Action-driven Learning of Structured Representations for Sequential Decision Making

By

Rafael Alberto Rodriguez Sanchez

B.S., Universidad Simon Bolivar, 2016

M.S., Politecnico di Milano, 2018

Thesis

Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Computer Science at Brown University

PROVIDENCE, RHODE ISLAND

May 2026

This dissertation by Rafael Alberto Rodriguez Sanchez is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

George Konidaris, Advisor

Recommended to the Graduate Council

Date _____ _____

Michael Littman, Reader

Date _____ _____

Ron Parr, Reader

Approved by the Graduate Council

Date _____ _____

David P. Lindstrom, Dean of the Graduate School

Curriculum Vitae

Rafael A. Rodriguez Sanchez grew up in Venezuela. He studied Electronics Engineering at Universidad Simon Bolivar, Caracas, Venezuela, and graduated *summa cum laude* in 2016 with a Bachelor's degree. He then transitioned to computer science, attending Politecnico di Milano, in Milan, Italy, on a scholarship from the Italian Ministry of Foreign Affairs and International Cooperation (MAECI), where he received a Master's degree *cum laude* in 2018 and developed his interest in Reinforcement Learning and Artificial Intelligence. Afterwards, he attended Brown University in Providence, Rhode Island. During his Ph.D, he interned at Amazon Alexa in Cambridge, Massachusetts in 2021. His research focuses on action-driven representation learning and structured world models for sequential decision-making.

To my parents

Acknowledgments

When I decided to do a Ph.D I did not *truly* know what I was getting myself into. It has been a long journey with many lows, many highs and a lot of soul-searching. And, though a Ph.D is, by nature, a solo journey deep into research and creativity, it definitely would have been impossible without the support of so many people I am lucky to have around.

I've never been the kind of person that has role models but I think, George, you have become one for me. I thank you for taking a chance on me, your curiosity and courage to pursue the hard problems truly inspired me all these years. Your optimism, support in the lowest moments, and our (almost always) encouraging weekly conversations were truly the foundation that made this dissertation possible.

Michael and Ron, thank you for being on my committee, and your support of this thesis. John and Lauren, thank you for making the bureaucracy bearable and possible.

My life in Providence wouldn't have been the same without many people that made this journey easier and worth it.

First, the golden boys in CIT243. Saket, thank you for always being around and motivating me to socialize, even at the moments I really needed it but did not want it. Akhil, for every drama you heard from me, every deep chat about life and science, I thank you. These truly made me happy and aware of how much I enjoyed science and research. Sam, my last roommate, thank you for all the support and always being the optimistic

soul we all needed around.

Anna, Anita, Ben, Alper, Cam, David P, and the IRL lab—thank you for every conversation we had during these past years. Thanks to my second floor friends: Ashley, Dominik, and Zhizheng.

Alessio, thanks for being a great roommate and friend, and for all those neat Diplomaticos we got to drink together at Preston St and elsewhere. Nedi, Gabby, Esen, your friendship made life in Providence (and NYC weekend escapades) so much fun.

Ji Won, thank you for your friendship all these years, since that moment we met in Preston St. Thank you for every time you unstuck us from CIT243 and made us go on mini adventures. The realization of this *delulu* thesis could not have happened without the daily support you gave me the last few months.

To the friends that have been cheering me on longer than I remember: Manu, Riccardo, Francesca, Jose Manuel, Arturo, and Andreina. Thank you for the words of encouragement and for receiving me in your homes every time I wanted to escape my Ph.D and Providence. All our trips together really replenished my good spirits to keep going.

Gaby, it is incredible to think that you have been supporting me since we went to those Math Olympiads when we were just little kids. Thank you for so much, for supporting me, for our trips, for hosting me when I was stranded without a visa. There are truly no words that can convey how much you helped through it all.

Jorge, thank you for your friendship and your support. Even the idea of a Ph.D at Brown would not have been possible without your support. I am grateful to you and Laura for all those weekend barbecues during your time in Boston. They got me through a lot of things.

Finally, mamá, papá, and Daniel. You have been supporting me since I was born and encouraging me to pursue every short-lived, crazy project, and long-term hyper-fixations that ended up with me pursuing this Ph.D. I am who I am today because of you. *Gracias.*

Abstract of *Action-driven Learning of Structured Representations for Sequential Decision Making*, by Rafael A. Rodriguez Sanchez, Brown University, May 2026

Generally intelligent agents must learn and adapt by interacting with a complex world. In order to be generally capable of performing diverse tasks in their lifetime they must perceive the world through rich, high-dimensional sensors and have access to adaptable, fine controls. This, however, makes the learning problem intractable. In order to learn and act efficiently, they must use abstractions of state and time: they have to focus only on the relevant information and reason at the right time scale.

Traditionally we have provided the problem formulation to our agents; implicitly giving them access to privileged knowledge about the abstractions and structure of the world. However, agents must be able to learn about these by themselves. In this thesis, we will focus on the problem of learning state representations directly from high-dimensional observations and show that agents' actions are the common thread, providing rich learning signal across two axes: abstraction and factorization.

First, we focus on state abstraction: the agent must learn a representation that contains only the relevant information for planning. Specifically, we propose an algorithm to learn minimal continuous representations that are sufficient for planning with skills and show empirically that the learned model can be reused effectively to plan for different tasks.

Second, we explore learning disentangled representations by discovering underlying factors of variation from raw observations: we introduce a contrastive algorithm that leverages the agent's actions to discover the independently controllable factors directly from pixels. That is, we leverage the agent's interventions in the dynamics of the world to uncover a signal that disentangles the controllable factors without any prior knowledge.

Next, we propose an approach to balance multiple sparsity conditions—action-effect sparsity and temporal-dependency sparsity—to recover the Dynamics Bayesian Network

(DBN) by showing that the disentangled representation is the Pareto-optimal solution of a cooperative game between multiple constraints over a shared encoder. Finally, we generalize this idea to mechanism shifts that arise naturally in dynamical systems in which contacts can create new relations in the DBN, providing additional structural signal for disentanglement.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The field of Artificial Intelligence (AI) has seen many breakthroughs since its inception, every time pushing the limits of computation. We have seen the development of agents capable of beating chess masters (Campbell et al., 2002), beat the humans at Go (Silver et al., 2016, 2017), Dota 2 and Starcraft (Vinyals et al., 2019), Poker (Brown and Sandholm, 2018, 2019), Diplomacy (Meta Fundamental AI Research Diplomacy Team (FAIR) et al., 2022) and, most recently, we have seen the birth of a new class of artificial agent, one trained with so much more knowledge than any human on its own can ever absorb (Large Language Models (Brown et al., 2020)) that has changed forever the way we work and interact with machines.

However, the physical world remains the biggest challenge that artificial agents have to face. Animals and humans are capable of learning to perform a wide variety of tasks in an ever-changing complex world during a lifetime. Yet, they do not nearly need as much computing power and massive amount of data of modern AI. However, this thesis will focus on another problem, the problem of growing a mind (Tenenbaum et al., 2011) of an interactive agent.

1.1 Embodied Agents and the Reinforcement Learning Problem

Formally, this thesis centers the embodied agent in an unknown world. The agent can act and interact with its environment through a set of actions and observe the world continually through its sensors. This agent is on its own out in the world. It has to learn from pure interaction how to maximize its own interests, represented by a reward function, that the agent must accumulate during its lifetime. This is the reinforcement learning problem (Sutton and Barto, 1998) and it captures very generally the problem we still have to solve.

This agent doesn't come with a manual to understand its world or what it's supposed to do. It is not pretrained on the whole information available in the world. It begins from scratch and has to grow to understand the environment, understand its actions, learn to communicate, play chess, Atari...

To build this agent, we need to provide fine actuators that are flexible enough to adapt to unknown tasks in the agent lifetime and we have to provide sensors that capture enough information to understand the necessary elements of any task. This makes a hard problem impossible. Learning to play chess while making decisions from images and controlling muscles is definitely the wrong approach.

Abstractions are a necessity (Konidaris, 2019) and the agent must learn through interactions how to build them. There are two types of abstractions:

Temporal Abstractions or Skills allow agents to deal with long-horizon problems by encapsulating directed control policies that can be composed to reason, learn and act more effectively. Skills are what allows us to think in chess moves and strategies instead of muscle control (Sutton et al., 1999; Barto and Mahadevan, 2003);

State Abstractions (Li et al., 2006; Abel et al., 2018) is the complement. It's the task

of compressing the observation into a summary of the information that is required for the agent decision process, it must be (1) compact and minimal to allow the agent to focus on what's important (e.g., to reason about chess moves, the agent might not need to think about the weather), and (2) it must reveal structural knowledge of the world to enable the agent better understanding of its world.

But, how does the agent learn these abstractions? Traditionally, we provide them: we formulate their tasks at the correct level of abstractions. For instance, we provide matrix forms of board states for Go and Chess that only contain features that relevant for playing and the actions are at the right temporal scale, we let them choose what piece to move and where. Our Atari playing agent was given only the images of the game but it did not have to learn how to use the joystick (Mnih et al., 2015). Even language models are given the right tokenization of language (Kudo and Richardson, 2018; Sennrich et al., 2016) that supports their learning and reasoning. Agents must learn all these by itself through interaction and they remain open challenges in Hierarchical RL's skill discovery and state representation learning. The latter is the focus of this dissertation.

1.2 Action as the driver of World Knowledge

Traditionally, the problem of state representation has been seen as the responsibility of unsupervised and self-supervised learning: approaches like VAEs (Kingma and Welling, 2013), contrastive learning (Gutmann and Hyvärinen, 2010; Oord et al., 2018a), object-oriented deep architectures (Locatello et al., 2020b; Greff et al., 2019), BYOL (Grill et al., 2020) and, more recently, diffusion models (Ho et al., 2020) have been ported to RL in the hopes of separating the state representation problem from that of behavior learning. In this dissertation, we take a different approach. Actions will be the drivers of representation learning and structure discovery.

Our agents do not learn from static datasets of the world and they are not merely

passive observers of the world. *They act.* And their actions and decisions are interventions in the causal structure of the world. A passive observer can only learn about correlations in their observations, an acting agent can gain grounded knowledge of the world, that is, the causal mechanisms that governs the world dynamics. This is what lies at the core of every chapter that follows.

1.3 Contributions

In Chapter 3, we explore the problem of factorization and disentanglement from the lens of control. When an agent receives high-dimensional observations, the relevant variables, those that we would provide if we hand-design them, are entangled together—e.g., the door state and positions are all represented in the segment of the image. We want the agent to discover those variables. This chapter introduces Action Controllable Factorization (ACF), a contrastive approach to disentangling the independently controllable factors. It leverages the fact that agent actions create interventional shifts with respect to the natural dynamics of the world as the factorization signal.

Chapter 4 addresses one limitation of ACF: not every important variable is independently controllable, but inter-factor dependencies are naturally sparse. Revealing these temporal relationships gives us two important things. First, it provides us an additional signal to further improve the latent factorization, and second, it pushes our world models to learn real relationship between the factors instead of capturing spurious correlation in the dynamics. Naive optimization of action effect sparsity and temporal sparsity leads to degenerate solutions. We introduce a novel cooperative framing to achieve factorization, Nash-ACF, that finds a Pareto efficient solution to learning the latent Dynamics Bayes Network (DBN).

In Chapter 5, we take a detour, and focus on hierarchical problem where the agent has access to temporally-extended skills and how these drive representation. Precisely, we

introduce a sufficient, and necessary, minimal state abstraction required for planning with the skills without simulating the low-level trajectories. We realize this via an information maximization algorithm, Contrastive Abstract Model for Planning (CAMP), effectively generalizing the Skills to Symbols framework (Konidaris et al., 2018; Ahmetoglu et al., 2025) to continuous latent states. The work presented in this chapter was previously published at RLC 2024 (Rodriguez-Sanchez and Konidaris, 2024).

Chapter 6 unifies our insights within one framework leveraging the concept of natural mechanism shifts in dynamical systems. Chapter 3 assumed that variables are directly controllable and Chapter 4 recovered the DBN but assumed that it held globally. Both assumptions break easily. This last chapter acknowledges this and introduces the natural mechanism shifts (NMS) as the missing interventional piece that allows factorization of indirectly controlled variable via transient couplings and temporally-extended behavior. This chapter introduces NMS-ACF, the multi-time extension of ACF.

Finally, Chapter 7 summarizes the insights developed in this dissertation and looks forward to what this entails for the reinforcement learning agent problem.

CHAPTER 2

Background

This chapter introduces the foundational mathematical formalisms, concepts, and tools used throughout this dissertation, and surveys the seminal works that shape the literature it builds on. We begin with the reinforcement learning problem and its classical formulation as a Markov decision process (Section 2.2). We then refine this formulation with factored structure, reviewing the line of work that exploits independence between state variables for planning and learning. Because general-purpose agents must act from high-dimensional sensors rather than enumerated states, we next discuss abstractions: the classical theory of how to compress state and time without losing what matters for decision-making (Section 2.4). Finally, we turn to representation learning (Section 2.5), the modern toolkit with which those abstractions are learned rather than designed. This section introduces the self-supervised and energy-based methods the thesis uses, formalizes identifiability as the property we want recovered representations to have, and reviews how representation learning has been deployed within RL.

2.1 Notation

Throughout this thesis we adopt the following conventions. Variables are denoted by lowercase italic letters and, unless stated otherwise, are vectors; the i-th component of a vector is indicated by a *superscript*, so s^i is the i-th factor of state s. Indexing over time uses a *subscript* t, so s_t is the state at time t and s_t^i its i-th factor. We write $s' := s_{t+1}$ for the next-step state when no ambiguity arises. Random variables are uppercase italic (S, A, Z). Sets, function classes, and aggregate quantities (losses, graphs) are calligraphic uppercase ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{F}$, $\mathcal{L}$, $\mathcal{G}$); numeric cardinalities such as the number of factors K and the option duration κ are plain. Dimensions of spaces carry a subscripted d (d_s, d_a, d_x, d_z).

We reserve ϕ for the parameter vector of the encoder and θ for the parameter vector of the world model (energies, transition, reward). Functions are subscripted by their parameters: f_ϕ is the learned encoder, E_θ a (per-factor) energy, π_w a policy with parameters w. Learned or estimated quantities carry a hat ($\widehat{T}, \widehat{R}$), abstract counterparts carry a bar ($\bar{s}, \bar{\mathcal{S}}, \bar{M}$), and rotated or transformed quantities carry a tilde ($\tilde{z}, \tilde{H}$). Loss functions come in two forms: pointwise losses evaluated at a single sample are denoted by lowercase script ℓ_{name}, and their expectations —the quantities actually optimized—by $\mathcal{L}_{\text{name}} := \mathbb{E}[\ell_{\text{name}}]$. Loss weights are written λ_{name} throughout.

The unknown observation (mixing) function from the ground-truth factor space to observations is $g_\star : \mathcal{S} \to \mathcal{X}$; the star distinguishes it from any learned decoder. Options are denoted $o \in \mathcal{O}$, each specified by an initiation set $\mathcal{I} \subseteq \mathcal{S}$, an internal policy π, and a termination probability β; κ denotes an option's expected execution time and, in Chapter 6, a multi-step horizon. The inverse-dynamics classifier under policy π is $\iota^\pi(a \mid z, z')$. The parent function in a dynamic Bayesian network is $\mathrm{pa}(s^i)$ by default; when the graph is indexed by an action or a phase we write $\mathrm{pa}^a(s^i)$ or $\mathrm{pa}^\varphi(s^i)$, respectively. Phases are denoted $\varphi \in \Phi$, and the set of factors whose mechanism changes across a phase transition is $\mathcal{J}(\varphi_0, \varphi_1) \subseteq [K]$. The score of the transition is $\psi(s', s, a) := \nabla_{s'} \log T(s' \mid s, a)$; its interventional difference relative to the observational action a_0 is

$\Delta^a\psi(s', s) := \psi(s', s, a) - \psi(s', s, a_0)$, and in the continuous-action setting of Chapter 6 we use the action-Jacobian J^a_ψ with $[J^a_\psi]^{ij} = \partial\psi^i/\partial a^j$.

2.2 The Reinforcement Learning Problem

This thesis is concerned with the problem of the generally-intelligent agent that interacts with its environment in a continuous loop of interaction. This agent must learn about the world, how it works and how to act accordingly to achieve its goals. The reinforcement learning (RL) problem (Sutton and Barto, 1998) formalizes this agent concisely as an agent that makes decisions sequentially to maximize a scalar reward function that represents its goals. This formalization is driven by the reward hypothesis (Sutton and Barto, 1998; Silver et al., 2021) that posits that a scalar signal is sufficient to represent any goals the agent must learn to achieve. In this section, we introduce the formalization of these concepts.

2.2.1 The Agent and its World: Markov Decision Processes

A Markov decision process (MDP; Bellman, 1954; Puterman, 1994) is a fundamental model of discrete-time, sequential decision-making and lies at the heart of the RL formulation. Let $\Delta(\mathcal{X})$ represent the set of probability density functions (p.d.f.) over a set $\mathcal{X}$. An MDP M is defined as a tuple $M = (\mathcal{S}, \mathcal{A}, T, R, \gamma, p_0)$ where:

- $\mathcal{S}$ is the set of possible states of the world. For the rest of this thesis, we will consider that $\mathcal{S} \subseteq \mathbb{R}^{d_s}$ for some dimension $d_s \in \mathbb{N}$;
- $\mathcal{A}$ is the set of actions the agent can take to affect the world. This set can be discrete set of possible actions—e.g., the buttons in a joystick—or, it could be $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ for some $d_a \in \mathbb{N}$—e.g., the control signals in dynamical systems;
- $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function and it models the dynamics of the world, by giving a probability density function (p.d.f.) over the possible next states

s' when action $a \in \mathcal{A}$ is executed in state $s \in \mathcal{S}$;

- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the scalar reward function the agent must maximize;
- γ is the discount factor;
- and, $p_0 \in \Delta(\mathcal{S})$ is the initial state p.d.f. or distribution.

2.2.2 Policies and Value Functions

A solution to an MDP is a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that prescribes a behavior that maximizes the agent's expected *return* U. There are multiple definitions of return in the literature. However, in this thesis we will focus in the most common case: the discounted sum of rewards.

$$U = \sum_{t=0}^{\infty} \gamma^t R_t \tag{2.1}$$

where the discount factor γ ensures convergence of the series and can interpreted as geometrically encoding a preference for more recent reward than rewards far into the future.

Policy Evaluation Given a policy π, we can *evaluate* its goodness with respect to the reward function by computing the expected return when the agent follows the policy. Formally, we define the **value function** as $v^\pi : \mathcal{S} \to \mathbb{R}$ for policy π as

$$v^\pi(s) = \mathbb{E}\left[U \mid \pi, T, S_0 = s\right]. \tag{2.2}$$

Therefore, the agent is concerned with learning the optimal policy π^* that maximizes the expected return, that is,

$$\pi^*(s) \in \arg\max_{\pi \in \Pi} v^\pi(s). \tag{2.3}$$

Similarly, we can define an **action-value function** [1] $q^\pi : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ that is defined as the expected value of taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and the following policy π. Formally,

$$q^\pi(s, a) := R(s, a) + \gamma \mathbb{E}[v^\pi(s')] \tag{2.4}$$

Policy Improvement Interestingly, if we have access to q^π, then we can actually compute a **better** policy π':

$$\pi'(s) := \arg\max_{a \in \mathcal{A}} q^\pi(s, a); \tag{2.5}$$

and it is guaranteed by the policy improvement theorem that $q^{\pi'}(s, a) \geq q^\pi(s, a)$ (Puterman, 1994).

The Bellman equation Therefore, the remaining question is how can we compute the value function for a policy π in order to improve it. The value function v^π must satisfy the following recursion, known as the Bellman equation,

$$v^\pi(s) = \mathbb{E}\left[R(s, a) + \gamma v^\pi(s') \mid a \sim \pi, s' \sim T, S_0 = s\right]. \tag{2.6}$$

Value Iteration The value function has a very interesting property. For instance, if we were to start with an estimate V_0 of v^π and, iteratively, use the Bellman equation to update our estimate:

[1] We will use Q-function and action-value function interchangeably for the rest of this thesis.

$$V_{n+1}(s) \leftarrow \mathbb{E}\left[R(s,a) + \gamma V_n(s') \mid \pi, T, S_0 = s\right] \ \forall s \in \mathcal{S}; \tag{2.7}$$

we are guaranteed to converge to v^π when $n \to \infty$. In fact, this algorithm is known as value iteration, a foundational algorithm for MDP planning.

The Bellman optimality equation establishes a similar recursive relationship for the optimal value function

$$q^*(s,a) = R(s,a) + \gamma \mathbb{E}\left[\max_{a' \in \mathcal{A}} q^*(s,a') \mid T, S_0 = s, A_0 = a\right] \tag{2.8}$$

Similarly, this holds for the value function

$$v^*(s) = \max_{a \in \mathcal{A}} \mathbb{E}\left[R(s,a) + \gamma v^*(s') \mid T, S_0 = s\right]. \tag{2.9}$$

2.2.3 Model-free RL

Without a model, we cannot fully compute the value iteration update. Instead, we can estimate the value function directly from data generated from interacting with the environment by taking an action following policy π and getting the experience tuple (s, a, r, s')

TD Learning estimates the value function of a policy π, V^π by leveraging the following update rule:

$$V^\pi_{n+1}(s) \leftarrow V^\pi_n(s) + \alpha\left(r + \gamma V^\pi_n(s') - V^\pi_n(s)\right), \tag{2.10}$$

where α is a step size and the difference $\delta := r + \gamma V^\pi_n(s') - V^\pi_n(s)$ is called the TD (Temporal Difference; Sutton and Barto, 1998) error. This update rule leverages previous

estimates of the value function V_n^π to compute a new estimate, that is, it bootstraps from its own estimates. This update rule is at the core of both classical and modern deep RL algorithms.

Q-Learning estimates analogously the Q-function from experience using an update rule based on the Bellman optimality equation to estimate the optimal Q-function. Let (s, a, r, s') obtained by interacting with the environment with any policy π, then the Q-Learning rule (Watkins and Dayan, 1992) is

$$Q_{n+1}(s,a) \leftarrow Q_n(s,a) + \alpha \left(r + \gamma \max_{a' \in \mathcal{A}} Q_n(s,a') - Q_n(s,a) \right), \tag{2.11}$$

where α is a step size and the error $\delta := r + \gamma \max_{a' \in \mathcal{A}} Q_n(s,a') - Q_n(s,a)$ is the TD error.

Modern algorithms such as DQN (Deep Q Networks; Mnih et al., 2015) and SAC (Soft Actor-Critic; Haarnoja et al., 2018) are based on this rule.

2.2.4 Model-based RL

Instead of directly estimating value from interactions, an agent can estimate a *world model*, that is, an approximate transition and reward functions $(\hat{T}, \hat{R})$ directly from interaction tuples (s, a, s', r), and use this approximate model to plan. Model-based RL (MBRL; Sutton, 1991) does exactly this. This approach effectively separates the problem of learning about the world dynamics from the problem of learning an optimal policy for a given reward function. Therefore, the world model can be re-used to learn multiple policies with the same understanding of the dynamics. In principle, this promises more sample-efficient learning.

The Dyna architecture (Sutton, 1991) instantiates this by interleaving dynamics estimation with value-based planning. A long line of seminal work studies the theoretical

properties and efficiency of this type of approaches (Brafman and Tennenholtz, 2002; Kearns and Koller, 1999). PILCO (Deisenroth and Rasmussen, 2011) scaled this idea to continuous control using Gaussian processes and showed that sample-efficiency improved by order of magnitudes over model-free baselines. However, model-based approaches suffer significantly from compounding errors; a problem that is exacerbated when the model is learned directly in the space of high-dimensional observations the agent receives. Therefore, scaling model-based RL is an active area of research.

The modern deep MBRL algorithms leverage deep neural networks and state-of-the-art generative modeling to scale MBRL agents and leverage different measure of uncertainty to tame the compounding error problem (Chua et al., 2018; Nagabandi et al., 2018; Janner et al., 2019). However, as the dimensionality of the observation space increases, more capacity is wasted predicting unnecessary information (e.g., the shadows in an image, the background evolution), exacerbating the compounding error problem and the generalization capabilities of these models.

Hence, learning compact state representations has also been important for model-based RL. Learning models in a more compact state space improves performance by enabling simpler models, reducing computational cost of planning (models do not need to generate full observations), and abstract away irrelevant details from the model.

VAEs (Kingma and Welling, 2013) have been used to compress the observations to latent states that can be used for model learning (Ha and Schmidhuber, 2018). However, these approaches do not take into account the temporal nature of the data. In contrast, there have been VAE-based losses that do take the temporal properties of the data by deriving ELBO bounds to estimate the transition dynamics (Krishnan et al., 2015; Gregor et al., 2018; Buesing et al., 2018; Zhang et al., 2019).

Dreamer V1 (Hafner et al., 2019), V2 (Hafner et al., 2021b), and V3 (Hafner et al., 2025) leverage this kind of formulation to learn recurrent transition functions both with

continuous and discrete latent spaces.

MuZero (Schrittwieser et al., 2020) learns a latent representation that is sufficient for reward and value prediction and learns a forward model over that representation. Subsequently, EfficientZeroV1 and V2 (Ye et al., 2021; Wang et al., 2024a) build on top of MuZero by improving the representations by introducing a self-prediction loss that ensures the next latent state is predictable from the current latent state and action. Similarly, TD-MPC (Hansen et al., 2023, 2024) introduce a model-based approach for continuous control that leverages representations learned without reconstructions that are self-predictive and are value-preserving.

State-of-the-art MBRL shows that the world model can be very useful, if the representation is good.

2.3 Factored Reinforcement Learning

MDPs do not assume any type structure in the world. The state is treated as an atomic unit. However, the physical world has natural decomposition that capture the causal structure of the world. For instance, not every object dynamics depend on every other in the world, we need a small subset of knowledge to understand an object dynamics.

This structure is quite appealing for many reasons. First, and the most obvious, is the size of the state space and the planning complexity. The larger the state space the more expensive is to plan. Think about it this way, if we have n binary state variables, then we have a state that grows exponentially, 2^n, with the number of variables. However, not every possible state carry new information about a given variable. For instance, the dynamics of an object free falling is independent of the time of day. Hence, an agent does not need to see the ball fall at every possible time of day. The agent can *generalize* to every situation requiring less experience to learn effectively. Indeed, this is the insight that underlies RL algorithms in factored settings. Moreover, this enables transferability

and fast adaptation. The ball dynamics do not depend on every variable describing the room the agent is in, thus, changing rooms will not change the ball dynamical model.

In this section, we formalize the model we need to exploit this structure. Factored Markov decision processes (FMDP; Boutilier and Dearden, 1996). This will be the core formalism we will be using throughout this dissertation.

2.3.1 Factored Markov Decision Processes

Consider that the state space is a set such that $\mathcal{S} = \mathcal{S}^1 \times \cdots \times \mathcal{S}^K$ where for all $\mathcal{S}^i \subseteq \mathbb{R}^{d_i}$ for some dimension $d_i \in \mathbb{N}$ and they represent *factors* of the state. FMDPs have a factorized transition function that can be represented graphically by a dynamic Bayesian network (DBN; Boutilier et al., 1995). It is called dynamic, because it shows the relation between variables in a one-timestep slice.

A DBN is a bipartite directed acyclic graph $\mathcal{G}_z = (\mathcal{V}, \mathcal{E})$ where the two disjoint set of nodes corresponds to $\mathcal{V}_t = \{S_t^i\}_{i=1}^K$ and $\mathcal{V}_{t+1} = \{S_{t+1}^i\}_{i=1}^K$ that are the random variables representing factors at time t and $t+1$, respectively. Edges $\mathcal{E}$ in this network represent temporal dependencies between factors. Moreover, given a FMDP with discrete actions, we consider that for each action $a \in \mathcal{A}$ we have a corresponding DBN $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{E}_a)$ where $\mathcal{V}_a = \{A^i\}_{i=1}^{d_a}$ are an element of action (continuous vector if action is continuous or one-hot vector if actions are discrete). We will use adjacency matrices to represent these graphs throughout this thesis: $G_z \in \{0,1\}^{K \times K}$ and $G_a \in \{0,1\}^{K \times d_a}$ for $\mathcal{G}_z$ and $\mathcal{G}_a$ respectively.

Let $\mathrm{Pa}^{\mathcal{G}} : \mathcal{V} \to \mathcal{V}$ be the parent function that takes a factor (a node of $\mathcal{G}$) and returns the set of parents of the factors in G. To avoid clutter, whenever unambiguous, we will use a and z as subscripts of pa to indicate parents of $\mathcal{G}_a$ and $\mathcal{G}_z$ Thus, the transition function can be written as

$$T(s_{t+1} \mid s_t, a_t) = \prod_{i=1}^{K} T(s_{t+1}^i \mid \mathrm{pa}^{z,a}\big(s_{t+1}^i\big)). \tag{2.12}$$

This factorization of the dynamics is why the s_is are called factors. These are the

variable where the independence relations hold.

2.3.2 Factored Planning and Learning

Naturally, there is a long line of work that used FMDPs for more efficient planning by leveraging the ideas value iteration and policy iteration together with the structure encoded in the DBN: *Structured Value Iteration* (Boutilier et al., 1995, 2000) and *Structured Policy Iteration* (Boutilier et al., 1995; Koller and Parr, 2000) extended the core algorithms to the factored, tabular setting. Guestrin et al. (2003) extended these ideas to use linear function approximation to estimate the structured value functions.

In the context of RL, there has been work that leverages the knowledge of the DBN and access to a factored planner to design efficient PAC RL algorithms such as factored E^3 algorithm (Kearns and Singh, 2002). Similarly, a factored RMax algorithm (Guestrin et al., 2002) was proposed by leveraging factored linear value iteration (Guestrin et al., 2003). Further theoretical work includes regret bounds for factored RL in the episodic (Osband and Van Roy, 2014; Tian et al., 2020) and non-episodic settings (Xu and Tewari, 2020). This classical line of work showed theoretically that if we had access to the DBN, we would have an exponential gain in sample efficiency.

The main limitation of these works, though, is assuming access to the DBN itself. Therefore, the natural continuation was to estimate the DBN directly from data. Thus, algorithms such as SLF-RMax (Strehl et al., 2007), Met-RMax (Diuk et al., 2009) and SPITI (Degris et al., 2006) relax this assumption and learn the structure, the DBN edges, directly from data. Vigorito and Barto (2009) extends the structure discovery to continuous state-action spaces.

Moreover, the notion of factors have been leveraged for other purposes such exploration (Seitzer et al., 2021; Wang et al., 2023), skill discovery (Vigorito and Barto, 2010; Hu et al., 2024; Wang et al., 2024b; Chuck et al., 2024, 2025), learning modular world model that are robust (Wang et al., 2022; Ke et al., 2021), non-parametric experience generation

by using the DBN structure to recombine factor values (Pitis et al., 2020, 2022).

Nonetheless, all these approaches assume that we have access to the variables where the DBN can be learned. Unfortunately, this is almost never true. We want to our agents to learn directly from their high-dimensional observations, where the DBN nodes—i.e., the factors—are not directly measured and must be inferred from the observation. This dissertation is concerned with the problem of discovering *those* variables, where the structural dependencies are effectively revealed.

2.4 Abstractions

The general purpose learning agents is meant to learn a wide variety of tasks during its lifetime. Therefore, it needs to have information-rich sensors and fine-grained control signals. Such is the case of an autonomous robot that sees the world from cameras and LIDARs, and control its joints via 1KHz control frequency. Learning to play chess through pixels and high-frequency control signal is impossible. Therefore, abstractions are a necessity (Konidaris, 2019).

In this section, we define the two types of abstractions: temporal abstractions and state abstractions. These sit at the core of hierarchical RL (Barto and Mahadevan, 2003; Dietterich, 2000; Parr and Russell, 1997) and this dissertation.

2.4.1 Temporal Abstraction and the Options framework

The further a reward is into the future the harder it is to estimate the value of an action now. This issue is pervasive in RL where the long-term credit assignment problem makes value estimation intractable very quickly. In fact, recall the problem of playing chess with a robot that has to make high-frequency decisions about the torques to apply to its joints. It is clear that this is not the right way to learn how to play chess. However, it is a valid and real setting for a general-purpose agent. Moreover, it is clear that the

natural decisions the robot should be at the higher level of piece moves. This idea is formalized by hierarchical RL (HRL) and temporal abstractions (Barto and Mahadevan, 2003).

The options framework (Sutton et al., 1999) formalizes this notion of high-level actions as *options* that are formally defined as a triple (I, π, β) where $I \subseteq \mathcal{S}$ is the initiation set, the set of states where the option can start executing, $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is the option's closed-loop policy, and $\beta : \mathcal{S} \to [0, 1]$ is the termination condition function that gives the probability of the terminating the option's execution at a given state.

To plan with a set of options we require an *option model* that describes the long-term effects of executing an option. The option model for an option o is defined as the tuple (R_o, P_o) where R_o is the expected discounted reward accumulated during the option execution from state s_t until termination, i.e.,

$$R_o(s_t) = \mathbb{E}\left[\sum_{\tau=t}^{T} \gamma^{\tau-t} r_\tau\right], \tag{2.13}$$

and P_o represents the discounted probability of ending in state s' when executing option o from state s_t, i.e.,

$$P_o(s' \mid s) = \sum_{k=1}^{\infty} \gamma^k p(s', k \mid s, \pi_o), \tag{2.14}$$

where $p(s', k \mid s, \pi_o)$ is the probability of ending in s' after k steps following the option's policy (Sutton et al., 1999). This option model is known as the Multi-time model of options (MTM). A simpler option model is the expected-length model of options (Abel et al., 2019). This model simplifies MTM by modeling the option's length, or duration of execution, independently from the next-state distribution. We will use this approximation in Chapter 5 to build abstract models automatically.

2.4.2 State Abstractions and Representations

The dual abstraction is state abstraction.

The state space of an MDP can become intractably complex. In fact, general-purpose agents—e.g., robots, must perceive the world from sensors that provide high-dimensional signals. However, MDP planning and learning scales poorly with the size of the state space (Kearns and Koller, 1999; Brafman and Tennenholtz, 2002); tabular RL methods become impractical quickly. State abstractions aim to alleviate the problem by noticing that grouping *similar* states effectively reduce the overall size of the state space leading to gains in computational complexity and generalization.

Formally, state abstractions (Sutton and Barto, 1998; Li et al., 2006; Abel et al., 2016, 2020) are *non-injective* functions $f : \mathcal{S} \to \bar{\mathcal{S}}$ that map *ground* state $s \in \mathcal{S}$ to a new abstract state $\bar{s} \in \bar{\mathcal{S}}$. These functions are necessarily lossy; the ground state cannot be fully recovered from the abstract state. This is a desired characteristic because by only focusing in the relevant information contained in the state the agent can learn more efficiently. However, if the abstraction filters out relevant information for the task, it can hurt the final performance.

More recently, deep reinforcement learning (Tesauro et al., 1995; Lillicrap et al., 2016; Haarnoja et al., 2018; Schulman et al., 2017) alleviated the problem of having to hand-design state abstractions (Sutton, 1995; Konidaris and Osentoski, 2009; Mahadevan and Maggioni, 2007) by learning end-to-end from raw observations to actions.

However, it has become clear that state abstractions are important and *representation learning* has become an active research field within deep RL (Jaderberg et al., 2016). In this thesis, we will take a comprehensive approach to representation learning and we develop methods to learn deep latent representations that are both abstract and structured for RL.

In the next few sections, we will introduce the most relevant concepts in state repre-

sentation learning in RL that will be important in the development of this thesis.

Bisimulation and Model-preserving Abstractions

A bisimulation relation is an equivalence relation, defined as follows:

Let $s_1, s_2 \in \mathcal{S}$ be two different states and let B be an equivalence relation. Moreover, let $\mathcal{S}/\equiv_B$ be an element of the quotient, the partition of the set into equivalent classes $\bar{s}$. Moreover, we refer to a particular state in an equivalent class as $s \in \bar{s}$.

$s_1 \equiv_B s_2$ if and only if it the following holds:

1. $R(s_1, a) = R(s_2, a) \;\; \forall a \in \mathcal{A}$;
2. $\sum_{s' \in \bar{s}} P(s' \mid s_1, a) = \sum_{s' \in \bar{s}} P(s' \mid s_2, a) \;\; \forall a \in A, \bar{s} \in \mathcal{S}/\equiv_B$

The abstraction defined by the quotient set defined by the bisimulation relations are also known as *model-preserving* state abstractions (Li et al., 2006).

However, because bisimulation relations are defined by exact equivalences they are not amenable to optimization and they are sensitive in practice to approximate abstractions. Therefore, bisimulation metrics soften these relations (Ferns et al., 2004, 2011; Castro and Precup, 2010) by introduce a pseudo-metric space based on a distance function defined over the rewards and transition functions, that measures how close two states are from being bisimilar.

$$d(s_i, s_j) = \max_{a \in \mathcal{A}} (1 - c)|R(s_i, a) - R(s_j, a)| + cW_1(p(\cdot \mid s_i, a), p(\cdot \mid s_j, a); d), \tag{2.15}$$

where $c \in [0, 1)$ and $W_1(\cdot, \cdot; d)$ is the 1st Wasserstein distance using the distance function d.

Both bisimulation metrics and bisimulation relations have been used to learn represen-

tations in RL. Deep Bisimulation for Control (DBC;(Zhang et al., 2020)) used Equation 2.15 to learn a reconstruction-free representation. Bisimulation relation conditions are leveraged by DeepMDP (Gelada et al., 2019) and Markov state abstractions (Allen et al., 2021). Finally, the following two sections will leverage more of the model-preservation (bisimulation) conditions to learn representations while leveraging techniques developed in self-supervised learning.

Contrastive Representations in RL

Contrastive approaches have been extensively applied to self-supervised learning and RL. This is because contrastive learning allows learning representations without the need for explicitly reconstructing the observations without getting collapsed representations that lose too much information to make them impractical.

Contrastive representations for RL (CURL; Laskin et al., 2020) introduced an auxiliary task based using InfoNCE to learn representations directly from pixels by creating negative samples and positive samples by random image perturbations (e.g., noise, color jitter, cropping, translations). Therefore, CURL requires some domain knowledge about the observations for the algorithm to work.

Temporal predictive coding (TPC; Nguyen et al., 2021) modifies InfoNCE to learn representations that makes a transition predictable from the current latent state. In this way, it guarantees Markovianity with respect the transition dynamics. In changes the perspective with respect to CURL because it uses the temporal nature of the data to generate negative samples (fake transitions) instead of domain knowledge about the observations.

Contrastive learning has also been used for goal-conditioned RL, where the problem of estimating goal-conditioned value function is reframed as leveraging NCE or InfoNCE to estimate the discounted probability of future states (Eysenbach et al., 2021, 2022; Zheng et al., 2024).

Self-predictive Representations in RL

Although contrastive learning approaches work in practice, they can become impractical because mining for negative samples can become a complex task. Moreover, large batch sizes are required to improve losses like InfoNCE.

Therefore, a new class of representations, self-predictive representations, was proposed. These are mostly inspired by *Bootstrap Your Own Latent* (BYOL; Grill et al., 2020), a self-supervised framework that differentiates itself from contrastive learning by only needing positive samples.

