Proceedings of Machine Learning Research vol 275:731–756, 2025 4th Conference on Causal Learning and Reasoning

Transfer Learning in Latent Contextual Bandits with Covariate Shift Through Causal Transportability

Mingwei Deng MINGWEI.DENG@AALTO.FI
Ville Kyrki VILLE.KYRKI@AALTO.FI
Dominik Baumann DOMINIK.BAUMANN@AALTO.FI
Department of Electrical Engineering and Automation
Aalto University
Espoo, Finland

Editors: Biwei Huang and Mathias Drton

Abstract

Transferring knowledge from one environment to another is an essential ability of intelligent systems. Nevertheless, when two environments are different, naively transferring all knowledge may deteriorate the performance, a phenomenon known as *negative transfer*. In this paper, we address this issue within the framework of multi-armed bandits from the perspective of causal inference. Specifically, we consider transfer learning in *latent contextual bandits*, where the actual context is hidden, but a potentially high-dimensional proxy is observable. We further consider a *covariate shift* in the context across environments. We show that naively transferring all knowledge for classical bandit algorithms in this setting led to negative transfer. We then leverage *transportability* theory from causal inference to develop algorithms that explicitly transfer effective knowledge for estimating the causal effects of interest in the target environment. Besides, we utilize variational autoencoders to approximate causal effects under the presence of a high-dimensional proxy. We test our algorithms on synthetic and semi-synthetic datasets, empirically demonstrating consistently improved learning efficiency across different proxies compared to baseline algorithms, showing the effectiveness of our causal framework in transferring knowledge.

Keywords: Transfer learning, transportability, contextual bandits, causal inference

1. Introduction

Consider an autonomous vehicle equipped with several pre-trained driving policies, each optimized for different weather conditions. We want to train an agent to select the driving policy that maximizes the expected cumulative reward based on camera images. Thus, it must infer the relevant weather conditions from these high-dimensional images. Typically, this would require extensive data collection from the environment, which, in real-world experiments with the car, is time-consuming and costly (Zhang et al., 2023). Therefore, data-efficient algorithms are needed. The car example can be formulated as a latent contextual bandit problem, where the agent only has access to a high-dimensional proxy variable instead of the actual context.

One approach to improve the sample efficiency is transfer learning, which allows the agent to reuse knowledge gained from a related environment to accelerate the learning process in the target environment. The expectation is that the transferred knowledge can serve as a warm start for the agent to converge to the optimal policy (e.g., the optimal selection of driving policy in the example above) faster. However, transferring knowledge may lead to *negative transfer* (Wang et al., 2019) if the environment difference is too large. For instance, if we reuse the knowledge of driving a car

from a dataset collected in the summer but want to train a model to drive in the winter, the model may initially be too aggressive for the winter conditions.

Unlike earlier works on transfer learning in bandit problems [1] (Deshmukh et al., 2017; Zhang and Bareinboim, 2017; Liu et al., 2018; Cai et al., 2024; Bellot et al., 2024), this paper focuses on tackling the challenge of negative transfer in *latent contextual bandits* (Zhou and Brunskill, 2016; Sen et al., 2017). Specifically, the objective is to define a contextual bandit agent that can improve sample efficiency by exploiting a dataset collected in another environment, given the awareness of the potential environmental discrepancy, while avoiding negative transfer. In causal inference, the study of *transportability* (Pearl and Bareinboim, 2011; Bareinboim and Pearl, 2012, 2016) focuses on determining what knowledge can be transferred from one environment to infer causal effects in another, where the two environments are assumed to be different. Our approach employs the *transport formula* derived from the transportability theory to explicitly determine the knowledge we can transfer from the data. Furthermore, to address the high-dimensional proxy, our approach builds on *Variational Autoencoders (VAEs)*, which are well-suited for capturing the structure of latent variables across various high-dimensional datasets (Hu et al., 2017; Lin et al., 2020; Peng et al., 2021) and infer the causal effect of interventions (Louizos et al., 2017).

Contributions. We propose a novel latent contextual bandit learning algorithm that can transfer knowledge across different environments based on the transport formula and improve sample efficiency. We test our method on several synthetic and semi-synthetic datasets, including settings where images serve as high-dimensional proxies. Our results show that the proposed method successfully avoids negative transfer compared to methods that naively extract knowledge from the data collected in another environment. Also, it consistently achieves higher sample efficiency over methods that do not consider any prior data.

2. Problem Formulation and Background

We consider a transfer learning setting for a latent contextual bandit problem, in which an agent seeks to minimize its cumulative regret in the target environment given a dataset collected from another environment. In this section, we introduce the concrete problem setting and necessary background to build our method. Then, based on the theory of transportability, we formalize the problem.

Throughout the paper, we use capital letters to denote random variables and small letters to denote their realizations, e.g., $X = x$. For sets of random variables, we use bold capital and small letters to denote them and their realizations, e.g., $\boldsymbol{X} = \boldsymbol{x}$. A distribution of a random variable Y is denoted by P_Y, and its conditional distribution is denoted by $P_{Y|x}$, an abbreviation of $P_{Y|X=x}$. The probability of a specific value x taken by X is denoted by $p(x)$, an abbreviation of $p(X = x)$.

2.1. Problem Setting

We start by considering a contextual bandit problem in which the true context variable is latent, but we have access to its proxy. Specifically, let the context variable be $Z \in \mathcal{Z} \subseteq \mathbb{R}^{d_1}$ and the proxy variable $W \in \mathcal{W} \subseteq \mathbb{R}^{d_2}$, where $d_1 \ll d_2$. At time step t, the environment samples z_t from a marginal distribution P_Z and w_t from a conditional distribution $P_{W|z_t}$, but only w_t is revealed to the agent. From the observed w_t, the agent is required to select an arm $x_t \in \mathcal{X} = \{1, \dots, K\}$.

1. A more detailed review of related work is provided in Appendix A.

In return, it receives a reward $y_t \in \mathcal{Y} \subseteq \mathbb{R}$. Different from the standard contextual bandit setting (Langford and Zhang, 2007), where the reward is sampled from a conditional distribution $P_{Y|w_t,x_t}$, the reward y_t here is drawn from $P_{Y|z_t,x_t}$. Let the optimal reward at step t be y_t^*. The objective of the agent is to minimize the expected cumulative regret

$$\mathbb{E}[R_T] = \sum_{t=0}^{T} \mathbb{E}[y_t^*] - \mathbb{E}[y_t], \tag{1}$$

where T is the total number of time steps.

