Partial Transportability for Domain Generalization

Kasra Jalaldoust* **Alexis Bellot*†** **Elias Bareinboim**
Causal Artificial Intelligence Lab
Columbia University
{kasra, eb}@cs.columbia.edu, abellot95@gmail.com

Abstract

A fundamental task in AI is providing performance guarantees for predictions made in unseen domains. In practice, there can be substantial uncertainty about the distribution of new data, and corresponding variability in the performance of existing predictors. Building on the theory of partial identification and transportability, this paper introduces new results for bounding the value of a functional of the target distribution, such as the generalization error of a classifier, given data from source domains and assumptions about the data generating mechanisms, encoded in causal diagrams. Our contribution is to provide the first general estimation technique for transportability problems, adapting existing parameterization schemes such Neural Causal Models to encode the structural constraints necessary for cross-population inference. We demonstrate the expressiveness and consistency of this procedure and further propose a gradient-based optimization scheme for making scalable inferences in practice. Our results are corroborated with experiments.

1 Introduction

In the empirical sciences, the value of scientific theories arguably depends on their ability to make predictions in a domain different from where the theory was initially learned. Understanding when and how a conclusion in one domain, such as a statistical association, can be generalized to a novel, unseen domain has taken a fundamental role in the philosophy of biological and social sciences in the early 21st century. As Campbell and Stanley [8, p. 17] observed in an early discussion on the interpretation of statistical inferences, "Generalization always turns out to involve generalization into a realm not represented in one's sample" where, in particular, statistical associations and distributions might differ, presenting a fundamental challenge.

As society transitions to become more AI centric, many of the every-day tasks based on predictions are increasingly delegated to automated systems. Such developments make various parts of society more efficient, but also require a notion of performance guarantee that is critical for the safety of AI, in which the problem of generalization appears under different forms. For instance, one critical task in the field is domain generalization, where one tries to learn a model (e.g. classifier, regressor) on data sampled from a distribution that differs in several aspects from that expected when deploying the model in practice. In this context, generalization guarantees must build on knowledge or assumptions on the "relatedness" of different training and testing domains; for instance, if training and testing domains are arbitrarily different, no generalization guarantees can be expected from any predictor [12, 40]. The question becomes how to link the domains of data that are used to train a model (a.k.a., the source domains) to the domain where this model is deployed in practice (a.k.a., the target domain).

*Equal Contribution.
†Now at Google DeepMind.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-4376

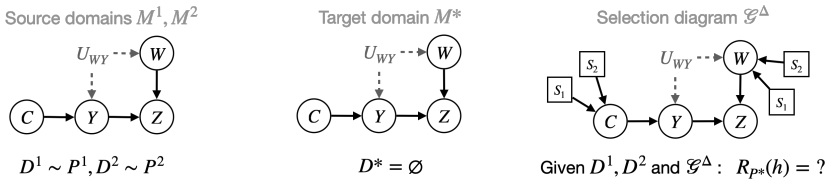


Figure 1: Illustration of the task of evaluating the generalization error of a model h. The mechanisms for C and W vary across domains.

To begin to answer this question, a popular type of assumption that relates source and target domains is *statistical* in nature: invariances in the marginal or conditional distribution of some variables across the source and target distributions. Examples include assumptions of covariate shift and label shift (among others) [35, 34]. Notably, generalization is justified by the stability and invariance of the causal mechanisms shared across the domains [14, 21], since the distributional/statistical invariances across the domains are consequences of mechanistic/structural invariances governing the underlying data generating process. Although the induced statistical invariances, once exploited correctly, can be used as bases for generalizability. Broadly, invariance-based approaches to domain generalization [27, 29, 2, 40, 24, 20, 7, 6, 13] search for predictors that not only achieves small error on the source data but also maintain certain notions of distributional invariance across the source domains. Since these statistical invariances can be viewed as proxies to structural invariances, in certain instances generalization guarantees can be provided through causal reasoning [17, 31, 39]. This idea can be illustrated in Fig. 1. The value of variables $\{C, Y, W, Z\}$ are determined as a stochastic function of variables pointing to it, while these functions may differ across domains. The challenge is to evaluate the generalization risk of a model, e.g. $R_{P*}(h) := \mathbb{E}_{P*}[(Y - h)^2]$ for $h := h(C, W, Z) = \mathbb{E}_{P^1}[Y \mid C, W, Z]$, without observations from the target P^*. General instances of this challenge have been studied under the rubric of the theory of causal transportability, where qualitative assumptions regarding the underlying structural causal models are encoded in a graphical object, and algorithms are designed to leverage these assumptions and compute certain statistical queries in the target domain in terms of the existing source data [26, 4, 5, 19, 11, 17].

Despite these advances, in practice, the combination of source data and graphical assumptions is not always sufficient to identify (uniquely evaluate) the desired statistical query, e.g., the average loss of a given predictor in the target domain. In this case, the query is said to be non-transportable[3]. For example, given Fig. 1, $R_{P*}(h)$ is non-transportable for the classifier $h := h(C, W, Z)$. In this paper, we study the fundamental task of computing tight upper-bounds for statistical queries in a new unseen domain. This allows us to assess worst-case performance of prediction models for the domain generalization task. Our contributions are as follows:

- **Sections 2 & 3.** We develop the first general estimation technique for bounding the value of queries across multiple domains (e.g., the generalization risk) in non-transportable settings (Def. 4). Specifically, we extend the formulation of canonical models [3, 42] to encode the constraints necessary for solving the transportability task, and demonstrate their expressiveness for generating distributions entailed by the underlying Structural Causal Models (SCMs) (Thm. 1).
- **Section 4.** We adapt Neural Causal Models (NCMs) [41] for the transportability task via a parameter sharing scheme (Thm. 2), similarly demonstrating their expressiveness and consistency for solving the partial transportability task. We then leverage the theoretical findings in sections 2 & 3 to implement a gradient-based optimization algorithm for making scalable inferences (Alg. 1), as well as a Bayesian inference procedure. Finally, we introduce Causal Robust Optimization (CRO) (Alg. 2), an iterative method to find a predictor with the best worst-case risk.

Preliminaries. We use capital letters to denote variables (X), small letters for their values (x), bold letters for sets of variables ($\boldsymbol{X}$) and their values ($\boldsymbol{x}$), and use supp to denote their domains of definition ($x \in \mathrm{supp}_X$). A conditional independence statement in distribution P is written as $(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \mathbf{Z})_P$. A d-separation statement in some graph $\mathcal{G}$ is written as $(\boldsymbol{X} \perp\!\!\!\perp_d \boldsymbol{Y} \mid \boldsymbol{Z})$. To denote $P(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{X} = \boldsymbol{x})$, we use the shorthand $P(\boldsymbol{y} \mid \boldsymbol{x})$. The basic semantic framework of our analysis relies on Structural Causal Models (SCMs) [25, Definition 7.1.1], which are defined below.

[3]The notion of non-transportability formalizes a type of aleatoric uncertainty [16] arising from the inherent variability within compatible data generating systems for the target domain. In particular, it cannot be explained away with increasing sample size from the source domains.

Definition 1. *An SCM M is a tuple $M = \langle \boldsymbol{V}, \boldsymbol{U}, \mathcal{F}, P \rangle$ where each observed variable $V \in \boldsymbol{V}$ is a deterministic function of a subset of variables $\boldsymbol{Pa}_V \subset \boldsymbol{V}$ and latent variables $\boldsymbol{U}_V \subset \boldsymbol{U}$, i.e., $v := f_V(\boldsymbol{pa}_V, \boldsymbol{u}_V)$, $f_V \in \mathcal{F}$. Each latent variable $U \in \boldsymbol{U}$ is distributed according to a probability measure $P(u)$. We assume the model to be recursive, i.e. that there are no cyclic dependencies among the variables.* □

SCM M entails a probability distribution $P^{\mathcal{M}}(\boldsymbol{v})$ over the set of observed variables $\boldsymbol{V}$ such that

$$P^{\mathcal{M}}(\boldsymbol{v}) = \int_{\text{supp}_{\boldsymbol{U}}} \prod_{V \in \boldsymbol{V}} P^{\mathcal{M}}(v \mid pa_V, \boldsymbol{u}_V) \cdot P(\boldsymbol{u}) \cdot d\boldsymbol{u}, \tag{1}$$

where the term $P(v \mid pa_V, \boldsymbol{u}_V)$ corresponds to the function $f_V \in \mathcal{F}$ in the underlying structural causal model M. It also induces a causal diagram $\mathcal{G}_{\mathcal{M}}$ in which each $V \in \boldsymbol{V}$ is associated with a vertex, and we draw a directed edge between two variables $V_i \to V_j$ if V_i appears as an argument of f_{V_j} in the SCM, and a bi-directed edge $V_i \leftrightarrow V_j$ if $\boldsymbol{U}_{V_i} \cap \boldsymbol{U}_{V_j} \neq \varnothing$, that is V_i and V_j share an unobserved confounder. Throughout this paper, we assume the observational distributions entailed by the SCMs satisfy the positivity assumption, that is, $P^M(\boldsymbol{v}) > 0$, for every $\boldsymbol{v}$. We will also operate non-parametrically, i.e., making no assumption about the particular functional form or the distribution of the unobserved variables.

2 Risk Evaluation through Partial Transportability

In this section, we focus on challenges of the domain generalization problem through a causal lens, in particular regarding assessment of average loss of a given classifier in the target domain. We study a system of variables $\boldsymbol{V}$ where $Y \in \boldsymbol{V}$ is a categorical outcome variable and consider a classifier $h : \text{supp}_{\boldsymbol{X}} \to \text{supp}_Y$ mapping a set of covariates $\boldsymbol{X} \subset \boldsymbol{V}$ to the domain of the outcome. The setting of domain generalization is characterized by multiple domains, defined by SCMs $\mathbb{M} : \{\mathcal{M}^1, \ldots, \mathcal{M}^K, \mathcal{M}^*\}$ that entail distributions $\mathbb{P} = \{P^{\mathcal{M}^1}, \ldots, P^{\mathcal{M}^K}\}$ and $P^{\mathcal{M}^*}$ over $\boldsymbol{V}$. We are given a classifier $h : \text{supp}_{\boldsymbol{X}} \to \text{supp}_Y$, and the objective is to evaluate its risk in the domain $\mathcal{M}^*$ which is defined as,

$$R_{P^*}(h) := \mathbb{E}_{P^*}[\mathcal{L}(Y, h(\boldsymbol{X}))], \tag{2}$$

where $\mathcal{L} : \text{supp}_Y \times \text{supp}_Y \to \mathbb{R}^+$ is a loss function. The following example illustrates these notions.

Example 1 (Covariate shift)**.** A common instance of the domain generalization problem considers source and target domains $\mathbb{M} : \{M^1, M^*\}$ over $\boldsymbol{V} = \{\boldsymbol{X}, Y\}$ and $\boldsymbol{U} = \{U_{\boldsymbol{X}}, U_Y\}$ defined by

$$\mathcal{M}^1 : \begin{cases} \mathcal{F}^1 : \begin{cases} \boldsymbol{X} \leftarrow f^1_{\boldsymbol{X}}(U_{\boldsymbol{X}}) \\ Y \leftarrow f_Y(\boldsymbol{X}, U_Y) \end{cases} \\ P^1(\boldsymbol{U}) = P^1(U_{\boldsymbol{X}}) \cdot P(U_Y) \end{cases} \qquad \mathcal{M}^* : \begin{cases} \mathcal{F}^* : \begin{cases} \boldsymbol{X} \leftarrow f^*_{\boldsymbol{X}}(U_{\boldsymbol{X}}) \\ Y \leftarrow f_Y(\boldsymbol{X}, U_Y) \end{cases} \\ P^*(\boldsymbol{U}) = P^*(U_{\boldsymbol{X}}) \cdot P(U_Y) \end{cases}$$

Here, because $P^1(U_{\boldsymbol{X}}) \neq P^1(U_{\boldsymbol{X}})$, this implies via Eq. 1 that the covariate distributions are different, i.e., $P^1(\boldsymbol{X}) \neq P^*(\boldsymbol{X})$. Still, the label distribution conditional on covariates is invariant, i.e., $P^1(Y \mid \boldsymbol{X}) = P^*(Y \mid \boldsymbol{X})$, also known as the covariate shift setting. Accordingly, the risk of a classifier $h := h(\boldsymbol{x})$ can be written as,

$$R_{P^*}(h) = \int_{\text{supp}_Y \times \text{supp}_{\boldsymbol{X}}} \mathcal{L}(y, h(\boldsymbol{x})) P^*(y, \boldsymbol{x}) \cdot dy d\boldsymbol{x} = \int_{\text{supp}_Y \times \text{supp}_{\boldsymbol{X}}} \mathcal{L}(y, h(\boldsymbol{x})) P^1(y \mid \boldsymbol{x}) P^*(\boldsymbol{x}) \cdot dy d\boldsymbol{x}. \tag{3}$$

We will consider the problem of quantifying the variation in $R_{P^*}(h)$ subject to variation in $P^*(\boldsymbol{x})$ that would be consistent with partial observations from these domains, e.g. samples from $P^1(\boldsymbol{X}, Y)$, and assumptions about the commonalities and discrepancies across the domains. □

To describe more general discrepancies in the mechanisms between the SCMs, we adapt the notion of domain discrepancy and selection diagram introduced in [19].

Definition 2 (Domain discrepancy)**.** For SCMs $\mathcal{M}^i, \mathcal{M}^j$ ($i, j \in \{*, 1, 2, \ldots, K\}$) defined over $\boldsymbol{V}$, the domain discrepancy set $\Delta_{ij} \subseteq \boldsymbol{V}$ is defined such that for every $V \in \Delta_{ij}$ there might exist a discrepancy $f^{\mathcal{M}^i}_V \neq f^{\mathcal{M}^j}_V$, or $P^{\mathcal{M}^i}(\boldsymbol{u}_V) \neq P^{\mathcal{M}^j}(\boldsymbol{u}_V)$. For abbreviation, we denote Δ_{i*} as Δ_i. □

In words, if an endogenous variable V is not in Δ_{ij}, this means that the mechanism for V (i.e., the function f_V and the distribution of exogenous variables $P(\boldsymbol{u}_V)$) are structurally invariant across $\mathcal{M}^i, \mathcal{M}^j$. What follows integrates the domain discrepancy sets into a generalization of causal diagrams to express qualitative assumptions about multiple SCMs [26, 11].

https://doi.org/10.52202/079017-4376

Definition 3 (Selection diagram)**.** The selection diagram $\mathcal{G}^{\Delta_i}$ is constructed from $\mathcal{G}^i$ ($i \in \{1, 2, \ldots, T\}$) by adding the selection node S_i to the vertex set, and adding the edge $S_i \to V$ for every $V \in \Delta_i$. The collection $\mathcal{G}^{\boldsymbol{\Delta}} = \{\mathcal{G}^*\} \cup \{\mathcal{G}^{\Delta_i}\}_{i \in \{1,2,\ldots,T\}}$ encodes the graphical assumptions. Whenever the causal diagram is shared across the domains, a single diagram can depict $\mathcal{G}^{\boldsymbol{\Delta}}$. □

Selection diagrams extend causal diagrams and provide a parsimonious graphical representation of the commonalities and disparities across a collection of SCMs. The following example illustrates these notions and highlights various subtleties in the generalization error of different predictors.

Example 2 (Generalization performance of classifiers)**.** Consider the SCMs $\mathcal{M}^i$ ($i \in \{1, 2, *\}$) over the binary variables $\boldsymbol{X} = \{C_1, C_2, \ldots, C_{10}\} \cup \{W, Z\}$ and Y, defined as follows:

$$P^i(\boldsymbol{U}): \begin{cases} \forall 1 \leqslant j \leqslant 10 : U_{C_j} \sim \text{Bern}(0.1) \text{ if } i = 1 \text{ Bern}(0.5) \text{ if } i = 2 \text{ Bern}(0.7) \text{ if } i = * \\ U_{YW} \sim \text{Bern}(0.2) \\ U_W \sim \text{Bern}(0.01) \text{ if } i = 1 \text{ Bern}(0.02) \text{ if } i = 2 \text{ Bern}(0.5) \text{ if } i = * \\ U_Z \sim \text{Bern}(0.9) \end{cases}$$

$$\mathcal{F}^i : \boldsymbol{C} \leftarrow \boldsymbol{U_C},\ Y \leftarrow U_{YW} \oplus \bigoplus_{C \in \boldsymbol{C}} C,\ W \leftarrow U_{YW} \oplus U_W,\ Z \leftarrow Y \cdot U_Z + W \cdot (1 - U_Z)$$

$\bigoplus$ denotes the xor operator, *i.e.*, $A \bigoplus B$ evaluates to 1 if $A \neq B$ and evaluates to 0 if $A = B$. Notice that the distribution of exogenous noise associated with $\boldsymbol{C}_{1:10}$ and $\{W\}$ differs across the domains. Consider three baseline classifiers $h_1(\boldsymbol{c}, w) := w \oplus \bigoplus_{c \in \boldsymbol{c}} c, h_2(\boldsymbol{c}) := \bigoplus_{c \in \boldsymbol{c}} c, h_3(z) := z$ evaluated on data from P^1, P^2, P^* with the symmetric loss function $\mathcal{L}(Y, h(\boldsymbol{X})) = \mathbb{1}\{Y \neq h(\boldsymbol{X})\}$. Their errors are given in Table 1. Notice that h_1 has almost perfect accuracy on both source distributions, but does not generalize to $\mathcal{M}^1$ as it uses the unstable feature W, incurring 50% loss. This observation indicates that mere minimization of the empirical risk might yield arbitrarily large risk in the unseen target domain. h_2 uses the features $\boldsymbol{C}$ that are the direct causes of Y, also known as the causal predictor [27, 2], and yields a stable loss of 20% across all domains. On the other hand, h_3 uses only Z that is a descendant of Y, yet achieves a small loss across all domains as the mechanism of Z is assumed to be invariant. This observation is surprising, because h_3 is neither a causal predictor nor the minimizer of the empirical risk, yet it performs nearly optimally on all domains. □

Example 2 illustrates potential challenges of the domain generalization problem, particularly regarding the variation of the risk of classifiers across the source and target domains. The following definition introduces the problem of "partial transportability" which is the main conceptual contribution of our paper. The objective is bounding a statistic of the target distribution using the data and assumptions available about related domains.

Classifier	$R_{P^{\mathcal{M}^1}}$	$R_{P^{\mathcal{M}^2}}$	$R_{P^{\mathcal{M}*}}$
$h_1(\boldsymbol{c}, w)$	1%	4%	49%
$h_2(\boldsymbol{c})$	20%	20%	20%
$h_3(z)$	3%	5%	4%

Table 1: Classifiers in Example 2.