Naively, these could cause the latent representation to collapse: there's no learning signal to learn the difference between samples. However, BYOL introduced the idea of using a previous version of the encoder (in practice an exponential moving average of the encoder) to generate a prediction target. The method consists in generating augmented views of a positive sample (e.g., random crops, color jitters, random noise) and predict from its encoding the latent representation of the original view. Hence, the learning proceeds by improving the representation by leveraging a previous version of the encoding: *bootstrapping* the previous encoding. This should encourage the network to learn transformation-invariant representations without the need of negative samples.

In a similar vein, self-predictive representations (SPR; (Schwarzer et al., 2021)) are the result of an auxiliary task that leverages this BYOL approach to improve representations for both model-free (Schwarzer et al., 2021) and model-based approaches (Ye et al., 2021; Hansen et al., 2023). However, instead of learning to predict a different view of the sample datum, it must predict the next latent state from the current latent state and action. In fact, this method encourages the network to learn a latent representation that is sufficient to predict the effect of the action; encouraging the representation to be Markov.

Probabilistic State Abstractions

So far, we have reviewed state abstractions that are formally non-injective *functions*. That is, these abstractions aggregate states into abstract states and no two states can belong to different abstract states.

A more general class of state abstractions was introduced in the context of classical planning with PDDL (Younes and Littman, 2004) in the framework of **Skills to Symbols** (S2S; Konidaris et al., 2018). This work takes a constructive approach to state abstractions and builds the minimal set of *symbols* that are required to plan with a set of skills. This work effectively bridged classical symbolic planning with the options framework (Sutton et al., 1999). In this section, we briefly introduce one of the fundamental innovation in the skills to symbols framework: probabilistic state abstractions.

Let $\bar{\mathcal{S}}$ be a set of abstract states, or symbols in the S2S framework, and, as usual, $\mathcal{S}$ the *ground* state space. Instead of partitioning the state space $\mathcal{S}$, as usual abstractions do, S2S introduces a grounding operator $\mathcal{G}$ that given an abstract state $\bar{s}$ would return a probability density function over the state space: $\mathcal{G}(\bar{s}) \in \Delta(\mathcal{S})$. This defines a more expressive family of abstractions: now a state s can belong to more than one abstract state $\bar{s}$ and its grounding $\mathcal{G}(\bar{s})$ fully represents the weight each ground state s contributes to the abstract state. This grounding approach to abstract state naturally comes from the *necessity* to compute the probability of success of a plan; without probabilistic groundings there would be no way to determine how likely is to be in a ground state s when only planning at the abstract level.

In this thesis, we will take this probabilistic approach to define abstract models driven by temporally-extended actions in Chapter 5.

2.5 Representation Learning

This section will introduce the most relevant techniques for learning representations in self-supervised learning (SSL;Balestriero et al. (2023))—learning generalizable representations by leveraging prediction tasks with unlabeled data—, and unsupervised learning—learning density models, generative models directly from unlabeled data. That will serve as the toolbox that we will use to develop the methods in the following chapters.

2.5.1 Variational Autoencoders

Autoencoders (Rumelhart et al., 1988; Hinton and Salakhutdinov, 2006) are the simplest deep learning architecture for learning to represent data using a deep embedding vector by leveraging an information bottleneck.

An autoencoder is composed of two neural networks with parameters (θ, ϕ): an encoder $f_\theta : \mathcal{X} \to \mathcal{Z}$ and a decoder $g_\phi : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{X}$ and $\mathcal{Z} \subseteq \mathbb{R}^n$ are the observation space, e.g., pixels, and the latent space, respectively, that correspond to n-dimensional real vectors.

These functions are trained to learn a lower-dimensional encoding that allows the decoder to reconstruct the observation by minimizing the mean square error loss (MSE).

$$\min_{\theta,\phi} \mathbb{E}_x \|x - g_\phi(f_\theta(x))\|_2^2 \tag{2.16}$$

Variational Autoencoders (VAEs) (Kingma and Welling, 2014) improve upon the vanilla autoencoder by reformulating the problem as a latent-variable model from a Bayesian perspective. Instead of mapping each input x deterministically to a single point in the latent space, VAEs assume that each input is generated from a distribution over latent variables z. Formally, we consider the generative process

$$z \sim p_\theta(z), \quad x \sim p_\theta(x \mid z), \tag{2.17}$$

where $p_\theta(z)$ is typically chosen as a standard Gaussian prior $\mathcal{N}(0, I)$, and $p_\theta(x \mid z)$ is the likelihood parameterized by a neural network (the decoder).

Since the true posterior $p_\theta(z \mid x)$ is intractable, VAEs introduce an approximate inference network, or encoder, $q_\phi(z \mid x)$, which is also modeled as a Gaussian with mean and variance predicted by a neural network. Learning is performed by maximizing the *evidence lower bound* (ELBO):

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x \mid z)\big] - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p_\theta(z)), \tag{2.18}$$

which balances two terms:

- a **reconstruction term**, encouraging the decoder to faithfully reconstruct x from latent samples z, and
- a **regularization term**, encouraging the approximate posterior $q_\phi(z \mid x)$ to stay close to the prior $p_\theta(z)$.

VAEs and variations lie at the core of many algorithms, because it allows to learn an effective lower-dimensional embedding that contains sufficient information from the observations to be reused for other downstream tasks. As we will see in the next section, VAEs also lie at the heart of representation learning in RL and proven to be useful in many state of the art model-based deep RL algorithms.

2.5.2 Contrastive Representation Learning

Autoencoders rely heavily in a reconstruction task and the information bottleneck, i.e., the dimensionality of the embedding vector, to compress the observation. However, the

information represented in the representation might not align well with the downstream task. That is, the information required to reconstruct might not be the information required for another task. Moreover, in practice, training decoders are computationally expensive, and VAEs are plagued with mode-collapse problems.

Contrastive methods (Hadsell et al., 2006; Gutmann and Hyvärinen, 2010; Oord et al., 2018a; Lee et al., 2021; Laskin et al., 2020) learn useful representations without decoders and reconstructions. It is based on the simple idea that useful representations must capture the important features of the observation as opposed to irrelevant variations such as noise and color jitters.

Consider a dataset of samples $\mathcal{D} = \{x_i\}$. For each sample x_i, contrastive methods generate a negative sample x_i^- and then train a classifier to discriminate between the negative and positive samples. This is known as Noise Contrastive Estimation (NCE; Gutmann and Hyvärinen, 2010). Formally, we consider that we have access to a noise distribution $p(x^-)$ from which we can sample negative samples and the real data distribution is $p(x)$ and can sample a dataset $\mathcal{D}^- = \{x_i^-\}$ of negative samples. Negative samples can be generated by applying transformations to the observations that do not modify the actual generating factors such as Gaussian noise, color jitters, random cropping and translations, etc.

Moreover, we consider the encoder $f_\phi : \mathcal{X} \to \mathcal{Z} \subseteq \mathbb{R}^n$. NCE trains a discriminative network $g_\theta : \mathcal{Z} \to [0, 1]$ that learns to predict whether a sample is a real sample from the latent representation:

$$\min_{\theta,\phi} \frac{1}{|\mathcal{D}| + |\mathcal{D}^-|} \sum_{x \in \mathcal{D} \cup \mathcal{D}^-} -\log \sigma(g_\theta(f_\phi(x))) \tag{2.19}$$

where σ is the sigmoid function.

An important extension of this contrastive idea is InfoNCE (Oord et al., 2018a). In this

case, the task is to learn a representation that can be predicted from a context variable c. This context variable can be an aggregation of past observation in time series, or information about neighboring patches or pixels in an image.

In practice, training proceeds by creating a batch of samples with $n-1$ negative samples and 1 positive sample.

Let $g_\phi : \mathcal{Z} \times \mathcal{C} \to \mathbb{R}$, hence we minimize

$$\min_{\theta,\phi} \frac{1}{|\mathcal{D}|} \sum_{x^+ \in \mathcal{D}} -\log \frac{e^{g_\phi(f_\theta(x^+),c)}}{\frac{1}{N}\left(e^{g_\phi(f_\theta(x^+),c)} + \sum_{x^-} e^{g_\phi(f_\theta(x^-),c)}\right)}. \tag{2.20}$$

Importantly, the InfoNCE estimates the mutual information between the context variable and the observation (Poole et al., 2019).

$$\mathbb{E}\left[\frac{1}{N}\log \frac{e^{g_\phi(f_\theta(x^+),c)}}{\frac{1}{N}\left(e^{g_\phi(f_\theta(x^+),c)} + \sum_{x^-} e^{g_\phi(f_\theta(x^-),c)}\right)}\right] \leq I(X;C) \leq \log N \tag{2.21}$$

2.5.3 Energy-Based Models

The contrastive estimators of the previous section belong to a broader family of models that we will use throughout this thesis: *energy-based models* (EBMs). An EBM defines an unnormalized probability density over a space $\mathcal{X}$ through an energy function $E_\theta : \mathcal{X} \to \mathbb{R}$,

$$p_\theta(x) \;=\; \frac{\exp(-E_\theta(x))}{Z(\theta)}, \qquad Z(\theta) \;=\; \int_{\mathcal{X}} \exp(-E_\theta(x'))\, dx', \tag{2.22}$$

where $Z(\theta)$ is the partition function that ensures the density integrates to one (LeCun et al., 2006; Hinton, 2002).

EBMs are attractive because any real-valued function—in particular, any neural network—can serve as an energy. This is a meaningful departure from classical parametric density models, where the architecture must encode normalizability: Gaussians, mixture

models (Bishop and Nasrabadi, 2006), autoregressive factorizations (Larochelle and Murray, 2011; Germain et al., 2015; Van Den Oord et al., 2016), and normalizing flows (Rezende and Mohamed, 2015; Dinh et al., 2017) all impose structural constraints on the function approximator so that Z is tractable by construction. EBMs drop this constraint and let the modeller choose any architecture that is convenient for the data. The cost is that $Z(\theta)$ becomes intractable for arbitrary E_θ, and both training and sampling must avoid computing it.

Training Much of the recent progress in deep generative modeling can be understood as scalable algorithms to train EBMs without evaluating Z. Noise-contrastive estimation (Gutmann and Hyvärinen, 2010) and its information-theoretic variant InfoNCE (Oord et al., 2018a), introduced in the previous Section 2.5.2, are EBM estimators: they recast density estimation as a classification problem between samples from the model and samples from a noise distribution, which cancels Z in the decision ratio. Score matching (Hyvärinen and Dayan, 2005; Song and Ermon, 2019) takes a different route, fitting the gradient of the log-density $\nabla_x \log p_\theta(x) = -\nabla_x E_\theta(x)$ rather than the density itself, which also removes any dependence on Z. Denoising score matching (Vincent, 2011) and its multi-scale variants (Song and Ermon, 2019), diffusion models (Ho et al., 2020; Song et al., 2020), and flow matching (Lipman et al., 2023) are refinements of this theme: each parameterizes a neural network that plays the role of a (conditional) energy gradient at one or more noise scales, and trains it with a loss that never evaluates the partition function. The scalability of these methods to high-dimensional image, audio, and video distributions is what makes EBMs a practical rather than purely theoretical tool today.

Compositionality. Beyond training convenience, the energy parameterization offers a property that is especially useful for *structured* representations: energies compose additively. If $x = (x_1, \ldots, x_K)$ and we write $E_\theta(x) = \sum_{i=1}^{K} E_\theta^i(x_i)$, the resulting density factorizes as a product $p_\theta(x) \propto \prod_i \exp(-E_\theta^i(x_i))$. The same observation underlies the

compositional guidance used at sampling time in diffusion (Du and Mordatch, 2019; Du et al., 2020) and the ability to combine conditional and unconditional score networks in classifier-free guidance (Ho and Salimans, 2022). In this thesis, we exploit exactly this additive structure: we associate one energy function with each latent factor of a factored MDP and train them jointly while preserving per-factor semantics (Chapters 3, 4, and 6).

Chapter 3 parameterizes the transition dynamics of a factored MDP as a sum of per-factor energy functions and trains them with InfoNCE, making the contrastive training of Section 2.5.2 an EBM estimator of the factored transition function. Chapter 4 retains the same parameterization and adds structural sparsity constraints on the energies. Chapter 6 extends the parameterization to multiple timescales, so that a single family of energy functions scores transitions over variable horizons.

2.6 Identifiable Representation Learning

Representation learning would be considerably easier if the data alone pinned down the "right" latent variables—i.e., the variables we need to learn the DBN of our FMDP. Given enough observations, we would like to claim that a learned encoder $f_\phi : \mathcal{X} \to \mathcal{Z}$ recovers the ground-truth generating variables S. This property, known as *identifiability*, asks whether the true sources can be determined uniquely—at least up to natural symmetries—from the observed distribution. It is the concept that unifies the two lines of work reviewed in the remainder of this section, nonlinear ICA (Comon, 1994; Hyvärinen and Pajunen, 1999) and causal representation learning (Schölkopf et al., 2021), and it is the central formal property we pursue throughout the thesis.

Definition 2.6.1 (Identifiability up to Permutation and Component-wise Bijection)**.** Let the ground-truth sources factorize as $S = S_1 \times \cdots \times S_K$ and let a learned encoder $f : \mathcal{X} \to \mathcal{Z}$ with $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_K$ produce latent coordinates $z = f(x)$. We say f *identifies* S if there exist a permutation $\rho : [K] \to [K]$ and component-wise invertible

maps $h_i : \mathcal{Z}_i \to S_{\rho(i)}$ such that $h_i(z_i) = S_{\rho(i)}$ almost surely.

Identifiability up to permutation and component-wise bijection is the strongest symmetry one can always expect to remain: the labels of the latent dimensions are arbitrary, and each coordinate can be monotonically reparameterized without changing what it represents. Stronger notions, such as identifiability up to affine or linear maps, are characteristic of linear ICA (Comon, 1994); weaker notions, such as block-wise identifiability, allow recovering groups of dimensions rather than individual ones (von Kügelgen et al., 2021; Lachapelle et al., 2024b, 2022). Throughout this thesis we use the component-wise notion unless explicitly stated otherwise.

Unfortunately, identifiability is not easy to attain. A central result by Locatello et al. (2019) re-shaped the field: without inductive biases beyond a factorized prior, the unsupervised learning of disentangled representations (Higgins et al., 2017a,b) is fundamentally ill-posed. Specifically, for any factorized generative model $p(s) = \prod_i p(s_i)$ and any observation function $g : \mathcal{S} \to \mathcal{X}$, there exist infinitely many alternative latent spaces—related to the true one by a volume-preserving diffeomorphism—that produce the exact same observation distribution $p(x)$ and remain factorized under it. The factorized prior alone leaves a large equivalence class of valid "disentangled" solutions, and the observational distribution cannot distinguish between them. This is deep learning counterpart of the classical nonlinear ICA impossibility (Hyvärinen and Pajunen, 1999): without additional assumptions, the mixing is fundamentally non-identifiable.

Hence, we need more information: the way out of this impossibility is to enrich the data with information beyond the marginal observation distribution. Nonlinear ICA recovers the sources by conditioning on an auxiliary variable u—such as time, class label, or environment index—under which the sources are *conditionally* independent (Hyvärinen and Morioka, 2016; Hyvärinen et al., 2019; Khemakhem et al., 2020). Causal representation learning (Schölkopf et al., 2021) recovers the sources by exploiting *interventional* data, where a mechanism $p(s_i \mid \mathrm{pa}(s_i))$ is altered while the others remain fixed (Schölkopf

et al., 2021; Varici et al., 2024; Lippe et al., 2022b, 2023; Locatello et al., 2020a). Both approaches break the symmetries that observational data alone leaves behind, and both recover a latent structure that is not only a statistically convenient factorization but one that aligns with the data-generating process itself.

The alignment is not just an interpretability preference: the causal variables are the ones that most naturally reflect the structure of the world. Moreover, the independent causal mechanisms (ICM) principle (Peters et al., 2017; Schölkopf et al., 2021) establishes that a mechanism $p(s_i \mid \mathrm{pa}(s_i))$ can change independently of the others. A representation that identifies the causal variables, therefore, isolates the *units of change* in the world: a new environment, a new task, or a new distribution typically differs from training in a sparse subset of mechanisms, not in the joint distribution as a whole. A representation that merely fits $p(x)$ has no reason to respect this modular structure and will generally break under any such shift. A representation that identifies the causal variables, by contrast, localizes the change to the factors actually affected, and can be updated accordingly. This is why causal variables are the variables we care about—they are the ones that generalize and transfer.

This is the thread followed throughout this dissertation. The agent's actions are the auxiliary signal—interventions, in a precise causal sense (Pearl, 2009)—that break the symmetries left by observational data and reveal the factors of the world that transfer.

CHAPTER 3

Agents Intervene to Disentangle

Disentangling Independently Controllable Factors via Energy Models

Imagine a simple desk lamp with two separate switches: one toggles the lamp's power, flipping it on or off, while the other cycles the bulb's color between warm and cold light. If you leave both switches untouched, the lamp may still occasionally flicker on or change color on its own, but with a significantly lower probability. By observing the lamp when you flip only the power switch versus doing nothing, you isolate the "on/off" factor; likewise, by pressing only the color switch versus leaving it alone, you isolate the "color" factor. Because each switch only affects one property while the other property evolves naturally, you can disentangle these two characteristics simply by contrasting action-driven changes with natural behavior. The lamp might have other characteristics like volume, weight, and shape; however, these are not factors that can be controlled by the agent. Here we focus on disentangling factors that *are* controllable.

3.1 Problem Formulation

Setting We consider that the agent does not have access to the ground truth factored state space $\mathcal{S}$. Instead, it gets high-dimensional observations that are generated by an unknown diffeomorphism $g_\star : \mathcal{S} \to \mathcal{X} \subseteq \mathbb{R}^{d_x}$. Hence, we are concerned with learning from the observed samples of $T(x_{t+1} \mid x_t, a_t)$ an encoder $f_\phi : \mathcal{X} \to \mathcal{Z}$, where $\mathcal{Z}$ factorizes as $\mathcal{Z} = \mathcal{Z}^1 \times \cdots \times \mathcal{Z}^K$, that *identifies* the underlying factors (See Definition 2.6.1).

3.1.1 Action Effect Sparsity

In many problems, the agent's actions have sparse effects on the environment: just a few factors are controlled, while others just follow their natural transition, unaffected by the agent. To help the agent understand its environment, we assume that the agent has a *special action* a_0 that corresponds to a *no-op* (or observe) action that allows the agent to observe the natural evolution of the environment without intervening.

Transition Dynamics Let $\Psi(s, a) = \mathcal{S} \times \mathcal{A} \to \mathcal{P}([1, 2, \ldots, K])$ be the set of variables affected by action a in state s. We assume the transition dynamics factorize as follows,

$$T(s_{t+1} \mid s_t, a_t) = \prod_{i \in \Psi(s_t, a_t)} T(s^i_{t+1} \mid s_t, a_t) \prod_{j \notin \Psi(s_t, a_t)} T(s^j_{t+1} \mid s_t, a_0); \tag{3.1}$$

where $T(s^i_{t+1} \mid s_t, a_0)$ represents the natural (or observational) dynamics. In here, we will consider conditioning the transition dynamics on the full current state s_t, instead of just the parents, given that $T(s^i_{t+1} \mid s_t, a_t) = T(s^i_{t+1} \mid \mathrm{pa}\left(s^i_{t+1}\right), a_t)$.

Moreover, for the unknown observation function $g_\star$, a diffeomorphism, we know that the *observed* dynamics follow (Boothby, 2003):

$$T(x_{t+1} \mid x_t, a_t) = |\det \left(J_{g_\star^{-1}}(x_{t+1})^\top J_{g_\star^{-1}}(x_{t+1})\right)|^{1/2} T(s_{t+1} \mid s_t, a_t), \tag{3.2}$$

This equation relates the observed dynamics $T(x_{t+1} \mid x_t, a_t)$ to the underlying ground

truth state dynamics $T(s_{t+1} \mid s_t, a_t)$ by the Jacobian matrix $J_{g_\star^{-1}}$, whose determinant quantifies the change in volume between the two spaces. This relation will allow to formalize the ACF algorithm in the next section.

3.2 ACF: Action Controllable Factorization

In this section, we describe the core contribution of this chapter: an algorithm to recover the independent controllable variables directly from the agent's observation.

3.2.1 Learning the Latent Dynamics

We parameterize the encoder by $f_\phi(x) \mapsto z$, with parameters ϕ, and, more importantly, we parameterize the transition function as the sum of energy functions (unnormalized probability densities; see Section 2.5.3) such that,

$$T(z_{t+1} \mid z_t, a_t) \propto \exp\left(\sum_{i=1}^{K} E_\theta^i(z_{t+1}^i, a_t, z_t)\right);$$

with $i \in [K]$ and parameters θ. This sum of energies reflects the factorized structure where each energy represent the transition dynamics of latent variable z^i.

Learning a Markov Representation In order to estimate these energy functions from data and learn a Markov representation suitable for decision making (Allen et al., 2021), we optimize the following training objectives. Firstly, we estimate the inverse dynamics ι^π using our energy functions, as follows,

$$\iota^\pi(a_t \mid z_t, z_{t+1}) = \frac{T(z_{t+1} \mid z_t, a_t)\pi(a_t \mid z_t)}{\sum_{a'} T(z_{t+1} \mid z_t, a')\pi(a' \mid z_t)}; \tag{3.3}$$

$$\propto \frac{\exp\left(\sum_i E_\theta(z_{t+1}^i, a_t, z_t)\right)\pi(a_t \mid z_t)}{\sum_{a' \in A} \exp\left(\sum_i E_\theta(z_{t+1}^i, a', z_t)\right)\pi(a' \mid z_t)}; \tag{3.4}$$

and because our action set is discrete, we can use a softmax multiclass classifier to learn our inverse function by minimizing the cross entropy loss:

$$\ell_{\text{inv}}^{\phi,\theta}(z_t, a_t, z_{t+1}) = -\log \iota^{\pi}(a_t \mid z_t, z_{t+1}). \tag{3.5}$$

Secondly, we use InfoNCE (Oord et al., 2018b) to maximize the mutual information between z_t and z_{t+1}: we use a batch B of $N-1$ negative samples and 1 positive samples, and minimize the following loss,

$$\ell_{\text{fwd}}^{\phi,\theta}(z_t, a_t, z_{t+1}) = -\log \frac{\exp\left(\sum_i E_\theta(z_{t+1}^i, a_t, z_t)\right)}{\sum_{z^{(j)} \in B} \exp\left(\sum_i E_\theta(z^{i,(j)}, a_t, z_t)\right)}. \tag{3.6}$$

Optimizing these losses guarantee that we learn a Markov representation that preserves the relevant information for action effects prediction (Allen et al., 2021) without requiring an explicit reconstruction loss. However, they do not ensure that the representation will align with the controllable factors.

To see this, consider an invertible mapping $g : \mathcal{S} \to \mathcal{Z}$ between the ground truth state s and another representation z. The relation between the densities is given by the following change of variable formula: $T(s_{t+1} \mid s_t, a_t) = |\det J_g(s_{t+1})| T(z_{t+1} \mid z_t, a_t)$. Therefore, if $|\det J_g(s_{t+1})| = 1$ (e.g., g is a rotation), the distribution will match even in the case we use a factorized prior (for an extended discussion, see Locatello et al. (2019); Hyvärinen et al. (2023)).

Therefore, we can achieve observational equivalence: our model can match the statistical properties of the data, but we are not yet guaranteed to identified the controllable variables.

3.2.2 Factorizing the Controllable Variables

We formalize our intuition and exploit the sparsity of the actions' effects to learn a latent representation Z that identifies the controllable factors.

The core idea is to contrast the effect of an action, the distribution $T(x_{t+1} \mid x_t, a_t)$, against the natural dynamics $T(x_{t+1} \mid x_t, a_0)$, where a_0 is the no-op action, using the following ratio:

$$\log r_a(x_{t+1}, x_t) = \log \frac{T(x_{t+1} \mid x_t, a_t)}{T(x_{t+1} \mid x_t, a_0)} \tag{3.7}$$

$$= \log \frac{|\det(J_{g_\star^{-1}}(x_{t+1})^T J_{g_\star^{-1}}(x_{t+1}))|^{1/2} \prod_i T(s^i_{t+1} \mid s_t, a_t)}{|\det(J_{g_\star^{-1}}(x_{t+1})^T J_{g_\star^{-1}}(x_{t+1}))|^{1/2} \prod_i T(s^i_{t+1} \mid s_t, a_0)}; \tag{3.8}$$

$$= \log \frac{T(s^j_{t+1} \mid s_t, a_t)}{T(s^j_{t+1} \mid s_t, a_0)}; \tag{3.9}$$

$$= \log r_a(s_{t+1}, s_t), \tag{3.10}$$

where s^j_{t+1} is the factor affected by a_t when executed in s_t. Therefore, this ratio is a function of the factor s^j_{t+1} and not the rest.

In practice, we can estimate these ratios from observed transitions using Noise Contrastive Estimation (NCE; Gutmann and Hyvärinen, 2010; Hyvärinen et al., 2019) and leveraging our energy parameterization:

$$\begin{aligned} \log r_a(z_{t+1}, z_t) &:= \log r_a(f_\phi(x_{t+1}), f_\phi(x_t)) \\ &:= \sum_i E_\theta(z^i_{t+1}, a_t, z_t) - E_\theta(z^i_{t+1}, a_0, z_t). \end{aligned}$$

Therefore, we train our energy functions to match the observed ratios by training $|A| - 1$ classifiers computed by $\sigma(\log r_a(z_{t+1}, z_t))$ where σ is the sigmoid function. We use the transitions of other actions as negative samples and minimize the following binary cross-entropy loss:

$$\ell^{\phi,\theta}_{\text{align}}(z_t, a_t, z_{t+1}) = \sum_{a' \in A} \mathbf{1}[a' = a_t] \log \sigma \left(\log r_a + \text{sg}(\zeta_a)\right) + \tag{3.11}$$

$$\mathbf{1}[a' \neq a_t] \log \left(1 - \sigma \left(\log r_a + \text{sg}(\zeta_a)\right)\right); \tag{3.12}$$

$$\zeta_a := \log \frac{\pi(a_t \mid z_t)}{\pi(a_0 \mid z_t)}; \tag{3.13}$$

where $\mathbf{1}[\cdot]$ is indicator functions that is 1 when the condition holds, and ζ_a are correction weights to account for the policy used to collect the data. In practice, we estimate the policy from the dataset and use the estimate to compute the loss. Finally, we minimize a weighted sum of these losses and use AdamW as out optimizer (Loshchilov and Hutter, 2019). Algorithm 1 formalizes the method.

3.2.3 Identifiability Guarantees

The core assumption of ACF is that variables are independently controllable, that is, for every state variable s^i, there exists a context $s \in \mathcal{S}$ and action $a \in A$, where the action effect is sufficiently different from the natural dynamics of the variable (a_0 effect). The following theorem establishes identifiability of independently controllable factors if the solution found is sparse.

Theorem 3.2.1 (Identifiability of the Independently Controllable Factors)**.** *Let the learned encoder $f : \mathcal{X} \to \mathcal{Z}$ be a diffeomorphism. If the following conditions hold*

1. *$\mathcal{S} \subset \mathbb{R}^K$ is connected and the unknown observation function $g_\star : \mathcal{S} \to X$ is a diffeomorphism.*

2. *The action effects are **sufficiently different** from the natural dynamics. That is, there exists $i \in [K]$*

$$\frac{\partial}{\partial s^i_t} \frac{T_i(s^i_{t+1} \mid s_t, a_t)}{T(s^i_{t+1} \mid s_t, a_0)} \neq 0$$

for $s_t \in \tilde{S} \subseteq \mathcal{S}$, almost surely. Moreover, there exists at least an action that affects each s^i (independent controllability)

3. *All energy function approximate the factor forward dynamics $E(z^i_{t+1}, a_t, z_t) \propto \log T(z^i_{t+1} \mid z_t, a_t)$;*

4. *(**Sparsity**) The score differences (gradients of the energies)*

$$\frac{\partial}{\partial z^i_{t+1}} \Delta E^a_i = \frac{\partial}{\partial z^i_{t+1}} \left[E(z^i_{t+1}, a_t, z_t) - E(z^i_{t+1}, a_0, z_t) \right] \neq 0$$

for at most one variable i and all actions.

then, there exists a factor-wise diffeomorphism $h : \mathcal{S} \to Z$ between the underlying ground truth factors of variation $\mathcal{S}$ and the learned encoding Z

Theorem 3.2.1 is the full-identification specialization of Lachapelle et al. (2024a, Theorem 2), the more general *partial* disentanglement results for discrete actions. We derived our specialization independently before becoming aware of their general result; for the thesis we defer the proof to theirs, since ours is strictly subsumed.

Under sufficient variability, and by enforcing the sparsity of action's effects and requiring that the desired factors are independently controllable—i.e., there exists a state and action in which a factor is controlled independently from the others— we can guarantee that the ground truth representation is identified.

We conjecture that the binary classifiers arising from $\mathcal{L}_{\text{align}}$ (see Equation 3.11) promote sparsity by competing to capture what makes each action distinct (the action effect not explained by a_0). Because $\mathcal{L}_{\text{align}}$ uses transitions from other actions as hard negatives, each classifier is pushed to rely on a different latent dimension—the one specifically influenced by its action—since sharing dimensions with another classifier would reduce discriminability.

In the following section, we will show empirically cases where this might not hold but

our algorithm still manages to identify some of these variables.

Algorithm 1 Action Controllable Factorization (ACF).

Require: Dataset $\mathcal{D} = \{(x_t, a_t, x_{t+1})\}$, encoder f_ϕ, set per-factor energy models $\{E_\theta^k\}_{k=1}^K$, policy π_w, Learning rate α, weights $\lambda_{\text{align}}, \lambda_{\text{fwd}}, \lambda_{\text{inv}}, \lambda_\pi$

for minibatch $\{(x^n, a^n, x'^n)\}_{n=1}^N \sim \mathcal{D}$ **do**

Encode: $z^n \leftarrow f_\phi(x^n),\ z'^n \leftarrow f_\phi(x'^n)$

Noise: $z^n \leftarrow z^n + \varepsilon^n,\ z'^n \leftarrow z'^n + \varepsilon'^n$

Negatives: $\mathcal{N} = \{(z^i, a^j, z'^j) \mid i, j = 1, \dots, N\}$

Energies: $E_{ij}(a) = \sum_k E_\theta^k(z^{k\prime j}, a, z^i)\ \forall\, i, j \in [N], k \in [K], a \in A$

Policy logits: $\pi^n_{\text{logits}} = \pi_w(z^n)\ \forall\, n$ ▷ Diagonal values are energies of real transitions

Ratios: $\log r_a^{nn} = E_{nn}(a) - E_{nn}(a_0)$

Policy weights: $\zeta_a^n = \log \frac{\pi(a_n|z^n)}{\pi(a_0|z^n)}$

$$\mathcal{L}_{\text{align}} = -\frac{1}{N}\sum_n\sum_a \Big([a^n{=}a] \log \sigma(\log r_a^{nn} + \text{sg}(\zeta_a^n)) + [a^n{\neq}a] \log(1 - \sigma(\log r_a^{nn} + \text{sg}(\zeta_a^n))) \Big)$$

$$\mathcal{L}_{\text{fwd}} = -\frac{1}{N}\sum_n \log \frac{e^{E_{nn}(a^n)}}{\sum_j e^{E_{nj}(a^n)}}$$

$$\mathcal{L}_{\text{inv}} = -\frac{1}{N}\sum_n \log \frac{\text{sg}(\pi(a^n \mid z^n)) e^{E_{nn}(a_n)}}{\sum_{a'} \text{sg}(\pi(a' \mid z^n)) e^{E_{nn}(a')}}$$

$$\mathcal{L}_\pi = -\frac{1}{N}\sum_n \log \frac{e^{\pi^n_{\text{logits}}[a^n]}}{\sum_{a'} e^{\pi^n_{\text{logits}}[a']}}$$

$\mathcal{L} = \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{fwd}}\mathcal{L}_{\text{fwd}} + \lambda_{\text{inv}}\mathcal{L}_{\text{inv}} + \lambda_\pi\mathcal{L}_\pi$

Update: $(\phi, \theta, w) \leftarrow \text{AdamW}((\phi, \theta, w), \alpha, \nabla\mathcal{L})$

end for

3.3 Evaluation

In this section, we empirically evaluate ACF in classical RL test domains. We consider a visual variation of the classical Taxi domain (Dietterich, 2000) and visual Minigrid environments (Chevalier-Boisvert et al., 2023): FourRooms (Sutton et al., 1999) and DoorKey.[1]

Baselines We consider GCL (Generalized Contrastive Learning; Hyvärinen et al., 2019) that can be seen as a vanilla contrastive-based disentanglement algorithm, and DMS

[1]We use Minigrid JAX (Bradbury et al., 2018) re-implementation (Pignatelli et al., 2024)

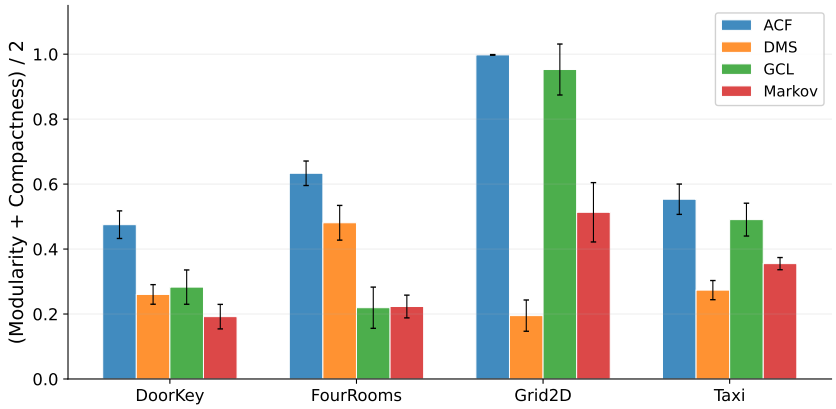


Figure 3.1: **Disentanglement Metrics**: ACF disentanglement consistently outperforms other three benchmarks, across all four domains (DoorKey, FourRooms, Grid2D, and Taxi). ACF achieves perfect disentanglement at Grid2D — a perfectly disentangled representation with respect to the ground truth would have modularity and compactness metrics close to 1. Further details in Table 3.1.

(Disentanglement via Mechanism Sparsity; Lachapelle et al., 2022), a VAE-based (Kingma and Welling, 2014) method that explicitly maximizes sparsity in state dependencies and action effects to drive disentanglement. Moreover, we consider MSA (Markov State Abstractions; Allen et al., 2021), a contrastive-based algorithm that leverages both forward and inverse dynamics to learn Markovian representations but does not explicitly optimize for disentanglement.

Evaluation Protocol To measure disentanglement, we consider test datasets of pairs of $\{(s^i, z^i)\}_i$ where s is the ground truth representation and z is the corresponding learned latent representation. Then, we fit factor-wise regressors (parameterized by feed-forward networks), $h_{ij}(z^i) \mapsto s^j$. The performance of h_{ij} is limited by the amount of information z^i contains about s^j, therefore we measure the quality of the learned regressor using the coefficient of determination R^2. Therefore, for each method we have a matrix R^2 (see Figure 3.2); this matrix would have 1 in the diagonal and low off-diagonal values if the ground truth variables were perfectly identified. We tune all methods via random search in their respective hyperparameter space and train 5 seeds for each method.

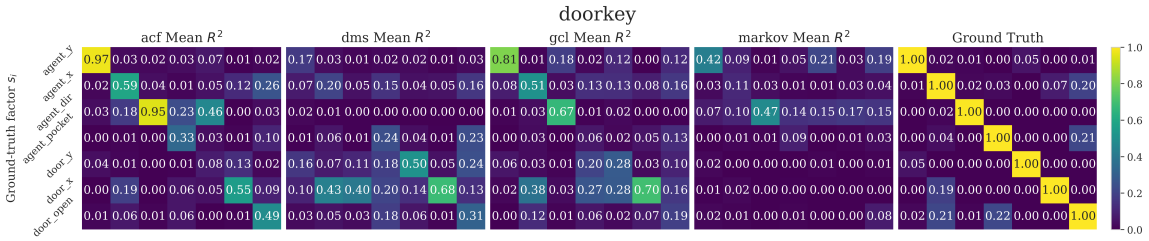


Figure 3.2: **Factorization matrices for DoorKey.** Mean R^2 matrices over 5 seeds.

Table 3.1: **Modularity and Compactness**: A perfectly disentangled representation with respect to the ground truth would have modularity and compactness close to 1. We report the mean values over 5 seeds and 95% CI.

	DoorKey		FourRooms		Grid2D		Taxi	
Method	M	C	M	C	M	C	M	C
ACF	**0.47** $\pm$ 0.04	**0.48** $\pm$ 0.08	**0.62** $\pm$ 0.05	**0.64** $\pm$ 0.06	**1.00** $\pm$ 0.00	**1.00** $\pm$ 0.00	**0.62** $\pm$ 0.05	**0.49** $\pm$ 0.08
DMS	0.25 $\pm$ 0.05	0.27 $\pm$ 0.03	0.28 $\pm$ 0.07	0.68 $\pm$ 0.08	0.19 $\pm$ 0.08	0.20 $\pm$ 0.06	0.31 $\pm$ 0.04	0.24 $\pm$ 0.04
GCL	0.27 $\pm$ 0.08	0.29 $\pm$ 0.07	0.20 $\pm$ 0.05	0.24 $\pm$ 0.12	0.99 $\pm$ 0.01	0.91 $\pm$ 0.16	0.56 $\pm$ 0.08	0.43 $\pm$ 0.06
Markov	0.16 $\pm$ 0.05	0.22 $\pm$ 0.06	0.12 $\pm$ 0.04	0.33 $\pm$ 0.06	0.30 $\pm$ 0.14	0.73 $\pm$ 0.12	0.38 $\pm$ 0.03	0.33 $\pm$ 0.02

Quantitative results Given a R^2 matrix, we search a permutation that maximizes the diagonal using the Hungarian algorithm (Kuhn, 1955). We then aggregate the matrices into two scores, the mean diagonal value, $\frac{1}{K}\sum_i^K R^2_{ii}$ and the mean off-diagonal score $\frac{1}{K}\sum_i \left(1 - R^2_{ii}/\sum_j R^2_{ij}\right)$. The former measures how well a latent factor represents the ground truth factor, and the latter measures how much of the latent factor's information is concentrated off the diagonal value. Ideally, this would mean a score of 1 for the mean diagonal and 0 for the off diagonal if the identification is perfect and the factors are fully independent. However, this is only an upper bound on perfect performance in many environments; e.g. taxi and passenger's position are not fully independent because the passenger can only move if it moves with the taxi. Figure 3.1 shows the results for all methods and domains. Table 3.2 summarizes the R^2 matrices for all domains and Figure 3.3 and Figure 3.4 show qualitative visualizations of the effects of traversing the latent variables learned via ACF.