We then extend the problem to a transfer learning setting, where we have two domains: the source domain π and the target domain π^*. The agent is given a fixed prior dataset collected from π. It includes W, Z, X, Y, and the arm X is selected based on Z. Formally, we denote the dataset to be $D := \{(w_i, z_i, x_i, y_i)\}_{i=1}^N$. Different from the *offline contextual bandit* setting (Strehl et al., 2010), where π^* is assumed identical to π, we assume a *covariate shift* (Sugiyama et al., 2007) on the latent context variable Z between the domains.

Assumption 1 (Covariate shift on bandit with latent context) *Given two domains π and π^*, the marginal distribution of the latent context variable is different in the two domains, i.e., $P_Z^{\pi} \neq P_Z^{\pi^*}$, while the conditional distributions that generate the proxy W and the reward Y remain the same, i.e., $P_{W|Z}^{\pi} = P_{W|Z}^{\pi^*}$ and $P_{Y|Z,X}^{\pi} = P_{Y|Z,X}^{\pi^*}$.*

Under this assumption, the objective of the agent is to minimize the cumulative regret in the target domain more efficiently by leveraging the prior dataset D.

2.2. Background

Our proposed method builds upon the framework of *structural causal models* and transportability theory, which we introduce in the following.

2.2.1. STRUCTURAL CAUSAL MODELS

A structural causal model (SCM) M is a tuple $M = \langle \boldsymbol{V}, \boldsymbol{U}, \mathcal{F}, P_{\boldsymbol{U}} \rangle$, where $\boldsymbol{V}$ is a set of endogenous variables that can be observed, $\boldsymbol{U}$ is a set of exogenous variables that are unobservable, $\mathcal{F} = \{f_1, \ldots, f_{|\boldsymbol{V}|}\}$ is a set of functions that represent the causal mechanisms generating X_i from its parents $\boldsymbol{Pa}_i \subseteq (\boldsymbol{V} \cup \boldsymbol{U}) \setminus \{X_i\}$, i.e., $X_i \leftarrow f_i(\boldsymbol{Pa}_i)$, and $P_{\boldsymbol{U}}$ is a joint distribution over the exogenous variables $\boldsymbol{U}$.

An SCM entails both an observational distribution and an *interventional distribution* (Peters et al., 2017). An intervention is defined as an operation that fixes values of a subset $\boldsymbol{X} \subset \boldsymbol{V}$ to a constant $\boldsymbol{x}$, denoted by $do(\boldsymbol{x})$, an abbreviation of $do(\boldsymbol{X} = \boldsymbol{x})$, which replaces functions $\{f_i : X_i \in \boldsymbol{X}\}$ that determine values of $\boldsymbol{X}$ in the original SCM M and generates a post-intervention SCM $M_{\boldsymbol{x}}$. The entailed distribution over $\boldsymbol{V}$ in M_x is the interventional distribution induced by $do(\boldsymbol{x})$, denoted by $P_{\boldsymbol{V}_{\boldsymbol{x}}} := P_{\boldsymbol{V}|do(\boldsymbol{x})}$.

Every SCM M has an associated *causal graph* $\mathcal{G} = (\boldsymbol{V}, \mathcal{E})$, which is a *directed acyclic graph* (DAG). Let $\boldsymbol{V_i} = \boldsymbol{Pa}_i \subseteq \boldsymbol{V}$ and $\boldsymbol{U_i} = \boldsymbol{Pa}_i \subseteq \boldsymbol{U}$. Given an SCM M, if $X_i \subseteq \boldsymbol{V}_j$, we draw a directed edge from X_i to X_j ($X_i \to X_j$) in a corresponding causal graph $\mathcal{G}$. If $X_i \subseteq \boldsymbol{U}_j$, we draw a dashed edge from X_i to X_j. Fig. 1(*a*) and Fig. 1(*b*) show corresponding causal graphs for the data-generating process of the latent contextual bandit in the source domain and the target domain,

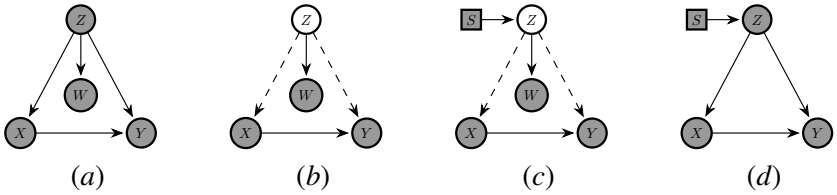


Figure 1: Causal Graphs of the data-generating processes. *Gray nodes are observable and white nodes are unobservable. Graphs in (a) and (b) depict causal graphs in the source domain and the target domain, respectively. The graph in (c) is the corresponding selection diagram for the two domains: the selection node S points to Z since $P_Z \neq P^*_Z$. The graph in (d) is a selection diagram encoding the domain discrepancy of an offline contextual bandit under a covariate shift.*

respectively. The causal graph of a post-intervention SCM $M_{\boldsymbol{x}}$ induced by $do(\boldsymbol{x})$ is similar to the original SCM M, except that the incoming edges of nodes associated with $\boldsymbol{X}$ are removed.

2.2.2. TRANSPORTABILITY

In the causal inference literature, the problem of identifying causal effects with potential domain discrepancy is studied under the theory of transportability. These works focus on identifying whether a causal effect can be estimated across domains and what can be transferred. Formally, the transportability is defined

Definition 2 (Transportability) *(Bareinboim and Pearl, 2012) Let π and π^* denote two domains, characterized by two observational distributions $P_{\boldsymbol{V}}$, $P^*_{\boldsymbol{V}}$, and two causal graphs $\mathcal{G}$ and $\mathcal{G}^*$, respectively. A causal effect $p(y|do(x))$ is transportable from π to π^* if $p(y|do(x))$ can be estimated from a set of interventions $\mathcal{I}$ on π, and $p^*(y|do(x))$ can be identified from $P_{\boldsymbol{V}}$, $P^*_{\boldsymbol{V}}$, $\mathcal{G}$, $\mathcal{G}^*$ and $\mathcal{I}$.*

Note, in Definition 2, $p(y|do(x))$ is identified from the source domain π, which is not necessarily the same as $p^*(y|do(x))$ in the target domain π^*. If it is, we say $p^*(y|do(x))$ is directly transportable. To check the transportability of a causal effect, the first step is to encode the domain discrepancy into a *selection diagram* $\mathcal{D}$ by adding a *selection node* pointing at the node that has a potential structural discrepancy. Fig. 1(c) shows a selection diagram encoding the domain discrepancy under Assumption 1. Given the selection diagram between two domains, one can identify whether a causal effect is transportable by applying the do-calculus (Pearl, 2009) on the causal effect and the corresponding selection diagram. If the causal effect can be transformed into an expression that only includes available interventional distributions and observational distributions in the source domain and the observational distribution in the target domain, we say the causal effect is transportable. The resulting expression of the causal effect is called a transport formula.