Definition 4 (Partial Transportability)**.** Consider a system of SCMs $\mathbb{M} : \{\mathcal{M}^1, \mathcal{M}^2, \ldots, \mathcal{M}^K, \mathcal{M}^*\}$ that induces the selection diagram $\mathcal{G}^{\boldsymbol{\Delta}}$ over the variables $\boldsymbol{V}$ and entails the distributions $\mathbb{P} : \{P^1(\boldsymbol{v}), P^2(\boldsymbol{v}), \ldots, P^K(\boldsymbol{v})\}$ and $P^*(\boldsymbol{v})$. A functional $\psi : \Omega_{\boldsymbol{V}} \to \mathbb{R}$ is partially transportable from $\mathbb{P}$ given $\mathcal{G}^{\boldsymbol{\Delta}}$ if,

$$\mathbb{E}_{P^{\mathcal{M}^*_0}}[\psi(\boldsymbol{V})] \leqslant q_{\max}, \forall \text{ SCMs } \mathbb{M}_0 \text{ that entail } \mathbb{P} \text{ and induce } \mathcal{G}^{\boldsymbol{\Delta}}, \tag{4}$$

where $q_{\max} \in \mathbb{R}$ is a constant that can be obtained from $\mathbb{P}$ given $\mathcal{G}^{\boldsymbol{\Delta}}$. □

For instance, finding the worst-case performance of a classifier based on the source distributions given the selection diagram is a special case of partial transportability with $\psi(\boldsymbol{x}, y) := \mathcal{L}(y, h(\boldsymbol{x}))$. In principle, this task is challenging as the exogenous distribution $P^*(\boldsymbol{U}_V)$ and structural assignments f^*_V of variables $V \in \boldsymbol{V}$ that do not match with any of the source domains could be arbitrary. In the following section, we will define tractable parameterization of $\{P(\boldsymbol{U}), \mathcal{F}\}$ to derive a systematic approach to solving partial transportability tasks.

3 Canonical Models for Partial Transportability

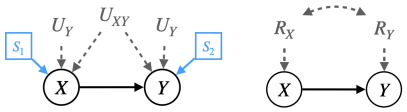
We begin with an example to illustrate how one might approach parameterizing a query such as $\mathbb{E}_{P^{\mathcal{M}*}}[\psi(\boldsymbol{V})]$, *e.g.*, the generalization error, to consistently solve the partial transportability task.

Example 3 (The bow model)**.** Let $\boldsymbol{X} := \{X\}$ be a single binary variable, and Y be a binary label. Consider two source domains defined by the following SCMs:

$$\mathcal{M}^1 : \begin{cases} P^1(\boldsymbol{U}) : \begin{cases} U_X \sim \text{Bern}(0.2) \\ U_Y \sim \text{Bern}(0.05) \\ U_{XY} \sim \text{Bern}(0.95) \end{cases} \\ \mathcal{F}^1 : \begin{cases} X \leftarrow U_X \oplus U_{XY} \\ Y \leftarrow (X \oplus U_{XY}) \oplus U_Y \end{cases} \end{cases} \qquad \mathcal{M}^2 : \begin{cases} P^2(\boldsymbol{U}) : \begin{cases} U_X \sim \text{Bern}(0.9) \\ U_Y \sim \text{Bern}(0.05) \\ U_{XY} \sim \text{Bern}(0.95) \end{cases} \\ \mathcal{F}^2 : \begin{cases} X \leftarrow U_X \oplus U_{XY} \\ Y \leftarrow (X \oplus U_{XY}) \vee U_Y \end{cases} \end{cases}$$

The task is to evaluate the generalization error of the classifier $h(x) = \neg x$. h can be shown optimal in both source domains: achieving $R_{P^1}(h) \approx 0.11$ and $R_{P^2}(h) \approx 0.06$. However, it is unclear whether it generalizes well to a target domain $\mathcal{M}^*$, given the domain discrepancy sets $\Delta_1 = \{X\}, \Delta_2 = \{Y\}$. □

Balke and Pearl [3] derived a canonical parameterization of SCMs such as $\{\mathcal{M}^1, \mathcal{M}^2, \mathcal{M}^*\}$ in Example 3. They showed that it is sufficient to parameterize $P(\boldsymbol{U})$ with correlated discrete latent variables R_X, R_Y, where R_X determines the value of X, and R_Y determines the functional that decides Y based on X. The causal diagrams are shown in Figure 2. Canonical SCMs entails the same set of distributions as the true underlying SCMs, *i.e.* are equally expressive. In particular, Zhang and Bareinboim [42] showed that for every SCM $\mathcal{M}$, there exists an SCM of the described form specified with only a distribution $P(r_X, r_Y)$, where, $\text{supp}_{R_X} = \{0, 1\}, \quad \text{supp}_{R_Y} = \{y = 0, y = 1, y = x, y = \neg x\}$. The joint distribution $P(r_X, r_Y)$ can be parameterized by a vector in 8-dimensional simplex, and entails all observational, interventional and counterfactual variables generated by the original SCM.



(a) $\mathcal{G}^{\boldsymbol{\Delta}}$ (b) $\mathcal{G}^{\mathcal{N}}$

Figure 2: Selection diagram & Canonical param.

The following definition by Zhang et al. [42] provides a general formulation of canonical models.

Definition 5 (Canonical SCM)**.** A canonical SCM is an SCM $\mathcal{N} = \langle \boldsymbol{U}, \boldsymbol{V}, \mathcal{F}, P(\boldsymbol{U}) \rangle$ defined as follows. The set of endogenous variables $\boldsymbol{V}$ is discrete. The set of exogenous variables $\boldsymbol{U} = \{R_V : V \in \boldsymbol{V}\}$, where $\text{supp}_{R_V} = \{1, \ldots, m_V\}$ and $m_V = |\{h_V : \text{supp}_{pa_V} \rightarrow \text{supp}_V\}|$. For each $V \in \boldsymbol{V}$, $f_V \in \mathcal{F}$ is defined as $f_V(pa_V, r_V) = h_V^{(r_V)}(pa_V)$.

Example 3 (continued)**.** Consider extending the canonical parameterization to to solve the partial transportability task by optimization. Each SCM $\mathcal{M}^1, \mathcal{M}^2, \mathcal{M}^*$ is associated with a canonical SCM $\mathcal{N}^1, \mathcal{N}^2, \mathcal{N}^*$. with exogenous variables $\{R_X, R_Y\}$ as above. The domain discrepancy sets $\boldsymbol{\Delta}$ indicate that certain causal mechanisms need to match across pairs of the SCMs. For example, $\Delta_1 = \{X\}$, which does not contain Y, and this means that (1) the function f_Y is the same across $\mathcal{M}^1, \mathcal{M}^*$, and (2) the distribution of unobserved variables that are arguments of f_Y, namely, U_{XY}, U_Y remains the same across $\mathcal{M}^1, \mathcal{M}^*$. Imposing these equalities on the canonical parameterization is straightforward as (1) the function f_Y is the same across all canonical SCMs by construction, and (2) the only unobserved variable pointing to variable V is R_V (for $V \in \{X, Y\}$). Following the selection diagram shown in Fig. 2a, $\mathcal{M}^1, \mathcal{M}^*$ agree on the mechanism of Y, which translates to the constraint $P^{\mathcal{N}^1}(r_Y) = P^{\mathcal{N}^*}(r_Y)$. Similarly, $\mathcal{M}^2, \mathcal{M}^*$ agree on the mechanism of X that translates to the constraint $P^{\mathcal{N}^2}(r_X) = P^{\mathcal{N}^*}(r_X)$. Putting these together, the optimization problem below finds the upper-bound for the risk $R_{P^*}(h)$ for the classifier $h(x) = \neg x$:

$$\max_{\mathcal{N}^1, \mathcal{N}^2, \mathcal{N}^*} P^{\mathcal{N}^*}(Y \neq \neg X) \tag{5}$$

$$\text{s.t. } P^{\mathcal{N}^1}(r_Y) = P^{\mathcal{N}^*}(r_Y), \quad P^{\mathcal{N}^2}(r_X) = P^{\mathcal{N}^*}(r_X) \qquad (Y \notin \Delta_1, \text{and } X \notin \Delta_2)$$

$$P^{\mathcal{N}^1}(x, y) = P^1(x, y), \quad P^{\mathcal{N}^2}(x, y) = P^2(x, y) \qquad (\text{matching source dists})$$

Notably, the above optimization has a linear objective with linear equality constraints. □

This example illustrates a more general strategy, in which probabilities induced by an SCM over discrete endogenous variables $\boldsymbol{V}$ may be generated by a canonical model. What follows is the main result of this section, and provides a systematic approach to partial transportability using the canonical models.

https://doi.org/10.52202/079017-4376

Theorem 1 (Partial-TR with canonical models)**.** *Consider the tuple of SCMs $\mathbb{M}$ that induces the selection diagram $\mathcal{G}^{\boldsymbol{\Delta}}$ over the variables $\boldsymbol{V}$, and entails the source distributions $\mathbb{P}$, and the target distribution P^*. Let $\psi : \Omega_{\boldsymbol{V}} \to \mathbb{R}$ be a functional of interest. Consider the following optimization scheme:*

$$\max_{\mathcal{N}^1,\mathcal{N}^2,\ldots,\mathcal{N}^*} \mathbb{E}_{P^{\mathcal{N}^*}}[\psi(\boldsymbol{V})] \text{ s.t. } P^{\mathcal{N}^i}(\boldsymbol{v}) = P^i(\boldsymbol{v}) \qquad \forall i \in \{1,2,\ldots,K,*\} \tag{6}$$

$$P^{\mathcal{N}^i}(r_V) = P^{\mathcal{N}^j}(r_V), \quad \forall i,j \in \{1,2,\ldots,K,*\} \quad \forall V \notin \Delta_{i,j}$$

where each $\mathcal{N}^i$ is a canonical model characterized by a joint distribution over $\{R_V\}_{V\in\boldsymbol{V}}$. The value of the above optimization is a tight upper-bound for the quantity $\mathbb{E}_{P^*}[\psi(\boldsymbol{V})]$ among all tuples of SCMs that induce $\mathcal{G}^{\boldsymbol{\Delta}}$ and entail $\mathbb{P}$. □

In words, this Theorem states that one may tightly bound the value of a target quantity $\mathbb{E}_{P^*}[\psi(\boldsymbol{V})]$ by optimizing over the space of canonical models subject to the proposed constraints, without any loss of information. An implementation of Thm. 1 approximating the worst-case error, by making inference on the posterior distribution of the target quantity, is provided in Appendix A.

4 Neural Causal Models for Partial Transportability

In this section we consider inferences in more general settings by using neural networks as a generative model, acting as a proxy for the underlying SCMs $\mathbb{M}$ with the potential to scale to real-world, high-dimensional settings while preserving the validity and tightness of bounds. For this purpose, we consider Neural Causal Models [41] and adapt them for the partial transportability task. What follows is an instantiation of [41, Definition 7].

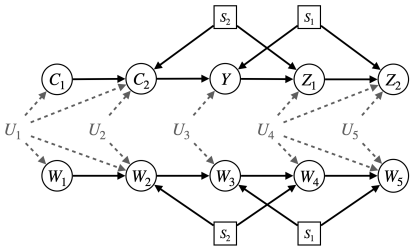


Figure 3: Selection diagram for Example 4.

Definition 6 (Neural Causal Model)**.** A Neural Causal Model (NCM) corresponding to the causal diagram $\mathcal{G}$ over the discrete variables $\boldsymbol{V}$ is is an SCM defined by the exogenous variables:

$$\boldsymbol{U} = \{U_{\boldsymbol{W}} \sim \text{unif}(0,1) : \boldsymbol{W} \subseteq \boldsymbol{V} \text{ s.t. } A \leftrightarrow B \in \mathcal{G}, \quad \forall A,B \in \boldsymbol{W}\}, \tag{7}$$

and the functional assignments $V \leftarrow f_{\theta_V}(Pa_V, \boldsymbol{U}_V)$, where $\boldsymbol{U}_V = \{U_{\boldsymbol{W}} \in \boldsymbol{U} : V \in \boldsymbol{W}\}$. The function f_{θ_V} is a feed-forward neural network parameterized with θ_V that outputs in supp_V. The distribution entailed by an NCM is denoted by $P(\boldsymbol{v};\theta)$, where $\theta = \{\theta_V\}_{V\in\boldsymbol{V}}$.

To illustrate how one might leverage this parameterization to define an instance of partial transportability task consider the following example.

Example 4. Let SCMs $\mathcal{M}^1, \mathcal{M}^2, \mathcal{M}^*$ induce $\mathcal{G}^{\boldsymbol{\Delta}}$ shown in Fig. 3 over the binary variables $\boldsymbol{X}, Y$, where $\boldsymbol{X} = \{C_1, C_2, Z_1, Z_2, W_1, \ldots, W_5\}$. Let $\theta^1, \theta^2, \theta^*$ be the parameters of NCMs constructed based on the causal diagram in Fig. 3 (without the s-nodes). The objective is to constrain these parameters to simulate a compatible tuple of NCMs $\mathcal{M}_{\theta^1}, \mathcal{M}_{\theta^2}, \mathcal{M}_{\theta^*}$ that equivalently entail $P^1(\boldsymbol{x}, y), P^2(\boldsymbol{x}, y)$ and induce $\mathcal{G}^{\boldsymbol{\Delta}}$.

For instance, the fact that S_2 is not pointing to Y suggests the invariance $f^*_Y = f^2_Y$ and $P^*(\boldsymbol{u}_Y) = P^2(\boldsymbol{u}_Y)$ for the true underlying SCMs. That same invariance may be enforced in the corresponding NCMs by relating the parameterization of $\mathcal{M}_{\theta^2}, \mathcal{M}_{\theta^*}$, i.e., imposing that $\theta^*_Y = \theta^2_Y$ for the NN generating Y. Similarly, the observed data D^1, D^2 from the source distributions $P^1(\boldsymbol{x}, y), P^2(\boldsymbol{x}, y)$, respectively, impose constraints on the parameterization of NCMs as plausible models must satisfy $P(\boldsymbol{x}, y; \theta^1) = P^1(\boldsymbol{x}, y)$ and $P(\boldsymbol{x}, y; \theta^2) = P^2(\boldsymbol{x}, y)$. This may be enforced, for instance, by maximizing the likelihood of data w.r.t. the NCM parameters: $\theta^i \in \arg\max_\theta \sum_{\boldsymbol{x},y\in D^i} \log P(\boldsymbol{x}, y; \theta^i)$, for $i \in \{1, 2\}$. By extending this intuition for all constraints imposed by the selection diagram and data, we narrow the set of NCMs $\mathcal{M}_{\theta^1}, \mathcal{M}_{\theta^2}, \mathcal{M}_{\theta^*}$ to a set that is compatible with our assumptions and data. Maximizing the risk of some prediction function $R_{P^*}(h)$ in this class of constrained NCMs might then achieve an informative upper-bound. □

Motivated by the observation in Example 4, we now show a more formal result (analogous to Thm. 1) that guarantees that the solution to the partial transportability task in the space of constrained NCMs achieves a tight bound on a given target quantity $\mathbb{E}_{P^*}[\psi(\boldsymbol{V})]$.

Theorem 2 (Partial-TR with NCMs)**.** *Consider a tuple of SCMs $\mathbb{M}$ that induces $\mathcal{G}^{\boldsymbol{\Delta}}, \mathbb{P}$ and P^* over the variables $\boldsymbol{V}$. Let $D^i \sim P^i(x, y)$ denote the samples drawn from the i-th source domain. Let θ^i denote the parameters of NCM corresponding to $\mathcal{M}^i \in \mathbb{M}$. Let $\mathbb{E}_{P^*}[\psi(\boldsymbol{V})]$ be the target quantity. The solution to the optimization problem,*

$$\hat{\Theta} \in \underset{\Theta:\langle\theta^1,\theta^2,\ldots,\theta^K,\theta^*\rangle}{\arg\max} \sum_{\boldsymbol{w}} \psi(\boldsymbol{w}) \cdot \sum_{\boldsymbol{v}\setminus\boldsymbol{w}} P(\boldsymbol{v};\theta^*) \tag{8}$$

$$\text{s.t. } \theta^i_V = \theta^j_V, \qquad \forall i,j \in \{1,2,\ldots,K,*\} \quad \forall V \notin \Delta_{i,j}$$

$$\theta^i \in \arg\max_{\theta} \sum_{\boldsymbol{v}\in D^i} \log P(\boldsymbol{v};\theta), \quad \forall i \in \{1,2,\ldots,K\}.$$

is a tuple of NCMs that induce $\mathcal{G}^{\boldsymbol{\Delta}}$, entails $\mathbb{P}$. In the large sample limit, the solution yields a tight upper-bound for $\mathbb{E}_{P^*}[\psi(\boldsymbol{V})]$. □

Theorem 2 establishes the expressive power of NCMs for solving partial transportability tasks. This formulation is powerful because it enables the use of gradient-based optimization of neural networks for learning and, in principle, might scale to large number of variables.

4.1 Neural-TR: An Efficient Implementation

We could further explore the efficient optimization of parameters by exploiting the separation between variables in the selection diagram. Rahman et al. [28], for instance, show that the NCM parameterization is modular w.r.t. the c-components of the causal diagram. We can similarly elaborate on this property, and leverage it for more efficient partial transportability.

In the following, we build towards an efficient algorithm for partial transportability using NCMs by first showing an example that describes how a given target quantity $\mathbb{E}_{P^*}[\psi(\boldsymbol{V})]$ might be decomposed for learning more efficiently.

Example 4 (continued)**.** $P(\boldsymbol{x}, y; \theta^*)$ in the objective in Eq. (8) may be decomposed as follows:

$$P(\boldsymbol{x}, y; \theta^*) = P^*(\underbrace{c_1, c_2, w_1, w_2}_{\boldsymbol{a}_1}; \theta^*_{\boldsymbol{A}_1}) \cdot P(\underbrace{y, w_3}_{\boldsymbol{a}_2} \mid \underbrace{c_2, w_2}_{\boldsymbol{b}_2}; \theta^*_{\boldsymbol{A}_2}) \cdot P(\underbrace{z_1, z_2, w_4, w_5}_{\boldsymbol{a}_3} \mid \underbrace{y, w_3}_{\boldsymbol{b}_3}; \theta^*_{\boldsymbol{A}_3}),$$

where the subsets $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3$ are the c-components of $\mathcal{G}^*$. Notice, S_2 is not pointing to any of the variables $\boldsymbol{A}_2$, which means that their mechanism is shared across $\mathcal{M}^2, \mathcal{M}^*$, and therefore,

$$P(\boldsymbol{a}_2 \mid \boldsymbol{b}_2; \theta^*_{A_2}) = P(\boldsymbol{a}_2 \mid \boldsymbol{b}_2; \theta^2_{A_2}) \approx P^2(\boldsymbol{a}_2 \mid \boldsymbol{b}_2). \tag{9}$$

This property is the basis of transportability algorithms [4, 10], and is known as the s-admissibility criterion [26], which allows us to deduce distributional invariances from structural invariances. By Eq. (9), we can replace the term $P(\boldsymbol{a}_2 \mid \boldsymbol{b}_2; \theta^*_{A_2})$ in the objective with the probabilistic model $P(\boldsymbol{a}_2 \mid \boldsymbol{b}_2; \eta^2)$ that is trained with D^2 to approximate $P^2(\boldsymbol{a}_2 \mid \boldsymbol{b}_2)$ and plug it into the objective $Eq.$ (8) as a constant.