The factor affected by an action depends on the current state s. Grid2D and FourRooms have different factorizations: In grid2D the agent can move up, down, left and right and 2 dimensions are enough as controllable factors, but in FourRooms

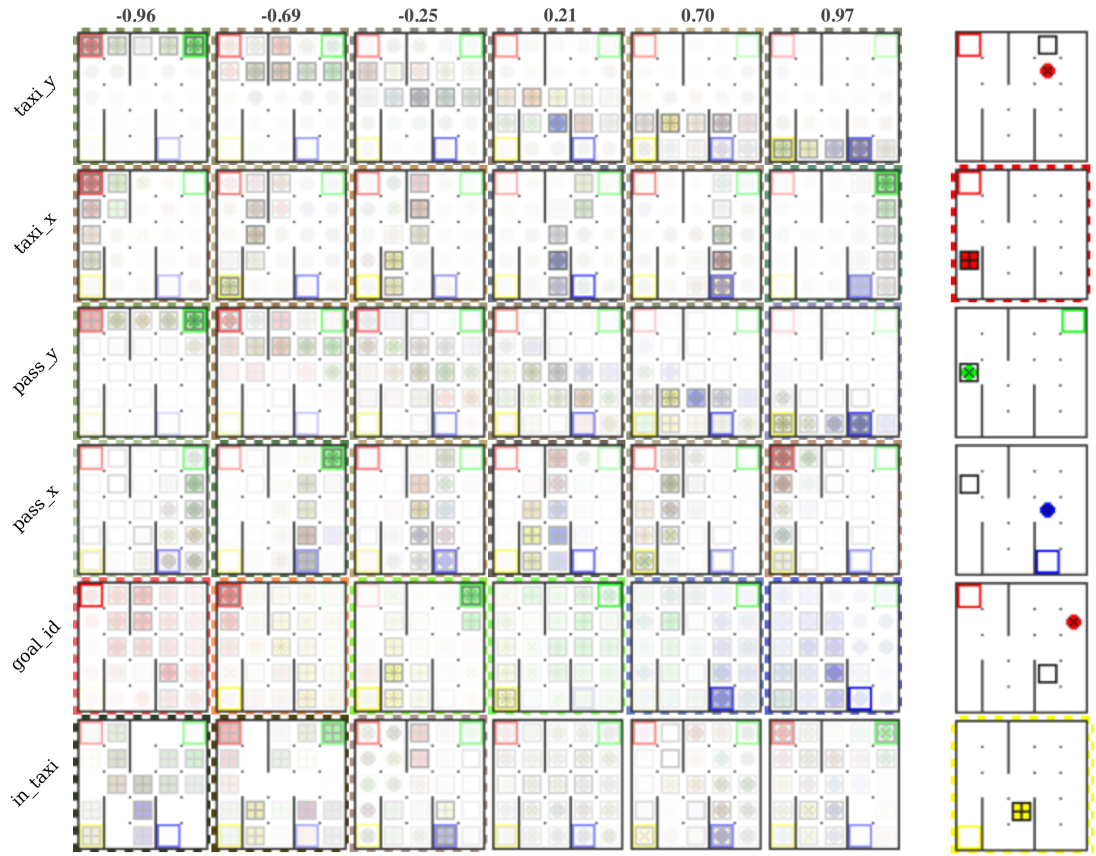


Figure 3.3: **Taxi latent traversals.** In this Taxi rendering, the taxi is represented by a hollow square, the passengers are circles with colors matching their goal positions. When a passenger is in the taxi, the border of the frame is highlighted with stripes. By varying the value of a latent variable (columns), we can see its effect on the mean observation. Each row represents different latent variables.

(Minigrid variant) can rotate, move forward, backward, left or right and, hence, 3 factors are required. More importantly, the factor an action affects is *relative to the agent's orientation* and, this, change causes difficulties for all baseline methods. In particular, DMS, which assumes a global sparse graph, struggles to converge.

Factors are not independent In the Taxi domain, factorization is more challenging because the taxi's position and the passenger's location are inherently coupled; the passenger can move only if it moves with the taxi. Our method outperforms the baselines in this case. Figure 3.3 shows qualitatively the effect of traversing the identified passengers position variables.

Identifying non-controllable variables In the DoorKey domain, not all factors are controllable by the agent: the door's position is sampled at the start of each episode and kept fixed throughout the episode. Although the agent must perceive the door's location to open it, that factor need not be disentangled. In fact, as seen in Figure 3.2, the door y coordinate is not identified. DMS, instead is able to partially identify these variables because it's not constrained to controllable elements and uses the sparsity of the state dependencies.

Table 3.2: **Factorization Metrics**: ACF mean diagonal value is closer to 1—the ideal factorization—than that of all three methods across all four domains. A mean diagonal value close to 1 and a maximum off-diagonal value (mean ± 95% CI across 5 seeds) close to 0 indicates perfect factorization.

Method	Metric	Grid2D	FourRooms	Taxi	DoorKey
ACF	Diag ↑	**0.996±0.000**	**0.792±0.090**	**0.635±0.091**	**0.494±0.042**
	Off ↓	**0.000±0.000**	0.003±0.002	0.024±0.009	0.032±0.007
	Max ↓	**0.000±0.000**	0.014±0.013	0.348±0.136	0.709±0.149
GCL	Diag ↑	0.982±0.024	0.151±0.093	0.532±0.061	0.402±0.057
	Off ↓	0.001±0.002	0.016±0.012	0.038±0.009	0.037±0.014
	Max ↓	0.002±0.003	0.083±0.057	0.526±0.082	0.457±0.077
DMS	Diag ↑	0.632±0.096	0.000±0.000	0.215±0.046	0.220±0.027
	Off ↓	0.049±0.056	**0.000±0.000**	0.019±0.007	0.039±0.009
	Max ↓	0.072±0.087	**0.000±0.000**	**0.221±0.068**	0.517±0.096
Markov	Diag ↑	0.521±0.016	0.081±0.119	0.281±0.062	0.128±0.103
	Off ↓	0.464±0.016	0.013±0.019	**0.016±0.007**	**0.015±0.016**
	Max ↓	0.874±0.026	0.061±0.092	0.313±0.140	**0.310±0.302**

3.3.1 Ablation: All ACF Losses are Important

We performed an ablation study in the Minigrid-DoorKey domain, the most challenging environment considered in the previous experiments. Table 3.3 shows that each loss term plays an important role in improving factorization. In addition, we evaluate the *Factored Markov* variant, which combines our unified factored energy parameterization with the MSA losses (forward and inverse). This variant achieves improved factorization compared to the original MSA, highlighting the importance of the our proposed parameterization.

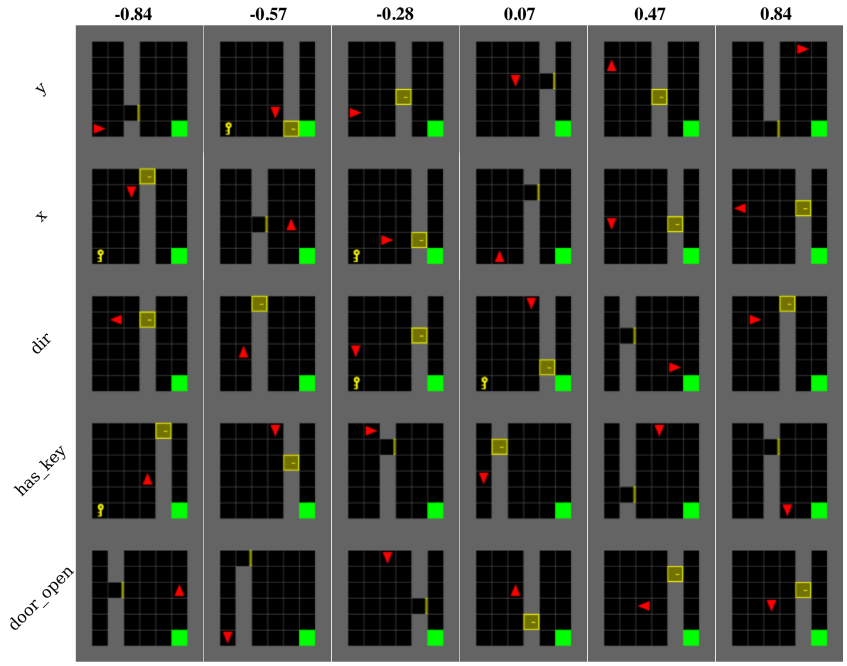


Figure 3.4: **DoorKey latent traversals**. For this domain, we show a random sample from observations that have a particular value of the latent dimension. We only show the controllable elements in DoorKey, that includes the agent position and orientation, the key and the door state. Different rows correspond to different latent variables and different columns represent different values for the corresponding latent variable.

Table 3.3: Ablation on Minigrid-DoorKey Environment over 5 seeds.

Experiment	**Diag (Mean ± 95%CI) ↑**	**MaxOff-Diag (Mean ± 95%CI)↓**
ACF (Full)	**0.5650** ± 0.0423	0.2499 ± 0.0213
Factored Markov	0.4861 ± 0.0491	0.2635 ± 0.0339
No Forward	0.1987 ± 0.0588	0.1028 ± 0.0340
No Inverse	0.5294 ± 0.0274	0.1918 ± 0.0280
No Policy	0.4630 ± 0.0694	0.2301 ± 0.0543
No Ratio	0.5083 ± 0.0767	0.2353 ± 0.0352

3.4 Planning with ACF Factors

To measure the quality of the learned representations for decision making, we train Gaussian predictive models on top of the learned representations in the Taxi domain. We first train the encoder on data covering the state-action space and then freeze it before training the transition and reward models. We consider two model variants: a *vanilla* model that does not assume the representation to be factored, and a *factored* model that learns the DBN structure using the ENCO discovery algorithm (Lippe et al.,

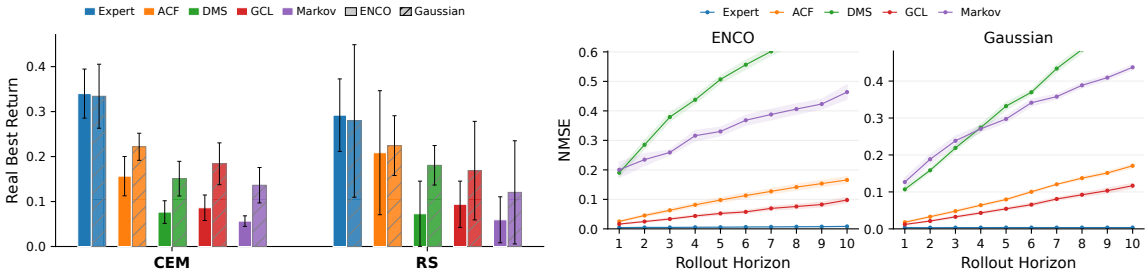
2022a). The latter learns independent predictors for each factor and uses the DBN to mask out unnecessary dependencies, effectively learning a factored transition model. We tune model-learning hyperparameters by randomly sampling the hyperparameter space and training over 5 seeds. We evaluate these models along two dimensions: (1) prediction error, measured by the mean squared error normalized by the variance (NMSE), and (2) planning performance.

To plan, we use two simple planning methods: random shooting (RS; Rao (2009); Nagabandi et al. (2018)) and the cross-entropy method (CEM; Busoniu et al. (2017)). We evaluate 20 episodes for all models across different planning horizons (Figure 3.5). Figure 3.5a shows the best-performing horizon averaged over 5 seeds and 20 episodes with a 95% CI. The expert representation—the one used in the identifiability experiments—achieves the highest performance, and ACF consistently achieves the second highest. The closer a learned representation is to the expert, the better the resulting model; this holds across both planning methods. Perhaps more interesting is the fact that ENCO performs significantly better on ACF than on the baseline methods—i.e., a method that relies on having access to a factored representation works significantly better with ACF, which obtains a factorization closer to the expert.

Finally, a similar pattern holds when evaluating prediction quality (Figure 3.5b): the expert achieves the lowest NMSE, while ACF achieves the third lowest, just above GCL. However, GCL's lower planning performance relative to ACF suggests that it may lose task-relevant information despite having less compounding prediction error.

3.5 Closing Thoughts

ACF showed that, even in simple environments and dynamics, an agent that can sparsely intervene in its environment is able to recover meaningful latent variables from high-dimensional observations—without prior knowledge about what those factors should



(a) Best Horizon Planning performance.

(b) NMSE vs. Prediction Horizon.

Figure 3.5: **Representation Quality**: We measure the quality of our learned representations on their capacity to build useful world models. On the left, we show the best performance achieved by the models using CEM and RS. On the right, we show the accumulation of prediction over horizon. Both plots show that ACF's structure enables models that are better for planning in Taxi.

be!

CRL (Schölkopf et al., 2021; Locatello et al., 2019; Varici et al., 2024; Lachapelle et al., 2022, 2024a) has studied identifiability under various interventional assumptions, but the connection to embodied agents has remained underexplored. ACF showed that by recognizing the agent's ability to intervene through their actions, we can ground CRL knowledge in RL and decision-making leading to methods that can actually recover useful representations for latent factored world models. However, this is not the end of the story.

In this chapter, we worked under the assumption that we had access to a set of actions that could independently control all factors that we were interested in; a tall order for real environments. In fact, our empirical evaluations directly violated this assumption. Remember, in incredibly simple and ubiquitous domain of RL benchmarks, we immediately found a violation: in the Taxi domain, the passenger can only be controlled via the Taxi. We will see in Chapter 6 that this is a very natural situation and, in fact, it is an important signal for learning factored models.

On the other hand, we assume that our per-factor energy functions, modeling the transition dynamics in latent space, have complete knowledge of the current state. Statistically, conditioning on extra information is harmless. But when learning a representation,

it allows the encoder to embed spurious correlations into the factors, obscuring the true temporal structure. Chapter 4 will tackle this issue head on.

CHAPTER 4

Recovering the Dynamics Bayesian Network: A Cooperative Approach

Disentangling via Sparse Temporal Dependencies

Consider an agent that has learned, through the methods of the previous chapter, that the door state is a factor it can control. ACF told the agent that its actions can change whether the door is open or closed; it identified the door as an independently controllable factor. But the agent still does not know *how* to open the door. It does not know that to change the door state, it must first possess the key; that the key's position depends on the agent's grasp; that the door is indifferent to the position of every other object in the room. This is the parent structure of the factored transition dynamics, the dynamic Bayesian network, and ACF does not recover it.

The problem is not that ACF failed. ACF did exactly what it was designed to do: it found which factors change under each action. But knowing that something changed is not the same as knowing what caused it to change. For planning in a factored MDP, the agent needs the full dependency graph: which factors influence which, under what conditions. This chapter is about learning a representation that reveals those dependencies.

The information is, in principle, available in the transition dynamics. If the door's next state depends on the key but not on the cube, then the cross-derivative $\partial^2 \log T / \partial s_{t+1}^{\text{door}} \partial s_t^{\text{cube}}$ is zero while $\partial^2 \log T / \partial s_{t+1}^{\text{door}} \partial s_t^{\text{key}}$ is not. The sparsity pattern of these cross-derivatives *is* the DBN. However, recovering this pattern from learned representations turns out to be subtle, because there are two very different kinds of sparsity at play, and they do not cooperate naturally.

4.1 Two Types of Sparsity

The agent has access to two sources of structural information, and they have fundamentally different character.

The first is *action sparsity*, the signal that ACF already exploits. When the agent takes an action, only some factors are affected. The interventional score difference $\Delta_a = \nabla_{s_{t+1}} \log T(s_{t+1} \mid s_t, a) - \nabla_{s_{t+1}} \log T(s_{t+1} \mid s_t, a_0)$ is sparse because each action intervenes on a small number of factors (Varıcı et al., 2024; Lachapelle et al., 2024a). This sparsity is *interventional knowledge*: it is tied to the causal structure of the world because the agent's action physically affects certain variables and not others. No change of representation can make an action affect a factor it does not physically touch. Action sparsity is grounded.

The second is *temporal sparsity*, the new signal in this chapter. The cross-Hessian of the transition score, $H_{ij} = \partial^2 \log T(s_{t+1} \mid s_t, a) / \partial s_{t+1}^i \partial s_t^j$, captures which factors influence which. If s_t^j is not a parent of s_{t+1}^i in the DBN, then $H_{ij} = 0$. The sparsity of H encodes the parent structure. But unlike action sparsity, temporal sparsity is *observational knowledge*: it reflects statistical dependencies in the dynamics that are not anchored to the agent's interventions. And observational knowledge is ambiguous.

Here is the problem. A sufficiently powerful encoder can find a representation in which the cross-Hessian is diagonal, perfectly sparse, even when the underlying factors have

genuine dependencies. The encoder rotates to align with the dynamics, diagonalizing H at the cost of mixing the true factors. In this basis the temporal dependencies look maximally sparse, but the "factors" are normal modes of the dynamics, not the physical quantities the agent needs for planning. Temporal sparsity, on its own, is ungrounded: it can be trivially satisfied by an encoder that aligns with the dynamics rather than with the causal structure. To make matters worse, the sparsity condition on the cross-hessian H is *local* a powerful encoder can change the basis of the representation continuously in the latent space and achieve (approximately) maximum sparsity.

Action sparsity is what anchors the representation. The agent's action, which in the ground truth affects only one factor, now affects all of them, because each factors were mixed to diagonalize the cross-Hessian. The action effect becomes dense. The ground-truth factored representation is the only one where *both* signals are simultaneously satisfied: the actions are sparse *and* the temporal dependencies reflect the true DBN.

4.1.1 A Simple Example

To make this concrete, consider two factors s_1 and s_2 with transition dynamics and its DBN

$$s^1_{t+1} = f_1(s^1_t, a) + \epsilon_1, \tag{4.1}$$

$$s^2_{t+1} = f_2(s^1_t, s^2_t) + \epsilon_2, \tag{4.2}$$

$$G_s = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad G_a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{4.3}$$

where s^1_t is a parent of s^2_{t+1} (but not vice versa) and the agent has an action that affects s^1_t only. In the ground-truth basis, the cross-Hessian has $H_{21} \neq 0$ and $H_{12} = 0$, reflecting the asymmetric dependency $s^1_t \to s^2_{t+1}$.

For simplicity, consider a rotation $z = R_\theta s$. In the rotated basis, the cross-Hessian

becomes $\tilde{H} = R_\theta H R_\theta^\top$. For the right angle θ, the encoder diagonalizes H: $\tilde{H}_{12} = \tilde{H}_{21} = 0$. Temporal sparsity is perfect. But the action, which in the ground truth produced the sparse score difference $\Delta = (\delta, 0)^\top$, now produces $\tilde{\Delta} = (\delta\cos\theta, \delta\sin\theta)^\top$. It is dense. The rotation that fixed temporal sparsity broke action sparsity.

The only angles where both are satisfied is $\theta \in [-\pi/2, 0, \pi/2, \pi]$: the ground-truth basis and any signed permutation. These are the Pareto solutions: any other rotation that improves one objective necessarily worsens the other.

In the following sections, we will show that this pattern holds more generally, in particular, when we train via gradient descent and expressive neural networks.

4.2 Pitfalls of Sparsity Optimization

4.2.1 Preliminaries

We will build on the theory of Lachapelle et al. (2024a). We will include the equivalence relations

Let $\vartheta = (\phi, \theta)$ be the parameters of our latent model, ϕ for the encoder and θ for the dynamics model T_θ.

Definition 4.2.1 (Observational equivalence)**.** We say that two models are observationally equivalent $\widehat{\vartheta} \sim_{\text{obs}} \vartheta$ if and only if the induced distributions over observed distributions (Equation 3.2) are the same.

$$T_\vartheta(x \mid x, a) = T_{\hat{\vartheta}}(x' \mid x, a);$$
$$\left|\det\left(J_{f_\phi^{-1}}(z')^\top J_{f_\phi^{-1}}(z')\right)\right|^{1/2} T(z' \mid z, a) = \left|\det\left(J_{f_{\hat{\phi}}^{-1}}(\hat{z}')^\top J_{f_{\hat{\phi}}}(\hat{z}')\right)\right|^{1/2} T(\hat{z}' \mid \hat{z}, a).$$

That is, both models match the observed distributions. This is what we achieve when

we train a latent representation that preserves the information of the observations and match the transitions dynamics.

Now, we define what it means to be consistent with the action and temporal graphs.

Definition 4.2.2 (**a**- and **z**-consistency)**.** Let $G^a \in \{0,1\}^{K \times d_a}$ denote the ground-truth action-to-next-state sparsity pattern and $G^z \in \{0,1\}^{K \times K}$ the previous-state-to-next-state pattern, with learned counterparts $\widehat{G}^a$ and $\widehat{G}^z$. A learned model is *a-consistent* with θ if $\widehat{\theta} \sim_{\text{obs}} \theta$ and $\widehat{G}^a$ matches G^a up to a permutation of latent indices, and analogously *z-consistent* for the temporal pattern. We write $\sim_a$ and $\sim_z$ for the induced relations.

Finally, we assume that we have sufficient variation in the dynamics to guarantee we have enough information to disambiguate the latent structure.

Assumption 4.2.3 (Action-Temporal Sufficient Variability; Lachapelle et al. (2024a, Assumption 7,8))**.** For almost all $z \in \mathbb{R}^{d_z}$ and every $\ell \in [d_a]$, there exist $\{(z_r, a_r, a_0)\}_{r=1}^{|\text{Ch}_\ell^{\mathbf{a}}|}$ and where a_r and a_0 differ only in coordinate ℓ such that

$$\text{span}\Big\{\nabla_{z_{r+1}} \log T\big(z_{r+1} \,\big|\, z_e,\, a_r\big)^\top - \nabla_{z_{r+1}} \log T\big(z \,\big|\, z_r,\, a_0\big)^\top\Big\}_{r=1}^{|\text{Ch}_\ell^{\mathbf{a}}|} \;=\; \mathbb{R}^{d_z}_{\text{Ch}_\ell^{\mathbf{a}}}.$$

Analogously for temporal sufficiency, for almost all $z \in \mathbb{R}^{d_z}$ and every $\ell \in [d_z]$, there exist $\{(z_r, a_r)\}_{r=1}^{|\text{Ch}_\ell^{\mathbf{z}}|}$

$$\text{span}\Big\{H_{z_{r+1},z_r} \log T\big(z_{r+1} \,\big|\, z_r,\, a_r\big)\Big\}_{r=1}^{|\text{Ch}_\ell^{\mathbf{z}}|} \;=\; \mathbb{R}^{d_z}_{\text{Ch}_\ell^{\mathbf{z}}}.$$

Lachapelle et al. tells us that under Assumption 4.2.3, observational equivalence, and maximizing sparsity of the induced graphs G^a and G^z, we can recover representations that are a, z-consistent. However, as we will show next, maximizing sparsity is hard in practice. In fact, ACF success can be partially attributed to the fact that sparsity is maximized implicitly in the algorithm design and not as an explicit objective of the loss function.

4.2.2 Maximizing Temporal Sparsity

If we optimize only for temporal sparsity, minimizing the off-diagonal entries of the cross-Hessian, the encoder finds can approximate to the representation that would give us the eigenbasis [1]. This is not a subtle failure; it is the expected behavior. The eigenbasis is the representation that diagonalizes the dynamics, and that is exactly what the optimizer is being asked to do. The resulting "factors" are linear combinations of the true factors with no causal meaning, and the action effects become dense. The temporal signal drifts to a mathematically correct but causally meaningless solution. This intuition holds more generally, as we will see next, and it is exacerbated when the encoder approximator is highly expressive—e.g, a neural network.

Consider how the encoder acts on the cross-Hessian H. The cross-Hessian $H_{ij} = \partial^2 \log T / \partial z^i_{t+1} \partial z^j_t$ has its two indices on two different tangent spaces—the next-state space and the current-state space—and a nonlinear encoder applies its Jacobian separately at z_{t+1} and at z_t. Under the change of variables $\tilde{z} = f_\phi(z)$ with Jacobian $J_\phi(z) := \partial f_\phi / \partial z$, the transformed cross-Hessian is

$$\tilde{H}(\tilde{z}_{t+1}, a, \tilde{z}_t) \;=\; J_\phi(z_{t+1})^{-\top}\, H(z_{t+1}, a, z_t)\, J_\phi(z_t)^{-1}, \tag{4.4}$$

$$\text{where } \; z_t = f_\phi^{-1}(\tilde{z}_t), \; z_{t+1} = f_\phi^{-1}(\tilde{z}_{t+1}). \tag{4.5}$$

The encoder's Jacobian at z_{t+1} multiplies on the left, the Jacobian at z_t multiplies on the right, and these are different matrices evaluated at different points in latent space. In the simpler case of linear encoder $f_\phi(z) = Wz$, the transformation is the simpler $W^{-T} H W^{-1}$. This restriction is what creates the only apparent obstacle to diagonalization: the antisymmetric part of H, which this congruence cannot eliminate. In the general nonlinear encoder case, the obstacle is gone.

To separate the algebra from the function-approximation machinery, fix a single

[1] If the cross-Hessian is has distinct eigenvalues.

realization of (z_{t+1}, z_t, a) and let $H \in \mathbb{R}^{K\times K}$ be the corresponding cross-Hessian. Let $A, B \in \mathrm{GL}(K)$ play the abstract roles of $J_\phi(z_{t+1})^{-\top}$ and $J_\phi(z_t)^{-1}$, and consider the off-diagonal loss

$$L_{\mathrm{temp}}(A, B) \;:=\; \sum_{i\neq j} \big[AHB^\top\big]_{ij}^2. \tag{4.6}$$

The identity is $(A, B) = (I, I)$ with $L_{\mathrm{temp}}(I, I) = \big\|H\big\|_{F,\,\mathrm{off}}^2$.

Proposition 4.2.4 (Two-sided sparsity is always locally reachable)**.** *For any $H \in \mathbb{R}^{K\times K}$,*

$$\inf_{A,B\in\mathrm{GL}(K)} L_{\mathrm{temp}}(A, B) \;=\; 0, \tag{4.7}$$

and the infimum is attained at an orthogonal pair $(A^\star, B^\star) \in \mathrm{O}(K) \times \mathrm{O}(K)$.

Proof. Take the singular value decomposition $H = U\Sigma V^\top$ with $U, V \in \mathrm{O}(K)$ and Σ diagonal and non-negative. Set

$$A^\star := U^\top, \qquad B^\star := V^\top. \tag{4.8}$$

Both are orthogonal, hence in $\mathrm{GL}(K)$. Direct computation gives

$$A^\star H (B^\star)^\top \;=\; U^\top\big(U\Sigma V^\top\big)V \;=\; (U^\top U)\,\Sigma\,(V^\top V) \;=\; \Sigma, \tag{4.9}$$

which is diagonal. All off-diagonal entries vanish, so $L_{\mathrm{temp}}(A^\star, B^\star) = 0$. Since $L_{\mathrm{temp}} \geq 0$ is a sum of squares, this is the infimum and it is attained. □

Remark 4.2.5 (Penalty-agnostic). Lemma 4.2.4 uses only that the SVD makes every off-diagonal entry of $\tilde{H}$ exactly zero. Consequently, for any penalty $\Omega : \mathbb{R} \to [0, \infty)$ with $\Omega(0) = 0$,

$$\inf_{A,B\in\mathrm{GL}(K)} \sum_{i\neq j} \Omega\big([AHB^\top]_{ij}\big) \;=\; 0, \tag{4.10}$$

attained at the same SVD pair. This covers the squared Frobenius loss ($\Omega(x) = x^2$),

the ℓ_1 LASSO penalty ($\Omega(x) = |x|$; Tibshirani, 1996), the Hoyer ℓ_0 surrogate used in Section 4.5.3 (Hoyer, 2004), and the ℓ_0.

This result holds locally, and a powerful enough encoder can warp the latent representation to attain approximately diagonal cross-Hessians. Moreover, the two-sided transform in Equation 4.4 is the same structure that Lachapelle et al. (2024a) use in their identifiability proof for temporal sparsity: their Equation (22) reads $\widehat{H} = Dv(z_{t+1})^\top H\, Dv(z_t)$ with Jacobians evaluated at two different points. This is precisely why they can extract both G_z-preservation and $(G_z)^\top$-preservation of the entanglement map, which is what resolves directional ambiguity between $(i \to j)$ and $(j \to i)$ edges. The pitfall (Lemma 4.2.4) and Lachapelle et al.'s identifiability (their Theorem 3) are two sides of the same coin: the very freedom that lets the encoder chase spurious minima is the freedom that lets structural zeros uniquely pin down the ground truth, provided sparsity is used as a *constraint* rather than a loss to be minimized.

Therefore, in practice, we need to maximize sparsity of the cross-Hessian while respecting the true edges of the DBN. In practice, we need to guarantee not to maximize sparsity beyond the true causal graph. However, naively, this would require knowing the DBN beforehand! Fortunately, we can adopt a cooperative game formulation to optimize the sparsity, and we will show that the Pareto set coincides with the set of G-preserving representations (Theorem 4.4.2). Therefore, by using methods that guarantee Pareto solutions we can design algorithms with more stable dynamics to learn our desired representation.

4.2.3 Maximizing Action Sparsity

Action Sparsity maximization is also prone to degenerate solutions. However, it is not as bad as the temporal one. The mechanism is *factor collapse*. If an action affects two factors jointly, an encoder is free to pick a basis in which the action appears to target a single "combined" factor, gaining one zero for free. This is intuitive. If two variables are always affected together by the actions, we do not have any signal to separate them.

Lemma 4.2.6 (Action sparsity is always reachable)**.** *Let $\Delta_a \in \mathbb{R}^K$ be the score difference of action a with support $S_a := \{j : (\Delta_a)_j \neq 0\}$. For any target index $i^\star \in [K]$ there exists an orthogonal $W \in \mathrm{O}(K)$ with*

$$[W\Delta_a]_j \;=\; 0 \qquad \textit{for all } j \neq i^\star. \tag{4.11}$$

In particular, whenever $|S_a| \geq 2$ the single-action off-$\{i^\star\}$ cost $\sum_{j\neq i^\star}[W\Delta_a]_j^2$ vanishes, while the corresponding identity-basis quantity $\sum_{j\neq i^\star}(\Delta_a)_j^2$ is strictly positive.

Proof. Build W by Gram–Schmidt: take the first row to be $\Delta_a^\top/\|\Delta_a\|$ and complete to an orthonormal basis. Then $W\Delta_a = \|\Delta_a\|\, e_1$. Permute rows to move the nonzero entry to position $i^\star$ if required. All other entries vanish by construction. □

This is the one-sided version of Lemma 4.2.4. Where the temporal sparsity exploits two-sided congruence to diagonalize H, the action pitfall exploits a single orthogonal rotation to concentrate Δ_a on one coordinate.

The one-sided nature of the transformation makes it harder for the encoder to achieve maximum sparsity. However, it can still create practical problems when it is not compatible with the temporal constraints, leading to bad local minima and training instabilities In Chapter 3, we showed that this ambiguities happen in practice when the latent variables are not independently controllable. This can create optimization instabilities when training for sparsity explicitly as DMS (Lachapelle et al., 2022). In fact, ACF better performance can be attributed to the fact that there is no explicit sparsity optimization—e.g., optimizing action masks.

4.3 A Cooperative Solution

The ground truth, or its equivalents, are the solutions where both objectives are simultaneously satisfied. The problem is that naive optimization cannot find it because

the two signals conflict locally. By re-framing the problem of maximizing action and temporal sparsity as a multi-objective optimization, we can keep over-sparsification in check. In this section we will show that the identifiable representation, the one we are interested in recovering, is actually a Pareto-optimal solution of this multi-objective problem.

4.3.1 Negotiating over the Encoder

Let's consider each sparsity constraint as a player that negotiates over a shared resource: the encoder f_ϕ. Each factor $i \in [K]$ wants to have sparse effects from the actions (interventional sparsity) and depend on as few as possible parents (temporal sparsity). Therefore, we can define players as follows.

The *observational sparsity* factor-players want to minimize the non-zeros they see in their cross-Hessian. That is, each factor-player i, has utility

$$\ell^i_{\text{xh}}(z_{t+1}, a, z_t) = \|H^i(z_{t+1}, a, z_t)\|_0 \tag{4.12}$$

$$= \sum_{j \neq i} \left[\frac{\partial^2 E^i_\theta}{\partial z^i_{t+1} \partial z^j_t} \neq 0 \right] \quad \forall i \in [K], \tag{4.13}$$

and they can always achieve maximum sparsity on their own. However, when cooperation fails, the action-players can cause maximum dependency. Hence, if cooperation can provide the factor-player more utility than the disagreement point of $d_f = K - 1$ zeros, then cooperation succeeds.

The *interventional sparsity* action-players i want to maximize the sparsity of the score differences that affect factor i.

$$\ell^i_{\text{action}}(z_{t+1}, z_t) = \sum_{a \in \mathcal{A} \setminus a_0} \left[\frac{\partial \left(E^i_\theta(z_{t+1}, a, z_t) - E^i_\theta(z_{t+1}, a_0, z_t) \right)}{\partial z^i_{t+1}} \neq 0 \right] \tag{4.14}$$

and when cooperation fails, the factor player can try to achieve maximum sparsity, as we showed before. Giving each action player a disagreement point of $d_a = |\mathcal{A}| - 1$, maximally dense vertors.

We will now show that under this formulation, the Pareto optimal solutions coincide with the desired representation. Moreover, we can optimize for Pareto optimal solutions using the Nash Bargaining Solution (see Section 4.5; Nash et al., 1950) as our objective.

4.4 Identifiability via the Sparsity Game

4.4.1 The Sparsity Game

Let $\vartheta = (g_\star, p)$ denote the ground-truth causal model with injective decoder $g_\star : \mathcal{Z} \to \mathcal{X}$, transition $p(z_{t+1} \mid z_t, a_t)$, and sparsity pattern $\mathbf{G} = [G_z \mid G_a] \in \{0,1\}^{K \times (K + d_a)}$. A learner fits $\widehat{\vartheta} = (\widehat{g}, \widehat{p})$ with induced sparsity $\widehat{G}$. We write $\widehat{\vartheta} \sim_{\text{obs}} \vartheta$ for observational equivalence.

Definition 4.4.1 ($\mathbf{G}$-sparsity game)**.** For $\mathbf{G} \in \{0,1\}^{K \times d}$ a sparsity pattern derived from $(g_\star, p)$, the $\mathbf{G}$-*sparsity game* has

- **Players** $j \in [K]$, one per latent dimension.
- **Strategy** $\widehat{\vartheta}$ with feasibility $\widehat{\vartheta} \sim_{\text{obs}} \vartheta$.
- **Loss of player** j the row-sparsity $\ell_j^{\mathbf{G}}(\widehat{\vartheta}) := |\widehat{\mathbf{G}}_{j,\cdot}|_0$.
- **Objective** minimize the loss vector $\boldsymbol{\ell}(\widehat{\vartheta}) \in \mathbb{N}^K$ in the Pareto sense.

Theorem 4.4.2 (Sparsity game achieves $\mathbf{G}$-preservation)**.** *Let $\mathbf{G} \in \{0,1\}^{K \times d}$ be a sparsity pattern satisfying, under $\widehat{\vartheta} \sim_{\text{obs}} \vartheta$ and Assumption 4.2.3, the row-wise support inclusion*

$$(P^\top \widehat{\mathbf{G}})_{j,\cdot} \supseteq \bigcup_{i \,:\, C_{j,i} \neq 0} \mathbf{G}_{i,\cdot}, \qquad \forall j \in [K], \tag{4.15}$$

for a permutation P aligning the reparameterization Jacobian $J_v = D(g_\star^{-1} \circ \widehat{g})$, with

$C := J_v \cdot P$ *having nonzero diagonal. Under Assumption A.1.1*[2]*, every Pareto-optimal* $\widehat{\vartheta}^\star$ *of the* $\mathbf{G}$*-restricted sparsity game satisfies:*

(i) **$\mathbf{G}$-*preservation:*** $P^\top \widehat{\mathbf{G}}^\star = \mathbf{G}$.

(ii) ***Irreducible loss vector:*** $\boldsymbol{\ell}^\star = \big(|\mathbf{G}_{\sigma(1),\cdot}|_0, \ldots, |\mathbf{G}_{\sigma(K),\cdot}|_0\big)$ *for some* $\sigma \in S_K$.

(iii) ***Flat Pareto front:*** *every Pareto point attains* $\boldsymbol{\ell}^\star$ *up to the permutation action of* S_K.

Proof sketch. The argument proceeds in five steps, detailed in Appendix A.

Step 1 (reduction to C). Observational equivalence gives a latent reparameterization $v = g_\star^{-1} \circ \widehat{g}$, and Lachapelle's alignment lemma supplies a permutation P making $C = J_v \cdot P$ diagonally nonzero. Lachapelle et al.'s support-inclusion result, rewritten in P-aligned coordinates, is precisely Equation 4.15.

Step 2 (row floor). Taking cardinalities in Equation 4.15 yields a lower bound on each player's loss: $\ell_j(\widehat{\vartheta}) \geq |\bigcup_{i:C_{j,i}\neq 0} \mathbf{G}_{i,\cdot}|_0 =: \mathcal{M}_j(C)$. Under Assumption A.1.1, this floor is achievable by zeroing slack parameters in $\widehat{p}$, reducing the game to minimization of $\mathcal{M}(C)$ over binary support patterns with nonzero diagonal.

Step 3 (dichotomy). Each off-diagonal $C_{j,i} \neq 0$ is classified as $\mathbf{G}$-*preserving* if $\mathbf{G}_{i,\cdot} \subseteq \mathbf{G}_{j,\cdot}$ (harmless: contributes nothing new to row j's floor) or *non-preserving* otherwise (strictly inflates row j's floor by at least one). Zeroing *all* non-preserving entries in a single row $j^\star$ is a *row-local* modification: row $j^\star$'s floor strictly decreases while every other row's floor is untouched.

Step 4 (Pareto forces preservation). If $C^\star$ has any non-preserving entry, the row-local modification of Step 3 produces a feasible $\widehat{\vartheta}^\dagger$ with strictly smaller $\ell_{j^\star}$ and unchanged ℓ_k for $k \neq j^\star$ — a Pareto improvement. Pareto optimality of $\widehat{\vartheta}^\star$ thus forbids non-preserving entries, i.e. $C^\star$ is $\mathbf{G}$-preserving. Consequently every row's floor collapses

[2]Unconstrained-encoder assumption, stated in Appendix A.

to $|\mathbf{G}_{j,\cdot}|_0$.

Step 5 (cardinality pinch). The inclusion Equation 4.15 gives $(P^\top \widehat{\mathbf{G}}^\star)_{j,\cdot} \supseteq \mathbf{G}_{j,\cdot}$; Step 4 forces equality of cardinalities $|(P^\top \widehat{\mathbf{G}}^\star)_{j,\cdot}|_0 = |\mathbf{G}_{j,\cdot}|_0$. Set inclusion plus cardinality match gives set equality, hence $P^\top \widehat{\mathbf{G}}^\star = \mathbf{G}$. This is (i).

Flatness. Every $\mathbf{G}$-preserving C produces the same floor vector, so the Pareto set in loss-space collapses to a single S_K-orbit. This gives (ii) and (iii). □ □

The intuition is that the row-wise support inclusion in Step 1 converts sparsity into a *combinatorial* problem on the binary pattern of C. The row-local dichotomy in Step 3 is what makes Pareto optimality useful here: non-preserving entries are strictly reducible by a single-row perturbation, so Pareto optimality sparsifies them them one row at a time. $\mathbf{G}$-preservation emerges not as an assumption but as the *only* obstruction-free support pattern compatible with Pareto optimality. Identifiability then follows from cardinality matching in Step 5.

Corollary 4.4.3. *Specializing Theorem 4.4.2 to the sparsity patterns relevant to Nash-ACF:*

- $\mathbf{G} = G^a$*:* a-consistency — *identifies the action→latent graph. Equation* (4.15) *follows from Lachapelle et al.'s interventional argument restricted to action columns.*
- $\mathbf{G} = G^z$*:* row-side z-consistency— *dentifies next-state row supports of the latent cross-Hessian. Equation 4.15 follows from the left-multiplicative side of* $\widehat{H} = J_v(z_{t+1})^\top H\, J_v(z_t)$ *(Lachapelle et al., 2024a).*
- $\mathbf{G} = (G^z)^\top$*:* column-side z-consistency—*identifies current-state column supports. The inclusion is the row inclusion applied to* $\widehat{H}^\top$*.*
- $\mathbf{G} = [G^z \mid G^a]$*:* joint row-side consistency.