2.3. Transportability of Bandits with Latent Contexts

Building on the preliminary discussion of transportability, we can formulate the problem described in Section 2.1 within this framework. Recall that the agent's goal is to minimize the cumulative regret in the target domain while leveraging access to a dataset collected from the source domain. In this context, the causal effects of an arm selection conditioned on observing the proxy w are

$p(y|w, do(x))$ and $p^*(y|w, do(x))$ in the source and target domains. We refer to this causal effect as a w-specific causal effect throughout the paper. Following the definition by Bellot et al. (2024), the problem outlined in Section 2.1 can be formulated formally as follows: let M and M^* denote the SCMs that characterize the source domain π and the target domain π^*, and $P_{W,Z,X,Y}$ and $P^*_{W,Z,X,Y}$ the distributions entailed by M and M^*. Given the selection diagram $\mathcal{D}$ in Fig. 1(*c*) that characterizes the domain discrepancy between π and π^* and a prior dataset $D \sim P_{W,Z,X,Y}$, at each step $t = 1, \ldots, T$ in the target domain, the agent observes a proxy $w_t \in W$ generated from the underlying context $z_t \in Z$. It then takes an action $x_t \in X$ and observes a reward $y_t \in Y$. The objective of the agent is minimizing the expected cumulative regret in π^*, i.e., minimizing

$$\mathbb{E}_{\pi^*}[R] \coloneqq \sum_{t=1}^{T} \mathbb{E}_{\pi^*}[Y|w_t, do(\tilde{x}_t)] - \mathbb{E}_{\pi^*}[Y|w_t, do(x_t)], \tag{2}$$

where $\tilde{x}_t = \arg\max_{x \in \mathcal{X}} \mathbb{E}_{\pi^*}[Y|w_t, do(x)]$. When the proxy variable W is high-dimensional, calculating $\mathbb{E}_{\pi^*}[Y|w_t, do(\tilde{x}_t)]$ analytically is infeasible. We therefore replace it with $\mathbb{E}_{\pi^*}[Y|z_t, do(\tilde{x}_t)]$ in (2) to denote the optimal reward in this case. Nevertheless, this results in an ill-posed regret definition, since $\mathbb{E}_{\pi^*}[Y|z_t, do(\tilde{x}_t)]$ always yields a higher return than $\mathbb{E}_{\pi^*}[Y|w_t, do(x_t)]$. As the agent cannot close this gap between $\mathbb{E}_{\pi^*}[Y|z_t, do(\tilde{x}_t)]$ and $\mathbb{E}_{\pi^*}[Y|w_t, do(x_t)]$, the best it can do is achieving a linear regret. Therefore, with the high-dimensional proxy, we evaluate the value of $\mathbb{E}_{\pi^*}[R]$ after T steps as the total cumulative regret, rather than evaluating $\mathbb{E}_{\pi^*}[R]$ per step, which is the conventional evaluation metric for bandit algorithms.

From (2), we can see that with an accurate estimation of the casual effect $\mathbb{E}_{\pi^*}[Y|w_t, do(x)]$ or $p^*(y|w_t, do(x))$, the expected cumulative regret can be minimized. Hence, our main focus in this paper is estimating this quantity using the prior data and the observational distribution in the target domain.

3. Estimating Causal Effects with Transportability

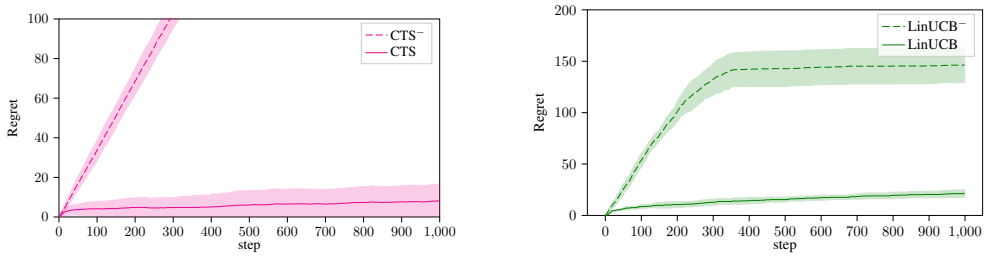
In this section, we derive two algorithms to estimate the causal effect $p^*(y|w, do(x))$ using prior data while avoiding negative transfer. One algorithm addresses a simple binary model, and the other generalizes the method to a high-dimensional proxy variable. We begin by highlighting the challenges of latent contexts and demonstrating negative transfer in common bandit algorithms.

3.1. Challenge of Negative Transfer

We first consider an offline contextual bandit setting, where the context is observable in both domains. Based on this assumption, one can draw a selection diagram of this setting under covariate shift, which is shown in Fig. 1(*d*). Since the agent can observe the context Z in both domains, it can evaluate $\tilde{x} = \arg\max_{x \in \mathcal{X}} \mathbb{E}_{\pi^*}[Y|z, do(x)]$ from the dataset D collected in the source domain. Therefore, the problem is equivalent to identifying the causal effect $\mathbb{E}_{\pi^*}[Y|z, do(x)]$ from the observational distribution $P_{X,Y,Z}$. Here, the causal effect of interest is directly transportable, and the optimal policy learned in the source domain can thus be directly transferred to the target domain.

Proposition 3 *For an offline contextual bandit under covariate shift, the z-specific causal effect $p(y|z, do(x))$, where $z \in \mathcal{Z}$, $x \in \mathcal{X}$ and $y \in \mathbb{R}$, is directly transportable. The transport formula of $p^*(y|z, do(x))$ is*

$$p^*(y|z, do(x)) = p(y|z, x). \tag{3}$$



(a) (b)

Figure 2: Negative transfer for classic bandit algorithms. *Fig. (a) shows negative transfer for binary model. Agents in this experiment utilize Thompson sampling (Thompson, 1933). Fig. (b) shows the negative transfer for the Linear Gaussian model with a scalar latent context Z and 5-dimensional proxy W. Agents in this experiment utilize LinUCB (Li et al., 2010).*

The proof can be found in Appendix B. This proposition suggests that the transportability of the z-specific causal effect is trivial and formally justifies the use of supervised learning in offline contextual bandits under covariate shift.