As a consequence, we do not need to optimize over the parameters $\theta^1_{\boldsymbol{A}_2}, \theta^2_{\boldsymbol{A}_2}, \theta^*_{\boldsymbol{A}_2}$ from the partial transportability optimization problem. Similarly, since S_1 does not point to $\boldsymbol{A}_1$, we can substitute $P(\boldsymbol{a}_1; \theta^*)$ with $P(\boldsymbol{a}_1; \eta^1)$, and pre-train it with data D^1. In the context of Example 4 and the evaluation of $R_{P^*}(h)$, the objective in Eq. (8) may be simplified to the substantially lighter optimization task:

$$\max_{\theta^1_{\boldsymbol{A}_3}, \theta^2_{\boldsymbol{A}_3}, \theta^*\boldsymbol{A}_3} \mathbb{E}_{\boldsymbol{A}_1 \sim P(\boldsymbol{a}_1;\eta^1)}\Big[\mathbb{E}_{\boldsymbol{A}_2 \sim P(\boldsymbol{a}_2|\boldsymbol{b}_2;\eta^2)}\Big[\sum_{\boldsymbol{a}_3} P(\boldsymbol{a}_3 \mid \boldsymbol{b}_3; \theta^*_{\boldsymbol{A}_3}) \cdot \mathbb{1}\{h(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 \backslash \{y\}) \neq y\}\Big]\Big]$$

$$\text{s.t. } \theta^i_{\boldsymbol{A}_3} \in \arg\max_{\theta_{\boldsymbol{A}_3}} \sum_{\boldsymbol{a}_3, \boldsymbol{b}_3 \in D^i} \log P(\boldsymbol{a}_3 \mid \boldsymbol{b}_3; \theta_{\boldsymbol{A}_3}), \quad \text{for } i \in \{1,2\}. \tag{10}$$

In general, the parameter space of NCMs can be cleverly decoupled and the computational cost of the optimization problem can be significantly improved since only a subset of the conditional distributions need to be parameterized and optimized. This observation motivates Alg. 1 designed to exploit these insights. It proceeds by first, decomposing the query, second, computing the identifiable components, and third, parameterizing the components that are not point identifiable and running the NCM optimization routine. The following proposition demonstrates the correctness of this procedure.

https://doi.org/10.52202/079017-4376

Algorithm 1 Neural-TR

Require: Source data $D^1, D^2, \ldots, D^K$; selection diagram $\mathcal{G}^{\boldsymbol{\Delta}}$; functional $\psi : \Omega_{\boldsymbol{W}} \to [0, 1]$.
Ensure: Upper-bound for $\mathbb{E}_{P^*}[\psi(\boldsymbol{W})]$

$\{\boldsymbol{A}_j\}_{j=1}^m \leftarrow$ c-components of $\boldsymbol{A} := \mathbf{An}_{\mathcal{G}^*}(\boldsymbol{W})$ in causal diagram $\mathcal{G}^*$.
$\Theta, \mathbb{C}_{\text{expert}} \leftarrow \varnothing, \quad \mathcal{L}_{\text{data}} \leftarrow 0$
$P^*(\boldsymbol{w}) := \sum_{\boldsymbol{a} \setminus \boldsymbol{w}} \prod_{j=1}^m P^*(\boldsymbol{a}_j \mid \text{do}(pa_{\boldsymbol{A}_j}))$
for $j = 1$ to m **do**
  if $\exists i \in \{1, 2, \ldots, K\}$ such that $\boldsymbol{A}_j \cap \Delta_{*i} = \varnothing$ **then**
    $\eta^i_{\boldsymbol{A}_j} \leftarrow \arg\max_{\eta_{\boldsymbol{A}_j}} \sum_{\boldsymbol{a}_j, pa_{\boldsymbol{A}_j} \in D^i} \log P(\boldsymbol{a}_j \mid \text{do}(pa_{\boldsymbol{A}_j}); \eta_{\boldsymbol{A}_j})$
    In $P^*(\boldsymbol{w})$, replace $P^*(\boldsymbol{a}_j \mid \text{do}(pa_{\boldsymbol{A}_j}))$ with $P(\boldsymbol{a}_j \mid \text{do}(pa_{\boldsymbol{A}_j}); \eta^i_{\boldsymbol{A}_j})$.
  else
    $\Theta \leftarrow \Theta \cup \{\theta^i_{\boldsymbol{A}_j}\}_{i \in \{1,2,\ldots,K,*\}}$
    In $P^*(\boldsymbol{w})$, replace $P^*(\boldsymbol{a}_j \mid \text{do}(pa_{\boldsymbol{A}_j}))$ with $P(\boldsymbol{a}_j \text{do}(pa_{\boldsymbol{A}_j}); \theta^*_{\boldsymbol{A}_j})$.
    $\mathbb{C}_{\text{expert}} \leftarrow \mathbb{C}_{\text{expert}} \cup \{\{\theta^i_V = \theta^*_V\}_{V \in \boldsymbol{A}_j \setminus \Delta_{*i}}\}_{i=1}^K$
    $\mathcal{L}_{\text{likelihood}} \leftarrow \mathcal{L}_{\text{likelihood}} + \sum_{\boldsymbol{a}_j, pa_{\boldsymbol{A}_j} \in D^i} \log P(\boldsymbol{a}_j, \text{do}(pa_{\boldsymbol{A}_j}); \theta^i_{\boldsymbol{A}_j})$.
  end if
end for
Return $\hat{\Theta} \leftarrow \arg\max_{\Theta} \sum_{\boldsymbol{w}} P^*(\boldsymbol{w}; \Theta) \cdot \psi(\boldsymbol{w}) + \Lambda \cdot \mathcal{L}_{\text{likelihood}}(\Theta)$ subject to $\mathbb{C}_{\text{expert}}$

Proposition 1. *Neural-TR (Algorithm 1) computes a tuple of NCMs compatible with the source data and graphical assumptions that yields the upper-bound for $\mathbb{E}_{P^*}[\psi(\boldsymbol{W})]$ in the large sample limit.* □

This result may be understood as an enhancement of Thm. 2 in which the factors that are readily transportable from source data are taken care of in a pre-processing step. The hybrid approach is especially useful in case researchers have pre-trained probabilistic models with arbitrary architecture that they can use off-the-shelf and avoid unnecessary computation.

4.2 Neural-TR for the Optimization of Classifiers

The Neural-TR algorithm can be viewed as an adversarial domain generator that takes a classifier $h(\boldsymbol{z})$ as the input, and then parameterizes a collection of SCMs to find a plausible target domain that yields the worst-case risk for the given classifier, namely, $\hat{\theta}^*$. By flipping $h(\boldsymbol{z})$ for some $\boldsymbol{z} \in \Omega$ we can reduce the risk of h under $\hat{\theta}^*$.

Algorithm 2 CRO (Causal Robust Optimization)

Require: $\mathbb{D} : \langle D^1, D^2, \ldots, D^K \rangle$; $\mathcal{G}^{\boldsymbol{\Delta}}$; $\delta > 0$
Ensure: $h(\boldsymbol{X})$ with the best worst-case risk.

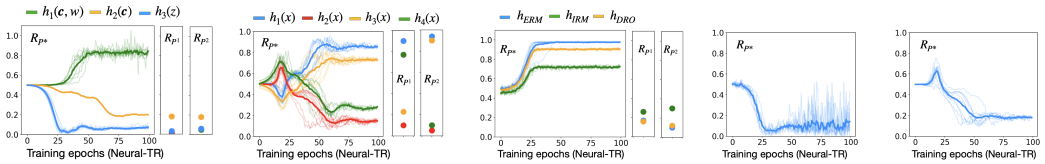
Initialize h randomly and $\mathbb{D}^* \leftarrow \varnothing$
$\hat{\Theta} \leftarrow$ Neural-TR$(\mathbb{D}, \mathcal{G}^{\boldsymbol{\Delta}}, \psi : \mathcal{L}(h(\boldsymbol{x}), y))$
while $R_{P(\boldsymbol{x}, y; \hat{\theta}^*)}(h) - \max_{D \in \mathbb{D}^*} R_D(h) > \delta$ **do**
  $\mathbb{D}^* \leftarrow \mathbb{D}^* \cup \{D^* \sim P(\boldsymbol{x}, y; \hat{\theta}^*)\}$
  $h \leftarrow \arg\min_h \max_{D \in \mathbb{D}^*} R_D(h)$
  $\hat{\Theta} \leftarrow$ Neural-TR$(\mathbb{D}, \mathcal{G}^{\boldsymbol{\Delta}}, \psi : \mathcal{L}(h(\boldsymbol{x}), y))$
end while
Return h

Interestingly, we can exploit Neural-TR to generate adversarial data for a given classifier and introduce an iterative procedure to progressively train classifiers with with minimum risk upper-bound. Algorithm 2 describes this approach. At each iteration, CRO uses Neural-TR as a subroutine to obtain an adversarially designed NCM $\hat{\theta}^*$ that yields the worst-case risk for the classifier at hand. Next, it collects data D^* from this NCM and adds it to a collection of datasets $\mathbb{D}^*$. Finally, it updates the classifier to be robust to the collection $\mathbb{D}^*$ by minimizing the maximum of the empirical risk $R_D(h) := \sum_{\boldsymbol{x}, y \in D} \mathcal{L}(y, h(x))$ across all $D \in \mathbb{D}^*$. We repeat this process until convergence of the upper-bound for risk. The following result justifies optimality of CRO for domain generalization; more discussion is provided in Appendix C.2.

Theorem 3 (Domain generalization with CRO)**.** *Algorithm 2 returns a worst-case optimal solution;*

$$\text{CRO}(\mathbb{D}, \mathcal{G}^{\boldsymbol{\Delta}}) \in \arg\min_{h: \Omega_{\boldsymbol{X}} \to \Omega_Y} \max_{\text{tuple of SCMs } \mathbb{M}_0 \text{ that entails } \mathbb{P} \text{ \& induces } \mathcal{G}^{\boldsymbol{\Delta}}} R_{P^{\mathcal{M}_0^*}}(h). \tag{11}$$

In words, Thm. 3 states that the classifier returned by CRO, in the large sample limit, minimizes worst-case risk in the target domain subject to the constraints entailed by the available data and induced by the structural assumptions.



(a) Example 2 (b) Example 3 (c) CMNIST (d) Example 2 (e) Example 3

Figure 4: (a-c): worst-case risk evaluation results as a function of Neural-TR (Alg. 1) training iterations. (d,e): worst-case risk evaluation of CRO.

5 Experiments

This section illustrates Algs. 1 and 2 for the evaluation and optimization of the generalization error on several tasks, ranging from simulated examples to semi-synthetic image datasets. The details of the experimental set-up and examples not fully described below, along with additional experiments, can be found in the Appendix.

5.1 Simulations

Worst-case risk evaluation Our first experiment revisits Examples 2 and 3 for the evaluation of the worst-case risk R_{P*} of various classifiers with Neural-TR (Alg. 1).

In Example 2 we had made (anecdotal) performance observations for the classifiers $h_1(\boldsymbol{c}, w) := w \oplus \bigoplus_{c \in \boldsymbol{c}} c, h_2(\boldsymbol{c}) := \bigoplus_{c \in \boldsymbol{c}} c, h_3(z) := z$ in a selected target domain $\mathcal{M}^*$. We now consider providing a worst-case risk *guarantee* with Neural-TR for *any* (compatible) target domain. The main panel in Fig. 4a shows the convergence of the worst-case risk evaluator over successive training iterations (line 15, Alg. 1), repeated 10 times with different model seeds and solid lines denoting the mean worst-case risk. The source performances R_{P_1}, R_{P_2} are given in the two right-most panels for reference. We observe that the good source performance of $h_2(\boldsymbol{c})$ and $h_3(z)$ generalizes to *all* possible target domains consistent with our assumptions, while the classifier $h_1(\boldsymbol{c}, w)$ diverges, with an error of 90% in the worst target domain. In Example 3, we consider the evaluation of binary classifiers $h \in \{h_1(x) := x, h_2(x) := \neg x, h_3(x) := 0, h_4(x) := 1\}$. $h_2(x) = \neg x$. Our results are given in Fig. 4b, highlighting the extent to which source performance need not be indicative of target performance. With these results, we are now in a position to confirm the desirable performance profile of h_2, even in the worst-case, as hypothesized in Example 3.

Worst-case risk optimization For each one of the examples above, we implement CRO (Alg. 2) to uncover the theoretically optimal classifier in the worst-case. The worst-case risks of the classifiers learned by CRO, denoted h_{CRO}, are given by 0.05 for Example 2 and 0.18 for Example 3. The worst-case risk evaluation results (with Neural-TR, as above) are given in Figs. 4d and 4e. It is interesting to note that these errors coincide with the best performing classifiers considered in the previous experiment, i.e. $h_3(z) := z$ for Example 2 and $h_2(x) = \neg x$ for Example 3. In fact, by comparing the outputs of CRO h_{CRO} with these classifiers, we can verify that the classifiers learned by CRO in these examples are precisely the mappings $h_{\text{CRO}}(z) := z$ and $h_{\text{CRO}}(x) = \neg x$ which is remarkable. By Thm. 3, $h_3(z) := z$ and $h_2(x) = \neg x$ are the theoretically best worst-case classifiers among all possible functions given the data and assumptions.

5.2 Colored MNIST

Our second experiment considers the colored MNIST (CMNIST) dataset that is used in the literature to highlight the robustness of classifiers to spurious correlations, e.g. see [2]. The goal of the classifier is to predict a binary label $Y \in \{0, 1\}$ assigned to an image $\boldsymbol{Z} \in \mathbb{R}^{28 \times 28 \times 3}$ based on whether the digit in the image is greater or equal to five. MNIST images $\boldsymbol{W} \in \mathbb{R}^{28 \times 28}$ are grayscale (and latent), and color $C \in \{\text{red}, \text{green}\}$ correlates with the class label Y.

Following standard implementations, we construct datasets from three domains with varying correlation strength between the color and image label: set to 90% agreement between the color $C =$red and label $Y = 1$ in source domain $\mathcal{M}^1$, and 80% in source domain $\mathcal{M}^2$. We consider performance evaluation and optimization in a target domain $\mathcal{M}^*$ with potential discrep-

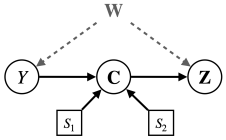
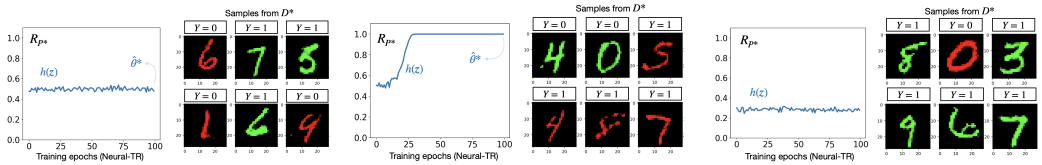


Figure 5: $\mathcal{G}^{\Delta}_{\text{CMNIST}}$

https://doi.org/10.52202/079017-4376



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

Figure 6: Illustration of the CRO training process (Alg. 2) on the colored MNIST task.

ancies in the mechanism for C, rendering the correlation between color and label unstable. The selection diagram is given in Figure 5.

Worst-case risk evaluation Consider a setting in which we are given a classifier $h : \Omega_{\boldsymbol{Z}} \to \Omega_Y$, and the task is to assess its generalizability with a symmetric 0-1 loss function. We use data drawn from $P^{1,2}(\boldsymbol{z}, y)$ to train predictors using Empirical Risk Minimization (ERM) [38], Invariant Risk Minimization (IRM) [2], and group Distributionally Robust Optimization (group DRO) [32], namely $h_{\text{ERM}}(\boldsymbol{z}), h_{\text{IRM}}(\boldsymbol{z})$, and $h_{\text{DRO}}(\boldsymbol{z})$ respectively; more detailed discussion about the role of invariance and robustness in domain generalization is available in appendix D. Using Neural-TR, we observe in Fig. 4c that the worst-case risk of h_{ERM} in a target domain with a discrepancy in the color assignment is approximately 0.95, h_{DRO} achieves 0.90 worst-case risk, and h_{IRM} achieves 0.65 worst-case risk. Either method perform worse than the baseline, that is random classification with risk 0.5. On this task, a classifier trained on gray-scale images $\boldsymbol{W}$ achieves a worst-case error of 0.25.

Worst-case risk optimization We now ask whether we could learn a theoretically optimal classifier in the worst-case with CRO (Alg. 2). Fig. 6 illustrates the training process over several iterations. Specifically, given a randomly initialized h, we infer the NCM $\hat{\theta}^*$ that entails worst-case performance of h (in this case, chance performance $R_{P^*}(h) = 0.5$) and generate data D^* from $\hat{\theta}^*$, shown in Fig. 6a. In a second iteration, a new candidate h is trained to minimize worst-case risk on $\mathbb{D} = D^*$. Note that in D^*, we observe an almost perfect association between the color C =green and label $Y = 1$: h therefore is encouraged to exploit color for prediction. Its worst-case error (inferred with Neural-TR) is accordingly close to 1, and the corresponding worst-case NCM $\hat{\theta}^*$ entails a distribution of data in which the correlation between color and label is flipped: with a strong association between the color C =red and label $Y = 1$, as shown in Fig. 6b. In a third iteration, a new candidate h is trained to minimize worst-case risk on the updated $\mathbb{D}^*$ with data samples from the previous two iterations (exhibiting opposite color-label correlations). By construction, this classifier is trained to ignore the spurious association between color and label, classifying images based on the inferred digit which leads to better behavior in the worst-case: achieving a final error of approximately 0.25, as shown in Fig. 6c, which is theoretically optimal. Note, however, that the poor performance of the baseline algorithms is not directly comparable to that of CRO, since CRO has access to background information (selection diagrams) that can not be communicated with the baseline algorithms. CRO may thus be interpreted as a meta-algorithm that operates with a broader range of assumptions encoded in a certain format (i.e., the selection diagram) that enable it to find the theoretically optimal classifier for domain generalization, in contrast to the baseline algorithms.

6 Conclusion

Guaranteeing the performance of ML algorithms implemented in the wild is a critical ingredient for improving the safety of AI. In practice, evaluating the performance of a given algorithm is non-trivial. Often the performance may vary as a consequence of our uncertainty about the possible target domain, also called a non-transportable setting. In this paper, we provide the first general estimation technique for bounding an arbitrary statistic such as the classification risk across multiple domains. More specifically, we extend the formulation of canonical models and neural causal models for the transportability task, demonstrating that tight bounds may be estimated with both approaches. Building on these theoretical findings, we introduce a Bayesian inference procedure as well as a gradient-based optimization algorithm for scalable inferences in practice. Moreover, we introduce Causal Robust Optimization (CRO), an iterative learning scheme that uses partial transportability as a subroutine to find a predictor with the best worst-case risk given the data and graphical assumptions.

7 Acknowledgement

This research was supported in part by the NSF, ONR, AFOSR, DARPA, DoE, Amazon, JP Morgan, and The Alfred P. Sloan Foundation.

References

[1] Isabela Albuquerque, João Monteiro, Tiago H Falk, and Ioannis Mitliagkas. Adversarial target-invariant representation learning for domain generalization. *arXiv preprint arXiv:1911.00804*, 2019.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[3] Alexander Balke and Judea Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American Statistical Association*, 92(439):1171–1176, 1997.

[4] Elias Bareinboim, Sanghack Lee, Vasant Honavar, and Judea Pearl. Transportability from multiple environments with limited experiments. *Advances in Neural Information Processing Systems*, 26, 2013.

[5] Elias Bareinboim and Judea Pearl. Transportability from multiple environments with limited experiments: Completeness results. *Advances in neural information processing systems*, 27, 2014.

[6] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

[7] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[8] Donald T Campbell and Julian C Stanley. *Experimental and quasi-experimental designs for research*. Ravenio books, 2015.

[9] J. Correa and E. Bareinboim. A calculus for stochastic interventions: Causal effect identification and surrogate experiments. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, 2020. AAAI Press.

[10] J. Correa and E. Bareinboim. General transportability of soft interventions: Completeness results. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10902–10912, Vancouver, Canada, Jun 2020. Curran Associates, Inc.

[11] Juan D Correa and Elias Bareinboim. From statistical transportability to estimating the effect of stochastic interventions. In *IJCAI*, pages 1661–1667, 2019.

[12] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136. JMLR Workshop and Conference Proceedings, 2010.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Francois Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[14] Stuart S Glennan. Mechanisms and the nature of causation. *Erkenntnis*, 44(1):49–71, 1996.

[15] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[16] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

 https://doi.org/10.52202/079017-4376

[17] Kasra Jalaldoust and Elias Bareinboim. Transportable representations for domain generalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):12790–12800, Mar. 2024.

