TECHNICAL REPORT
R-133
May, 2025

Adapting, Fast and Slow: A Causal Approach to Few-Shot Sequence Learning

Kasra Jalaldoust and **Elias Bareinboim**
Causal Artificial Intelligence Lab
Columbia University
{kasra,eb}@cs.columbia.edu

Abstract

Generalization to an unseen target domain is not possible without asserting a *causal structure* that constrains the source and target domains. We study a setting where ample source data is supplemented with limited target data, also known as *supervised domain adaptation* (DA). We consider multi-source DA with discrete-valued variables, and assume existence of a causal structure underlying the source and target domains. We design a structure-informed procedure that leverages qualitative knowledge of the structure – which is in form of causal graphs and domain discrepancies – to *transport* inferences from the source data to the target domain. We also design a structure-agnostic algorithm that would achieve performance guarantees *almost* as good as the structure-informed baseline, offering *few-shot learning* in certain instances. We extend our findings to the sequential prediction task, where knowledge of the complex causal structure allows the structure-informed procedure to learn modular predictors from different source domains and systematically recompose them for faster adaptation in the target domain, and we then show that in these scenarios the structure-agnostic approach would achieves similar fast rates as well. Our results characterize when and how few-shot sequence learning is possible, and provide a causal theoretical basis for data-driven domain adaptation through a unifying structure-agnostic scheme. Experiments corroborate our results.

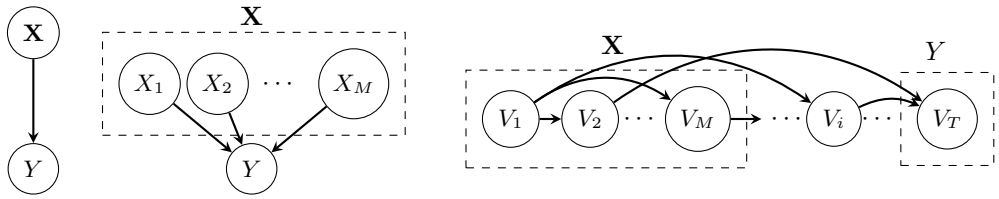
1 Introduction

Machine learning deals with generalizing patterns from finite samples to the distribution that generates these samples. The classical sample-to-population performance guarantees [34, 35] rely on the assumption that *target domain*, where the solution would be evaluated, entails a data distribution identical to the *source domain*, where the training data is obtained from. However, in practice the performance would take a serious even under small qualitative differences between source and target domains. This problem is known broadly as a *distribution shift* in ML, and generalizability or external validity in a broader scientific context. In particular, the *domain generalization* task refers to a situation where the learner has access to typically large data collected from one or multiple source domains and no data from the target domain. This is an extreme case of the *domain adaptation* problem where the learner also has access to a small amount of data collected from the target domain.

Theoretical understanding of generalization across domains is challenging. Arbitrary differences between the source and target domains inevitably imposes a barrier for learning, as there would be no basis for usefulness of source data in the target learning task. Thus, a formal approach to this problem necessitates establishing a notion of *structure* that specifies what the target domain can be in relation to the source domains. Then, one can imagine a carefully designed algorithm that leverages this structure and uses only the statistical associations present in the source data that would provably remain stable/invariant in the target, thus achieve a prediction rule with out-of-distribution guarantees.

Table 1: Overview of the settings considered in this paper

Setting	Causal diag.	Str.-informed	Str.-agnostic	Examples
Uni-cause (App. A)	Fig 1a	Alg 5, Lem A.3	Alg 6, Thm A.5	Ex A.1, Ex A.4
Multi-cause (Sec. 2)	Fig 1b	Alg 1, Lem 2.3	Alg 2, Thm 2.5	Ex 2.2, Ex 2.4
Sequential (Sec. 3)	Fig 1c	Alg 3, Lem 3.3	Alg 4, Thm 2.5	Ex 3.1



(a) Uni-cause (b) Multi-cause (c) Sequential

Figure 1: Causal diagram representing each of the settings considered in this paper.

Domain adaptation in prediction tasks involving covariates X and label Y has been studied in the literature [8, 9, 23, 6, 37, 15, 16], where various notions of divergence between the source and target X distribution are used as proxies for domain-relatedness. Other work in this area leverages distributional assumptions relating source and target, e.g., [10, 7, 5, 4] where learning in the source yields smaller complexity for learning in the target, e.g., through learning a shared *representation*.

Humans are particularly effective in transferring knowledge across domains [21, 25, 33], and causality is known to be the pillar of human understanding and decision making, especially under changing circumstances [14, 31]. Principles of generalization to the unseen from a causal perspective has been extensively studied under the rubrics of *transportability* [28, 2, 13, 12, 17, 18], and also through the lens of statistical invariances entailed by an implicit causal structure [29, 30, 20, 22]. In DA, since *some* target data is available, the learner would always have the choice of discarding the source data entirely, and relying solely on the target data. Thus, the theoretical question in DA is not whether it is possible to *learn*, but how fast learning can take place and how to best leverage the data from the source domains. In this paper, we seek to characterize the situations where certain aspects of the source data deems generalizable, thus allowing zero-shot/few-shot learning of the target (i.e., fast adaptation), and when learning from the source data hinders learning in the target (i.e., slow adaptation), and what lies in between these two extremes. Our contributions are the following:

1. **Causal structure for faster adaptation rates.** In Section 2, we illustrate the role of an underlying causal structure in the classification task, and introduce a more fine-grained causal structure that allows transportability through a structure-informed procedure. We provide target performance guarantees for the structure-informed predictor, and introduce a structure-agnostic procedure with a small excess risk compare to the structure-informed baseline, enabling few-shot adaptation.

2. **Extension to sequence adaptation.** In Section 3, we consider DA in sequential prediction task (Figure 1c) where the objective is to predict the last token of a sequence from a prefix of it, e.g., fine-tuning for reasoning in the language models. We introduce discrepancy oracle (Definition 3.2) that encodes when a common logic/circuit is used at to generate tokens at different position in different domains. We devise a structure-informed algorithm that leverages this elaborate structural knowledge to learn useful modular predictors from the combination of source and target data, and compose them for faster adaptation. Next, we introduce an structure-agnostic algorithm that competes with the structure-informed baseline with a $\mathcal{O}(\sqrt{\frac{\text{poly}(T)}{n}})$ margin where T is the length of the sequence and n the number of target data. Our findings shed light on possibility of agnostic adaptation in a multitude of setups where the existing baselines fail.

Preliminaries. We use capital letters to denote variables (X), small letters for their values (x), bold letters for sets of variables ($\mathbf{X}$) and their values ($\mathbf{x}$), and use caligraphic letters ($\mathcal{X}$) to denote their support. A conditional independence statement in distribution P is written as $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_P$. A d-separation statement in some graph $\mathcal{G}$ is written as $(\mathbf{X} \perp\!\!\!\perp_d \mathbf{Y} \mid \mathbf{Z})$. To denote $P(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x})$, we use the shorthand $P(\mathbf{y} \mid \mathbf{x})$. The basic semantic framework of our analysis relies on Structural Causal Models (SCMs) [27, Definition 7.1.1], which are defined below.

Definition 1.1. *An SCM $\mathcal{M}$ is a tuple $M = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}, P \rangle$ where each observed variable $V \in \mathbf{V}$ is a deterministic function of a subset of variables $\mathbf{Pa}_V \subset \mathbf{V}$ and latent variables $\mathbf{U}_V \subset \mathbf{U}$, i.e., $v := f_V(\mathbf{pa}_V, \mathbf{u}_V)$, $f_V \in \mathcal{F}$. The unobserved variables $\mathbf{U}$ follow a distribution $P(\mathbf{u})$.* ▫

We assume the model to be recursive, i.e. that there are no cyclic dependencies among the variables. SCM $\mathcal{M}$ entails a probability distribution $P^{\mathcal{M}}(\mathbf{v})$ over the set of observed variables $\mathcal{V}$ such that

$$P^{\mathcal{M}}(\mathbf{v}) = \int_{\mathcal{U}} \prod_{V \in \mathbf{V}} P^{\mathcal{M}}(v \mid \mathbf{pa}_V, \mathbf{u}_V) \cdot P(\mathbf{u}) \cdot d\mathbf{u}, \tag{1}$$

where the term $P(v \mid \mathbf{pa}_V, \mathbf{u}_V)$ corresponds to the function $f_V \in \mathcal{F}$ in the underlying structural causal model $\mathcal{M}$. It also induces a causal diagram $\mathcal{G}_{\mathcal{M}}$ in which each $V \in \mathbf{V}$ 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 $\mathbf{U}_{V_i} \cap \mathbf{U}_{V_j} \neq \varnothing$ or $P(\mathbf{U}_{V_i}, \mathbf{U}_{V_j}) \neq P(\mathbf{U}_{V_i}) \cdot P(\mathbf{U}_{V_j})$, that is V_i and V_j are confounded [3].

Throughout this paper, we only consider discrete-valued variable, and assume the observational distributions entailed by the SCMs satisfy strict positivity assumption, that is, $P^{\mathcal{M}}(\mathbf{v}) > \epsilon$, for every $\mathbf{v}$ and a known constant ϵ. We will also operate non-parametrically, i.e., making no assumption about the particular functional form or the distribution of the unobserved variables.

2 Transportability of modular predictors

We consider a classification problem where $\mathbf{X} = \{X_1, X_2, ..., X_M\}$ is multivariate, each taking value in a finite set $\mathcal{X}$, and Y taking value in the finite set $\mathcal{Y}$, which is is the last variable in the causal order. Further, we assume that Y is a downstream variable, i.e., last in the causal order. Also, we assume that Y is not confounded with any of the $\mathbf{X}$ variables, i.e., there exists no unobserved variable pointing to both Y and $\mathbf{X}$ variables in the causal diagram induced by the source and target SCMs. The objective is predicting the label Y using covariates $\mathbf{X}$, i.e., learning $P^*(y \mid x_1, ..., x_M)$. There is a loss function $\ell(\mu; y, \mathbf{x})$, and the risk is defined as the expected loss $R_{P*}(\mu) := \mathbb{E}_{P*}[\ell(\mu; Y, \mathbf{X})]$. The true risk minimizer is denoted as $\mu_* \in \arg\min_{\mu: \mathcal{X} \to \text{simplex}^{|\mathcal{Y}|}} R_{P*}(\mu)$, and the empirical risk minimizer w.r.t. data D is denoted as,

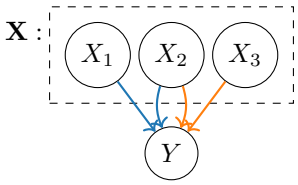


Figure 2: Causal diagrams corresponding to Example 2.2. Color-coded edges show parents of Y in each domain: blue for $\mathcal{M}^1$, orange for $\mathcal{M}^*$

$$\hat{P}(y \mid \mathbf{x}; D) \in \underset{\mu: \mathcal{X} \to \text{simplex}^{|\mathcal{Y}|}}{\arg\min} \sum_{y, \mathbf{x} \in D} \ell(\mu; y, \mathbf{x}). \tag{2}$$

We consider the loss to be the negative log-likelihood $\ell(\mu; y, x) := -\log \mu(y \mid \mathbf{x})$ in this work, and the objective is to minimize the excess risk denoted by $R_{P*}(\mu) - R_{P*}(\mu_*)$.

Suppose we have access to target data D^* drawn i.i.d. from the target domain π^* that entails the target distribution $P^*(x, y)$, as well as source data $D^1, D^2, \ldots, D^K$ from a set of source domains $\Pi^{\text{src}} = \{\pi^1, \pi^2, ..., \pi^K\}$ that entail the source distributions $P^{\text{src}} = \{P^1(x, y), P^2(x, y), \ldots, P^K(x, y)\}$. Let $n = |D^*|$ and $N = |D^j|$ for all $j \in [K]$, and suppose $N \gg n$. We assume strictly positive mass for every combination of the variables, i.e., $P^j(x, y) > \epsilon$ for all $j \in [K] \cup \{*\}$. To encode structural invariances between the domain, we use the following notation by [13, 1].

Definition 2.1 (Domain discrepancy sets)**.** The collection of subsets of observable variables $\Delta = \{\Delta_{j,j'}\}_{j,j' \in [K] \cup \{*\}}^{K}$ where $\Delta_{j,j'}$ contains a variable $V \in \mathbf{V}$ if there is a possible mismatch between the causal mechanism of V in domains $\pi^j, \pi^{j'}$, i.e., either $f_V^j \neq f_V^{j'}$ or $P^j(\mathbf{u}_V) \neq P^{j'}(\mathbf{u}_V)$. ▫

Example 2.2 (Modular transportability)**.** Suppose $X_1, X_2, X_3, Y \in \{0, 1, ..., 9\}$. There is a single source domain $\mathcal{M}^1$ and a target domain $\mathcal{M}^*$, described as follows:

$$
\begin{aligned}
U_{X_1}, \ldots, U_{X_M} &\sim P(u_{X_1}, \ldots, u_{X_m}) \\
U_Y &\sim \text{Multinomial}(\text{prob} : \{0.91, 0.01, ..., 0.01\}) \\
X_m &\leftarrow U_{X_m}, \quad \forall m \in [M] \\
Y &\leftarrow \begin{cases} X_1 - X_2 + U_Y \pmod{10} & \text{in } \mathcal{M}^1 \\ X_3 - X_2 + U_Y \pmod{10} & \text{in } \mathcal{M}^* \end{cases}
\end{aligned}
$$

The causal diagram corresponding to these SCMs is shown in Figure 2. The causal parents of Y are a different subset of covariates in the source and target domains, but the mechanism that decides Y based on these parents is shared between π^* and π^1; it is a noisy subtraction of second parent from the first parent.

Suppose we know the parents of Y in both source and target, i.e., we have access to the ordered sets $\mathbf{Pa}_Y^1 = \langle \mathbf{Pa}_Y^1[1], \mathbf{Pa}_Y^1[2] \rangle = \langle X_1, X_2 \rangle$ and $\mathbf{Pa}_Y^* = \langle \mathbf{Pa}_Y^*[1], \mathbf{Pa}_Y^*[2] = \rangle = \langle X_3, X_2 \rangle$. Moreover, suppose we have access to Δ, which indicates the mechanism sharing, and in this case $Y \notin \Delta_{1,*}$ implies $f_Y^*(a, b, u_Y) = f_Y^1(a, b, u_Y)$ for all $a, b \in \{0, 1, ..., 9\}$, and $P^*(u_Y) = P^1(u_Y)$. Using this information, we can train a modular predictor for Y using data from π^1, i.e, $\mu_1(y \mid a, b) = \hat{P}(y \mid X_1 = a, X_2 = b; D^1)$, and then, because we know the parents of Y in the target, we can plug them into this predictor in the appropriate order and predict Y with small error in the target domain without any target data.