One may consider substiting Z with its proxy W to derive the same result as in Proposition 3, i.e., $p^*(y|w, do(x)) = p(y|w, x)$. Therefore, concluding that the problem outlined in Section. 2.3 is also straightforward to solve, as estimating $p(y|w, x)$ from the prior dataset D is trivial. However, this conclusion is not valid.

Proposition 4 *For the problem described in Section. 2.3, the w-specific causal effect $p^*(y|w, do(x))$ is not directly transportable and the transport formula of $p^*(y|w, do(x))$ is*

$$p^*(y|w, do(x)) = \sum_{z \in \mathcal{Z}} p(y|z, x)p^*(z|w). \tag{4}$$

The proof can be found in Appendix B. Proposition 4 formally establishes that the w-specific causal effect is not directly transportable from the source domain. If one naively transfers $p(y|w, do(x))$ from the source domain as if it were $p^*(y|w, do(x))$, it can lead to a negative transfer. This is due to the fact that $p(z|w) \neq p^*(z|w)$ under the assumption of covariate shift on Z, i.e., $p(z) \neq p^*(z)$. We simulate two experiments on the binary model and linear Gaussian model to show the negative transfer in Fig. 2. The results demonstrate a clear negative transfer for both models, where agents that naively transfer $p(y|w, do(x))$ (CTS$^-$ and LinUCB$^-$) from the source domain suffer a linear increase of regret while regrets for agents that directly learn from the target domain (CTS and LinUCB) increase sub-linearly. The experiment details can be found in Appendix C.

Moreover, although (4) includes the quantity $p(y|z, x)$ that can be estimated from the prior dataset D, it is still not possible to fully compute (4), as the posterior distribution $p^*(z|w)$ is not estimable without observing the context Z in the target domain. Consequently, this raises concerns about the formulation of the problem in Section 2.3 within the framework of transportability. However, in the next section, we will show that it is possible to recover the unknown posterior distribution from the observational distribution in the target domain. This allows us to leverage (4) to estimate the w-specific causal effect in the target domain using prior data.

3.2. Transfer Learning Guided by Transport Formula

This section describes a transfer learning algorithm guided by the transport formula we derived in Proposition 4. We start by restoring the unknown posterior distribution $p^*(z|w)$.

3.2.1. RESTORING UNKNOWN POSTERIOR FOR BINARY MODEL

We first consider a binary model [2]. In this case, one can derive an analytic solution for computing the unknown posterior with Bayes' theorem

$$p^*(z|w) = \frac{p^*(w|z)p^*(z)}{p^*(w)}. \tag{5}$$

If we can estimate the three terms on the right-hand side, we can solve for the unknown $p^*(z|w)$. Under the assumption of covariate shift, we have $p^*(w|z) = p(w|z)$, where $p(w|z)$ can be estimated from the prior dataset D. The denominator, $p^*(w)$, can be directly computed from the observation of W, given that W is binary. Therefore, the only term that can not be directly computed is $p^*(z)$. For the binary model, we have the following proposition

Proposition 5 *Let Z and W in Fig. 1(c) be binary. Then, the distribution of Z in the target domain can be estimated by*

$$p^*(z=1) = \frac{p^*(w=1) - p(w=1|z=0)}{p(w=1|z=1) - p(w=1|z=0)}. \tag{6}$$

The derivation of (6) is based on the marginalization of $p^*(w=1)$; we provide the details in Appendix D. Note, the terms on the right-hand side of (6) can be estimated either from the observation in the target domain or the prior dataset collected from the source domain.

Eq. (6) provides a complete solution to (5), thus restoring the unknown posterior $p^*(z|w)$ from the prior dataset and the observation in the target domain. Once the posterior is restored, we can calculate the w-specific causal effect by the transport formula derived in (4). The concrete algorithm is given in Alg. 1. However, when we drop the assumption of the binary model and consider a high-dimensional proxy W governed by a flexible generating mechanism $P_{W|Z}$, we do not have an analytic solution of (5). Nevertheless, inspired by the Causal Effect Variational Autoencoder (CEVAE) Louizos et al. (2017), we will show how to approximate (5) for a high-dimensional proxy using VAEs in the next section.

3.2.2. RESTORING UNKNOWN POSTERIOR FOR HIGH-DIMENSIONAL PROXY

Recall that we start the derivation of (6) from the marginal distribution of the proxy W. We then derive the solution for the unknown posterior. However, it is generally intractable to compute the marginal for a high-dimensional W (Blei et al., 2017). Therefore, we propose leveraging a VAE to restore the marginal distribution based on the *latent-variable model*. It maximizes the marginal distribution of the observation by maximizing its evidence lower bound (ELBO)

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{z\sim q_\phi(z|w)}[\log p_\theta(w|z) - \text{KL}(q_\phi(z|w)||p(z))], \tag{7}$$

2. We notice the connection between the binary model and a transfer learning problem of *bandits with unobserved confounders*. Interested readers are referred to Remark 9 in Appendix C

Algorithm 1: Bandit with transportability for binary model.

Input: Prior dataset D
Estimate $p(w|z) \leftarrow D$
Estimate $p(y|x,z) \leftarrow D$
for $t \in \{1,\dots,T\}$ **do**
  observe w_t;
  Update $p^*(w)$;
  Update $p^*(z)$ by (6);
  for $x \in \{0,1\}$ **do**
    Compute expected reward $\mathbb{E}_{\pi^*}[Y|w_t, do(x)]$ by (4);
  $x_t \leftarrow \arg\max_x \mathbb{E}_{\pi^*}[Y|w_t, do(x)]$;
  take x_t and get y_t;

Algorithm 2: Bandit with transportability for high-dimensional proxy.