[18] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5815–5826. PMLR, 18–24 Jul 2021.

[19] Sanghack Lee, Juan D Correa, and Elias Bareinboim. Generalized transportability: Synthesis of experiments from heterogeneous domains. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020.

[20] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[21] Peter Machamer, Lindley Darden, and Carl F Craver. Thinking about mechanisms. *Philosophy of science*, 67(1):1–25, 2000.

[22] Sara Magliacane, Thijs Van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. *Advances in neural information processing systems*, 31, 2018.

[23] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[24] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[25] Judea Pearl. *Causality*. Cambridge university press, 2009.

[26] Judea Pearl and Elias Bareinboim. Transportability of causal and statistical relations: A formal approach. In *Twenty-fifth AAAI conference on artificial intelligence*, 2011.

[27] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

[28] Md Musfiqur Rahman and Murat Kocaoglu. Modular learning of deep causal generative models for high-dimensional causal inference. In *Forty-first International Conference on Machine Learning*, 2024.

[29] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

[30] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021.

[31] Dominik Rothenhäusler, Nicolai Meinshausen, Peter Bühlmann, and Jonas Peters. Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):215–246, 2021.

[32] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[33] Xinwei Shen, Peter Bühlmann, and Armeen Taeb. Causality-oriented robustness: exploiting general additive interventions. *arXiv preprint arXiv:2307.10299*, 2023.

[34] Amos Storkey. When training and test sets are different: characterizing learning transfer. 2008.

[35] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

[36] Jin Tian and Judea Pearl. *A general identification condition for causal effects*. eScholarship, University of California, 2002.

[37] US Department of Health and Human Services. The health consequences of smoking—50 years of progress: a report of the surgeon general, 2014.

[38] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

[39] Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[40] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[41] Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The causal-neural connection: Expressiveness, learnability, and inference. *Advances in Neural Information Processing Systems*, 34:10823–10836, 2021.

[42] Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. *arXiv preprint arXiv:2110.05690*, 2021.

 https://doi.org/10.52202/079017-4376

Appendix

Table of Contents

A Partial Transportability as a Bayesian Inference Task

Consider a system of multiple SCMs $\mathcal{M}^1, \mathcal{M}^2, \ldots, \mathcal{M}^K, \mathcal{M}^*$ that induces the selection diagram $\mathcal{G}^{\boldsymbol{\Delta}}$, and entails the source distributions $P^1, P^2, \ldots, P^K$, and the target distribution P^* over the variables $\boldsymbol{V}$. Let $\psi : \Omega_{\boldsymbol{X}} \to \mathbb{R}$ be a functional of interest. Consider the following optimization scheme:

$$\begin{aligned} \hat{q}_{\max} = \max_{\mathcal{N}^1, \mathcal{N}^2, \ldots, \mathcal{N}^*} & \mathbb{E}_{P^{\mathcal{N}^*}}[\psi(\boldsymbol{X})] \\ \text{s.t. } & P^{\mathcal{N}^i}(r_V) = P^{\mathcal{N}^j}(r_V), \qquad \forall i, j \in \{1, 2, \ldots, K, *\} \quad \forall V \notin \Delta_{i,j} \\ & P^{\mathcal{N}^i}(\boldsymbol{v}) = P^i(\boldsymbol{v}) \qquad \forall i \in \{1, 2, \ldots, K, *\}, \end{aligned} \tag{12}$$

where each $\mathcal{N}^i$ is a canonical model characterized by a joint distribution over $\{R_V\}_{V \in \boldsymbol{V}}$.

This section describes an Markov Chain Monte Carlo (MCMC) algorithm to approximate the optimal scalar $\hat{q}_{\max}$ upper bounding the query $\phi_{\mathcal{N}^*} := \mathbb{E}_{P^{\mathcal{N}^*}}[\psi(\boldsymbol{X})]$ above from finite samples drawn from input distributions $P^1, P^2, \ldots, P^K$. Formally, we aim to infer the value,

$$\hat{q}_{\max} : P(\phi_{\mathcal{N}^*} < \hat{q}_{\max} \mid \bar{\boldsymbol{v}}) = 1 \tag{13}$$

where $\bar{\boldsymbol{v}} := (\bar{\boldsymbol{v}}_{P^1}, \ldots, \bar{\boldsymbol{v}}_{P^k})$, $\bar{\boldsymbol{v}}_{P^i} = \{\boldsymbol{v}_{P^i}^{(j)} : j = 1, \ldots, n_i\}$ denote n_i independent sampled drawn from P^i.

We consider a setting in which we are provided with prior distributions (possibly uninformative) over parameters of the family of compatible CMs $\mathcal{N}^1, \mathcal{N}^2, \ldots, \mathcal{N}^*$. In particular, we assume that for each CM, probabilities of $P(U), U \in \boldsymbol{U}$ are drawn from uninformative Dirichlet priors; and $\mathcal{F}$ are drawn uniformly from the finite class of possible structural functions. That is, for every $U \in \boldsymbol{U}$ and every $V \in \boldsymbol{V}$,

$$P(U) \sim \texttt{Dirichlet}(\alpha_1, \ldots, \alpha_{d_U}), \quad f_V \sim \texttt{Uniform}(\Omega_{PA_V} \times \Omega_{\boldsymbol{U}_V} \mapsto \Omega_V) \tag{14}$$

where $d_U = \prod_{V \in Pa(\boldsymbol{C}_U)} |\Omega_V|$ and $\alpha_1 = \ldots = \alpha_{d_U} = 1$.

The total collection of parameters is given by the set $\{(\boldsymbol{\theta}^{\mathcal{N}^1}, \boldsymbol{\xi}^{\mathcal{N}^1}), \ldots, (\boldsymbol{\theta}^{\mathcal{N}^*}, \boldsymbol{\xi}^{\mathcal{N}^*})\}$. Among them $\boldsymbol{\theta} = \{\boldsymbol{\theta}_U \in [0,1]^{d_U} : U \in \boldsymbol{U}\}$ define the parameterization of exogenous probabilities while $\boldsymbol{\xi} = \{\xi_V^{(pa_V, \boldsymbol{u}_V)} \in \text{supp}_V : PA_V \subset \boldsymbol{V}, \boldsymbol{U}_V \subset \boldsymbol{U}\}$ define the structural functions, one set of each CM separately.

We design a Gibbs sampler to evaluate posterior distributions over these parameters. For simplicity, we describe each step of the gibbs sampler for a single domain and input dataset, and consider the implementation of constraints below.

A.1 Gibbs Sampling

The Gibbs sampler iterates over the following steps, each parameter conditioned on the current values of the remaining terms in the parameter vector.

1. *Sample* $\boldsymbol{u}$. Let $u \in \Omega_U, U \in \boldsymbol{U}$. For each observed data example across all domains $\boldsymbol{v}^{(n)} \in \bar{\boldsymbol{v}}$, $n = 1, \ldots, \sum_i n_i$, we sample corresponding exogeneous variables $U \in \boldsymbol{U}$ from the conditional distribution,

$$P(\boldsymbol{u}^{(n)} \mid \boldsymbol{v}^{(n)}, \boldsymbol{\xi}, \boldsymbol{\theta}) \propto P(\boldsymbol{u}^{(n)}, \boldsymbol{v}^{(n)} \mid \boldsymbol{\xi}, \boldsymbol{\theta}) = \prod_{V \in \boldsymbol{V}} \mathbb{1}\{\xi_V^{(pa_V^{(n)}, \boldsymbol{u}_V^{(n)})} = v^{(n)}\} \prod_{U \in \boldsymbol{U}} \theta_u. \tag{15}$$

2. *Sample* $\boldsymbol{\xi}$. Parameters $\boldsymbol{\xi}$ define deterministic causal mechanisms. For a given parameter $\xi_V^{(pa_V, \boldsymbol{u}_V)} \in \boldsymbol{\xi}$ its conditional distribution is given by $P(\xi_V^{(pa_V, \boldsymbol{u}_V)} = v \mid \bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) = 1$ if there exists a sample $(\boldsymbol{v}^{(n)}, pa_V^{(n)}, \boldsymbol{u}^{(n)})$ for some n, where n iterates over the samples of $\boldsymbol{u}$ from step 1 and $\boldsymbol{v}$ associated with the *subset* of domains in which exogeneous probabilities match the target domain, such that $\xi_V^{(pa_V^{(n)}, \boldsymbol{u}_V^{(n)})} = v^{(n)}$. Otherwise, $P(\xi_V^{(pa_V, \boldsymbol{u}_V)} = v \mid \bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})$ is given by a uniform discrete distribution over its support supp_V.

https://doi.org/10.52202/079017-4376

3. *Sample* $\boldsymbol{\theta}$. Let $\boldsymbol{\theta}_U = (\theta_1, \ldots, \theta_{d_U}) \in \boldsymbol{\theta}$ be the parameters that define the probability vector of possible values of variables $U \in \boldsymbol{U_C}$. Its conditional distribution is given by,

$$\boldsymbol{\theta}_U \mid \bar{\boldsymbol{v}}, \bar{\boldsymbol{u}} \sim \texttt{Dirichlet}\left(\alpha_1 + \beta_1, \ldots, \alpha_{d_U} + \beta_{d_U}\right),$$

where $\beta_i := \sum_n \mathbb{1}\{u^{(n)} = u_i\}$. Similarly, n iterates over the samples of $\boldsymbol{u}$ from step 1 associated with the subset of domains in which exogeneous probabilities match the target domain.

A.2 Implementing Constraints

Iterating this procedure forms a Markov chain with the invariant distribution $P(\boldsymbol{u}, \boldsymbol{\xi}, \boldsymbol{\theta} \mid \bar{\boldsymbol{v}})$. This naturally enforces the soft constraint $P^{\mathcal{N}^i}(\boldsymbol{v}) = P^i(\boldsymbol{v}), i \in \{1, 2, \ldots, K, *\}$ for the CMs defined by the sampled parameters. The posterior distributions of the subset of $(\boldsymbol{\theta}^{\mathcal{N}^*}, \boldsymbol{\xi}^{\mathcal{N}^*})$ for which invariances across domains are assumed are then matched with the posterior distribution inferred from source data. The constraint $P^{\mathcal{N}^i}(r_V) = P^{\mathcal{N}^*}(r_V), i \in \{1, 2, \ldots, K, *\}, V \notin \Delta_{i,*}$ is enforced by generating $\boldsymbol{\theta}_U^{\mathcal{N}^*}$ from the prior such that $P^{\mathcal{N}^*}(r_V) := \sum_{u \in \Omega} \theta_u^{\mathcal{N}^*} = \sum_{u \in \Omega} \theta_u^{\mathcal{N}^i} := P^{\mathcal{N}^i}(r_V), V \notin \Delta_{i,*}$ where Ω denotes the partition of supp_U that is expressed by R_V.

The query is then approximated by plugging the T MCMC samples into the query $\phi_{\mathcal{N}*}$ to obtain $\phi_{\mathcal{N}*}^{(1)}, \ldots, \phi_{\mathcal{N}*}^{(T)}$ and

$$\hat{q}_{\max} := \sup\{x : \sum_t \mathbb{1}\{\phi_{\mathcal{N}*}^{(t)} \leqslant x\} = \alpha\}. \tag{16}$$

for a chosen value of confidence value α.

Example 5 (Example 3 continued)**.** Consider again the evaluation of the risk $R_{P*}(h) := P^{\mathcal{N}^*}(Y \neq h(X))$ given the classifier $h(x) = \neg x$. We are data sampled from $P^1(x, y), P^2(x, y)$. For every SCM $\mathcal{M}$, there exists an SCM of the described format specified with only a distribution $P(r_X, r_Y)$, where,

$$\text{supp}_{R_X} = \{0, 1\}, \quad \text{supp}_{R_Y} = \{y = 0, y = 1, y = x, y = \neg x\}. \tag{17}$$

Thus, the joint distribution $P(u_{XY}) = P(r_X, r_Y)$ can be parameterized by a vector in 8-dimensional simplex. The canonical SCMs associated with each of the SCMs $\mathcal{M}^1, \mathcal{M}^2, \mathcal{M}^*$, are denoted $\mathcal{N}^1, \mathcal{N}^2, \mathcal{N}^*$, for which $\boldsymbol{V} = \{X, Y\}, \boldsymbol{U} = \{U_{XY}\}$ and $\text{supp}_{U_{XY}} = \{1, \ldots, 8\}$. The partial task can be translated into an optimization problem aiming to to find the upper-bound for the risk $R_{P*}(h)$ for the classifier $h(x) = \neg x$:

$$\begin{aligned} &\max_{\mathcal{N}^1, \mathcal{N}^2, \mathcal{N}*} P^{\mathcal{N}^*}(Y \neq \neg X) \qquad (18)\\ &\text{s.t. } P^{\mathcal{N}^1}(r_Y) = P^{\mathcal{N}^*}(r_Y), \quad P^{\mathcal{N}^2}(r_X) = P^{\mathcal{N}^*}(r_X) \qquad (Y \notin \Delta_1, \text{and } X \notin \Delta_2)\\ &\qquad P^{\mathcal{N}^1}(x, y) = P^1(x, y), \quad P^{\mathcal{N}^2}(x, y) = P^2(x, y) \qquad (\text{matching source dists}) \end{aligned}$$

With the Gibbs sampler outlined above, we obtain samples from the posterior distribution $P(\theta^{\mathcal{N}^1}, \theta^{\mathcal{N}^2}, \xi^{\mathcal{N}^1}, \xi^{\mathcal{N}^2} \mid \bar{\boldsymbol{v}})$. $\theta^{\mathcal{N}^1}, \theta^{\mathcal{N}^2}$ encode the probabilities $P^{\mathcal{N}^1}(U_{XY} = u), P^{\mathcal{N}^2}(U_{XY} = u)$ and are instantiated as two-dimensional arrays of shape $(2, 4)$ such that, e.g., $P^{\mathcal{N}^1}(r_Y) = \sum_{\text{dim. } 1} \theta^{\mathcal{N}^1}$, with $r_Y \in \{1, 2, 3, 4\}$ and similarly $P^{\mathcal{N}^1}(r_X) = \sum_{\text{dim. } 0} \theta^{\mathcal{N}^1}$, with $r_X \in \{1, 2\}$.

To enforce the constraints $P^{\mathcal{N}^1}(r_Y) = P^{\mathcal{N}^*}(r_Y), \quad P^{\mathcal{N}^2}(r_X) = P^{\mathcal{N}^*}(r_X)$ it thus suffices to sample $\theta^{\mathcal{N}^*}$ from the prior Dirichlet distribution (as it has not been updated with data) and re-scale the outcomes such that the partial row and column sums satisfy the corresponding partial row and column sums computed from the MCMC samples of $P(\theta^{\mathcal{N}^1}, \theta^{\mathcal{N}^2} \mid \bar{\boldsymbol{v}})$. The resulting MCMC parameters $(\boldsymbol{\theta}^{\mathcal{N}^*}, \boldsymbol{\xi}^{\mathcal{N}^*})$ are then valid samples from the posterior distribution $P(\theta^{\mathcal{N}^*}, \xi^{\mathcal{N}^*} \mid \bar{\boldsymbol{v}})$ subject to assumed constraints, and the risk could be computed by plugging those samples into $R_{P*}(h) := P^{\mathcal{N}^*}(Y \neq h(X))$ to obtain $R_{P*}(h)^{(1)}, \ldots, R_{P*}(h)^{(T)}$ and evaluating

$$\hat{q}_{\max} := \sup\{x : \sum_t \mathbb{1}\{R_{P*}(h)^{(t)} \leqslant x\} = \alpha\}. \tag{19}$$

for a chosen value of confidence value α. □

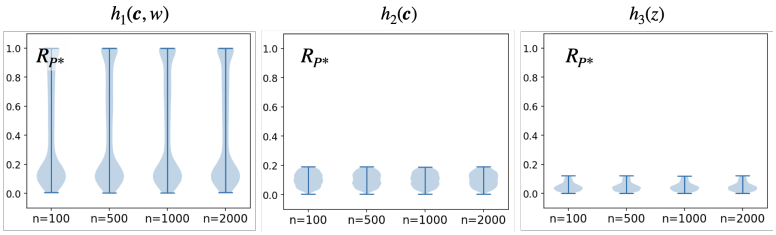


Figure 7: Violin plots that describe MCMC samples of $R_{P*}(h)$ for Example 2. The upper end-point is an estimate of $\max R_{P*}(h)$. n stands for the number of source domain samples used as a conditioning set in the posterior evaluation.

The following Theorem shows that $\hat{q}_{\max}$ converges to the true (tight) bounds $q_{\max}$ for the unknown query $R_{P*}(h)$.

Theorem 4. *$\hat{q}_{\max}$ defined in Eq. (19) is a valid upper bound on $q_{\max}$ for any sample size, and coincides with $q_{\max}$ as the sample size increases to infinity.*

Proof. Let $\boldsymbol{\Theta}$ denote the collection of parameters $\boldsymbol{\xi}, \boldsymbol{\theta}$ of discrete SCMs that generate the observed data from $P^1, P^2, \ldots$. We assume that the prior distribution on $\boldsymbol{\xi}, \boldsymbol{\theta}$ has positive support over the domain of $\boldsymbol{\Theta}$. That is, the probability density function $\rho(\boldsymbol{\xi}) > 0$ and $\rho(\boldsymbol{\theta}) > 0$ for every possible realization of $\boldsymbol{\xi}, \boldsymbol{\theta}$. By the definition of $\boldsymbol{\Theta}$, for every pair of parameter $(\boldsymbol{\xi}, \boldsymbol{\theta}) \in \boldsymbol{\Theta}$, it must be compatible with the dataset $\bar{\boldsymbol{v}}$, i.e., $P(\bar{\boldsymbol{v}} \mid \boldsymbol{\xi}, \boldsymbol{\theta}) > 0$. Similarly, given that the prior has positive support in $\boldsymbol{\Theta}$, $P(\boldsymbol{\xi}, \boldsymbol{\theta} \mid \bar{\boldsymbol{v}}) > 0$.

Note that parameters $(\boldsymbol{\xi}, \boldsymbol{\theta}) \in \boldsymbol{\Theta}$ fully determine the optimal upper bound $q_{\max}$ for $R_{P*}(h)$. And so this implies that $P(R_{P*}(h) < q_{\max} \mid \bar{\boldsymbol{v}}) > 0$, which by definition of a 100% credible interval means that $R_{P*}(h) < \hat{q}_{\max}$.

Next we show convergence of the posterior by way of convergence of the likelihood of the data given one SCM $\mathcal{M}$. For increasing sample size the posterior will, with increasing probability, be low for any parameter configuration, i.e. for any $(\boldsymbol{\xi}, \boldsymbol{\theta}) \notin \boldsymbol{\Theta}$. By the definition of the optimal upper bound $q_{\max}$ given by the solution to the partial identification task,

$$P(\bar{\boldsymbol{v}} \mid R_{P*}(h) < q_{\max}) \to_p 1. \tag{20}$$

Therefore if the prior on parameters $(\boldsymbol{\xi}, \boldsymbol{\theta})$ defining SCMs is non-zero for any $\mathcal{M}$ compatible with the data and assumptions, also the posterior converges,

$$P(R_{P*}(h) < q_{\max} \mid \bar{\boldsymbol{v}}) \to_p 1, \tag{21}$$

which is the definition of the credible value $\hat{q}_{\max}$ as the 100^{th} quantile of the posterior distribution, which coincides with $q_{\max}$ asymptotically. □

B Additional Experiments and Details

This section includes experimental details not covered in the main body of this paper as well as additional examples to illustrate our methods, including the Bayesian inference approach.