Note that for the full covariates, $f_Y^1(\mathbf{x}, u_Y) \neq f_Y^*(\mathbf{x}, u_Y)$, so if we treat this example as a uni-cause case (Appendix A), then $Y \mid \mathbf{X}$ wouldn't be invariant (i.e., $Y \in \Delta_{1,*}$), and therefore, the structure-aware strategy would discard the source data. However, once we unfold $\mathbf{X}$ into $\{X_1, X_2, X_3\}$, with the more elaborate structure that involves the ordered parents of Y in each domain and Δ, it would be possible to transport across the domains. In situations where $Y \in \Delta_{j,*}$ for all $j \in [K]$, no transport is possible. □

Notably, because the parents of Y are different between the domains, the existing notions of transportability (e.g., [13]) do not license transport in the case of Example 2.2. However, we can transport since a common causal module $f_Y^* = f_Y^1$ governs generation of the label Y across the domains. More broadly, if we were to consider the less-granular causal model based on the causal diagram $\mathbf{X} \rightarrow Y$, then applying the same machinery would not license transport, as it is leveraging *modularity* of a more-granular mechanism between source and target; a more detailed discussion on the *uni-cause model* is provided in Appendix A, and serves as a useful introduction to the general approach we will develop subsequently. Algorithm 1 summarizes the strategy discussed in Example 2.2, and what follows its rate.

Algorithm 1 Str.-informed DA (multi-cause)

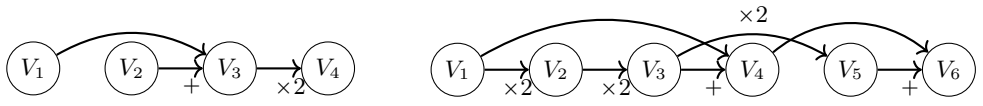
Require: $D^1, D^2, \ldots, D^K, D^*; \Delta, \{\mathbf{Pa}_Y^j\}$
Ensure: $\mu_{\text{TR}}(y \mid \mathbf{x}) \approx P^*(y \mid \mathbf{x})$
1: $\mathcal{J} \leftarrow \{j \in [K] \text{ and } Y \notin \Delta_{j,*}\}$
2: $c \leftarrow |\mathbf{Pa}_Y^*|$
3: $D_{\mathbf{R}}^{\text{TR}} \leftarrow \bigcup_{j \in \mathcal{J}} D^j[Y, \mathbf{R}_{1:c} = \mathbf{pa}_Y^j[1:c]]$
4: $D_{\mathbf{R}}^* \leftarrow D^*[Y, \mathbf{R}_{1:c} = \mathbf{Pa}_Y^*[1:c]]$
5: $\mu(y \mid \mathbf{r}) \leftarrow \hat{P}(y \mid \mathbf{r}_{1:c}; D_{\mathbf{R}}^{\text{TR}} \cup D_{\mathbf{R}}^*)$
6: **Return** $\mu_{\text{TR}} \leftarrow \mu(y \mid \mathbf{r} = \mathbf{pa}_Y^*)$

Proposition 2.3 (Structure-informed DA rate; multi-cause)**.** *In Algorithm 1 with high probability,*

$$
R_{P^*}(\mu_{\text{TR}}) - R_{P^*}(\mu_*) = \begin{cases} \mathcal{O}(\frac{|\mathcal{X}|^c \cdot |\mathcal{Y}|}{\epsilon^2 \cdot N}) & \text{if } \mathcal{J} \neq \varnothing \\ \mathcal{O}(\frac{|\mathcal{X}|^c \cdot |\mathcal{Y}|}{\epsilon \cdot n}) & \text{otherwise} \end{cases} \tag{3}
$$

where $c = |\mathbf{Pa}_Y^*| \leqslant M$. □

All proofs are in Appendix B. In words, if $\mathcal{J}$ is empty, it means that $Y \in \Delta_{j,*}$ for all source domains π^j, thus no source data can help learning in the target. Thus, our error rate decays similar to ERM on the target D^j. If there are any sources that can be used for learning in the target, knowledge of the domain discrepancies and the parent sets allows us to pool the data from relevant sources, reorder the covariates to match with the parents of Y in the target, and train a single predictor using large data pooled from the sources and target. In this case, what allows zero-shot generalization is large source data supplemented with domain knowledge about the causal structure within each of the domains and the mechanism discrepancies across the domains. In the next example we demonstrate a strategy that probes whether $Y \in \Delta_{j,*}$ leveraging the target data, yet achieves competitive rates.



(a) $\mathcal{G}^1$ (b) $\mathcal{G}^*$

Figure 3: Causal diagrams corresponding to Example 3.1. The labels $+, \times 2$ indicate true mechanisms that are noisy summation and doubling. The query of interest is $P^*(v_6 \mid v_1) \approx 1_{\{v_6 = 13 \times v_1 (\text{mod} 10)\}}$

Example 2.4 (Agnostic approach: multi-cause prediction)**.** In the context of Example 2.2, suppose we do not have access to domain discrepancy sets Δ and the parent sets $\mathbf{Pa}_Y^1, \mathbf{Pa}_Y^*$. To compensate for the lack of knowledge, we take the following approach: we train a collection of predictors that would contain at least one whose prediction error in the target is as good as what is achievable through structure-informed DA, and then use held-out target data to find the best performing one. In particular, for every $c \in \{0, 1, 2, 3\}$, we take two ordered set of the covariates of size c, and regress Y on these c variables in these order, once using target data only and once using source and target data combined. this results in a total of at most $\sum_{c=0}^{3} (\binom{3}{c} \cdot c!)^2 + \binom{3}{c} \cdot c! = 98$ predictors. Notice that this number does not depend on dimensionality of X, Y. Finally, we use held-out target data to choose the best performing one in this pool of predictors. This imposes a fixed excess risk on top of what is achievable through structure-informed DA. ▫

Algorithm 2 generalizes the approach described in Example 2.4, and what follows is its performance guarantee.

Proposition 2.5 (Structure-agnostic DA rate: multi-cause.)**.** *Let $\mu_{\text{TR}}, \mu_{\text{Ag}}$ be learned by the structure-informed and agnostic DA procedures (Algorithms 1 and 2), respectively. We have,*

$$R_{P*}(\mu_{\text{Ag}}) = \mathcal{O}(R_{P*}(\mu_{\text{TR}}) + \sqrt{\frac{K \cdot M \cdot \log M}{n}}), \tag{4}$$

where K is the number of source domains, M is the number of covariates, and n is the number of data from the target domain. ▫

Algorithm 2 Structure-agnostic DA (multi-cause)

Require: $D^1, D^2, \ldots, D^K, D^*$
Ensure: $\mu_{\text{Ag}}(y \mid \mathbf{x}) \approx P(y \mid \mathbf{x})$
1: $D^*_{\text{tr}}, D^*_{\text{te}} \leftarrow \text{partition}(D^*)$
2: $\mathcal{H} \leftarrow \varnothing$
3: **for** $c \in \{0, 1, ..., M\}$ **do**
4: **for** $\{\mathbf{Pa}_Y^j[1:c] \subseteq [M]\}$ and $\mathcal{S} \subseteq [K]$ **do**
5: $D^{\text{TR}} \leftarrow \bigcup_{j \in \mathcal{S}} D^j[Y, \mathbf{R} = \mathbf{Pa}_Y^j]$
6: $\mathcal{H} \leftarrow \mathcal{H} \cup \{\hat{P}(y \mid \mathbf{r}; D^*_{\text{tr}}[Y, \mathbf{R} = \mathbf{Pa}_Y^*] \cup D^{\text{TR}})\}$
7: **end for**
8: **end for**
9: **Return** $\mu_{\text{Ag}} \leftarrow \arg\min_{\mu \in \mathcal{H}} \ell(\mu; D^*_{\text{te}})$

In words, structure-agnostic DA yields a target risk that is slightly worse than what is achievable via the structure-informed DA. In a way, there is a cost for learning Δ and $\mathbf{Pa}_Y^*$ since these are assumed unknown.

3 Few-shot sequence learning

Suppose the observable variables in domain π^j ($j \in [K] \cup \{*\}$) are $\mathbf{V} = \{V_1, V_2, ..., V_{T_j}\}$ (possibly different number across domains), and every V_i takes value in a finite set $\mathcal{V}$, shared across all positions and domains. Assume the causal order $V_1 \prec V_2 \prec ... \prec V_{T_j}$, and that there exists no unobserved confounding. Let $\mathcal{G}^j = \{\mathbf{Pa}_i^j\}_{i=1}^T$ be the causal diagram underlying the SCM $\mathcal{M}^j$ (e.g., Figure 3). The goal of the task is to predict the last token of the sequences in the target using the first M tokens, i.e., to learn the conditional distribution $P^*(v_{T_*} \mid \mathbf{v}_{1:M})$. This is an extension of the multi-cause problem which has $Y = V_T$ and $\mathbf{X} = V_{1:M}$. Below is a illustrative example.

Example 3.1. Consider a single source domain π^1 represented by $\mathcal{M}^1$, and the target domain π^* represented by $\mathcal{M}^*$. The variables in both SCMs have a support of $\{0, 1, \ldots, 9\}$. The noise U_i in all domains and for all variables follows the distribution $\text{Multinomial}(\text{prob} : \{0.91, 0.01, ..., 0.01\})$; having most of the mass on 0 and uniformly small mass on other outcomes. Let $g_1(x_1, u) = 2 \times x_1 + u$ and $g_2(x_1, x_2, u) = x_1 + x_2 + u (\text{mod} 10)$ be noisy doubling and summation, respectively, and suppose the function f_i^j determining V_i in the domain j is equal to either g_1 or g_2. In the causal diagrams in

Figures 3a and 3b, for each variable the function that does the value assignment is shown. Suppose we know the causal diagrams (i.e., the parents for each variable in both domains) and the domain discrepancies as well as which variables share the same function in their value assignment. Since the distribution of U_i is the same for all variables and there exists no confounders, knowledge of domain discrepancies and the graph yields invariance of certain conditional distributions across different positions and domains, e.g.,

$$P^*(V_3 = y \mid V_2 = x_1) = \sum_{u=0}^{9} P(u) \cdot 1_{\{g_1(x_1,u)=y\}} = P^1(V_4 = y \mid V_3 = x_1). \tag{5}$$

Using these probabilistic invariances derived from the causal structure, we attempt to estimate the query of interest $\mathcal{Q} = P^*(v_6 \mid v_1)$. First, we train modular conditional probability distributions $\hat{P}_a(y \mid x_1), \hat{P}_b(y \mid x_1, x_2)$ using data from variables and their parents across different domains if they share the functional assignment. For example V_3 from target and V_4 from the source have a shared mechanism, as shown in Equation (5), entail the same conditional given their parents, thus, their data would be pooled to be used for estimation of $\hat{P}_a(y \mid x_1)$. Finally, we can transport $\mathcal{Q}$ as follows:

$$\begin{aligned}
\mathcal{Q} &= \sum_{v_{2:5}} P^*(v_{2:6} \mid v_1) && (6)\\
&= \sum_{v_{2:5}} P^*(v_2 \mid v_1) \cdot P^*(v_3 \mid v_1, v_2) \cdot P^*(v_4 \mid v_3) \cdot P^*(v_5 \mid v_4, v_2) \cdot P^*(v_6 \mid v_5, v_4) && (7)\\
&\approx \sum_{v_{2:5}} \hat{P}_a(v_2 \mid v_1) \cdot \hat{P}_b(v_3 \mid v_1, v_2) \cdot \hat{P}_a(v_4 \mid v_3) \cdot \hat{P}_b(v_5 \mid v_4, v_2) \cdot \hat{P}_b(v_6 \mid v_5, v_4). && (8)
\end{aligned}$$

The large source data ($N \gg n$) can be used for training of both $\hat{P}_a, \hat{P}_b$, so the above transportation formula can be considered a zero-shot generalization, i.e., even with no target data the above estimator retains accuracy due to transportation of conditionals from the large source data. □

As seen in the above example, the domain discrepancies may be more complicated in the sequential setting, allowing a match between mechanisms from different positions i, i' and across different domains j, j'. Below is an extension of domain discrepancies useful to accommodate such invariances.

Definition 3.2 (Discrepancy oracle)**.** Let $\Delta(i, j; i', j')$ be a boolean function that returns one if either $f_i^j \neq f_{i'}^{j'}$ or $P^j(u_i) \neq P^{j'}(u_{i'})$, and returns zero otherwise. □

In the context of Example 3.1, for example, $\Delta(4, 1; 3, *) = 0$ and $\Delta(6, *; 3, *) = 1$. In structure-informed procedure (Algorithm 3) the structure encoded by the discrepancy oracle Δ is used and the domain-specific causal diagrams $\mathcal{G}^1, \ldots, \mathcal{G}^K, \mathcal{G}^*$ to pool data that is causally relevant to each of the variables in the target sequence. Then, this data is used to learn the conditional distribution $P^*(v_i \mid \mathbf{pa}_i^*)$. These conditional distributions are then composed to yield an estimator of $P^*(v_{M+1:T_*} \mid v_{1:M})$, and finally the variables $V_{M+1}, ..., V_{T_*-1}$ are marginalized out to obtain an estimation of the target quantity $P^*(v_{T_*} \mid v_{1:M})$.

