Learning Nonlinear Causal Reductions to Explain Reinforcement Learning Policies

Armin Kekić[1] **Jan Schneider**[1] **Dieter Büchler**[1,2] **Bernhard Schölkopf**[1,3,4,*] **Michel Besserve**[1,5,*]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]University of Alberta, Canada
[3]Tübingen AI Center, Tübingen, Germany
[4]ELLIS Institute, Tübingen, Germany
[5]Technische Universität Braunschweig, Germany
[*]Joint supervision

Abstract

Why do reinforcement learning (RL) policies fail or succeed? This is a challenging question due to the complex, high-dimensional nature of agent-environment interactions. In this work, we take a causal perspective on explaining the behavior of RL policies by viewing the states, actions, and rewards as variables in a low-level causal model. We introduce random perturbations to policy actions during execution and observe their effects on the cumulative reward, learning a simplified high-level causal model that explains these relationships. To this end, we develop a nonlinear Causal Model Reduction framework that ensures approximate interventional consistency, meaning the simplified high-level model responds to interventions in a similar way as the original complex system. We prove that for a class of nonlinear causal models, there exists a unique solution that achieves exact interventional consistency, ensuring learned explanations reflect meaningful causal patterns. Experiments on both synthetic causal models and practical RL tasksincluding pendulum control and robot table tennisdemonstrate that our approach can uncover important behavioral patterns, biases, and failure modes in trained RL policies.

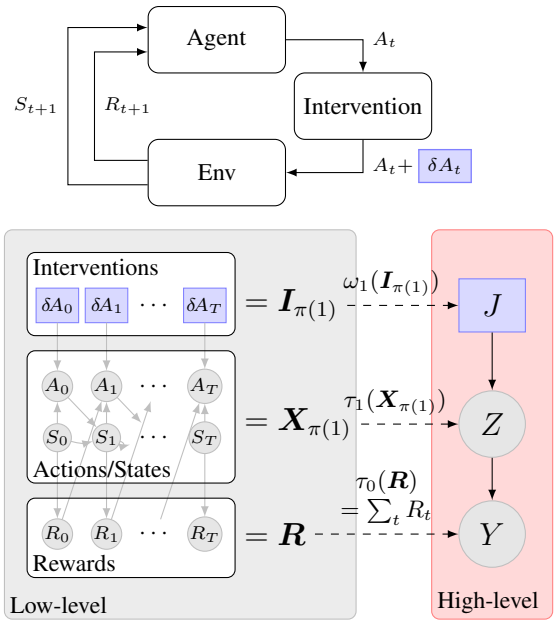
1 INTRODUCTION

In recent years, reinforcement learning (RL) has demonstrated remarkable successes in diverse domains, from achieving superhuman performance in games like Go [1] and Atari [2], to enabling sophisticated control in robotics [3] and optimizing resource management in computer networks [4]. The field has also seen significant adoption in real-world applications, including autonomous systems [5], recommendation engines [6], and process optimization [7]. As RL systems continue to be deployed in increasingly consequential settings, understanding the behavior and decision-making processes of trained policies becomes a practical necessity for ensuring reliability, safety, and trust. Hence, a natural and critical question arises: *"Why did a policy fail or succeed?"* Practitioners need robust explanations when and why policies exhibit unexpected behaviors. Identifying specific failure modes can guide more efficient training regimes and enable improvements to policy architecture or learning algorithms.

Despite their impressive capabilities, understanding the behavior of trained RL policies is challenging. These policies often use parameter-rich neural networks to map from complex observation spaces to actions through mechanisms that are not easily accessible to human intuition. Standard performance metrics, such as cumulative reward, provide only limited insight into the behavior of an RL agent. Moreover, the credit assignment problemdetermining which specific actions contributed most significantly to eventual outcomesis a fundamental obstacle to developing policy explanations.

In this work, we address these challenges by taking a causal perspective on explaining the behavior of learned RL policies, summarized in Fig. 1. We formulate the problem as a *Causal Model Reduction* (CMR), where we treat the joint system of actions, environment variables, and rewards as a complex *low-level causal model*. To probe the low-level model, we introduce random perturbations to policy actions as interventions and observe their impact on cumulative rewards. We learn a simplified *high-level causal model* that summarizes the most important factors determining differences in the expected reward. As our main learning signal, we use *interventional consistency*: the low- and high-level models should respond to interventions in a similar fashion. The map from the original causal model to the reduced

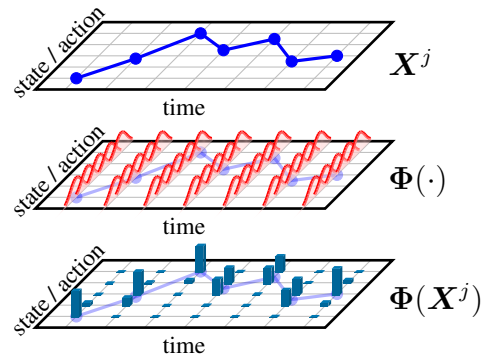
Accepted for the 1st Workshop on Causal Abstractions and Representations (CAR) at UAI 2025 (CAR 2025).



(a) **From Reinforcement Learning Policies to Causal Explanations.**

$$\begin{tikzcd}
P_{\mathcal{L}}(X) \arrow[r, "\tau"] \arrow[d, "i"'] & \widehat{P}_{\tau}^{(0)}(Z,Y) \approx P_{\mathcal{H}}(Z,Y) \arrow[d, "\omega(i)"] \\
P_{\mathcal{L}}^{(i)}(X) \arrow[r, "\tau"] & \widehat{P}_{\tau}^{(i)}(Z,Y) \approx P_{\mathcal{H}}^{(\omega(i))}(Z,Y)
\end{tikzcd}$$

(b) **Approximately Commutative Diagram.**



(c) **Interpretable Reduction Function Class.**

Figure 1: **Learning Causal Explanations of RL Policies.** (a) shows how RL policies are translated to a Causal Model Reduction problem. We sample episodes from the interactions between a trained agent and its environment, where the sampled actions are augmented through shift interventions δA_t before they are executed. We treat the episode variables as nodes in a low-level causal graph, with the shifts δA_t acting as interventions on the actions A_t. Map τ condenses the low-level causal variables to a simpler high-level model with two variables: the target Y, summarizing the rewards, and the cause Z, summarizing the subset $\pi(1)$ of low-level nodes. Similarly, ω_1 maps the vector of low-level interventions $\boldsymbol{I}$ to its high-level counterpart J. The main learning signal is shown in (b). We learn the maps τ and ω by making the diagram approximately commutative by minimizing the divergence between the distributions on each side of the $\approx$ sign. (c) shows a nonlinear interpretable function class that can be used to learn the reduction maps τ_1 and ω_1. A state/action variable's trajectory $\boldsymbol{X}^j$ is encoded through equally spaced Gaussian kernels to a feature vector $\boldsymbol{\Phi}(\boldsymbol{X}^j)$.

one can thus be used to explain the behaviors that are most influential on the success or failure of the RL policy.

Our contributions are: (i) formulating the problem of explaining RL policy behavior as a Causal Model Reduction and developing a nonlinear extension of Targeted Causal Reduction (TCR) [8] that summarizes the main factors explaining a target phenomenon in complex systems, (ii) providing theoretical guarantees of solution uniqueness for a broad class of nonlinear models, ensuring unambiguous explanations despite the identifiability challenges of nonlinear systems, (iii) introducing a class of interpretable nonlinear reduction functions that help explain the behavior captured by the learned reductions, (iv) demonstrating experimentally that our approach uncovers behavioral patterns and biases in two trained RL tasks.

2 BACKGROUND

Notation. We use lowercase letters for deterministic variables and capital letters for random variables. $X \sim P$ means X has distribution P. We use boldface for column vectors, and $\boldsymbol{X}_S$ for the subvector of $\boldsymbol{X}$ restricted to the components in set S. The number of elements in a set S is $\#S$.

2.1 STRUCTURAL CAUSAL MODELS (SCMS)

SCMs are a mathematical framework for representing cause-effect relationships in complex systems.

Definition 2.1 (Structural Causal Model [9, 10])**.** An n-dimensional SCM is a triplet $\mathcal{M}{=}(\mathcal{G}, \mathbb{S}, P_{\boldsymbol{U}})$ consisting of: (i) a directed acyclic graph $\mathcal{G}$ with n vertices, (ii) a joint distribution $P_{\boldsymbol{U}}$ over exogenous or noise variables $\{U_j\}_{j \leq n}$, (iii) a set $\mathbb{S}{=}\{X_j := f_j(\mathbf{Pa}_j, U_j), j{=}1, \ldots, n\}$ of structural equations, where $\mathbf{Pa}_j$ are the variables indexed by the set of parents of vertex j in $\mathcal{G}$. This induces a joint distribution $P_{\mathcal{M}}$ over the endogenous variables $\boldsymbol{X} = [X_1, \ldots, X_n]^\intercal$.[1]

The endogenous variables $\boldsymbol{X}$ encode the system's observ-

[1]This SCM definition allows for confounding between variables through the potential lack of independence between the exogenous variables $\{U_j\}$. Although Def. 2.1 uses the common assumption of acyclic graphs for simplicity, our approach is also compatible with some families of causal graphs with cycles. See App. A.1.

ables, where each variable X_j is determined through the deterministic causal mechanism f_j, its parent variables $\mathbf{Pa}_j$, and the exogenous noise variable U_j. One often encountered class of SCMs are *additive noise models* where structural equations take the simplified form $X_j := f_j(\mathbf{Pa}_j) + U_j$.

Interventions are encoded in SCMs by replacing one or several structural equations. An intervention transforms the original model $\mathcal{M}{=}(\mathcal{G}, \mathbb{S}, P_{\boldsymbol{U}})$ into an intervened model $\mathcal{M}^{(\boldsymbol{i})}{=}(\mathcal{G}^{(\boldsymbol{i})}, \mathbb{S}^{(\boldsymbol{i})}, P_{\boldsymbol{U}}^{(\boldsymbol{i})})$, where $\boldsymbol{i}$ is the vector parameterizing the intervention. The base probability distribution of the unintervened model is denoted $P_{\mathcal{M}}^{(0)}$ or simply $P_{\mathcal{M}}$ and the interventional distribution associated with $\mathcal{M}^{(\boldsymbol{i})}$ is denoted $P_{\mathcal{M}}^{(\boldsymbol{i})}$. In this work, we focus on *shift interventions*, which modify the structural equation of variable X_l by shifting it by a scalar i_l

$$\{X_l := f_l(\mathbf{Pa}_l, U_l)\} \mapsto \{X_l := f_l(\mathbf{Pa}_l, U_l) + i_l\}\,. \quad (2.1)$$

These can be combined to form multi-node interventions with vector parameter $\boldsymbol{i}$.

2.2 CAUSAL MODEL REDUCTIONS (CMRS)

Causal models with a large number of variables can be difficult to interpret and work with. Causal Model Reductions (CMRs) [8] are dimensionality reduction approaches that map such detailed low-level causal models to approximate high-level descriptions with fewer variables while preserving the essential causal properties. They closely relate to several notions of *causal abstraction* [11–17].

Low- and High-level SCMs. We consider two causal models:

- A *low-level SCM* $\mathcal{L}$ with endogenous variables $\boldsymbol{X} \sim P_{\mathcal{L}}$ (on range $\mathcal{X}$) with exogenous variables $\boldsymbol{U} \sim P_{\boldsymbol{U}}$, and set of possible interventions $\mathcal{I}$, leading to interventional distributions $(P_{\mathcal{L}}^{(\boldsymbol{i})})_{\boldsymbol{i}\in\mathcal{I}}$.
- A lower-dimensional *high-level SCM* $\mathcal{H}$ with endogenous variables $\boldsymbol{Z} \sim P_{\mathcal{H}}$ (on range $\mathcal{Z}$), exogenous variables $\boldsymbol{W} \sim P_{\boldsymbol{W}}$, set of interventions $\mathcal{J}$, and distributions $(P_{\mathcal{H}}^{(\boldsymbol{j})})_{\boldsymbol{j}\in\mathcal{J}}$.

$\boldsymbol{X}$ is then *reduced* into a $\mathcal{Z}$-valued high-level random variable using a deterministic map $\tau : \mathcal{X} \to \mathcal{Z}$.

Interventional Consistency. The key criterion for a good reduction is *interventional consistency*: the high-level model should respond to interventions in ways that correspond to the original model's behavior under analogous interventions (*cf.* [11, 16–18]). To formalize this, we consider $\tau_{\#}[P_{\mathcal{L}}]$, the so-called *push-forward distribution* by τ of the low level model distribution, such that

$$\tau(\boldsymbol{X}) \sim \tau_{\#}[P_{\mathcal{L}}]\,. \quad (2.2)$$

Interventional distributions are pushed forward in the same way from the low-level to the high-level, allowing to define the notion of an *exact transformation*:

Definition 2.2 (Exact transformation [17])**.** A map $\tau : \mathcal{X} \to \mathcal{Z}$ is an exact transformation from $\mathcal{L}$ to $\mathcal{H}$ if it is surjective, and there exists a surjective *intervention map* $\omega : \mathcal{I} \to \mathcal{J}$ such that for all $\boldsymbol{i} \in \mathcal{I}$

$$\tau_{\#}\left[P_{\mathcal{L}}^{(\boldsymbol{i})}\right] = P_{\mathcal{H}}^{(\omega(\boldsymbol{i}))}\,. \quad (2.3)$$

That is, regardless of whether we first intervene through $\boldsymbol{i}$ in the low-level model $\mathcal{L}$ and then map to the high-level $\mathcal{H}$, or first map to $\mathcal{H}$ and intervene with $\omega(\boldsymbol{i})$, we arrive at the same distribution.

2.3 TARGETED CAUSAL REDUCTION (TCR)

While exact transformations are theoretically elegant, they present two practical challenges: (i) achieving perfect equality in Eq. (2.3) is often infeasible, so we need a systematic method to learn approximately consistent reductions from data, and (ii) there are many possible consistent reductions and finding a meaningful one needs additional constraints. *Targeted Causal Reduction* (TCR) [8] addresses these challenges by focusing on explaining a specific target variable of interest and providing a concrete learning objective. Below, we describe the case of a single high-level cause, the general case is described in App. A.2.