For the approximation of credible intervals and expectations required for the Bayesian inference approach, we draw 10,000 samples from posterior distributions $P(\cdot \mid \bar{\boldsymbol{v}})$ after discarding $2,000$ samples as burn-in. The results will be given a violin plots that encode the full posterior distribution of the query of interest, here the target error $R_{P*}(h)$ of a classifier h. The worst-case target error can then be read as the upper end-point of the posterior distribution.

For completeness, we provide MCMC results for Examples 2 and 3, analyzed in the main body of this paper, in Figs. 7 and 8, respectively. One could check that the upper bounds match with the analysis in the main body of this paper.

B.1 Additional Examples

This section adds additional synthetic examples to illustrate our methods.

https://doi.org/10.52202/079017-4376

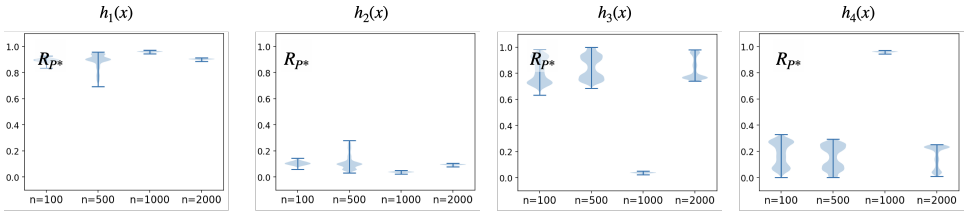
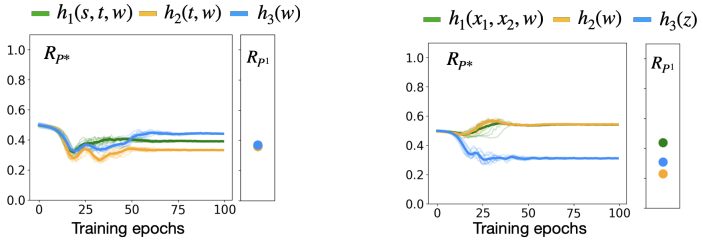


Figure 8: Violin plots that describe MCMC samples of $R_{P*}(h)$ for Example 3. The upper end-point is an estimate of $\max R_{P*}(h)$. n stands for the number of source domain samples used as a conditioning set in the posterior evaluation. Recall that $h_1(x) := x, h_2(x) := \neg x, h_3(x) := 0, h_4(x) := 1$.



(a) Example 6 (b) Example 7

Figure 9: NCM experimental results on Examples 6 and 7.

Example 6. This experiment is inspired by the debate around the relationship between smoking and lung cancer in the 1950's [37], and the corresponding selection diagram is shown in Figure 12a. We consider $\mathbb{M} : \{\mathcal{M}^1, \mathcal{M}^*\}$ that describe the effect of an individual's smoking status S on lung cancer C, including related measured variables such presence of tar in the lungs T, and demographic factors $\boldsymbol{W}$. The data generating mechanism is given by

$$\mathcal{M}^i = \begin{cases} \boldsymbol{V} & = \{\boldsymbol{W}, S, T, C\} \\ \boldsymbol{U} & = \{\boldsymbol{U}_W, U_S, U_T, U_{SC}\} \\ \mathcal{F} & = \begin{cases} f_W(u_W) & = u_W, \text{ for } W \in \boldsymbol{W}, U_W \in \boldsymbol{U}_W \\ f_S(\boldsymbol{w}, u_S, u_{SC}) & = \begin{cases} 1, & \text{if } \sum_i \frac{w_i}{d} + u_{SC} + 1.5 * u_s - 1 > 0 \text{ and } i = 1 \\ 1, & \text{if } \sum_i \frac{w_i}{d} + u_{SC} + u_S - 2 > 0 \text{ and } i = * \\ 0, & \text{otherwise} \end{cases} \\ f_T(s, u_T) & = \begin{cases} 1, & \text{if } s - 0.5u_T - 1 > 0 \\ 0, & \text{otherwise} \end{cases} \\ f_C(\boldsymbol{w}, u_C, u_{SC}) & = \begin{cases} 1, & \text{if } t - \sum_i \frac{w_i}{d} + u_{SC} - 1 > 0 \\ 0, & \text{otherwise} \end{cases} \end{cases} \\ P(\boldsymbol{U}) & \text{defined such that } U_S, U_T, U_{SC} \sim Bern(0.5), U_W \sim N(0,1), W \in \boldsymbol{W}, \end{cases}$$

Note that $\boldsymbol{\Delta} = \{S\}$ as the mechanism for S differs across domains while the mechanisms for all other variables are assumed invariant. The quantity to upper-bound is the target mean squared error: $R_{P*}(h) := \mathbb{E}_{P*}[(C-h)^2]$ of cancer prediction algorithms $h \in \{h_1(w,s,t) = \mathbb{E}_{P^1}[C \mid w,s,t], h_2(w,t) = \mathbb{E}_{P^1}[C \mid w,t], h_3(w) = \mathbb{E}_{P^1}[C \mid w]\}$ given data from P^1 and $\mathcal{G}^{\boldsymbol{\Delta}}$.

The results for the NCM approach are given in Fig. 9a. We observe that despite the discrepancy in S, all methods maintain an error of close to 0.4.

The results for the Gibbs sampling approach are given in Fig. 10. The violin plots encode the full posterior distribution of the query of interest, here the target error $R_{P*}(h)$ of a classifier h. The worst-case target error can then be read as the upper end-point of the posterior distribution. We observe that the upper-bounds from the NCM and MCMC approach approximately match. □

Example 7. This experiment considers the design of prediction rules for the development of Alzheimer's disease in a target hospital $\mathcal{M}^*$ in which no data could be recorded, and the corresponding selection diagram is shown in Figure 12b. The observed variables are given by $\boldsymbol{V} = \{X_1, X_2, W, Y, Z\}$. Among those, X_1 and X_2 are treatments for hypertension and clinical depression, respectively, both known to influence Alzheimer's disease Y, and blood pressure W. Z is a symptom of Alzheimer's. Their biological mechanisms are somewhat understood, e.g. the

https://doi.org/10.52202/079017-4376

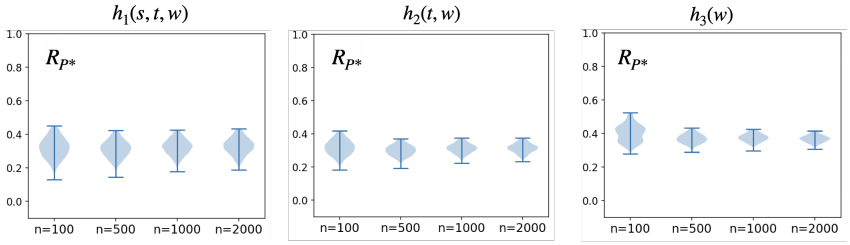


Figure 10: Violin plots that describe MCMC samples of $R_{P*}(h)$ for Example 6. The upper end-point is an estimate of $\max R_{P*}(h)$. n stands for the number of source domain samples used as a conditioning set in the posterior evaluation.

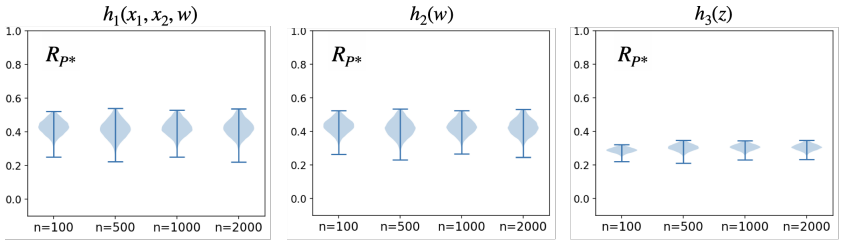


Figure 11: Violin plots with MCMC samples for Example 7. n stands for the number of source domain samples used as a conditioning set in the posterior evaluation.

effect of hypertension is mediated by blood pressure W, although several unobserved factors, such as physical activity levels and diet patterns, are expected to simultaneously affect both conditions. We assume that hypertension and clinical depression are not known to affect each other, although it's common for patients with clinical depression to simultaneously be at risk of hypertension (expressed through the presence of an unobserved common cause). More specifically, investigators have access to data from a related study conducted in domain $\mathcal{M}^1$. SCMs $\mathbb{M} : \{\mathcal{M}^1, \mathcal{M}^*\}$ are given as follows,

$$\mathcal{M}^i = \begin{cases} \boldsymbol{V} & = \{X_1, X_2, W, Y, Z\} \\ \boldsymbol{U} & = \{U_{WY}, U_{X_2}, U_W, U_{X_1X_2}, U_Z\} \\ \mathcal{F} & = \begin{cases} f_{X_1}(U_{X_1X_2}) & = \begin{cases} 1, & \text{if } U_{X_1X_2} > 0 \\ 0, & \text{otherwise} \end{cases} \\ f_{X_2}(U_{X_1X_2}, U_{X_2}) & = \begin{cases} 1, & \text{if } U_{X_1X_2} + U_{X_2} > 0 \\ 0, & \text{otherwise} \end{cases} \\ f_W(X_1, U_{WY}, U_W) & = \begin{cases} 1, & \text{if } X_1 + U_{WY} + 1.5U_W - 1 > 0 \text{ and } i = * \\ 1, & \text{if } X_1 + U_{WY} - U_W + 1 > 0 \text{ and } i = 1 \\ 0, & \text{otherwise} \end{cases} \\ f_Y(W, X_1, U_{WY}) & = \begin{cases} 1, & \text{if } W - U_{WY} + 0.1X_1 - 1 > 0 \\ 0, & \text{otherwise} \end{cases} \\ f_Z(Y, U_Z) & = \begin{cases} 1, & \text{if } Y + U_Z > 0.5 \\ 0, & \text{otherwise} \end{cases} \end{cases} \\ P(\boldsymbol{U}) & \text{defined such that } U_{WY}, U_{X_2}, U_W, U_{X_1X_2}, U_Z \sim \mathcal{N}(0,1), \end{cases}$$

Note that $\boldsymbol{\Delta} = \{W\}$ as the mechanism for W differs across domains while the mechanisms for all other variables are assumed invariant. In this example, we aim at upper-bounding the target mean squared error: $R_{P*}(h) := \mathbb{E}_{P*}[(C-h)^2]$ of cancer prediction algorithms $h \in \{h_1(x_1,x_2,w) = \mathbb{E}_{P^1}[Y \mid x_1,x_2,w], h_2(w) = \mathbb{E}_{P^1}[Y \mid w], h_3(z,t) = \mathbb{E}_{P^1}[Y \mid z]\}$ given data from P^1 and $\mathcal{G}^{\Delta*1}$.

The results for the NCM approach are given in Fig. 9b. We observe that the discrepancy in W leads to poor performance for all methods (chance level) except for h_3 that outperforms.

The results for the Gibbs sampling approach are given in Fig. 11. The violin plots encode the full posterior distribution of the query of interest, here the target error $R_{P*}(h)$ of a classifier h. The worst-case target error can then be read as the upper end-point of the posterior distribution. We observe that the upper-bounds from the NCM and MCMC approach approximately match. □

https://doi.org/10.52202/079017-4376

(a) Example 6 (b) Example 7

Figure 12: Selection diagrams for additional experiments

B.2 More on Colored MNIST

Consider handwritten grayscale digits $\boldsymbol{W} \in [0,1]^{28\times 28}$ that are annotated with $Y \in \{0,1,\dots,9\}$ and colored with $C \in \{\text{red}, \text{green}\}$, resulting in colored images $\boldsymbol{Z} \in [0,1]^{28\times 28\times 3}$. What follows describes the underlying SCM for domain $i \in \{1,2,*\}$:

$$\mathcal{M}^i : \begin{cases} \boldsymbol{W}, U_Y, \boldsymbol{U_C}, \boldsymbol{U_Z} \sim P(\boldsymbol{w}) \cdot P(u_Y) \cdot P(\boldsymbol{u}_C) \cdot P(\boldsymbol{u_Z}) \\ \mathcal{F}^i : \begin{cases} Y \leftarrow f_Y(\boldsymbol{W}, U_Y) \quad \text{(The annotation mechanism)} \\ C \leftarrow f_C^i(Y, \boldsymbol{U_C}) \quad \text{(The choice of color based on digit)} \\ \boldsymbol{Z} \leftarrow \boldsymbol{W} \cdot C^\top + \boldsymbol{U_Z} \quad \text{(Coloring image } \boldsymbol{W} \text{ with color } C) \end{cases} \end{cases}$$

In words, the grayscale image of handwritten digits $\boldsymbol{W}$ is generated according to a distribution $P(\boldsymbol{w})$ shared across all domains. The label Y is the annotation of the image with the corresponding digit through mechanism f_Y shared across all domains; the variable U_Y accounts for the possible error in annotation. Next, the color is chosen based on the digit Y following some stochastic policy $f_C^i(\cdot, U_C)$ that changes across the source and target domains. Finally, the colored image $\boldsymbol{Z}$ is produced by product of the grayscale image $\boldsymbol{W}$ and the color C; exogenous variable $\boldsymbol{U_Z}$ accounts for possible noise in coloring.

We have a classifier $h : \Omega_{\boldsymbol{Z}} \to \Omega_Y$ at hand, and the task is to assess its generalizability. Consider the following derivation:

$$\begin{aligned} P^*(\boldsymbol{z}, y) &= \sum_{\boldsymbol{c}} P^*(y, \boldsymbol{c}, \boldsymbol{z}) && (22) \\ &= \sum_{\boldsymbol{c}} P^*(y) \cdot P^*(\boldsymbol{c} \mid y) \cdot P^*(\boldsymbol{z} \mid \boldsymbol{c}, y) && (23) \\ &= P^*(y) \sum_{\boldsymbol{c}} P^*(\boldsymbol{c} \mid y) \cdot P^{1,2}(\boldsymbol{z} \mid c, y) \qquad S_1, S_2 \perp\!\!\!\perp_d \boldsymbol{Z} \mid \boldsymbol{C}, Y && (24) \\ &= P^{1,2}(y) \sum_{\boldsymbol{c}} P^*(c \mid y) \cdot P^{1,2}(\boldsymbol{z} \mid \boldsymbol{c}, y) \qquad S_1, S_2 \perp\!\!\!\perp_d Y && (25) \end{aligned}$$

Motivated by the above derivation, we use the source data drawn from P^1, P^2 and train the generative models $P(y; \eta_Y), P(\boldsymbol{z} \mid y, c; \eta_{\boldsymbol{Z}})$ to approximate sampling from the distributions $P^{1,2}(y), P^{1,2}(\boldsymbol{z} \mid y, c)$, respectively. The former generates a random digit Y according to the distribution of label in the source domain, and the latter generates a colored picture $\boldsymbol{Z}$ by taking color C and digit Y as the input. Also, we use an NCM with parameter θ_C^* to model the c-factor $P^*(c \mid do(y)) = P^*(c \mid y)$. We can now rewrite the risk as follows:

$$\begin{aligned} R_{P^*}(h) &= \sum_{\boldsymbol{z}, y} |y - h(\boldsymbol{z})| \cdot P^{1,2}(y) \sum_c P^*(c \mid y) \cdot P^{1,2}(\boldsymbol{z} \mid c, y) && (26) \\ &= \mathbb{E}_{Y \sim P(y;\eta_Y)}\big[\sum_c P(c \mid y; \theta_C^*) \cdot \mathbb{E}_{\boldsymbol{Z} \sim P(\boldsymbol{z} \mid c, y; \eta_{\boldsymbol{Z}})}[|Y - h(\boldsymbol{Z})|]\big]. && (27) \end{aligned}$$

By maximizing the above w.r.t. the free parameter θ_C^*, we achieve the worst-case risk of the classifier.

B.3 Reproducibility

For the synthetic experiments, we used feed-forward neural networks with 7 layers and 128×128 neurons in each layer. The activation for all layers is ReLu, but for the last layer which is a sigmoid since f_{θ_V} outputs the probability of $V = 1$. For evaluation, at each epoch, we used 1000 samples from the joint distribution. The data generative process for all experiments is provided in the corresponding

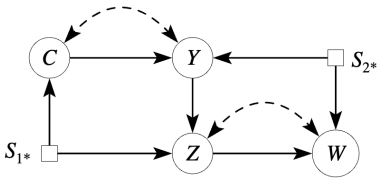


Figure 13: Selection diagram of Example 8

example. We used Adam optimizer for training the Neural networks. In CMNIST example, we used a standard implementation of a conditional GAN [23] trained over 200 epochs with a batch-size of 64. The learning rate of Adam was set to 0.0002. The architecture of the generator is given by a 5 layer feed-forward neural network with Batch normalization and Leaky-ReLu activations.

C Extended Discussion on Algorithms

In this section, we elaborate more on the algorithms presented in the paper.

C.1 Examples of Neural-TR (Algorithm 1)

In the next examples, we follow Algorithm 1 to compute the worst-case risk of a classifier.