Algorithm 3 Structure-informed DA (sequential)

Require: $D^1, D^2, \ldots, D^K, D^*; \Delta; \{\mathcal{G}^j : \{\mathbf{Pa}_i^j\}\}$
Ensure: $\mu_{\text{TR}}(v_{T_*} \mid v_{1:M}) \approx P^*(v_{T_*} \mid v_{1:M})$
1: **for** $i \in \{M+1, ..., T_*\}$ **do**
2: $\quad \mathcal{J}_i \leftarrow \{(i', j') : \Delta(i, *; i', j') = 0 \text{ and } j \in [K]\}$
3: $\quad D_i^{\text{TR}} \leftarrow \bigcup_{i', j' \in \mathcal{J}_i \cup \{(i,*)\}} D^{j'}[Y : V_{i'}, \mathbf{X}_{1:c} : \mathbf{Pa}_{i'}^{j'}]$
4: **end for**
5: $\mu_{\text{TR}} = \prod_{i=M+1}^{T_*} \hat{P}(Y = v_i \mid \mathbf{X} = \mathbf{pa}_i^*; D_i^{\text{TR}})$
6: **Return** $\mu_{\text{TR}}(v_{T_*} \mid v_{1:M}) \leftarrow \sum_{v_{M+1:T_*-1}} \mu_{\text{TR}}$

Remark that in the multi-cause scenarios discussion in earlier, the structure-informed DA procedure (Algorithm 1) either achieves zero-shot generalization (i.e., rates in terms of N) or uses only the target data (rates in terms of n) (Lemma A.3 and Proposition 2.3). However, in sequential prediction, the predictor μ_{TR} returned by Algorithm 3 may lie in between the two extremes; in particular, if all conditional distributions $\{P^*(v_i \mid \mathbf{Pa}_i^*)\}_{i=M+1}^{T_*}$ are transported, i.e., data from at least one of the sources is pooled for estimation, then all of them would have low error, resulting in a target risk guarantee for μ_{TR} that depends on N. However, if none of the conditionals $\{P^*(v_i \mid \mathbf{Pa}_i^*)\}_{i=M+1}^{T_*}$ can be transported, then the guarantee would be in terms of the target data size n only. One can

imagine in-between situations where some of the conditionals are transported and others must be learned with target data only; in these situations, the risk bound would have a fast and slow components which decay with n and N, respectively. Below is an upper-bound for the target risk of structure-informed DA in sequential prediction (for a more refined analysis, refer to Appendix C).

Theorem 3.3 (Structure-informed DA rates; sequential prediction)**.** *In Algorithm 3 with high probability,*

$$R_{P^*}(\mu_{\text{TR}}) - R_{P^*}(\mu_*) = \begin{cases} \mathcal{O}(\frac{|\mathcal{V}|^T}{\epsilon^2 N}) & \text{if } \forall i \in \{M+1, ..., T\} : \mathcal{J}_i \neq \varnothing \\ \mathcal{O}(\frac{|\mathcal{V}|^{M+1}}{\epsilon n}) & \text{otherwise} \end{cases} \tag{9}$$

where T is the length of the longest sequence.

In words, the guarantee offered by Theorem 3.3 decays with N (i.e., zero-shot generalization) if all components of the sequence from $M+1$ to T can be transported from one of the sources. Transportability can be interpreted as a *causal interpolation* of the source domains, since each target mechanism f_i^* must be present in at least one position of one of the source domains. On the other hand, when at least one component cannot be transported, then the rate would involve a term which decays with n, i.e., a slow adaptation.

In more realistic settings where one does not have access to the discrepancy oracle and the causal diagrams, we pursue a strategy analogous to the agnostic procedure in the previous section. Each instance of the the domain knowledge (discrepancies and graphs) yields an estimator of $P^*(v_T \mid v_{1:M})$ trained using a combination of the source and target data (Algorithm 3). Since the domain knowledge $\Delta, \{\mathcal{G}^j\}_{j\in[K]\cup\{*\}}$ is a discrete object, there exists finitely many distinct estimators of $P^*(v_T \mid v_{1:M})$ considering all possibilities of the domain knowledge. We partition the target data into training and validation sets of size $\frac{n}{2}$, and use the training part along with the source data to obtain all possible estimators of $P^*(v_T \mid v_{1:M})$. Finally, we use the target validation data to pick the best performing estimator from the pool. Algorithm 4 summarizes this approach, and what follows is its performance guarantee.

Algorithm 4 Structure-agnostic DA (sequential)

Require: $D^1, D^2, \ldots, D^K, D^*; \Delta; \{\mathcal{G}^j\}$
Ensure: Target classifier $\hat{\mathbb{E}}_{P^*}[Y \mid x]$
1: Let $\mathcal{E} = \{(i,j) : i \in [T_j], j \in [K] \cup \{*\}\}$ and $\mathcal{H} \leftarrow \{\}$
2: $D^*_{\text{tr}}, D^*_{\text{te}} \leftarrow \text{partition}(D^*)$
3: **for** every partition of $\mathcal{E}$ into subsets $\mathcal{S} = \{\mathcal{E}_l\}_l$ **do**
4: $\quad \Delta(i,j;i',j') \leftarrow \begin{cases} 0 \text{ if } \exists \mathcal{E}_l \in \mathcal{S} \text{ s.t. } \in (i,j),(i',j') \in \mathcal{E}_l \\ 1 \text{ otherwise} \end{cases}$
5: $\quad$ **for** every set of graphs $\{G^1, \mathcal{G}^2, ..., \mathcal{G}^K, \mathcal{G}^*\}$ **do**
6: $\quad\quad \mathcal{H} \leftarrow \mathcal{H} \cup \{\text{StrInf}(D^1, ..., D^K, D^*_{\text{tr}}; \Delta; \{\mathcal{G}^j\})\}$
7: $\quad$ **end for**
8: **end for**
9: **Return** $\mu_{\text{Ag}} \leftarrow \arg\min_{\mu\in\mathcal{H}} \ell(\mu; D^*_{\text{te}})$

Theorem 3.4 (Structure-agnostic DA rate: sequential prediction)**.** *Let $\mu_{\text{TR}}, \mu_{\text{Ag}}$ be learned by the structure-informed and agnostic DA procedures (Algorithms 3 and 4). We have,*

$$R_{P^*}(\mu_{\text{Ag}}) = \mathcal{O}(R_{P^*}(\mu_{\text{TR}}) + \sqrt{\frac{K \cdot T^3 \cdot \log T}{n}}) \tag{10}$$

where K is the number of sources, and T is the length of the longest sequence.

The agnostic procedure would have a guarantee that is only marginally worse than what it achieved through the structure-informed procedure. For example, if the quantity of interest $P^*(v_T \mid v_{1:M})$ is transportable from the sources given $\Delta, \mathcal{G}^j$, i.e., the rate would only depend on N, then the agnostic procedure would adapt with a fast rate since the following upper-bound can be achieved:

$$R_{P^*}(\mu_{\text{Ag}}) - R_{P^*}(\mu_*) = R_{P^*}(\mu_{\text{Ag}}) - R_{P^*}(\mu_{\text{TR}}) + R_{P^*}(\mu_{\text{TR}}) - R_{P^*}(\mu_*) \tag{11}$$

$$= \mathcal{O}(\sqrt{\frac{K \cdot T^3 \cdot \log T}{n}} + \frac{|\mathcal{V}|^T}{\epsilon^2 N}). \tag{12}$$

In the above, for large enough N, the term involving n is dominant, thus we would achieve arbitrarily small target error with target data size polynomial in the number of variables. This can be compared with target-only estimation through ERM which achieves $R_{P^*}(\hat{P}(v_t \mid v_{1:M}; D^*)) - R_{P^*}(\mu_*) = \mathcal{O}(\frac{2^M}{n})$, which is exponential in the number of variables.

4 Two-stage adaptation

Algorithm 4 iterates over all combinations of discrepancy oracle and graphs, and this makes it computationally intractable. To resolve this issue, we introduce an alternative approach that is equivalent to Algorithm 4. It is a two-stage procedure that involves certain *pretraining* a single model on source data, and then reusing the model components for a separate *fine-tuning* step using the target data.

4.1 The architecture and pretraining

For simplicity, suppose the same number of variables are observed in all domains, i.e., $T_j = T$. Further, suppose $|\mathbf{Pa}_i^j| \leqslant 1$ for all $i \in [T], j \in [K] \cup \{*\}$; we remove this condition in Appendix D. The goal of pretraining is to use the large source data and learn the following mappings that satisfy the following desired properties.

1. The mechanism indicator $\phi : [T] \times [K] \to [d]$, such that,
$$\Phi(i,j) \neq \Phi(i',j') \text{ if and only if } \Delta(i,j;i',j') = 1 \tag{13}$$
2. The parent matrix: $A^j \in [0,1]^{T\times T}$ for all $j \in T$ is lower-diagonal, such that,
$$A^j_{i,i'} = 1 \text{ iff } \mathbf{Pa}_i^j = V_{i'} \tag{14}$$
3. Universal predictor $\Psi : \mathcal{V} \times \mathcal{V} \times [d] \to [0,1]$:
$$\forall y, x \in \mathcal{V}, i \in [T], j \in [K] : \Psi(y \mid x; \phi = \Phi(i,j)) = P^j(V_i = y \mid \mathbf{Pa}_i^j = x) \tag{15}$$

In words, it is desirable $\Phi(i,j)$ encodes the discrepancy oracle by clustering the position-domain pair into the categories $\{1,...,d\}$, and $\{A^j\}_{j=1}^K$ encode the causal diagram in each domain. It is clear that once the mappings satisfy the properties Equations (13) to (15), then we can have optimal prediction in the source domain: for predicting v_i in domain π^j take, $\hat{v}_i \sim \Psi(v_i \mid X = A^j_{i,\cdot} \cdot v_{1:T}; \Phi(i,j))$. Interestingly, any instantiation of $\Phi, \{A^j\}, \Psi$ that maximizes a penalized likelihood on source populations satisfies the above properties, as stated below.

Theorem 4.1 (Pretraining)**.** *Let θ^{src} be a set of parameters for the mappings above such that,*

$$\theta^{\text{src}} \in \arg\min_{\theta\in\Theta} \Big(\sum_{|\mathcal{V}|^T} \sum_{j=1}^{K} P^j(v_t) \sum_{i=1}^{T} -\log \Psi_\theta(v_i \mid A^j_{\theta\, i,\cdot} \cdot v_{1:T}; \Phi_\theta(i,j))\Big) + \lambda(d + \|A^{\cdot}\|_1), \tag{16}$$

where Θ denotes all parameterizations for the mappings, $[d]$ is the range of Φ, and $\|A^{\cdot}\|_1$ denotes sum of the entries of the parent matrices. $\Phi_{\theta^{\text{src}}}, \{A^j_{\theta^{\text{src}}}\}, \Psi_{\theta^{\text{src}}}$ satisfy Equations (13) to (15). □

Thus, maximizing the likelihood w.r.t. large enough source data yields the desired properties guaranteed in Theorem 4.1. Next, we leverage these properties for fast adaptation to target data.

4.2 Fine-tuning

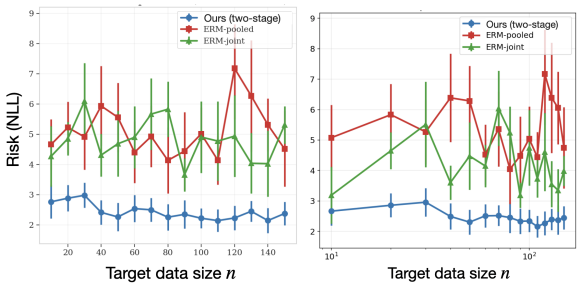
We partition the target data D^* into $D^*_{\text{tr}}, D^*_{\text{ft}}, D^*_{\text{te}}$ of proportionate size. Recall $[d]$ as the range of the mapping Φ. Once pretrained, each $\phi \in [d]$ corresponds to a subset of position-domain pairs in the source that share the causal mechanism; this is due to Theorem 4.1 that ensures Equation (13). Suppose for a position $i \in [T]$, it holds that $\Delta(i,*;i',j') = 0$. Thus, for all $x, y \in \mathcal{V}$,

$$P^*(V_i = y \mid \mathbf{Pa}_i^* = x) = P^{j'}(V_{i'} = y \mid \mathbf{Pa}_i^{j'} = x) \qquad \text{(Due to } \Delta(i,*;i',j') = 0) \tag{17}$$

$$= \Psi_{\theta^{\text{src}}}(V_{i'} = y \mid X = x; \phi = \Phi(i',j')) \quad \text{(Equation (15))} \tag{18}$$

Notably, $\Psi_{\theta^{\text{src}}}$ is learned in the pretraining stage, yet we need to discover $\phi, \mathbf{Pa}_i^*$ at each position $i \in [T]$. To this end, we take a target parent matrix $A^* \in [0,1]^{T\times T}$ to encode $\mathbf{Pa}_i^*$ via $A^*_{i,i'} = 1$ if $\mathbf{Pa}_i^* = V_{i'}$, and also a target mechanism indicator $\Phi^* : [T] \to [d]$. Next, we use D^*_{ft} to learn,

$$\theta^{\text{trg}} \in \arg\min_{\theta\in\Theta} \sum_{v_{1:T}\in D^*_{\text{ft}}} \sum_{i=1}^{T} -\log \Psi_{\theta^{\text{src}}}(Y = v_i \mid X = A^*_{\theta\, i,\cdot} \cdot v_{1:T}; \Phi^*_\theta(i)). \tag{19}$$



(a) Multi-cause settings (b) Sequential setting

Figure 4: Performance of our algorithm (blue) in comparison with two baselines. The baselines have the same architecture as our algorithm (Section 4.1), and combine the source and target data for a single-stage training. ERM-joint preserves the domain labels (i.e., only pretraining), while ERM-pool drops them (Φ constant). For more details see Appendix E.

This optimization considered at each position $i \in [T]$ is equivalent to using $|D^*_{\text{ft}}|$ target data to pick the best-performing predictor for V_i in π^* among a pool of $d \cdot (i-1)$ candidates from the sources.