Definition 2.3 (General TCR Framework)**.** In TCR, we learn transformations (τ, ω) between models $\mathcal{L}$ and $\mathcal{H}$ under the following assumptions:

1. **Target**: $\mathcal{H}$ contains only two nodes: a predefined scalar target variable $Y = \tau_0(\boldsymbol{X})$ that quantifies a phenomenon of interest, and a high-level cause $Z = \tau_1(\boldsymbol{X})$ that needs to be learned.
2. **Parameterized High-level Models**: The high-level SCM is constrained to a class of linear additive Gaussian noise models $\{\mathcal{H}_{\boldsymbol{\gamma}}\}_{\gamma\in\Gamma}$ with parameters $\boldsymbol{\gamma}$ to be learned, implying that (i) the mechanism $Z \to Y$ is linear, with (linear) causal coefficient α and additive noise W_0 and (ii) the exogenous variables $\{W_0, W_1 = Z\}$ have a factorized Gaussian distribution $P_{\boldsymbol{W}} = P_{W_0}P_{W_1}$.
3. **Constructive Transformations**: The respective components of the maps τ and ω associated to the high-level cause and effect depend on disjoint subsets $\pi(1)$ and $\pi(0)$ of low-level variables:

$$\tau(\boldsymbol{x}) = (\bar{\tau}_0(\boldsymbol{x}_{\pi(0)}), \bar{\tau}_1(\boldsymbol{x}_{\pi(1)})\,, \quad (2.4)$$

$$\omega(\boldsymbol{i}) = (0, \bar{\omega}_1(\boldsymbol{i}_{\pi(1)}))\,. \quad (2.5)$$

the first component of ω is set to zero to prevent direct high-level interventions on Y, forcing changes in Y to be explain through high-level interventions on Z.

4. **Approximate Consistency Objective:** Rather than requiring exactness of Eq. (2.3), TCR minimizes the consistency loss, quantifying an average discrepancy D between the push-forward interventional distributions and their corresponding high-level model distributions:

$$\mathcal{L}_{\text{cons}} = \mathbb{E}_{\boldsymbol{i}\sim P_{\boldsymbol{I}}}\left[\mathrm{D}\left(\widehat{P}_{\tau}^{(\boldsymbol{i})}(Y,\boldsymbol{Z}) \| P_{\mathcal{H}}^{(\omega(\boldsymbol{i}))}(Y,\boldsymbol{Z})\right)\right], \tag{2.6}$$

with $\widehat{P}_{\tau}^{(\boldsymbol{i})} = \tau_{\#}\left[P_{\mathcal{L}}^{(\boldsymbol{i})}\right]$.

The relationship between those distributions and the learned reduction maps is shown in Fig. 1b. The expectation is taken over a prior distribution of interventions $P_{\boldsymbol{I}}$. [2]

Intuition behind TCR. TCR provides a form of *etiological explanation* for the target phenomenon Y, identifying the origins of its variation. It accomplishes this by creating an interpretable formulation where changes in Y due to low-level interventions are explained through a high-level causal mechanism. TCR forces explanations to flow through the learned high-level cause Z. When asked *"What causes changes in Y?"*, TCR answers with a specific causal statement: *"Y is changed by acting on the high-level cause $Z = \tau_1(X_{\pi(1)})$ through intervention $\omega(i_{\pi(1)})$"*. The interpretability of the maps τ and ω provides valuable insights: τ reveals which properties of the system influence Y, while ω shows which types of interventions most effectively change these properties.

Linear TCR. The TCR formulation by Kekić et al. [8] imposes some significant constraints: (i) **Linear transformations** for τ and ω, simplifying analysis and facilitating interpretability but limiting expressivity; and (ii) **Gaussian KL-divergence approximation** for the discrepancy D in Eq. (2.6), where distributions are approximated as Gaussians with matched means and variances (see App. A.3), taking a differentiable closed-form and allowing easy optimization.

3 FROM RL POLICIES TO CAUSAL MODEL REDUCTION

RL episodes inherently form causal chains where agent observations lead to actions, which cause environmental changes resulting in new states and rewards. These interactions naturally map to structural causal models. To identify the causal impact of specific actions on overall performance, we need to observe counterfactual outcomes—what would have happened if the agent had acted differently. We achieve this by introducing controlled perturbations to the policy's chosen actions and observing their effects. Specifically, we construct episodes where each action A_t selected by the policy is perturbed by a small random shift $\delta A_t \sim \mathcal{N}(0, \sigma_t)$ before being executed in the environment:

$$(S_0,\, A_0{+}\delta A_0,\, R_1, \ldots,\, A_{T-1}{+}\delta A_{T-1},\, R_T,\, S_T) \tag{3.1}$$

These perturbations serve as shift interventions in our causal framework and allow us to observe how deviations from the policy's intended actions propagate to the cumulative reward.

To formalize this as a TCR problem, we define the low-level endogenous variables $\boldsymbol{X}$ consisting of states and actions $\boldsymbol{X}_{\pi(1)}{=}(A_0, \ldots, A_{T-1}, S_0, \ldots, S_T)^\intercal$ and rewards $\boldsymbol{X}_{\pi(0)}{=}\boldsymbol{R}{=}(R_1, \ldots, R_T)^\intercal$, as shown in Fig. 1a. The interventions are $\boldsymbol{I}_{\pi(1)}{=}(\delta A_0, \ldots, \delta A_{T-1}, 0, \ldots, 0)^\intercal$, where we intervene only on actions and not on states, and the target variable is the cumulative reward $Y{=}\sum_{t=1}^{T} R_t$.

With this formulation, we can now apply the TCR framework to learn a simplified high-level model with one high-level cause Z that best explains variations in the target variable Y. The reduction maps τ_1 and ω_1 identify which aspects of the states and actions most significantly influence the overall performance, thereby providing an interpretable causal explanation of policy behavior.

4 NONLINEAR TARGETED CAUSAL REDUCTION (NTCR)

While the linear TCR framework introduced in Sec. 2.3 provides a principled approach to learning simplified causal models, many real-world phenomenaparticularly in reinforcement learninginvolve inherently nonlinear relationships that cannot be adequately captured by linear maps. In this section, we extend TCR to nonlinear settings while preserving its key benefits of interpretability and causal consistency. This extension, which we call nonlinear TCR (nTCR), allows us to discover and represent complex causal patterns in high-dimensional systems such as RL policies. Despite the added flexibility of nonlinear functions, we show that our approach maintains strong theoretical guarantees. Specifically, we demonstrate both the existence and uniqueness of reductions with exact interventional consistency for a broad class of nonlinear causal models.

4.1 NORMALITY REGULARIZATION

The original discrepancy of linear TCR in Eq. (2.6) does not enforce distributions to fully match (see App. A.3), and the nonlinear reduction can therefore lead to a highly non-Gaussian high-level cause. We therefore introduce a normality regularization encouraging $\tau_1(\boldsymbol{X})$ to be Gaussian. In

[2] Remarks: In practice, $\pi(0)$ and $\pi(1)$ can be fixed based on domain knowledge, with $\pi(1)$ typically encompassing all variables potentially influencing the target that are not in $\pi(0)$. The intervention prior $P_{\boldsymbol{I}}$ can be uninformative (*e.g.* i.i.d. Gaussian) or informed by domain expertise.

addition to being more faithful to the initial idea of Def. 2.2, this allows for easier interpretation of the learned causes (as it enforces a simple unimodal cause distribution). The theory behind the choice of consistency objective is further discussed in App. D.

Given the push-forward distribution $\widehat{P}_{\tau}^{(\boldsymbol{i})}(Z)$ (with CDF $F_{\widehat{P}_{\tau,\text{std}}^{(\boldsymbol{i})}(Z)}(x)$) of the high-level cause under intervention $\boldsymbol{i}$, we first standardize this distribution to zero mean and unit variance to obtain $\widehat{P}_{\tau,\text{std}}^{(\boldsymbol{i})}(Z)$. We then measure the deviation from normality as the 1-Wasserstein distance W_1 between this standardized distribution and the standard normal distribution $\mathcal{N}(0,1)$ (with CDF $\Phi(x)$):

$$\mathcal{L}_{\text{norm}} = \mathbb{E}_{\boldsymbol{i}\sim P(\boldsymbol{i})}\left[W_1(\widehat{P}_{\tau,\text{std}}^{(\boldsymbol{i})}(Z), \mathcal{N}(0,1))\right] \tag{4.1}$$

$$= \mathbb{E}_{\boldsymbol{i}\sim P(\boldsymbol{i})}\left[\int_{\mathbb{R}} |F_{\widehat{P}_{\tau,\text{std}}^{(\boldsymbol{i})}(Z)}(x) - \Phi(x)|\, dx\right] . \tag{4.2}$$

The total optimization objective for nTCR then becomes $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cons}} + \eta_{\text{norm}}\mathcal{L}_{\text{norm}}$. In practice, we compute $\mathcal{L}_{\text{norm}}$ using a differentiable approximation of the CDF and evaluating the L_1 distance at a finite set of points, as presented in App. C.1. This approach allows for efficient optimization while maintaining the desired distributional properties of the high-level causes.

4.2 EXACT SOLUTIONS

Nonlinear TCR allows a much larger space of functions to fit the learning objective. This can lead to overfitting and even potential non-identifiability issues where multiple reductions could exist, which would strongly limit the interpretability of the approach. The following statement provides identifiability guarantees even in the nonlinear case, as long as we obtain exact solutions.

Proposition 4.1 (Uniqueness)**.** *Assume (i) the low-level model is an additive noise SCM of the form*

$$\begin{bmatrix}\boldsymbol{X}_{\pi(1)},\\ \boldsymbol{X}_{\pi(0)}\end{bmatrix} := \begin{bmatrix}\boldsymbol{f}_1(\boldsymbol{X}_{\pi(1)}) + \boldsymbol{U}_{\pi(1)} + \boldsymbol{i}_{\pi(1)}\,,\\ \boldsymbol{f}_0(\boldsymbol{X}_{\pi(0)}, \boldsymbol{X}_{\pi(1)}) + \boldsymbol{U}_{\pi(0)}\end{bmatrix}, \tag{4.3}$$

$$\boldsymbol{U}_{\pi(1)} \sim P_1 \text{ indep. of } \boldsymbol{U}_{\pi(0)} \sim P_0\,, \tag{4.4}$$

(ii) P_1 admits a probability density with non-vanishing Fourier transform, (iii) The high-level model has a non-zero causal effect ($\alpha \neq 0$). Then, if there exists a constructive transformation following Def. 2.3 that is exact for all $\boldsymbol{i} \in \mathbb{R}^{\#\pi(1)}$, it is also unique up to multiplicative and additive constants.

See App. B.2 for the proof. The above result does not guarantee the existence of such exact transformations. The following proposition shows we can construct families of low-level additive noise models that admit a nonlinear exact transformation $\tau_1(\boldsymbol{X}_{\pi(1)}) \to \tau_0(\boldsymbol{X}_{\pi(0)})$.

Proposition 4.2 (Existence)**.** *Assume a low-level additive SCM following Eq.* (D.2)*. Assume additionally that (i) $\boldsymbol{U}$ is jointly Gaussian, (ii) $\boldsymbol{f}_0(\boldsymbol{X}_{\pi(0)}, \boldsymbol{X}_{\pi(1)}) = \boldsymbol{h}_0(\boldsymbol{X}_{\pi(0)}) + B(\boldsymbol{X}_{\pi(1)} - \boldsymbol{f}_1(\boldsymbol{X}_{\pi(1)}))$, for some functions $\boldsymbol{h}_0, \boldsymbol{f}_1$, some matrix $B \in \mathbb{R}^{\#\pi(0)\times\#\pi(1)}$, and (iii) $Y = \tau_0(\boldsymbol{X}_{\pi(0)}) = \boldsymbol{a}^\top(\boldsymbol{X}_{\pi(0)} - \boldsymbol{h}_0(\boldsymbol{X}_{\pi(0)}))$ for arbitrary non-vanishing $\boldsymbol{a} \in \mathbb{R}^{\#\pi(0)}$, then the transformation defined by the following maps is exact*

$$\bar{\tau}_1(\boldsymbol{X}_{\pi(1)}) = \boldsymbol{a}^\top B(\boldsymbol{X}_{\pi(1)} - \boldsymbol{f}_1(\boldsymbol{X}_{\pi(1)}))\,, \tag{4.5}$$

$$\bar{\omega}_1(\boldsymbol{i}_{\pi(1)}) = \boldsymbol{a}^\top B\boldsymbol{i}_{\pi(1)}\,. \tag{4.6}$$

See App. B.1 for the proof. Although we do not expect to cover all possible nonlinear exact transformations with this class of models, this can be used to validate our algorithms, as discussed in Sec. 5.1.

4.3 INTERPRETABLE NONLINEAR FUNCTION CLASS

When the reduction is allowed to be nonlinear, it can be challenging to interpret what the high-level cause and intervention represent in the system without constraints on the mappings τ_1 and ω_1. For example, RL episodes possess inherent temporal structure that we can exploit to build more interpretable function classes for these mappings. We can express $\mathcal{X}$ as a Cartesian product across d features, with each feature represented as a time series $\mathcal{X}=\mathcal{X}^1\times\cdots\times\mathcal{X}^d$, where $\mathcal{X}^j$ represents the space of possible trajectories for the j-th feature over time. Each feature's trajectory space can be further decomposed as $\mathcal{X}^j=\mathcal{X}_1^j\times\cdots\times\mathcal{X}_T^j$. The intervention space $\mathcal{I}$ can be decomposed similarly. Leveraging this structure, we define our interpretable nonlinear function class for τ_1 as:

$$\tau_1(\boldsymbol{X}) = \sum_{j=1}^{d}\tau_1^j(\boldsymbol{X}^j) = \sum_{j=1}^{d}\sum_{t=1}^{T} w_{j,t}\cdot\phi_{j,t}(X_t^j) \tag{4.7}$$

$$= \sum_{j=1}^{d}\sum_{t=1}^{T} w_{j,t}\cdot\exp\left(-\frac{(x-\mu_{j,t})^2}{2\sigma_{j,t}^2}\right) . \tag{4.8}$$

Where $w_{j,t}$ are learned weights, and X_t^j is the value of feature j at time t. For our implementation, we use Gaussian kernels $\phi_{j,t}$ as the basis functions, where $\{\mu_{j,t}, \sigma_{j,t}\}$ are fixed parameters, set to span the range of values typically encountered, see Fig. 1c.[3] This can be viewed as a continuous one-hot encoding of variable-value pairs across the episode, allowing for smooth interpolation between similar states. This approach ensures that our nonlinear reduction remains interpretable, as we can identify which features, at which time steps, contribute most significantly to the high-level causal explanation by examining the learned weights $w_{j,t}$.