Example 8 (Simplify)**.** Consider a system of SCMs $\mathcal{M}^1, \mathcal{M}^2\mathcal{M}^*$ over $\boldsymbol{X} = \{C, Z, W\}$ and Y that induces the selection diagram shown in Figure 13. Suppose we would like to assess the risk of a classifier $h(z)$. Following Theorem 2, the naive approach requires us to parameterize three NCMs $\theta^1, \theta^2, \theta^*$ over the variables $\boldsymbol{X}, Y$, and then proceed with the maximization of the target quantity $R_{P\mathcal{N}*}(h) = \mathbb{E}_{P*}[\mathbb{1}\{Y = h(Z)\}]$. Notably, the latter depends only on $P^*(y, z)$. We can rewrite the risk of h as follows:

$$\mathbb{E}_{Y,Z\sim P^*(y,z)}[\mathbb{1}\{Y \neq h(Z)\}] = \sum_{z,y} \mathbb{1}\{y \neq h(z)\} \cdot P^*(y, z) \tag{28}$$

$$= \sum_{z,y} \mathbb{1}\{y \neq h(z)\} \cdot P^*(y) \cdot P^*(z \mid y) \qquad \text{(factorization)} \tag{29}$$

$$= \sum_{z,y} \mathbb{1}\{y \neq h(z)\} \cdot P^*(y) \cdot P^2(z \mid y) \qquad (Y \mid Z \perp\!\!\!\perp_d S_2) \tag{30}$$

$$= \sum_{z,y} \mathbb{1}\{y \neq h(z)\} \cdot P^2(z \mid y) \cdot \sum_{c} P^*(y, c) \tag{31}$$

This new expression for the objective function depends only on the unknown $P^*(y, c)$, a so-called ancestral c-factor, that can generally be expressed as $P^*(\boldsymbol{a} \mid do(pa_{\boldsymbol{A}})), \boldsymbol{A} = \{C, Y\}$. In the following, we argue that to partially transport the risk we only need to parameterize the SCMs over ancestral c-factors that are not transportable. Specifically, the partial transportation problem can be restated as follows:

$$\max_{\theta^1,\theta^2,\theta*} \quad \mathbb{E}_{\boldsymbol{U}_{CY}}[\sum_{y,c,z} P(y, c \mid \boldsymbol{U}_{CY}; \theta^*) \cdot P(z \mid y; \eta^2) \cdot \mathbb{1}\{y \neq h(z)\}] \tag{32}$$

$$+ \Lambda \cdot \big(\sum_{y,c\in D^1} \mathbb{E}_{\boldsymbol{U}_{CY}}[\log P(y, c \mid U_{CY}; \theta^1)] + \sum_{y,c\in D^2} \mathbb{E}_{\boldsymbol{U}_{CY}}[\log P(y, c \mid U_{CY}; \theta^2)] \big)$$

$$\text{s.t. } \theta^*[C] = \theta^2[C], \quad \theta^*[Y] = \theta^1[Y].$$

In the above, $D^i \sim P^i(c, y, z, w)$ denotes the source data, and $P(z \mid y; \eta^2)$ is a probabilistic model of $P^2(z \mid y)$ learned using the data D^2. □

Example 9 (Partial-TR illustrated)**.** Consider a system of SCMs $\mathcal{M}^1, \mathcal{M}^2, \mathcal{M}^*$ over the binary variables $\boldsymbol{X} = \{X_1, X_2, \ldots, X_9\}$ and Y that induces the selection diagram shown in Figure 14. Consider the classifier $h(x_1, x_4) = x_1 \vee x_4$. The objective is partial transportation of the risk of h, expressed as follows:

$$R_{P*}(h) = P^*(Y \neq h(X_1, X_4)) \tag{33}$$

$$= \mathbb{E}_{X_1,X_4,Y\sim P*}[\mathbb{1}\{Y \neq X_1 \vee X_4\}]. \tag{34}$$

https://doi.org/10.52202/079017-4376

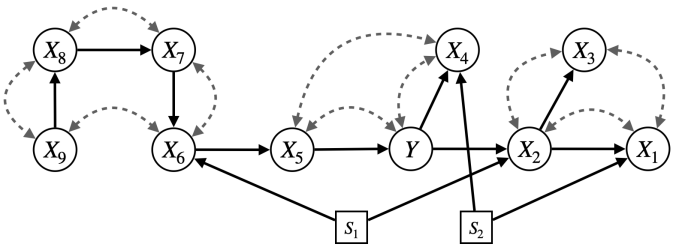


Figure 14: Selection diagram of Example 9

The latter indicates that $\psi(X_1, X_4, Y) := \mathbb{1}\{Y \neq X_1 \vee X_4\}$ must be passed to the algorithm. The objective function is then expressed as:

$$R_{P*}(h) = \mathbb{E}_{P*}[\psi(X_1, X_4, Y)] = \sum_{x_1, x_4, y} \mathbb{1}\{Y \neq X_1 \vee X_4\} \cdot P^*(x_1, x_4, y). \tag{35}$$

Next, we focus on transporting $P^*(x_1, x_4, y)$. First, we compute the ancestral set using the selection diagram;

$$\boldsymbol{A} = An(X_1, X_4, Y) = \{X_1, X_2, X_4, Y, X_5, X_6, X_7, X_8, X_9\}. \tag{36}$$

and we decompose this set into c-components:

$$\boldsymbol{A}_1 = \{X_1, X_2\}, \quad \boldsymbol{A}_2 = \{X_4, Y, X_5\}, \quad \boldsymbol{A}_3 = \{X_6, X_7, X_8, X_9\}. \tag{37}$$

Next, we form the expression below:

$$P^*(x_1, x_4, y) := \sum_{x_2, x_5, \ldots, x_9} P^*(x_1, x_2 \mid do(y)) \cdot P^*(y, x_4, x_5 \mid do(x_6)) \cdot P^*(x_6, \ldots, x_9). \tag{38}$$

Notice,

$$P^*(\boldsymbol{a}_2 \mid do(x_6)) \overset{\text{rule 2 do-calc.}}{=} P^*(\boldsymbol{a}_2 \mid x_6) \overset{S_1 \perp\!\!\!\perp_d Y, X_4, X_5 \mid X_6}{=} P^1(\boldsymbol{a}_2 \mid x_6), \tag{39}$$

$$P^*(\boldsymbol{a}_3) \overset{S_2 \perp\!\!\!\perp_d \boldsymbol{A}_3 \mid X_6}{=} P^2(\boldsymbol{a}_3). \tag{40}$$

Thus, we use the source data D^1, D^2 to learn the generative model $P(\boldsymbol{a}_2 \mid x_6; \eta^1_{\boldsymbol{A}_2}), P(\boldsymbol{a}_3; \eta^2_{\boldsymbol{A}_3})$ to approximate sampling from $P^1(\boldsymbol{a}_2 \mid x_6), P^2(\boldsymbol{a}_3)$ respectively. We plug these models as constants into Eq. 35.

Since S_{*1}, S_{*2} are pointing to the variables X_2, X_1, respectively, the first term $P^*(x_1, x_2) \mid do(y))$ in Eq. 38 can not be directly transported from neither of the source domains. Thus, we need to parameterize this c-factor using NCMs across all domains. We require the following properties:

1. **Parameter sharing**: Since $X_4, Y, X_5, X_7, X_8, X_9$ are not pointed by S_1, we share their mechanisms across all domains. Also, since X_2, X_6 are not pointed by S_2, we set $\theta^*_{\{X_2, X_6\}} = \theta^2_{\{X_2, X_6\}}$. These constraints are stored in $\mathbb{C}_{\text{expert}}$ in the Algorithm.

2. **Source data**: To enforce θ^1, θ^2 to be compatible with the source data D^1, D^2, we compute the likelihood of the data w.r.t. the parameters, as follows:

$$\mathcal{L}_{\text{likelihood}} := \sum_{i=1}^{2} \Big(\sum_{\langle x_1, x_2, y \rangle \in D^i} \mathbb{E}_{\boldsymbol{U}_{X_1, X_2}}[\log P(x_1, x_2 \mid y, \boldsymbol{U}_{X_1, X_2}; \theta^i_{X_2})] \tag{41}$$

We plug $P(x_1, x_2 \mid do(y); \theta^*_{\boldsymbol{A}_1})$ into Eq. 38. Finally, we use stochastic gradient ascent to maximize the objective function in Eq. 35 regularized by an additive term $\Lambda \cdot \mathcal{L}_{\text{likelihood}}$ that encourages the likelihood of the data w.r.t. the parameters of the source NCMs. □

C.2 Illustration of CRO (Algorithm 2)

First, we initialize with a random classifier. One may also warm start with a reasonable guess such as empirical risk minimizer defined as,

$$h_{\text{ERM}} \in \underset{h:\Omega_{\boldsymbol{X}}\to\Omega_{\boldsymbol{Y}}}{\arg\min} \sum_{i=1}^{K} \sum_{\boldsymbol{x},y\in D^i} \mathcal{L}(y, h(\boldsymbol{x})). \tag{42}$$

Throughout the runtime of the algorithm we accumulate instances of distributions that we obtain via Neural-TR (Alg. 1). At each step, these distribution are aimed to maximize the risk of the classifier at hand. In this sense, Neural-TR can be viewed as an adversary, and the CRO can be viewed as a game between two players:

1. **Neural-TR.** Searches over the spaces of plausible target domains that are characterized by the source data and the domain relatedness encoded in the selection diagram, to find a distribution that is hard to generalize to using the classifier at hand.
2. **group DRO [32]** Updates the classifier at hand by minimizing the maximum risk over the distributions produced by Neural-TR so far, that is,

$$\min_{h:\Omega_{\boldsymbol{X}}\to\Omega_{\boldsymbol{Y}}} \max_{D\in\mathbb{D}^*} \frac{1}{|D|} \cdot \sum_{\boldsymbol{x},y\in D} \mathcal{L}(y, h(\boldsymbol{x})). \tag{43}$$

For more information about group DRO, see Appendix D.2.

The equilibrium of the above happens if the worst-case risk obtained by Neural-TR almost coincides with the risk obtained by group DRO, i.e.,

$$R_{P(\boldsymbol{x},y;\hat{\theta})}(h) - \max_{D\in\mathbb{D}^*} \frac{1}{|D|} \cdot \sum_{\boldsymbol{x},y\in D} \mathcal{L}(y, h(\boldsymbol{x})) < \delta. \tag{44}$$

Once this is achieved, we stop the search and return the classifier at hand. When the game is not at equilibrium, we would have a difference larger than δ, meaning that the new target domain $\hat{\theta}^*$ has enough novelty to forces the classifier at hand to perform at least δ worse than what it achieves over the existing distributions in $\mathbb{D}^*$. Therefore, we draw samples $D^* \sim P(\boldsymbol{x}, y; \hat{\theta})$ and add them to our collection $\mathbb{D}^*$. As shown in Theorem 3 this game reaches the equilibrium in finitely many steps, and the classifier that we return has the best worst-case risk w.r.t. the selection diagram $\mathcal{G}^{\boldsymbol{\Delta}}$ and the source distributions $\mathbb{P}$. The conceptual Figure 15 shows the process of convergence of CRO.

It is important to note that although we employ group DRO as a subroutine in our CRO algorithm, we do not use the source distributions directly. Instead, we use group DRO on the distributions obtained from Neural-TR. Note that under the assumptions encoded in the selection diagram, the target distribution distribution may be geometrically unrelated to the source distributions; the reason is that mechanistic relatedness of the target domain to the source domains (as indicated by the graph) do not translate directly to closeness of the entailed distributions under known distributional distance measures.

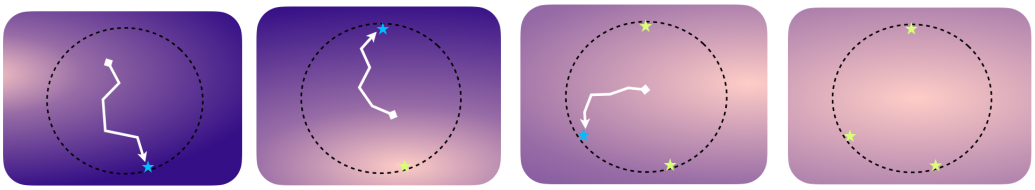
D Extended Related Work

In this section, we discuss some learning schemes based on invariant and robust learning that are proposed for domain generalization, including IRM and group DRO that are discussed in the experiments.

D.1 Invariant Learning for Domain Generalization

Several common invariance criteria are extensively studied in the literature and proposed for the domain generalization task. A prominent idea is label conditional distribution invariance that seeks a representation ϕ such that $P^i(Y \mid \phi(\boldsymbol{X}))$ is equal across the source domains [29, 1, 13, 22]. These notions do not explicitly rely on an underlying structural causal model (SCM), although invariances are often justified by an underlying causal model [27, 2, 39, 31, 33]. Jalaldoust & Bareinboim

 https://doi.org/10.52202/079017-4376



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Equilibrium

Figure 15: Conceptual illustration of CRO. The rectangle represents the space of all distributions over $\boldsymbol{X}, Y$, and the circle inside it represents the subset of that are plausible target distributions, as characterized by the source distributions and selection diagram. Iteration 1: At first we start with some classifier that may or may not perform well for all distributions in the plausible subset; the darker spots indicate distributions that yield higher risk for the classifier at hand. Neural-TR uses gradient ascend steps to find an SCM that entails a distribution which yields the highest risk for the classifier at hand, i.e., the darkest spot within the plausible subset (likely at the boundary of it), shown by the star blue in Fig. (a). We register this distribution by taking samples from it and adding them to the collection $\mathbb{D}^*$. Iteration 2: We update the classifier at hand to have group robustness to the collection of distributions $\mathbb{D}^*$; in this case, only risk minimizer, since there is only one distribution in the collection. Now the distributions that are *close* to the registered distribution would entail small risk, thus, the region around the first star is now brighter. Once again, using Neural-TR we find a distribution that yields high risk for the classifier at hand. Iteration 3: We update the classifier, this time to minimize the risk on both registered distributions indicated with yellow starts using group DRO. Now the risk is smaller in most parts of the plausible set, though Neural-TR still finds another distribution at the boundary with high risk. Equilibrium: We update the classifier using group DRO over the three registered distributions. This time, the registered distributions correctly represent the plausible set, meaning that the maximum risk inside the plausible set is not significantly larger than what is achieved at the registered points through group DRO.

[17] studied the implicit assumptions that license generalizability of representations that satisfy the probabilistic relation $P^i(Y \mid \phi(\boldsymbol{X}))$. Although searching for such representation is practically challenging and in cases theoretically intractable. Thus, one may resort to achieving an approximate notion that serve as a proxy to invariance of $P^i(Y \mid \phi(\boldsymbol{X}))$; A well-known instance of such effort is invariant risk minimization [2], discussed below.

The paper [2] studies a constrained optimization problem called invariant risk minimization (IRM) in the context of domain generalization. In the notation of our paper, the IRM problem can be written as follows:

$$\begin{aligned} \min_{\phi,h} \quad & \sum_{i=1}^{K} \mathbb{E}_{P^i}[Y \neq h \circ \phi(\boldsymbol{X})] \\ \text{s.t.} \quad & h \in \operatorname*{arg\,min}_{\tilde{h}:\Omega_{\boldsymbol{R}} \to \{0,1\}} \mathbb{E}_{P^i}[Y \neq \tilde{h} \circ \phi(\boldsymbol{X})] \quad \forall i, \end{aligned} \tag{45}$$

Where $\phi : \Omega_{\boldsymbol{X}} \to \Omega_{\boldsymbol{R}}$ is a representation, and $h : \Omega_{\boldsymbol{R}} \to \{0, 1\}$ is a classifier defined based on it. In words, a pair h, ϕ satisfies the invariant risk minimization property if $h \circ \phi$ attains the minimum risk across all classifiers defined based on ϕ, across all source domains. The search procedure suggests choosing the classifier that satisfies the mentioned constraint, and achieves minimum risk on the pooled source data. The constrained optimization program above is highly non-convex and hard to solve in practice. To approximate the solution, the paper considers the Langrangian form below:

$$h_{\text{IRM}} \in \min_{h_\theta:\Omega_{\boldsymbol{X}} \to \{0,1\}} \quad \sum_{i=1}^{K} \mathbb{E}_{P^i}[Y \neq h_\theta(\boldsymbol{X})] + \lambda \cdot \|\nabla_\theta \mathbb{E}_{P^i}[Y \neq h_\theta(\boldsymbol{X})]\|^2. \tag{46}$$

In this program, θ parametrizes the classifier h, and the penalty term λ accounts for how restrictive one wants to enforce the IRM constraint. In the extreme $\lambda = 0$ the objective equates to the vanilla ERM with all data pooled; on the other extreme, for $\lambda \to \infty$ ascertains that the solution is guaranteed to satisfy the IRM constraint.

Consider a representation that satisfies the original IRM constraint in Eq. 45. The optimal classifier defined over this representation is the bayes classifier, that uses $\frac{1}{2}$ level set of $P^i(Y = 1 \mid \phi(\boldsymbol{X}))$ as the decision boundary. This means that satisfying the IRM constraint implies a match between $\frac{1}{2}$ level-sets of $P^i(Y = 1 \mid \phi(\boldsymbol{X}))$ across all source domains. On the other hand, invariance of

$P^i(Y \mid \phi(\boldsymbol{X}))$ requires coincidence of every level-set across the source domains, and in this sense, the IRM constraint can be viewed as a proxy to the invariance property of $P^i(Y \mid \phi(\boldsymbol{X}))$. One can speculate that since IRM yields a proxy to invariance of $P^i(Y \mid \phi(\boldsymbol{X}))$, it might still exhibit generalization, though slightly weaker than what is derived from invariance of $P^i(Y \mid \phi(\boldsymbol{X}))$. However, IRM is shown to have poor domain generalizability, both theoretically (e.g., [30]) and empirically (e.g., [15]). Still, due to popularity of this method in the literature, we find it insightful to use the Neural-TR algorithm to find out what would be the worst-case risk of IRM. As shown in Fig. 4c, the worst-case performance of IRM is much worse than what is reported by [2] and [15]; the reason is that Neural-TR does not commit to one held-out domain, and instead it constructs an SCMs that is tailored to yield the poorest performance subject to the graph and source distributions.

D.2 Group Robustness for Domain Generalization

Group Distributionally robust optimization (group DRO) [32] has been employed in the broad context of learning under uncertainty. In group DRO one seeks a single classifier that minimizes the risk on multiple distributions simultaneously. More specifically, the objective is minimizing the maximum risk among the source distributions, i.e.,

$$h_{\text{DRO}} \in \arg\min \max_{i \in \{1,2,\dots,K\}} R_{P^i}(h) \tag{47}$$

This approach ensures that the learned classifier is optimal w.r.t. an unknown target domain that lies in the convex hull of the source distributions. In this sense, group DRO objective interpolates the perturbations that are represented in the source data to define an uncertainty set for the target distribution. On the other hand, in invariant learning the objective is to extrapolate the perturbations that are observed among the source domain by learning a representation that shields the label from these changes. In particular, [18, 31, 33] highlight the invariant-robust spectrum, and propose methods that have a free parameter which allows interpolating the two. In our experiments, we considered group DRO as a representative of methods in this category, and evaluated its worst-case performance in the Colored MNIST task, as shown in Figure 4c. Once again, we emphasize that this worst-case risk is much larger than what is shown in the benchmarks, e.g., by [15]. The reason is that the worst-case performance is obtained by Neural-TR that operates as an adversary, seeking a plausible target domain that is hardest to generalize to, subject to the assumptions encoded in the graph and the source data.

E Proofs

Proof of Theorem 1

Our results rely on the expressiveness of discrete SCMs, i.e. defined over variables $\{\boldsymbol{V}, \boldsymbol{U}\}$ with finite cardinalities. Discrete SCMs, introduced first in [3] and then in [42] have been shown to be "canonical" in the sense that they could represent all counterfactual distributions entailed by any SCM with the same induced causal diagram defined over finite $\boldsymbol{V}$. The following example illustrates this observation.

Example 10 (The double bow)**.** Let $\{X, Y, Z\}$ be binary variables. Consider two source domains defined based on the following SCMs:

$$\mathcal{M}^1 : \begin{cases} P^1(\boldsymbol{U}) : \begin{cases} U_X \sim \text{Normal}(0,1) \\ U_{XY} \sim \text{Normal}(0,1) \\ U_{ZY} \sim \text{Normal}(0,1) \end{cases} \\ \mathcal{F}^1 : \begin{cases} X \leftarrow \mathbb{1}\{U_X + U_{XY} > 0\} \\ Y \leftarrow \mathbb{1}\{X - U_{XY} > 0\} \\ Z \leftarrow \mathbb{1}\{Y \cdot U_{ZY} > 0\} \end{cases} \end{cases} \qquad \mathcal{M}^* : \begin{cases} P^*(\boldsymbol{U}) : \begin{cases} U_X \sim \text{Normal}(0,1) \\ U_{XY} \sim \text{Normal}(0,1) \\ U_{ZY} \sim \text{Normal}(0,1) \end{cases} \\ \mathcal{F}^* : \begin{cases} X \leftarrow \mathbb{1}\{U_X + U_{XY} > 0\} \\ Y \leftarrow \mathbb{1}\{-U_{XY} + 0.5 > 0\} \\ Z \leftarrow \mathbb{1}\{Y \cdot U_{ZY} > 0\} \end{cases} \end{cases}$$

The SCM M^1 induces a counterfactual probabilities, e.g. $P^{M^1}(x, y_x, z_y)$ for outcomes $x, y_x, z_y \in \{0, 1\}$. [3] observed that such probabilities, defined over a finite set of events, may be generated with an equivalent model with a potentially large but finite set of discrete exogenous variables. [3] derived a canonical parameterization for the SCMs that induces the same graph but instead involves possibly correlated discrete latent variables R_X, R_Y, R_Z, where R_X determines the functional that decides

https://doi.org/10.52202/079017-4376

X, R_Y determines the functional that decides Y based on X, and R_Z determines the functional that decides Z based on Y. [3] showed that for every SCM $\mathcal{M}$ with the same induced graph as $\mathcal{M}^1$ there exists an SCM of the described format specified with only a distribution $P(r_X, r_Y, r_Z)$, where,

$$\begin{aligned}\text{supp}_{R_X} &= \{0, 1\},\\ \text{supp}_{R_Y} &= \{y = 0, y = 1, y = x, y = \neg x\},\\ \text{supp}_{R_Z} &= \{z = 0, z = 1, z = y, z = \neg y\}.\end{aligned}$$

Thus, the joint distribution $P(r_X, r_Y, r_Z)$ can be parameterized by an 32-dimensional vector. □

This example illustrates a more general procedure, in which probabilities induced by an SCM over discrete endogenous variables $\boldsymbol{V}$ may be generated by a canonical model. This is formalized in the following lemma.