Each specialization inherits the three conclusions of Theorem 4.4.2.

Remark 4.4.4 (Joint preservation across all three is strictly stronger). Two algebraic decompositions clarify the relationship between flavors:

$$G\text{-preserving} \iff G^a\text{-preserving} \wedge G^z\text{-preserving}, \tag{4.16}$$

$$\text{directed-}G^z \text{ identification} \iff G^z\text{-preserving} \wedge (G^z)^\top\text{-preserving}. \tag{4.17}$$

Equation 4.16 follows because the row-inclusion $G_{i,\cdot} \subseteq G_{j,\cdot}$ in $[K + d_a]$ decomposes into a latent-part inclusion in $[K]$ and an action-part inclusion in $[K+1, K+d_a]$. Equation 4.17 captures directional identifiability of the latent graph: G^z-row-preservation alone permits edge flips $(i \to j) \leftrightarrow (j \to i)$ that preserve row supports while inverting column supports.

The strictest preservation is therefore G^a-preserving$\wedge G^z$-preserving$\wedge(G^z)^\top$-preserving, equivalent to the Pareto set of the *composite sparsity game* whose players are the union of row-players for $\{G^a, G^z, (G^z)^\top\}$. Using Equation 4.16, this is achievable with $2K$ players: K row-players on G (simultaneously covering G^a- and G^z-row-preservation) plus K row-players on $(G^z)^\top$ (column-side $\mathbf{z}$-preservation). The G-only sparsity game of Definition 4.4.1 delivers joint row-side consistency but *not* column-side z-consistency.

Remark 4.4.5 (Flatness of the Pareto front and selector independence). Theorem 4.4.2(iii) asserts that every Pareto point attains the same loss vector (up to permutation). Consequently every reasonable Pareto selector— sum-of-losses, max-of-losses, Chebyshev scalarization, or NBS—agrees on the sparsity vector at optimality. The specific $\widehat{\vartheta}^\star$ chosen may differ across selectors, but the row-sparsity counts of $\widehat{\mathbf{G}}^\star$ are invariant. Selector choice is a training-dynamics question not an identifiability question.

4.5 A Cooperative ACF

By framing our optimization problem as a cooperative game, the Nash bargaining solution (NBS; Nash et al., 1950) provides exactly this.

4.5.1 Preliminaries: NBS

Setup Consider n players that can cooperate to achieve some outcome in a feasible set of solutions $S \subseteq \mathbb{R}^n$. There exists also a disagreement point $d \in \mathbb{R}^n$; the payoff each player get if negotiation fails.

Nash et al. proposed an axiomatic bargaining approach to this problem based on the following desirable axioms:

Pareto Optimality : the solution must lie in the Pareto frontier. There is no other feasible point where some player is strictly better without making any other player worse;

Symmetry : if the feasible set S and the disagreement point are symmetric with respect to the players, then the solution must give all players equal utility;

Invariance to Affine Transformations of the Utilities : if the utilities are scaled or shifted, the solution transforms accordingly. That is the solution should not depend on arbitrary changes of scale or origin;

Independence of Irrelevant Alternatives: if we reduce the feasible set to $S' \subset S$ and the optimal solution is $s^* \in S'$, then s^* is also the solution to the reduced problem. That is, removing irrelevant options shouldn't change the solution;

and proved that the unique solution satisfying all these axioms is the NBS:

$$s^* = \arg \max_{s \in S, s \geq d} \prod_{i=1}^{n} (s_i - d_i). \tag{4.18}$$

Corollary 4.5.1 (NBS identifies ϑ)**.** *The Nash Bargaining Solution of the composite sparsity game over $\{G^a, G^z, (G^z)^\top\}$ is Pareto-optimal (Nash et al., 1950), hence simultaneously* **a***-consistent and directionally* **z***-consistent with ϑ by Theorem 4.4.2 applied to each component.*

4.5.2 Nash-ACF

Nash-ACF is the cooperative instantiation of ACF. This algorithm is designed to incorporate the observational temporal dependencies to guide factorization while grounding our factors in the interventional knowledge provided by the agent's actions.

As in ACF, we learn an encoder $f_\phi : \mathcal{X} \to \mathcal{Z}$ and a set of per-factor energy functions $E^i_\theta(z^i_{t+1}, a, z_t)$. Let K be the number of latent factors. We optimize the forward and inverse losses,

$$\mathcal{L}_{\text{acf}}(\phi, \theta) = \mathbb{E}_{z_{t+1},a,z_t} \left[\lambda_{\text{fwd}} L_{\text{fwd}} + \lambda_{\text{inv}} L_{\text{inv}} + \lambda_{\text{align}} L_{\text{align}}\right] \tag{4.19}$$

to estimate our energy function and encoder. Chapter 3 showed that these losses allow to learn simultaneously an encoder and the energies that model the latent transition dynamics while aligning with the independently controllable factors.

4.5.3 A Smooth ℓ_0 Surrogate

As is common in sparsity optimization, the ℓ_0 is not a norm and is not continuous; it is a discrete object. In order to optimize via gradient descent, we need a smooth approximation.

We can count zeros smoothly via the inverse participation ratio (Wegner, 1980), also known as the Hoyer measure (Hoyer, 2004; Hurley and Rickard, 2009),

$$M_{\text{hoyer}}(v) = \frac{\|v\|_1^2}{\|v\|_2^2}, \tag{4.20}$$

that smoothly counts the *effective* number of non-zeros in vector v. M_{hoyer} has some nice properties that help our optimization:

Bounded $0 \leq M_{\text{hoyer}}(v) \leq n$. Moreover, $M_{\text{hoyer}}(v) = 1$ if and only if v has exactly one non-zero element, and $M_{\text{hoyer}}(v) = n$ if and only if all elements of v are non-zero and equal magnitude;

Exact Counting : $M_{\text{hoyer}}(v) = s \in \{0 \dots n\}$ is exactly the number of non-zeros if all non-zeros have the same magnitudes;

Scale Invariant : $M_{\text{hoyer}}(\alpha v) = M_{\text{hoyer}}(v)$ for all $\alpha \neq 0$;

Smooth : $M_{\text{hoyer}}(v)$ is infinitely differentiable if $\|v\|_2 \neq 0$.

Therefore, we can use the following surrogate,

$$\ell_0^{\text{hoyer}}(v) = n - M_{\text{hoyer}}(v) \tag{4.21}$$

$$= n - \frac{\|v\|_1^2}{\|v\|_2^2}, \tag{4.22}$$

that smoothly counts the effective number of zeros of v.

4.5.4 The Algorithm

We integrate ACF and NBS in two phases (Algorithm 2):

ACF pretraining phase We learn a first encoder $g_\varphi : \mathcal{X} \to \tilde{\mathcal{Z}}$ and its corresponding energies $\tilde{E}_\theta^i$ using the ACF loss (Chapter 3). This will give us a first representation and its energies that we will need in order to compute the NBS losses.

NBS phase We learn an adapter encoder $f_\phi : \tilde{\mathcal{Z}} \to \mathcal{Z}$ by optimizing the Nash log sum objective. We build cross-Hessian matrices,

$$[\tilde{H}(\tilde{z}', a, \tilde{z})]_{ij} = \frac{\partial^2 \tilde{E}}{\partial \tilde{z}'_i \partial \tilde{z}_j}; \tag{4.23}$$

$$H(z', a, z) = J_{f_\phi}^{-\top}(\tilde{z}') \tilde{H}(\tilde{z}', a, \tilde{z}) J_{f_\phi}^{-1}(\tilde{z}); \tag{4.24}$$

and the score matrices,

$$\tilde{s}_a(\tilde{z}', \tilde{z}) = \nabla_{\tilde{z}'} \tilde{E}(\tilde{z}', a, \tilde{z}) \tag{4.25}$$

$$\tilde{\Delta}^a(\tilde{z}', \tilde{z}) = \tilde{s}_a(\tilde{z}', \tilde{z}) - \tilde{s}_{a_0}(\tilde{z}', \tilde{z}); \tag{4.26}$$

$$\Delta(z', z) = J_f^{-\top}(\tilde{z}')[\tilde{\Delta}_{a_1} \mid \cdots \mid \tilde{\Delta}_{a_{|\mathcal{A}|}}] \tag{4.27}$$

Finally, we can compute the NBS objective and optimize the encode f_ϕ via gradient ascent:

$$\ell^i_{\text{action}} = \mathbb{E}\left[M_{\text{hoyer}}(\Delta_{i,\cdot})\right]; \tag{4.28}$$

$$\ell^i_{\text{row-xh}} = \mathbb{E}[M_{\text{hoyer}}(H_{i,\cdot})]; \tag{4.29}$$

$$\ell^i_{\text{col-xh}} = \mathbb{E}[M_{\text{hoyer}}([H^\top]_{i,\cdot})]; \tag{4.30}$$

$$\mathcal{L}_{\text{nash}} = \sum_i \log\big(d_a - \ell^i_{\text{action}}\big) + \sum_i \log\big(d_f - \ell^i_{\text{row-xh}}\big) + \sum_i \log\big(d_f - \ell^i_{\text{col-xh}}\big) \tag{4.31}$$

Algorithm 2 Nash-ACF (two-phase).

Require: Dataset $\mathcal{D} = \{(x, a, x')\}$, factors K, actions A, baseline a_0
Require: Disagreement points d_a, d_f

// **Phase 1: ACF pretraining**
Initialize encoder $g_\varphi : \mathcal{X} \to \tilde{\mathcal{Z}}$ and per-factor energies $\{\tilde{E}^i_\theta\}_{i=1}^K$
for each pretraining step **do**
    Sample $\{(x, a, x')\} \sim \mathcal{D}$; encode $\tilde{z} = g_\varphi(x)$, $\tilde{z}' = g_\varphi(x')$
    $\mathcal{L}_{\text{acf}} \leftarrow \lambda_{\text{fwd}} L_{\text{fwd}} + \lambda_{\text{inv}} L_{\text{inv}} + \lambda_{\text{align}} L_{\text{align}}$
    Update (φ, θ) by gradient descent on $\mathcal{L}_{\text{acf}}$
end for

// **Phase 2: Adapter learning under Nash sparsity**
Freeze $(g_\varphi, \tilde{E}_\theta)$; initialize bijective adapter $f_\phi : \tilde{\mathcal{Z}} \to \mathcal{Z}$
for each adapter step **do**
    Sample $\{(x, a, x')\} \sim \mathcal{D}$
    Encode and adapt: $\tilde{z} = g_\varphi(x)$, $\tilde{z}' = g_\varphi(x')$; $z = f_\phi(\tilde{z})$, $z' = f_\phi(\tilde{z}')$
    Adapter Jacobians: $J = J_{f_\phi}(\tilde{z})$, $J' = J_{f_\phi}(\tilde{z}')$
    // $\tilde{z}$**-space sparsity signals (frozen energies)**
    $\tilde{\Delta}_a = \nabla_{\tilde{z}'} \tilde{E}(\tilde{z}', a, \tilde{z}) - \nabla_{\tilde{z}'} \tilde{E}(\tilde{z}', a_0, \tilde{z})$ for $a \in A \setminus \{a_0\}$
    $\tilde{H} \in \mathbb{R}^{K \times K}$ with rows $\tilde{H}_{i,\cdot} = \nabla_{\tilde{z}} \nabla_{\tilde{z}'_i} \tilde{E}^i(\tilde{z}'_i, a, \tilde{z})$
    // **Pull back to** z**-space**
    $\Delta_a = J'^{-\top} \tilde{\Delta}_a$; assemble $\Delta \in \mathbb{R}^{(|A|-1) \times K}$ with $\Delta_{i,\cdot}$ the i-th factor's responses across actions
    $H = J'^{-\top} \tilde{H} J^{-1}$
    // **Per-factor utilities (Hoyer** ℓ_0 **surrogate)**
    $\ell^i_{\text{action}} \leftarrow \mathbb{E}[M_{\text{hoyer}}(\Delta_{i,\cdot})]$ ▷ factor i across actions
    $\ell^i_{\text{row-xh}} \leftarrow \mathbb{E}[M_{\text{hoyer}}(H_{i,\cdot})]$ ▷ $\tilde{z}'_i$-row
    $\ell^i_{\text{col-xh}} \leftarrow \mathbb{E}\big[M_{\text{hoyer}}([H^\top]_{i,\cdot})\big]$ ▷ $\tilde{z}_i$-column
    // **Nash product**
    $\mathcal{L}_{\text{nash}} \leftarrow \sum_i \log\big(d_a - \ell^i_{\text{action}}\big) + \sum_i \log\big(d_f - \ell^i_{\text{row-xh}}\big) + \sum_i \log\big(d_f - \ell^i_{\text{col-xh}}\big)$
    Update ϕ by gradient ascent on $\mathcal{L}_{\text{nash}}$ ▷ $g_\varphi, \tilde{E}_\theta$ frozen
end for

4.6 Evaluation

We evaluate on controlled domains where the ground-truth factored structure is known, allowing direct measurement of whether the learned representation recovers the DBN.

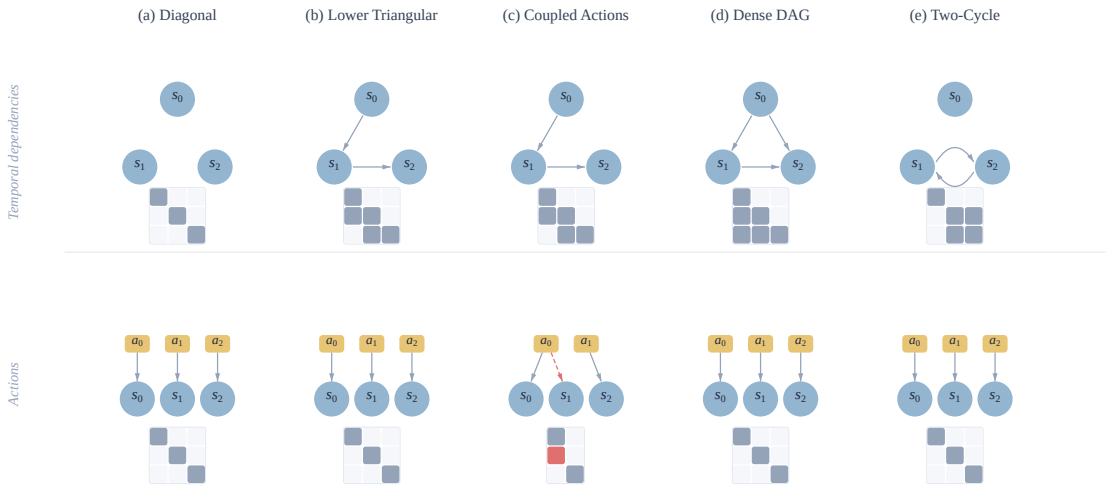


Figure 4.1: Toy domains that explore different configurations of temporal sparsity and action sparsity as defined by Lachapelle et al. (2022).

4.6.1 Metrics

For each pair of learned latent dimension z_i and ground-truth factor s_j, we fit a nonlinear regressor $h_{ij} : z_i \to s_j$ and measure the coefficient of determination R^2_{ij}. A perfectly factored representation yields an R^2 matrix that is a permutation of the identity. Because ground-truth factors may be naturally correlated (passenger position is correlated with taxi position when the passenger is riding), we correct for these baselines:

$$R^2_{\text{corr},ij} = \frac{\max(R^2_{ij} - b_j,\, 0)}{1 - b_j}, \tag{4.32}$$

where b_j is the R^2 achievable by predicting factor j from all other ground-truth factors. After Hungarian alignment of the corrected matrix, we report the mean diagonal $\bar{d}$ (how well each latent captures its matched factor; ideal: 1), the maximum off-diagonal $o_{\max}$(worst-case entanglement; ideal: 0) and the mean off-diagonal value $\bar{o}$.

4.6.2 Domains

We evaluate on a family of domains following Lachapelle et al. (2024a) with varying temporal and action graphs. Most domains are designed to test different topologies of

the temporal graph G_z with actions that independently control the factors. The domain "coupled actions" have a partial control of the factors, which creates ambiguity between two of them, but that can be resolved by the temporal dependencies. Figure 4.1 shows a summary of all domains. For evaluation, the observation is mixed with a known, non-lossy, random MLP to simulate a nonlinear mixing function.

4.6.3 Baselines

1. ACF (Chapter 3) the basis for Nash-ACF;
2. DMS (Lachapelle et al., 2022) operationalizes the sparsity principles that have driven this chapter leveraging explicit differentiable masks for temporal dependencies and action effects in a VAE framework;
3. GCL (Hyvärinen et al., 2019) uses the auxiliary variable for ICA identification and uses NCE (Gutmann and Hyvärinen, 2010) to learn the representation. This is the most general version of ACF, without leveraging the structure provided by the action's interventions;
4. Markov (Allen et al., 2021) uses NCE (Gutmann and Hyvärinen, 2010) to estimate inverse and forward models for representation learning. ACF unifies this under a single energy model and leverage the factorization prior to learn disentangled representations. It's ACF without explicit design for factored models;
5. VAE (Kingma and Welling, 2013) is the base of many representation learning methods in model-based RL (Higgins et al., 2017a; Ha and Schmidhuber, 2018; Hafner et al., 2019, 2021b, 2025) and it uses a Gaussian with diagonal variance to induce factored latent models.

Table 4.1: Aggregate results: Nonlinear Dynamics and Nonlinear Mix. We report IQM ± IQR across domains to show the performance across domains.

Method	$\bar{d}$ (IQM±IQR) ↑	$\bar{o}$ (IQM±IQR) ↓	$o_{\max}$ (IQM±IQR) ↓
Baselines			
GCL	0.799±0.121	0.086±0.018	0.348±0.145
VAE	0.586±0.110	0.125±0.058	0.438±0.104
Markov	0.672±0.077	0.122±0.025	0.440±0.080
DMS	0.558±0.039	0.110±0.034	0.527±0.134
ACF	0.840±0.067	0.036±0.016	0.175±0.068
Nash-ACF			
Action & Temporal	**0.917**±0.121	**0.009**±0.036	**0.045**±0.174
Action	0.894±0.159	0.009±0.032	0.047±0.151
Temporal	0.662±0.193	0.040±0.054	0.236±0.325
Weighted sum			
Action+Temporal	0.842±0.367	0.014±0.036	0.063±0.161
Action	0.843±0.417	0.013±0.031	0.063±0.166
Temporal	0.606±0.223	0.063±0.072	0.233±0.288

4.6.4 Validating Nash-ACF on toy domains from Lachapelle et al. (2022).

We validate Nash-ACF performance on toy domains from Lachapelle et al. (2022) that evaluate different topologies of the temporal dependencies. It is worth noting that most of them are purely recoverable by the actions effects. Except *coupled actions* where two variables s_0 and s_1 are always acted upon together. Figure 4.1 shows the action graphs and temporal graphs of all domains.

We use a random MLP to simulate non-linear entanglement of the observations. We learn an MLP encoder using the different baselines algorithms. Table 4.2 show the results for all methods including Nash-ACF with action-player and temporal-player only. Similarly for Weighted sum optimization of the same Hoyer sparsity loss as a comparison. Table 4.1 summarizes the performance of all methods over all domains via the Interquantile Mean (IQM) and the Interquantile Range(IQR).

Table 4.2: Per-domain results: Nonlinear Dynamics + Nonlinear Mixing function. Best hyperparameter configuration per (method, domain), mean ± std over 5 seeds (tuned by random search).

Method	Metric	Diagonal	Lower Tri.	Coupled Act.	Dense DAG	Two-Cycle
Baselines						
GCL	$\bar{d} \uparrow$	$0.855_{\pm.156}$	$0.894_{\pm.039}$	$0.734_{\pm.065}$	$0.809_{\pm.180}$	$0.704_{\pm.208}$
	$\bar{o} \downarrow$	$0.101_{\pm.095}$	$0.051_{\pm.008}$	$0.096_{\pm.016}$	$0.078_{\pm.070}$	$0.084_{\pm.007}$
	$o_{\max} \downarrow$	$0.269_{\pm.160}$	$0.255_{\pm.020}$	$0.551_{\pm.093}$	$0.359_{\pm.330}$	$0.415_{\pm.057}$
VAE	$\bar{d} \uparrow$	$0.795_{\pm.156}$	$0.648_{\pm.181}$	$0.572_{\pm.100}$	$0.538_{\pm.055}$	$0.441_{\pm.167}$
	$\bar{o} \downarrow$	$0.065_{\pm.058}$	$0.159_{\pm.115}$	$0.193_{\pm.103}$	$0.101_{\pm.027}$	$0.115_{\pm.114}$
	$o_{\max} \downarrow$	$0.177_{\pm.128}$	$0.394_{\pm.260}$	$0.498_{\pm.076}$	$0.507_{\pm.087}$	$0.421_{\pm.277}$
Markov	$\bar{d} \uparrow$	$0.813_{\pm.047}$	$0.713_{\pm.105}$	$0.636_{\pm.233}$	$0.666_{\pm.138}$	$0.616_{\pm.121}$
	$\bar{o} \downarrow$	$0.116_{\pm.043}$	$0.113_{\pm.055}$	$0.232_{\pm.027}$	$0.108_{\pm.048}$	$0.138_{\pm.074}$
	$o_{\max} \downarrow$	$0.335_{\pm.169}$	$0.399_{\pm.154}$	$0.723_{\pm.098}$	$0.442_{\pm.127}$	$0.480_{\pm.255}$
DMS	$\bar{d} \uparrow$	$0.892_{\pm.079}$	$0.538_{\pm.317}$	$0.577_{\pm.202}$	$0.558_{\pm.341}$	$0.422_{\pm.248}$
	$\bar{o} \downarrow$	$0.052_{\pm.041}$	$0.128_{\pm.036}$	$0.120_{\pm.037}$	$0.087_{\pm.046}$	$0.121_{\pm.030}$
	$o_{\max} \downarrow$	$0.192_{\pm.102}$	$0.614_{\pm.188}$	$0.564_{\pm.066}$	$0.441_{\pm.267}$	$0.576_{\pm.112}$
ACF	$\bar{d} \uparrow$	$0.929_{\pm.043}$	$0.883_{\pm.095}$	$\mathbf{0.815}_{\pm.120}$	$0.821_{\pm.143}$	$0.758_{\pm.263}$
	$\bar{o} \downarrow$	$0.045_{\pm.019}$	$0.025_{\pm.015}$	$0.054_{\pm.070}$	$0.036_{\pm.055}$	$0.028_{\pm.035}$
	$o_{\max} \downarrow$	$0.097_{\pm.050}$	$0.145_{\pm.096}$	$\mathbf{0.215}_{\pm.241}$	$0.213_{\pm.328}$	$0.168_{\pm.209}$
Nash-ACF						
Action+Temporal	$\bar{d} \uparrow$	$\mathbf{0.98}_{\pm.00}$	$\mathbf{0.95}_{\pm.01}$	$0.61_{\pm.09}$	$\mathbf{0.94}_{\pm.02}$	$0.80_{\pm.12}$
	$\bar{o} \downarrow$	$0.02_{\pm.01}$	$0.00_{\pm.00}$	$0.07_{\pm.03}$	$0.00_{\pm.00}$	$0.03_{\pm.01}$
	$o_{\max} \downarrow$	$\mathbf{0.07}_{\pm.03}$	$0.01_{\pm.02}$	$0.42_{\pm.18}$	$\mathbf{0.00}_{\pm.00}$	$0.15_{\pm.09}$
Action	$\bar{d} \uparrow$	$\mathbf{0.97}_{\pm.00}$	$0.93_{\pm.03}$	$0.55_{\pm.03}$	$0.92_{\pm.02}$	$\mathbf{0.81}_{\pm.05}$
	$\bar{o} \downarrow$	$0.04_{\pm.01}$	$0.01_{\pm.01}$	$0.04_{\pm.03}$	$0.01_{\pm.01}$	$0.01_{\pm.01}$
	$o_{\max} \downarrow$	$0.15_{\pm.05}$	$0.05_{\pm.06}$	$0.23_{\pm.20}$	$0.03_{\pm.05}$	$\mathbf{0.05}_{\pm.04}$
Temporal	$\bar{d} \uparrow$	$\mathbf{0.99}_{\pm.00}$	$0.71_{\pm.12}$	$0.60_{\pm.09}$	$0.62_{\pm.12}$	$0.61_{\pm.02}$
	$\bar{o} \downarrow$	$0.00_{\pm.00}$	$0.04_{\pm.02}$	$0.04_{\pm.04}$	$0.05_{\pm.02}$	$0.08_{\pm.02}$
	$o_{\max} \downarrow$	$0.02_{\pm.02}$	$0.23_{\pm.13}$	$0.25_{\pm.24}$	$0.30_{\pm.11}$	$0.49_{\pm.12}$
Weighted sum						
Action+Temporal	$\bar{d} \uparrow$	$0.95_{\pm.02}$	$0.81_{\pm.24}$	$0.42_{\pm.10}$	$0.90_{\pm.03}$	$0.66_{\pm.24}$
	$\bar{o} \downarrow$	$0.03_{\pm.03}$	$0.01_{\pm.02}$	$0.09_{\pm.04}$	$0.01_{\pm.01}$	$0.01_{\pm.01}$
	$o_{\max} \downarrow$	$0.09_{\pm.06}$	$0.07_{\pm.09}$	$0.44_{\pm.19}$	$0.03_{\pm.06}$	$0.03_{\pm.03}$
Action	$\bar{d} \uparrow$	$0.96_{\pm.00}$	$0.92_{\pm.03}$	$0.50_{\pm.19}$	$0.91_{\pm.03}$	$0.53_{\pm.18}$
	$\bar{o} \downarrow$	$0.03_{\pm.02}$	$0.01_{\pm.01}$	$0.06_{\pm.02}$	$0.00_{\pm.01}$	$0.01_{\pm.01}$
	$o_{\max} \downarrow$	$0.11_{\pm.07}$	$0.05_{\pm.06}$	$0.33_{\pm.12}$	$0.02_{\pm.04}$	$0.03_{\pm.04}$
Temporal	$\bar{d} \uparrow$	$0.81_{\pm.07}$	$0.58_{\pm.08}$	$0.55_{\pm.10}$	$0.52_{\pm.08}$	$0.60_{\pm.33}$
	$\bar{o} \downarrow$	$0.10_{\pm.04}$	$0.06_{\pm.04}$	$0.02_{\pm.04}$	$0.06_{\pm.03}$	$0.07_{\pm.04}$
	$o_{\max} \downarrow$	$0.27_{\pm.11}$	$0.29_{\pm.21}$	$0.08_{\pm.15}$	$0.27_{\pm.16}$	$0.28_{\pm.18}$

Full Nash-ACF effectively improves over simple ACF and all baselines even if this small, toy domains. Moreover, the ablations show that the main culprit for additive under-performance is due to the temporal sparsity. This aligns with the driving argument of this chapter: the observational nature of the temporal sparsity, reflected in the bilinear transformation of the cross-Hessian causes undesirable minima for the temporal sparsity optimization. Indeed, the action only ablation does not suffer as much.

4.6.5 Improving Taxi and DoorKey using Nash-ACF

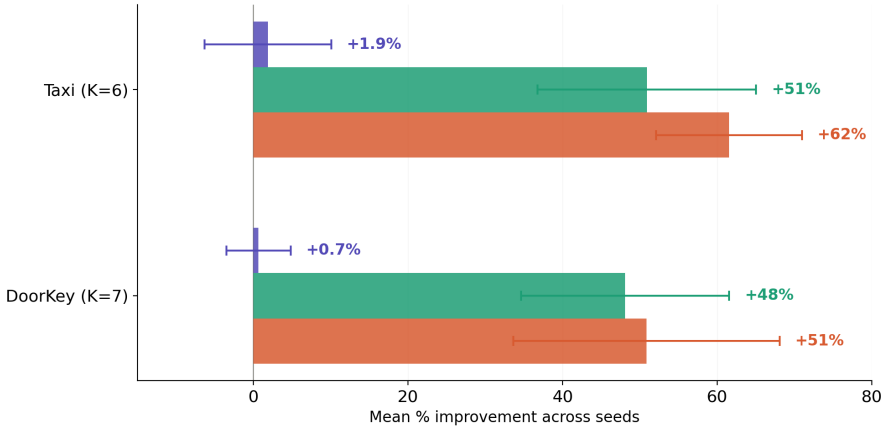


Figure 4.2: Improvement over pure ACF of Visual Taxi and DoorKey factorization when using Nash-ACF.

To validate that incorporating the explicit temporal and action sparsity via Nash-ACF can improve performance in more practical domains, we use the pretrained energies for visual Taxi and DoorKey from Section 3.3.

We use the energies from ACF to learn an adapter $f_\phi : \mathcal{Z}_{\text{acf}} \to \mathcal{Z}$. The sparsity signals come from the local transformation using the Jacobians for the adapter function.

$$\nabla_{z'} E(z', a, z) = J_{f_\phi}^{-\top} \nabla E(z'_{\text{acf}}, a, z_{\text{acf}});$$
$$H_{z',z}(z', a, z) = J_{f_\phi}^{-\top} H_{z'_{\text{acf}}, z_{\text{acf}}} J_{f_\phi}^{-1}$$

Table 4.3 shows the resulting metrics (diagonal, off-diagonal means) before and after, and Figure 4.2 shows the relative improvement over ACF. Both domains clearly benefit from including the temporal sparsity signal, in particular over the off-diagonal values. The leakage reduced considerably, which reflects the disentanglement caused by aligning the variables with the ones that allows sparse dependencies. The performance is not hurt by including the temporal sparsity through Nash-ACF showing that both action and temporal signals balance effectively.

Table 4.3: Nash-ACF results on Taxi and DoorKey. We report the per-seed results and their mean performance.

(a) Taxi ($K = 6$).

	Before (encoder)			After (adapter)		
Seed	cd	co	$\text{co}_{\max}$	cd	co	$\text{co}_{\max}$
106	**0.719**	0.022	0.445	0.697	**0.015**	**0.180**
107	0.552	0.040	0.395	**0.636**	**0.021**	**0.181**
108	0.500	0.014	0.302	**0.508**	**0.004**	**0.068**
110	**0.675**	0.053	0.519	0.633	**0.025**	**0.234**
Mean	$0.611_{\pm 0.089}$	$0.032_{\pm 0.015}$	$0.415_{\pm 0.079}$	$\mathbf{0.619}_{\pm 0.069}$	$\mathbf{0.016}_{\pm 0.008}$	$\mathbf{0.166}_{\pm 0.061}$

(b) DoorKey ($K = 7$).

	Before (encoder)			After (adapter)		
Seed	cd	co	$\text{co}_{\max}$	cd	co	$\text{co}_{\max}$
142	0.450	0.011	0.175	**0.485**	**0.008**	**0.121**
143	**0.490**	0.027	0.606	0.485	**0.013**	**0.196**
144	**0.572**	0.017	0.255	0.556	**0.006**	**0.081**
145	**0.393**	0.035	0.562	0.388	**0.018**	**0.356**
Mean	$0.476_{\pm 0.065}$	$0.022_{\pm 0.009}$	$0.399_{\pm 0.187}$	$\mathbf{0.479}_{\pm 0.060}$	$\mathbf{0.011}_{\pm 0.005}$	$\mathbf{0.188}_{\pm 0.105}$

4.7 Closing Thoughts

Nash-ACF resolved the tension between discovering the intervened variables, the ones ACF made possible to recover, and aligning the representation to reveal the sparse temporal dependencies. By framing the problem as a cooperation, a multi-objective optimization, that required Pareto optimality, we effectively solved the DBN learning

problem. This now allows agents to recover the latent factored MDP, the one we have had to hand-designed in order to exploit effectively.

Moreover, it opens up new opportunities for agent exploration. As we saw, temporal sparsity is not an interventional signal, it's observational. Therefore, some ambiguities remain (Lachapelle et al., 2024a) and only agent active intervention can resolve them.

Finally, there is still some glaring limitations in this chapter. First, we still assume we can observe the action effects in just one time step, essentially we expect the consequences of intervention to be immediately available for the agent to measure. This also implicitly assumes that all variables evolve in a similar frequency, something that does not hold naturally in many domains. For instance, a robotic manipulator stacking cubes naturally handles variables that evolve at different frequencies: in one time step the agent can measure changes in the joint velocities but the cube has barely moved.

Second, we assume that everything is *directly* controllable and that the temporal dependencies hold globally, instead of being a local, transient property. This breaks quite easily. Think of the Taxi domain. The passenger can only be moved by the taxi and it couples them perfectly. Think of the robotic manipulator. The cube and arm become a rigid body by contact and the cube can only be move that way. Control can be *indirect*, caused by dynamical properties of the system, and the couplings and dependencies of the DBN change with the state. Optimizing for a global DBN makes the taxi and passenger, and the arm and cube, one factor.

Chapter 5 recognizes the first limitation and taht temporally-extended actions drive state abstractions. Chapter 6 tackle the second limitation head on, and, we will show that this is in fact an important interventional signal, not a mere complication.

CHAPTER 5

Temporally-extended Interactions drive Abstract World Models

Previous chapters have focused on how agent's action in the world reveal factors. This chapter asks a complementary question: if the agent has access to skills to interact with its world, what is the correct *abstract* state for planning.

Consider a robot that has to control its joints to complete tasks, say stack some blocks. The agent must take high-frequency decisions to achieve a goal that is many time steps ahead and that delays the feedback. This is equivalent to us making decisions about how to stack the block by thinking of each muscle to twitch. The problem is clearly intractable and the model is wasteful because task decisions clearly live in a different timescale. This chapter acknowledges precisely that.

In the following sections, we will construct an abstract state that we can prove sufficient, and necessary, for planning with skills without having to simulate low-level details of the controls, extending the Skill to Symbols framework (Konidaris et al., 2018; Ahmetoglu et al., 2025) to continuous abstract state spaces.

5.1 State Abstractions and Probabilistic Groundings

State abstractions (or state aggregation) have commonly been defined in the form of non-injective functions $f_\phi : \mathcal{S} \to \bar{\mathcal{S}}$ where $\bar{\mathcal{S}}$ is an abstract state space. Recently, Konidaris et al. (2018) propose probabilistic groundings to define a new class of state abstractions. These groundings are defined by $G : \bar{\mathcal{S}} \to \Delta(\mathcal{S})$ and, contrary to state aggregation approaches, these can have overlapping support. That is, for a state s and abstract states $\bar{s}^1$ and $\bar{s}^2$, we can have that $G_{\bar{s}^1}(s) > 0$ and $G_{\bar{s}^2}(s) > 0$. In state aggregation methods, one state has just one abstract state to map to. Therefore, this provides a more expressive framework to build abstractions.

5.2 Building a Value-preserving Abstract MDP

5.2.1 Ground and Abstract MDPs

We start by defining the ground MDP M, the environment that the agent observes by only executing the options.

Definition 5.2.1 (Ground MDP)**.** Let $\mathcal{O}$ be a set of options defined over the agent's state-action space. The ground MDP is $M = (\mathcal{S}, \mathcal{O}, T, R, \gamma, \kappa, p_0)$. $T(s_{t+1} \mid s_t, o)$ is the next-state probability density function seen by the agent when executing option o at s_t and its accumulated discounted reward is $R(s, o) = \mathbb{E}_\kappa \left[\sum_{t'=1}^{\kappa} \gamma^{t'} R(S_{t'}, A_{t'}) \mid s, o\right]$, and $\kappa : \mathcal{S} \times \mathcal{O} \to [0, \infty)$ is the expected option's execution time of option o when initiated at state s.[1]

Definition 5.2.2 (Abstract MDP)**.** The abstract MDP is $\bar{M} = (\bar{\mathcal{S}}, \mathcal{O}, \bar{T}, \bar{R}, \gamma, \bar{\kappa}, \bar{p}_0)$ where $\bar{\mathcal{S}}$ is the abstract state space, $\bar{T} : \bar{\mathcal{S}} \times \mathcal{O} \to \Delta(\bar{\mathcal{S}})$ is the abstract transition kernel, $\bar{R} : \bar{\mathcal{S}} \times \mathcal{O} \to \mathbb{R}$ is the abstract reward function, γ is the discount factor, $\bar{\kappa} : \bar{\mathcal{S}} \times \mathcal{O} \to [0, \infty)$ is the option's duration model and $\bar{p}_0$ is the initial abstract state distribution.

[1]The ground MDP would be an SMDP if we used the multi-time model of options (Sutton et al., 1999).

Given that the objective is to compute plans in the abstract model, we will only consider policies of the form $\pi : \bar{\mathcal{S}} \to \mathcal{O}$ in the rest of the paper. Moreover, to connect the abstract MDP to the ground MDP, we use a grounding function defined in terms of probability density functions, as introduced by Konidaris et al. (2018). The grounding of an abstract state $\bar{s}$ is defined by the probability of the agent being in a state s.

Definition 5.2.3 (Grounding function)**.** Let M be a ground MDP and $\bar{M}$ be an abstract MDP. A grounding function $G : \bar{\mathcal{S}} \to \Delta(\mathcal{S})$ maps $\bar{s}$ to probability measures over $\mathcal{S}$ of M. Given an abstract state $\bar{s}$, we denote by $G_{\bar{s}}$ its grounding probability density. We will denote the tuple $(M, \bar{M}, G)$ as a grounded abstract model.

Definition 5.2.4 (Future State Distribution)**.** Let the tuple $(M, \bar{M}, G)$ be a grounded abstract model. Let the future state distribution be B_t, and defined recursively as follows,

$$B_0(s_0) = p_0(s_0);$$
$$B_t(s_t, \ldots, s_0 \mid o_0, \ldots, o_{t-1}) = T(s_t \mid s_{t-1}, o_{t-1})\, B_{t-1}(s_{t-1}, \ldots, s_0 \mid o_0, \ldots, o_{t-2});$$

and the grounded future state distribution $\bar{B}_t$ is the estimate obtained by grounding the estimate obtained by simulating trajectories in the abstract model $\bar{M}$

$$P(s_t, \bar{s}_t, \ldots, s_0, \bar{s}_0 \mid o_0, \ldots, o_{t-1}) = G_{\bar{s}_t}(s_t)\, \bar{T}(\bar{s}_t \mid \bar{s}_{t-1}, o_{t-1})\, P_{t-1}(s_{t-1}, \bar{s}_{t-1}, \ldots, s_0, \bar{s}_0 \mid o_0, \ldots, o_{t-2});$$
$$\bar{B}_t(s_t, \ldots, s_0 \mid o_0, \ldots, o_{t-1}) = \int P(s_t, \bar{s}_t, \ldots, s_0, \bar{s}_0 \mid o_0, \ldots, o_{t-1})\, d\bar{s}_0 \cdots d\bar{s}_t;$$

Definition 5.2.5 (Dynamics Preserving Abstraction)**.** Let f_ϕ be a mapping $f_\phi : \mathcal{S} \to \mathcal{Z} \subseteq \mathbb{R}^{d_z}$ for some dimension $d_z \in \mathbb{N}$, typically with $d_z \ll d_s$. If for all $o \in \mathcal{O}$ and all $s \in \mathcal{S}$ that are reachable with probability greater than 0, the following holds,

$$T(s_{t+1} \mid s_t, o) = T(s_{t+1} \mid f_\phi(s_t), o); \tag{5.1}$$
$$\Pr(\mathcal{I}_o = 1 \mid s) = \Pr(\mathcal{I}_o = 1 \mid f_\phi(s)); \tag{5.2}$$

where $\mathcal{I}_o$ is an indicator variable corresponding to the option's initiation set. Then, we say that f_ϕ is dynamics-preserving. That is, the information in $f_\phi(s)$ is sufficient to predict the option's effect and determine if an option is executable.