We use D^*_{tr} to learn a separate target-only model at every position i; in particular, let $\mu^*_i := \hat{P}(v_i \mid v_{1:i-1}; D^*_{\text{tr}})$. Finally, at each position i, we choose a linear interpolation of the best transported predictor and μ^*_i. This is performed via learning the *transport indicators* $s^*_1, \dots, s^*_T \in [0,1]$ through,

$$s^*_i \in \arg\min_{s\in[0,1]} \sum_{v\in D^*_{\text{te}}} \sum_{i=1}^{T} -\log(s \cdot \Psi_{\theta^{\text{src}}}(v_i \mid A^*_{\theta^{\text{trg}}{}_{i,\cdot}} \cdot v_{1:T}; \Phi^*_{\theta^{\text{trg}}}(i)) + (1-s)\cdot \mu^*_i(v_i \mid v_{1:i-1})) \quad (20)$$

In words, $s_i \approx 0$ indicates that the target-only model performs best on the held-out data D^*_{te}, thus deciding on no transport from the sources. On the other hand, $s_i \approx 1$ indicates a decision to transport. Finally, we compute the estimation of the query of interest $P^*(v_t \mid v_{1:M})$:

$$\hat{\mu}_{\text{ft}}(v_T \mid v_{1:M}) = \sum_{v_{M+1:T-1}} \prod_{i=M+1}^{T} s_i \cdot \mu^*_i(v_i \mid v_{1:i-1}) + (1-s_1)\cdot \Psi_{\theta^{\text{src}}}(v_i \mid A^*_{\theta^{\text{trg}}{}_{i,\cdot}} \cdot v_{1:T}; \Phi^*_{\theta^{\text{trg}}}(i)) \quad (21)$$

What follows justifies equivalence of two-stage adaptation with Algorithm 4.

Theorem 4.2 (Fine-tuning rate)**.** *Let μ_{Ag} be learned by Algorithm 4 and μ_{ft} (Equation (21)) be the result of the two-stage adaptation. We have, $R_{P*}(\mu_{\text{ft}}) = \mathcal{O}(R_{P*}(\mu_{\text{Ag}}))$*

4.3 Empirical evaluation

We evaluate the performance of the two-step adaptation in Figure 4 in both multi-cause ($M = 9$) and sequential settings ($M = 3$). Here, $K = 1, T = 10, \mathbf{Pa}^j_i \leqslant 1$, with $\mathcal{V} = \{0, 1, ..., 9\}$, i.e., single source adaptation from 10-token sequences of digits. We have $N = 10^4$ data from source. and the causal diagrams $\mathcal{G}^1, \mathcal{G}^*$ are encoded by $A^{1,\text{true}}, A^{*,\text{true}}$, randomly generated. Each causal mechanism is selected at random from $\mathcal{F} = \{g_{2\times}, g_{\text{copy}}, g_{+1}, g_{-1}, g_{\text{unif}}\}$, which are noisy operators (see Example 2.2). We ensure that the case is transportable by using all mechanisms $\mathcal{F}$ in both source and target. Thus, the structure-informed algorithm would achieve zero-shot generalization. We consider the performance of the two-stage adaptation (as a proxy to structure-agnostic DA in Algorithm 4), and as the baselines we use two variants of ERM on combined source and target data. We keep the architecture (Section 4.1) common between the models to isolate the effect of the adaptation procedure. More evaluations are provided in Appendix E.

5 Conclusions

We proposed a causal framework for supervised domain adaptation, introducing structure-informed and structure-agnostic algorithms. The causal structure enables learning by identifying which components of a model can be reliably transported across domains. Even in the absence of structural knowledge, agnostic procedures can achieve near-optimal performance. Finally, we developed a two-stage learning procedure that is theoretically equivalent to an exhaustive agnostic procedure.

Acknowledgments

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

References

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

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

[3] Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, page 507–556. Association for Computing Machinery, New York, NY, USA, 1st edition, 2022.

[4] Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28(1):7–39, 1997.

[5] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[6] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. doi: 10.1007/s10994-009-5187-8.

[7] Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Proceedings of the 16th Annual Conference on Learning Theory (COLT)*, pages 567–580, 2003.

[8] 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.

[9] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/42e77b63637ab381e8be5f8318cc28a2-Paper.pdf.

[10] John Blitzer, Sham Kakade, and Dean Foster. Domain adaptation with coupled subspaces. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 173–181. PMLR, 2011.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. ISBN 9780521833783.

[12] 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.

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

[14] Alison Gopnik, Clark Glymour, David M. Sobel, Laura E. Schulz, Tamar Kushnir, and David Danks. A theory of causal learning in children: Causal maps and bayes nets. *Psychological Review*, 111(1):3–32, 2004. doi: 10.1037/0033-295X.111.1.3.

[15] Steve Hanneke and Samory Kpotufe. On the value of target data in transfer learning. *NeurIPS*, 2019. URL https://openreview.net/forum?id=SJG4yBHx8S.

[16] Steve Hanneke and Samory Kpotufe. A more unified theory of transfer learning. *arXiv preprint arXiv:2408.16189*, 2024.

[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. doi: 10.1609/aaai.v38i11.29175. URL https://ojs.aaai.org/index.php/AAAI/article/view/29175.

[18] Kasra Jalaldoust, Alexis Bellot, and Elias Bareinboim. Partial transportability for domain generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=2V5LTfhcfd.

[19] Sham M Kakade and Ambuj Tewari. On the generalization ability of online learning algorithms. *IEEE Transactions on Information Theory*, 55(7):2917–2923, 2009.

[20] Masanori Koyama and Shoichiro Yamaguchi. When is invariance useful in an out-of-distribution generalization problem ?, 2021.

[21] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017. doi: 10.1017/S0140525X16001837.

[22] 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.

[23] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009. URL http://dblp.uni-trier.de/db/conf/colt/colt2009.html#MansourMR09.

[24] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *22nd Conference on Learning Theory (COLT)*, 2009.

[25] Gary Marcus. *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. MIT Press, Cambridge, MA, 2001.

[26] Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

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

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

[29] 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.

[30] 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.

[31] Laura E. Schulz and Alison Gopnik. Causal learning across domains. *Developmental Psychology*, 40(2):162–176, 2004. doi: 10.1037/0012-1649.40.2.162.

[32] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge, 2014.

[33] Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, 2011. doi: 10.1126/science.1192788.

[34] V. Vapnik. Principles of risk minimization for learning theory. In J. Moody, S. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991.

[35] Vladimir Vapnik. Statistical learning theory wiley. *New York*, 1(624):2, 1998.

[36] Vladimir N Vapnik and Alexey Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2): 264–280, 1971.

[37] Liu Yang, Steve Hanneke, and Jaime Carbonell. A theory of transfer learning with applications to active learning. *Carnegie Mellon University*, 2010. URL https://www.cs.cmu.edu/~shanneke/papers/a_theory_of_transfer_learning.pdf.

Supplementary Material

A Multi-source domain adaptation in uni-cause case

Consider the classification task where Y takes value in a finite support $\mathcal{Y}$, and the covariate X that takes value in a finite support $\mathcal{X}$; in particular the objective is learning $P^*(Y = y \mid X = x)$ within the hypothesis class containing all functions $\mu(y \mid x) : \mathcal{X} \to \text{simplex}^{|\mathcal{Y}|}$. There is a loss function $\ell(\mu; y, \mathbf{x})$, and the risk is defined as the expected loss $R_{P*}(\mu) := \mathbb{E}_{P*}[\ell(\mu; Y, \mathbf{X})]$. The true risk minimizer is denoted as $\mu_* \in \arg\min_{\mu:\mathcal{X}\to\text{simplex}^{|\mathcal{Y}|}} R_{P*}(\mu)$, and the empirical risk minimizer w.r.t. data D is denoted as,

Algorithm 5 Str.-informed DA ($X \to Y$)

Require: $D^1, D^2, \ldots, D^K, D^*; \Delta$
Ensure: $\mu_{\text{TR}} \approx P^*(y \mid x)$
1: $\mathcal{J} \leftarrow \{j \in [K] \text{ and } Y \notin \Delta_{j,*}\}$
2: $D^{\text{TR}} \leftarrow \bigcup_{j\in[K] \text{ s.t. } Y\notin\Delta_{j,*}} D^j$
3: **Return** $\mu_{\text{TR}} \leftarrow \hat{P}(y \mid x; D^{\text{TR}} \cup D^*)$

$$\hat{P}(y \mid \mathbf{x}; D) \in \underset{\mu:\mathcal{X}\to\text{simplex}^{|\mathcal{Y}|}}{\arg\min} \sum_{y,\mathbf{x}\in D} \ell(\mu; y, \mathbf{x}). \tag{22}$$

We consider the loss to be the negative log-likelihood $\ell(\mu; y, x) := -\log \mu(y \mid \mathbf{x})$ in this work, and the objective is to minimize the excess risk denoted by $R_{P*}(\mu) - R_{P*}(\mu_*)$.

Suppose we have access to target data D^* drawn i.i.d. from the target domain π^* that entails the target distribution $P^*(x, y)$, as well as source data $D^1, D^2, \ldots, D^K$ from a set of source domains $\Pi^{\text{src}} = \{\pi^1, \pi^2, ..., \pi^K\}$ that entail the source distributions $P^{\text{src}} = \{P^1(x, y), P^2(x, y), \ldots, P^K(x, y)\}$. Let $n = |D^*|$ and $N = |D^j|$ for all $j \in [K]$, and suppose $N \gg n$. We assume strictly positive mass for every combination of the variables, i.e., $P^j(x, y) > \epsilon$ for all $j \in [K] \cup \{*\}$.

Example A.1 (Classification in $X \to Y$ case)**.** Suppose the source domains are governed by SCMs $\mathcal{M}^1, \mathcal{M}^2, ..., \mathcal{M}^K$, and the target domain π^* is governed by the SCM $\mathcal{M}^*$; For domain π^j the SCM $\mathcal{M}^j$ is denoted as follow:

$$\begin{aligned} U_X, U_Y &\sim \text{unif}([0, 1]) \\ X &\leftarrow f_X^j(U_X) \\ Y &\leftarrow f_Y^j(X, U_Y). \end{aligned}$$

The source and target SCMs all induce the same causal diagram $X \to Y$, which indicates that X is the cause of Y, and no unobserved confounders are present. Notice that without further assumptions the source data is unrelated to classification in the target; for example, in a case of $X \in \{0, 1\}$ it is possible that $f_Y^1 = f_y^2 = ... = f_Y^K : 1_{\{U_Y>0.5\}}$, which means that $Y \perp\!\!\!\perp X$ across all sources, but $f_Y^*(x, u_Y) = X \oplus 1_{\{U_Y>0.9\}}$, which means that $P^*(Y = 1 \mid X = 0) = 0.1$ and $P^*(Y = 1 \mid X = 1) = 0.9$ in the target domain. □

In the next example we use the domain discrepancy sets in an instance of the domain adaptation task.

Example A.2 (Δ might allow direct-transport, or suggest no transport)**.** In the context of Example A.1, suppose we have access to Δ. If there exists $j \in [K]$ such that $Y \notin \Delta_{j,*}$, then $f_Y^* = f_Y^j$ and $P^*(u_Y) = P^j(u_Y)$, and therefore,

$$\begin{aligned} P^*(y \mid x) &= \sum_{u_Y\in\mathcal{Y}} P^*(y \mid u_Y, x) \cdot P^*(u_Y \mid x) && \text{(introduce } U_Y\text{)} \\ &= \sum_{u_Y\in\mathcal{Y}} 1_{\{f_Y^*(x,u_Y)=y\}} \cdot P^*(u_Y) && \text{(defn. of } f_Y^* \text{ \& } U_Y \perp\!\!\!\perp X\text{)} \\ &= \sum_{u_Y\in\mathcal{Y}} 1_{\{f_Y^j(x,u_Y)=y\}} \cdot P^j(u_Y) = P^j(Y \mid x) && (Y \notin \Delta_{j,*}) \end{aligned}$$

Thus, to predict the label in the target, it suffices to estimate $P^j(Y \mid x)$ using large source data available from π^j. However, if $Y \in \Delta_{j,*}$ for all $j \in [K]$, then domain discrepancy sets reject use of any source data, as the real SCMs may be similar to what was discussed in Example A.1. □

A simple procedure in Algorithm 5 describes the above approach, and below is a guarantee.

Lemma A.3 (Structure-informed DA rate; $X \to Y$ case)**.** *In Algorithm 5 with high probability,*

$$R_{P*}(\mu_{\text{TR}}) - R_{P*}(\mu^*) = \begin{cases} \mathcal{O}(\frac{|\mathcal{X}|\cdot|\mathcal{Y}|}{\epsilon^2\cdot N}) & \text{if } \mathcal{J} \neq \varnothing \\ \Omega(\frac{|\mathcal{X}|\cdot|\mathcal{Y}|}{\epsilon\cdot n}) & \text{otherwise} \end{cases} \tag{23}$$

where $\mathcal{J}$ is obtained through Algorithm 5. □

In words, if the mechanism of Y in the target matches with that of any of the sources, then because of support overlap (due $P(x, y) \geqslant \epsilon$) structure-informed DA can achieve fast rates that depend on the source data size N which is typically large. This case is called *transportable* in the causal inference literature, or domain generalization or zero-shot learning in the literature. In the other cases (i.e., the non-transportable scenario), even with access to the structure Δ, there might exist vastly different realities (i.e., a tuple of source and target SCMs) which admit the structural assumption encoded in Δ but they do not agree on the classification rule in the target domain, thus, no adaptation is possible. These two extreme cases happen as a byproduct of the discrete nature of the domain discrepancy sets; if $Y \notin \Delta_{j,*}$ then the mechanism determining Y matches perfectly between π^*, π^j, and since there is no confounding between Y and X (i.e., U variable pointing to both X and Y), we ensure that $P^*(y \mid x) = P(y \mid x)$. Note that not having a confounder is critical to this conclusion based on the structure Δ; in **??**, we discuss how confounders can complicate structure-informed DA.

Notably, the risk upper-bound provided in Lemma A.3 is not tight, e.g., ϵ^2 in the denominator of the transportable case can be improved through adaptive procedures. Since we are assuming that ϵ is a constant and $N \gg n$, the bounds serve the purpose for this work. Our main focus throughout is identifying which source domains contain useful information for prediction in the target, and how that information can be incorporated in learning, given the structure Δ, thus we rely on the covariates overlap. On the other hand, in many theoretical work on DA, it is presumed that the sources are all useful for prediction in target, e.g., there exists a unique best hypothesis h^* for all domains [8, 24], thus, in these work the main complexity of DA comes from lack of overlap between the covariate distributions across the domains.

We treat the structure-informed DA procedures (such as Algorithm 5) as the best one can do given knowledge of the structural properties of the problem. In reality, access to such structure may not be viable, and in the next example we would like to consider adaptation in situations where Δ is unknown.