Definition 7 (Canonical SCM)**.** A canonical SCM is an SCM $\mathcal{N} = \langle \boldsymbol{U}, \boldsymbol{V}, \mathcal{F}, P(\boldsymbol{U})\rangle$ defined as follows. The set of endogenous variables $\boldsymbol{V}$ is discrete. The set of exogenous variables $\boldsymbol{U} = \{R_V : V \in \boldsymbol{V}\}$, where $\text{supp}_{R_V} = \{1, \ldots, m_V\}$ (where $m_V = |\{h_V : \text{supp}_{pa_V} \to \text{supp}_V\}|$) for each $V \in \boldsymbol{V}$. For each $V \in \boldsymbol{V}$, $f_V \in \mathcal{F}$ is defined as $f_V(pa_V, r_V) = h_V^{(r_V)}(pa_V)$.

The following lemma establishes the expressiveness of canonical SCMs.

Lemma 1 (Thm. 2.4 [42])**.** *For an arbitrary SCM $M = \langle \boldsymbol{U}, \boldsymbol{V}, \mathcal{F}, P(\boldsymbol{U})\rangle$, there exists a canonical SCM $\mathcal{N}$ such that 1. M and $\mathcal{N}$ are associated with the same causal diagram, i.e., $\mathcal{G}_M = \mathcal{G}_\mathcal{N}$. 2. For any set of counterfactual variables $\boldsymbol{Y}_{\boldsymbol{x}}, \ldots, \boldsymbol{Z}_{\boldsymbol{w}}$, $P^M(\boldsymbol{Y}_{\boldsymbol{x}}, \ldots, \boldsymbol{Z}_{\boldsymbol{w}}) = P^\mathcal{N}(\boldsymbol{Y}_{\boldsymbol{x}}, \ldots, \boldsymbol{Z}_{\boldsymbol{w}})$.*

In words, finite exogenous domains in canonical SCMs are sufficient for capturing all the uncertainties and randomness introduced by the (potentially) continuous latent variables in SCMs. Our goal will be to adapt the canonical parameterization of SCMs such that they entail the equality constraints specified by $\mathcal{G}^{\boldsymbol{\Delta}}$. The next example illustrates the implication of the constraints induced by $\mathcal{G}^{\boldsymbol{\Delta}}$ on the construction of canonical SCMs.

Example 11 (Example 10 continued.)**.** Consider $\mathcal{M}^1$ and $\mathcal{M}^*$ given in Example 10. The domain discrepancy set $\boldsymbol{\Delta}$ indicates that certain causal mechanisms need to match across pairs of the SCMs. For example, $\Delta_{1*} = \{Y\}$, which does not contain $\{X, Z\}$, and this implies that the functions f_X, f_Z are invariant across $\mathcal{M}^1, \mathcal{M}^*$, and that the distribution of unobserved variables that are arguments of f_Y, f_Z, namely, $\{U_X, U_{XY}, U_{YZ}\}$ are invariant across $\mathcal{M}^1, \mathcal{M}^*$. The canonical parameterization of $\mathcal{M}^1$ is given by

$$\mathcal{N}^1 = \begin{cases} \boldsymbol{V} & = \{X, Y\} \\ \boldsymbol{U} & = \{R_X, R_Y, R_Z\} \\ \mathcal{F}^1 & = \begin{cases} f_X^1 : \text{supp}_{R_X} \to \text{supp}_X \\ f_Y^1 : \text{supp}_{R_Y} \times \text{supp}_X \to \text{supp}_Y \\ f_Z^1 : \text{supp}_{R_Z} \times \text{supp}_Y \to \text{supp}_Z \end{cases} \\ P^1(\boldsymbol{U}) & = P^1(R_X, R_Y, R_Z) \end{cases}$$

Analogously, the canonical parameterization of $\mathcal{M}^*$ is given by

$$\mathcal{N}^1 = \begin{cases} \boldsymbol{V} & = \{X, Y\} \\ \boldsymbol{U} & = \{R_X, R_Y, R_Z\} \\ \mathcal{F}^* & = \begin{cases} f_X^* : \text{supp}_{R_X} \to \text{supp}_X \\ f_Y^* : \text{supp}_{R_Y} \times \text{supp}_X \to \text{supp}_Y \\ f_Z^* : \text{supp}_{R_Z} \times \text{supp}_Y \to \text{supp}_Z \end{cases} \\ P^*(\boldsymbol{U}) & = P^*(R_X, R_Y, R_Z) \end{cases}$$

With these definitions, the restrictions in $\boldsymbol{\Delta}_{1*}$ impose straightforward constraints on the parameterization of the canonical models given directly from the definition of discrepancy set:

$$\begin{aligned} f_X^1(r_X) = f_X^*(r_X), P^1(r_X) = P^*(r_X), \quad X \notin \boldsymbol{\Delta}_{1*} \\ f_Z^1(y, r_Z) = f_Z^*(y, r_Z), P^1(r_Z) = P^*(r_Z), \quad Z \notin \boldsymbol{\Delta}_{1*} \end{aligned}$$

for any input x, y, r_Y, r_X, r_Z. □

The next lemma formalizes the observation made in the example above, showing that if a pair SCMs and a pair of associated canonical models induce the same distributions and causal diagram, their discrepancies must also agree.

Lemma 2. *For a pair of SCMs M^i, M^j ($i, j \in \{*, 1, 2, \dots, T\}$) defined over $\boldsymbol{V}$ with discrepancy set $\Delta_{ij} \subseteq \boldsymbol{V}$, let $\mathcal{N}^i, \mathcal{N}^j$ be associated canonical SCMs that induce the same causal graphs and entail the same distributions over $\boldsymbol{V}$. Then the discrepancy sets of the pairs of SCMs and canonical SCMs must agree, i.e. $V \in \Delta_{ij}$ if and only if either $f_V^{N^i} \neq f_V^{N^j}$, or $P^{N^i}(u_V) \neq P^{N^j}(u_V)$.*

Proof. Let $V \in \Delta_{ij}$, and fix M^i, M^j such that $P^{M^i}(v \mid do(pa_V)) \neq P^{M^j}(v \mid do(pa_V))$. This is possible since the interventional probabilities are parameterized by the mechanism of V which could vary across M^i, M^j. Assume for a contradiction that $f_V^{N^i} = f_V^{N^j}$ and $P^{N^i}(u_V) = P^{N^j}(u_V)$ for two canonical models N^i, N^j constructed to match all L_3 statements induced by M^i, M^j. This implies in particular that $P^{\mathcal{N}^i}(v \mid do(pa_V)) = P^{\mathcal{N}^j}(v \mid do(pa_V))$ and therefore $\mathcal{N}^i, \mathcal{N}^j$ do not induce the same probabilities as M^i, M^j. This contradicts the assumption that the pair of canonical SCMs matches the pair of SCMs in all L_3 statements.

For the converse, we proceed similarly. For fixed M^i, M^j, assume for a contradiction that $f_V^{N^i} \neq f_V^{N^j}$, or $P^{N^i}(u_V) \neq P^{N^j}(u_V)$ such that $P^{\mathcal{N}^i}(v \mid do(pa_V)) \neq P^{\mathcal{N}^j}(v \mid do(pa_V))$ for two canonical models N^i, N^j constructed to match all L_3 statements induced by M^i, M^j, but nevertheless $V \notin \Delta_{ij}$. The discrepancy set ensures that $P^{M^i}(v \mid do(pa_V)) = P^{M^i}(v \mid do(pa_V))$ but the same relation is not true for N^i, N^j as $P^{\mathcal{N}^i}(v \mid do(pa_V)) \neq P^{\mathcal{N}^j}(v \mid do(pa_V))$ by assumption and therefore $\mathcal{N}^i, \mathcal{N}^j$ do not induce the same probabilities as M^i, M^j. This contradicts the assumption that the pair of canonical SCMs matches the pair of SCMs in all L_3 statements. □

Lemma 3. *Consider a system of multiple SCMs $\mathbb{M} : \{\mathcal{M}^1, \mathcal{M}^2, \dots, \mathcal{M}^K, \mathcal{M}^*\}$ that induces a selection diagram and entails the source distributions $\mathbb{P} : \{P^1, P^2, \dots, P^K, P^*\}$ over the variables $\boldsymbol{V}$. Then there exists a system of canonical SCM $\mathbb{N} : \{\mathcal{N}^1, \mathcal{N}^2, \dots, \mathcal{N}^K, \mathcal{N}^*\}$ such that*

1. *$\mathbb{M}$ and $\mathbb{N}$ are associated with the same set of causal diagrams and selection diagrams.*
2. *For any set of counterfactual variables $\boldsymbol{Y_x}, \dots, \boldsymbol{Z_w}$, $P^{M^*}(\boldsymbol{Y_x}, \dots, \boldsymbol{Z_w}) = P^{\mathcal{N}^*}(\boldsymbol{Y_x}, \dots, \boldsymbol{Z_w})$.*

Proof. For (1), Thm. 2.4 [42] gives that SCMs $\mathbb{M} : \{\mathcal{M}^1, \mathcal{M}^2, \dots, \mathcal{M}^K, \mathcal{M}^*\}$ and canonical SCMs $\mathbb{N} : \{\mathcal{N}^1, \mathcal{N}^2, \dots, \mathcal{N}^K, \mathcal{N}^*\}$ induce the same causal diagrams. Lem. 2 gives that for every pair of SCMs M^i, M^j ($i, j \in \{*, 1, 2, \dots, T\}$), their discrepancy set is the same as that of $\mathcal{N}^i, \mathcal{N}^j$ ($i, j \in \{*, 1, 2, \dots, T\}$). As selection diagrams are constructed deterministically from causal diagrams and discrepancy sets, $\mathbb{M}$ and $\mathbb{N}$ must share the same set of selection diagrams.

(2) is given by Thm. 2.4 [42]. □

Theorem 1 (restated). *Consider a system of multiple SCMs $\mathcal{M}^1, \mathcal{M}^2, \dots, \mathcal{M}^K, \mathcal{M}^*$ that induces the selection diagram $\mathcal{G}^{\boldsymbol{\Delta}}$ and entails the source distributions $P^1, P^2, \dots, P^K$ and the target distribution P^* over the variables $\boldsymbol{V}$. Let $\psi(P^*) \in [0, 1]$ be the target quantity. Consider the following optimization scheme:*

$$\hat{q}_{\max} = \max_{\mathcal{N}^1, \mathcal{N}^2, \dots, \mathcal{N}^*} \psi(P^{\mathcal{N}^*}) \tag{48}$$
$$\text{s.t. } P^{\mathcal{N}^i}(r_V) = P^{\mathcal{N}^j}(r_V), \qquad \forall i, j \in \{1, 2, \dots, K, *\} \quad \forall V \notin \Delta_{i,j}$$
$$P^{\mathcal{N}^i}(\boldsymbol{v}) = P^i(\boldsymbol{v}) \qquad \forall i \in \{1, 2, \dots, K, *\},$$

where each $\mathcal{N}^i$ is a canonical model characterized by a joint distribution over $\{R_V\}_{V \in \boldsymbol{V}}$. The value of the above optimization, namely $\hat{q}_{\max}$, is a tight upper-bound for the quantity $\psi(P^*)$ among all tuples of SCMs that induce the selection diagram and entail the source distributions at hand. □

 https://doi.org/10.52202/079017-4376

Proof. Note that,

$$\hat{q}_{\max} = \max_{\mathcal{M}^1,\mathcal{M}^2,\ldots,\mathcal{M}^*} \psi(P^{\mathcal{M}^*}) \tag{49}$$
$$\text{s.t. } P^{\mathcal{M}^i}(\boldsymbol{u}_V) = P^{\mathcal{M}^j}(\boldsymbol{u}_V), f_V^{\mathcal{M}^i} = f_V^{\mathcal{M}^j}, \quad \forall i,j \in \{1,2,\ldots,K,*\} \quad \forall V \notin \Delta_{i,j}$$
$$P^{\mathcal{M}^i}(\boldsymbol{v}) = P^i(\boldsymbol{v}) \qquad \forall i \in \{1,2,\ldots,K,*\},$$

is a tight upper bound to the target $\psi(P^{\mathcal{M}^*})$ among all tuples of SCMs that induce the selection diagram and entail the source distributions at hand, by construction. It follows from Lem. 3 that for any tuple of SCMs $\{\mathcal{M}^1,\mathcal{M}^2,\ldots,\mathcal{M}^K,\mathcal{M}^*\}$, that induce the selection diagram and entail the source distributions, there exists a tuple of canonical SCMs $\mathcal{N}^1,\mathcal{N}^2,\ldots,\mathcal{N}^*$, that induce the selection diagram and entail the source distributions such that,

$$P^{M^*}(\boldsymbol{Y}_{\boldsymbol{x}},\ldots,\boldsymbol{Z}_{\boldsymbol{w}}) = P^{\mathcal{N}^*}(\boldsymbol{Y}_{\boldsymbol{x}},\ldots,\boldsymbol{Z}_{\boldsymbol{w}}).$$

The reverse direction of the above equations also holds since a a family of canonical SCMs is an instance of a family of SCMs. This means that solutions for optimization problems in Eq. (48) and Eq. (49) must coincide. □

E.1 Proof of Theorem 2

To prove this result, we need to show the following:

1. **Necessity.** Every tuple of NCMs Θ that are constraint by conditions in Eq. 8 represents a tuple of SCMs that entails $\mathbb{P}$ and induces $\mathcal{G}^{\boldsymbol{\Delta}}$.
2. **Sufficiency.** For every tuple of SCMs $\mathbb{M}$ that entails $\mathbb{P}$ and induces $\mathcal{G}^{\boldsymbol{\Delta}}$, there exists a tuple of NCMs Θ that admits the constraints in Eq. 8, and for every $i \in \{*,1,2,\ldots,K\}$, we have $P(\boldsymbol{y}_{\boldsymbol{x}},\boldsymbol{z}_{\boldsymbol{w}};\theta^i) = P^{\mathcal{M}^i}(\boldsymbol{y}_{\boldsymbol{x}},\boldsymbol{z}_{\boldsymbol{w}})$, where $\boldsymbol{y}_{\boldsymbol{x}},\boldsymbol{z}_{\boldsymbol{w}}$.

Necessity. Consider a tuple of NCMs Θ that are constraint by the conditions in Eq. 8.

- **$\mathcal{G}^{\boldsymbol{\Delta}}$-consistency.** Since these NCMs are constructed based on the common causal diagram $\mathcal{G}$, they all induce $\mathcal{G}$ (Theorem 2 by Xia et al. [41]). Moreover, the parameter sharing constraint states that $V \notin \Delta_{ij}$ if and only if $\theta_V^i = \theta_V^j$. This implies that the NCMs parameterized by Θ induce the same domain discrepancy sets as $\mathcal{G}^{\boldsymbol{\Delta}}$. Thus, the selection diagram induced by the NCMs parameterized by Θ is exactly $\mathcal{G}^{\boldsymbol{\Delta}}$.
- **$\mathbb{P}$-expressivity.** The data likelihood condition for source distribution $P^i(\boldsymbol{v})$ states the following:
$$\theta^i \in \underset{\mathcal{G}-\text{constrained } \theta}{\arg\max} \sum_{\boldsymbol{v}\in D^i} \log P(\boldsymbol{v};\theta). \tag{50}$$
For large enough samples size $|D^i| \sim P^i(\boldsymbol{v})$, and enough model complexity in θ, Theorem 1 by Xia et al. [41] shows that there exists a $\mathcal{G}$-constrained NCM θ that induces the distribution entailed by the true SCM $\mathcal{M}^i$. Thus, by imposing Eq. (50) we assure that $P(\boldsymbol{v};\theta^i) = P^i(\boldsymbol{v})$. By imposing all data likelihood conditions, in the limit of sample size and model complexity, we ensure that the source NCMs induce the source distributions.

In conclusion, the tuple of NCMs are necessarily representing a plausible target domain since (1) they induce $\mathcal{G}^{\boldsymbol{\Delta}}$ and (2) they entail $\mathbb{P}$.

Sufficiency. Consider a tuple of SCMs $\mathbb{M} = \langle \mathcal{M}^1,\mathcal{M}^2,\ldots,\mathcal{M}^K,\mathcal{M}^*\rangle$ that induce $\mathcal{G}^{\boldsymbol{\Delta}}$ and entail $\mathbb{P}$. Theorem 1 by Xia et al. [41] shows that for every SCM $\mathcal{M}$ that induces $\mathcal{G}$, there exists a $\mathcal{G}$-constraint NCM parameterized by θ such that $P^{\mathcal{M}}(\boldsymbol{v}) = P(\boldsymbol{v};\theta)$ (as a consequence of L3-consistency). The proof is constructive, and for every $V \in \boldsymbol{V}$ the construction of the neural network θ_V depends on (1) the function f_V and (2) the distribution $P^{\mathcal{M}}(\boldsymbol{u}_V)$.

Consider two SCMs $\mathcal{M}^i,\mathcal{M}^j$ $(i,j \in \{*,1,2,\ldots,K\})$ that induce domain discrepancy set Δ_{ij}. Follow the construction by Xia et al. [41] to obtain the corresponding NCMs parameterized by θ^i,θ^j. For every $V \notin \Delta_{i,j}$, we have, $\theta_V^i = \theta_V^j$ since the construction depends on $f_V^i = f_V^j$ and $P^{\mathcal{M}^i}(\boldsymbol{u}_V) = P^{\mathcal{M}^j}(\boldsymbol{u}_V)$. Thus, the domain discrepancy set induced by θ^i,θ^j matches with $\Delta_{i,j}$ induced by the

SCMs $\mathcal{M}^i, \mathcal{M}^j$. Therefore, By constructing the NCM θ^i from $\mathcal{M}^i$ ($i \in \{*, 1, 2, \dots, K\}$), we are guaranteed to have a tuple of NCMs $\mathbb{N}$ that (1) induce $\mathcal{G}^{\boldsymbol{\Delta}}$ and (2) entails $\mathbb{P}$.