[3]The function ω_1 is defined analogously, operating on the intervention space $\mathcal{I}_{\pi(1)}$.

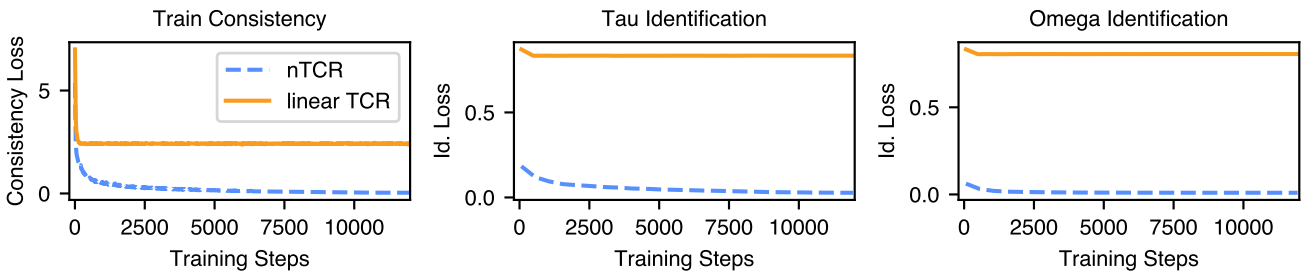


Figure 2: **Identification of Ground-Truth Solutions for Synthetic Low-Level Models.** Consistency loss (left) and the identification losses measuring agreement with the ground-truth solutions (definition in App. E.1) for the τ- and ω-functions (middle and right) over the reduction training run.

5 CASE STUDIES

5.1 EXPERIMENTS ON SYNTHETIC DATA

Before applying our approach to RL problems, we first validate its theoretical properties on synthetic data generated from known low-level causal models. This controlled setting serves two important purposes: (i) it demonstrates that for this model class, nTCR can indeed find the unique solution that perfectly minimizes the causal consistency loss, confirming our theoretical guarantees; and (ii) it validates that our training methodology successfully converges to the correct solution.

We generate synthetic data from 10 randomly sampled low-level causal models following the structure described in Prop. 4.2 with $\dim(\boldsymbol{X}_0) = 1$, $\dim(\boldsymbol{X}_1) = 9$, and $\boldsymbol{h}_0(\boldsymbol{X}_0) \equiv 0$. Each entry in the matrix B is sampled from a standard normal distribution, and we set $\boldsymbol{a} = 1$. For these experiments, we use neural networks as generic function approximators for both τ_1 and ω_1. For more details on the experimental setting, please refer to App. E.1.

Fig. 2 shows the consistency loss and an identification measure that quantifies how closely the learned τ- and ω-maps match the ground truth solutions.[4] We observe that nTCR successfully finds reductions with near-zero consistency loss and converges to the theoretical solution outlined in Prop. 4.2, validating both our theory and implementation.

5.2 PENDULUM

Pendulum [19] is a classic swing-up control task where a pendulum must be brought to the upright position. With limited torque, the agent must develop momentum through strategic oscillations rather than directly moving to the goal position. The state space comprises three variables: x-y coordinates $(\cos(\theta), \sin(\theta))$ and angular velocity, while the action is the applied torque. The reward function penalizes deviation from the upright position, angular velocity, and action magnitude. For the reductions, we choose the more natural feature θ instead of the x- and y-positions. More details on the experiment setting, and the agent and nTCR training are given in App. E.2.

Fig. 3 (left, Policy A) shows that nTCR identifies two trajectory groups with significant reward variations: pendulums starting in the bottom-right corner swinging clockwise (red, higher rewards) versus those starting bottom-left swinging counterclockwise (blue, lower rewards). The reduction shows that the agent performs better clockwise than counterclockwisea surprising bias given that initial conditions are uniformly sampled and the environment has mirror symmetry along the axis of gravitation.

For Policy B (Fig. 3, right), the learned omega map reveals that shifting torque to more negative values would improve performance, particularly towards the end of the episode. Analysis confirms this (Fig. 3, bottom-right): the policy fails to consistently stabilize the pendulum upright, allowing it to tip over in the positive θ direction before applying positive torque for a full rotation recovery. nTCR correctly identifies that applying more negative torque at the top position would prevent this instability.

5.3 TABLE TENNIS

We evaluate our approach on a robot table tennis simulation based on the real-world setup developed in [20, 21]. The environment features a four degrees-of-freedom robot arm actuated by pneumatic artificial muscles, positioned on one side of a table tennis table, shown in Fig. 4 (a,b). The robot's task is to return incoming balls launched by a ball gun to a desired landing point on the opponent's side of the table. The state space includes the robot state (joint angles, joint velocities, and air pressures in each muscle) and the ball state (position and velocity). The action space consists of the changes in desired pressures for each of the eight pneumatic muscles. The task requires precise timing and positioning to successfully intercept the ball and direct it toward the target. More details on the environment, the RL- and nTCR-training are given in App. E.3.

[4] A normalized L2 loss between the learned and ground-truth high-level cause; see App. E.1 for more details.

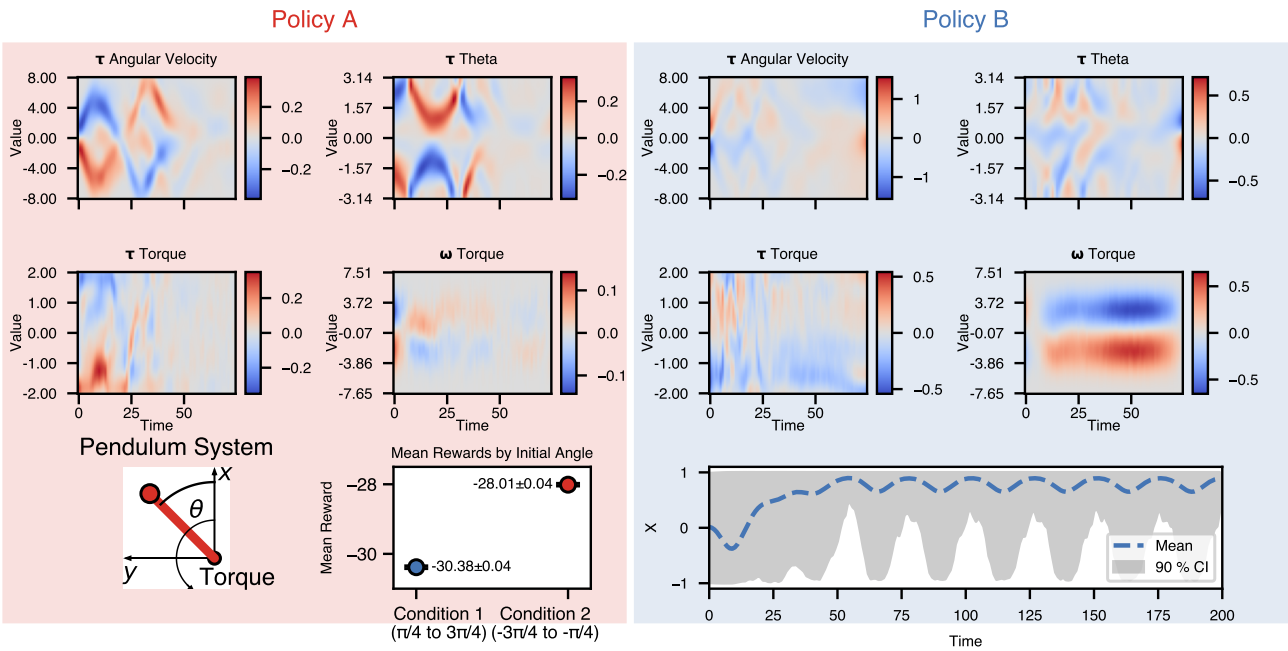


Figure 3: **Pendulum task.** The top two rows show the learned nTCR τ- and ω-maps for two policies A and B. The heatmaps show the learned reductions $\tau_1^j(\boldsymbol{x}^j)$ and $\omega_1^j(\boldsymbol{i}^j)$, where j indexes the state/action variables (angular velocity, theta, and torque). Note that since we only intervene on the torque, this is the only variable for which there is a nonzero ω-map. The bottom left plot shows the pendulum system setting. The middle plot on the bottom row shows the mean reward for Policy A for pendulums starting in the left quadrant (Condition 1) and those starting in the right quadrant (Condition 2). The standard error of the mean is shown (the error bars are smaller than the data point in the plot). The bottom right plot shows the mean x-position of the pendulum for episodes under Policy B and the 90% confidence interval.

The learned τ maps for the joint positions 0-2 and the ball positions are shown in Fig. 4. We make two key observations: one regarding the movements of the robot arm and another with respect to performance differences across ball trajectories. Joint 0 is the main axis used to swing the robot arm to accelerate the ball (see Fig. 4 (b)). The τ-map for this variable is shown in Fig. 4 (d). The map highlights two critical state regions for joint position 0 during the period when the arm is swinging towards the ball (regions A and B in Fig. 4 (d)): (A) the arm swings back (negative angle values) before the ball arrives and forward when the ball arrives, leading to high rewards, and (B) the arm swings forward too early, leading to low rewards.

The TCR reductions for the ball position reveal two clear trends: (i) balls that travel further toward the outside edge of the table are more difficult to hit (Fig. 4 (g)), and (ii) balls that bounce further from the robot, or closer to the net, present greater challenges (Fig. 4 (h, i)). We validate these findings with an analysis of the missed balls, shown in Fig. 4(c). We observe that more balls miss the racket toward the top compared to the bottom, and more to the left (corresponding to the direction toward the outside of the table) than to the right. These observations are consistent with the τ reduction values for joint positions 1 and 2, shown in Fig. 4 (h, i). Around the time of ball contact, they have their largest positive contributions near $\pi/4$—corresponding to a relatively high and further out racket—and their largest negative contributions for larger angles—corresponding to a lower racket closer to its base.

6 RELATED WORK

Explainable reinforcement learning (XRL) focuses on understanding the decision-making processes of learning RL agents. Milani et al. [22] propose a taxonomy for XRL approaches into three categories: feature importance (FI) methods explaining the immediate context for single actions (*e.g.* [23–25]), learning process methods (LPM) that reveal influential experiences from training (*e.g.* [26–28]), and policy-level (PL) methods that summarize the agent's behavior as a whole. Our work is a policy-level explanation [29–31], and specifically, our nTCR is a technique that extracts abstract states, as we reduce the dimensionality of the states, actions, and reward spaces into high-level variables. A notable causal approach to XRL is Madumal et al. [32], who introduce action influence models based on SCMs to generate contrastive explanations for agent behavior through counterfactual analysis, though their approach differs from ours in that they require a predefined causal graph structure and focus on explaining individual actions rather than identifying high-level causal patterns across episodes.

Our approach is an extension of linear TCR [8] which is itself grounded in theoretical work on causal abstraction [11–14, 17]. While these works establish formal conditions for

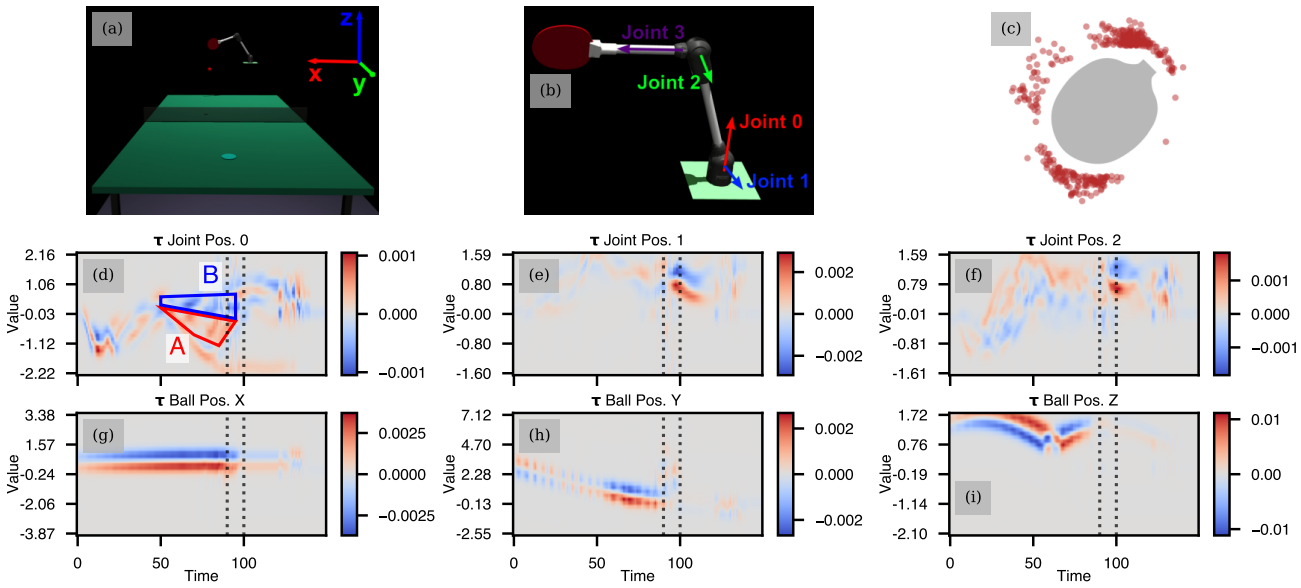


Figure 4: **Table tennis task.** The task involves training a robot arm to return incoming balls to a target location on the opponent's side of the table (a). The robot arm and its rotational axes are shown in (b). (c) shows the positions of 400 balls that the robot missed relative to the racket. The racket is shown in the typical angle it is in when it hits the ball. (d-i) show the learned τ reduction maps for the joint positions 0-2 and the ball position. The reduction maps for the other variables are shown in App. E.3. The dotted vertical lines show the time range where the robot typically hits the ball.

valid causal abstractions, relatively few have addressed the challenge of learning such abstractions from low-level systems. Geiger et al. [15, 33] focus on language models, where high-level variables and interpretations are already available and can be used to constrain a neural network to align with a desired high-level causal structure. Our approach takes the opposite direction: we develop a general approach to building the high-level abstraction from the ground up. This is conceptually related to causal feature learning [16], which partitions input and output spaces to identify discrete high-level models consistent with low-level observations and interventions that fix the value of the causes. In contrast, TCR differs in two key aspects: it uses shift interventions that are more natural perturbations for many natural and engineered low-level systems, and focuses on continuous rather than strictly discrete high-level representations.