Input: Prior dataset D
Estimate $p_{\theta_1}(w|z) \leftarrow D$
Estimate $p_{\theta_2}(y|x,z) \leftarrow D$
Initialize replay buffer $\mathcal{B}$ with capacity N
Initialize ϕ for the encoder $q_\phi(z|w)$
for $t \in \{1,\dots,T\}$ **do**
  observe w_t;
  $\hat{z}_t \leftarrow q_\phi(z|w_t)$;
  $x_t \leftarrow \arg\max_x p_{\theta_2}(y|x,\hat{z}_t)$;
  Take x_t, get y_t, store (w_t, x_t, y_t) in $\mathcal{B}$;
  if *t is a gradient step* **then**
    $\mathbf{W}^M \leftarrow$ Sample a random mini-batch of M data points from $\mathcal{B}$;
    Perform gradient step on (9) with respect to θ_1, θ_2 and ϕ using $\mathbf{W}^M$;

where $\mathrm{KL}(\cdot|\cdot)$ is the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951). Note, $p(z)$ here denotes the prior we set for z. Two neural networks are used to parameterize the approximated posterior $q_\phi(z|w)$ and the conditional $p_\theta(w|z)$, denoted by the encoder and the decoder, respectively.

The encoder can be used to approximate the posterior in (5), and we could directly transfer a decoder from the source domain as the mechanism that generates W from Z is invariant across the domains given Assumption 1. Namely, we can have the optimal decoder $p_\theta(w|z) = p^*(w|z)$ in the target domain and this can drive the encoder $q_\phi(z|w)$ to approximate the true $p^*(z|w)$.

One issue with this approach is that we do not know the prior distribution $p(z)$ in (7). Typically, the prior is chosen as $\mathcal{N}(0, \mathrm{I})$, which drives the encoder to map the high-dimensional input w to a range close to $\mathcal{N}(0, \mathrm{I})$. However, this may be problematic in our setting, as the true posterior does not necessarily match with $\mathcal{N}(0, \mathrm{I})$.

Nevertheless, $\mathcal{N}(0, \mathrm{I})$ could be an effective prior for the encoder to start with if we apply less regularization in the latent space, i.e., a smaller coefficient on the KL divergence term. This allows the encoder to learn a more flexible posterior. When combined with an optimal decoder, this approach encourages the encoder to approximate the true posterior $p^*(z|w)$ more effectively, rather than being heavily constrained by an overly simplistic assumption about the latent space.

This approach leads us to leverage the objective function of β-VAE (Higgins et al., 2017), which adds a constraint coefficient β on the KL term in (7),

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{z\sim q_\phi(z|w)}[\log p_\theta(w|z) - \beta \mathrm{KL}(q_\phi(z|w)||p(z))]. \tag{8}$$

Notice that β is set to be greater than 1 in the original β-VAE since the objective there is to learn a disentangled representation. In our approach, β is set to be smaller than 1 as we want the encoder to learn a flexible posterior.

So far, we have made the design choice of restoring the unknown posterior. We still need to estimate the w-specific causal effect in (4), which we show in the next section.

Remark 6 *Restoring the unknown posterior $p^*(z|w)$ entirely is impossible without any inductive bias due to the unidentifiability of VAEs (Locatello et al., 2019; Rissanen and Marttinen, 2021). However, under certain assumptions, the identifiability of the latent distribution $P(Z)$ can be achieved (Kivva et al., 2022). Let f be the function that generates W from Z, the weakest assumptions that guarantee the identifiability of $P(Z)$ up to an affine transformation are (1) $P(Z)$ is a Gaussian mixture model; (2) f is a piecewise affine function; (3) f is weakly injective.*

3.2.3. CAUSAL EFFECT VAE WITH TRANSPORTABILITY

The transport formula in (4) suggests that both $p(y|z,x)$ and $p^*(z|w)$ are required to estimate the w-specific causal effect $p^*(y|w,do(x))$. The encoder $q_\phi(z|w)$ in (8) can be used to approximate $p^*(z|w)$, and we can use another neural network to estimate $p(y|z,x)$ from the prior data. Yet, in practice, we may still need to adapt the estimated $p(y|z,x)$ in the target domain since the distribution of z is potentially different from the source domain. One way to adapt this neural network in the target domain is to jointly train it with the β-VAE by adding it as another decoder. Therefore, we obtain a new objective function,

$$\mathcal{L}_{\text{Trans}} = \mathbb{E}_{z\sim q_\phi(z|w)}[\log p_{\theta_1}(w|z) + \log p_{\theta_2}(y|x,z) - \beta \text{KL}(q_\phi(z|w)||p(z))]. \tag{9}$$

Note that the two decoders $p_{\theta_1}(w|z)$ and $p_{\theta_2}(y|x,z)$ can be estimated from the prior dataset D, and both of them encourage the encoder to approximate the true posterior.

A remaining issue is how we can train this VAE in the target domain, given that we have only one observation at each step. The solution is using a *replay buffer* (Mnih, 2013) to store the observation pair (w,x,y) and train the VAE by sampling a batch from it for a fixed frequency. The concrete algorithm is given in Alg. 2, and the architecture is shown in Fig. 3. We provide an implementation at https://github.com/Marksmug/CEVAE-Transportability.

Since Alg. 2 uses stochastic gradient descent for optimizing (9), we use the *parameter counting argument* (Bengio et al., 2019) to explain the benefit of our method over naively transferring $p(y|w,do(x))$ from the source domain. In practice, the latter can be considered as transferring a full autoencoder in Fig. 3 from the source domain.

Corollary 7 *Given the setting in Section 2.3 and Assumption 1. Let $\boldsymbol{V} = \{X,Y,Z,W\}$ be the set of all variables, and $P(\boldsymbol{V})$ be the joint distribution learned from the source domain by the two decoders in Fig. 3 and $P'(\boldsymbol{V})$ be the one learned by a full autoencoder in Fig. 3. The expected gradient w.r.t. the parameter θ_1 of the log-likelihood under the target domain will be zero for the former one and non-zero for the latter, i.e.,*

$$\mathbb{E}_{\pi^*}\left[\frac{\partial \log P'(\boldsymbol{V})}{\partial \theta_1}\right] \neq \mathbb{E}_{\pi^*}\left[\frac{\partial \log P(\boldsymbol{V})}{\partial \theta_1}\right] = 0. \tag{10}$$

Corolarry 7 (see proof in Appendix E) formally justifies the adaptation benefit of our approach over naively transferring $p(y|w,do(x))$ using CEVAE-like latent variable models. It suggests that our approach guided by the transport formula (4) needs to adapt fewer parameters in the target domain than the naively transferring approach. This consequently affects the number of samples required for the adaptation in the target domain.