Algorithm 6 Structure-agnostic DA ($X \to Y$)

Require: $D^1, D^2, \dots, D^K, D^*$
Ensure: $\mu_{\text{Ag}}(y \mid x) \approx P^*(y \mid x)$
1: $D^*_{\text{tr}}, D^*_{\text{te}} \leftarrow \text{partition}(D^*)$
2: $\psi_{\mathcal{S}} \leftarrow \hat{P}(y \mid x; D^*_{\text{tr}} \cup \bigcup_{j \in \mathcal{S}} D^j)$ for all $\mathcal{S} \subseteq [K]$
3: **Return** $\mu_{\text{Ag}} \leftarrow \arg\min_{\mathcal{S} \subseteq [K]} \ell(\psi_{\mathcal{S}}; D^*_{\text{te}})$

Example A.4 (Agnostic approach; $X \to Y$ case)**.** In the context of Example A.1, suppose we the structure Δ is unknown, yet we would like to achieve guarantees not much worse than what is achievable using Δ. Note that structure-informed DA (Algorithm 5) pools data from the source domain π^j with the target data whenever Δ implies that $P^*(y \mid x) = P^j(y \mid x)$. If any of the source data is pooled, then the error decays with N which is typically very large, otherwise, it would decay with n, the number of target data points.

We can take the following approach to benefit from the source data even without the structure. Partition the target data into $D^* = D^*_{\text{tr}} \sqcup D^*_{\text{te}}$ of equal size $\frac{n}{2}$. Then, for each subset of the sources such as $\mathcal{S} \subseteq [K]$, learn a predictor $\psi_{\mathcal{S}}(y \mid x) := \hat{P}(y \mid x; D^*_{\text{tr}} \cup \bigcup_{j \in \mathcal{S}} D^j)$. Finally, use D^*_{te} to choose the best of the 2^K predictors learned from each of the domains, i.e., $\mu_{\text{Ag}} \leftarrow \arg\max_{\psi_{\mathcal{S}}: \mathcal{S} \subseteq [K]} \sum_{y,x \in D^*_{\text{te}}} \ell(\psi_{\mathcal{S}}; y, x)$. The best error achievable using the structure Δ can be achieved by at least one of the 2^K predictors we have learned. Thus, the extra risk of the above procedure compared to the structure-informed case is equivalent to learning from a finite hypothesis class of size 2^K using D^*_{te} data, which is bounded by $\mathcal{O}(\sqrt{\frac{K}{n}})$. □

Example A.4 shows that through a simple training-validation procedure, without access to the domain discrepancies, it is possible to achieves rates that are only slightly worse than what is achievable through explicit access to the domain discrepancies Δ. We call such strategies *Agnostic* throughout this work, and Algorithm 6 summarizes this approach. What follows is a formal statement.

Theorem A.5 (Structure-agnostic DA rate; $X \to Y$ case)**.** *Let $\mu_{\text{TR}}, \mu_{\text{Ag}}$ be learned using structure-informed and agnostic DA procedures (Algorithms 5 and 6), respectively. We have,*

$$R_{P^*}(\mu_{\text{Ag}}) = \mathcal{O}(R_{P^*}(\mu_{\text{TR}}) + \sqrt{\frac{K}{n}}), \tag{24}$$

where K is the number of source domains, and n is the number of target data. □

B Proofs

Definition B.1 (strongly convex functions)**.** A function $f(x)$ is m-strongly convex if,

$$f(x') \geqslant f(x) + \nabla f(x)^T(x' - x) + \frac{m}{2}\|x' - x\|^2, \quad \forall x, x' \in \mathcal{X}, \tag{25}$$

□

Next, we show that the risk in our problem is strongly convex under strict positivity assumption $P^*(x, y) > \epsilon$.

Lemma B.2 (strongly convex risk.)**.** *$R_{P^*}(\mu)$ is ϵ-strongly convex w.r.t. μ under the assumption of $P^*(x, y) > \epsilon$.*

Proof. Recall the loss function $\ell(\mu; y, x) = -\log \mu(y \mid x)$. Thus, the true risk of μ_{TR} can be expressed as:

$$R_{P^*}(\mu_{\mathrm{TR}}) = \mathbb{E}_{P^*}[-\log \mu_{\mathrm{TR}}(Y \mid X)] \tag{26}$$

$$= \sum_x P^*(x) \cdot \sum_y P^*(y \mid x) \cdot \log \frac{1}{\mu_{\mathrm{TR}}(y \mid x)} \tag{27}$$

$$= \sum_x P^*(x) \cdot D_{\mathrm{KL}}\big(P^*(\cdot \mid x) \| \mu_{\mathrm{TR}}(\cdot \mid x)\big). \tag{28}$$

For a fixed $x \in \mathcal{X}$, the KL loss $D_{\mathrm{KL}}\big(P^*(\cdot \mid x) \| \mu_{\mathrm{TR}}(\cdot \mid x)\big)$ is 1-strongly convex for the interior of the simplex, i.e., under positivity. Thus, the weighted sum $\sum_x P^*(x) \cdot D_{\mathrm{KL}}\big(P^*(\cdot \mid x) \| \mu_{\mathrm{TR}}(\cdot \mid x)\big)$. with $P^*(x) > \epsilon$ would be ϵ-strongly convex [11]. □

What follows is standard high-probability bound for excess risk of ERM with strongly convex risk, adapted from [19].

Corollary B.3. *Let the ERM solution be,*

$$\mu_{\mathrm{ERM}} := \hat{P}(y \mid x; D^*) \in \underset{\mu: \mathcal{X} \to \mathrm{simplex}^{|\mathcal{Y}|}}{\arg\min} \sum_{x,y \in D^*} -\log \mu(y \mid x), \tag{29}$$

and let the true risk minimizer be,

$$\mu_* = \underset{\mu: \mathcal{X} \to \mathrm{simplex}^{|\mathcal{Y}|}}{\arg\min} \mathbb{E}_{Y,X \sim P^*}[-\log \mu(y \mid x)]. \tag{30}$$

Under $P^*(x, y) > \epsilon$, for any $\delta > 0$ the following holds with probability $1 - \delta$:

$$R_{P^*}(\mu_{\mathrm{ERM}}) - R_{P^*}(\mu_*) \leqslant \frac{8 \cdot \left(\ln \frac{1}{\delta} + |\mathcal{X}| \cdot |\mathcal{Y}| \cdot \ln(1 + \frac{n}{|\mathcal{X}| \cdot |\mathcal{Y}|})\right)}{\epsilon \cdot n} = \mathcal{O}(\frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{\epsilon \cdot n}), \tag{31}$$

where $n = |D^*|$. □

We also show the following bound for the risk of the transported estimators.

Lemma B.4. *Suppose $P^*(y \mid x) = P(y \mid x)$. Define the transported predictor as the ERM over $D \sim P(x, y)$:*

$$\mu_{\mathrm{TR}} := \hat{P}(y \mid x; D) \in \underset{\mu: \mathcal{X} \to \mathrm{simplex}^{|\mathcal{Y}|}}{\arg\min} \sum_{x,y \in D} -\log \mu(y \mid x), \tag{32}$$

and let the true risk minimizer be defined in Equation (30). Suppose $P^*(x, y), P(x, y) > \epsilon$. For any $\delta > 0$ the following holds with probability $1 - \delta$:

$$R_{P^*}(\mu_{\mathrm{TR}}) - R_{P^*}(\mu_*) = \mathcal{O}(\frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{\epsilon^2 \cdot N}), \tag{33}$$

where $N = |D|$.

Proof. Equal conditionals, i.e., $P^*(y \mid x) = P(y \mid x)$, implies that the true risk minimizer matches under both distributions, i.e.,

$$\mu_* \in \arg\min_{\mu} \mathbb{E}_{P^*}[\ell(\mu; y, x)] \iff \mu_* \in \arg\min_{\mu} \mathbb{E}_{P}[\ell(\mu; y, x)]. \tag{34}$$

Since μ_{TR} is the solution of ERM under P, based on Corollary B.3, we have:

$$R_P(\mu_{\mathrm{TR}}) - R_P(\mu_*) = \mathcal{O}(\frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{\epsilon \cdot N}). \tag{35}$$

Let $\alpha = \max_{x \in \mathcal{X}} \frac{P^*(x)}{P(x)}$. For any $\mu : \mathcal{X} \to \mathrm{simplex}^{|\mathcal{Y}|}$, we related the risk under P and P^*:

$$\begin{aligned}
R_{P^*}(\mu_{\mathrm{TR}}) - R_{P^*}(\mu_*) &= \sum_x P^*(x) \cdot \sum_y P(y|x) \cdot (\log \mu_*(y|x) - \log \mu_{\mathrm{TR}}(y|x)) && (36)\\
&= \sum_x \frac{P^*(x)}{P(x)} \cdot P(x) \cdot \sum_y P(y|x) \cdot (\log \mu_*(y|x) - \log \mu_{\mathrm{TR}}(y|x)) && (37)\\
&= \mathbb{E}_{(X,Y)\sim P}\left[\frac{P^*(X)}{P(X)} \cdot (\log \mu_*(y|x) - \log \mu_{\mathrm{TR}}(y|x))\right] && (38)\\
&\leqslant \alpha \cdot \mathbb{E}_{(X,Y)\sim P}\left[\log \mu_*(y|x) - \log \mu_{\mathrm{TR}}(y|x)\right] && (39)\\
&= \alpha \cdot \left(R_P(\mu_{\mathrm{TR}}) - R_P(\mu_*)\right) && (40)\\
&= \mathcal{O}(\frac{\alpha \cdot |\mathcal{X}| \cot |\mathcal{Y}|}{\epsilon \cdot N}) = \mathcal{O}(\frac{|\mathcal{X}| \cot |\mathcal{Y}|}{\epsilon^2 \cdot N}). && (41)
\end{aligned}$$

The last line follows from strict positivity:

$$\alpha = \max_{x \in \mathcal{X}} \frac{P^*(x)}{P(x)} \leqslant \frac{\max_{x \in \mathcal{X}} P^*(x)}{\min_{x \in \mathcal{X}} P(x)} \leqslant \frac{1}{\epsilon} \tag{42}$$

□

B.1 Proof of Lemma A.3

If $\mathcal{J} = \varnothing$, then the algorithm learns $\mu_{\mathrm{TR}} \leftarrow \hat{P}(y \mid x; D^*)$, i.e., ERM on the target data only. Followed from Corollary B.3, we obtain the desired guarantee.

If $\mathcal{J} \neq \varnothing$, then there exists at least one source domain $j \in \mathcal{J}$ for which $P^j(y \mid x) = P^*(y \mid x)$, and we transport the predictor from that domain. Since $|D^j| = N$, using Lemma B.4, we obtain the desired result.

B.2 Proof of Theorem A.5

In Algorithm 6 we first compute a collection of predictors $\{\psi_{\mathcal{S}}\}_{\mathcal{S} \subseteq [K]}$. Then we use held-out target data to choose the one with smallest risk. Let,

$$\tilde{\mu} \in \arg\min_{\mu \in \{\psi_{\mathcal{S}}\}_{\mathcal{S} \subseteq [K]}} R_{P^*}(\mu), \tag{43}$$

And let μ_{TR} be obtained from Algorithm 5. Firstly, consider the following cases:

1. $\mathcal{J} = \varnothing$: In this case, μ_{TR} is obtained through ERM on the target data, i.e., $\mu_{\mathrm{TR}} = \hat{P}(y \mid x; D^*)$, achieving the following guarantee (Corollary B.3):

$$R_{P^*}(\mu_{\mathrm{TR}}) - R_{P^*}(\mu_*) = \mathcal{O}(\frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{\epsilon \cdot n}). \tag{44}$$

Notably, for $\mathcal{S} = \varnothing$, we have $\psi_{\mathcal{S}} = \hat{P}(y \mid x; D^*_{\mathrm{tr}})$, where $|D^*_{\mathrm{tr}}| = \frac{n}{2}$. Thus,

$$\begin{aligned}
R_{P^*}(\tilde{\mu}) - R_{P^*}(\mu_*) &\leqslant R_{P^*}(\psi_{\varnothing}) - R_{P^*}(\mu_*) && (45)\\
&= \mathcal{O}(\frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{\epsilon \cdot n}). && (46)
\end{aligned}$$

2. $\mathcal{J} \neq \varnothing$: In this case, μ_{TR} is obtained through ERM on source data from domains π^j for $j \in \mathcal{J}$, which would achieving the following guarantee (Lemma B.4):

$$R_{P*}(\mu_{\text{TR}}) - R_{P*}(\mu_*) = \mathcal{O}(\frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{\epsilon^2 \cdot N}). \tag{47}$$

Notably, for $\mathcal{S} = \mathcal{J}$, we have $\psi_{\mathcal{S}}$ trained using the same pooled data as μ_{TR}, where $|D^*_{\text{tr}}| = \frac{n}{2}$. Thus,

$$R_{P*}(\tilde{\mu}) - R_{P*}(\mu_*) \leqslant R_{P*}(\psi_{\mathcal{J}}) - R_{P*}(\mu_*) \tag{48}$$

$$= \mathcal{O}(\frac{|\mathcal{X}| \cdot |\mathcal{Y}|}{\epsilon^2 \cdot N}). \tag{49}$$

Comapring these rates with Lemma A.3 confirms:

$$R_{P*}(\tilde{\mu}) = \mathcal{O}(R_{P*}(\mu_{\text{TR}})). \tag{50}$$

Next, we show that empirical version of $\tilde{\mu}$, namely μ_{Ag}, achieves the desirable excess compared to $\tilde{\mu}$.