7 DISCUSSION

In this work, we have presented nonlinear Targeted Causal Reduction (nTCR), a nonlinear dimensionality reduction framework, and have shown that it can be used to generate policy-level explanations for the behavior of RL approaches. We have experimentally demonstrated that nTCR can identify biases and failure modes in trained RL policies on Pendulum and a robot table tennis simulation.

Our theoretical analysis in Sec. 4 provides uniqueness and existence guarantees of non-linear exact transformations. However, those are limited to the class of nonlinear causal models presented in Props. 4.1 and 4.2, and extending these guarantees to more general model classes should be addressed by future work. Notably, dropping the linear and Gaussian high-level model assumptions may be needed in some applications, although it poses further interpretability challenges.

While nTCR overcomes the limitations of linear TCR by allowing for arbitrary functions as reduction maps, one of the central challenges for methods generating explanations is balancing expressivity with interpretability. Our Gaussian kernel-based function class (Sec. 4.3) represents a middle ground, offering more expressivity than linear maps while maintaining interpretability. More complex or less structured state/action data, such as for policies using on image inputs may require additional steps for representing them appropriately, *e.g.* segmentation methods [34] or saliency maps [23, 35].

Finally, we use shift interventions in our theory and experiments, which naturally align with continuous action spaces in RL. However, there are no fundamental restrictions preventing the use of other interventions more adapted to discrete action spaces, which are common in many RL settings. Those could be implemented as perturbations of the continuous parameters controlling the policy function used to sample discrete actions.

References

[1] 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*, 2016.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv:1312.5602*, 2013.

[3] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 2013.

[4] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM workshop on hot topics in networks*, 2016.

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv:1604.07316*, 2016.

[6] M Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 2022.

[7] Rui Nian, Jinfeng Liu, and Biao Huang. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 2020.

[8] Armin Kekić, Bernhard Schölkopf, and Michel Besserve. Targeted Reduction of Causal Models. *UAI*, 2024.

[9] Judea Pearl. *Causality*. Cambridge university press, second edition, 2009.

[10] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference – Foundations and Learning Algorithms*. MIT Press, 2017.

[11] Sander Beckers and Joseph Y Halpern. Abstracting Causal Models. *AAAI*, 2019.

[12] Sander Beckers, Frederick Eberhardt, and Joseph Y Halpern. Approximate Causal Abstractions. *UAI*, 2020.

[13] Riccardo Massidda, Atticus Geiger, Thomas Icard, and Davide Bacciu. Causal Abstraction with Soft Interventions. *CLeaR*, 2023.

[14] Eigil F Rischel and Sebastian Weichwald. Compositional abstraction error and a category of causal models. *UAI*, 2021.

[15] Atticus Geiger, Chris Potts, and Thomas Icard. Causal abstraction for faithful model interpretation. *arXiv:2301.04709*, 2023.

[16] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. *UAI*, 2015.

[17] Paul K Rubenstein, Sebastian Weichwald, Stephan Bongers, Joris M Mooij, Dominik Janzing, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Causal consistency of structural equation models. *UAI*, 2017.

[18] Hamza Keurti, Hsiao-Ru Pan, Michel Besserve, Benjamin F Grewe, and Bernhard Schölkopf. Homomorphism Autoencoder–Learning Group Structured Representations from Observed Transitions. *ICML*, 2023.

[19] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv:2407.17032*, 2024.

[20] Dieter Büchler, Simon Guist, Roberto Calandra, Vincent Berenz, Bernhard Schölkopf, and Jan Peters. Learning to play table tennis from scratch using muscular robots. *IEEE Transactions on Robotics*, 2022.

[21] Simon Guist, Jan Schneider, Hao Ma, Le Chen, Vincent Berenz, Julian Martus, Heiko Ott, Felix Grüninger, Michael Muehlebach, Jonathan Fiene, Bernhard Schölkopf, and Dieter Büchler. Safe & accurate at speed with tendons: A robot arm for exploring dynamic motion. In *Robotics: Science and Systems*, 2024.

[22] Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. Explainable reinforcement learning: A survey and comparative review. *ACM Computing Surveys*, 2024.

[23] Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding Atari agents. *ICML*, 2018.

[24] Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[25] Alexander Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo Jimenez Rezende. Towards interpretable reinforcement learning using attention augmented agents. *NeurIPS*, 2019.

[26] Francisco Cruz, Richard Dazeley, Peter Vamplew, and Ithan Moreira. Explainable robotic systems: Understanding goal-driven actions in a reinforcement learning scenario. *Neural Computing and Applications*, 2023.

[27] Andrew Anderson, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Alan Fern, and Margaret Burnett. Explaining reinforcement learning to mere mortals: An empirical study. *IJCAI*, 2019.

[28] Ioana Bica, Daniel Jarrett, Alihan Hüyük, and Mihaela van der Schaar. Learning "what-if" explanations for sequential decision-making. *ICLR*, 2021.

[29] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. Graying the black box: Understanding DQNs. *ICML*, 2016.

[30] Nicholay Topin and Manuela Veloso. Generation of policy-level explanations for reinforcement learning. *AAAI*, 2019.

[31] Sarath Sreedharan, Siddharth Srivastava, and Subbarao Kambhampati. TLdR: Policy summarization for factored SSP problems using temporal abstractions. *ICAPS*, 2020.

[32] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. *AAAI*, 2020.

[33] Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *NeurIPS*, 2021.

[34] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *ICCV*, 2023.

[35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *ICCV*, 2017.

[36] Stephan Bongers, Patrick Forré, Jonas Peters, and Joris M Mooij. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 2021.

[37] Jean-François Cardoso. Dependence, correlation and gaussianity in independent component analysis. *JMLR*, 2003.

[38] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines 3: Reliable reinforcement learning implementations. *JMLR*, 2021.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

Learning Nonlinear Causal Reductions to Explain Reinforcement Learning Policies
(Supplementary Material)

Armin Kekić[1] **Jan Schneider**[1] **Dieter Büchler**[1,2] **Bernhard Schölkopf**[1,3,4,*] **Michel Besserve**[1,5,*]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]University of Alberta, Canada
[3]Tübingen AI Center, Tübingen, Germany
[4]ELLIS Institute, Tübingen, Germany
[5]Technische Universität Braunschweig, Germany
[*]Joint supervision

Appendix

CONTENTS

A Additional Background **12**
A.1 Cyclic Structural Causal Models . . . 12
A.2 Targeted Causal Reduction (Multiple Causes) . . . 12
A.3 KL Divergence and Information Geometry . . . 13

B Proofs of Main Text Results **14**
B.1 Analytical Solution . . . 14
B.2 Uniqueness of Exact Transformation . . . 15

C Additional Method Details **15**
C.1 Differentiable Normality Regularization . . . 15

D Additional Theoretical Results **16**

E Experimental Details **17**
E.1 Synthetic Experiments . . . 17
E.2 Pendulum . . . 19
E.3 Table Tennis . . . 20

F Additional Experimental Results **22**
F.1 Effect of Normality Regularization . . . 22
F.2 Linear TCR for Pendulum Policies . . . 23

Accepted for the 1st Workshop on Causal Abstractions and Representations (CAR) at UAI 2025 (CAR 2025).

F.3 Linear TCR for Table Tennis Policies 24

A ADDITIONAL BACKGROUND

A.1 CYCLIC STRUCTURAL CAUSAL MODELS

Definition A.1 (SCM (with cycles allowed))**.** An n-dimensional SCM is a triplet $\mathcal{M} = (\mathcal{G}, \mathbb{S}, P_{\boldsymbol{U}})$ consisting of:

- a joint distribution $P_{\boldsymbol{U}}$ over exogenous variables $\{U_j\}_{j \leq n}$,
- a directed graph $\mathcal{G}$ with n vertices,
- a set $\mathbb{S} = \{X_j := f_j(\mathbf{Pa}_j, U_j), j = 1, \ldots, n\}$ of structural equations, where $\mathbf{Pa}_j$ are the variables indexed by the set of parents of vertex j in $\mathcal{G}$,

such that for almost every $\boldsymbol{u}$, the system $\{x_j := f_j(\mathbf{pa}_j, u_j)\}$ has a unique solution $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{u})$, with $\boldsymbol{g}$ measurable.

The unique solvability condition allows to consider a very general class of SCMs by allowing cycles ($\mathcal{G}$ may not be a DAG, while a DAG would imply unique solvability). Moreover, we allow hidden confounding through the potential lack of independence between the exogenous variables $\{U_j\}$. See Bongers et al. [36] for a thorough study of these models. Under these conditions, the distribution $P_{\boldsymbol{U}}$ entails a well-defined joint distribution over the endogenous variables $P(\boldsymbol{X})$. By unique solvability, we define the mapping from exogenous to endogenous variables of this model under interventions as $M^{(\boldsymbol{i})} : \boldsymbol{u} \mapsto \boldsymbol{x}$ such that $M^{(\boldsymbol{i})}(\boldsymbol{U}) \sim P^{(\boldsymbol{i})}_{\mathcal{M}}(\boldsymbol{X})$.

A.2 TARGETED CAUSAL REDUCTION (MULTIPLE CAUSES)

In this paper, we focus on a single high-level cause for Targeted Causal Reduction, as this is case used to develop our RL explainability method. Here, we present the more general TCR formulation with multiple high-level causes.

TCR Framework with Multiple Causes. The general TCR framework with multiple causes extends the single-cause version as follows:

1. **Target-oriented Structure**: We designate a scalar target variable $Y = \tau_0(\boldsymbol{X})$ that quantifies a phenomenon of interest. The high-level model has $(n{+}1)$ endogenous variables $\{Y, Z_1, \ldots, Z_n\}$, where $Z_1, \ldots, Z_n$ are the learned high-level causes of Y.
2. **Parameterized High-level Models**: The high-level SCM is constrained to a class of linear additive Gaussian noise models $\{\mathcal{H}_{\boldsymbol{\gamma}}\}_{\gamma \in \Gamma}$ with parameters $\boldsymbol{\gamma}$ to be learned. The exogenous variables $\{W_0, W_1, \ldots, W_n\}$ have a factorized Gaussian distribution $P_{\boldsymbol{W}} = \prod P_{W_k}$.
3. **Constructive Transformations**: The linear maps τ and ω are constructive, meaning each dimension depends only on a designated subset of low-level variables:

$$\tau = (\tau_0, \tau_1, \ldots, \tau_n) \text{ with } \tau_k : \boldsymbol{x} \mapsto \bar{\tau}_k(\boldsymbol{x}_{\pi(k)}) \tag{A.1}$$

$$\omega = (\omega_0, \omega_1, \ldots, \omega_n) \text{ with } \omega_k : \boldsymbol{i} \mapsto \bar{\omega}_k(\boldsymbol{i}_{\pi(k)}) \,. \tag{A.2}$$

   Here, the alignment function π partitions the N low-level variables into non-overlapping subsets. Specifically, $\pi(0)$ identifies indices of low-level variables that determine the target Y, while $\pi(1), \ldots, \pi(n)$ identify indices for each of the high-level causes. These subsets satisfy $\pi(k) \cap \pi(l) = \emptyset$ for all $k \neq l$, ensuring each low-level variable contributes to at most one high-level variable.

Our non-linear TCR framework may in principle be extended to this multiple cause setting, but it would require the introduction of extra regularization term in the optimized objective in order to constrain the method to separate distinct causes, as elaborated in [8].

A.3 KL DIVERGENCE AND INFORMATION GEOMETRY

The Kullback-Leibler divergence can be used as a pseudo-metric in the space of probability densities with common support over a given space that we take to be $\mathbb{R}[d]$ for simplicity. By definition for two densities p, q over the same support

$$D_{KL}(p\|q) = \int p(x) \log \frac{p(x)}{q(x)} dx \,.$$

It turns out this can be used to define projections on specific manifolds of probability densities. In particular, one can define the projection of density p on the manifold of Gaussian densities $\mathcal{M}_G$.

$$Proj_G(p) = \mathcal{N}(\boldsymbol{\mu}(p), \mathbf{Var}(p))$$

where $\boldsymbol{\mu}(p)$ and $\mathbf{Var}(p)$ are the mean and covariance matrix associated to density p.

It is easy to check that $Proj_G(p)$ is the closest point to p on the manifold of Gaussian densities in the sense that

$$D_{KL}(p, Proj_G(p)) = \min_{q \in \mathcal{M}_G} D_{KL}(p, q)$$

This moreover leads to a property analogous to the Pythagorean theorem in Euclidean spaces: for any Gaussian density q and any (possibly non-Gaussian) density q we have the decomposition (see [37] for details)

$$D_{KL}(p, q) = D_{KL}(p, Proj_G(p)) + D_{KL}(Proj_G(p), q) \,.$$

While the second term $D_{KL}(Proj_G(p), q)$ has a classic analytic expression that is optimized in linear TCR [8], the first term can be written

$$D_{KL}(p, Proj_G(p)) = H(Proj_G(p)) - H(p)$$

where $H(q) = -\int p(x) \log(p(x)) dx$ denotes the entropy associated to density q. Using invariance of the KL divergence by parameter transformation, we also have

$$D_{KL}(p, Proj_G(p)) = D_{KL}(p_{norm}, \mathcal{N}(0, I_d)) = H(\mathcal{N}(0, I_d)) - H(p_{norm}) \,,$$

where p_{norm} denotes the density obtain by normalizing the first and second order statistics of p (by removing the mean and dividing by the standard deviation), and I_d denotes the identity matrix for the dimension d of the considered space. This term is called negentropy, as it is up to an additive constant the opposite of the entropy, with the additional specificity that it vanishes when the density is Gaussian (hence with maximum entropy).