This is similar to model-preserving abstractions (Li et al., 2006) and bisimulation (Givan et al., 2003; Ferns et al., 2004). However, (1) it is stronger in the sense that z must be a sufficient statistic for next-state prediction, and more importantly, (2) this does not impose a condition over the ground reward function. Because we want to build an abstract model to be re-used for task-specific rewards, the ground reward function is considered as a way to measure the cost (negative reward) of executing a skill—retaining Markovianity with respect to the ground reward function would limit how much information can potentially be abstracted away.

5.2.2 Dynamics-preserving abstractions are sufficient to Plan

The following result establishes that dynamics-preserving abstractions are sufficient and necessary to plan with a given set of skills with an abstract model.

Theorem 5.2.6. *Let the tuple $(M, \bar{M}, G)$ be a grounded abstract model and a function $f_\phi : \mathcal{S} \to \mathcal{Z} \subseteq \mathbb{R}^{d_z}$. The model satisfies that $B_t(\cdot \mid o_0, \ldots, o_{t-1}) = \bar{B}_t(\cdot \mid o_0, \ldots, o_{t-1})$ if and only if f_ϕ is dynamics-preserving.*

Proof. See Appendix B. □

Based on this result, we can build a sensible abstract MDP $\bar{M}$, as follows. Let $f_\phi : \mathcal{S} \to \mathcal{Z}$ be a dynamics-preserving abstraction. Given that $T(s'|s, o) = T(s'|z, o)$, where $f_\phi(s) = z$, then we can build a transition function in $\mathcal{Z}$-space, $T(z'|z, o)$, and a grounding function G, that can let us reconstruct $T(s'|z, o)$.

$$
\begin{aligned}
p_0(z) &= \int p_0(s)\mathbf{1}[f_\phi(s) = z]\, ds; \\
T(z'|z,o) &= \int T(s'|z,o)\mathbf{1}[f_\phi(s') = z']\, ds'; \\
G(s'|z,o,z') &= \begin{cases} \frac{p_0(s')\mathbf{1}\left[f_\phi(s')=z'\right]}{p_0(z')} & \text{if } z' \text{is an initial state (there is not previous } (z,o)) \\ \frac{T(s'|z,o)\mathbf{1}\left[f_\phi(s')=z'\right]}{T(z'|z,o)} & \text{otherwise} \end{cases}
\end{aligned}
$$

Given that just knowing z is not enough to determine its grounding distribution, we can build an abstract state space $\bar{\mathcal{S}} \triangleq \mathcal{Z} \times \mathcal{O} \times \mathcal{Z}$ of transition tuples—with special values $z_\perp$ and $o_\perp$ to form $\bar{s}_0 = (z_\perp, o_\perp, z_0)$ for initial abstract states. Let $\bar{s} = (\hat{z}, \hat{o}, \hat{z}')$ and $\bar{s}' = (z, o, z')$ be two abstract states in $\bar{\mathcal{S}}$, we define the abstract MDP functions in this new $\bar{\mathcal{S}}$, as follows.

$$
\begin{aligned}
G_{\bar{s}}(\cdot) &= G(\cdot|\hat{z}, \hat{o}, \hat{z}'); \\
\bar{T}(\bar{s}'|\bar{s}, o) &= \begin{cases} T(z'|z,o) & \text{if } \hat{z}' = z \\ 0 & \text{otherwise} \end{cases}; \\
\bar{R}(\bar{s}, o) &= \mathbb{E}_{s\sim G_{\bar{s}}}\left[R(s,o)\right]; \quad \bar{\tau}(\bar{s}, o) = \mathbb{E}_{s\sim G_{\bar{s}}}\left[\tau(s,o)\right];
\end{aligned}
$$

That is, if the tuples corresponding to $\bar{s}$ and $\bar{s}'$ are not compatible, we define its transition probability as 0, and we define the abstract reward and abstract option's execution length as their corresponding expected values under the grounding function. Finally, the following theorem formally states that this construction is actually useful for planning.

Corollary 5.2.7. *Let the tuple $(M, \bar{M}, G)$ be a grounded abstract model. If the dynamics preserving property holds then the value of policy π computed in abstract model $\bar{M}$ satisfies that $v^\pi(\bar{s}) = \mathbb{E}[v^\pi(s) \mid s \sim G_{\bar{s}}]$. That is, the grounded abstract model preserves the expected*

value under the grounding G.

Proof. Given that we have that, by definition, $T(s_{t+1} \mid s_t, o) = T(s_{t+1} \mid \bar{s}, o) = \mathbb{E}_{\bar{s}_{t+1} \sim \bar{T}(\cdot|\bar{s},o)}[G_{\bar{s}_{t+1}}(s_{t+1})]$. It follows that

$$\begin{aligned}
\mathbb{E}_{s\sim G_{\bar{s}}}[v^\pi(s)] &= \mathbb{E}_{s\sim G_{\bar{s}}}\left[\mathbb{E}_{o\sim\pi}\left[R(s,o) + \mathbb{E}_{s_{t+1}\sim T(s_{t+1}|s,o)}\left[\gamma^\kappa\, v^\pi(s_{t+1})\right]\right]\right] \\
&= \mathbb{E}_{o\sim\pi}\left[\mathbb{E}_{s\sim G_{\bar{s}}}\left[R(s,o)\right] + \mathbb{E}_{s\sim G_{\bar{s}},\, s_{t+1}\sim T(s_{t+1}|s,o)}\left[\gamma^\kappa\, v^\pi(s_{t+1})\right]\right] \\
&= \mathbb{E}_{o\sim\pi}\left[\bar{R}(\bar{s},o) + \mathbb{E}_{\bar{s}_{t+1}\sim\bar{T}(\cdot|\bar{s},o)}\mathbb{E}_{s_{t+1}\sim G_{\bar{s}_{t+1}}}\left[\bar{\gamma}\, v^\pi(s_{t+1})\right]\right] \\
&= \mathbb{E}_{o\sim\pi}\left[\bar{R}(\bar{s},o) + \mathbb{E}_{\bar{s}_{t+1}\sim\bar{T}(\cdot|\bar{s},o)}\left[\bar{\gamma}\, v^\pi(\bar{s}_{t+1})\right]\right] = v^\pi(\bar{s}).
\end{aligned}$$

□

5.3 CAMP: Contrastive Abstract Model for Planning

The mutual information (MI) between random variables X and Y, $MI(X;Y)$, measures the information that each variable holds about the other. We are interested in finding a function f_ϕ that is dynamics-preserving such that we can build our abstract MDP. By Definition 5.2.5, we want to learn $f_\phi(s)$ that is maximally predictive of the effect of o when executed in s and to predict if option o is executable. That is, we want to maximize the following:

$$\max_{f_\phi\in\mathcal{F}} MI(S', I; f_\phi(S), O) \equiv \max_{f_\phi\in\mathcal{F}} MI(S'; f_\phi(S), O) + MI(I; f_\phi(S)), \tag{5.3}$$

where $\mathcal{F}$ is a class of functions that map the high-dimensional ground states to lower-dimensional space. $\mathcal{I}$ is a binary random variable for the initiation set prediction. S', S, O are random variables over the ground states $\mathcal{S}$ and the options set $\mathcal{O}$.

In general, by the data processing inequality, $MI(S'; f_\phi(S), O)$ is upper-bounded by $MI(S'; S, O)$. Therefore, we can show that optimizing the above objective results in a

bounded value loss when using the abstract model to plan. To see this, we first note that by compressing through f_ϕ, we lose information $\Delta MI \triangleq MI(S';S,O) - MI(S';Z,O)$, where $Z = f_\phi(S)$, in the transition dynamics simulation. We show that,

$$\begin{aligned}\Delta MI &\overset{(a)}{=} \mathbb{E}_{p(s)}\left[D_{\mathrm{KL}}\Big(T(s_{t+1} \mid s_t, o) \,\Big\|\, \tilde{T}(s_{t+1} \mid z, o)\Big)\right] \\ &\overset{(b)}{\geq} 2\ln 2 \cdot \mathbb{E}_{p(s)}\left[\|T(s_{t+1} \mid s_t, o) - \tilde{T}(s_{t+1} \mid z, o)\|_1^2\right].\end{aligned}$$

where $p(s)$ is a distribution over s that will depend on the data collection policy and (a) follows from the definition of the KL divergence and (b) from the well-known bound relating the KL divergence and L1 norm[2]. Therefore, the error in the learned transition dynamics is minimized by our objective and this implies, by the following theorem, that this objective also minimizes the value loss resulting from the approximation.

5.3.1 Contrastive Abstract Model Learning

We maximize the previous Infomax objective (5.3) as follows. The term $MI(\mathcal{I}; Z)$ reduces to a cross entropy loss, so we will focus on estimating the term $MI(S'; Z, O)$: we can prove that maximizing both sides of the identity $MI(Z'; Z, O) = (MI(S'; Z') - MI(S'; Z' \mid Z, O))$ implicitly maximizes $MI(S'; Z, O)$ (see extended derivation details in Appendix B.0.1). Intuitively, the first term $MI(Z'; Z, O)$ makes z_{t+1} predictable from knowing the option executed and the previous z_t. The second term avoids collapsing f_ϕ to a trivial solution: maximizing $MI(S'; Z') - MI(S'; Z' \mid Z, O)$ makes f_ϕ retain information about the ground state s (avoiding collapse of the representation) that is maximally predicted by the previous (z_t, o).

We choose to maximize these mutual information terms contrastively using InfoNCE (Oord et al., 2018a) to avoid making assumptions about tractable density models (other MI estimators (Poole et al., 2019; Alemi et al., 2017; Belghazi et al., 2018) can be used).

[2] $D_{\mathrm{KL}}(P \,\|\, Q) \geq 2\ln 2 \cdot \|P - Q\|_1^2$

Using these estimators allows the model to implicitly learn complex grounding functions that improve the quality of the abstract state space. Note that by using InfoNCE for the terms above, this algorithm corresponds to Temporal Predictive Coding (TPC; Nguyen et al., 2021) which proposes abstract states without reconstruction objectives. Therefore, our formulation corresponds to the TPC algorithm in the degenerate case of options being the primitive actions.[3]

In practice, we assume that we have access to a dataset of transition samples $\mathcal{D} = \{(s_t, o_t, r_t^\gamma, s_{t+1}, \kappa_t, \mathcal{I}_t)\}_{t=1}^N$ that correspond to the execution of option o_t from state s_t, terminating in s_{t+1} with a duration of κ_i and accumulated return $r_i^\gamma = \sum_{t'=0}^{\kappa_i - 1} \gamma^{t'} r_{t'}$. $\mathcal{I}_i$ corresponds to the initiation sets of all options in state s_t. This dataset might be initialized by rolling out trajectories with a random agent and further enhanced during the agent's learning (see Algorithm 3).

We propose to learn the abstract model $\widehat{M} = (\widehat{T}, \widehat{R}, \widehat{\mathcal{I}}, \widehat{\kappa})$ with parameters (θ, ϕ), based on the abstraction f_ϕ with parameters ϕ. Notice, that because we need to guarantee good initiation sets by $MI(\mathcal{I}; f_\phi(S))$, the initiation set loss also affects the learning of f_ϕ:

$$\begin{aligned}
\mathcal{L}_{\text{info}} &= -MI_\phi(Z'; Z, O) - MI_\phi(S'; Z'); \\
\mathcal{L}_{\mathcal{I}} &= -\log \widehat{\mathcal{I}}(\mathcal{I}_t \mid f_\phi(s_t)); \\
\mathcal{L}_T &= -\log \widehat{T}(f_\phi(s_{t+1}) \mid f_\phi(s_t), o_t);
\end{aligned}$$

Therefore, $\mathcal{L}_{\text{info}}$, $\mathcal{L}_{\mathcal{I}}$ and $\mathcal{L}_T$ are used to learn the abstraction function f_ϕ. Moreover, to compensate for any imbalances in the data, we use a weighted negative log-likelihood loss for the initiation loss to learn an initiation classifier to be used during planning. To learn the rest of the model, we consider f_ϕ fixed and minimize the following losses and consider samples of the form $(s_{t-1}, o_{t-1}, s_t, o_t, r_t^\gamma, \kappa_t)$ which can be obtained by slicing trajectories

[3]Extended discussion in Appendix B.0.1

appropriately. We map them considering f_ϕ and minimize the following,

$$\mathcal{L}_R = (\widehat{R}(z_{t-1}, o_{t-1}, z_t, o_t) - r_t^\gamma)^2; \quad \mathcal{L}_\kappa = (\widehat{\kappa}(z_{t-1}, o_{t-1}, z_t, o_t) - \kappa_i)^2;$$

Finally, we minimize $\mathcal{L} = \lambda_{\text{info}}\,\mathcal{L}_{\text{info}} + \lambda_{\mathcal{I}}\,\mathcal{L}_{\mathcal{I}} + \lambda_T\,\mathcal{L}_T + \lambda_R\,\mathcal{L}_R + \lambda_\kappa\,\mathcal{L}_\kappa$. In our experiments, all constants were $\lambda_{\text{info}} = \lambda_{\mathcal{I}} = \lambda_T = \lambda_R = \lambda_\kappa = 1$.

5.3.2 Goal-based Planning with an Abstract Model

Consider a goal set $\mathcal{G}_{\text{task}} \subset \mathcal{S}$ and $\mathcal{G}^\phi_{\text{task}} \subset \mathcal{Z}$, its mapping to $\mathcal{Z}$. In order to define the task MDP $M_{\mathcal{G}_{\text{task}}}$ (Algorithm 3, Line 3) for the agent to plan in, we define the task reward function for abstract state $\bar{s} = (\hat{z}, \hat{o}, z)$ as $R_{\mathcal{G}_{\text{task}}}(\bar{s}, o) = \widehat{R}(\bar{s}, o) + R_{\text{task}}\,\mathbf{1}\Big[z \in \mathcal{G}^\phi_{\text{task}}\Big]$ where R_{task} is the goal reward. The first term can be interpreted as the base cost/reward of executing a skill while the second term indicates to the agent the task-specific rewarding states. Moreover, we augment the transition dynamics and set all $z \in \mathcal{G}^\phi_{\text{task}}$ as terminating states by setting $\bar{T}_{\mathcal{G}_{\text{task}}}(z_{\text{done}} \mid z, o) = \mathbf{1}\Big[z \in \mathcal{G}^\phi_{\text{task}}\Big]$. The agent uses the task MDP $\bar{M}_{\mathcal{G}_{\text{task}}}$ to simulate trajectories and improves its policy (Algorithm 3, Line 8) and it can rollout the policy in the environment to collect new data (Algorithm 3, Line 7) that further improves the abstract model.

Algorithm 3 Planning and Learning with an Abstract Model.

Require: Agent π, Ground Environment M,
1: Abstract Model $\bar{M}$, Goal $\mathcal{G}$
2: Initialize dataset $\mathcal{D}$ by rolling out N trajectories
3: $\bar{M} \leftarrow$ PretrainAbstractMDP($\mathcal{D}$)
4: $\bar{M} \leftarrow$ MakeTaskMDP($\bar{M}$, $\mathcal{G}$)
5: **while** true **do**
6: $\quad\mathcal{D} \leftarrow$ Roll out for L steps.
7: $\quad$**if** H steps have passed **then**
8: $\quad\quad\bar{M} \leftarrow$ TrainModel($\bar{M}$, $\mathcal{D}$)
9: $\quad\quad\pi \leftarrow$ TrainAgentImagination($\bar{M}$, π)
10: $\quad$**end if**
11: **end while**

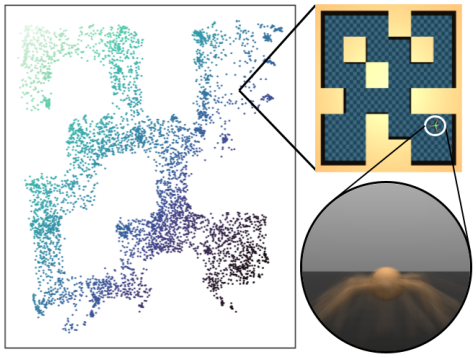
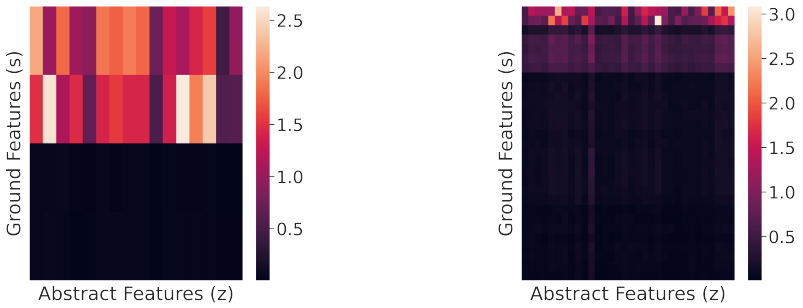


Figure 5.1: **Medium Antmaze**. 2D MDS projection of the learned f_ϕ: it learns to represent the position in the maze. The average grounding shows possible configurations of the ant joints when it is in the represented position.

5.4 Evaluation

Pinball environment (Konidaris and Barto, 2009) This domain has a continuous state space with position vector $(x, y) \in [0, 1]^2$ and velocities $(\dot{x}, \dot{y}) \in [-1, 1]^2$. As opposed to its original formulation, we consider a variant with continuous actions that decrease or increase the velocity by $\Delta(\dot{x}, \dot{y}) \in [-1, 1]^2$. Moreover, we also consider the top view pixel observation of the environment as the agent's observation. As options, we handcrafted position controllers implemented as PID controllers that move the ball in the coordinate directions by a fixed step size. **Antmaze** We consider the problem of controlling a Mujoco (Todorov et al., 2012; Fu et al., 2020) Ant to navigate through a maze. The state space is a 29-dimensional vector that contains the position of the ant in the maze and the ant's proprioception. We consider the Medium Play maze as defined by Fu et al. (2020). We use 8 options learned using TD3 (Fujimoto et al., 2018) that move the ant in the coordinate directions (north, south, east, west and the diagonal directions) in the maze by a fixed step size.



(a) Pinball. (b) Medium Antmaze.

Figure 5.2: MI matrix: ground features s are in the vertical axis and abstract features z are in the horizontal axis. High MI (first two rows) corresponds to the position of the ball or the ant.

5.4.1 Abstract State Space Preserves Relevant Information for Planning

Our main hypothesis is that abstract actions drive state abstraction because the information needed to plan with a structured option set will be less than the ground perception space of the agent. To quantify this, we measure the information contained in the abstract state space about the ground features by estimating the MI using non-parametric methods based on k-nearest neighbors (Kozachenko and Leonenko, 1987). We use Scikit-learn implementation (Pedregosa et al., 2011). In Figure 5.2a, we show the MI matrix between Pinball's ideal features (position and velocities) and the learned features from the pixel observations. For Antmaze (Figure 5.2b), we purposely over-parameterized the abstract space to give enough capacity to learn the full observation, if necessary. However, we can see that features that are not necessary for planning with the skills are effectively abstracted away. In the case of Pinball only the first two dimensions corresponding to the ball position have high MI. In the Antmazes, similarly, the first 7 dimensions have the highest MI which corresponds to position in the maze (first two dimensions) and orientation of the ant's torso. Qualitatively, we can visualize the learned abstract state space using Multidimensional Scaling (MDS; Borg and Groenen, 2005). Figure 5.1 shows the abstract state space learned for the Antmaze and it reveals the

pattern of the coordinate positions of the ant in the maze. Additionally, we show grounded observations that correspond to an abstract state: the ant at the represented position in the maze with many different configurations of the joints and torso.

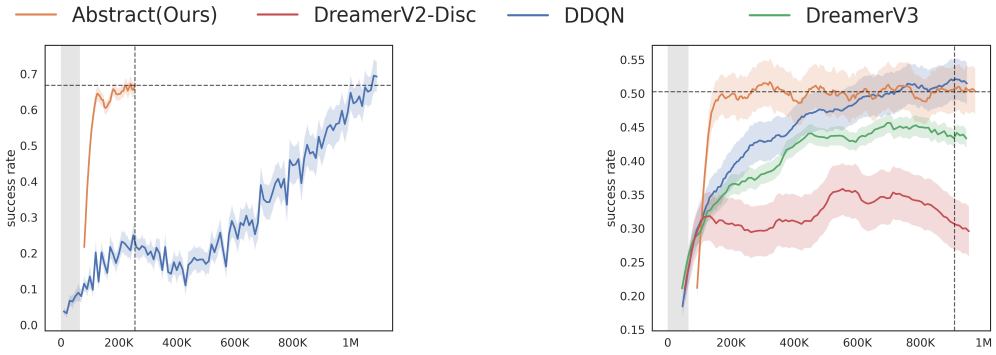
5.4.2 Planning with an Abstract MDP

To evaluate the effectiveness of these models for multiple goal-based tasks, we pretrained abstract models and use them to plan in imagination using Double DQN (Van Hasselt et al., 2016): the DDQN agent rolls out imagined trajectories to improve its policy and then rolls it out in the ground environment to collect new data that is used to learn the task reward function (we keep fix the rest of the model). As our baseline, we use DDQN tuned to learn a policy with the same options but interacting with the ground MDP. In Figure 5.3, we show learning curves (success rate vs. ground environment steps) averaged over different goals and seeds. The error areas represent one standard deviation.

For the pinball domain we use pixel observations as input. In Figure 5.3a, we compare learning curves averaged over 8 goals and 5 seeds where the gray area represent the number of samples used for pretraining phase of the model. These curves show that planning in the abstract model achieves similar performance to the same agent learning directly in the ground MDP which showcases the gain obtained in terms of sample efficiency.

Figure 5.3b shows an analogous plot for Antmaze (9 goals and 5 seeds). In this domain we provide additional results for state-of-the-art model-based RL methods: DreamerV2 and DreamerV3 (Hafner et al., 2021a, 2023). These methods have been shown to work in diverse domains by building (discrete) latent states based on reconstruction losses. However, their performance is limited in comparison to our abstract model: (1) notice that after the gray area our abstract model collects data only to improve the goal reward prediction, whereas the baselines continuously collect data that further improves their models which shows the sample efficiency afforded by our skill-driven abstraction, and (2) our simple DDQN agent learns faster in imagination that the more sophisticated planning

agents of the baselines.



(a) Pinball.

(b) Antmaze.

Figure 5.3: Planning with an abstract model. Success rate v. Environment steps averaged over goals and 5 seeds. The gray area represents the offset for the steps needed to pre-train the model.

5.5 Closing Thoughts

Skills drive state abstractions (Konidaris et al., 2018; Ahmetoglu et al., 2025). This chapter has shown that the right kind of temporally-extended skills provide structure that we can leverage for planning. Planning chess moves was never meant to be done at the muscle control level, it's intractable and it's wasteful. Our agents should simulate the minimum information necessary for planning.

Our method learned the minimal information necessary for planning, but it did not reveal fine grained structure as ACF and Nash-ACF did. But it did reveal temporal structure: variables preserved are variables that are naturally slower than low-level variables. For instance, our MuJoCo Ant learns at the skills timescale, joint information is not important, they change too fast, but global position is relevant at the coarser timescale.

Chapter 6 will unify these concepts under a multi-time ACF model that we can use to reveal variables at different timescales and factorize the independently controllable factors. This will allow us to build a fully abstract factored model that can be used for planning.

CHAPTER 6

Factored World Models via Natural Mechanism Shifts

Consider a robot arm in front of a table with a cube. At high-frequency control, the robot sends torque commands to its joints and observes the resulting state at every timestep. From these single-step transitions, the methods of Chapters 3 and 4 can identify the arm's end-effector position as an independently controllable factor and recover its dependencies in the DBN. But the cube barely moves in a single timestep. Its position changes meaningfully only when the arm executes a multi-step grasp: approach, make contact, close the gripper, lift, move, release. At the single-step resolution, the cube is nearly invisible.

This is the *one-step trap*. Factors that respond only to temporally-extended behavior cannot be identified from single-step transitions, no matter how sophisticated the factorization. The methods of the previous chapters assumed the agent's actions have immediate, observable effects on the factors they control. For slow factors like object positions, this assumption fails.

Something else is happening during the grasp that the previous chapters cannot

capture either. When the arm makes contact with the cube, the dynamics change: the arm and cube become a coupled rigid body. The cube's transition mechanism, which was independent of the arm in free motion, now depends directly on the arm's state. When the arm releases, the coupling disappears. This is a phase transition: a *structural* signal: a natural realization of the sparse mechanism shift hypothesis (Schölkopf et al., 2021). Only the coupled factors' mechanisms change; every other factor's dynamics are unaffected. Moreover, the shift is *interventional*: contact is caused by the agent's actions and determined by the state. It is grounded in the causal structure of the world, just as primitive actions are.

Phase transitions serve two roles at once. First, they *extend the agent's control*: in free motion, the agent controls only the arm; in the grasping phase, it controls the arm *and* the cube, because the coupling lets the arm's actions propagate to the cube's state. The cube was always a factor, but it was unreachable by primitive actions alone. The phase transition is what makes it controllable. Second, phase transitions *reveal the coupling structure*: the fact that the arm and cube become coupled during grasping, while the door state remains independent, is direct evidence that these are separate factors with a conditional dependency. The mechanism shift tells you both that the factors exist and how they interact.

This chapter develops a framework that exploits these ideas. We extend the energy-based factorization of the previous chapters in three directions: continuous actions, multi-step transitions, and phase-dependent dynamics. The result is a multi-time, phase-conditioned energy model trained with a cooperative sparsity framework that balances the interventional, temporal, and phase-shift signals across multiple timescales. We treat the two problems in sequence: first the temporal-resolution problem (Section 6.1–6.2), then the mechanism-shift problem (Section 6.4–6.5), then their composition and identifiability (Section 2.6), and finally the evaluation (Section 6.7).

6.1 Beyond Primitive Actions

Two limitations of the single-step, discrete-action setting of Chapters 3 and 4 must be lifted before we can reach slow factors: the discreteness of the action space, and the assumption that one timestep is enough to see the effect of an action.

Continuous actions. In continuous control, the action space is continuous: joint torques, end-effector velocities, gripper commands. The inverse dynamics loss in ACF (Chapter 3) relies on a discrete action set; with continuous actions, this is no longer possible. We treat the inverse loss as a continuous contrastive problem, paralleling the forward loss of ACF, and use InfoNCE (Oord et al., 2018a): for the forward loss, we sample negative next-states z' from other transitions in the batch; for the inverse loss, we sample negative actions from the batch. The energy model scores positive pairs higher than negative pairs without requiring enumeration of the action space.

Given a batch of negative action samples B_a,

$$\ell_{\text{inv}}^{\Phi,\theta}(z', a, z) = -\log \frac{\exp(E_\theta(z', a, z))}{\sum_{a_j \in B_a} \exp(E_\theta(z', a_j, z))}. \tag{6.1}$$

The score sparsity condition of Chapter 4 also needs a continuous reformulation. Where it previously compared a discrete action a against a reference a_0, we now perturb the action infinitesimally and look at the score difference ψ

$$\Delta_h \psi(s', a, s) = \psi(s', a, s) - \psi(s', a + h, s), \tag{6.2}$$

and take the limit $h \to 0$ to obtain

$$\left[J_\psi^a(s', a, s)\right]_{ij} = \frac{\partial \psi^i}{\partial a'_j}, \tag{6.3}$$

the Jacobian matrix of the score with respect to the action. Lachapelle et al. (2024a)

showed that maximizing sparsity of J_ψ^a leads to identifiability under sufficient variability. This is the continuous analog of the action-sparsity signal we used in the discrete setting, and it slots into Nash-ACF in the same way: each column of J_ψ^a corresponds to an action dimension, and a per-factor action player constrains how each factor responds to it.

Delayed effects and multi-time models. At high-frequency control, the cube's displacement in a single timestep is negligible compared to the arm's. The signal-to-noise ratio for slow factors is too low to identify them from $k = 1$ transitions. We need multi-time transitions: the energy $E(z_{t+\kappa}, a, z_t, \kappa)$ models the relationship between the state at time t and the state at time $t + \kappa$, conditioned on the initial action and the number of steps. At $\kappa = 1$, the energy captures the immediate dynamics (fast factors). At larger κ, it captures the cumulative effect over many steps (slow factors). Different values of κ probe the dynamics at different temporal resolutions, naturally separating factors by the timescale at which they respond to the agent's control.

6.2 Multi-Time Energies

We parameterize the transition dynamics with a phase-conditioned energy model that operates over multi-step transitions. Given a transition from state s at time t to state s' at time $t + \kappa$, under initial action a_0 and phase φ, the energy factorizes per factor:

$$E(z_{t+\kappa}, a_t, z_t, \varphi, \kappa) = \sum_{i=1}^{K} E_i(z_{t+\kappa}^i, \text{summary}(z_t, a_t, \varphi, \kappa), \varphi, \kappa), \tag{6.4}$$

where $z = f_\phi(x)$ is the learned representation, and the summary compresses the multi-step context into a fixed-size vector. We will define formally what a phase is in Section 6.4, and for the rest of this chapter, we will assume that phase labels are provided.

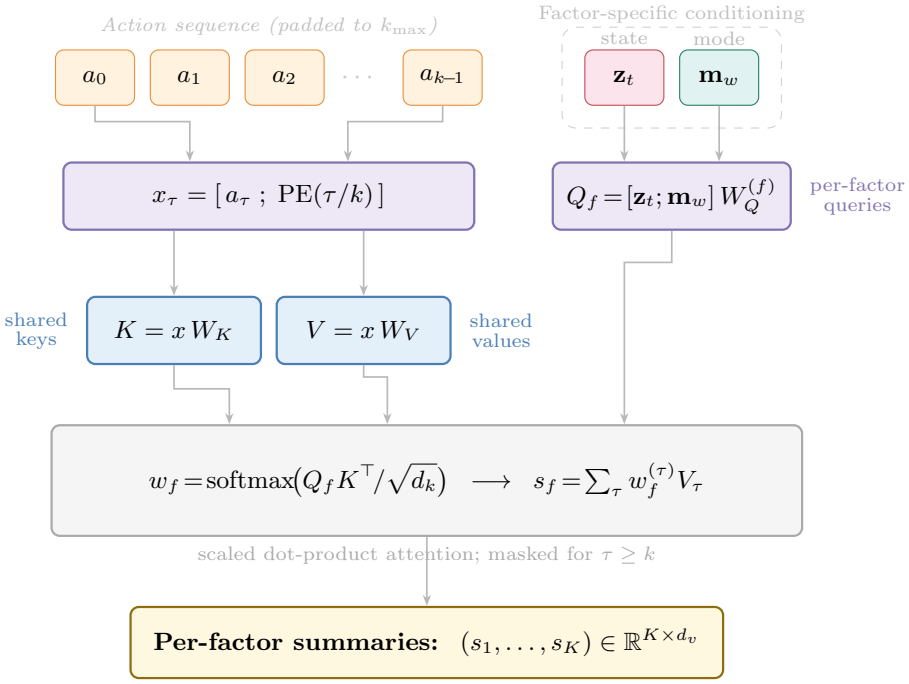


Figure 6.1: Attention for action-sequence encoding.

6.2.1 Architecture

The model has four components.

1. The *encoder* maps observations to a factored latent space, $z = f_\phi(x)$.

2. The *phase embedding* represents each phase as a learned vector $m_\varphi \in \mathbb{R}^{d_m}$. The embeddings are shared across all components of the model: the phase identity conditions the energy, the summary, and the subgoal prediction.

3. The *summary network* compresses the multi-step context $(z, a_0, m_\varphi, \kappa)$ into a fixed-size summary vector via temporal attention (Figure 6.1). The summary captures what happened over the k steps in a form the per-factor energy heads can use. It also incorporates a *subgoal*: a gated encoding of the segment endpoint that captures where the current phase is heading. The subgoal is computed from the start and end states of the segment and the phase embedding, and is stop-gradiented during training to prevent it from collapsing.

The *per-factor energy heads* E_θ^i are separate networks (one per factor) that take the summary, the phase embedding, and the next-state component $z_{t+\kappa}^i$ as input. Separate networks are necessary; shared backbones fail to specialize in practice.

An *action decoder* reconstructs the action sequence $a_{t:t+\kappa}$ from the summary and the initial state. This serves as a regularizer: the summary must contain enough information about what happened during the transition to reconstruct the actions, preventing it from collapsing to a phase-label-only representation.

6.2.2 Contrastive Losses

The energy model is trained with forward and inverse InfoNCE losses, following Chapter 3 but adapted to continuous actions.

Forward. Given a batch of transitions, the energy $E(z_i, z'_j, a_0, \varphi, \kappa)$ should be high for the true pair ($i = j$) and low for negative pairs ($i \neq j$)[1].

$$\mathcal{L}_{\text{fwd}} = -\frac{1}{N} \sum_i \log \frac{e^{E(z^{(i)}, a_0^{(i)}, z'^{(i)}, \varphi, \kappa)}}{\sum_j e^{E(z^{(i)}, a_0^{(i)}, z'^{(j)}, \varphi, \kappa)}}. \tag{6.5}$$

Inverse. The energy should distinguish the correct action from other actions in the batch:

$$\mathcal{L}_{\text{inv}} = -\frac{1}{N} \sum_i \log \frac{e^{E(z^{(i)}, a_0^{(i)}, z'^{(i)}, \varphi, \kappa)}}{\sum_j e^{E(z^{(i)}, a_0^{(j)}, z'^{(i)}, \varphi, \kappa)}}. \tag{6.6}$$

Negative actions are sampled from other transitions in the batch, avoiding the need to enumerate a discrete action set.

[1]We use the superscript $(\cdot)$ to indicate the index of the sample to avoid confusion with the factor index.

6.3 Why Multi-Time Alone Is Not Enough

Multi-time energies solve the temporal-resolution problem: with $k \gg 1$, the cube's displacement becomes visible and the slow factor can be identified. But the methods of Chapter 4 assume a global DBN: a single sparsity pattern that holds across all transitions in the dataset. This assumption breaks under contact.

In free motion, the arm and the cube are independent: the cube has no parents in the dynamics, and the arm's actions affect only the arm. During grasping, the two are coupled: the cube's next state depends on the arm's state, and the arm's next state depends on the cube. These are not the same DBN. They are not even close. A single graph forced to cover both regimes must either include the contact edge always (entangling the two factors in free motion) or omit it always (failing to capture the grasp). Neither choice respects the structure.

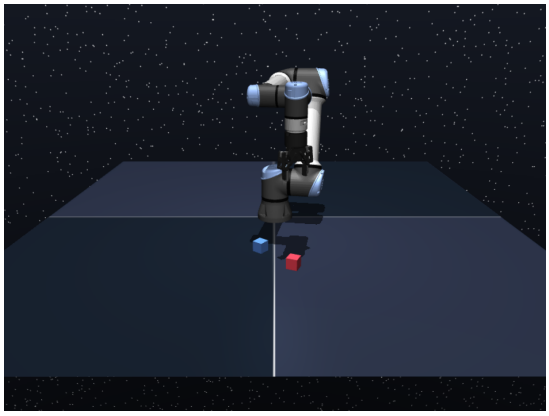
Worse, the rare-but-informative regime is exactly the one a sparsity-driven optimizer will sacrifice. Free-motion transitions vastly outnumber contact transitions in any reasonable dataset. The two factors are strongly correlated during the rare contact phase; under a single-DBN assumption, the optimizer will treat the correlation as noise and absorb it into the encoder. The very signal we need—that contact creates a new dependency—gets washed out.

The fix is to drop the single-DBN assumption. Different regimes correspond to different DBNs, with shared factors but distinct edge sets. The transitions between regimes are not noise: they are the cleanest interventional signal the agent can observe. We formalize this next.

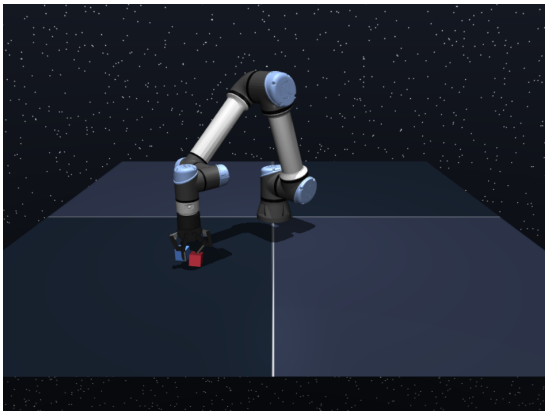
6.4 Phases as Sparse Mechanism Shifts

Consider a dynamical system with continuous state-action space, e.g., a robotic manipulator, where the natural state space factorizes in K blocks $\mathcal{S} = \mathcal{S}^1 \times \mathcal{S}^2 \times \cdots \times \mathcal{S}^K$. The

blocks need not be one-dimensional. Our robot setting (Figure 6.2) partitions naturally into the end-effector state $\mathcal{S}_{\text{ee}}$ and each cube's state $\mathcal{S}_{\text{cube\_x}}$.



(a) Free motion phase φ_f.



(b) Cube 1 grasped, phase φ_c.

Figure 6.2: Double-cube environment from OGBench (Park et al., 2024).

When the arm moves on its own—the free motion phase φ_f—the three factors are independent and their dynamics decouple:

$$T^{\varphi_f}(s' \mid s, a) = T(s'_{\text{ee}} \mid s_{\text{ee}}, a) \cdot T(s'_{\text{cube\_1}} \mid s_{\text{cube\_1}}) \cdot T(s'_{\text{cube\_2}} \mid s_{\text{cube\_2}}). \tag{6.7}$$

When the arm grasps a cube, the dynamics shift and the cube and arm become a rigid body. This is a new phase φ_c:

$$T^{\varphi_c}(s' \mid s, a) = T(s'_{\text{ee}} \mid s_{\text{ee}}, s_{\text{cube\_1}}, a) \cdot T(s'_{\text{cube\_1}} \mid s_{\text{cube\_1}}, s_{\text{ee}}, a) \cdot T(s'_{\text{cube\_2}} \mid s_{\text{cube\_2}}, a). \tag{6.8}$$

The transition $\varphi_f \to \varphi_c$ creates an observable shift in dynamics—a mechanism shift in the causal sense (Schölkopf et al., 2021).

These transitions are *natural experiments*. They reveal new structure that the agent did not know before, and they are caused by the agent's own actions: the agent moved its arm to make contact, and the contact triggered the mechanism shift. This makes the phase-transition signal *interventional*, grounded in the causal structure rather than merely

observational. The agent's extended behavior—its temporally-extended skills—generates the mechanism shifts that reveal the coupling structure.

This is also why multi-time and phases must be combined. Multi-time transitions are needed to observe slow factors: the cube moves meaningfully only over $k \gg 1$. Phases are needed to observe mechanism shifts: the coupling appears only during contact. Together, they provide the temporal resolution and the structural resolution needed to identify the full factored dynamics.