Remark 8 *The architecture and the objective function of Alg. 2 is similar to CEVAE, which estimates $p(w,z,x,y)$ from the observation of (w,x,y). The objective of the two methods is the same.*

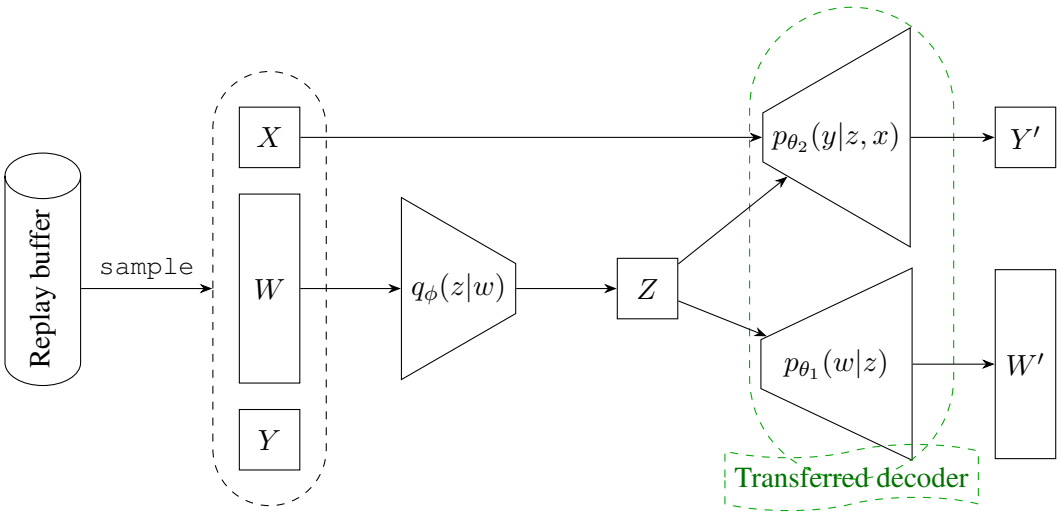


Figure 3: Architecture of VAE with transportability.

Both of them estimate the causal effect of the action x given the observation of the proxy w. The differences between them are (1) CEVAE is trained in an offline setting, while Alg. 2 is trained online; (2) instead of restoring $p(w, z, x, y)$, Alg. 2 only restores $p(z|w)$, reducing the number of parameters required to adapt (as shown in Corollary 7) compared to CEVAE; (3) Alg. 2 does not restore $p(x|z)$ using another network, while CEVAE does.

4. Evaluation

In this section, we evaluate the proposed method on both synthetic and semi-synthetic datasets. Specifically, we first evaluate the method described in Alg. 1 on a simulated binary model. We then extend to high-dimensional proxy variables and demonstrate the effectiveness of our method in real-world datasets. In particular, we modify the Infant Health and Development Program (IHDP) dataset (Hill, 2011) and test our method on it. Moreover, we want to investigate the performance of our method on an even higher dimension of the proxy. Inspired by Rissanen and Marttinen (2021), we employ the image dataset MNIST (LeCun, 1998) to create a setting where the proxy is an image.

For the binary model experiment, we run multiple simulations and evaluate the performance using the average cumulative regret in (2) within the target domain. For the high-dimensional proxy case, we use averaged total cumulative regret against gradient steps as the evaluation metric.We refer to our algorithms as causal agents. There are three main questions we want to answer: (1) Does our method avoid the negative transfer in the target domain? (2) Does our method improve the sample efficiency in the target domain? (3) Can our method restore the unknown posterior $p^*(z|w)$ in the target domain? To answer the first one, we compare our method with baselines that transfer $\mathbb{E}[Y|x, w]$ from the prior dataset. For the second, we compare our method with a baseline without access to the prior dataset. For the third one, we compare the posterior $p^*(z|w)$ learned by our method with the true distribution of the latent context P_Z^*.

4.1. Synthetic Binary Model

We begin by evaluating Alg. 1 on a binary model. We consider a bandit setting characterized by the causal graph shown in Fig. 1(*b*), involving proxy, context, action, and reward variables

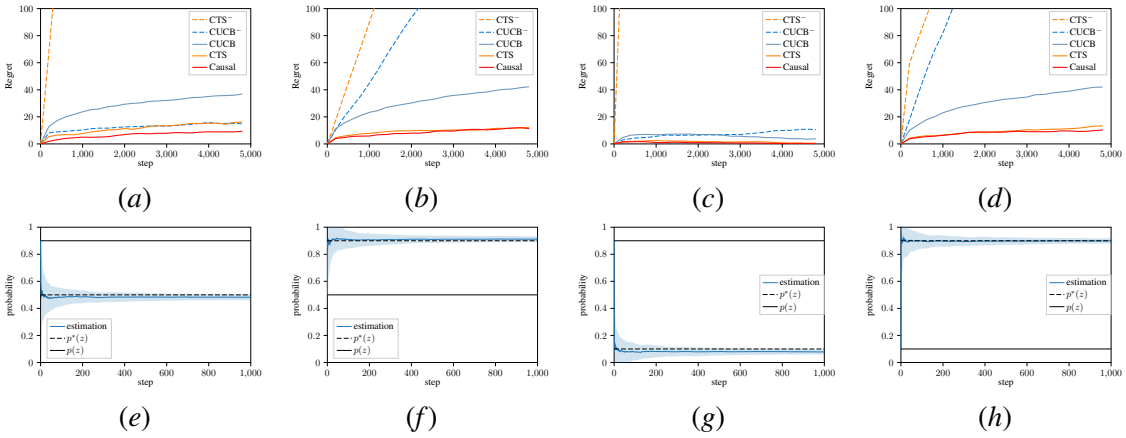


Figure 4: Cumulative regret and the distribution comparison for the binary model. *Each column shows the result of one setting, where context Z follows different Bernoulli distributions. The top row shows the cumulative regret averaged over 100 simulations for Alg. 1 and the baselines. The bottom row shows the convergence of the estimation (blue) to the true distribution of Z in the target domain (dashed black) and the distribution of Z in the source domain (solid black).*

$W, Z, X, Y \in \{0, 1\}$, respectively. We set the optimal action as opposite to the context. Details of the data-generating process are provided in Appendix F.