The entropy is very challenging to estimate for arbitrary distributions. In contrast, the KL divergence between two (possibly multivariate) Gaussians has an analytic expression.

$$\begin{aligned} &D_{KL}(\mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1) \| \mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2)) \\ &\qquad = \frac{1}{2} \left[\log(\det(\Sigma_2)/\det(\Sigma_1)) - d + \mathrm{tr}(\Sigma_2^{-1}\Sigma_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \Sigma_2^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \right] \end{aligned} \tag{A.3}$$

Hence, [8] use a Gaussian KL-divergence approximation to optimize linear TCR, defined as

$$D_G(p, q) = D_{KL}(Proj_G(p), q) \,. \tag{A.4}$$

where q is the density of the high-level model, and p is the push-forward density of the low-level model. Because q is assumed Gaussian, this discrepancy has an analytic expression which allows optimizing the loss of Eq. (2.6). From the above development we immediately get that

$$D_G(p, q) \leq D_{KL}(p, q) = D_{KL}(p, Proj_G(p)) + D_{KL}(Proj_G(p), q) \,. \tag{A.5}$$

and in particular, the inequality is strict if p is not Gaussian. As a consequence, minimizing D_G does not guarantee a priori that the KL divergence is minimized.

The resulting expression of the consistency loss for linear TCR used by Kekić et al. [8] is thus

$$\mathcal{L}_{\text{cons}}(\tau, \omega, \gamma) = \mathbb{E}_{\boldsymbol{i} \sim P_{\boldsymbol{I}}} \left[D_G \left(\widehat{P}_\tau^{(\boldsymbol{i})}(Y, \boldsymbol{Z}) \| P_{\mathcal{H}}^{(\omega(\boldsymbol{i}))}(Y, \boldsymbol{Z}) \right) \right] \,, \text{ with } \widehat{P}_\tau^{(\boldsymbol{i})} = \tau_\# \left[P_{\mathcal{L}}^{(\boldsymbol{i})} \right] \,, \tag{A.6}$$

with the above defined D_G.

B PROOFS OF MAIN TEXT RESULTS

B.1 ANALYTICAL SOLUTION

Proposition 4.2 (Existence)**.** *Assume a low-level additive SCM following Eq.* (D.2)*. Assume additionally that (i)* $\boldsymbol{U}$ *is jointly Gaussian, (ii)* $\boldsymbol{f}_0(\boldsymbol{X}_{\pi(0)}, \boldsymbol{X}_{\pi(1)}) = \boldsymbol{h}_0(\boldsymbol{X}_{\pi(0)}) + B(\boldsymbol{X}_{\pi(1)} - \boldsymbol{f}_1(\boldsymbol{X}_{\pi(1)}))$*, for some functions* $\boldsymbol{h}_0, \boldsymbol{f}_1$*, some matrix* $B \in \mathbb{R}^{\#\pi(0)\times\#\pi(1))}$*, and (iii)* $Y = \tau_0(\boldsymbol{X}_{\pi(0)}) = \boldsymbol{a}^\top(\boldsymbol{X}_{\pi(0)} - \boldsymbol{h}_0(\boldsymbol{X}_{\pi(0)}))$ *for arbitrary non-vanishing* $\boldsymbol{a} \in \mathbb{R}^{\#\pi(0)}$*, then the transformation defined by the following maps is exact*

$$\bar{\tau}_1(\boldsymbol{X}_{\pi(1)}) = \boldsymbol{a}^\top B(\boldsymbol{X}_{\pi(1)} - \boldsymbol{f}_1(\boldsymbol{X}_{\pi(1)}))\,, \tag{4.5}$$

$$\bar{\omega}_1(\boldsymbol{i}_{\pi(1)}) = \boldsymbol{a}^\top B\boldsymbol{i}_{\pi(1)}\,. \tag{4.6}$$

Proof. Because of the choice of τ_0, we have

$$Y = \boldsymbol{a}^\top B\left(\boldsymbol{X}_1 - \boldsymbol{f}_1(\boldsymbol{X}_1)\right) + \boldsymbol{a}^\top\boldsymbol{U}_0 = \boldsymbol{a}^\top B\left(\boldsymbol{U}_1 + \boldsymbol{i}_{\pi(1)}\right) + \boldsymbol{a}^\top\boldsymbol{U}_0\,.$$

Such that for some constant "Cst." independent of $\boldsymbol{i}_{\pi(1)}$

$$\mathbb{E}[Y] = Cst. + \boldsymbol{a}^\top B\boldsymbol{i}_{\pi(1)}\,.$$

Moreover, the high-level model's interventional marginal distribution of Y is given by

$$Y = \alpha Z + W_0 = \tau_1(\boldsymbol{X}^{(0)}_{\pi(1)}) + \alpha\omega_1(\boldsymbol{i}_{\pi(1)}) + W_0 = \alpha\tau_1((\mathbf{id} - \boldsymbol{f}_1)^{-1}(\boldsymbol{U}_1)) + \alpha\omega_1(\boldsymbol{i}_{\pi(1)}) + W_0\,,$$

with W_0 an exogenous variable independent of $\tau_1(\boldsymbol{X}_1)$, where we use the map "$(\mathbf{id} - \boldsymbol{f}_1)^{-1}$" from exogenous to endogenous variables of the low-level model restricted to nodes in $\pi(1)$. Note that "$(\mathbf{id} - \boldsymbol{f}_1)$" is invertible by directed acyclic assumption on the graph. If we would allow cycles in the graph, this condtion would be obtained by the unique solvability assumption of [36]. By matching the expectation of Y, this implies that up to an arbitrary additive constant β (that depends on the choice of τ_1, we have

$$\alpha\omega_1(\boldsymbol{i}_1) = \beta + \boldsymbol{a}^\top B\boldsymbol{i}_{\pi(1)}\,.$$

and we fix $\alpha = 1$ to remove one degree of freedom (τ_1 can be chosen accordingly). Now by cause consistency we get that the pushforward distribution associated to

$$\tau_1(\boldsymbol{X}^{(\boldsymbol{i})}_{\pi(1)}) = \tau_1((\mathbf{id} - \boldsymbol{f}_1)^{-1}(\boldsymbol{U}_1 + \boldsymbol{i}_{\pi(1)}))\,,$$

matches the shifted high level distribution

$$W_1 + \beta + \boldsymbol{a}^\top B\boldsymbol{i}_{\pi(1)}\,.$$

If we reparametrize using $\tau_1(.) = \gamma_1 \circ (\mathbf{id} - \boldsymbol{f}_1)$ we get

$$\gamma_1(\boldsymbol{U}_{\pi(1)} + \boldsymbol{i}_{\pi(1)}) = W_1 + \beta + \boldsymbol{a}^\top B\boldsymbol{i}_{\pi(1)}$$

A Gaussian high-level cause satisfying consistency can thus be achieved by using $\gamma_1 : \boldsymbol{u} \mapsto \beta + \boldsymbol{a}^\top B\boldsymbol{u}$ (because a linear combination of jointly Gaussian random variables is Gaussian, and distribution shifts due to interventions are consistent). Such that we obtain

$$\tau_1 : \boldsymbol{x} \mapsto \beta + \boldsymbol{a}^\top B(\boldsymbol{x}_{\pi(1)} - \boldsymbol{f}_1(\boldsymbol{x}_{\pi(1)}))$$

This is also a sufficient condition to have a valid exact transformation given the Gaussian assumptions.

□

B.2 UNIQUENESS OF EXACT TRANSFORMATION

Proposition 4.1 (Uniqueness)**.** *Assume (i) the low-level model is an additive noise SCM of the form*

$$\begin{bmatrix} \boldsymbol{X}_{\pi(1)}, \\ \boldsymbol{X}_{\pi(0)} \end{bmatrix} := \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{X}_{\pi(1)}) + \boldsymbol{U}_{\pi(1)} + \boldsymbol{i}_{\pi(1)}\,, \\ \boldsymbol{f}_0(\boldsymbol{X}_{\pi(0)}, \boldsymbol{X}_{\pi(1)}) + \boldsymbol{U}_{\pi(0)} \end{bmatrix}\,, \tag{4.3}$$

$$\boldsymbol{U}_{\pi(1)} \sim P_1 \text{ indep. of } \boldsymbol{U}_{\pi(0)} \sim P_0\,, \tag{4.4}$$

(ii) P_1 admits a probability density with non-vanishing Fourier transform, (iii) The high-level model has a non-zero causal effect ($\alpha \neq 0$). Then, if there exists a constructive transformation following Def. 2.3 that is exact for all $\boldsymbol{i} \in \mathbb{R}^{\#\pi(1)}$, it is also unique up to multiplicative and additive constants.

Proof. If there is an exact transformation, marginal probability densities of the target and its corresponding push-forward are matched for all interventions. So the expectations of the target with respect to these densities are matched across interventions, such that

$$\mathbb{E}_{\hat{P}^{(i)}}[Y] = \mathbb{E}_{P^{(\omega(i))}}[Y]$$

which means (using linearity of the high-level cause-effect mechanism)

$$\mathbb{E}_{\hat{P}^{(i)}}[\tau_0(\boldsymbol{X}_0)] = \alpha\mathbb{E}_{P^{(\omega(i))}}[Z_1] + \mathbb{E}[W_0] = \alpha\mathbb{E}_{\hat{P}^{(\boldsymbol{i})}}[Z_1] = \alpha\mathbb{E}_{P^{(\boldsymbol{i})}}[\tau_1(\boldsymbol{X}_1)] + \mathbb{E}[W_0]$$

Assuming there exist a solution achieving consistency for all $\boldsymbol{i}$, it must therefore satisfy

$$\alpha\mathbb{E}_{P^{(\boldsymbol{i})}}[\tau_1(X_1)] = \mathbb{E}_{P^{(\boldsymbol{i})}}[\tau_0(X_0)]\,.$$

Consider an alternative solution

$$\alpha\mathbb{E}_{P^{(\boldsymbol{i})}}[\tilde{\tau}_1(X_1)] = \mathbb{E}_{P^{(\boldsymbol{i})}}[\tau_0(X_0)]\,.$$

Then for all $\boldsymbol{i}_{\pi(1)} \in \mathbb{R}^{\#\pi(1)}$

$$\alpha\mathbb{E}_{P^{(\boldsymbol{i})}}[\tilde{\tau}_1(X_1) - \tau_1(X_1)] = 0\,.$$

Assume a non-trivial model, then $\alpha \neq 0$. Using $g = (\mathbf{id} - \boldsymbol{f}_1)^{-1}$, the mapping from exogenous to endogenous variables (restricted to nodes in $\pi(1)$), we can rewrite the condition as

$$\mathbb{E}_{P_1(U-\boldsymbol{i})}[\tilde{\tau}_1(g(U)) - \tau_1(g(U))] = \int p_1(u-\boldsymbol{i})\delta\tau_1(g(u))du = p_1 * \delta\tau_1 \circ g(\boldsymbol{i}) = 0\,,$$

where "$*$" denotes the convolution operation. Applying the Fourier transform we get

$$\mathcal{F}\left[p_1 * \delta\tau_1 \circ g\right](\xi) = \mathcal{F}[p_1](\xi) \cdot \mathcal{F}[\delta\tau_1 \circ g](\xi) = 0, \text{ for all } \xi \in \mathbb{R}^{\#\pi(1)}$$

we get that if the Fourier transform of p does not vanish, then the Fourier transform of $\delta\tau_1 \circ g$ must vanish everywhere, leading to $\tau_1 \circ g = 0$. Assuming g invertible, we get $\delta\tau_1 = 0$ so the reductions are identical.

□

C ADDITIONAL METHOD DETAILS

C.1 DIFFERENTIABLE NORMALITY REGULARIZATION

In Sec. 4.1, we introduced a normality regularization term that encourages the high-level cause distribution to be Gaussian by measuring the Wasserstein distance between the standardized pushforward distribution $\widehat{P}^{(\boldsymbol{i})}_{\tau,\text{std}}(Z)$ and the standard normal distribution $\mathcal{N}(0,1)$. To implement this regularization in a way that allows end-to-end gradient-based optimization, we need a differentiable approximation of both the empirical cumulative distribution function (CDF) and the distance measure.

Our approach consists of three main steps:

Standardization. Given mini-batches of samples from the pushforward distribution of the high-level cause, we first standardize these samples to have zero mean and unit variance:

$$z_{\text{std}} = \frac{z - \mu_z}{\sigma_z} \tag{C.1}$$

where μ_z and σ_z are the empirical mean and standard deviation of the samples.

Differentiable CDF approximation. For a differentiable approximation of the empirical CDF, we add smooth step functions of the samples:

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} \sigma((x - z_i) \cdot s) \tag{C.2}$$

where $\sigma(t) = \frac{1}{1+e^{-t}}$ is the sigmoid function serving as a smooth version of the step function, z_i are the standardized samples, n is the number of samples, and s is a smoothing factor that controls the sharpness of the transitions in the CDF approximation. Higher values of s create sharper transitions that better approximate the true step function but may lead to sharper gradients.

Wasserstein distance approximation. We approximate the 1-Wasserstein distance between distributions by computing the L_1 distance between their CDFs at a finite set of evaluation points $\{x_1, x_2, \ldots, x_m\}$ uniformly spaced in the interval $[-4, 4]$ (covering most of the probability mass of a standard normal distribution):

$$\hat{W}_1 \approx \frac{1}{m} \sum_{j=1}^{m} |\hat{F}(x_j) - \Phi(x_j)| \tag{C.3}$$

where Φ is the CDF of the standard normal distribution, computed using the error function: $\Phi(x) = \frac{1}{2}(1 + \text{erf}(x/\sqrt{2}))$.