Partial-TR via NCMs. Due to necessity and sufficiency above, we conclude that a tuples of NCMs satisfies the parameter sharing and data likelihood conditions stated in Eq. 8, if and only if there exists a tuple of SCMs $\mathbb{M}$ that induce $\mathcal{G}^{\boldsymbol{\Delta}}$ and entail $\mathbb{P}$ such that $P(\boldsymbol{v}; \theta^i) = P^{\mathcal{M}^i}(\boldsymbol{v})$ for all $i \in \{*, 1, 2, \dots, K\}$. Therefore, by solving the following optimization problem,

$$\begin{aligned}\hat{\Theta} \in \underset{\Theta:\langle\theta^1,\theta^2,\dots,\theta^K,\theta*\rangle}{\arg\max} & \sum_{\boldsymbol{w}} \psi(\boldsymbol{w}) \cdot \sum_{\boldsymbol{v}\setminus\boldsymbol{w}} P(\boldsymbol{v}; \theta^*) \\ \text{s.t. } & \theta^i_V = \theta^j_V, \qquad \forall i, j \in \{1, 2, \dots, K, *\} \quad \forall V \notin \Delta_{i,j} \\ & \theta^i \in \arg\max_{\theta} \sum_{\boldsymbol{v} \in D^i} \log P(\boldsymbol{v}; \theta), \quad \forall i \in \{1, 2, \dots, K\}.\end{aligned} \tag{51}$$

we achieve a tight upper-bound for the query $\mathbb{E}_{P*}[\psi(\boldsymbol{W})]$ w.r.t. $\mathcal{G}^{\boldsymbol{\Delta}}, \mathbb{P}$. □

E.2 Proof of Proposition 1

Consider the objective of Theorem 2;

$$\begin{aligned}\hat{\Theta} \in \underset{\Theta:\langle\theta^1,\theta^2,\dots,\theta^K,\theta*\rangle}{\arg\max} & \sum_{\boldsymbol{w}} \psi(\boldsymbol{w}) \cdot \sum_{\boldsymbol{v}\setminus\boldsymbol{w}} P(\boldsymbol{v}; \theta^*) \\ \text{s.t. } & \theta^i_V = \theta^j_V, \qquad \forall i, j \in \{1, 2, \dots, K, *\} \quad \forall V \notin \Delta_{i,j} \\ & \theta^i \in \arg\max_{\theta} \sum_{\boldsymbol{v} \in D^i} \log P(\boldsymbol{v}; \theta), \quad \forall i \in \{1, 2, \dots, K\}.\end{aligned} \tag{52}$$

No need to parameterize non-ancestors of $\boldsymbol{W}$. Let $\boldsymbol{T} = \boldsymbol{V} \setminus An_{\mathcal{G}*}(\boldsymbol{W})$. By applying Rule 3 of σ-calculus [9] we realize that,

$$P(\boldsymbol{w}; \theta^*_{\boldsymbol{V}\setminus\boldsymbol{T}}, \theta^*_{\boldsymbol{T}}) = P(\boldsymbol{w}; \theta^*_{\boldsymbol{V}\setminus\boldsymbol{T}}, \tilde{\theta}_{\boldsymbol{T}}). \tag{53}$$

The latter indicates that the parameters $\{\theta^*_T\}_{T\in\boldsymbol{T}}$ are irrelevant to the joint distribution $P(\boldsymbol{w})$, and therefore, can be dropped from the NCMs used for partial transportability of $\mathbb{E}_{P*}[\psi(\boldsymbol{W})] = \sum_{\boldsymbol{w}} P^*(\boldsymbol{w}) \cdot \psi(\boldsymbol{w})$.

Let $\boldsymbol{A} = An_{G*}(\boldsymbol{W})$. We drop the non-ancestors, and rewrite the objective as follows:

$$\begin{aligned}\hat{\Theta_{\boldsymbol{A}}} \in \underset{\Theta_{\boldsymbol{A}}:\langle\theta^1_{\boldsymbol{A}},\theta^2_{\boldsymbol{A}},\dots,\theta^K_{\boldsymbol{A}},\theta^*_{\boldsymbol{A}}\rangle}{\arg\max} & \sum_{\boldsymbol{w}} \psi(\boldsymbol{w}) \cdot \sum_{\boldsymbol{a}\setminus\boldsymbol{w}} P(\boldsymbol{a}; \theta^*) \\ \text{s.t. } & \theta^i_V = \theta^j_V, \qquad \forall i, j \in \{1, 2, \dots, K, *\} \quad \forall V \notin \Delta_{i,j} \\ & \theta^i_{\boldsymbol{A}} \in \arg\max_{\theta} \sum_{\boldsymbol{a} \in D^i} \log P(\boldsymbol{a}; \theta_{\boldsymbol{A}}), \quad \forall i \in \{1, 2, \dots, K\}.\end{aligned} \tag{54}$$

Next, we add the likelihood terms to the main objective regularized by a coefficient Λ to achieve a single-objective optimization.

$$\begin{aligned}\hat{\Theta_{\boldsymbol{A}}} \in \underset{\Theta_{\boldsymbol{A}}:\langle\theta^1_{\boldsymbol{A}},\theta^2,\dots,\theta^K,\theta*\rangle}{\arg\max} & \sum_{\boldsymbol{w}} \psi(\boldsymbol{w}) \cdot \sum_{\boldsymbol{a}\setminus\boldsymbol{w}} P(\boldsymbol{a}; \theta^*) + \Lambda \cdot \sum_{i=1}^{K} \sum_{\boldsymbol{a}\in D^i} \log P(\boldsymbol{a}; \theta^i_{\boldsymbol{A}}) \\ \text{s.t. } & \theta^i_V = \theta^j_V, \qquad \forall i, j \in \{1, 2, \dots, K, *\} \quad \forall V \notin \Delta_{i,j}\end{aligned} \tag{55}$$

https://doi.org/10.52202/079017-4376

For $\Lambda \to \infty$, the new optimization problem matches with that of Thm. 2. Now, we focus on the likelihood expression, and rewrite it following a causal order of $\mathcal{G}^*$, namely, $A_1 \prec A_2 \prec \cdots \prec A_N$.

$$\log P(\boldsymbol{a}; \theta^i_{\boldsymbol{A}}) = \sum_{l=1}^{N} \log P(a_l \mid a_{l-1}, \ldots, a_1; \theta^i_{\boldsymbol{A}}) \qquad \text{(factorization)} \tag{56}$$

$$= \sum_{l=1}^{N} \log \mathbb{E}_{\boldsymbol{U}_{\boldsymbol{A}}}[P(a_l \mid v_{l-1}, \ldots, v_1, \boldsymbol{U}; \theta^i_{\boldsymbol{A}})] \qquad \text{(conditioning on } \boldsymbol{U}) \tag{57}$$

$$= \sum_{l=1}^{N} \log \mathbb{E}_{\boldsymbol{U}_{A_l}}[P(a_l \mid pa_{A_l}, \boldsymbol{U}_{A_l}; \theta^i)] \qquad \text{(Rule 1 of do-calc)} \tag{58}$$

$$= \sum_{l=1}^{N} \log \mathbb{E}_{\boldsymbol{U}_{A_l}}[P(a_l \mid pa_{A_l}, \boldsymbol{U}_{A_l}; \theta^i_A)] \qquad \text{(Rule 3 of do-calc)} \tag{59}$$

Let $\{\boldsymbol{A}_j\}_{j=1}^m$ be the c-components of $\mathcal{G}^*_{[\boldsymbol{A}]}$, which is the graph induced by nodes $\boldsymbol{A}$. We rewrite the above objective in terms of the c-factors:

$$\log P(\boldsymbol{a}; \theta^i_{\boldsymbol{A}}) = \sum_{j=1}^{m} \sum_{A \in \boldsymbol{A}_j} \log \mathbb{E}_{\boldsymbol{U}_A}[P(a \mid pa_A, \boldsymbol{U}_A; \theta^i_A)] \qquad \text{(c-factor decomp.)} \tag{60}$$

$$= \sum_{j=1}^{m} \log \prod_{A \in \boldsymbol{A}_j} \mathbb{E}_{\boldsymbol{U}_A}[P(a \mid pa_A, \boldsymbol{U}_A; \theta^i_A)] \qquad \text{(sum-of-log to log-of-prod)} \tag{61}$$

$$= \sum_{j=1}^{m} \log \mathbb{E}_{\boldsymbol{U}_{\boldsymbol{A}_j}}[\prod_{A \in \boldsymbol{A}_j} P(a \mid pa_A, \boldsymbol{U}_A; \theta^i_A)] \qquad \text{(mutually indep. } \boldsymbol{U}_A) \tag{62}$$

$$= \sum_{j=1}^{m} \log P(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}); \theta^i_{\boldsymbol{A}_j}) \qquad \text{(trunc. fact. prod.)} \tag{63}$$

From the last expression, we can observe that the NCM parameterization is modular w.r.t. the c-components, as Rahman et al. [28] also discusses. We rewrite the full optimization program again:

$$\hat{\Theta_{\boldsymbol{A}}} \in \underset{\Theta_{\boldsymbol{A}}: \langle \theta^1_{\boldsymbol{A}}, \theta^2, \ldots, \theta^K, \theta^* \rangle}{\arg\max} \sum_{\boldsymbol{a}} \exp\{\sum_{j=1}^{m} \log P(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}); \theta^*_{\boldsymbol{A}_j})\} \cdot \psi(\boldsymbol{a}) \tag{64}$$

$$+ \Lambda \cdot \sum_{i=1}^{K} \sum_{j=1}^{m} \sum_{\boldsymbol{a}_j \in D^i} \log P(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}); \theta^i_{\boldsymbol{A}_j}) \tag{65}$$

$$\text{s.t. } \theta^i_V = \theta^j_V, \qquad \forall i, j \in \{1, 2, \ldots, K, *\} \quad \forall V \notin \Delta_{i,j}$$

Let $\boldsymbol{A}_j$ be a c-component that S_i is not pointing to it in $\mathcal{G}^{\boldsymbol{\Delta}}$, i.e., $\boldsymbol{A}_j \cap \Delta_i = \varnothing$. The latter means that the parameter sharing $\theta^*_V = \theta^i_V$ is enforced for all $V \in \boldsymbol{A}_j$; we call these parameters $\theta^{i,*}_{\boldsymbol{A}_j}$. We notice that $\theta^{i,*}_{\boldsymbol{A}_j}$ only appears through the term $\log P(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}); \theta_{\boldsymbol{A}_j})$ in the score function; once in the main objective as $\theta^*_{\boldsymbol{A}_j}$ and once in the regularizer as $\theta^i_{\boldsymbol{A}_j}$. For $\Lambda \to \infty$, the regularizer enforces $\theta^{i,*}_{\boldsymbol{A}_j}$ to satisfy the following criterion:

$$\theta^{i,*}_{\boldsymbol{A}_j} \in \underset{\theta_{\boldsymbol{A}_j}}{\arg\max} \sum_{\boldsymbol{a}_j \in D^i} \log P(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}); \theta_{\boldsymbol{A}_j}) \tag{66}$$

This criterion is in fact an interventional (L2) constraint [41] enforced on $\theta^{i,*}_{\boldsymbol{A}_j}$ that requires $\theta^{i,*}_{\boldsymbol{A}_j}$ to approximate $P^i(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}))$ using the observational data D^i. Since $P^i(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}))$ is a complete c-factor, it is identifiable from $P^i(\boldsymbol{a}_j, pa_{\boldsymbol{A}_j})$ [36]. Therefore, by increasing the sample size $|D^i| \to \infty$ and the model complexity of $\theta^{i,*}_{\boldsymbol{A}_j}$, satisfying the criterion in Eq. 66 guarantees arbitrarily accurate approximation of the interventional quantities $P^i(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}))$ [41]. This implies that we can replace the terms involving the parameters $\theta^{i,*}_{\boldsymbol{A}_j}$ with any consistent approximation

of $P^i(\boldsymbol{a}_j \mid do(pa_{\boldsymbol{A}_j}))$ as constants. To get the approximation, we are free to use any probabilistic model and architecture depending on the context; this includes the option to train the NCM parameters $\theta^{i,*}_{\boldsymbol{A}_j}$ in the pre-training.

This adjustment gets us to the exact procedure pursued in Algorithm 1, thus proves consistency of it with what we would achieve via Theorem 2. □

E.3 Proof of Theorem 3

For this proof, it is useful to define the worst-case risk w.r.t. the selection diagram and the source distributions.

Definition 8 (Worst-case risk)**.** For selection diagram $\mathcal{G}^{\boldsymbol{\Delta}}$ and source distributions $\mathbb{P}$, the worst-case risk of classifer $h : \Omega_{\boldsymbol{X}} \to \Omega_Y$ is denoted by $R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h)$ and defined as the solution of partial transportation task for the query $\mathbb{E}_{P^*}[\mathcal{L}(Y, h(\boldsymbol{X}))]$, where $\mathcal{L}(y, \hat{y})$ is a loss function. Formally,

$$R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h) := \max_{\text{tuple of SCMs } \mathbb{M}_0 \text{ that entails } \mathbb{P} \text{ \& induces } \mathcal{G}^{\boldsymbol{\Delta}}} R_{P^{\mathcal{M}^*_0}}(h). \tag{67}$$

□

Theorem 3 (restated). *For discrete $\boldsymbol{X}, Y$ CRO terminates. Furthermore, for large enough data across all source domain, the worst-case risk of CRO is at most ϵ away from the worst-case optimal classifier w.r.t. selection diagram $\mathcal{G}^{\boldsymbol{\Delta}}$ and source data $\mathbb{P}$. Formally,*

$$\lim_{n\to\infty} P(R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h^{\text{CRO}}_n) - \min_{h:\Omega_{\boldsymbol{X}}\to\Omega_Y} R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h) > \epsilon) \to 0 \tag{68}$$

where $h^{\text{CRO}}_n := \text{CRO}(\mathbb{D}_n, \mathcal{G}^{\boldsymbol{\Delta}})$, and $\mathbb{D}_n = \langle D^1, D^2, \ldots, D^K \rangle$ is a collection of datasets that each contain at least n datapoints. □

Proof. Soundness of CRO relies on consistency of Neural-TR (Alg. 1 as a subroutine; we pick the data size large enough to satisfy this condition according to Theorem 2.

Termination. Let $\hat{\theta}^*_1, \hat{\theta}^*_2, \ldots$ be the sequence of target NCMs produced during the runtime of CRO, and let $h_1, h_2, \ldots$ be the sequence of classifiers obtained after each iteration. Let Π denote the space of all distributions over $\boldsymbol{X}, Y$. For discrete $\boldsymbol{X}, Y$, the space Π is a compact subspace of some Euclidean space. Thus, every sequence in Π has a convergent subsequence, especially the sequence $\{P(\boldsymbol{x}, y; \hat{\theta}^*_m)\}_m \subset \Pi$; let $\{P_l\}_l$ be this convergent subsequence. Every convergent subsequence is Cauchy, which means,

$$\forall \tau > 0 \quad \exists n > 0 \quad \forall l, l' > n : d(P_l - P_{l'}) < \tau, \tag{69}$$

where d is an appropriate metric over the probability space. Choose τ small enough w.r.t. the convergence tolerance $\delta > 0$ to ensure,

$$\forall P, P' \text{ where } d(P, P') < \tau \implies \forall h : \Omega_{\boldsymbol{X}} \to \Omega_Y \quad |R_P(h) - R_{P'}(h)| \leqslant \delta. \tag{70}$$

The above is possible, since the mapping $R_P(h)$ is a bounded and continuous mapping on the space Π. Now, we are guaranteed to find an index l such that,

$$|R_{P(\boldsymbol{x},y;\hat{\theta}^*_{l+1})}(h_l) - R_{P(\boldsymbol{x},y;\hat{\theta}^*_l)}(h_l)| < \delta. \tag{71}$$

Notice that by definition, $\hat{\theta}^*_{l+1}$ is obtained by Neural-TR (Alg. 1) to attain the worst-case risk of h_l, i.e.,

$$R_{P(\boldsymbol{x},y;\hat{\theta}^*_{l+1})}(h_l) = R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h_l). \tag{72}$$

Moreover,

$$R_{P(\boldsymbol{x},y;\hat{\theta}^*_l)}(h_l) \leqslant \max_{i\in\{1,2,\ldots,l\}} R_{D^*_i}(h_l). \tag{73}$$

Putting the last three equations together we have,

$$R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h_l) \leqslant \max_{i\in\{1,2,\ldots,l\}} R_{D^*_i}(h_l) + \delta, \tag{74}$$

which invokes the termination.

https://doi.org/10.52202/079017-4376

Worst-case optimality. Suppose h^{CRO} is returned by CRO, and let h^* be the true worst-case optimal classifier defined as,

$$h^* \in \min_{h:\Omega_{\boldsymbol{X}} \to \Omega_Y} R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h). \tag{75}$$

Let $\mathbb{D}$ denote the collection of datasets collected by the algorithm before termination. We know that h^{CRO} is robust to $\mathbb{D}^*$, i.e.,

$$h^{\text{CRO}} \in \operatorname*{arg\,min}_{h:\Omega_{\boldsymbol{X}} \to \Omega_Y} \max_{D \in D^*} R_D(h) \implies \max_{D \in D^*} R_D(h^{\text{CRO}}) \overset{\text{opt. } h^{\text{CRO}}}{\leqslant} \max_{D \in D^*} R_D(h^*) \tag{76}$$

Moreover, every distribution in $\mathbb{D}^*$ is entailed by an NCM that represents a possible target domain. Therefore, the worst-case risk is at least as large as the worst-case empirical risk on the set of distribution $\mathbb{D}$, i.e.,

$$\max_{D \in D^*} R_D(h^*) \leqslant R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h^*) \tag{77}$$

Since that algorithm has terminated we have,

$$R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h^{\text{CRO}}) < \max_{D \in D^*} R_D(h^{\text{CRO}}) + \delta. \tag{78}$$

where $\delta > 0$ is the tolerance for the convergence condition in the algorithm. Putting all inequalities together, we have,

$$\begin{aligned} R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h^{\text{CRO}}) - \delta &\leqslant \max_{D \in D^*} R_D(h^{\text{CRO}}) && (79)\\ &\leqslant \max_{D \in D^*} R_D(h^*) && (80)\\ &\leqslant R_{\mathcal{G}^{\boldsymbol{\Delta}},\mathbb{P}}(h^*), && (81) \end{aligned}$$

which indicates that the worst-case risk of h^{CRO} is at most δ larger than the optimal worst-case risk. □

F Broader Impact and Limitations

Our work investigates the design of algorithms and conditions under which knowledge acquired in one domain (e.g., particular setting, experimental condition, scenario) can be generalized to a different one that may be related, but is unlikely to be the same. As alluded to in this paper, under-identifiability issues and the difficulty of stating realistic assumptions that are conducive to extrapolation guarantees are pervasive throughout the data sciences. Our hope is that our analysis with a more surgical encoding of structural differences between domains that allow the empirical investigator to determine whether (and how) her/his understanding of the underlying system is sufficient to support the generalization of prediction algorithm is an important addition towards safe and reliable AI. This approach is not without limitations, however. We have shown that selection diagrams are sufficient to ensure consistent domain generalization (through bounds instead of point estimates) but arguably restrict the analysis to a narrow class of problems as graphs or super-structures need to be defined. This stands in contrast with representation learning methods that operate on higher-dimensional spaces, e.g. text, images, which are difficult to reason about in a causal framework. The trade-off is that guarantees for consistent extrapolation are difficult to define and that one-size-fits-all assumptions are difficult to justify in practice. Partial transportability may be understood as a complementary view-point on this problem, applicable in a different class of problems in which structural knowledge is available implying that non-trivial guarantees for extrapolation can be established. Pushing the boundaries of methods based on causal graphs to reach compelling real-world applications is arguably one the most important frontiers for the causal community as a whole. In this work, there is scope for improving posterior estimation and for introducing assumptions on the class of SCMs that are modelled, e.g. linear Gaussian models, etc., that could lead to efficient predictors in higher-dimensional spaces. Similarly, relaxations of selection diagrams, e.g. in the form of equivalence classes or partially-known graphs, could be developed for applications in domains where knowledge of graph structure is unrealistic.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see the contribution bullets.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See appendix F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-4376

Justification: All proofs are provided in the appendix with a more detailed restatement of the theorems and needed assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: The code will be accessible through git-hub after publication. We ensured that the results are reproducible; please see Appendix B.3 for some details on the used architecture.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is provided.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See the corresponding examples for each experiment, and the reproducibility note in B.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: All the examples are small, so it is not applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

 https://doi.org/10.52202/079017-4376

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were executed on a Macbook Pro M2 32 GB RAM.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We respect NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See appendix F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work is theoretical and bears no such risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All related and used results are cited properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-4376

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The code is annotated and provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.