6.4.1 Formal Definitions

Definition 6.4.1 (Phase)**.** A *phase* $\varphi \in \Phi$ is a dynamical regime characterized by its own factored transition kernel:

$$T_\varphi(s' \mid s, a) = \prod_{i=1}^{K} T_\varphi^i(s'_i \mid \mathrm{pa}_\varphi^{s,a}(s'_i)), \tag{6.9}$$

where $\mathrm{pa}_\varphi^{s,a}(s'_i) \subseteq \{s_1, \ldots, s_K, a_1, \ldots, a_{d_a}\}$ are the parents of factor i under phase φ. The DBN graph $G_\varphi^s = (V, E_\varphi)$ has edges $E_\varphi = \{(j, i) : s_j \in \mathrm{pa}_\varphi(s'_i)\}$, and similarly the action graph G_φ^a.

The set of factors V is shared across phases, but the edges E_φ change. In free motion, G_{free} has no edge from the end-effector to the cube. In grasping, G_{grasp} acquires that edge. Phases also induce a per-phase action graph $G_\varphi^a \in \{0, 1\}^{K \times d_a}$, encoding which actions affect which factors under phase φ. This graph also changes across phases: a gripper command, for instance, has no effect on the cube during free motion, but becomes a parent of the cube once contact is established.

Definition 6.4.2 (Mechanism Shift)**.** A *mechanism shift* between phases φ_0 and φ_1 is a change in the transition mechanisms. The *involved set* $\mathcal{J}(\varphi_0, \varphi_1) \subseteq \{1, \ldots, K\}$ contains

the factors whose mechanisms changed:

$$i \in \mathcal{J}(\varphi_0, \varphi_1) \iff T^i_{\varphi_0} \neq T^i_{\varphi_1}. \tag{6.10}$$

Factors outside $\mathcal{J}$ are *uninvolved*: their mechanisms are preserved across the phase transition.

Assumption 6.4.3 (Sparse Mechanism Shift Hypothesis (Schölkopf et al., 2021))**.** Each phase transition involves few factors: $|\mathcal{J}(\varphi_0, \varphi_1)| \ll K$. Most mechanisms are preserved across any phase transition.

This is not arbitrary; it reflects the structure of real-world dynamics. When the arm grasps the cube, only the cube's mechanism changes (it becomes coupled to the arm). The door, the table, every other object's dynamics are unaffected. Sparsity of mechanism shifts is a consequence of the locality of physical interaction.

Phase transitions are state-determined interventions. Unlike the regime changes in the causal representation learning literature (Lachapelle et al., 2022; Locatello et al., 2019; Schölkopf et al., 2021), where different "environments" are provided externally, phase transitions in our setting are caused by the agent and determined by the state. Contact occurs when the arm reaches the cube; this is a function of s_{ee} and s_{cube}. The agent's extended behavior—approaching, making contact—is what triggers the mechanism shift. This makes the phase-transition signal interventional and grounded, not merely observational.

Phases partition the state-action space into stable structural signatures. We can then use the cooperative sparsity framework of Chapter 4 naturally, with the key difference that we now have sparsity signals at multiple timescales and across phases. The NBS must balance all of them without sacrificing rare phase-transition signals.

6.5 Phases as Players in Nash-ACF

In Chapter 4, we defined two kinds of players: an action-player and a factor-player, each constraining a different sparsity pattern. We generalize these players in two ways.

(a) *Continuous actions.* Action interventions are no longer discrete. Instead of score-difference vectors we have the Jacobian matrix of the score with respect to the action vector (Equation 6.3). Each column corresponds to an action dimension, and each factor responds differently to it. The action player becomes a per-factor action player.

(b) *Phase-conditioned players.* Structural patterns change across phases. A mechanism shift can change a factor's response to a particular action dimension, and it can create or destroy temporal dependencies. Each phase contributes its own set of players, each constraining the per-phase sparsity pattern.

A shared encoder must satisfy every per-phase, per-player constraint simultaneously. This is exactly the cooperative setting Nash-ACF was designed for: each player is a stakeholder, the Nash product is the aggregator, and the maximizer is the Pareto-optimal allocation that respects every constraint at once. The phase-conditioned Nash product factorizes across phases and players:

$$\max \prod_{\varphi \in \Phi} \prod_{i=1}^{K} \left(d_a - \mathbb{E}\left[\ell^{i,\varphi}_{\text{action}}\right]\right) \prod_{i=1}^{K} \left(K - 1 - \mathbb{E}\left[\ell^{i,\varphi}_{\text{xh}}\right]\right) \tag{6.11}$$

$$= \sum_{\varphi \in \Phi} \sum_{i=1}^{K} \log\left(d_a - \mathbb{E}\left[\ell^{i,\varphi}_{\text{action}}\right]\right) + \sum_{i=1}^{K} \log\left(K - \mathbb{E}\left[\ell^{i,\varphi}_{\text{xh}}\right]\right). \tag{6.12}$$

The product structure is what protects rare-but-informative phases. A weighted sum could trade away a sparsity violation in φ_c for a small gain in the dominant φ_f; the product cannot, because driving any single factor toward zero collapses the entire objective.

6.6 Identifiability

The identifiability theorem for phase-conditioned Nash-ACF rests on a purely structural fact about G-preservation. The encoder must choose a representation that preserves the local graphs for all phases, instead of the union of the graphs. It is the union of the phase graphs what would be preserved if we applied blindly Nash-ACF to these problems.

In this section, we can prove that if each phase can be identified individually, then we can build a representation that respect all of them. We show this by inductively applying Lachapelle et al. (2024a, Proposition 6). We state the preservation of a set of graphs, then its iterated form, and then specialize to the phase-conditioned setting.

6.6.1 Joint Preservation via Iterated Merging

Definition 6.6.1 (Graph family and $\mathcal{F}$-preservation)**.** A *graph family* on $[K]$ is a finite *set* of binary matrices

$$\mathcal{F} \;=\; \big\{G^{(1)}, G^{(2)}, \ldots, G^{(|\mathcal{F}|)}\big\}, \qquad G^{(r)} \in \{0,1\}^{K \times m_r},$$

where each $G^{(r)}$ has K rows and some number of columns m_r (which may vary across r). Duplicate graphs are identified; if two graphs in the family are equal as matrices, they are treated as a single element of the set.

A diffeomorphism $c : \mathcal{Z} \to \mathcal{Z}$ is *$\mathcal{F}$-preserving* if it is G-preserving (in the sense of Lachapelle et al. (2024a, Definition 12)) for every $G \in \mathcal{F}$ simultaneously. We write $\mathrm{Diff}_{\mathcal{F}}$ for the set of $\mathcal{F}$-preserving diffeomorphisms.

Set union of families versus union graph. The notation $\mathcal{F}_1 \cup \mathcal{F}_2$ throughout this chapter denotes the ordinary set-theoretic union of families: the set whose elements are the graphs appearing in either $\mathcal{F}_1$ or $\mathcal{F}_2$ (with duplicates collapsed). It is a set of several graphs, not a single combined graph. This is distinct from the *union graph* $G_\cup := \bigvee_\varphi G_\varphi$

introduced in Section 6.3, which is a single binary matrix obtained by entrywise logical OR of its constituents. The two operations produce different objects:

- $\mathcal{F}_1 \cup \mathcal{F}_2$: a *set* of graphs. A $(\mathcal{F}_1 \cup \mathcal{F}_2)$-preserving diffeomorphism preserves each graph in the set separately.
- $\bigvee_\varphi G_\varphi$: a *single* graph whose edges are the union of the constituents' edges. A $(\bigvee_\varphi G_\varphi)$-preserving diffeomorphism preserves only the zeros common to all constituents.

The former is a strictly stronger condition than the latter whenever the constituent graphs disagree on any edge — a phase-toggling edge in the phase-conditioned setting.

Remark 6.6.2 (Group structure). $\mathrm{Diff}_{\mathcal{F}} = \bigcap_{G \in \mathcal{F}} \mathrm{Diff}_G$ is an intersection of groups (Lachapelle et al. (2024a, Proposition 5) establishes each Diff_G as a group under composition), hence a group. The consistency relation $\sim^{\mathcal{F}}_{\mathrm{con}}$ defined below inherits its equivalence-relation structure from this group property: reflexivity (identity preserves every G), transitivity (compositions stay in the intersection), and symmetry (inverses stay in the intersection) all follow from standard group closure applied in each factor.

Lemma 6.6.3 (Merge of two preservation families)**.** *Let $v : \mathcal{Z} \to \mathcal{Z}$ be a diffeomorphism with two decompositions*

$$v \;=\; c_1 \circ P_1^\top \;=\; c_2 \circ P_2^\top, \tag{6.13}$$

where P_1, P_2 are permutation matrices and c_r is $\mathcal{F}_r$-preserving for $r \in \{1, 2\}$. Then there exist a permutation P_{12} and a diffeomorphism c_{12} such that

$$v \;=\; c_{12} \circ P_{12}^\top \qquad \textit{and} \qquad c_{12} \textit{ is } (\mathcal{F}_1 \cup \mathcal{F}_2)\textit{-preserving.} \tag{6.14}$$

Proof sketch. The argument reproduces the four steps of Lachapelle's proof of Proposition 6, with $\mathcal{F}_1, \mathcal{F}_2$ in place of $\{G^a\}, \{G^z, (G^z)^\top\}$.

Step 1. Apply Lemma 2 of Lachapelle et al. (2024a) to the invertible Jacobian of c_1 to obtain a permutation P_0 with $P_0^\top \subseteq C_1$ (where C_1 is the dependency graph of c_1).

Step 2. By Proposition 3 of Lachapelle et al. (2024a), $P_0^\top$ inherits $\mathcal{F}_1$-preservation from C_1. The composition $c_1 \circ P_0$ is $\mathcal{F}_1$-preserving by closure under composition.

Step 3. Combining $c_1 \circ P_1^\top = c_2 \circ P_2^\top$ with $P_0^\top \subseteq C_1$ produces $P_0^\top P_1^\top P_2 \subseteq C_2$, so this permutation is $\mathcal{F}_2$-preserving. Closure under composition promotes $c_1 \circ P_0$ to $\mathcal{F}_2$-preserving as well.

Step 4. Set $c_{12} := c_1 \circ P_0$ and $P_{12} := P_1 P_0$. Then $v = c_{12} \circ P_{12}^\top$ and c_{12} is $(\mathcal{F}_1 \cup \mathcal{F}_2)$-preserving by Steps 2–3.

Full details are in Appendix C.0.2. □

Proposition 6.6.4 (Iterated merge of n preservation families)**.** *Let $v : \mathcal{Z} \to \mathcal{Z}$ be a diffeomorphism with n decompositions*

$$v \;=\; c_k \circ P_k^\top, \qquad k \in [n], \tag{6.15}$$

where each c_k is $\mathcal{F}_k$-preserving and each P_k is a permutation. Then there exist a permutation $P_\star$ and a diffeomorphism $c_\star$ such that

$$v \;=\; c_\star \circ P_\star^\top \qquad \textit{and} \qquad c_\star \textit{ is } \big(\textstyle\bigcup_{k=1}^n \mathcal{F}_k\big)\textit{-preserving.} \tag{6.16}$$

Proof sketch. Induction on n. The base case $n = 1$ is trivial (set $c_\star = c_1$, $P_\star = P_1$). The case $n = 2$ is Lemma 6.6.3. For $n \geq 3$, apply the inductive hypothesis to $\{c_k, P_k\}_{k=1}^{n-1}$ to obtain $(c_{[n-1]}, P_{[n-1]})$ merging $\mathcal{F}_1 \cup \cdots \cup \mathcal{F}_{n-1}$. Apply Lemma 6.6.3 to the two decompositions $v = c_{[n-1]} \circ P_{[n-1]}^\top = c_n \circ P_n^\top$ (taking $\mathcal{F}_1 := \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_{n-1}$ and $\mathcal{F}_2 := \mathcal{F}_n$ in the lemma) to obtain $(c_\star, P_\star)$ merging all n families. Full proof in Appendix C.0.3. □

The proposition says nothing about phases specifically: it is an abstract statement about merging any n decompositions of the same diffeomorphism, each carrying its own preservation family. Phase-conditioned consistency is the special case where each family

encodes one phase's graph structure.

6.6.2 Phase-Conditioned Consistency

We instantiate Proposition 6.6.4 for phase-conditioned models. Fix phases $\Phi = \{1, \ldots, |\Phi|\}$ with per-phase graphs $\{(G^a_\varphi, G^z_\varphi)\}_{\varphi\in\Phi}$ and set

$$\mathcal{F}_\varphi \;:=\; \big\{G^a_\varphi,\; G^z_\varphi,\; (G^z_\varphi)^\top\big\}, \qquad \mathcal{F}_\Phi \;:=\; \bigcup_{\varphi\in\Phi} \mathcal{F}_\varphi. \tag{6.17}$$

Each $\mathcal{F}_\varphi$ is a set of (up to) three graphs — one action-type and two latent-type — and $\mathcal{F}_\Phi$ is the set-union of these phase-wise families across all phases. It contains at most $3|\Phi|$ graphs in total, with duplicates collapsed where distinct phases happen to share a graph.

Definition 6.6.5 (Phase-conditioned (a, z)-consistency)**.** Two phase-conditioned models $\theta_\Phi := \{(f, p_\varphi, G_\varphi)\}_\varphi$ and $\tilde\theta_\Phi := \{(\tilde f, \tilde p_\varphi, \tilde G_\varphi)\}_\varphi$ sharing decoders $f, \tilde f$ across phases are (a, z)-*consistent relative to* Φ, written $\theta_\Phi \sim^{a,z}_{\mathrm{con},\Phi} \tilde\theta_\Phi$, iff there exists a permutation P such that:

(i) For every $\varphi \in \Phi$: $\theta_\varphi \sim_{\mathrm{diff}} \tilde\theta_\varphi$, $\tilde G^a_\varphi = P^\top G^a_\varphi$, and $\tilde G^z_\varphi = P^\top G^z_\varphi P$.

(ii) The entanglement map $v := f^{-1} \circ \mathit{statedelsewhereinthischapterasAssumption}\tilde f$ satisfies $v = c \circ P^\top$ stated elsewhere in this chapter as Assumptionwith c being $\mathcal{F}_\Phi$-preserving.

Remark 6.6.6. $\sim^{a,z}_{\mathrm{con},\Phi}$ is an equivalence relation by the group structure of $\mathrm{Diff}_{\mathcal{F}_\Phi}$. It is strictly stronger than $\sim^{a,z}_{\mathrm{con}}$ applied to the union graph $G_\cup := \bigvee_\varphi G_\varphi$: the former requires preservation of every per-phase zero, the latter only the zeros common to all phases. The gap is nonzero whenever the phases have phase-toggling edges.

Corollary 6.6.7 (Phase-conditioned identifiability)**.** *Let $\theta_\Phi, \tilde\theta_\Phi$ be phase-conditioned models with shared decoders. Assume:*

(a) Per-phase observational equivalence: $\theta_\varphi \sim_{\mathrm{obs}} \tilde\theta_\varphi$ for every $\varphi \in \Phi$.

(b) Per-phase sufficient influence: for each $\varphi \in \Phi$, the transition p_φ satisfies Assumptions 4.2.3 on supp_φ.

(c) Per-phase sparsity: $\|\tilde{G}^a_\varphi\|_0 \le \|G^a_\varphi\|_0$ and $\|\tilde{G}^z_\varphi\|_0 \le \|G^z_\varphi\|_0$ for every $\varphi \in \Phi$.

Then $\theta_\Phi \sim^{a,z}_{\mathrm{con},\Phi} \tilde{\theta}_\Phi$.

Proof. Per-phase application of Theorems 1 (or 2) and 3 of Lachapelle et al. (2024a), combined via Proposition 6 of the same reference, yields $\theta_\varphi \sim^{a,z}_{\mathrm{con}} \tilde{\theta}_\varphi$ for each φ. Each relation provides a decomposition $v = c_\varphi \circ P_\varphi^\top$ with c_φ being $\mathcal{F}_\varphi$-preserving. Apply Proposition 6.6.4 with $\{(c_\varphi, P_\varphi, \mathcal{F}_\varphi)\}_{\varphi\in\Phi}$ to obtain a single $(c_\star, P_\star)$ with $c_\star$ being $\mathcal{F}_\Phi$-preserving. The graph-alignment conditions $P_\star^\top \tilde{G}^a_\varphi = G^a_\varphi$ and $P_\star^\top \tilde{G}^z_\varphi P_\star = G^z_\varphi$ follow from the inductive construction in Appendix C.0.3 (Step 4, generalized). □

6.6.3 Nash-ACF Achieves the Hypotheses

Corollary 6.6.7 takes per-phase sparsity and observational equivalence as inputs. Chapter 4 produces both for Nash-ACF maximizers.

Corollary 6.6.8 (Phase-conditioned Nash-ACF identifies up to $\sim^{a,z}_{\mathrm{con},\Phi}$)**.** *Let $\hat{\theta}_\Phi$ be a maximizer of the phase-conditioned Nash product (Equation 6.11), and let θ_Φ be the ground truth. Assume per-phase sufficient influence (condition (b) of Corollary 6.6.7). Then $\theta_\Phi \sim^{a,z}_{\mathrm{con},\Phi} \hat{\theta}_\Phi$.*

Proof. By Theorem 4.4.2 (Chapter 4) applied per phase, maximizing Π_φ enforces $\hat{\theta}_\varphi \sim_{\mathrm{obs}} \theta_\varphi$ and $\|\hat{G}^a_\varphi\|_0 \le \|G^a_\varphi\|_0$, $\|\hat{G}^z_\varphi\|_0 \le \|G^z_\varphi\|_0$ simultaneously for every φ. Since Equation 6.11 factorizes across phases, maximizing the per-phase NBS is equivalent to maximizing every phase sub-problem simultaneously, which is feasible because the ground-truth θ_Φ is a joint maximizer. Hence conditions (a) and (c) of Corollary 6.6.7 hold, and the claim follows. □

6.6.4 Complete Disentanglement via Multi-Phase Criterion

Phase-conditioned (a, z)-consistency is partial disentanglement in general. Complete disentanglement requires the preservation family $\mathcal{F}_\Phi$ to pin down the entanglement map up to permutation and element-wise transformation. The following criterion, generalizing Lachapelle et al. (2024a, Assumption 5) to a family of graphs, does exactly this.

Definition 6.6.9 (Multi-phase graphical criterion)**.** The family $\{G_\varphi\}_{\varphi\in\Phi}$ satisfies the *multi-phase graphical criterion* if for every pair $i \neq j \in [K]$, there exists a phase $\varphi \in \Phi$ and an index ℓ (either a latent index or an action index) such that

$$(G_\varphi)_{i,\ell} \neq (G_\varphi)_{j,\ell} \quad \text{or} \quad (G^z_\varphi)_{\ell,i} \neq (G^z_\varphi)_{\ell,j}, \tag{6.18}$$

i.e., some phase provides a structural distinction between i and j via row or column supports of its graph.

Corollary 6.6.10 (Complete disentanglement)**.** *Under the hypotheses of Corollary 6.6.8, if the family $\{G_\varphi\}_{\varphi\in\Phi}$ satisfies the multi-phase graphical criterion, then $\hat{\theta}_\Phi \sim_{\text{perm}} \theta_\Phi$, i.e., phase-conditioned Nash-ACF completely disentangles the representation.*

Proof. By Corollary 6.6.8, $\theta_\Phi \sim^{a,z}_{\text{con},\Phi} \hat{\theta}_\Phi$, giving $v = c \circ P^\top$ with c being $\mathcal{F}_\Phi$-preserving. By Proposition 3 of Lachapelle et al. (2024a) applied to each $G \in \mathcal{F}_\Phi$, the multi-phase criterion implies that for every $i \neq j$ some graph $G \in \mathcal{F}_\Phi$ has $G_{i,\cdot} \not\subseteq G_{j,\cdot}$, forcing $C_{i,j} = 0$ in the dependency graph of c. Hence c has diagonal dependency graph, i.e., c is an element-wise transformation, and $v = c \circ P^\top$ is a permutation composed with an element-wise transformation. This is precisely $\sim_{\text{perm}}$ (Lachapelle Def. 6). □

Table 6.1: Nash-ACF and NMS-ACF on Taxi ($K = 6$): corrected nonlinear R^2 metrics per checkpoint seed, compared against the ACF encoder. Best hyperparameter random search over 5 seed.

Method	Metric	seed 106	seed 107	seed 108	seed 110	Mean
ACF	$\bar{d}\uparrow$	0.680	0.541	0.459	0.609	$0.602_{\pm.094}$
	$\bar{o}\downarrow$	0.004	0.014	**0.002**	0.009	$0.006_{\pm.005}$
	$o_{\max}\downarrow$	0.107	0.242	**0.050**	0.185	$0.122_{\pm.082}$
Nash-ACF	$\bar{d}\uparrow$	0.697	**0.636**	**0.508**	**0.633**	$0.619_{\pm.069}$
	$\bar{o}\downarrow$	0.015	0.021	0.004	0.025	$0.016_{\pm.008}$
	$o_{\max}\downarrow$	0.180	0.181	0.068	0.234	$0.166_{\pm.061}$
NMS-ACF	$\bar{d}\uparrow$	**0.717**	0.631	0.496	0.626	$\mathbf{0.646}_{\pm.090}$
	$\bar{o}\downarrow$	**0.004**	**0.005**	**0.002**	**0.003**	$\mathbf{0.003}_{\pm.001}$
	$o_{\max}\downarrow$	**0.110**	**0.058**	**0.064**	**0.057**	$\mathbf{0.069}_{\pm.020}$

6.7 Evaluation

6.7.1 Domains

Visual Taxi Taxi domain from Chapter 3.

cube-double-noisy and cube-triple-noisy Two cubes and three cubes manipulation enviorments from OGBench (Park et al., 2024). Ground-truth factors: end-effector (3D), cube 1 (3D), cube 2 (3D), cube 3 (3D). Phases include free motion, grasping cube 1, and grasping cube 2. This is harder because the two cubes are symmetric and must be distinguished by their independent dynamics.

6.7.2 Metrics

We use the block R^2 metric introduced in Chapter 4, adapted to the multi-dimensional factor setting. For each pair of learned latent block z_{block_i} and ground-truth factor block s_{block_j}, we fit a nonlinear regressor and compute R^2, corrected for ground-truth correlations. After Hungarian alignment, we report the corrected diagonal (how well each latent block captures its matched factor) and the maximum off-diagonal (worst-case cross-contamination between factors).

For the two-cube domain, we additionally report the *object cross-contamination*: the maximum R^2 between the two cube factors' learned representations, which should be zero if the cubes are properly disentangled.

6.7.3 Results

We empirically validate that phase awareness can improve a learned encoder. We start from an ACF-learned encoder, freeze the backbone, and attach a phase-aware adapter trained with the Nash product. Table 6.1 reports the result across five seeds and compare with respect to the results of pure Nash-ACF.

The adapter improves the diagonal R^2 and reduces the maximum off-diagonal. The phase awareness avoids collapsing to the union graph where the passenger and taxi are strongly correlated.

Table 6.2 show the results of NMS-ACF applied to the cube domains. These domains receive the observations nonlinearly mixed by a random MLP. Similar to Taxi, the results show that phase information improves the factorization by reducing the entanglement of the cubes and end effector in the representation.

Table 6.2: ACF baseline and NMS-ACF refinement (mean ± std).

Domain	**Metric**	**ACF**	**+ NMS-ACF**	**Improvement (%)**
cube-double-noisy	$\bar{d} \uparrow$	0.358 ± 0.047	0.368 ± 0.101	2.7%
	$\bar{o} \downarrow$	0.057 ± 0.021	0.019 ± 0.014	66%
	$o_{\max} \downarrow$	0.187 ± 0.083	0.054 ± 0.040	72.0%
cube-triple-noisy	$\bar{d} \uparrow$	0.211 ± 0.085	0.254 ± 0.060	20%
	$\bar{o} \downarrow$	0.006 ± 0.004	0.001 ± 0.001	83%
	$o_{\max} \downarrow$	0.043 ± 0.030	0.014 ± 0.010	64.4%

6.8 Closing Thoughts

The core of this thesis has been: the agent must act to discover what its world is made of. ACF gave the agent a way to discover *what* it controls. Nash-ACF gave it a principled

way to balance the multiple sparsity signals that this discovery generates. This chapter pushes the same idea further. The agent's behavior is not just a sequence of primitive actions; it is a sequence of *phases*, and each phase give us new structural knowledge about the world's dynamics. The agent grasps a cube, and the act of grasping reveals a coupling free-motion data couldn't reveal.

The framework presented here is the natural composition of the previous chapters with one additional ingredient: the recognition that a single global DBN is the wrong abstraction for an agent that interacts with a structured world. Phases are the right abstraction. They are interventional, sparse, state-determined, and grounded in the agent's own causal structure. The multi-time energy model gives us the temporal resolution to see slow factors; the phase-conditioned NBS gives us the structural resolution to see how those factors couple and decouple. The identifiability theory—iterated graph preservation across a family of phase-specific graphs—is the formal closure of the loop: it explains why the phase-conditioned Nash maximizer cannot collapse into the wrong solution.

What remains open is the discovery of phases themselves. We assumed access to phase labels in this chapter; in practice, an agent must segment its own experience into regimes without supervision. Closing this gap— learning the phases jointly with the factorization—is the natural next step, and one of the directions taken up in Chapter 7.

More broadly, this chapter is the place where the thesis crosses from primitive control into structured behavior. The same energy-based machinery that recovered the end-effector from torque commands, scaled with the right inductive biases, recovers the cube from a grasp.

CHAPTER 7

Looking Forward

This thesis started from a simple observation: an agent that acts in the world is not a passive observer. Its actions are interventions on the dynamics, and the structure those interventions reveal is not arbitrary; it reflects the causal mechanisms governing the world. We developed this observation into a framework for learning structured representations for sequential decision making, progressing from primitive actions to temporally-extended skills, and from single-step factorization to multi-phase cooperative sparsity.

Chapter 3 showed that contrasting the effect of an action against the natural dynamics isolates the independently controllable factors. The ratio of interventional to observational transition densities is sparse, because each action affects only a few factors, and the contrastive energy-based model recovers this sparsity. The key insight was that the agent's interventional knowledge is grounded: the sparsity of action effects is tied to the causal structure of the world and cannot be explained away by a change of representation.

Chapter 4 extended this to the full dependency structure. ACF identifies the independently controllable factors but doesn't minimize its dependence on other factors. The cross-Hessian of the transition function captures which factors influence which, but this temporal sparsity is observational and ambiguous; a powerful encoder can always rotate

to the eigenbasis, achieving perfect temporal sparsity in the wrong basis. We showed that the ground-truth factored representation is the unique Pareto optimum of both action sparsity and temporal sparsity, and that the Nash Bargaining Solution is the optimization principle that finds it without letting either signal be sacrificed.

Chapter 5 showed a complementary axis: temporally-extended skills compress state to the slow variables relevant for planning. The abstract representation retains only what is needed to predict a skill's effect, discarding the high-frequency detail. The result is a minimal representation sufficient for value-preserving planning with options. Thus allowing to build a continuous abstract world model, abstract in time and in state.

Chapter 6 unified both threads. Phase transitions in manipulation are structural mechanism shifts: they extend the agent's control to new factors (grasping lets you move the cube) and reveal the coupling structure between factors (arm and cube become a rigid body). Multi-time energy models probe the dynamics at multiple temporal resolutions, revealing slow factors invisible to single-step methods. The four-player NBS balances all signals across phases and timescales, recovering the factored structure that neither single-step methods nor monolithic abstractions can achieve.

The common thread across all chapters is that the agent's actions, at every temporal scale, are the source of structural knowledge about the world. The structure is not imposed by the designer; it is discovered through interaction. And it is the right structure for planning, because the causal mechanisms that actions reveal are the same mechanisms that planning must reason about.

7.1 Looking Forward

The framework developed in this thesis opens two families of questions. In the near term: how can the agent use the structure it has learned to discover *more* structure? This is the discovery cycle: act to learn structure, use structure to act better, act better to

learn more structure. In the long term: how can the discovered structure serve as the foundation for compositional reasoning, enabling agents to generalize across tasks and environments?

7.1.1 Structure Discovery as Intrinsic Motivation

The combined variability condition (Assumption 4.2.3) tells us what the agent needs to identify the factored structure: sufficient diversity in its interventions, and sufficient coverage of mechanism shifts across phases. In the current framework, this coverage is assumed to exist in the data. But an acting agent can seek it out.

This connects to a broader vision of intrinsic motivation. Most curiosity-driven exploration methods optimize for prediction error or information gain about the dynamics. But prediction error does not distinguish between dynamics that are hard to predict because the model is wrong and dynamics that are hard to predict because the environment is stochastic. Structural uncertainty is more targeted: the agent seeks transitions that reveal new edges in the DBN, new phase transitions that activate new couplings, new actions that differentiate previously indistinguishable factors. The curiosity is about the *structure* of the world, not just its content.

From a practical standpoint, the agent should learn skills that maximize structural coverage. A good skill, in this view, is one that creates a mechanism shift: it takes the system from one phase to another, activating new couplings and extending the agent's control to new factors. Skill discovery and structure discovery become the same problem. The skills that are useful for planning are exactly the skills that produce the interventional and mechanism-shift signals needed for identification. The agent learns skills to discover structure, and uses the discovered structure to learn better skills.

7.1.2 Scaling to Large World Models

This thesis developed action-driven identifiability at the scale where its theoretical claims could be tested cleanly: pixel-RL benchmarks, manipulation environments with few objects, and small-to-mid model architectures. The natural question is whether the framework remains relevant at the scale of contemporary Vision-Language-Action models and large-scale world models. We believe it does, and that the empirical record of the last two years has sharpened the case rather than weakened it.

The dominant recipe at scale is "scale plus diversity": pretrain a large vision-language backbone, graft an action head, train on as much heterogeneous robot data as can be pooled. This recipe is now hitting structural failure modes. Naively pooling data across embodiments produces negative transfer rather than gains (Wang et al., 2026). Within-environment performance saturates in demonstration count, while the real lever is environment and object diversity (Lin et al., 2025). State-of-the-art VLAs collapse to near-zero on previously unseen embodiments (Zha et al., 2026). The common pattern is that the learned representation conflates what the agent can control with what is embodiment-specific or distractor-correlated, and observational training has no way to disambiguate the two. This is a non-identifiability, not a coverage problem; the recipe will not fix it by becoming larger.

The latent-action VLA family makes the diagnosis precise. LAPO (Schmidt and Jiang, 2024), LAPA (Ye et al., 2025), villa-X (Chen et al., 2026), and Genie (Bruce et al., 2024) all place a discrete bottleneck between an inverse and forward dynamics model, treating the bottleneck as the agent's action. Lachapelle (2025) recently proved that this bottleneck recovers the true action up to permutation, but only when the dynamics are deterministic, injective, and free of distractors. Empirical settings violate these assumptions routinely, and Nikulin et al. (2025) showed exactly the predicted breakdown: under action-correlated distractors, the latent collapses onto the distractor and downstream success drops by an order of magnitude. The repair is the same as in our setting. The agent's action is an

intervention; the controllable factors should respond to it, and the exogenous factors should not. Replacing the monolithic latent with an action-indexed factorization $z = (z_c, z_e)$ and enforcing mechanism sparsity on the action-Jacobian extends Lachapelle's identifiability guarantee past the distractor-free regime. This is ACF at VLA scale.

A second insertion point exists in the non-generative world-model line, of which V-JEPA 2 (Assran et al., 2025) is the strongest current instance. V-JEPA 2 freezes a one-billion-parameter video encoder and post-trains a small action-conditioned predictor on 62 hours of robot data. The predictor is a six-layer transformer with frame-causal attention; it consumes per-frame visual tokens with action, state, and extrinsic tokens prepended, and predicts the next-frame visual tokens via ℓ_1 loss in encoder feature space. The architecture is ready-made for our framework. The two sparsities central to Nash-ACF, action-locality and temporal-dependency, can be encoded directly in the predictor's attention mask: the attention pattern *is* the dependency graph. This formulation does not require touching the large encoder, which keeps the experiment within an academic compute budget.

The most direct empirical step is a controlled study of identifiability under exogenous dynamics, building on the Distracting Control Suite results of Nikulin et al. (2025) and extending to manipulation benchmarks at LIBERO-class scale. The setup is surgical: same encoder, same data, same training schedule, with the only difference being an ACF-structured factorization and the mechanism-sparsity regularizer. The framework predicts the modified system should match the latent-action baseline in the regime where Lachapelle's theorem already applies, and outperform it as exogenous-noise strength increases. A claim about VLA fragility becomes a falsifiable prediction.

The broader conjecture is that the embodied scaling hypothesis is incomplete in a way the language case obscures. Language tokens arrive pre-equipped with compositional syntax, which supplies the auxiliary structure that nonlinear-ICA identifiability requires; embodied perception does not. Scaling alone cannot supply this structure. Action-driven identifiability is what supplies it. Demonstrating this at the scale where the prevailing

recipe breaks is the natural target for the next phase of the program.

7.1.3 Modular World Models

The factored energy model is a step toward modular world models, where each factor's transition mechanism is an independent module that can be composed, reused, and transferred.

In the current framework, each per-factor energy head E_i already functions as a module: it captures the dynamics of factor i conditioned on its parents and the current phase. The sparsity of the DBN means each module depends on only a few other factors. The phase conditioning means the module's behavior changes across regimes in a structured way. In principle, an agent that has learned the modules for "arm dynamics," "cube dynamics under contact," and "door dynamics given key" can compose them to reason about new situations: a new cube, a different door, a rearranged room.

Skills to Symbols (Konidaris et al., 2018; Ahmetoglu et al., 2025) showed that we can automatically learn PDDL operators from skills, but they require access to well-designed representation to begin with. Our own Chapter 5, extended the Skills to Symbols idea to the continuous case, by generalizing the Subgoal condition. This dissertation opens up the possibility of scaling these approaches to problems with high-dimensional observation.

The open challenge is composition at test time. The current model is trained on a fixed set of phases and factors. A truly modular world model would allow the agent to encounter a new object, learn its dynamics as a new module, and plug it into the existing model without retraining the other modules. This requires the modules to have compatible interfaces, the per-factor energy architecture provides a natural interface (the shared latent space and the phase embedding). Lifelong learning (Mendez and Eaton, 2021) has studied this class of compositional structure for varied learning problems. This dissertation gives a concrete way we can do this for SDM agents.

The long-term vision is an agent that builds its world model the way we build with LEGO: standardized pieces that snap together in new configurations. The pieces are learned, not designed, and the structure that determines how they connect is discovered through the agent's own interventions.

7.1.4 Language Grounding

The factored structure the agent discovers has a compositional organization: independently identifiable factors, skills with sparse effects, phases with structured mechanism shifts. Natural language has the same kind of compositionality: nouns refer to objects, verbs refer to actions, adjectives refer to properties, and sentences describe state transitions (Rodriguez-Sanchez et al., 2020)

This suggests that the learned factored representation provides a natural grounding space for language. Rather than learning a mapping from words to a monolithic embedding, the agent maps words to the components of its own internal model. "Cube" grounds to a factor. "Push" grounds to a skill whose effect set includes that factor. "Red" grounds to a value of a factor's property. "Push the red cube to the left" grounds to: activate the skill "push," applied to the factor matching "red cube," with the directional subgoal "left."

The compositional alignment between language and the factored model has a practical consequence for generalization. If the agent has learned "push the red cube" and "grasp the blue cube" separately, it should be able to understand "push the blue cube" without new training, because "push" and "blue cube" are compositional references to independently learned components.

This kind of compositional generalization is notoriously difficult for end-to-end models that lack internal structure, but it falls naturally out of a factored representation where skills and factors combine independently. LLMs (Large Language Models) and VLA (Vision Language Action models) exhibit these types of generalization. However, they require massive amounts of data to achieve this. Moreover, data is curated and artificially

structured to ensure correct learning—e.g., aligned datasets of language, videos, and actions—something that truly general and independent agents might not have.

Perhaps, most interesting is the question of grounding and language emergence in artificial agents (Spiegel et al., 2025a). If language emerges from a need to communicate, having a unified representation grounded in the causal structure of the world provides the initial common set of structures agents can agree on.

The deeper question is whether the relationship goes both ways. Can language *accelerate* structure discovery by providing prior knowledge about what the factors should be? A human telling the agent "there are two cubes on the table" is, in effect, communicating the number of factors before the agent has discovered them. A human saying "the key opens the door" is communicating a DBN edge. If the agent's learned structure is compatible with linguistic descriptions, then language becomes a channel for transferring structural knowledge, dramatically reducing the amount of interaction needed to build a world model. RLang (Rodriguez-Sanchez et al., 2023; Spiegel et al., 2025b), a domain-specific language for RL, explored this very question. Now, we can work on discovering the very grounding structure assumed by these works.

7.1.5 Causal Discovery and Active Experimentation

The framework in this thesis is, at its core, a causal discovery method (Pearl, 2009; Peters et al., 2017; Schölkopf et al., 2021): the agent's actions are interventions, the phase transitions are mechanism shifts, and the factored representation is the causal graph (variables and edges). We have shown that this framing leads to identifiability guarantees and principled optimization. But we have only scratched the surface of the connection to the causal inference literature.

Active causal discovery asks: given a partially known causal graph, which experiment should the agent perform next to maximally reduce its uncertainty about the graph structure? In our framework, this translates directly: given a partially recovered DBN

(some structural zeros confirmed, others uncertain), which action or skill should the agent execute to provide the missing interventional or mechanism-shift coverage?

This also connects to the question of sample efficiency. The current framework trains on a fixed dataset. An agent that actively seeks structurally informative transitions, rather than collecting data uniformly, could discover the factored structure with dramatically fewer interactions.

There is also a deeper theoretical connection to Pearl's do-calculus and the front-door criterion that we have not yet explored. The agent's phase transitions resemble soft interventions: they change some mechanisms while preserving others. The conditions under which the factored structure is identifiable from a mix of observational data and soft interventions are studied in the causal inference literature, and connecting these results to the RL setting could strengthen the theoretical foundations of the approach.

7.1.6 Partial Observability and Memory

The framework in this thesis assumes that the agent's observations are generated by a diffeomorphism from the true state: everything is visible, nothing is hidden. This is a strong assumption that real-world agents violate constantly. Objects go behind other objects. Relevant state variables (battery charge, joint wear, the contents of a closed drawer) are not visible in the current observation. The agent must maintain beliefs about hidden state and update them over time.

From the causal perspective, partial observability introduces hidden confounders in the causal graph. If a hidden variable h influences both factor s_i and factor s_j, the agent observes a correlation between s_i and s_j that cannot be explained by any edge in the observed DBN. The factorization framework, which assumes the observed variables are the complete set of factors, would incorrectly infer a direct dependency between s_i and s_j. Detecting and handling hidden confounders is a fundamental challenge.

One direction is to extend the energy model to operate on beliefs rather than observations. If the agent maintains a belief state $b_t = p(s_t \mid o_{1:t}, a_{1:t-1})$ via a recurrent encoder or a state-space model, the factored energy could be defined over the belief: $E(b'_i, b, a, \varphi, k)$. The question is whether the identifiability results extend to this setting. The combined variability condition would need to hold in belief space, which is a much richer and more complex space than the state space. It is unclear whether the agent's interventions provide sufficient signal to identify factors in belief space, or whether additional assumptions (e.g., temporal persistence of hidden factors, sparsity of the belief update) are needed.