This approximation gives us a fully differentiable loss function that effectively measures how closely the distribution of our high-level cause follows a Gaussian distribution. In practice, we use the same number of evaluation points as we have samples in our mini-batch, and we set the smoothing factor $s = 10.0$, which provides a good balance between approximation accuracy and gradient stability.

D ADDITIONAL THEORETICAL RESULTS

As explained in App. A.3, the consistency loss $\mathcal{L}_{cons}$ used in linear TCR [8] is only a KL divergence between the Gaussian approximation of the push-forward distribution of the low-level model and the (Gaussian) high-level model. Therefore, it does not explicitly enforce a perfect match between distributions, which is the theoretical requirement of an exact transformation according to Eq. (2.3).

In principle, however, the setting of Prop. 4.1 does provide additional constraints that theoretically allow identifiability when $\mathcal{L}_{cons}$ vanishes, as shown below.

Proposition D.1 (Identifiability using D_G)**.** *We assume that the prior $P_{\boldsymbol{I}}$ over interventions has a strictly positive density with respect to the Lebesgue measure. Additionally, we assume the setting of Prop. 4.1: (i) the low-level model is an additive noise SCM of the form*

$$\begin{bmatrix} \boldsymbol{X}_{\pi(1)}, \\ \boldsymbol{X}_{\pi(0)} \end{bmatrix} := \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{X}_{\pi(1)}) + \boldsymbol{U}_{\pi(1)} + \boldsymbol{i}_{\pi(1)}, \\ \boldsymbol{f}_0(\boldsymbol{X}_{\pi(0)}, \boldsymbol{X}_{\pi(1)}) + \boldsymbol{U}_{\pi(0)} \end{bmatrix}, \quad \boldsymbol{U}_{\pi(1)} \sim P_1 \text{ indep. of } \boldsymbol{U}_{\pi(0)} \sim P_0, \tag{D.1}$$

(ii) P_1 admits a probability density with non-vanishing Fourier transform, (iii) The high-level model has a non-zero causal effect ($\alpha \neq 0$). Then, if there exists a constructive transformation that makes $\mathcal{L}_{\text{cons}}$ vanish, it is also unique up to multiplicative and additive constants.

Proof. We notice that the proof of Prop. 4.1 only exploits equality between the expectations of Z or Y obtained under, on the one hand, the high-level model distribution and, on the other hand, the push-forward low-level distribution (for all possible interventions). Since we assume that $\mathcal{L}_{\text{cons}}$ vanishes, then the discrepancy D_G must vanish for all $\boldsymbol{i}$[1] and therefore there is a match between the interventional expectations of $\tau_1(\boldsymbol{X})$ for all $\boldsymbol{i}$, and therefore τ_1 is identified following the same argumentation. □

[1] by continuity in $\boldsymbol{i}$ and because its probability density is strictly positive, if the discrepancy was non-zero at any point it would lead to a strictly positive expectation, thus a strictly positive $\mathcal{L}_{\text{cons}}$.

Potential Limitations to Identifiability based on $\mathcal{L}_{\text{cons}}$. However, the above theoretical identifiability guaranty requires optimizing with respect to an expectation that integrates over all possible values of the intervention vector, which cannot be evaluated in practice, as we necessarily sample from a finite number of interventions. In contrast, linear TCR has theoretical identifiability guaranties under a finite number of interventions [8]. While we could not derive an equivalent result for nonlinear TCR, we conjecture instead a negative result for the loss $\mathcal{L}_{\text{cons}}$ of linear TCR that illustrates the practical difficulties of the optimization of $\mathcal{L}_{\text{cons}}$.

Conjecture D.2. We assume that the prior $P_{\mathbf{I}}$ has a density that vanishes on an open subset of its domain. Additionally, we assume the setting of Prop. 4.1: (i) the low-level model is an additive noise SCM of the form

$$\begin{bmatrix} \boldsymbol{X}_{\pi(1)}, \\ \boldsymbol{X}_{\pi(0)} \end{bmatrix} := \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{X}_{\pi(1)}) + \boldsymbol{U}_{\pi(1)} + \boldsymbol{i}_{\pi(1)}, \\ \boldsymbol{f}_0(\boldsymbol{X}_{\pi(0)}, \boldsymbol{X}_{\pi(1)}) + \boldsymbol{U}_{\pi(0)} \end{bmatrix}, \quad \boldsymbol{U}_{\pi(1)} \sim P_1 \text{ indep. of } \boldsymbol{U}_{\pi(0)} \sim P_0, \tag{D.2}$$

(ii) P_1 admits a probability density with non-vanishing Fourier transform, (iii) The high-level model has a non-zero causal effect ($\alpha \neq 0$). Then, if there exists a constructive transformation that makes $\mathcal{L}_{\text{cons}}$ vanish, there are infinitely many other constructive transformations that also make $\mathcal{L}_{\text{cons}}$ vanish.

Justification for the Conjecture.

Having $\mathcal{L}_{\text{cons}}$ vanish only entails equality between first and second order moments of the low-level and high-level push-forward distributions lead to constraints on the convolutions of τ_1 with the exogenous density p_1. As exemplified in the proof of Prop. 4.1, using $g = (\mathbf{id} - \boldsymbol{f}_1)^{-1}$ leads to conditions of the form

$$\mathbb{E}_{P(U-\boldsymbol{i})}[\tilde{\tau}_1(g(U)) - \tau_1(g(U))] = \int p_1(u - \boldsymbol{i})\delta\tau_1(g(u))du = p_1 * \delta\tau_1 \circ g(\boldsymbol{i}) = 0$$

on the support of $P_{\mathbf{I}}$. So we can choose any non-vanishing function $\delta\tau_1$ such that $p_1 * \delta\tau_1 \circ g$ vanishes outside the support of $P_{\mathbf{I}}$ to obtain a different $\tilde{\tau}_1$ satisfying this constraint. This can be done by choosing a function in the Fourier domain of the form $h(\xi) = \mathcal{F}\{p_1\}^{-1}(\xi)\mathcal{F}\{k\}(\xi)$ with k non-zero but vanishing on the support of $P_{\mathbf{I}}$, such that h is k deconvolved by p_1. Then we obtain that $p_1 * h = k$, such that $\delta\tau = k \circ g^{-1}$. Given the set of such k is infinite, we conjecture that it is possible to find a class of function satisfying both the constraints on first and second order moments at the same time.

Practical Issues with the Optimization of $\mathcal{L}_{\text{cons}}$ and Enforcing Non-Gaussianity. Following the reasoning of the above conjecture, we speculate that optimizing $\mathcal{L}_{\text{cons}}$ with a highly expressive function class may lead to overfitting, in the sense that many highly nonlinear functions can be found that satisfy the first and second order moment constraints of the Gaussian approximation of the KL divergence. Enforcing an exact match between distributions instead of only first and second order moments may help avoid this behavior, *e.g.* by avoiding the nonlinear function to generate multimodal distributions.

While this can be done by adding to $\mathcal{L}_{\text{cons}}$ an addition negentropy term enforcing Gaussianity (see App. A.3), entropy is very challenging to estimate from finite samples and use as a differentiable objective. Instead, we resort to the differentiable measure of non-Gaussianity introduced in Sec. 4.1 based on a Wasserstein distance (see App. C.1) that proved stable in our experiments, and is known to have numerical benefits over the KL divergence.

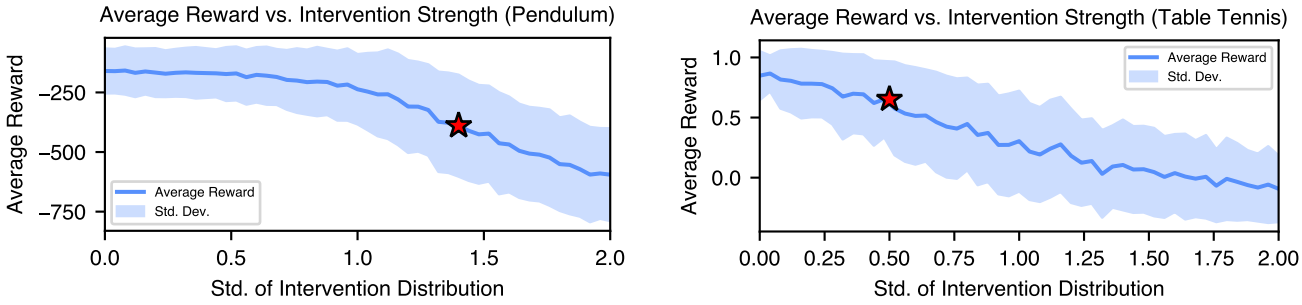
Experiments using this measure as a regularizer are shown in App. F.1 and are compatible with our theoretical insights, in the sense that it improve performance when the sampling of the intervention does not cover a broad enough range of the exogenous distribution.

E EXPERIMENTAL DETAILS

Data. For each experiment we sample N_{int} distinct interventions and for each distinct intervention, we generate N_{ep} RL episodes that apply that intervention. For the synthetic experiments, we equivalently sample N_{ep} SCM samples that apply the same intervention. That is, we keep the intervention fixed but re-sample the exogenous noise. For all datasets, we use 80% for training and 10% for validation and test, respectively.

E.1 SYNTHETIC EXPERIMENTS

Nonlinear Causal Model for Synthetic Data. For our synthetic experiments, we implement a class of nonlinear causal models that follow the structure described in Prop. 4.2. That is, we sample SCMs with the structure given in Eq. (D.2),



(a) **Pendulum Task** (b) **Table Tennis Task**

Figure A: **Relationship Between Intervention Strength and Task Performance.** These plots illustrate how the average episode reward varies as a function of the intervention strength σ (standard deviation of the Gaussian perturbations $\delta A_t \sim \mathcal{N}(0, \sigma^2)$) for both RL tasks. The shaded regions represent ± 1 standard deviation of the reward distribution across episodes. The star markers indicate our selected intervention strengths, which were chosen at the threshold where increased perturbation begins to significantly degrade policy performance. This selection balances between introducing sufficient causal signal for the reduction algorithm and preserving the policy's core behavior patterns.

where, $|\pi(0)| = 1$ and $|\pi(1)| = 9$, , creating a 10-dimensional system (9 low-level cause variables plus 1 target variable). We set $\boldsymbol{h}_0(\boldsymbol{X}_0) \equiv 0$ and $\boldsymbol{a} = 1$. The matrix $B \in \mathbb{R}^{1\times 9}$ has entries sampled from a standard normal distribution, providing the linear mapping from causes to effect as required by Prop. 4.2. The exogenous noise variables $\boldsymbol{U}_{\pi(1)}$ and $\boldsymbol{U}_{\pi(0)}$ are sampled from a normal distribution with zero mean and standard deviation 1.0. The nonlinear function $\boldsymbol{f}_1$ consists of component-wise nonlinear mechanisms represented by randomly initialized two-layer neural networks with 10 hidden units and $\tanh$ activation functions.

Intervention Generation. For each experiment, we generate interventions by sampling from a standard normal distribution, creating shift interventions $\boldsymbol{i}_{\pi(1)} \sim \mathcal{N}(0, \mathbb{1})$. We generate $N_{\text{int}} = 10^7$ distinct interventions, and for each intervention, we sample $N_{\text{ep}} = 1024$ data points from the causal model.

Neural Network Architecture for nTCR. For these synthetic data experiments, the reduction maps τ_1 and ω_1 are parameterized using neural networks with residual connections. Each network consists of a first linear layer that maps the input dimension to a hidden dimension of 256, followed by 8 hidden layers with Softplus activation functions and residual connections. The final layer maps the hidden representation to the high-level cause/intervention. This architecture with residual connections helps in learning complex functions while maintaining gradient flow during training. This amounts to approximately $1.2 \cdot 10^7$ parameters for τ_1, ω_1 and the high-level causal model combined. Unlike our RL experiments, we do not use the Gaussian kernel representation described in Sec. 4.3 since we are only interested in verifying the identification properties of our approach, not interpretability.

nTCR Training. We train our nTCR model using the Adam optimizer with an initial learning rate of $5 \cdot 10^{-4}$ and a cosine annealing learning rate scheduler that gradually reduces the learning rate to zero over the course of training. For the normality regularization, we set $\eta_{\text{norm}} = 1.0$ to encourage the learned high-level cause to follow a Gaussian distribution. We use a batch size of 512, meaning that each training step processes 512 distinct interventions sampled from our prior distribution $P(\boldsymbol{i})$.

Identification Metric. To quantify how well our learned reduction maps align with the ground truth solutions from Prop. 4.2, we employ a normalized, debiased L2 loss. Let $z_i^{\text{learned}} = \tau_1^{\text{learned}}(X_i)$ be the high-level cause computed using our learned reduction, and $z_i^{\text{gt}} = \tau_1^{\text{gt}}(X_i)$ be the corresponding ground truth value for sample i. We first standardize both sets of samples:

$$\tilde{z}_i^{\text{learned}} = \frac{z_i^{\text{learned}} - \overline{z^{\text{learned}}}}{\sqrt{\sum_j (z_j^{\text{learned}} - \overline{z^{\text{learned}}})^2}} \tag{E.1}$$

$$\tilde{z}_i^{\text{gt}} = \frac{z_i^{\text{gt}} - \overline{z^{\text{gt}}}}{\sqrt{\sum_j (z_j^{\text{gt}} - \overline{z^{\text{gt}}})^2}} \tag{E.2}$$

where $\overline{z^{\text{learned}}}$ and $\overline{z^{\text{gt}}}$ are the respective sample means. We then find the optimal rescaling coefficient c that minimizes the L2 distance:

$$c = \arg\min_c \sum_i (c \cdot \tilde{z}_i^{\text{learned}} - \tilde{z}_i^{\text{gt}})^2 = \frac{\sum_i \tilde{z}_i^{\text{learned}} \cdot \tilde{z}_i^{\text{gt}}}{\sum_i (\tilde{z}_i^{\text{learned}})^2} \tag{E.3}$$

The final identification loss is the normalized minimum L2 distance:

$$\mathcal{L}_{\text{id}}(\tau) = \frac{1}{\sqrt{n}} \sqrt{\sum_i (c \cdot \tilde{z}_i^{\text{learned}} - \tilde{z}_i^{\text{gt}})^2} \tag{E.4}$$

where n is the number of samples. The same procedure is applied to evaluate the ω_1 map. This metric accounts for the fact that the theoretical solutions are only identified up to multiplicative and additive constants, while measuring how well the learned reductions capture the structural properties of the ground truth solutions.