μ_{Ag} in Algorithm 6 is achieved by minimizing the empirical risk over the finite collection $\{\psi_{\mathcal{S}}\}_{\mathcal{S}\subseteq[K]}$ using data D^*_{te} of size $\frac{n}{2}$. Standard uniform convergence guarantees of finite hypothesis classes ([36, 32]) imply that for any $\delta > 0$, with probability $1 - \delta$ the excess risk can be upper-bounded as:

$$R_{P*}(\mu_{\text{Ag}}) - R_{P*}(\tilde{\mu}) \leqslant \sqrt{\frac{\log|\{\psi_{\mathcal{S}}\}_{\mathcal{S}\subseteq[K]}|}{2 \cdot \frac{n}{2}}} + \sqrt{\frac{\log(1/\delta)}{2 \cdot \frac{n}{2}}}. \tag{51}$$

This is due to the fact that $|\{\psi_{\mathcal{S}}\}_{\mathcal{S}\subseteq[K]}| = \mathcal{O}(2^K)$. Finally, Equation (50) implies,

$$R_{P*}(\mu_{\text{Ag}}) \leqslant R_{P*}(\tilde{\mu}) + \sqrt{\frac{\log|\{\psi_{\mathcal{S}}\}_{\mathcal{S}\subseteq[K]}|}{2 \cdot \frac{n}{2}}} + \sqrt{\frac{\log(1/\delta)}{2 \cdot \frac{n}{2}}} \tag{52}$$

$$= \mathcal{O}(R_{P*}(\mu_{\text{TR}}) + \sqrt{\frac{K}{n}}) \tag{53}$$

B.3 Proof of Proposition 2.3

Proof follows the logic of the proof of Lemma A.3 (Appendix B.1). In the transportable case we would have $\mathcal{J} \neq \varnothing$ in Algorithm 1. Thus, the data used for estimation of μ_{TR} is pooled from at least one source domain π^j for $j \in \mathcal{J}$, where $|D^j| = N$. The conditional distribution to be estimated is $\mu_{\text{TR}}(y \mid \mathbf{pa}^*_Y)$, and for $c = |\mathbf{Pa}^*_Y|$, following Lemma B.4, we get the bound,

$$R_{P*}(\mu_{\text{TR}}) - R_{P*}(\mu_*) = \mathcal{O}(\frac{|\mathcal{X}|^c \cdot |\mathcal{Y}|}{\epsilon^2 \cdot N}). \tag{54}$$

On the other hand, in the non-transportable case, we would have $\mathcal{J} = \varnothing$, thus μ_{TR} is trained using only data from the target domain, and due to Corollary B.3, achieves,

$$R_{P*}(\mu_{\text{TR}}) - R_{P*}(\mu_*) = \mathcal{O}(\frac{|\mathcal{X}|^c \cdot |\mathcal{Y}|}{\epsilon \cdot n}). \tag{55}$$

B.4 Proof of Proposition 2.5

The proof follows the logic the proof of Theorem A.5 (Appendix B.2). Algorithm 6 computes the collection of predictors $\mathcal{H}$ by iterating over all possible values of $c = |\mathbf{Pa}^*_Y|$, all possible combinations of the parents across the domains, $\{\mathbf{Pa}^j_Y\}_{j\in[K]\cup\{j\}}$, and all subsets of the source domains $\mathcal{S} \subseteq [K]$. For each combination, we pool the source data from the source domains $\mathcal{S}$, and index them according to the ordered parent sets, and compute the ERM. This process generates at most,

$$|\mathcal{H}| \leqslant 2^K \sum_{c=0}^{M} (c! \cdot \binom{M}{c})^K \leqslant 2^K \cdot M \cdot (M!)^K \leqslant ((2M)^M)^K. \tag{56}$$

distinct predictors. For possible instance of $\Delta, \{\mathbf{Pa}_Y^j\}_{j\in[K]\cup\{*\}}$ there exists a predictor in $\mathcal{H}$ that is learned using the same data as what would have been learned using the knowledge of the structure, namely, μ_{TR}. Thus, the true risk minimizer in $\mathcal{H}$, namely,

$$\tilde{\mu} \in \arg\min_{\mu\in\mathcal{H}} R_{P*}(\mu), \tag{57}$$

achieves the same risk bound as μ_{TR}. Therefore, the empirical risk minimizer in $\mathcal{H}$ computed using the held-out target data D^*_{te}, i.e.,

$$\mu_{\text{Ag}} \in \arg\min_{\mu\in\mathcal{H}} \frac{1}{|D^*_{\text{tr}}|} \cdot \sum_{y,x\in D^*_{\text{te}}} \ell(\mu; y, x), \tag{58}$$

would achieve an excess risk ([36, 32]),

$$R_{P*}(\mu_{\text{Ag}}) - R_{P*}(\tilde{\mu}) \leqslant \mathcal{O}(\frac{\log|\mathcal{H}|}{n}). \tag{59}$$

Finally, since $R_{P*}(\tilde{\mu}) = \mathcal{O}(R_{P*}(\mu_{\text{TR}}))$ as discussed above, we have,

$$R_{P*}(\mu_{\text{Ag}}) \leqslant \mathcal{O}(R_{P*}(\mu_{\text{TR}}) + \frac{K\cdot M\cdot \log M}{n}). \tag{60}$$

B.5 Proof of Theorem 3.3

The query of interest $P^*(v_T \mid v_{1:M})$ can be computed through the following formula by introducing the intermediate variables $V_{M+1:T-1}$ and then marginalizing them out:

$$P^*(v_T \mid v_{1:M}) = \sum_{v_{M+1:T-1}} P^*(v_{M+1:T} \mid v_{1:M}). \tag{61}$$

Following the causal order, and the causal diagram of the target domain, we can write $P^*(v_{M+1:T} \mid v_{1:M})$ as a product of conditionals on the parents:

$$P^*(v_T \mid v_{1:M}) = \sum_{v_{M+1:T-1}} \prod_{i=M+1}^{T} P^*(v_i \mid \mathbf{pa}_i^*). \tag{62}$$

To transport the above, we attempt a multi-cause problem instance at every position i: for $i \in \{M+1, \cdots, K\}$ if there exists a position $i' \in [T]$ and domain index $j' \in [K]$ such that $\Delta(i, *; i', j') = 0$,

$$P^*(V_i = y \mid \mathbf{Pa}_i^* = \mathbf{x}) = P^j(V_{i'} = y \mid \mathbf{Pa}_{i'}^j = \mathbf{x}). \tag{63}$$

This allows us to pool data from all position-domain pairs i', j' such that $\Delta(i, *; i', j') = 0$, and use it to estimate $P^*(V_i = y \mid \mathbf{Pa}_i^* = \mathbf{x})$. In Algorithm 1 $\mathcal{J}_i$ denotes this subset of position-domain pairs. Next, based on the parents in each of the source domains, we pool the data corresponding to $\mathcal{J}_i$, namely D_i^{TR}, with the size of $|\mathcal{J}_i|\cdot N + n$; the n term is due to the fact that $\Delta(i, *; i, *) = 0$ is guaranteed. Next, compute the ERM using D_i^{TR} to obtain $\mu_{\text{TR}}^i := \hat{P}(y \mid \mathbf{x}; D_i^{\text{TR}})$. Finally, we compose all these predictors, and marginalize out the intermediate variables to achieve an estimation of $P^*(v_T \mid v_{1:M})$:

$$\mu_{\text{TR}}(v_T \mid v_{1:M}) = \sum_{v_{M+1:T-1}} \prod_{i=M+1}^{T} \mu_{\text{TR}}^i(Y = v_i \mid \mathbf{X} = \mathbf{pa}^i). \tag{64}$$

Next, we decompose the risk of μ_{TR} in terms of the risk of the predictors different positions:

$$R_{P*}(\mu_{\text{TR}}) = \mathbb{E}_{P*}[-\log \mu_{\text{TR}}(v_T \mid v_{1:M})] \qquad (\ell(\mu; y, \mathbf{x}) = -\log \mu(y \mid \mathbf{x}))$$

(65)

$$= \mathbb{E}_{P*}\big[\mathbb{E}_{P*}[-\log \mu_{\text{TR}}(v_T \mid v_{1:M}) \mid v_{M+1:T-1}]\big] \qquad \text{(law of iterated expectation.)}$$

(66)

$$= \mathbb{E}_{P*}\big[\mathbb{E}_{P*}[-\log \sum_{v_{M+1:T-1}} \prod_{i=M+1}^{T} \mu^i_{\text{TR}}(v_i \mid \mathbf{pa}^*_i) \mid v_{M+1:T-1}]\big] \qquad \text{(intermediate variables.)}$$

(67)

$$\leqslant \mathbb{E}_{P*}\big[\mathbb{E}_{P*}[\sum_{v_{M+1:T-1}} -\log \prod_{i=M+1}^{T} \mu^i_{\text{TR}}(v_i \mid \mathbf{pa}^*_i) \mid v_{M+1:T-1}]\big] \qquad \text{(concavity of log \& Jensen ineq.)}$$

(68)

$$= \mathbb{E}_{P*}\big[\sum_{v_{M+1:T-1}} \sum_{i=M+1}^{T} \mathbb{E}_{P*}[-\log \mu^i_{\text{TR}}(v_i \mid \mathbf{pa}^*_i) \mid v_{M+1:T-1}]\big] \qquad \text{(linearity of expectation)}$$

(69)

$$= \sum_{i=M+1}^{T} \mathbb{E}_{P*}\big[\sum_{v_{M+1:i-1}} \mathbb{E}_{P*}[-\log \mu^i_{\text{TR}}(v_i \mid \mathbf{pa}^*_i) \mid v_{M+1:i}]\big] \qquad \text{(Markovianity)}$$

(70)

$$= \sum_{i=M+1}^{T} \mathbb{E}_{P*}[-\log \mu^i_{\text{TR}}(v_i \mid \mathbf{pa}^*_i)] \qquad \text{(marginalize interm. vars.)}$$

(71)

$$= \sum_{i=M+1}^{T} R_{P*}(\mu^i_{\text{TR}}) \qquad \text{(risk of sub-task)}$$

(72)

Let $\mu^i_* := P^*(v_i \mid \mathbf{pa}^*_i)$, so that,

$$\mu_*(v_t \mid v_{1:M}) = \sum_{v_{M+1:T-1}} \prod_{i=M+1}^{T} \mu^i_*(v_i \mid \mathbf{pa}^*_i). \tag{73}$$

Due to Lemma B.4 and Corollary B.3, we have the following risk bound for the predictor at position i:

$$R_{P*}(\mu^i_{\text{TR}}) - R_{P*}(\mu^i_*) \leqslant \mathcal{O}\Big(\frac{|\mathcal{V}|^{|\mathbf{Pa}^*_i|+1}}{|\mathcal{J}_i| \cdot \epsilon^2 \cdot N + \epsilon \cdot n}\Big). \tag{74}$$

Therefore, we have the bound,

$$R_{P*}(\mu_{\text{TR}}) - R_{P*}(\mu_*) \leqslant \sum_{i=M+1}^{T} R_{P*}(\mu^i_{\text{TR}}) - R_{P*}(\mu^i_*) \tag{75}$$

$$= \mathcal{O}\Big(\sum_{i=M+1}^{T} \frac{|\mathcal{V}|^{|\mathbf{Pa}^*_i|+1}}{|\mathcal{J}_i| \cdot \epsilon^2 \cdot N + \epsilon \cdot n}\Big). \tag{76}$$

Let,

$$\mathcal{I} = \{i \in [T] \text{ s.t. } \mathcal{J}_i \neq \varnothing\}, \tag{77}$$

denote the positions for which $P^*(v_i \mid \mathbf{pa}^*_i)$ is transportable. Also, let $c = \max_{i \in [T]} |\mathbf{Pa}^*_i|$. We have,

$$R_{P*}(\mu_{\text{TR}}) - R_{P*}(\mu_*) = \mathcal{O}\Big(\frac{|\mathcal{I}| \cdot |\mathcal{V}|^{c+1}}{\epsilon^2 \cdot N} + \frac{(T - M - |\mathcal{I}|) \cdot |\mathcal{V}|^{c+1}}{\epsilon \cdot n}\Big). \tag{78}$$

The latter justifies the claim of Theorem 3.3. We discuss the different rates achievable above in Appendix C.

B.6 Proof of Theorem 3.4

The proof follows the logic the proof of Theorem A.5 (Appendix B.2). The structure-informed procedure (Algorithm 3) partitions the position-domain pairs i, j into clusters. The pairs $(i, *)$ each fall into a cluster, and we use the pooled data corresponding to the position-domain pairs in that cluster to estimate $P^*(v_i \mid \mathbf{pa}_j^*)$. In particular, $\mathcal{S} = \{\mathcal{E}_l\}$ in Algorithm 4 denotes these clusters, and we iterate over all of them. Next, we consider all combination of causal diagrams for the source and target domains; this allows the structure-informed procedure to match the scope of the parents across the domains. For each combination above that corresponds to a selection diagram (i.e., $\Delta, \{\mathcal{G}^j\}_{j\in[K]\cup\{*\}}$), we use Algorithm 3 as a subroutine (StrInf) to compute a an estimation of $P^*(v_T \mid v_{1:M})$.

Notably, for all possible structures encoded as $\Delta, \{\mathcal{G}^j\}_{j\in[K]\cup\{*\}}$, we have a candidate in $\mathcal{H}$. Thus, the rate achieved by the structure-informed procedure is matched by the minimum risk in $\mathcal{H}$, i.e.,

$$R_{P^*}(\tilde{\mu} \in \arg\min_{\mu\in\mathcal{H}} R_{P^*}(\mu)) = \mathcal{O}(R_{P^*}(\mu_{\text{TR}})). \tag{79}$$

Computing $\tilde{\mu}$ is only possible with large target data, however, we can compute an empirical risk minimzer within $\mathcal{H}$ using held-out target data to achieve similar rates. Let,

$$\mu_{\text{Ag}} \in \arg\min_{\mu\in\mathcal{H}} \frac{1}{|D^*_{\text{te}}|} \cdot \sum_{x,y\in D^*_{\text{te}}} \ell(\mu; y, x). \tag{80}$$

In computing μ_{Ag} we used held-out target data D^*_{te} of size $\frac{n}{2}$, thus, we have,

$$R_{P^*}(\mu_{\text{Ag}}) - R_{P^*}(\tilde{\mu}) = \mathcal{O}(\sqrt{\frac{\log|\mathcal{H}|}{n}}). \tag{81}$$

We can bound the size of $\mathcal{H}$ as,

$$|\mathcal{H}| \leqslant \underbrace{(KT)^{KT}}_{\text{different partitions } \mathcal{S}} \cdot \underbrace{((2T)!)^{T^{K+1}}}_{\text{causal diagrams for all domains}} \tag{82}$$

which gives,

$$\log|\mathcal{H}| \leqslant KT \cdot (\log K + \log T) + (K+1)T \cdot T\log T = \mathcal{O}(KT^3 \log T). \tag{83}$$

The latter justifies the claim of Theorem 3.4:

$$R_{P^*}(\mu_{\text{Ag}}) = \mathcal{O}(R_{P^*}(\mu_{\text{TR}}) + \sqrt{\frac{K \cdot T^3 \log T}{n}}). \tag{84}$$

B.7 Proof of Theorem 4.1

Define,

$$\mu_\theta(v_i \mid v_{1:i-1}; j) := \Psi_\theta(v_i \mid A^j_{\theta_{i,\cdot}} \cdot v_{1:T}; \Phi_\theta(i, j)). \tag{85}$$

We can rewrite the objective of Equation (16) as,

$$\mathcal{L}(\theta) := \lambda \cdot \underbrace{(d_\theta + \sum_{j=1}^{K}\sum_{i,i'=1}^{T} A^j_{\theta_{i,i'}})}_{\text{penalty}} + \underbrace{\sum_{j=1}^{K}\sum_{i=1}^{T} \mathbb{E}_{P^j}\left[D_{\text{KL}}\left(P^j(\cdot \mid V_{1:i-1}) \| \mu_\theta(\cdot \mid V_{1:i-1}; j)\right)\right]}_{\text{match with source distributions}}. \tag{86}$$

In words, the objective in Equation (16) ensures that the solution entails a distribution that matches the sources at all conditionals and all domains, while preferring parameters with smaller mechanism indicator range d and fewer edges in the graphs encoded by $\{A^j\}$.