Another direction draws on the causal inference literature. The front-door criterion provides a way to identify causal effects even in the presence of hidden confounders, by finding mediating variables that block the confounding path. In the RL setting, the agent's actions might serve as mediators: the action is caused by the agent (not by the hidden confounder) and its effect on the next state is observable. Whether this reasoning can be formalized to provide identifiability of factored representations under partial observability is an open theoretical question.

This is the least developed direction in this thesis, and perhaps the most important for real-world deployment. An agent that can discover factored structure only when everything is visible is fundamentally limited. Extending the framework to partial observability is necessary for the vision of a general-purpose agent that learns the structure of its world through interaction.

7.2 Closing Remarks

This thesis has argued that the structure an agent needs for planning is discovered, not imposed. The agent's interventions, from primitive actions to temporally-extended skills, reveal the independently controllable factors of the world, the dependencies between them, and the phases under which those dependencies change. The Nash Bargaining framework

ensures that every signal, whether abundant or rare, contributes to this discovery.

The directions outlined above, from skill discovery to modular world models to language grounding, all build on the same foundation: the agent's ability to intervene in the world generates structural knowledge, and that structure enables increasingly sophisticated reasoning. In the near term, structure drives exploration; the agent learns skills to discover more structure, and uses structure to learn better skills. In the long term, structure enables composition; the agent assembles modular world models, grounds language in its own experience, and plans with symbols it discovered rather than symbols it was given.

The general-purpose agent we envision does not start with a model of the world. It builds one, piece by piece, through its own actions. The pieces are the factors, the wiring is the DBN, and the blueprint is the causal structure that the agent's interventions make visible. The thesis has shown that this construction is possible. The question now is how far it can go.

References

David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In *International Conference on Machine Learning*, pages 2915–2923. PMLR, 2016.

David Abel, Dilip Arumugam, Lucas Lehnert, and Michael Littman. State abstractions for lifelong reinforcement learning. In *International Conference on Machine Learning*, pages 10–19. PMLR, 2018.

David Abel, John Winder, Marie DesJardins, and Michael Littman. The expected-length model of options. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 1951–1958, 2019.

David Abel, Nate Umbanhowar, Khimya Khetarpal, Dilip Arumugam, Doina Precup, and Michael Littman. Value preserving state-action abstractions. In *International Conference on Artificial Intelligence and Statistics*, pages 1639–1650. PMLR, 2020.

Alper Ahmetoglu, Steven James, Cameron Allen, Sam Lobel, David Abel, and George Konidaris. Skill-driven neurosymbolic state abstractions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=ggP2r9S9wY`.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Cameron Allen, Neev Parikh, Omer Gottesman, and George Konidaris. Learning Markov state abstractions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8229–8241, 2021.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-JEPA

2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(4):341–379, 2003.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

William M Boothby. *An introduction to differentiable manifolds and Riemannian geometry, Revised*, volume 120. Gulf Professional Publishing, 2003.

Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

Craig Boutilier and Richard Dearden. Approximate value trees in structured dynamic programming. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, pages 54–62, 1996.

Craig Boutilier, Richard Dearden, Moisés Goldszmidt, et al. Exploiting structure in policy construction. In *IJCAI*, volume 14, pages 1104–1113, 1995.

Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial intelligence*, 121(1-2):49–107, 2000.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct): 213–231, 2002.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365 (6456):885–890, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=bJbSbJskOS`.

Lars Buesing, Theophane Weber, Sébastien Racaniere, SM Eslami, Danilo Rezende, David P Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, et al.

Learning and querying fast generative models for reinforcement learning. *arXiv preprint arXiv:1802.03006*, 2018.

Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2017.

Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep Blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

Pablo Castro and Doina Precup. Using bisimulation for policy transfer in MDPs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pages 1065–1070, 2010.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, Jianyu Chen, and Jiang Bian. villa-X: Enhancing latent action modeling in vision-language-action models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=y5CaJb17Fn`.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & Miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Caleb Chuck, Kevin Black, Aditya Arjun, Yuke Zhu, and Scott Niekum. Granger causal interaction skill chains. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=iA2KQyoun1`.

Caleb Chuck, Fan Feng, Carl Qi, Chang Shi, Siddhant Agarwal, Amy Zhang, and Scott

Niekum. Null counterfactual factor interactions for goal-conditioned reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3): 287–314, 1994.

Thomas Degris, Olivier Sigaud, and Pierre-Henri Wuillemin. Learning the structure of factored Markov decision processes in reinforcement learning problems. In *Proceedings of the 23rd international conference on Machine learning*, pages 257–264, 2006.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

Thomas G Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=HkpbnH9lx`.

Carlos Diuk, Lihong Li, and Bethany R Leffler. The adaptive k-meteorologists problem and its application to structure learning and feature selection in reinforcement learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 249–256, 2009.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina

Balcan, and Hsuan-Tien Lin, editors, *Annual Conference on Neural Information Processing Systems*, 2020.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=tc5qisoB-C`.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, page 162–169, Arlington, Virginia, USA, 2004. AUAI Press. ISBN 0974903906.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous Markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model

minimization in Markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pages 2424–2433. PMLR, 2019.

Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, and Theophane Weber. Temporal difference variational auto-encoder. In *International Conference on Learning Representations*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Carlos Guestrin, Relu Patrascu, and Dale Schuurmans. Algorithm-directed exploration for model-based reinforcement learning in factored MDPs. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 235–242, 2002.

Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19: 399–468, 2003.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolu-

tion. In *Advances in Neural Information Processing Systems*, pages 2450–2462. Neural information processing systems foundation, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations*, 2021a.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations*, 2021b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pages 1–7, 2025.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models

for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017a. URL `https://openreview.net/forum?id=Sy2fzU9gl`.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pages 1480–1490. PMLR, 2017b.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL `https://arxiv.org/abs/2207.12598`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004.

Jiaheng Hu, Zizhao Wang, Peter Stone, and Roberto Martín-Martín. Disentangled unsupervised skill discovery for efficient hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 37:76529–76552, 2024.

Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. *Advances in neural information processing systems*, 29, 2016.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Aapo Hyvärinen, Ilyes Khemakhem, and Hiroshi Morioka. Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning. *Patterns*, 4 (10), 2023.

Aapo Hyvärinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2016.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Nan Rosemary Ke, Aniket Rajiv Didolkar, Sarthak Mittal, Anirudh Goyal, Guillaume Lajoie, Stefan Bauer, Danilo Jimenez Rezende, Yoshua Bengio, Christopher Pal, and Michael Curtis Mozer. Systematic evaluation of causal discovery in visual model based reinforcement learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=gWIbXsrtOCc`.

Michael Kearns and Daphne Koller. Efficient reinforcement learning in factored MDPs. In *Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2*, pages 740–747, 1999.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49:209–232, 2002.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Daphne Koller and Ronald Parr. Policy iteration for factored MDPs. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 326–334, 2000.

G.D. Konidaris and A.G. Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1015–1023, 2009.

George Konidaris. On the necessity of abstraction. *Current opinion in behavioral sciences*, 29:1–7, 2019.

George Konidaris and Sarah Osentoski. Value function approximation using the Fourier basis. In *the Multidisciplinary Symposium on Reinforcement Learning, Montreal, Canada*, 2009.

George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.

Lyudmyla F Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

Rahul G Krishnan, Uri Shalit, and David Sontag. Deep Kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing: System demonstrations*, pages 66–71, 2018.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Sébastien Lachapelle. On the identifiability of latent action policies. *arXiv preprint arXiv:2510.01337*, 2025.

Sébastien Lachapelle, Pau Rodriguez, Yash Sharma, Katie E Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In *Conference on Causal Learning and Reasoning*, pages 428–484. PMLR, 2022.

Sébastien Lachapelle, Pau Rodriguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Nonparametric partial disentanglement via mechanism sparsity: Sparse actions, interventions and sparse temporal dependencies. *arXiv preprint arXiv:2401.04890*, 2024a.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien.

Additive decoders for latent variables identification and Cartesian-product extrapolation. *Advances in Neural Information Processing Systems*, 36, 2024b.

Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 29–37. JMLR Workshop and Conference Proceedings, 2011.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pages 5639–5650. PMLR, 2020.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John Canny, and Ian Fischer. Compressive visual representations. *Advances in Neural Information Processing Systems*, 34:19538–19552, 2021.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for MDPs. volume 1, page 3, 2006.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *4th International Conference on Learning Representations*, 2016.

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=pISLZG7ktL`.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Confer-*

ence on Learning Representations, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Phillip Lippe, Taco Cohen, and Efstratios Gavves. Efficient neural causal discovery without acyclicity constraints. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=eYciPrLuUhG.

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Stratis Gavves. CITRIS: Causal identifiability from temporal intervened sequences. In *International Conference on Machine Learning*, pages 13557–13603. PMLR, 2022b.

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Efstratios Gavves. BISCUIT: Causal representation learning from binary interactions. In *Uncertainty in Artificial Intelligence*, pages 1263–1273. PMLR, 2023.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International conference on machine learning*, pages 6348–6359. PMLR, 2020a.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(10), 2007.

Jorge A Mendez and Eric Eaton. Lifelong learning of compositional structures. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=ADWd4TJO13G`.

Meta Fundamental AI Research Diplomacy Team (FAIR), Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7559–7566. IEEE, 2018.

John F Nash et al. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

Tung D Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal predictive coding for model-based planning in latent space. In *International Conference on Machine Learning*, pages 8130–8139. PMLR, 2021.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Lyubaykin Nikita, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=2gcEQCT7QW`.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018a.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018b.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored MDPs. *Advances in Neural Information Processing Systems*, 27, 2014.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned RL. *arXiv preprint arXiv:2410.20092*, 2024.

Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*, 10, 1997.

Judea Pearl. *Causality*. Cambridge university press, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jonas Peters, Dominik Janzing, and Bernhard Scholkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

Eduardo Pignatelli, Jarek Liesen, Robert Tjarko Lange, Chris Lu, Pablo Samuel Castro, and Laura Toni. NAVIX: Scaling MiniGrid environments with JAX. *arXiv preprint arXiv:2407.19396*, 2024.

Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems*, 33: 3976–3990, 2020.

Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. Mocoda: Model-based

counterfactual data augmentation. *Advances in Neural Information Processing Systems*, 35:18143–18156, 2022.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN 978-0-47161977-2. doi: 10.1002/9780470316887.

Anil V Rao. A survey of numerical methods for optimal control. *Advances in the astronautical Sciences*, 135(1):497–528, 2009.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Rafael Rodriguez-Sanchez and George Konidaris. Learning abstract world models for value-preserving planning with options. *Reinforcement Learning Journal*, 4:1733–1758, 2024.

Rafael Rodriguez-Sanchez, Roma Patel, and George Konidaris. On the relationship between structure in natural language and models of sequential decision processes. In *Language in Reinforcement Learning Workshop at ICML 2020*, 2020.

Rafael Rodriguez-Sanchez, Benjamin Adin Spiegel, Jennifer Wang, Roma Patel, Stefanie Tellex, and George Konidaris. Rlang: A declarative language for describing partial world knowledge to reinforcement learning agents. In *International Conference on Machine Learning*, pages 29161–29178. PMLR, 2023.

DE Rumelhart, GE Hinton, and RJ Williams. Learning internal representations by error propagation. In *Neurocomputing: foundations of research*, pages 673–695. 1988.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *The Twelfth In-*

ternational Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=rvUq3cxpDF.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:22905–22918, 2021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725, 2016.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021. ISSN 0004-3702.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Benjamin A Spiegel, Lucas Gelfond, and George Konidaris. Visual theory of mind enables the invention of proto-writing. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47, 2025a.

Benjamin Adin Spiegel, Ziyi Yang, William Jurayj, Ben Bachmann, Stefanie Tellex, and George Konidaris. Informing reinforcement learning agents by grounding language to Markov decision processes, 2025b. URL `https://openreview.net/forum?id=1EEst6oDU7`.

Alexander L. Strehl, Carlos Diuk, and Michael L. Littman. Efficient structure learning in factored-state MDPs. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, page 645–650. AAAI Press, 2007. ISBN 9781577353232.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in neural information processing systems*, 8, 1995.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning - An Introduction*.

Adaptive Computation and Machine Learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL http://www.incompleteideas.net/book/first/the-book.html.

Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, 2011.

Gerald Tesauro et al. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Yi Tian, Jian Qian, and Suvrit Sra. Towards minimax optimal reinforcement learning in factored Markov decision processes. *Advances in Neural Information Processing Systems*, 33:19896–19907, 2020.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

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.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Burak Varici, Emre Acartürk, Karthikeyan Shanmugam, and Ali Tajer. General identifia-

bility and achievability for causal representation learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2314–2322. PMLR, 2024.

Burak Varıcı, Emre Acartürk, Karthikeyan Shanmugam, and Ali Tajer. Score-based causal representation learning: Linear and general transformations. *arXiv preprint arXiv:2402.00849*, 2024.

Christopher M Vigorito and Andrew G Barto. Incremental structure learning in factored MDPs with continuous states and actions. *University of Massachusetts Amherst-Department of Computer Science, Tech. Rep*, 2009.

Christopher M. Vigorito and Andrew G. Barto. Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Transactions on Autonomous Mental Development*, 2(2):132–143, 2010. doi: 10.1109/TAMD.2010.2050205.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. In *Advances in Neural Information Processing Systems*, 2021.

Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. EfficientZero V2: mastering discrete and continuous control with limited data. In *Proceedings of the 41st International Conference on Machine Learning*, pages 51041–51062, 2024a.

Ye Wang, Sipeng Zheng, Hao Luo, Wanpeng Zhang, Haoqi Yuan, Chaoyi Xu, Haiweng

Xu, Yicheng Feng, Mingyang Yu, Zhiyu Kang, et al. Rethinking visual-language-action model scaling: Alignment, mixture, and regularization. *arXiv preprint arXiv:2602.09722*, 2026.

Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23151–23180. PMLR, 17–23 Jul 2022.

Zizhao Wang, Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. Elden: Exploration via local dependencies. *Advances in Neural Information Processing Systems*, 36:15456–15474, 2023.

Zizhao Wang, Jiaheng Hu, Caleb Chuck, Stephen Chen, Roberto Martín-Martín, Amy Zhang, Scott Niekum, and Peter Stone. SkiLD: Unsupervised skill discovery guided by factor interactions. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Franz Wegner. Inverse participation ratio in 2+ ε dimensions. *Zeitschrift für Physik B Condensed Matter*, 36(3):209–214, 1980.

Ziping Xu and Ambuj Tewari. Near-optimal reinforcement learning in factored MDPs: Oracle-efficient algorithms for the non-episodic setting. *Advances in Neural Information Processing Systems*, 33, 2020.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng,

Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VYOe2eBQeh.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering Atari games with limited data. *Advances in neural information processing systems*, 34: 25476–25488, 2021.

Håkan LS Younes and Michael L Littman. PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects. *Techn. Rep. CMU-CS-04-162*, 2: 99, 2004.

Lihan Zha, Asher J. Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z. Ren, and Anirudha Majumdar. LAP: Language-action pre-training enables zero-shot cross-embodiment transfer, 2026. URL https://arxiv.org/abs/2602.10556.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2020.

Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International conference on machine learning*, pages 7444–7453. PMLR, 2019.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=0akLDTFR9x.

Appendix A

Nash-ACF: Proofs for the Sparsity Game Identifiability

A.1 Proofs

This appendix gives the full proof of Theorem 4.4.2 and its consequences. The argument is organized in five steps, each occupying one subsection. Notation follows Section 4.4.

A.1.1 Preliminaries

Let $\vartheta = (g_\star, p)$ denote the ground-truth model; let $\widehat{\vartheta} = (\widehat{g}, \widehat{p})$ be the learned counterpart satisfying $\widehat{\vartheta} \sim_{\text{obs}} \vartheta$. By observational equivalence there exists a diffeomorphism $v : \mathcal{Z} \to \mathcal{Z}$ with $\widehat{g} = g_\star \circ v$; we set $v := g_\star^{-1} \circ \widehat{g}$ and denote its Jacobian by $J_v \in \mathbb{R}^{K \times K}$.

Assumption A.1.1 (Unconstrained function class)**.** $\widehat{g}$ and $\widehat{p}$ range over unconstrained smooth function classes: any invertible smooth $v : \mathcal{Z} \to \mathcal{Z}$ is realizable as $g_\star^{-1} \circ \widehat{g}$ for some feasible $\widehat{g}$, and $\widehat{p}$ can be chosen with any parametric sparsity pattern consistent with observational equivalence.

Lemma A.1.2 (Diagonal alignment; Lachapelle Lemma 2)**.** *Under sufficient influence*

(Assumption 4.2.3) and $\widehat{\vartheta} \sim_{\text{obs}} \vartheta$, there exists a permutation $P \in \{0,1\}^{K\times K}$ such that

$$C := J_v \cdot P \tag{A.1}$$

has nonzero diagonal $C_{j,j} \neq 0$ for all $j \in [K]$ on a full-measure subset of $\mathcal{Z}$.

Lemma A.1.3 (Support inclusion; Lachapelle Eq. (33) / two-sided variant of Eq. (22))**.** *Under Assumption 4.2.3 and $\widehat{\vartheta} \sim_{\text{obs}} \vartheta$, for every $\ell \in [K + d_a]$ and every i with $G_{i,\ell} = 1$,*

$$(J_v)^\top e_i \in \mathbb{R}^K_{\widehat{G}_{\cdot,\ell}}. \tag{A.2}$$

Equivalently, $G_{i,\ell} = 1 \wedge (J_v)_{j,i} \neq 0 \Rightarrow \widehat{G}_{j,\ell} = 1$. For $\mathbf{G} = G^z$, the analogous inclusion is derived from $\widehat{H} = J_v(z_{t+1})^\top H\, J_v(z_t)$ (Lachapelle et al., 2024a); for $\mathbf{G} = (G^z)^\top$, from the transposed cross-Hessian.

Theorem A.1.4 (Identifiability; Lachapelle Theorem 4 and two-sided extension)**.** *Under Assumption 4.2.3 and $\widehat{\vartheta} \sim_{\text{obs}} \vartheta$ with $|\widehat{\mathbf{G}}|_0 \leq |\mathbf{G}|_0$, there exists a permutation P with $P^\top \widehat{\mathbf{G}} = \mathbf{G}$, for $\mathbf{G} \in \{G^a, G^z, (G^z)^\top, G\}$. The action and row-$G^z$ cases follow from the one-sided interventional argument of Lachapelle et al. (2022); the column-G^z case uses the two-sided structure of Lachapelle et al. (2024a).*

Our task is to show that Pareto optimality of the sparsity game forces the cardinality hypothesis of Theorem A.1.4, hence identifiability, without needing to assume it.

A.1.2 Step 1: Reduction to the Jacobian support pattern

Fix a feasible $\widehat{\vartheta}$ with associated v, J_v, and permutation P from Lemma A.1.2. Let $C = J_v \cdot P$ and define the P-aligned learned pattern $\tilde{\mathbf{G}} := P^\top \widehat{\mathbf{G}}$.

Proposition A.1.5 (Row-wise support inclusion)**.** *For every $j \in [K]$,*

$$\tilde{\mathbf{G}}_{j,\cdot} \supseteq \bigcup_{i\,:\,C_{j,i}\neq 0} \mathbf{G}_{i,\cdot} \tag{A.3}$$

Proof. Lemma A.1.3 asserts $\mathbf{G}_{i,\ell} = 1$ and $(J_v)_{j,i} \neq 0$ imply $\widehat{\mathbf{G}}_{j,\ell} = 1$. Substituting $J_v = CP^\top$ gives $(J_v)_{j,i} = C_{j,P(i)}$; relabeling $i' := P(i)$ yields $\mathbf{G}_{i',\ell} = 1 \wedge C_{j,i'} \neq 0 \Rightarrow (P^\top \widehat{\mathbf{G}})_{j,\ell} = 1$, which is (A.3) row-by-row. □

A.1.3 Step 2: Row-wise lower bound and achievability

Definition A.1.6 (Row floor)**.** For C with nonzero diagonal, define

$$\mathcal{M}_j(C) := \left| \bigcup_{i\,:\,C_{j,i}\neq 0} \mathbf{G}_{i,\cdot} \right|_0, \qquad \mathcal{M}(C) := \big(\mathcal{M}_1(C), \ldots, \mathcal{M}_K(C)\big). \tag{A.4}$$

Lemma A.1.7 (Lower bound)**.** *For any feasible $\widehat{\vartheta}$ with associated C, $\ell_j^{\mathbf{G}}(\widehat{\vartheta}) \geq \mathcal{M}_j(C)$ for every j.*

Proof. Applying $|\cdot|_0$ to (A.3) and using permutation-invariance of cardinality gives the claim. □

Lemma A.1.8 (Achievability)**.** *Under Assumption A.1.1, for any C realizable by some feasible $\widehat{\vartheta}$, there exists a feasible $\widehat{\vartheta}'$ with the same C such that $\ell_j^{\mathbf{G}}(\widehat{\vartheta}') = \mathcal{M}_j(C)$ for all j simultaneously.*

Proof. Given $\widehat{\vartheta}$ realizing C, set $\tilde{\mathbf{G}}'_{j,\cdot} := \bigcup_{i:C_{j,i}\neq 0} \mathbf{G}_{i,\cdot}$. Construct $\widehat{p}'$ with parametric support $\tilde{\mathbf{G}}'_{j,\cdot}$ by zeroing all slack parameters of $\widehat{p}$ (those declaring dependence on parents outside the forced support). Any functional dependence outside the forced support would contradict (A.3) upward, so these parameters carried zero functional weight and zeroing them preserves observational equivalence. Under Assumption A.1.1, $\widehat{p}'$ is in the feasible class. □

Remark A.1.9 (Support-level abstraction). Both $\mathcal{M}_j(C)$ and the classification in Appendix A.1.4 depend only on $\mathrm{supp}(C)$, not on real values. The argument that follows lives on the finite lattice of binary patterns with nonzero diagonal.

A.1.4 Step 3: Dichotomy on off-diagonals

Definition A.1.10 (Type classification)**.** Let C have nonzero diagonal. An off-diagonal entry $C_{j,i}$ with $i \neq j$ and $C_{j,i} \neq 0$ is $\mathbf{G}$-*preserving* if $\mathbf{G}_{i,\cdot} \subseteq \mathbf{G}_{j,\cdot}$, and *non-preserving* otherwise.

Definition A.1.11 ($\mathbf{G}$-preserving matrix)**.** C is $\mathbf{G}$-*preserving* if every off-diagonal nonzero is $\mathbf{G}$-preserving; equivalently, for all $i \neq j$, $C_{j,i} \neq 0 \Rightarrow \mathbf{G}_{i,\cdot} \subseteq \mathbf{G}_{j,\cdot}$.

Remark A.1.12 (Collapse under preservation)*.* C is $\mathbf{G}$-preserving iff for every $j \in [K]$, $\bigcup_{i:C_{j,i}\neq 0} \mathbf{G}_{i,\cdot} = \mathbf{G}_{j,\cdot}$.

Proof. ($\Rightarrow$) The diagonal term $\mathbf{G}_{j,\cdot}$ is always present; every off-diagonal contribution $\mathbf{G}_{i,\cdot}$ with $C_{j,i} \neq 0$ and $i \neq j$ satisfies $\mathbf{G}_{i,\cdot} \subseteq \mathbf{G}_{j,\cdot}$ by preservation, so the union equals $\mathbf{G}_{j,\cdot}$. ($\Leftarrow$) If the union collapses to $\mathbf{G}_{j,\cdot}$ for all j, then any off-diagonal $C_{j,i} \neq 0$ satisfies $\mathbf{G}_{i,\cdot} \subseteq \mathbf{G}_{j,\cdot}$. □

Proposition A.1.13 (Row-local domination)**.** *Let C have nonzero diagonal. Then:*

(i) If C is $\mathbf{G}$-preserving, $\mathcal{M}_j(C) = |\mathbf{G}_{j,\cdot}|_0$ for every j.

(ii) If C is not $\mathbf{G}$-preserving, there exists $j^\star \in [K]$ and a matrix C' obtained from C by zeroing all non-preserving entries in row $j^\star$, such that

$$\mathcal{M}_{j^\star}(C') < \mathcal{M}_{j^\star}(C), \tag{A.5}$$

$$\mathcal{M}_k(C') = \mathcal{M}_k(C) \qquad \forall k \neq j^\star. \tag{A.6}$$

C' has nonzero diagonal, and its row-$j^\star$ off-diagonals are exactly the $\mathbf{G}$-preserving off-diagonals of C.

Proof. **(i)** Immediate from Remark A.1.12 and Definition A.1.6.

(ii) Since C is not $\mathbf{G}$-preserving, some row $j^\star$ contains a non-preserving off-diagonal.

Define

$$C'_{j,i} := \begin{cases} 0 & \text{if } j = j^\star,\ i \neq j^\star,\ \mathbf{G}_{i,\cdot} \not\subseteq \mathbf{G}_{j^\star,\cdot}, \\ C_{j,i} & \text{otherwise.} \end{cases}$$

Diagonal preserved: $C'_{j^\star,j^\star} = C_{j^\star,j^\star} \neq 0$, and $C'_{k,k} = C_{k,k} \neq 0$ for $k \neq j^\star$.

Other rows unchanged: For $k \neq j^\star$, row k of C' coincides with row k of C, hence $\mathcal{M}_k(C') = \mathcal{M}_k(C)$.

Row $j^\star$ floor strictly decreases: The active index set in C' is $\{j^\star\} \cup \{i \neq j^\star : C_{j^\star,i} \neq 0,\ \mathbf{G}_{i,\cdot} \subseteq \mathbf{G}_{j^\star,\cdot}\}$. All remaining $\mathbf{G}_{i,\cdot}$'s are subsets of $\mathbf{G}_{j^\star,\cdot}$, so $\mathcal{M}_{j^\star}(C') = |\mathbf{G}_{j^\star,\cdot}|_0$. For $\mathcal{M}_{j^\star}(C)$: by hypothesis there is a non-preserving entry $(j^\star, i_0)$, giving $\ell_0 \in \mathbf{G}_{i_0,\cdot} \setminus \mathbf{G}_{j^\star,\cdot}$; this ℓ_0 lies in the C-union but not in $\mathbf{G}_{j^\star,\cdot}$, so $\mathcal{M}_{j^\star}(C) \geq |\mathbf{G}_{j^\star,\cdot}|_0 + 1 > \mathcal{M}_{j^\star}(C')$. □

Corollary A.1.14 (Pareto-minimal floors are **G**-preserving)**.** *Every Pareto-minimal element of $\{\mathcal{M}(C) : C \text{ realizable}\}$ is attained by some **G**-preserving C, and all **G**-preserving C's attain the common floor vector $\big(|\mathbf{G}_{1,\cdot}|_0, \ldots, |\mathbf{G}_{K,\cdot}|_0\big)$.*

A.1.5 Step 4: Pareto optimality forces G-preservation

Theorem A.1.15 (Pareto optima are **G**-preserving)**.** *Under Assumption A.1.1, every Pareto-optimal $\widehat{\vartheta}^\star$ of the **G**-restricted sparsity game has **G**-preserving $C^\star = J_v^\star \cdot P$, and*

$$\boldsymbol{\ell}(\widehat{\vartheta}^\star) \;=\; \big(|\mathbf{G}_{1,\cdot}|_0, \ldots, |\mathbf{G}_{K,\cdot}|_0\big) \quad \textit{up to permutation by } P. \tag{A.7}$$

Proof. Suppose for contradiction that $\widehat{\vartheta}^\star$ is Pareto-optimal but $C^\star$ is not **G**-preserving.

Apply Proposition A.1.13(ii). There exist $j^\star$ and a row-local modification $C^\dagger$ with $\mathcal{M}_{j^\star}(C^\dagger) < \mathcal{M}_{j^\star}(C^\star)$ and $\mathcal{M}_k(C^\dagger) = \mathcal{M}_k(C^\star)$ for $k \neq j^\star$.

Realize $C^\dagger$ by a feasible $\widehat{\vartheta}^\dagger$. Under Assumption A.1.1, any invertible Jacobian structure is realizable by some diffeomorphism $v^\dagger$. Since $C^\dagger$ differs from $C^\star$ only in row $j^\star$, preserves

the diagonal, and only zeros off-diagonals, invertibility holds on a dense set (and remains so under arbitrarily small smooth perturbation). Define $\widehat{g}^\dagger := g_\star \circ v^\dagger$ and $\widehat{p}^\dagger := v^\dagger_\# p$; observational equivalence holds by construction.

Apply Lemma A.1.8. Choose $\widehat{p}^\dagger$ with minimal parametric support, giving $\boldsymbol{\ell}(\widehat{\vartheta}^\dagger) = \mathcal{M}(C^\dagger)$.

Pareto domination.

$$\begin{aligned}\ell_{j^\star}(\widehat{\vartheta}^\dagger) = \mathcal{M}_{j^\star}(C^\dagger) < \mathcal{M}_{j^\star}(C^\star) = \ell_{j^\star}(\widehat{\vartheta}^\star),\\ \ell_k(\widehat{\vartheta}^\dagger) = \mathcal{M}_k(C^\dagger) = \mathcal{M}_k(C^\star) = \ell_k(\widehat{\vartheta}^\star) \qquad \forall k \neq j^\star.\end{aligned}$$

This contradicts Pareto optimality of $\widehat{\vartheta}^\star$, so $C^\star$ is $\mathbf{G}$-preserving. By Proposition A.1.13(i) and Lemma A.1.8, $\boldsymbol{\ell}(\widehat{\vartheta}^\star) = \big(|\mathbf{G}_{1,\cdot}|_0, \ldots, |\mathbf{G}_{K,\cdot}|_0\big)$. □

A.1.6 Step 5: G-preservation implies identifiability

Proposition A.1.16 (Cardinality pinch)**.** *If $\widehat{\vartheta}^\star$ is Pareto-optimal with $\mathbf{G}$-preserving $C^\star$ and $\tilde{\mathbf{G}} := P^\top \widehat{\mathbf{G}}^\star$, then $\tilde{\mathbf{G}}_{j,\cdot} = \mathbf{G}_{j,\cdot}$ for every $j \in [K]$.*

Proof. Lower bound. By Proposition A.1.5, $\tilde{\mathbf{G}}_{j,\cdot} \supseteq \bigcup_{i:C^\star_{j,i}\neq 0} \mathbf{G}_{i,\cdot} \supseteq \mathbf{G}_{j,\cdot}$, the last inclusion using $C^\star_{j,j} \neq 0$.

Cardinality match. By Theorem A.1.15, $|\tilde{\mathbf{G}}_{j,\cdot}|_0 = \ell_j(\widehat{\vartheta}^\star) = \mathcal{M}_j(C^\star) = |\mathbf{G}_{j,\cdot}|_0$, the last equality by Remark A.1.12.

Conclusion. Set inclusion with equal cardinalities forces $\tilde{\mathbf{G}}_{j,\cdot} = \mathbf{G}_{j,\cdot}$. □

Corollary A.1.17 (Identifiability)**.** *Every Pareto-optimal $\widehat{\vartheta}^\star$ of the $\mathbf{G}$-restricted sparsity game satisfies $P^\top \widehat{\mathbf{G}}^\star = \mathbf{G}$ for some permutation P.*

Proof. Proposition A.1.16 gives $P^\top \widehat{\mathbf{G}}^\star = \mathbf{G}$ row-by-row. □

This establishes Theorem 4.4.2(i). Parts (ii) and (iii) are Theorem A.1.15 and Corollary A.1.14.

Appendix B

CAMP: Proofs

Theorem B.0.1. *Let the tuple $(M, \bar{M}, G)$ be a grounded abstract model and a function $\phi : \mathcal{S} \to \mathcal{Z} \subseteq \mathbb{R}^{d_z}$. The model satisfies that $B_t(\cdot \mid o_0, ..., o_{t-1}) = \bar{B}_t(\cdot \mid o_0, ..., o_{t-1})$ if and only if ϕ is dynamics preserving.*

Proof. Let $\phi^{-1}(z) = \{s \in \mathcal{S} \mid \phi(s) = z\}$. We construct $\bar{T}$ and G such that it satisfies that,

$$\bar{T}(z'|z, o) = \int_{s' \in \phi^{-1}(z')} T(s'|z, o) ds';$$
$$G(s'|z, o, z') = \frac{T(s'|z, o)\mathbb{1}[\phi(s') = z']}{\bar{T}(z'|z, o)}$$

If the dynamics preserving property holds, we have that there exists a mapping ϕ such that $T(s'|s, o) = T(s'|\phi(s), o)$. Hence, by defining that abstract state as $\bar{s} = (z, o, z')$, we can build the grounded abstract model such that it follows that $B_t = \bar{B}_t$, by construction.

To prove the converse, we assume that $B_t = \bar{B}_t$.

Hence, by construction, we have that

$$\mathbb{P}(s_t, ..., s_0|o_0, z_0, ..., o_{t-1}, z_{t-1}) = \prod_t \mathbb{P}(s_t|o_0, z_0, ..., o_{t-1}, z_{t-1}).$$

Therefore, we have that

$$\begin{aligned}
\bar{B}_t(s_t, ..., s_0|o_0, ..., o_{t-1}) &= \int \prod_{i=0}^{t} \mathbb{P}(s_i|o_0, z_0, ..., o_{i-1}, z_{t-1})\mathbb{P}(z_i, ..., z_0|o_0, ..., o_{i-1})dz_0...z_t \\
&= \int \prod_{i=0}^{t} \mathbb{P}(s_i|z_i, o_{i-1})\mathbb{P}(z_i, ..., z_0|o_0, ..., o_{i-1})dz_0...z_t \\
&= \int \prod_{i=0}^{t} G(s_i|z_{i-1}, o_{i-1}, z_i)\mathbb{P}(z_i, ..., z_0|o_0, ..., o_{i-1})dz_0...z_t \\
&= \prod_{i=0}^{t} \int G(s_i|z_{i-1}, o_{i-1}, z_i)\mathbb{P}(z_i, z_{i-1}|o_0, ..., o_{i-1})dz_i z_{i-1} \\
&= \prod_{i=0}^{t} \int G(s_i|z_{i-1}, o_{i-1}, z_i)\bar{T}(z_i|z_{i-1}, o_{i-1})\mathbb{P}(z_{i-1}|o_0, ..., o_{i-2})dz_i z_{i-1} \\
&= \prod_{i=0}^{t} \int \tilde{T}(s_i|z_{i-1}, o_{i-1})\mathbb{P}(z_{i-1}|o_0, ..., o_{i-2})dz_{i-1}
\end{aligned}$$

$$\begin{aligned}
B_t(s_t, ..., s_0|o_0, ..., o_{t-1}) &= p_0(s_0) \prod_{i=1}^{t} T(s_i|s_{i-1}, o_{i-1}) \\
&= \prod_{i=1}^{t} T(s_i|s_{i-1}, o_{i-1})\mathbb{P}(s_{i-1}|o_0, ..., o_{t-2})
\end{aligned}$$

Hence, we must have that for all $s_{i-1} \in z_{i-1}$ and all $i \in [t]$ and $t \geq 0$

$$\int T(s_i|s_{i-1}, o_{i-1})\mathbb{P}(s_{i-1}|o_0, ..., o_{t-2})ds_{i-1} = \int \tilde{T}(s_i|z_{i-1}, o_{i-1})\mathbb{P}(z_{i-1}|o_0, ..., o_{i-2})dz_{i-1}$$

That is,

$$\begin{cases} \mathbb{P}(s_0) = p_0(s_0) = \int G(s|z_0)p_0(z_0)ds & \text{for } t = 0 \\ \mathbb{P}(s_1|o_0) = \int T(s_1|s_0, o_0)p_0(s_0)ds_0 = \int \tilde{T}(s_0|z_0, o_0)p_0(z_0)dz_0 & \text{for } t = 1 \end{cases}$$

By definition, $t = 0$ holds. For $t = 1$, we have

$$\begin{aligned} \mathbb{P}(s_1|o_0) &= \int T(s_1|s_0, o_0)p_0(s_0)ds_0 \\ &= \int T(s_1|s_0, o_0)G(s_0|z_0)p_0(z_0)dz_0ds_0 \\ &= \int \tilde{T}(s_1|z_0, o_0)p_0(z_0)dz_0 \end{aligned}$$

which follows from the equation at $t = 0$. Hence, it must be true that for any $s_0 \in \phi^{-1}(z_0)$, for any z_0 with $p_0(z_0) > 0$.

$$\tilde{T}(s_1|z_0, o_0) = \int T(s_1|s_0, o_0)G(s_0|z_0)ds_0$$

We can see that for any $s_0 \in \phi^{-1}(z_0)$ such that $T(s_1|s_0, o_0) \neq \tilde{T}(s_1|z_0, o_0)$, the abstract model would commit a non-zero error in its prediction. Hence, it must be that $T(s_1|s_0, o_0) = \tilde{T}(s_1|z_0, o_0)$ for $s_0 \in \phi^{-1}(z_0)$.

Let the equations at time $t = i - 1$ and $t = i - 2$ hold, then

$$\begin{aligned}
&\mathbb{P}(s_i|o_0, ..., o_{i-1}) = \int T(s_i|s_{i-1}, o_{i-1}) p_{i-1}(s_{i-1}|o_0, ...o_{i-2}) ds_{i-1} \\
&= \int T(s_i|s_{i-1}, o_{i-1}) \tilde{T}(s_{i-1}|z_{i-2}, o_{i-2}) p_{i-2}(z_{i-2}|o_0, ..., o_{i-3}) ds_{i-1} dz_{i-1} dz_{i-2} \\
&= \int T(s_i|s_{i-1}, o_{i-1}) G(s_{i-1}|z_{i-2}, o_{i-2}, z_{i-1}) \bar{T}(z_{i-1}|z_{i-2}, o_{i-2}) p_{i-2}(z_{i-2}|o_0, ..., o_{i-3}) ds_{i-1} dz_{i-1} dz_{i-2} \\
&= \int \tilde{T}(s_i|z_{i-1}, o_{i-1}) p_{i-1}(z_{i-1}|o_0, ..., o_{i-2}) dz_{i-1}
\end{aligned}$$

Because $p_{i-1}(z_{i-1}|o_0, ..., o_{i-2}) = \int \bar{T}(z_{i-1}|z_{i-2}, o_{i-2}) p_{i-2}(z_{i-2}|o_0, ..., o_{i-3}) dz_{i-2}$ hold by construction of the abstract MDP, we need the following to hold.

$$\tilde{T}(s_i|z_{i-1}, o_{i-1}) = \int T(s_i|s_{i-1}, o_{i-1}) G(s_{i-1}|z_{i-2}, o_{i-2}, z_{i-1}) ds_{i-1}. \tag{B.1}$$

Therefore, as in the base case, we need that $\tilde{T}(s_i|z_{i-1}, o_{i-1}) = T(s_i|s_{i-1}, o_{i-1})$ for all $s_{i-1} \in \phi^{-1}(z_{i-1})$ that have $G(s_{i-1}|z_{i-2}, o_{i-2}, z_{i-1}) > 0$. Then, ϕ must be dynamics preserving.

□

Corollary B.0.2. *Let the tuple $(M, \bar{M}, G)$ be a grounded abstract model. Let the strong subgoal property (Konidaris et al., 2018) for an option o be defined as, $Pr(s'|s, o) = Pr(s'|o)$. The dynamics preserving property holds with a finite abstract state space $\mathcal{Z} = [N]$ for some $N \in \mathbb{N}$ if and only if the strong subgoal property holds.*

Proof. If the strong subgoal property holds, we have that $Pr(s'|s, o) = Pr(s'|o)$. Then, for any function $\phi : S \to \mathcal{Z}$, it holds that $\mathbb{P}(s'|\phi(s), o) = \mathbb{P}(s'|s, o)$.