Compute Resources. All synthetic experiments were conducted on NVIDIA Quadro RTX 6000 GPUs with 4 CPU cores allocated per run. Each complete training run required approximately 48 hours of computation time. We performed 10 different runs, each with a different randomly generated causal model.

E.2 PENDULUM

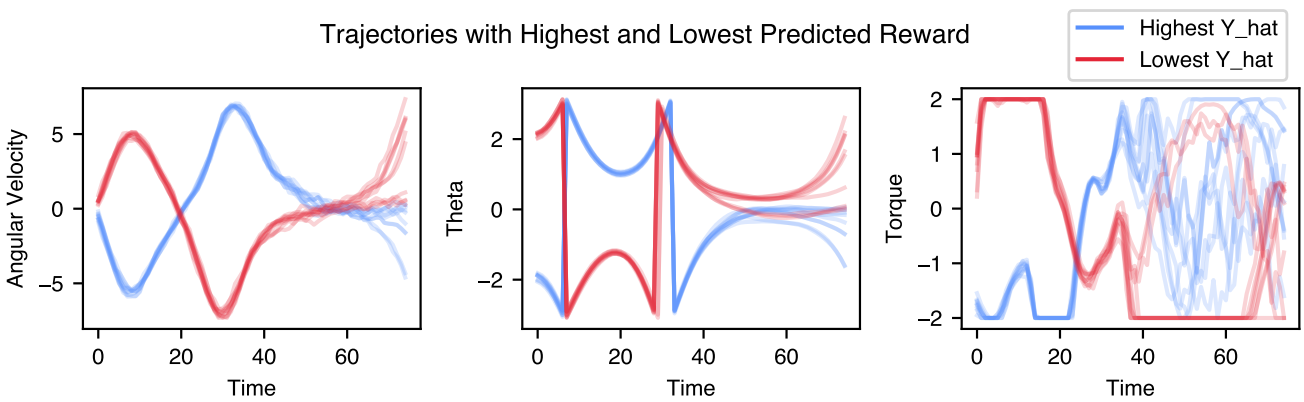


Figure B: **Episodes with Highest and Lowest Predicted Reward.** The plot shows episodes from the test dataset with the highest and lowest cumulative reward predicted by nTCR for Policy A. The red lines display the 10 episodes with highest predicted rewards, while the blue lines show the 10 episodes with lowest predicted rewards.

Each episode in the Pendulum task [19] has a length of $T = 200$ steps. We apply interventions to the actions for the first 75 steps, and define the target variable as the cumulative reward for the remainder of the episode $Y = \sum_{t=76}^{200} R_t$. The reward of the Pendulum task is defined as $R_t = -\theta_{t-1}^2 - 0.1\,\dot{\theta}_{t-1}^2 - 0.001\,A_t^2$, which penalizes deviations from the upright position, large angular velocities, and large torque applied by the policy.

RL Agent Training. We train the policies for Sec. 5.2 with the Stable-Baselines 3 [38] implementation of Proximal Policy Optimization (PPO) [39]. For the Pendulum experiments, we use the default hyperparameters of Stable-Baselines 3. The RL agent collects 1 million environment transitions, and the total wall-clock training time is 50 minutes on an Intel Xeon W-2245 CPU. The training requires less than 1 GB of memory.

Policy Data. For the analysis of the Pendulum policy, we collect a dataset of $N_{\text{int}} = 100{,}000$ interventions with $N_{\text{ep}} = 100$ episodes per intervention. Collecting the datasets of Policy A and B in Sec. 5.2 takes about 150 CPU core-hours each on a compute cluster, with 4GB of memory per core.

nTCR. We train our nTCR model using the Adam optimizer with an initial learning rate of 0.01 and a cosine annealing learning rate scheduler that reduces the learning rate to zero over the course of training. We use a batch size of 64 and train for 90 epochs with weight decay of 0.01. For the normality regularization, we set $\eta_{\text{norm}} = 10$.

For our interpretable nonlinear function class (Sec. 4.3), we use 128 Gaussian kernels for each variable at each time step. Before training, we determine the minimum and maximum values $x^j_{\min}$ and $x^j_{\max}$ for each variable j across all episodes. The kernel centers $\{\mu_{j,t,k}\}_{k=1}^{128}$ are equally spaced over the range $[x^j_{\min}, x^j_{\max}]$, and the kernel widths are set as:

$$\sigma_{j,t} = c \cdot \frac{x^j_{\max} - x^j_{\min}}{128} \tag{E.5}$$

where the constant $c = 8$ acts as a smoothing parameter—higher values lead to smoother learned reduction functions.

For computational convenience, we pre-sample all episodes under interventions and store the data rather than simulating episodes during training, though the latter approach would also be feasible in principle.

Episodes with High and Low Predicted Reward. Fig. B displays episodes from the test set with the highest and lowest predicted cumulative rewards according to the learned nTCR model for Policy A. These episodes correspond to the characteristic trajectory patterns identified in Fig. 3(left): clockwise swinging motions starting from the right quadrant tend to have higher rewards than episodes with counterclockwise motion originating from the left quadrant.

Compute Resources for nTCR. All nTCR training experiments for the Pendulum task were conducted on NVIDIA Quadro RTX 6000 GPUs with 8 CPU cores allocated per run. Each complete training run required approximately 2 hours of computation time and used roughly 4 GB of GPU memory.

E.3 TABLE TENNIS

Table A: Stable-Baselines 3 PPO hyperparameters used for training the table tennis policy.

Parameter name	Value
batch_size	310
clip_range	0.4
clip_range_vf	0.3
ent_coef	7×10^{-6}
gae_lambda	1
gamma	0.9999
learning_rate	0.00015
max_grad_norm	0.1
n_epochs	34
n_steps	5000
num_hidden	380
num_layers	1
vf_coef	0.56

In the simulated table tennis task [20, 21], episodes terminate when the racket hits the ball, which typically happens between steps 90 and 100. The environment then continues simulating the ball to determine its landing position. If the agent does not hit the ball, the episode terminates as soon as the ball falls below the height of the table, which typically happens after 125 to 135 steps. Since our method requires episodes of fixed length, we pad the data to $T = 150$ steps by appending zeros after episode termination.

The agent receives a non-zero reward only at the end of the episode, which depends on whether the agent managed to hit the ball. This reward is given by

$$R_t = \begin{cases} R^{\text{tt}} & \text{racket touches the ball} \\ R^{\text{hit}} & \text{ball below the table} \\ 0 & \text{otherwise}\,, \end{cases} \tag{E.6}$$

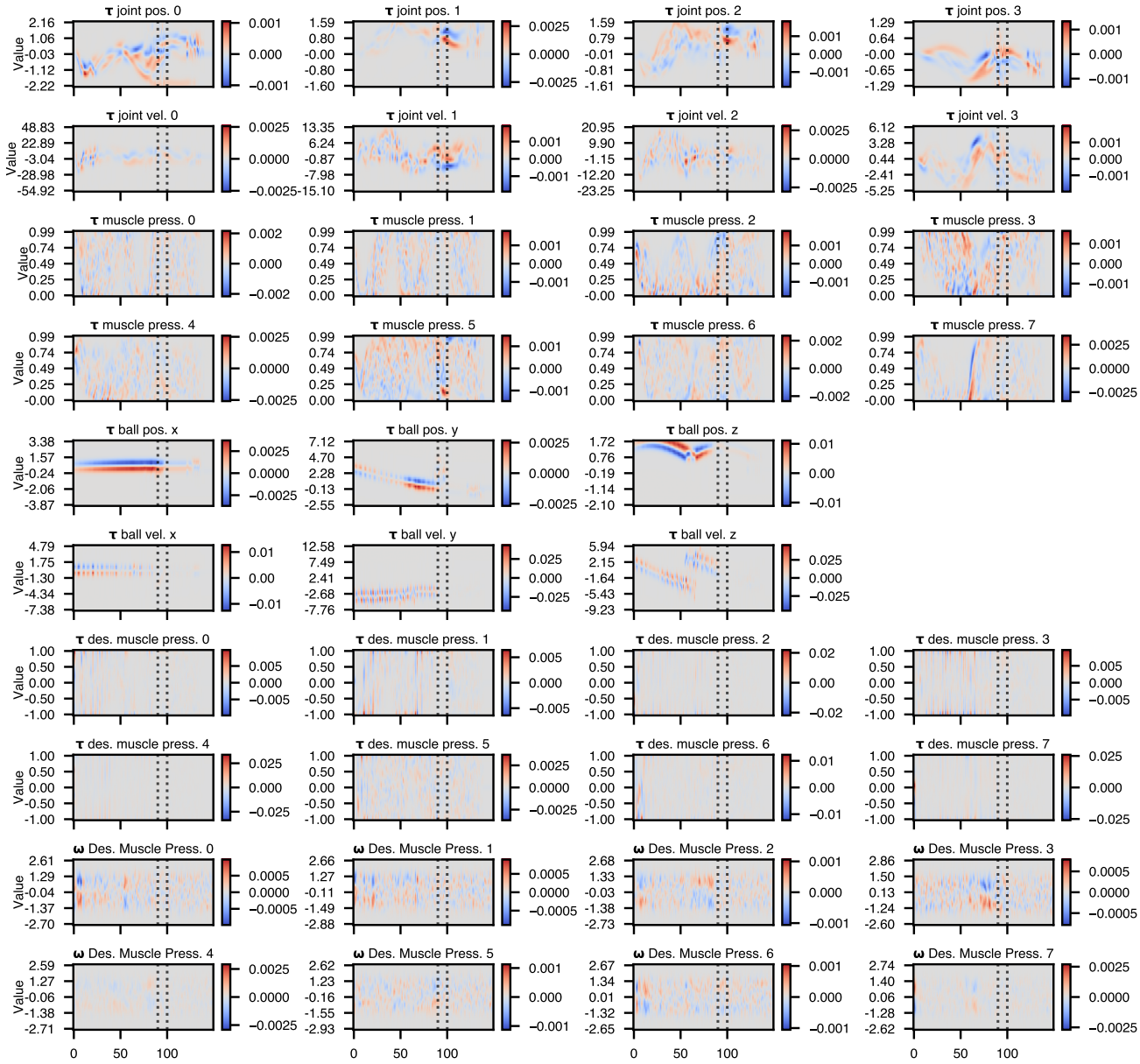


Figure C: **nTCR Reduction Maps for Table Tennis Task.** Linear parameter in high-level model 0.0013 (bias 0.5964).

where R^{tt} rewards the agent for hitting the ball close to the desired position on the table and R^{hit} penalizes missing the ball depending on the minimum distance between the ball and the racket. These reward terms are defined as

$$R^{\text{tt}} = \max\left(1 - \left(\frac{\|\boldsymbol{p}^{\text{land}} - \boldsymbol{p}^{\text{des}}\|}{3}\right)^{\frac{3}{4}}, -0.2\right) \tag{E.7}$$

$$R^{\text{hit}} = -\min_t \|\boldsymbol{p}_t^{\text{ball}} - \boldsymbol{p}_t^{\text{racket}}\|_2, \tag{E.8}$$

where $\boldsymbol{p}^{\text{land}}$ is the landing point of the ball on the table, $\boldsymbol{p}^{\text{des}}$ is the desired landing point at the center of the opponent's side, and $\boldsymbol{p}_t^{\text{ball}}$ and $\boldsymbol{p}_t^{\text{racket}}$ are the 3D positions of ball and racket at timestep t, respectively.

RL Agent Training. Similar to the Pendulum task, we train the table tennis policies in Sec. 5.3 with the Stable-Baselines 3 [38] implementation of PPO [39]. We use the algorithm hyperparameters tuned by Guist et al. [21], which are displayed in Tab. A. The RL agent collects 3 million environment transitions, and the total wall-clock training time is roughly 10 hours on an Intel Xeon W-2245 CPU. Running the simulation and the RL training requires 8 GB of memory.

Policy Data. For the table tennis analysis, collecting the dataset of $N_{\text{int}} = 100{,}000$ interventions with $N_{\text{ep}} = 100$ episodes per intervention takes about 3400 CPU core-hours, with 8GB of memory per core.

nTCR. We use similar nTCR training settings as for the Pendulum task, with the following modifications: learning rate of 0.0001 and weight decay of 0.1 (manually tuned for optimization stability), 32 Gaussian kernels per variable-time pair (chosen to fit GPU memory constraints), and smoothing parameter $c = 1$ (smaller than Pendulum to provide necessary resolution and avoid oversmoothing with fewer kernels). Fig. C shows the full set of nTCR maps for the table tennis task.

Compute Resources for nTCR. All nTCR training experiments for the table tennis task were conducted on NVIDIA A100-SXM4-40GB GPUs with 8 CPU cores allocated per run. Each complete training run required approximately 5 hours of computation time and used around 20 GB of GPU memory.

F ADDITIONAL EXPERIMENTAL RESULTS

F.1 EFFECT OF NORMALITY REGULARIZATION

To better understand the role of our normality regularization introduced in Sec. 4.1, we conduct an ablation study on synthetic data across different noise regimes. We vary the variance of the intervention distribution while keeping the exogenous noise levels in the low-level SCM constant, effectively creating different signal-to-noise ratios.