We provide 1000 prior samples collected from the source domain and average the cumulative regret in the target domain over 100 simulations. We compare our method with standard baselines, including Upper Confidence Bound (UCB) (Auer, 2002) and Thompson Sampling (TS) (Thompson, 1933). We condition both algorithms on w, referring to them as Context-UCB (CUCB) and Context-TS (CTS). We refer to their counterparts that directly transfer $\mathbb{E}[Y|w, do(x)]$ from the prior dataset, as CUCB$^-$ and CTS$^-$, respectively.

The first row of Fig. 4 shows the cumulative regret comparisons for all agents under four different settings, where the optimal policy is different across domains. We observe that both the CUCB$^-$ and CTS$^-$ agents exhibit significantly worse performance than their counterparts without access to prior data. This observation suggests that both agents suffer from negative transfer in these settings. In contrast, the causal agent, which transfers knowledge explicitly via the transport formula (4), successfully avoids negative transfer. However, due to the relatively simple binary model, the causal agent does not demonstrate significantly better performance than the CTS agent, which learns directly from the target domain. The second row shows the estimation of $p^*(z)$ (blue) based on (6) and the distribution of Z in the source domain (solid black) as well as the target domain (dashed black). It can be seen the estimation converges to $p^*(z)$ in all settings.

4.2. Semi-synthetic Data

In this section, we evaluate Alg. 2 on two real-world datasets: IHDP and MNIST. IHDP is typically used to evaluate individual-level causal inference, and MNIST is a high-dimensional image dataset. Following Rissanen and Marttinen (2021), we modify both datasets to make the data-generating process compatible with our assumption; details can be found in Appendix F.

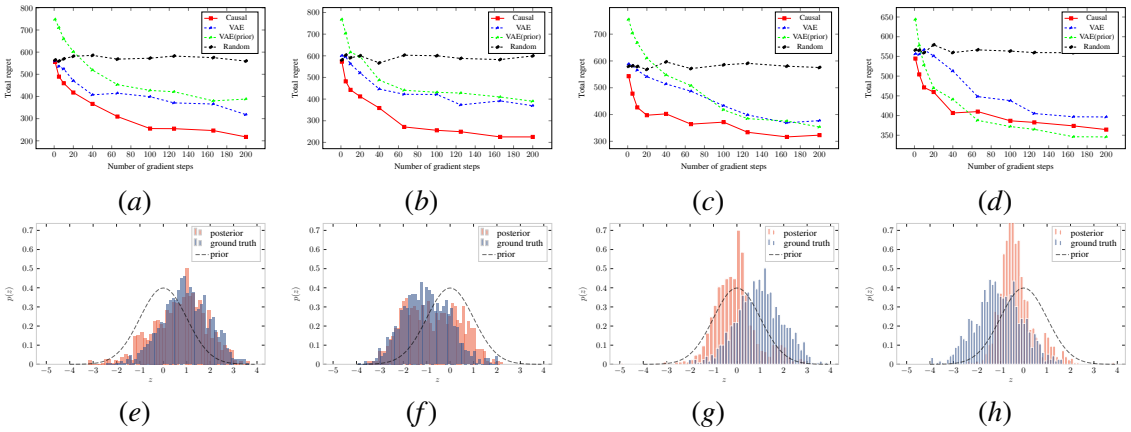


Figure 5: Total cumulative regrets and distributions comparison for IHDP and MNIST dataset. *The first two columns are results for the IHDP dataset, and the last two are for the MNIST dataset. More results can be found in Appendix H.*

For both experiments, we use the total cumulative regret after 1000 steps as our evaluation metric, averaged over 5 random seeds. We define the following three agents: (1) **VAE (prior)**: we pre-train a VAE on the prior data by reconstructing W as well as predicting Y, and start the for loop in Alg. 2. This agent directly transfers the interventional distribution $p(y|w, do(x))$ from the source domain according to the transport formula (4) [3]; (2) **VAE**: we initialize a random VAE and start the for loop in Alg. 2 without any prior knowledge; (3) **Causal**: we pre-train two decoders on the prior data as shown in Alg. 2, which are two neural nets with one mapping context z to proxy w and the other one mapping context z and action x to reward y. For fail comparison, we use the same architecture and objective function for all agents. The details of VAEs used in the experiments can be found in Appendix G.

4.2.1. PROXY IHDP DATASET

The IHDP dataset was created to study the individual causal effects of intensive child care on future cognitive test scores. It collects 25 different measurements of the children and their mothers, some continuous and some categorical, and the corresponding binary treatment they take in a randomized experiment as well as the continuous synthetically generated score. Given the limited size of the original dataset (747 samples), we train a VAE on the measurements with the latent dimension to be 1 and use it to generate more samples of the measurements. To make the data-generating process compatible with our assumption, we use the latent variable of the VAE as the latent context variable Z and the generated measurement as the proxy variable W. Additionally, we use a simple function to regenerate the score Y for each sample we generated from the VAE, such that the optimal action is 0 if $z < 0$ and 1 otherwise.

We want to compare the sample efficiency of the three agents, which can be shown by the total cumulative regret over the number of gradient steps. In this case, fewer gradient steps and smaller total cumulative regret means better sample efficiency. We provide 1000 samples from the source domain, and show our results in the first two columns of Fig. 5 with $P_Z = \mathcal{N}(1, 1)$, $P_Z^* = \mathcal{N}(-1, 1)$

3. Since the encoder that learns the posterior $p(z|w)$ in the source domain is transferred.

and $P_Z = \mathcal{N}(-1, 1)$, $P_Z^* = \mathcal{N}(1, 1)$, respectively. Here, the causal agent shows a constantly higher sample efficiency than other agents. It overcomes the negative transfer that shows in the VAE (prior) agent, and it shows a better sample efficiency than the VAE agent.

Additionally, we plot the posterior distribution (blue) learned by the encoder and the true P_Z^* (orange) in the second row of Fig. 5. Recall (9), where the prior we set for the encoder to learn is $\mathcal{N}(0, 1)$. However, the encoder instead learns a shift to the true P_Z^*. This shows that our method can approach the true P_Z^* even without observing it.

4.2.2. PROXY MNIST DATASET

We use the MNIST dataset to evaluate the performance of our method on image proxies. To make the data-generating process match with the causal graphs, we train a generative adversarial network (GAN) (Goodfellow et al., 2014) on the MNIST dataset to generate images from a three-dimensional Gaussian noise. Its first dimension is then used as the context Z, and we generate binary X and real-valued Y from the context Z for each sample. The function that generates y from z and x is similar to the one used in the IHDP dataset. The optimal action for $z < 0$ is 0 and 1 otherwise.