Therefore, it is only important to be able to know if a given option is executable in a given abstract state. Therefore, we can construct the function $I_{\mathcal{O}}(s) = [I_0(s), ..., I_{|\mathcal{O}|}(s)]$ that returns a binary vector that indicates which options are executable in s.

Define the equivalence relation $s_0 \sim_{\mathcal{O}} s_1$ iff $I_{\mathcal{O}}(s_1) = I_{\mathcal{O}}(s_2)$. We can define the abstract state space as $Z \triangleq S/\sim_{\mathcal{O}}$, that is, the set of equivalent classes. Given that there at most $2^{|\mathcal{O}|} \in \mathbb{N}$ classes, then the abstract MDP is finite.

We assume that the dynamics preserving property holds and that the abstract state space Z is finite to prove the converse. Then, there exists $\phi : S \to \mathcal{Z}$ such that $\mathbb{P}(s'|\phi(s), o) = \mathbb{P}(s'|s, o)$ and $\mathbb{P}(I_o = 1|s) = \mathbb{P}(I_o = 1|\phi(s))$.

We can construct a factored $\phi(s) = [\phi_D(s), \phi_I(s)]$, such that, $\mathbb{P}(s'|\phi(s), o) = \mathbb{P}(s'|\phi_D(s), o)$ and $\mathbb{P}(I_o = 1|\phi(s)) = \mathbb{P}(I_o = 1|\phi_I(s))$.

If we define ϕ_I based on the function $I_{\mathcal{O}}$, as before, then ϕ_I maps to a set of at most $2^{|\mathcal{O}|}$ elements. As $\mathcal{Z} = \mathcal{Z}_D \times \mathcal{Z}_I$ is finite, then $\mathcal{Z}_D$ is also finite. Thus, we construct $Z_D = [M]$ and for each option o and equivalence class $m \in [M]$ options from each option o such that $Pr(s'|o_m) \triangleq Pr(s'|m, o)$. Then, the strong subgoal property holds for every o_m.

□

Proposition B.0.3. *Let ϕ be a dynamics-preserving abstraction and $\bar{s} = (\hat{z}, \hat{o}, z)$. For $\epsilon > 0$, if $\|G_z(s) - G_{\bar{s}}(s)\|_1^2 \leq \epsilon$, then there exists $\epsilon_T > 0$ and $\epsilon_R > 0$ such that $\|T(s'|s, o) - \tilde{T}(s'|z, o)\|_1^2 \leq \epsilon_T$ and $\|R(s, o) - \tilde{R}(z, o)\|_1^2 \leq \epsilon_R$.*

Proof. First, we prove that the bounded grounding error implies bounded transition distribution error. If ϕ is a dynamics abstraction, then we can learn $\tilde{T}(z'|z, o)$ and we have that $T(s'|s, o) = T(s'|z, o) = \int G_{\bar{s}}(s)\bar{T}(z'|z, o)dz'$ and its corresponding approximation $\tilde{T}(s'|z, o) = \int G_{z'}(s)\bar{T}(z'|z, o)dz'$

$$\begin{aligned}\|T(s'|s,o) - \tilde{T}(s'|z,o)\|_1 &= \left| \int \left(G'_{\bar{s}}(s)\bar{T}(z'|z,o) - G_{z'}(s)\bar{T}(z'|z,o) \right) dz' \right| \\ &\leq \int \bar{T}(z'|z,o)|G_{\bar{s}'}(s) - G_{z'}(s)|dz'ds \\ &\leq \sqrt{\epsilon}\end{aligned}$$

Analogously, we can bound the error of the reward function.

$$\begin{aligned}\|\bar{R}(z',o) - \tilde{R}(z',o)\|_1 &= \left| \int G_{\bar{s}'}(s)R(s,o)ds - \int G_{z'}(s)R(s,o)ds \right| \\ &\leq \int |G_{\bar{s}'}(s) - G_{z'}(s)|\,|R(s,o)|\,ds \\ &\leq RMax \int |G_{\bar{s}'}(s) - G_{z'}(s)|\,ds \\ &\leq RMax\sqrt{\epsilon}\end{aligned}$$

Then, it follows from Minkowski's inequality that

$$\begin{aligned}\|R(s,o) - \bar{R}(z',o)\|_1 &= \|R(s,o) - \tilde{R}(z',o) + \tilde{R}(z',o) - \bar{R}(z',o)\|_1 \\ &\leq \|R(s,o) - \tilde{R}(z',o)\|_1 + \|\tilde{R}(z',o) - \bar{R}(z',o)\|_1 \\ &\leq \sqrt{\epsilon} + RMax\sqrt{\epsilon} = \sqrt{\epsilon_R}\end{aligned}$$

□

Theorem B.0.4 (Value Loss Bound)**.** *Let $(M, \bar{M}, G)$ be a grounded abstract model and $\tilde{T}(s'|\bar{s},o) = \int G_{\bar{s}'}(s')\bar{T}(\bar{s}'|\bar{s},o)d\bar{s}'$ be the approximate transition dynamics from the grounded model. If the following conditions hold for all $o \in \mathcal{O}$ and all $s \in \mathcal{S}$ with $G_{\bar{s}}(s) > 0$:*

(1) $\|T(s'|s,o) - \tilde{T}(s'|\bar{s},o)\|_1^2 \leq \epsilon_T$*, and (2)*$|R(s,o) - \bar{R}(\bar{s},o)|^2 \leq \epsilon_R$*; then, for any policy* π*,*

$$|Q^\pi(s,o) - Q^\pi(\bar{s},o)| \leq \frac{\sqrt{\epsilon_R} + \gamma V Max \sqrt{\epsilon_T}}{1-\gamma}.$$

Proof. We proceed by induction on $Q_n^\pi(\bar{s},o)$, where

$$v_0^\pi(\bar{s}) = \mathbb{E}_{s\sim\bar{s}}\left[v^\pi(s)\right], \tag{B.2}$$

$$Q_1^\pi(\bar{s},o) = \int_{s\in\mathcal{S}} P(s)\left(R(s,o) + \gamma^\tau v_0^\pi(\bar{s}')\right) ds, \tag{B.3}$$

$$= \int_{s\in\mathcal{S}} P(s)\left(R(s,o) + \gamma^\tau \int_{s'\in\mathcal{S}} T^{s,o,s'} v^\pi(s') ds'\right) ds, \tag{B.4}$$

$$Q_i^\pi(\bar{s},o) = \int_{s\in\mathcal{S}} P(s)\left(R(s,o) + \gamma^\tau v_{i-1}^\pi(\bar{s}')\right) ds, \tag{B.5}$$

with $\bar{s}' = T(\cdot \mid s,o)$. I use $P(s)$ as shorthand for $P(s\sim\bar{s})$ and $T^{s,o,s'}$ for $T(s' \mid s,o)$, and let

$$\varepsilon_{Q,n} = \sum_{i=0}^{n} \sqrt{\varepsilon_R} + \gamma^i \left(\text{VMax}\sqrt{\varepsilon_T}\right). \tag{B.6}$$

Base Case: $Q^\pi \approx Q_1^\pi$.

$$Q^\pi(s,o) - Q_1^\pi(\bar{s},o) \tag{B.7}$$

$$= R(s,o) + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \int_s P(s)\left(R(\bar{s},o) - \gamma^\tau v_0^\pi(\bar{s}') ds\right), \tag{B.8}$$

$$= \underbrace{R(s,o) - R(\bar{s},o)}_{\leq\sqrt{\varepsilon_R}} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \int_s P(s)\gamma^\tau v_0^\pi(\bar{s}') ds, \tag{B.9}$$

$$\leq \sqrt{\varepsilon_R} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_s P(s)\mathbb{E}_{s'\sim\bar{s}'}[v^\pi(s')] ds \tag{B.10}$$

$$\leq \sqrt{\varepsilon_R} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_s P(s) \int_{s'} P(s'\sim\bar{s}') v^\pi(s') ds'\ ds, \tag{B.11}$$

$$\leq \sqrt{\varepsilon_R} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_s P(s) \int_{s'} T^{s,o,s'} v^\pi(s') ds'\ ds, \tag{B.12}$$

$$\leq \sqrt{\varepsilon_R} + \gamma^\tau \text{VMax} \underbrace{\int_{s'} T^{s,o,s'} - \int_s P(s) T^{s,o,s'} ds\ ds'}_{\leq\sqrt{\varepsilon_T}}, \tag{B.13}$$

$$\leq \sqrt{\varepsilon_R} + \gamma^\tau \text{VMax}\sqrt{\varepsilon_T}. \tag{B.14}$$

This concludes the base case. □

Inductive Case: $Q^\pi \approx Q^\pi_n \implies Q^\pi \approx Q^\pi_{n+1}$. We assume that, for every $s \in \mathcal{S}$ and any o,

$$Q^\pi(s,o) - Q^\pi_n(\bar{s},o) \leq \varepsilon_{Q,n}, \tag{B.15}$$

and prove that

$$Q^\pi(s,o) - Q^*_{n+1}(\bar{s},o) \leq \varepsilon_{Q,n+1}. \tag{B.16}$$

By algebra,

$$\begin{aligned}
&Q^\pi(s,o) - Q^\pi_{n+1}(\bar{s},o) && \text{(B.17)}\\
&= R(s,o) + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \int_s P(s) \left(R(s,o) + \gamma^\tau v^\pi_n(\bar{s}')\right) ds, && \text{(B.18)}\\
&= \underbrace{R(s,o) - R(\bar{s},o)}_{\leq \sqrt{\varepsilon_R}} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_s P(s) v^\pi_n(\bar{s}') ds, && \text{(B.19)}\\
&\leq \sqrt{\varepsilon_R} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_s P(s) v^\pi_n(\bar{s}') ds, && \text{(B.20)}\\
&= \sqrt{\varepsilon_R} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_s P(s) \underbrace{v^\pi_n(\bar{s}')}_{\geq \mathbb{E}_{s' \sim \bar{s}'}[v^\pi(s')] - \varepsilon_{Q,n}} ds, && \text{(B.21)}
\end{aligned}$$

$$\leq \sqrt{\varepsilon_R} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_s P(s) \left(\mathbb{E}_{s' \sim \bar{s}'}[v^\pi(s')] - \varepsilon_{Q,n}\right) ds, \tag{B.22}$$

$$= \sqrt{\varepsilon_R} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_s P(s) \int_{s'} T^{s,o,s'} v^\pi(s') ds' \; ds + \gamma^\tau \varepsilon_{Q,n}, \tag{B.23}$$

$$= \sqrt{\varepsilon_R} + \gamma^\tau \int_{s'} T^{s,o,s'} v^\pi(s') ds' - \gamma^\tau \int_{s'} \underbrace{\int_s P(s) T^{s,o,s'}}_{=T^{\bar{s},o,s'}} v^\pi(s') ds \; ds' + \gamma^\tau \varepsilon_{Q,n}, \tag{B.24}$$

$$\leq \sqrt{\varepsilon_R} + \gamma^\tau \text{VMAX} \underbrace{\int_{s'} T^{s,o,s'} - T^{\bar{s},o,s'} ds'}_{\leq \sqrt{\varepsilon_T}} + \gamma^\tau \varepsilon_{Q,n}, \tag{B.25}$$

$$\leq \sqrt{\varepsilon_R} + \gamma^\tau \text{VMAX} \sqrt{\varepsilon_T} + \gamma^\tau \varepsilon_{Q,n}, \tag{B.26}$$

$$\leq \sqrt{\varepsilon_R} + \gamma \text{VMAX} \sqrt{\varepsilon_T} + \gamma \varepsilon_{Q,n}, \tag{B.27}$$

$$= \varepsilon_{Q,n+1}. \tag{B.28}$$

This concludes the inductive case. □

Thus, by induction and the convergence of the geometric series, for any s, o, π, we conclude that

$$Q^\pi(s,o) - Q^\pi(\bar{s},o) \leq \frac{\sqrt{\varepsilon_R} + \gamma \text{VMAX} \sqrt{\varepsilon_T}}{1-\gamma}. \tag{B.29}$$

□

B.0.1 TPC is Dynamics Preserving

We start by considering that by learning an abstract state space such that $MI(S'; Z, O)$ is maximized. The following decomposition based on the mutual information chain rule corresponds to the TPC algorithm (Nguyen et al., 2021). In the original paper, they work at the primitive action level and all actions available always, hence, there's no need to consider initiation sets.

$$
\begin{aligned}
MI(S', Z'; Z, O) &\overset{(a)}{=} MI(S'; Z, O) + \underbrace{MI(Z'; Z, O|S')}_{=0}; \\
&\overset{(b)}{=} MI(Z'; Z, O) + \underbrace{MI(S'; Z, O|Z')}_{(1)}; \\
&\overset{(c)}{=} MI(Z'; Z, O) + MI(S'; Z, A) - MI(S'; Z') + MI(S'; Z'|Z, O);
\end{aligned}
$$

where (a) follows from the fact that give s' we can determine z', (b) follows from decomposing the term on the left-hand size and (c) from decomposing term (1).

The above implies that $MI(Z'; Z, O) = MI(S'; Z') - MI(S'; Z'|Z, O)$. Therefore, if we maximize both sides of this identity, we must have a latent space that preserve *only* the information of the state s' that is predictable from the previous (z, a) pair. $MI(Z'; Z, O)$ ensures that the next abstract state is predictable from the (z, o) tuple. $MI(S; Z)$ ensures that the abstract state has information about the ground state which is measured by $g(s|z)$.

$$
MI(S; O) = \int p(s, z) \log \frac{g(s|z)}{p(s)} ds dz \tag{B.30}
$$

The following decomposition shows the two extra terms required by the TPC algorithm to estabilize the optimization. Term (a) is the (differential) entropy of ϕ which tends to infinity for a deterministic function. This is solved by smoothing it with Gaussian noise of 0 mean and fixed standard deviation, as done in TPC. The second term (b) corresponds to the consistency term, that is, the transition function $p(z'|z, a)$ must have low entropy, which ensures that the abstract dynamics are learnt.

$$
\begin{aligned}
M(S';Z'|Z,O) &= \int p(s',z',z,o)\log\frac{p(s',z'|z,o)}{p(s'|z,o)p(z'|z,o)}ds'dz'dzdo \\
&= \int p(s',z',z,o)\log\frac{p(z'|s')}{p(z'|z,o)} \\
&= \underbrace{\int p(s',z')\log p(z'|s')ds'dz'}_{(a)} - \underbrace{\int p(z',z,o)\log p(z'|z,o)dz'dzdo}_{(b)}
\end{aligned}
$$

By maximizing $MI(Z';Z,O)$ and $MI(S';Z')$ using InfoNCE (Oord et al., 2018a), we obtain the TPC algorithm.

Appendix C

NMS-ACF: Proofs for Phase-Conditioned Identifiability

This appendix provides the full proofs of Lemma 6.6.3 (merge of two preservation families) and Proposition 6.6.4 (iterated merge of n preservation families). The merge lemma reproduces the four-step argument of Lachapelle et al. (2024a, Proposition 6) with abstract preservation families $\mathcal{F}_1, \mathcal{F}_2$ in place of $\{G^a\}$ and $\{G^z, (G^z)^\top\}$; Proposition 6.6.4 iterates this lemma via induction.

For completeness we first restate the supporting lemmas and propositions from Lachapelle et al. (2024a) on which the proofs rely. All numbering is that of the cited reference; results are stated verbatim up to notational substitutions.

C.0.1 Supporting Results from Lachapelle et al. 2024

G-preserving maps. Given a binary matrix $G \in \{0,1\}^{K \times m}$ and a diffeomorphism $c : \mathcal{Z} \to \mathcal{Z}$ with dependency graph $C \in \{0,1\}^{K \times K}$ (Definition 2 of the cited reference), c is *G-preserving* if C viewed as a linear map on the space $\mathbb{R}_G^{K \times m}$ of G-supported matrices maps this space into itself. Formally:

Definition C.0.1 (G-preserving matrix; Lachapelle et al., 2024a, Definition 11)**.** A matrix $C \in \mathbb{R}^{K\times K}$ is *G-preserving* if $C^\top \mathbb{R}_G^{K\times m} \subseteq \mathbb{R}_G^{K\times m}$, where $\mathbb{R}_G^{K\times m} := \{M : G_{i,j} = 0 \Rightarrow M_{i,j} = 0\}$.

Definition C.0.2 (G-preserving diffeomorphism; Lachapelle et al., 2024a, Definition 12)**.** A diffeomorphism $c : \mathcal{Z} \to \mathcal{Z}$ with dependency graph C is *G-preserving* if C is G-preserving in the matrix sense (Definition C.0.1).

The characterization of G-preservation via the row-inclusion lattice of G is central to the proofs:

Proposition C.0.3 (Lachapelle et al., 2024a, Proposition 3)**.** *A function c with dependency graph C is G-preserving if and only if*

$$G_{i,\cdot} \not\subseteq G_{j,\cdot} \implies C_{i,j} = 0, \qquad \forall i, j \in [K].$$

Proof. ($\Rightarrow$): Assume $G_{i,\cdot} \not\subseteq G_{j,\cdot}$. Then there is some k with $G_{i,k} = 1$ but $G_{j,k} = 0$. The matrix $e_i e_k^\top \in \mathbb{R}_G^{K\times m}$, and so by G-preservation $C^\top(e_i e_k^\top) \in \mathbb{R}_G^{K\times m}$. The (j,k) entry of $C^\top e_i e_k^\top$ equals $C_{i,j}$, and since $G_{j,k} = 0$, this must be zero, giving $C_{i,j} = 0$.

($\Leftarrow$): Take $A \in \mathbb{R}_G^{K\times m}$ and (i,j) with $G_{i,j} = 0$. Then $(C^\top A)_{i,j} = \sum_k C_{k,i} A_{k,j}$. If $A_{k,j} \neq 0$ then $G_{k,j} = 1$, so $G_{k,\cdot} \not\subseteq G_{i,\cdot}$ (the latter has a zero at column j whereas the former does not), whence $C_{k,i} = 0$ by hypothesis. Each term vanishes and $(C^\top A)_{i,j} = 0$. □

Group structure of G-preserving maps. Both G-preserving matrices and G-preserving diffeomorphisms form groups under their respective operations:

Proposition C.0.4 (Lachapelle et al., 2024a, Proposition 4)**.** *Invertible G-preserving matrices form a group under matrix multiplication. Specifically, the identity matrix is G-preserving; the product of two G-preserving matrices is G-preserving; and the inverse of an invertible G-preserving matrix is G-preserving.*

Proof. Identity: $I^\top \mathbb{R}_G^{K\times m} = \mathbb{R}_G^{K\times m}$.

Closure under multiplication: If C_1, C_2 are G-preserving, then $(C_1C_2)^\top \mathbb{R}_G^{K\times m} = C_2^\top C_1^\top \mathbb{R}_G^{K\times m} \subseteq C_2^\top \mathbb{R}_G^{K\times m} \subseteq \mathbb{R}_G^{K\times m}$.

Closure under inversion: Let C be invertible and G-preserving. Viewing $C^\top$ as a linear map on $\mathbb{R}^{K\times m}$, invertibility of C makes $C^\top$ an invertible linear map, so $\dim(C^\top \mathbb{R}_G^{K\times m}) = \dim(\mathbb{R}_G^{K\times m})$. Combined with $C^\top \mathbb{R}_G^{K\times m} \subseteq \mathbb{R}_G^{K\times m}$, this gives equality: $C^\top \mathbb{R}_G^{K\times m} = \mathbb{R}_G^{K\times m}$. Applying $C^{-\top}$ to both sides, $\mathbb{R}_G^{K\times m} = (C^{-1})^\top \mathbb{R}_G^{K\times m}$, i.e., C^{-1} is G-preserving. □

Proposition C.0.5 (Lachapelle et al., 2024a, Proposition 5)**.** *The set of G-preserving diffeomorphisms forms a group under composition: it contains the identity diffeomorphism, is closed under composition, and is closed under inversion.*

Proof. We use the pointwise-Jacobian characterization: a differentiable c is G-preserving iff its Jacobian $Dc(z)$ is G-preserving at every $z \in \mathcal{Z}$ (Lemma 1 of Lachapelle et al. (2024a); follows from Proposition C.0.3 combined with Proposition 1 of the cited reference linking dependency graphs to Jacobians).

Identity: Its Jacobian is I, which is G-preserving.

Composition: For diffeomorphisms c, c' with dependency graphs C, C', the chain rule gives $D(c \circ c')(z) = Dc(c'(z)) \cdot Dc'(z)$. Each factor is G-preserving pointwise; the product is G-preserving by Proposition C.0.4. Hence $c \circ c'$ is G-preserving.

Inversion: The inverse function theorem gives $Dc^{-1}(z) = (Dc(c^{-1}(z)))^{-1}$. The matrix being inverted is G-preserving, so by Proposition C.0.4 its inverse is G-preserving pointwise. Hence c^{-1} is G-preserving. □

Structural lemmas on invertible matrices. The following lemma provides the bridge permutation P_0 used throughout the merge argument:

Lemma C.0.6 (Lachapelle et al., 2024a, Lemma 2)**.** *Let $L \in \mathbb{R}^{K\times K}$ be an invertible*

matrix. Then there exists a permutation σ on $[K]$ such that $L_{i,\sigma(i)} \neq 0$ for all $i \in [K]$. Equivalently, writing P for the permutation matrix associated to σ (so that $Pe_i = e_{\sigma(i)}$), we have $P^\top \subseteq L$, meaning the support of $P^\top$ is contained in the support of L. In particular, both LP and PL have nonzero diagonals.

Proof. Invertibility gives $\det(L) \neq 0$. The Leibniz formula expands the determinant as

$$\det(L) \;=\; \sum_{\sigma \in S_K} \operatorname{sign}(\sigma) \prod_{i=1}^{K} L_{i,\sigma(i)},$$

where S_K is the symmetric group on K elements. A nonzero sum requires at least one nonzero summand, so there is some σ with $\prod_i L_{i,\sigma(i)} \neq 0$, hence $L_{i,\sigma(i)} \neq 0$ for every i. □

The pointwise permutations produced by Lemma C.0.6 at each $z \in \mathcal{Z}$ can be replaced by a single globally constant permutation, exploiting continuity:

Lemma C.0.7 (Lachapelle et al., 2024a, Lemma 12, Appendix A.5)**.** *Let $J : \mathcal{Z} \to \mathrm{GL}(K)$ be a continuous map from the connected space $\mathcal{Z}$ into invertible matrices. Suppose that at each $z \in \mathcal{Z}$ there exists a permutation $P(z)$ such that $J(z) \cdot P(z)$ has nonzero diagonal. Then there exists a* single *permutation P, independent of z, such that $J(z) \cdot P$ has nonzero diagonal for almost every $z \in \mathcal{Z}$.*

Proof sketch. The argument uses connectedness of $\mathcal{Z}$ to show that a topological invariant — the equivalence class of the pointwise permutation — cannot vary continuously. Let G be the set of G-preserving matrices (for some fixed graph; in our application G is the dependency graph union considered in Step 3 of Lemma 6.6.3), and let

$$\mathcal{L} \;:=\; \Big(\bigcup_{\pi \in S_K} GP_\pi \Big) \setminus \{\text{singular matrices}\}.$$

Define an equivalence relation on S_K by $\pi \sim \pi'$ if $P_\pi P_{\pi'}^\top$ is G-preserving, and for each equivalence class $\Pi \in S_K/\sim$ let $V_\Pi := GP_\pi \setminus \{\text{singular}\}$ for any representative $\pi \in \Pi$

(well-defined since $\pi \sim \pi'$ implies $GP_\pi = GP_{\pi'}$).

Partition claim: The sets $\{V_\Pi\}_{\Pi \in S_K/\sim}$ form a partition of $\mathcal{L}$. Suppose for contradiction that some invertible $A \in GP_\pi \cap GP_{\pi'}$ with $\pi \not\sim \pi'$. By Lemma C.0.6 there is a permutation Q with $AQ^\top$ having nonzero diagonal, and without loss of generality $Q \not\sim \pi$, making $P_\pi Q^\top$ non-G-preserving. Then $AQ^\top \in GP_\pi Q^\top$ forces a zero on the diagonal of $AQ^\top$ (by closure of G-preservation under multiplication applied to the permutation factor), a contradiction.

Each V_Π is clopen in $\mathcal{L}$: Closedness follows from $V_\Pi = \mathcal{L} \cap GP_\pi$ (intersection of $\mathcal{L}$ with a closed subspace of $\mathbb{R}^{K\times K}$); openness from $V_\Pi = \mathcal{L} \setminus \bigcup_{\Pi' \neq \Pi} V_{\Pi'}$ (complement of a finite union of closed sets).

Conclusion: The image $J(\mathcal{Z}) \subseteq \mathcal{L}$ is connected (continuous image of a connected space) and the V_Π partition $\mathcal{L}$ into clopen sets. Hence $J(\mathcal{Z})$ lies entirely within a single $V_{\Pi^\star}$. Picking any representative $P^\star \in \Pi^\star$ gives a single permutation such that $J(z)P^\star$ is G-preserving — and hence has nonzero diagonal — for all z up to a set where continuity breaks down (a set of measure zero, as handled by the companion argument in Lemma 11 of the cited reference).

Full details, including the extension from "almost everywhere" to "everywhere" via closure under continuity, are provided in Lachapelle et al. (2024a, Appendix A.5). □

Consistency relations and their combination. Definitions 13, 14, and 15 of the cited reference introduce the one-phase consistency relations $\sim^a_{\text{con}}$, $\sim^z_{\text{con}}$, and $\sim^{a,z}_{\text{con}}$ respectively. Their precise form is:

Definition C.0.8 (Lachapelle et al., 2024a, Definitions 13–15)**.** Two models $\theta, \tilde{\theta}$ are:

- *a-consistent*, $\theta \sim^a_{\text{con}} \tilde{\theta}$, if there is a permutation P with $\tilde{G}^a = P^\top G^a$ and $v = c \circ P^\top$ where c is G^a-preserving.
- *z-consistent*, $\theta \sim^z_{\text{con}} \tilde{\theta}$, if there is a permutation P with $\tilde{G}^z = P^\top G^z P$ and $v = c \circ P^\top$

where c is both G^z-preserving and $(G^z)^\top$-preserving.

- (a, z)-*consistent*, $\theta \sim^{a,z}_{\text{con}} \tilde{\theta}$, if both hold with a *single common permutation* P and a *single common map* c that is simultaneously G^a-, G^z-, and $(G^z)^\top$-preserving.

The core structural fact from which our iterated merge is built is that (a, z)-consistency is equivalent to the conjunction of the two one-sided consistencies:

Proposition C.0.9 (Lachapelle et al., 2024a, Proposition 6)**.** *For any two models $\theta, \tilde{\theta}$ satisfying Assumptions 1–3 of the cited reference,*

$$\theta \sim^{a,z}_{\text{con}} \tilde{\theta} \iff \theta \sim^{a}_{\text{con}} \tilde{\theta} \ \wedge \ \theta \sim^{z}_{\text{con}} \tilde{\theta}.$$

The nontrivial direction is ($\Leftarrow$): given separate witnesses (c, P) for $\sim^a_{\text{con}}$ and $(\bar{c}, \bar{P})$ for $\sim^z_{\text{con}}$, a shared permutation $P^\star = \bar{P} P_0$ and shared map $c^\star = \bar{c} \circ P_0$ exist, with P_0 extracted via Lemma C.0.6 applied to the dependency graph of $\bar{c}$.

Per-phase identifiability. The one-phase identifiability theorems produce the hypotheses of our Corollary 6.6.7. They assume the sufficient-influence conditions of Lachapelle et al. (2024a, Assumptions 6, 7, and 8).

Theorem C.0.10 (Lachapelle et al., 2024a, Theorem 1)**.** *Let $\theta, \tilde{\theta}$ satisfy Assumptions 1–4 of the cited reference, Assumption 6 (sufficient influence of a, continuous), and $\theta \sim_{\text{obs}} \tilde{\theta}$. If $\|\tilde{G}^a\|_0 \leq \|G^a\|_0$, then $\theta \sim^a_{\text{con}} \tilde{\theta}$.*

Theorem C.0.11 (Lachapelle et al., 2024a, Theorem 2)**.** *Same as Theorem C.0.10, but with Assumption 7 (discrete a) in place of Assumption 6.*

Theorem C.0.12 (Lachapelle et al., 2024a, Theorem 3)**.** *Let $\theta, \tilde{\theta}$ satisfy Assumptions 1–4 of the cited reference, Assumption 8 (sufficient influence of z), and $\theta \sim_{\text{obs}} \tilde{\theta}$. If $\|\tilde{G}^z\|_0 \leq \|G^z\|_0$, then $\theta \sim^z_{\text{con}} \tilde{\theta}$.*

Graphical criterion for complete disentanglement. For completeness we restate Assumption 5 and Proposition 7, which together give the single-phase graphical criterion generalized by our multi-phase criterion (Definition 6.6.9).

Assumption C.0.13 (Lachapelle et al., 2024a, Assumption 5)**.** Let $G = [G^z \mid G^a]$. For all $i \in [K]$,

$$\Big(\bigcap_{j \in \mathrm{Ch}_i^z} \mathrm{Pa}_j^z\Big) \cap \Big(\bigcap_{j \in \mathrm{Pa}_i^z} \mathrm{Ch}_j^z\Big) \cap \Big(\bigcap_{\ell \in \mathrm{Pa}_i^a} \mathrm{Ch}_\ell^a\Big) \;=\; \{i\}.$$

Proposition C.0.14 (Lachapelle et al., 2024a, Proposition 7)**.** *Let $\theta, \tilde{\theta}$ be models satisfying Assumptions 1–3. If $\theta \sim_{\mathrm{con}}^{a,z} \tilde{\theta}$ and G satisfies Assumption C.0.13, then $\theta \sim_{\mathrm{perm}} \tilde{\theta}$, i.e., complete disentanglement holds.*

With these preliminaries in place we can now prove Lemma 6.6.3 and Proposition 6.6.4.

C.0.2 Proof of Lemma 6.6.3

Proof. Recall the setup: $v = c_1 \circ P_1^\top = c_2 \circ P_2^\top$ with c_r being $\mathcal{F}_r$-preserving for $r \in \{1, 2\}$ and P_1, P_2 permutations. Let C_1, C_2 denote the dependency graphs of c_1, c_2.

Step 1: Extract a bridge permutation P_0. Since c_1 is a diffeomorphism, its Jacobian is invertible pointwise. By Lachapelle et al. (2024a, Lemma 2), there exists a permutation $P_0(z)$ such that

$$P_0(z)^\top \;\subseteq\; C_1 \qquad \text{for almost every } z \in \mathcal{Z}. \tag{C.1}$$

By Lachapelle et al. (2024a, Lemma 12) (Appendix A.5), $P_0(z)$ can be chosen to be a globally constant permutation P_0 by continuity of the Jacobian.

Step 2: P_0 inherits $\mathcal{F}_1$-preservation. By Lachapelle et al. (2024a, Proposition 3), membership $P_0^\top \subseteq C_1$ and $\mathcal{F}_1$-preservation of C_1 imply that $P_0^\top$ is itself $\mathcal{F}_1$-preserving. Specifically, for each $G \in \mathcal{F}_1$ and any indices i, j with $G_{i,\cdot} \not\subseteq G_{j,\cdot}$, Proposition 3 gives $(C_1)_{i,j} = 0$, and since $P_0^\top \subseteq C_1$, also $(P_0^\top)_{i,j} = 0$. Hence $P_0^\top$ is G-preserving for every

$G \in \mathcal{F}_1$.

By Lachapelle et al. (2024a, Propositions 4 and 5) (closure of G-preserving diffeomorphisms under composition, for each $G \in \mathcal{F}_1$), the composition $c_1 \circ P_0$ is $\mathcal{F}_1$-preserving.

Step 3: $c_1 \circ P_0$ inherits $\mathcal{F}_2$-preservation. This is the cross-relation step. From the two decompositions of v:

$$c_1 \circ P_1^\top \;=\; c_2 \circ P_2^\top \qquad \Longrightarrow \qquad C_1 P_1^\top \;=\; C_2 P_2^\top \tag{C.2}$$

$$\Longrightarrow \qquad C_1 P_1^\top P_2 \;=\; C_2. \tag{C.3}$$

Combining (C.3) with (C.1):

$$P_0^\top P_1^\top P_2 \;\subseteq\; C_1 P_1^\top P_2 \;=\; C_2. \tag{C.4}$$

By Lachapelle et al. (2024a, Proposition 3) applied to $\mathcal{F}_2$-preserving C_2, the permutation $P_0^\top P_1^\top P_2$ is $\mathcal{F}_2$-preserving.

We now show $c_1 \circ P_0$ is $\mathcal{F}_2$-preserving. From (C.4), the permutation $P_0^\top P_1^\top P_2$ is $\mathcal{F}_2$-preserving. Its inverse $P_2^\top P_1 P_0$ is also $\mathcal{F}_2$-preserving by Lachapelle et al. (2024a, Proposition 4) (closure under inversion).

Now consider the product $C_2 \cdot (P_2^\top P_1 P_0)$:

$$C_2 \cdot (P_2^\top P_1 P_0) \;=\; (C_1 P_1^\top P_2)(P_2^\top P_1 P_0) \;=\; C_1 P_0, \tag{C.5}$$

where the first equality uses (C.3). The LHS is C_2 (which is $\mathcal{F}_2$-preserving) times a $\mathcal{F}_2$-preserving permutation, so the LHS is $\mathcal{F}_2$-preserving by Lachapelle et al. (2024a, Proposition 4) (closure under multiplication). Therefore $C_1 P_0$ is $\mathcal{F}_2$-preserving.

Since $C_1 P_0$ is the dependency graph of $c_1 \circ P_0$, we conclude that $c_1 \circ P_0$ is $\mathcal{F}_2$-preserving.

Step 4: Combine into a single decomposition. Set $c_{12} := c_1 \circ P_0$ and $P_{12} := P_1 P_0$. Then

$$c_{12} \circ P_{12}^\top \;=\; (c_1 \circ P_0) \circ (P_1 P_0)^\top \;=\; (c_1 \circ P_0) \circ (P_0^\top P_1^\top) \;=\; c_1 \circ P_1^\top \;=\; v, \tag{C.6}$$

using $P_0 P_0^\top = I$.

From Step 2, c_{12} is $\mathcal{F}_1$-preserving; from Step 3, it is $\mathcal{F}_2$-preserving. Hence c_{12} is $(\mathcal{F}_1 \cup \mathcal{F}_2)$-preserving.

Graph alignments. The final step is to verify that the merged permutation $P_{12} = P_1 P_0$ correctly aligns the graphs under the consistency condition, i.e., that for every $G \in \mathcal{F}_1 \cup \mathcal{F}_2$ of the appropriate type (action-type or latent-type), the pre-merge alignment condition lifts.

For $G^a \in \mathcal{F}_1 \cup \mathcal{F}_2$: suppose $\tilde{G}^a = P_r^\top G^a$ for $r \in \{1, 2\}$ (from the pre-merge consistency). We need $\tilde{G}^a = P_{12}^\top G^a = (P_1 P_0)^\top G^a$. For $r = 1$: $(P_1 P_0)^\top G^a = P_0^\top P_1^\top G^a = P_0^\top \tilde{G}^a$. By Step 2, $P_0^\top$ is $\mathcal{F}_1$-preserving, hence preserves the row structure of $\tilde{G}^a$: $P_0^\top \tilde{G}^a = \tilde{G}^a$. Similar argument for $r = 2$ using Step 3's conclusion that $P_0^\top P_1^\top P_2$ is $\mathcal{F}_2$-preserving.

For $G^z \in \mathcal{F}_1 \cup \mathcal{F}_2$ (latent-type, two-sided alignment): suppose $\tilde{G}^z = P_r^\top G^z P_r$. We need $\tilde{G}^z = P_{12}^\top G^z P_{12}$. The argument is analogous, using the two-sided $\mathcal{F}_r$-preservation of P_0 (which includes both G^z-preservation and $(G^z)^\top$-preservation, because $\mathcal{F}_r$ includes both when we track the temporal case properly).

This completes the proof of Lemma 6.6.3. □

C.0.3 Proof of Proposition 6.6.4

We prove the proposition by induction on n.

Base case: $n = 1$**.** Trivial: set $c_\star := c_1$ and $P_\star := P_1$. The conclusion $v = c_\star \circ P_\star^\top$ with $c_\star$ being $\mathcal{F}_1$-preserving holds by hypothesis.

Base case: $n = 2$**.** Lemma 6.6.3 applied directly to the two decompositions $v = c_1 \circ P_1^\top = c_2 \circ P_2^\top$. The merged $(c_\star, P_\star) := (c_{12}, P_{12})$ is $(\mathcal{F}_1 \cup \mathcal{F}_2)$-preserving.

Inductive step: $n - 1 \to n$**, for** $n \geq 3$**.** Assume the proposition holds for $n - 1$. Given n decompositions $\{v = c_k \circ P_k^\top\}_{k=1}^n$, apply the inductive hypothesis to the first $n - 1$ decompositions to obtain a merged pair $(c_{[n-1]}, P_{[n-1]})$ with

$$v \;=\; c_{[n-1]} \circ P_{[n-1]}^\top, \qquad c_{[n-1]} \text{ is } \big(\textstyle\bigcup_{k=1}^{n-1} \mathcal{F}_k\big)\text{-preserving.} \tag{C.7}$$

Now we have two decompositions of v:

$$v \;=\; c_{[n-1]} \circ P_{[n-1]}^\top \;=\; c_n \circ P_n^\top, \tag{C.8}$$

with $c_{[n-1]}$ being $(\bigcup_{k<n} \mathcal{F}_k)$-preserving and c_n being $\mathcal{F}_n$-preserving.

Apply Lemma 6.6.3 with

$$(\mathcal{F}_1', \mathcal{F}_2') \;:=\; \big(\textstyle\bigcup_{k=1}^{n-1} \mathcal{F}_k,\; \mathcal{F}_n\big) \tag{C.9}$$

as the two families being merged. The lemma produces $(c_\star, P_\star)$ with $v = c_\star \circ P_\star^\top$ and $c_\star$ being $\big(\bigcup_{k=1}^{n-1} \mathcal{F}_k \cup \mathcal{F}_n\big)$-preserving. This equals $\big(\bigcup_{k=1}^{n} \mathcal{F}_k\big)$-preserving.

Induction closes.

Explicit form of the merged permutation. Tracing the construction: at each inductive step, the merged permutation is $P_{[k]} = P_{[k-1]} P_0^{(k)}$, where $P_0^{(k)}$ is the bridge

permutation introduced by Lemma 6.6.3 at step k. Unrolling:

$$P_\star \;=\; P_{[n]} \;=\; P_1 \cdot P_0^{(2)} \cdot P_0^{(3)} \cdots P_0^{(n)}. \tag{C.10}$$

Each $P_0^{(k)}$ is the permutation extracted from $C_{[k-1]}$ via Lachapelle et al. (2024a, Lemma 2) at inductive step k. Note that $P_\star$ is not in general equal to any single P_k; it is a genuine combination built up through the merging process.

The graph alignments propagate similarly: at each inductive step, the alignment $\tilde{G}_\varphi = P_{[k]}^\top G_\varphi P_{[k]}$ (or one-sided for action graphs) is preserved by the bridge construction, as verified in the "Graph alignments" paragraph of Appendix C.0.2.