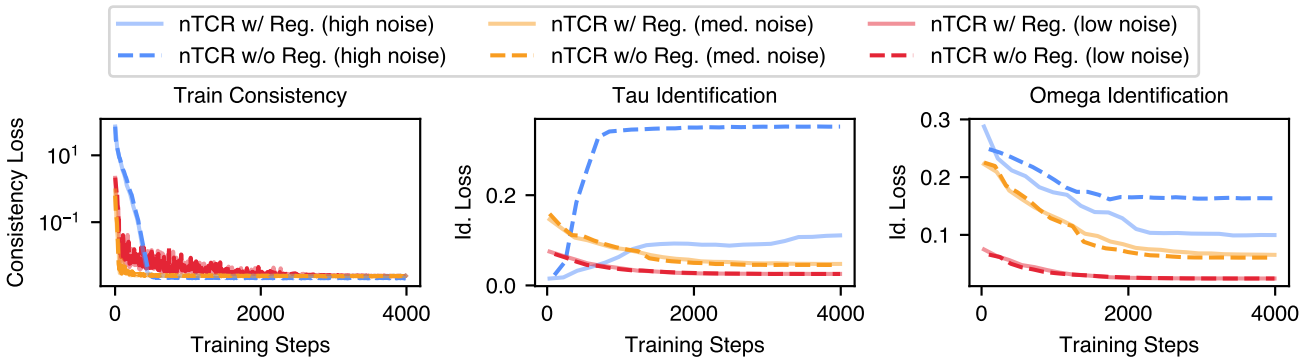


Figure D: **Effect of Normality Regularization on Achieved Consistency and Identification of Ground-Truth Solution.** The figure shows the consistency loss (left) and the identification losses measuring agreement with the ground-truth solutions (definition in App. E.1) for the τ- and ω-functions (middle and right) over the reduction training run for synthetic low-level SCMs with 3 low-level variables. The interventions are sampled as $\boldsymbol{i}_{\pi(1)} \sim \mathcal{N}(0, h \cdot \mathbb{1})$, with $h \in \{0.1, 1.0, 10.0\}$ for the high-, medium- and low-exogenous-noise settings, respectively. Note that, since we keep the exogenous noise levels in the low-level SCM constant, a low variance of the intervention distribution corresponds to high exogenous noise relative to the interventions. The lines for normality regularization have $\eta_{\text{norm}} = 1$; to turn off regularization we set $\eta_{\text{norm}} = 0$. We show average values over 10 sampled SCMs.

Fig. D shows several important insights about the effect of normality regularization across different noise regimes. First, the identifiability metrics demonstrate that learning ground truth reduction maps becomes progressively more challenging as we move from the low-noise setting (high intervention variance) to the high-noise setting (low intervention variance). This reflects the fundamental signal-to-noise ratio: when interventions have small variance relative to the exogenous noise, the signal about the most influential factors for the target becomes harder to detect and extract.

Most significantly, we observe that normality regularization has the strongest beneficial effect in the high-noise regime, where it substantially improves the identification of ground truth solutions for both τ and ω maps. In contrast, for the low- and medium-noise settings, the regularization provides minimal additional benefit, suggesting that when the signal-to-noise ratio is favorable, the consistency loss alone is sufficient to guide the learning toward the correct solution. To understand why normality regularization is particularly crucial in the high-noise regime, Fig. E shows the empirical CDF of the learned

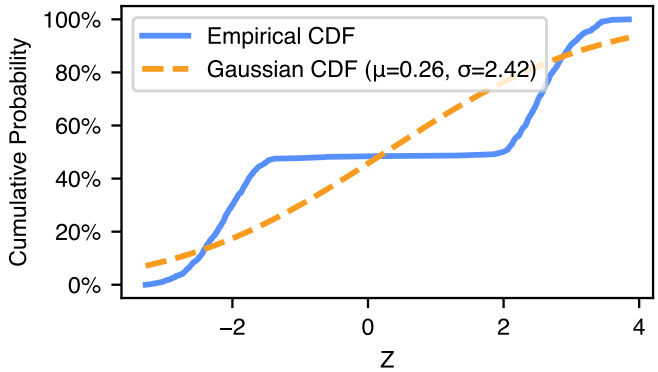
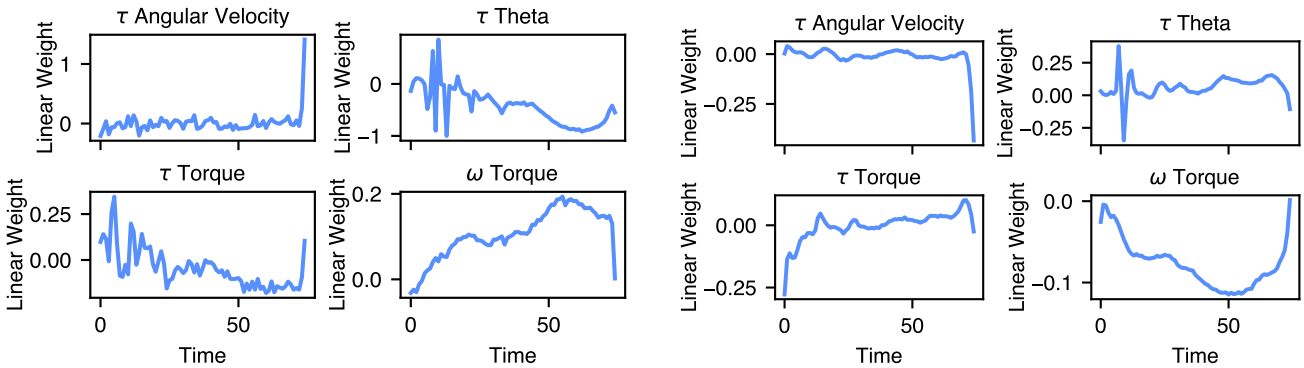


Figure E: **Non-Gaussianity of the high-level cause.** The figure shows the empirical CDF (blue) for the high-level cause for an SCM sampled in Fig. E (high noise setting) for an nTCR with $\eta_{\text{norm}} = 0$. The yellow dashed line shows the CDF for a Gaussian distribtuion with the same mean and variance.

high-level cause distribution in the high-noise setting without normality regularization. The distribution shows highly non-Gaussian behavior with multiple peaks, indicating that the τ reduction is exploiting the flexibility of the nonlinear function class to minimize the consistency loss by artificially shaping the high-level pushforward distribution. Such multi-modal distributions are undesirable for interpretability, as they introduce additional complexity that obscures the underlying causal structure (see also our discussion of Gaussianity in App. D).

These results demonstrate that normality regularization serves as an important inductive bias that prevents overfitting to the training data while maintaining the interpretability of the learned high-level causes. The regularization ensures that interventions must be sufficiently strong relative to the exogenous noise to be effectively learneda principle that aligns with the fundamental requirement that causal interventions should produce detectable changes in the target phenomenon (see also Fig. A).

F.2 LINEAR TCR FOR PENDULUM POLICIES



(a) **Policy A** (b) **Policy B**

Figure F: **Linear TCR [8] for Pendulum Task.** Linear parameters in high-level model: 5.6821 (bias −28.5229) for Policy A and 94.0380 (bias −115.0152) for Policy B.

The reduction maps for linear TCR [8] for policies A and B are shown in Figs. Fa and Fb, respectively. These provide a baseline for comparison with our nonlinear approach (nTCR).

Policy A: Limitations of Linear TCR. For Policy A, linear TCR identifies predominantly negative weights in the second half of the episode for the τ-maps for Theta and Torque. While this correctly captures that deviations from the upright position (nonzero Theta) and large corrective actions (high Torque) late in episodes correlate with lower rewards, the linear approach fails to capture the more nuanced pattern revealed by nTCR. For the first half of the episode, the linear weights show inconsistent, unstable patterns without a clear interpretable structure. The τ-map for Angular Velocity appears largely

neutral except for an unexplained positive peak at the final time step.

In contrast, nTCR (as shown in Fig. 3, left) clearly distinguishes between two distinct trajectory classes: clockwise swinging (from the right quadrant) versus counterclockwise swinging (from the left quadrant). This critical asymmetry in policy performance is entirely missed by the linear model, which can only capture monotonic relationships between state variables and expected reward.

Policy B: Observations Consistent with nTCR. For Policy B, linear TCR shows a strongly negative τ-map for Torque at the start of episodes and a relatively neutral pattern in the second half. This suggests episodes starting with positive torque tend to have worse outcomes. The ω-map indicates shifting torque toward more negative values would be beneficial, which aligns with nTCR's findings. These two observations are consistent with Policy B's failure mode—allowing the pendulum to tip over in the positive θ direction before applying corrective torque—can be captured by linear relationships. In this case, both linear TCR and nTCR correctly identify the key intervention (applying more negative torque) that would improve policy performance.

Comparative Advantages of nTCR. Overall, we observe that linear TCR has fundamental limitations in capturing complex behavioral patterns in RL policies:

- **Non-monotonic relationships:** Linear TCR cannot capture U-shaped or other non-monotonic relationships between states and expected rewards.
- **Trajectory classes:** Linear TCR fails to distinguish between qualitatively different trajectory classes (like clockwise vs. counterclockwise motion) that are not linearly separable.

These results empirically validate the need for our nonlinear extension to TCR, demonstrating that nTCR can uncover important qualitative patterns in policy behavior that remain hidden to linear approaches.

F.3 LINEAR TCR FOR TABLE TENNIS POLICIES

The reduction maps for linear TCR [8] for the table tennis task are shown in Fig. G. While linear TCR provides a computationally simpler baseline, it reveals fundamental limitations when applied to complex robotic control tasks.

Temporal Importance Identification. Linear TCR successfully identifies that the most crucial periods for policy success or failure occur just before and during ball contact. This can be observed from the largest magnitude contributions in the linear reduction maps, which appear predominantly around the time window when the robot typically hits the ball (indicated by the dotted vertical lines). This temporal insight aligns with the physical intuition that precise timing and positioning during ball contact are critical for successful returns.

Limited State Representation. However, the resolution of linear TCR is fundamentally constrained: it can only indicate whether a particular variable at a specific time contributes positively or negatively to the expected reward *on average* over the entire state and action distribution observed in the dataset. Linear reductions cannot distinguish between different values that a variable might take at any given time step. For instance, while nTCR can identify that balls bouncing closer to the net present greater challenges for the robot (Fig. 4(h,i)), it would be hard to deduce such value-dependent relationships from linear TCR. The linear approach averages over all ball positions at each time step, obscuring the specific spatial patterns that influence task difficulty.

Expressivity vs. Simplicity Trade-off. While linear TCR offers benefits in terms of training simplicity and straightforward interpretation of the learned reductions, its limited capacity to represent state-dependent relationships severely constrains its applicability to complex control scenarios. The table tennis task exemplifies this limitation: successful policy explanation requires understanding not just *when* certain variables matter, but also *which specific values* of those variables lead to success or failure. This distinction between temporal importance and state-value dependencies shows the necessity of our nonlinear extension for capturing meaningful causal patterns in more complex tasks.

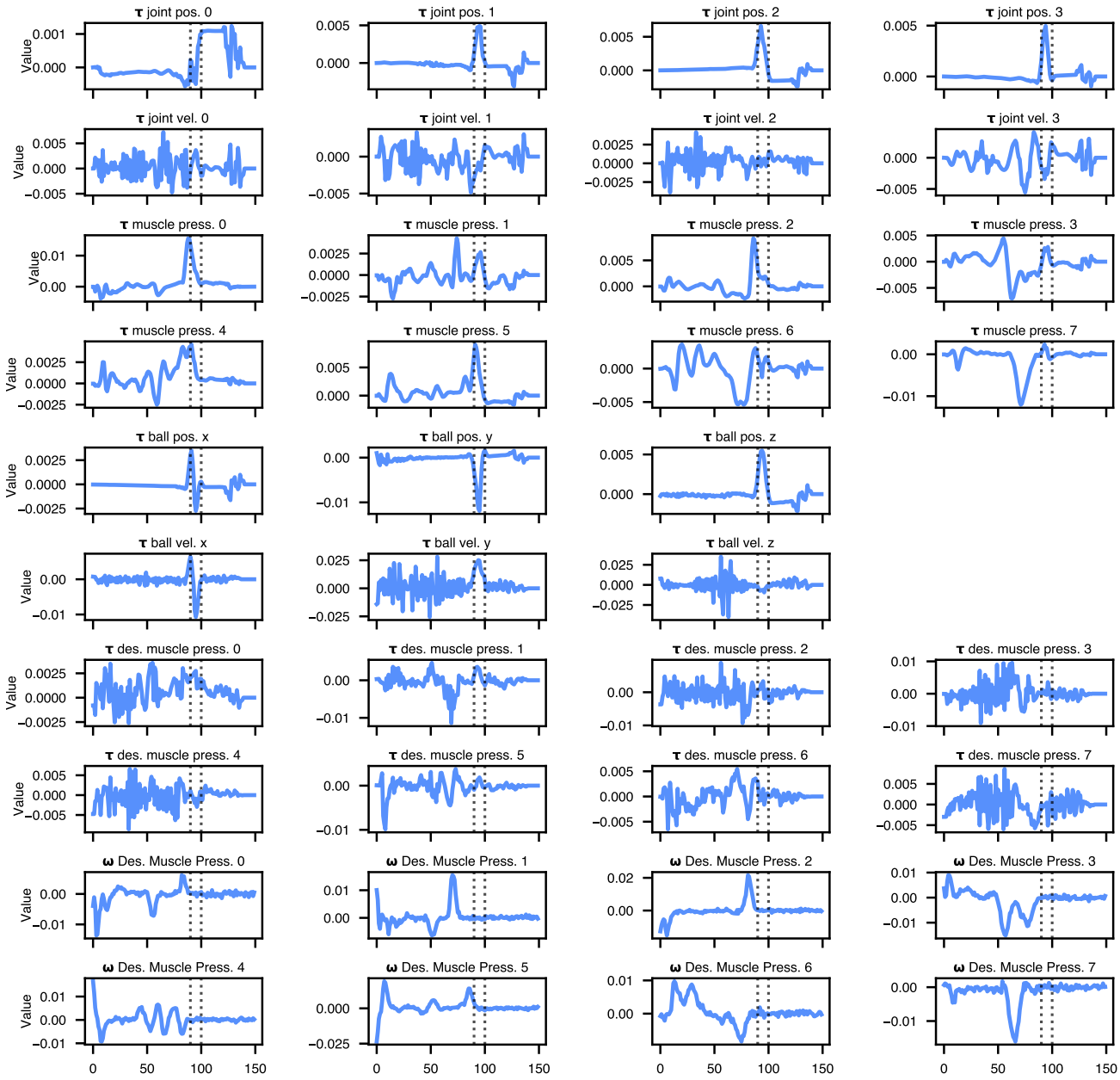


Figure G: **Linear Reduction Maps for Table Tennis Task.** Linear parameter in high-level model 0.6933 (bias 0.6780).