The score can be decomposed into K objectives as follows:

$$\mathcal{L}(\theta) = \lambda \cdot d_\theta + \sum_{j=1}^{K} \mathcal{L}_j(\theta), \tag{87}$$

where,

$$\mathcal{L}_j(\theta) = \lambda \cdot \|A^j_\theta\| + \sum_{i=1}^{T} \mathbb{E}_{P^j}\big[D_{\mathrm{KL}}\big(P^j(\cdot \mid V_{1:i-1}) \| \mu_\theta(\cdot \mid V_{1:i-1}; j)\big)\big]. \tag{88}$$

For small enough $\lambda > 0$, maximizing $\mathcal{L}_j(\theta)$ ensures that the parent matrix A^j has a one entry at the position of the true parents in each row, since $P^j(\cdot \mid v_{1:i-1}) = P^j(\cdot \mid \mathbf{pa}^j_i)$. Also, the penalty $\lambda \cdot \|A^j\|$ ensures that no additional one entries are kept in the parent matrix, thus, Equation (14) would be satisfied. Recall that d is the size of the range of the mechanism indicator mapping $\Phi : [T] \times [K] \to [d]$.

If $\Delta(i, j; i', j') = 0$, we would have $P^j(v_i \mid \mathbf{pa}^j_i) = P^{j'}(v_{i'} \mid \mathbf{pa}^{j'}_{i'})$. To satisfy Equation (13), the mechanism indicator $\Phi : [T] \times [K] \to [d]$ must map (i, j) and (i', j') to the same value in the range $[d]$, which makes $\mu_\theta(v_i \mid \mathbf{pa}^j_i; j) = \mu_\theta(v_{i'} \mid \mathbf{pa}^{j'}_{i'}; j)$. By minimizing $\lambda \cdot d$, we ensure that this happens, satisfying Equation (13). Finally, note that once Equations (13) and (14) are satisfied, minimizing the divergence between the true distribution $P^*(v_i \mid v_{1:i-1})$ and $\mu_\theta(v_i \mid v_{1:i-1}; j)$ occurs only when Equation (15) is satisfied.

B.8 Proof of Theorem 4.2

The parameters to be learned in fine-tuning stage are:

1. The target mechanism indicator $\Phi^* : [T] \to [d_{\theta^{\mathrm{src}}}]$.
2. The target parent matrix $A^* \in [0, 1]^{T \times T}$.
3. The target-only predictors $\mu^*_i(v_i \mid v_{1:i-1})$.
4. The transport indicators $s_1, ..., s_T \in [0, 1]$

Once the pretrained parameters θ^{src} satisfy Equations (13) to (15), consider the following values for the parameters of fine-tuning stage: Let A^* encode the true causal diagram $\mathcal{G}^*$, and for the transported conditionals $P^*(v_i \mid \mathbf{pa}^*_i)$ we set $s_i = 1$, and $\Phi^*(i) = \Phi(i', j')$ for some (i', j') which satisfies $\Delta(i, *; i', j') = 0$. For the non-transportable conditionals, we set $s_i = 0$ to use $\mu^*_i(v_i \mid v_{1:i-1})$ that is trained using the target data D^*_{tr} of size proportionate to n. Let $\tilde{\theta}$ encode the parameters for this assignment of the fine-tuning parameters. We have,

$$R_{P^*}(\mu^{\tilde{\theta}}_{\mathrm{ft}}(v_T \mid v_{1:M}) = \mathcal{O}(R_{P^*}(\mu_{\mathrm{TR}}(v_T \mid v_{1:M})), \tag{89}$$

Since this set of values for the parameters corresponds to the structure-informed solution. We discretize the fine-tuning parameter space into a set $\mathcal{H}$ of points , and consider only binary parent matrices and binary transport indicators. We can ensure that $\tilde{\theta}$ lies on this grid, among

$$|\mathcal{H}| \leqslant \underbrace{T^2}_{\text{parent matrix}} \cdot \underbrace{(KT)^T}_{\text{target mech. ind.}} \cdot \underbrace{2^T}_{\text{TR indicator}} \tag{90}$$

We use the held-out target data to obtain the best of these candidates,

$$\theta^* \in \arg\min_{\theta \in \mathcal{H}} \frac{1}{|D^*_{\mathrm{te}}|} \cdot \sum_{v_T, v_{1:M} \in D^*_{\mathrm{te}}} -\log \mu^{\theta}_{\mathrm{ft}}(v_T \mid v_{1:T}) \tag{91}$$

Compared to the best in class parameter set $\tilde{\theta}$, we would have an excess risk bounded as,

$$R_{P^*}(\mu^{\theta^*}_{\mathrm{ft}}) - R_{P^*}(\mu^{\tilde{\theta}}_{\mathrm{ft}}) = \mathcal{O}(\sqrt{\frac{\log |\mathcal{H}|}{n}}) = \mathcal{O}(\sqrt{\frac{T \cdot (\log K + \log T)}{n}}). \tag{92}$$

This proves,

$$R_{P^*}(\mu^{\theta^*}_{\mathrm{ft}}) = \mathcal{O}(R_{P^*}(\mu_{\mathrm{TR}}) + \sqrt{\frac{T \cdot (\log K + \log T)}{n}}) \tag{93}$$

$$= \mathcal{O}(R_{P^*}(\mu_{\mathrm{TR}}) + \sqrt{\frac{K \cdot T^3 \cdot \log T}{n}}) = \mathcal{O}(R_{P^*}(\mu_{\mathrm{Ag}})) \tag{94}$$

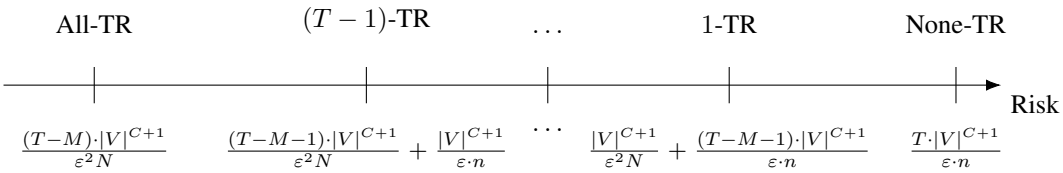


Figure 5: A schematic of risks obtained via the structure-informed procedure (Algorithm 3). In cases where all conditionals $P^*(v_i \mid v_{1:i-1})$ are transportable, we obtain a rate proportionate to $\frac{T-M}{\epsilon\cdot N}$. As more and more conditionals are non-transportable, they need to be estimated from the target data, adding a cost proportionate to $\frac{1}{n}$ for every non-transportable term. In the extreme case that no conditional is transportable, i.e., all target mechanisms are novel, we incur a risk proportionate to $\frac{T}{n}$.

C More details on structure-informed rates

In this section, we expand upon the possible rates in sequence adaptation via the structure-informed procedure. Please view the proof of Theorem 3.3 (Appendix B.5). We reduce the problem of estimating $P^*(v_T \mid v_{1:M})$ to estimating,

$$P^*(v_{M+1:T} \mid v_{1:M}) = \prod_{i=M+1}^{T} P^*(v_i \mid v_{1:i-1}). \tag{95}$$

Each of the conditionals, is either transported from a source domain (if there exists (i', j') such that $\Delta(i, *; i', j') = 0$), or is estimated from the target data alone. In the former, the excess risk associated to estimation of $P^*(v_i \mid v_{1:i-1})$ would be $\frac{|\mathcal{V}|^{c+1}}{\epsilon^2\cdot N}$, which is desirable since N is large, and in the latter, the risk would be bounded by $\frac{|\mathcal{V}|^{c+1}}{\epsilon\cdot n}$. The joint risk depends on how many of the $T - M$ components are transported, and how many must be estimated from the target data. This gives a variety of rates, shown in Figure 5. Notably, if $N \gg n$, then we achieve a fast rate only if all components are transported, but achieve slower and slower rates for more and more non-transportable components. This figure is informative in the case of structure-agnostic adaptation as well, since due to Theorem 3.4 the risk of the structure-agnostic method is bounded by a fixed margin of $\sqrt{\frac{K\cdot T^3\cdot\log T}{n}}$ compared to these rates, which is independent from the size of the vocabulary $\mathcal{V}$.

D Detailed model architecture

This section provides a comprehensive walkthrough of our model architecture, focusing on how multiple parents are identified and utilized for next-token prediction in a domain-adaptive manner.

Overall task and core design principle

The objective is sequence modeling, specifically to predict the next token in a sequence of discrete symbols (digits 0–9 in our experiments). The core idea is that the generation of a token at position i depends on:

1. A selected **causal function** (e.g., add, subtract, multiply)
2. One or more **parent tokens** from earlier positions in the sequence (i.e., positions $< i$)

Input representation and positional encoding

Input sequences: Sequences of integer token IDs from $\mathcal{V} = \{0, 1, \ldots, 9\}$.

Positional encoding (`PositionalEncoding` class): Each token at position (i, j) where i is the sequence position and j is the domain ID is mapped to a dense vector representation using:

- Standard sinusoidal positional encodings for both the position index (0 to $T - 1$) and the domain ID (0 to K)

- These two embeddings (each $r/2$ dimensional) are concatenated to form the initial r-dimensional embedding
- Input: (B, T) for positions and domains; Output: (B, T, d) embeddings, where B is the batch size.

This design allows the model to be aware of both absolute position within the sequence and the domain context, enabling domain-specific parent selection as described in the following sections.

Universal operator indicator

The `UniversalOperatorIndicator` class determines, for each token position, the probability distribution over a fixed set of operations $\mathcal{F}$ (e.g., add, subtract, multiply_two).

Mechanism:

1. The h-dimensional embedding of each token is passed through a linear projection to $|\mathcal{F}|$ dimensions
2. Layer normalization is applied: LayerNorm(Linear(embedding))
3. Softmax produces a probability distribution: softmax(LayerNorm(Linear(embedding)))

Universality: This module's parameters are shared across all domains and positions. The *choice* of operation is contextual based on the token's embedding, but the *meaning* of each operation is universal across domains.

Output: $(B, T, |\mathcal{F}|)$ operator probabilities where B is batch size.

Domain-specific parent selector

The `DomainSpecificParentSelector` class implements the core mechanism for identifying influential parent tokens, corresponding to the causal structure learning described in Algorithm 4.

Multi-head causal attention design: The mechanism uses C distinct attention heads (where C is the maximum number of parents). For our experiments, $C = 2$, meaning the model can identify up to two distinct parents for each token.

Domain-specificity: The key innovation is that query (Que) and key (Key) projection matrices are domain-specific:

$$\text{domain\_queries} \in \mathbb{R}^{(K+1)\times C\times r\times r} \tag{96}$$

$$\text{domain\_keys} \in \mathbb{R}^{(K+1)\times C\times r\times r} \tag{97}$$

During forward pass, for domain j and parent head h:

$$\text{Que}_{j,h} = \text{Embeddings} \cdot W^{\text{Que}}_{j,h} \tag{98}$$

$$\text{Key}_{j,h} = \text{Embeddings} \cdot W^{\text{Key}}_{j,h} \tag{99}$$

Parent selection process: For each domain j, parent head h:

1. **Attention scores**: $S_{j,h} = \frac{\text{Que}_{j,h}\text{Key}^T_{j,h}}{\sqrt{r}}$
2. **Causal masking**: Standard causal attention masking ensures token at position i only attends to positions $< i$
3. **Sharp softmax**: $A_{j,h} = \text{softmax}(S_{j,h}/\tau)$ where $\tau = 0.1$
4. **First position handling**: Weights for position 0 are zeroed as it has no parents

The temperature $\tau = 0.1$ makes the softmax significantly sharper, encouraging sparse selection of a small number of parents rather than soft averaging.

Output: For each domain j, a list of C attention weight matrices (B, T, T) representing parent selection distributions.

Feature preparation for conditional MLP

The `_prepare_mlp_features` method combines operator indicators and selected parent values to form input for the final prediction MLP.

Input components:

- sequences_onehot $\in \{0,1\}^{B\times T\times|\mathcal{V}|}$: One-hot encoded input sequences
- operator_indicators $\in [0,1]^{B\times T\times|\mathcal{F}|}$: From universal operator indicator
- parent_weights: From domain-specific parent selector
- domains $\in \{0,\ldots,K\}^{B\times T}$: Domain IDs

Feature construction: For each position i:

1. Operator indicators operator_indicators$[:,p,:]$ form the first part of the feature vector
2. For each parent head $h \in \{0,\ldots,C-1\}$:

$$\text{WeightedParentValue}_{h,p} = A_{j,h}[:,p,:] \cdot \text{sequences\_onehot} \tag{100}$$

   where j is the domain of position i. This produces a $(B,|\mathcal{V}|)$ vector for each parent head.
3. These C vectors are concatenated after the operator indicators

Feature dimension: $|\mathcal{F}| + (C \times |\mathcal{V}|)$ where $|\mathcal{F}|$ is the number of operations and $|\mathcal{V}| = 10$.

Conditional MLP for prediction

The `EfficientConditionalMLP` class predicts the next token's probability distribution based on the combined features, implementing the learned conditional distributions $\hat{P}(v_i|\text{pa}_i)$ from our theoretical framework.

Architecture:

1. **Input projection**: Linear layer from feature dimension to r
2. **Hidden layers**: Stack of linear layers ($r \to r$) with ReLU activations, dropout, and residual connections
3. **Output layer**: Linear layer from r to $|\mathcal{V}| = 10$

Output: Logits of shape $(B,T,|\mathcal{V}|)$ for next-token prediction.

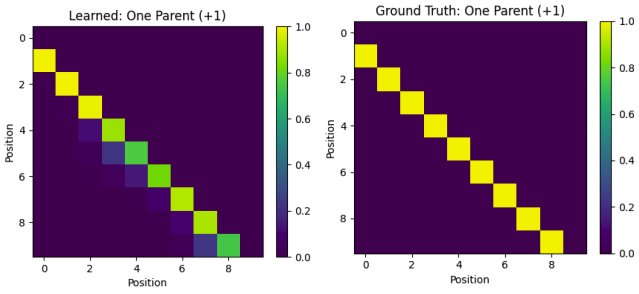
Training and fine-tuning protocol

Pre-training (source domains): The entire model, including domain-specific Que/Key matrices for source domains, is trained end-to-end using standard next-token prediction cross-entropy loss.

Fine-tuning (target domain adaptation): When adapting to target domain π^*:

- **Frozen components**:
  - Positional encoding parameters
  - Universal operator indicator parameters
  - Conditional MLP parameters
  - Source domain Que/Key matrices
- **Trainable components**:
  - New randomly initialized $W^{\text{Que}}_{*,h}, W^{\text{Key}}_{*,h}$ for target domain
  - New target-specific operator indicator

This modular design enables learning domain-specific parent selection while reusing universal causal functions, corresponding to the structure-agnostic adaptation strategy in Algorithm 4 with computational efficiency discussed in Section 4.



(a) Learned parents matrix (b) True parent matrix

Figure 6: The chain graph.

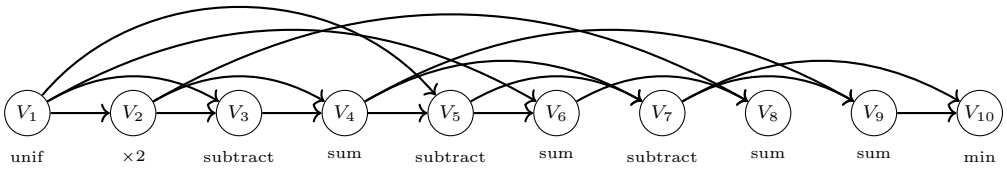


Figure 7: causal diagram and operators corresponding to the target domain.

E Experimental setup and reproducibility

The setting considered in Section 4.3 involves a source domain and a target domain with sequences of length 10, where variables follow a chain, i.e., $\mathbf{Pa}_i = V_{i-1}$, but for the first variable V_1 which has no parents and is generated at random. Investigation of the parameters from pretraining reveal that the parent matrix learned matches the underlying structure, as shown in Figure 6.

To handle more than one parent, we follow the design discussed in Appendix D. We experiment in settings where each variable has at most two parents randomly selected from the previous variables in the causal order. Note that the causal diagram of the source and target does not match necessarily, and is decided independently. The modules that determine the value of the variables are also drawn at random, from a pool containing null-ary, unary and binary noisy operators;

$$\mathcal{F} = \{g_{\text{unif}}, g_{\text{copy}}, g_{+1}, g_{-1}, g_{\times 2}, g_{\text{sum}}, g_{\min}, g_{\text{subtract}}, g_{\text{mult}}\}. \tag{101}$$

Such structure for $T = 10$ is shown in Figures 7 and 8 for a source domain. Next, we investigate pretraining and fine-tuning in these context of this SCMs.

E.1 Pretraining discovers causal structure

Firstly, we emphasize that in pretraining, as discussed in Section 4 and further in Appendix D, the underlying causal structure is discoverable once we train with large source data. In particular, the operation indicators align for position-domain pairs with matching causal mechanism across the sources, and the parent matrices are learned too. In Figure 9 we see the parent matrices learned in

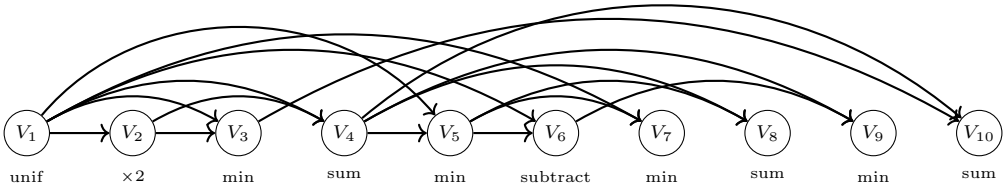


Figure 8: Causal diagram and operators corresponding to the source domain.

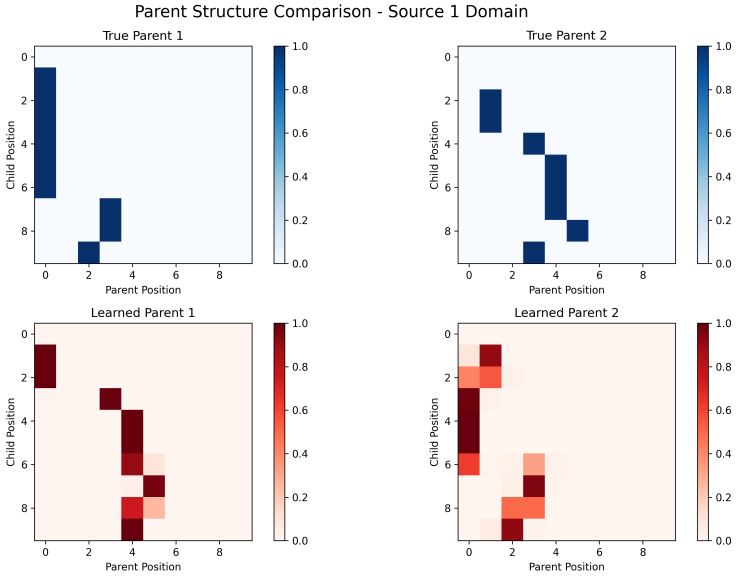
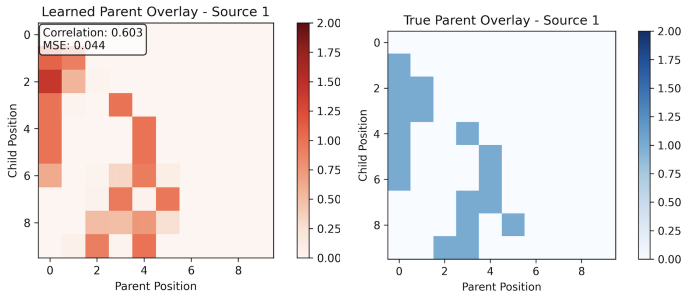


Figure 9: individual parent matrices of the source, generated by key/query pairs for each position in each domain.



(a) Learned parents (sum of parent matrices)

(b) True parents (sum of parent matrices)

Figure 10: Overlaying the parent matrices

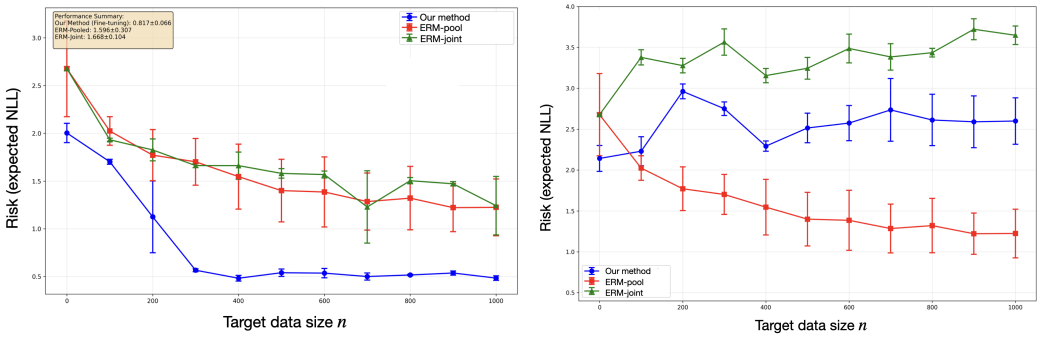
pretraining and the true parent matrices (source only). Since some operators like summation are symmetrical, the learned parent might mismatch in order with the true ones. Considering the overlaid matrices shown in Figure 10, we witness that the causal diagram is learned. It is worth to mention that in the context of transformers, [26] discovers that the transformer architecture learns such sequential dependencies, however, it is unclear whether complex architectures like transformer can be tied to the true causal mechanisms generating the data. Here, we show that this architecture captures not only the causal dependencies, but also mechanism match/mismatches across the domains, allowing it to better adapt to under-sampled domains.

E.2 Fine-tuning exhibits fast and slow adaptation

We consider our method in comparison with two baselines:

1. **ERM-pool.** We pool the source and target data together, dropping the domain indices, and treat them as a single domain.
2. **ERM-joint.** We keep the domain indices, and train the model with source and target data simultaneously, treating the target as another source in pretraining.

We use the same architecture of our method in both baselines to avoid discrepancies due to architecture, and isolate/emphasize the effect of our adaptation procedure.



(a) Adapting fast due to process supervision

(b) Adapting slow due to no process supervision

Figure 11: The performance of our method which is based on structure agnostic domain adaptation, in comparison with the baselines that train jointly on source and target data, either by discarding the domain indices (ERM-pool) or by keeping them (ERM-joint).

The objective is learning $P^*(v_{10} \mid v_{1:5})$. By tracing back the operators in Figure 7, we can deduce,

$$V_{10} \approx min(V_4 - V_5, 2 \times V_4 - V_5). \tag{102}$$

This operator is binary, using only V_4, V_5, and is the best predictor for V_{10}. However, it is not used at any position in the source sequence, thus can not be transported directly even with structural knowledge. However, the components that make it are shared between the source and target across different positions, and we have learned them in pretraining. In fine-tuning, the task of re-learning the composition is much simpler than re-learning the circuits themselves, as shown in Theorems 3.3 and 3.4. Figure 11a shows the performance of different methods in this task; our method achieves a small risk faster than the baselines.

Also, we tried the same task of learning $P^*(v_{10} \mid v_{1:5})$ but with hiding the intermediate tokens V_6, V_7, V_8, V_9 in the target data, and forcing the models to generate auto-regressively to predict the last token of the sequence based on the first 5 tokens. The results are shown in figure Figure 11b. In this case, all methods struggle to converge; our method and ERM-joint perform even worse than ERM-pool, due to higher model complexity. This observation emphasizes the importance of *process supervision* in sequence learning; access to intermediate token might enable structure-informed adaptation, which in turn allows a structure-agnostic method to also benefit from the unknown structure.

E.3 Reproducibility

Data generation We evaluate our approach on a synthetic arithmetic benchmark where each sequence represents a functional program executed on base-10 digits.

Model architecture The `DomainAdaptationModel` uses the following specification:

- **Hidden dimension**: $r = 128$ (`config.hidden_dim`)
- **Positional encoding**: Learned embeddings of length T and dimension r
- **Universal operator-indicator**: 2-layer MLP: $r \to d \to |\mathcal{F}|$ plus LayerNorm
- **Parent selector**: $H = 4$ causal attention heads with sharp-softmax temperature $\tau = 0.1$
- **Conditional MLP (token head)**: $2 \times (\text{Linear} + \text{ReLU}) + \text{LayerNorm}$, output size $|\mathcal{V}| = 10$
- **Maximum parents per head**: $C = 4$ (`config.max_parents`)
- **Parameter counts** (for $T = 20$): 43,868 total parameters, 20,946 trainable during fine-tuning
- **Activation dtype**: `float32` (no automatic mixed precision)

The **target-domain adapter** (our method) learns:

- New parent queries/keys (shape $[H, r, r]$ each)
- New operator-indicator MLP ($d \to d \to |\mathcal{F}|$)
- **Freezes**: positional encoding, base operator MLP, parent selector of source domain(s), and conditional MLP

Training hyperparameters **Global optimizer**: AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, weight decay 0.01, batch size 32 sequences ($\to 32 \cdot (T-1)$ tokens), no gradient clipping, constant learning rate schedule.

Pre-training (source only):

- Epochs: 150
- Learning rate: 10^{-3}
- Data: All source sequences (domain id 1)

Fine-tuning (our method - adapter only):

- Epochs: 15
- Learning rate: 10^{-3}
- Prefix length: $M = T/2$ (default, override with `-M`)
- Supervision modes:
  - **PS (process supervision)**: Full next-token cross-entropy
  - **NPS (no process supervision)**: Mask positions $\geqslant M - 1$

Baseline configurations:

- **ERM-Pooled**: Stage 1 (source only): 50 epochs, lr 10^{-3}; Stage 2 (add target): 15 epochs, lr 5×10^{-4}, all parameters trainable
- **ERM-Joint**: Start from source pre-trained model, 20 epochs, lr 5×10^{-4}, all parameters trainable
- **Early stopping**: None

Loss masking (NPS) In no-process-supervision (NPS) experiments, the cross-entropy at positions $\geqslant M - 1$ is excluded:

$$\text{mask}[i] = \begin{cases} 1 & \text{for } 0 \leqslant i < M - 1 \\ 0 & \text{for } i \geqslant M - 1 \end{cases} \tag{103}$$

Implemented in `build_loss_mask()` and applied on GPU.

Evaluation protocol

- **Input**: First M tokens of target sequence where $M \in \{T-1, T/2\}$
- **Generation**: Autoregressively sample positions $M, \ldots, T-2$ by greedy argmax; keep logits for final position
- **Metric**: CE(logits$_{T-1}$, target digit) averaged over 1000 held-out target sequences
- **Seeds**: 3 independent runs (seeds $0, 1, 2$) with mean $\pm$ std reported in plots

Computational environment

- **Hardware**: NVIDIA H100 80GB (PCIe), CUDA 12.1, driver 535
- **Software**: PyTorch 2.1.0, Python 3.12, numpy 1.26
- **Determinism**: `torch.use_deterministic_algorithms(True)` and `torch.backends.cudnn.deterministic = True`
- **Typical runtimes**: $T = 10$, $N = 10^4$, $3 \times 3 \times 2$ grid ≈ 6 minutes; $T = 20$, $N = 10^5$ pre-training ≈ 90 minutes
- **Peak GPU memory**: < 4GB per worker