Similar to the settings in the IHDP dataset, we divide the dataset according to the distribution of Z. In the first setting, P_Z is set to be $\mathcal{N}(1, 1)$ and P_Z^* is $\mathcal{N}(-1, 1)$ and vice versa in the second setting. We train all VAEs with a three-dimensional latent and choose the first dimension as the restored z. The prior dataset consists of 1000 samples from the source domain. We show the results of the two settings in the last two columns in Fig. 5. As expected, the VAE (prior) agent shows a negative transfer in both cases without updating in the target domain. As we increase the gradient steps, VAE (prior) adapts quickly in the second setting while relatively slow in the first setting. It takes 40 gradient steps to achieve a performance slightly better than the random agent. The causal agent, on the other hand, adapts quickly in both settings and constantly shows a lower regret than VAE agent, which has no prior knowledge of the source domain.

However, the posterior distributions learned by the encoder do not consistently converge to the true P_Z^*. In the first setting, the posterior aligns more with P_Z instead of P_Z^* (see Fig. 5(*g*)). In the second setting, the posterior shifts slightly towards P_Z^* (see Fig. 5(*h*)). We hypothesize this is because of the unidentifiability of VAEs discussed in Remark 6. Although the transferred decoder is assumed to serve as an inductive bias in the target domain, it is insufficient to restore the latent variable distribution with the dimension of three in this experiment.

5. Conclusion

In this paper, we addressed the challenge of negative transfer in a latent contextual bandit problem with a high-dimensional proxy, leveraging causal transportability theory. We begin by introducing a novel algorithm for the binary case, showing how to effectively transfer knowledge from a source to a target domain via the transport formula. We then extend this to handle high-dimensional proxies using VAEs. Our experiments on both synthetic and semi-synthetic datasets demonstrate that our method successfully avoids negative transfer and improves sample efficiency under the covariate shift on the latent context. These findings highlight the effectiveness of using causal transportability to tackle distribution shifts between environments and enhance the adaptability of bandit algorithms. Extending this framework to reinforcement learning, where a Markov decision process is considered to govern the data-generating process, could be an intriguing avenue for future work.

Acknowledgments

This work was supported by the Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence FCAI.

References

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Elias Bareinboim and Judea Pearl. Transportability of causal effects: Completeness results. In *AAAI Conference on Artificial Intelligence*, pages 698–704, 2012.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

Elias Bareinboim, Andrew Forney, and Judea Pearl. Bandits with unobserved confounders: A causal approach. *Advances in Neural Information Processing Systems*, 2015.

Alexis Bellot, Alan Malek, and Silvia Chiappa. Transportability for bandits with data from different environments. *Advances in Neural Information Processing Systems*, 2024.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Blair Bilodeau, Linbo Wang, and Dan Roy. Adaptively exploiting d-separators with causal bandits. *Advances in Neural Information Processing Systems*, pages 20381–20392, 2022.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Changxiao Cai, T Tony Cai, and Hongzhe Li. Transfer learning for contextual multi-armed bandits. *The Annals of Statistics*, 52(1):207–232, 2024.

Aniket Anand Deshmukh, Urun Dogan, and Clay Scott. Multi-task learning for contextual bandits. *Advances in Neural Information Processing Systems*, 2017.

Andrew Forney, Judea Pearl, and Elias Bareinboim. Counterfactual data-fusion for online reinforcement learners. In *International Conference on Machine Learning*, pages 1156–1164, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2014.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Joey Hong, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, and Craig Boutilier. Latent bandits revisited. *Advances in Neural Information Processing Systems*, pages 13423–13433, 2020.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596, 2017.

Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Identifiability of deep generative models without auxiliary information. *Advances in Neural Information Processing Systems*, 35:15687–15701, 2022.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Kevin Labille, Wen Huang, and Xintao Wu. Transferable contextual bandits with prior observations. In *Pacific-Asia Conference On Knowledge Discovery And Data Mining*, pages 398–410, 2021.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in Neural Information Processing Systems*, 2007.

Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal bandits: Learning good interventions via causal inference. *Advances in Neural Information Processing Systems*, 2016.

Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Sanghack Lee and Elias Bareinboim. Structural causal bandits: Where to intervene? *Advances in Neural Information Processing Systems*, 2018.

Sanghack Lee and Elias Bareinboim. Structural causal bandits with non-manipulable variables. In *AAAI Conference on Artificial Intelligence*, pages 4164–4172, 2019.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web*, pages 661–670, 2010.

Shuyu Lin, Ronald Clark, Robert Birke, Sandro Schönborn, Niki Trigoni, and Stephen Roberts. Anomaly detection for time series using VAE-LSTM hybrid model. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4322–4326, 2020.

Bo Liu, Ying Wei, Yu Zhang, Zhixian Yan, and Qiang Yang. Transferable contextual bandit for cross-domain recommendation. In *AAAI Conference on Artificial Intelligence*, 2018.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124, 2019.

Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in Neural Information Processing Systems*, 30, 2017.

Yangyi Lu, Amirhossein Meisami, and Ambuj Tewari. Causal bandits with unknown graph structure. *Advances in Neural Information Processing Systems*, pages 24817–24828, 2021.

Volodymyr Mnih. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

J Pearl. *Causality*. Cambridge University Press, 2009.

Judea Pearl and Elias Bareinboim. Transportability of causal and statistical relations: A formal approach. In *AAAI Conference on Artificial Intelligence*, pages 247–254, 2011.

Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021.

Jonas Peters, Dominik Janzing, and Bernhard Schlkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017.

Severi Rissanen and Pekka Marttinen. A critical look at the consistency of causal estimation with deep latent variable models. *Advances in Neural Information Processing Systems*, pages 4207–4217, 2021.

Rajat Sen, Karthikeyan Shanmugam, Murat Kocaoglu, Alex Dimakis, and Sanjay Shakkottai. Contextual bandits with latent confounders: An nmf approach. In *International Conference on Artificial Intelligence and Statistics*, pages 518–527, 2017.

Alex Strehl, John Langford, Lihong Li, and Sham M Kakade. Learning from logged implicit exploration data. *Advances in Neural Information Processing Systems*, 2010.

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

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11293–11302, 2019.

Zirui Yan, Dennis Wei, Dmitriy A Katz, Prasanna Sattigeri, and Ali Tajer. Causal bandits with general causal models and interventions. In *International Conference on Artificial Intelligence and Statistics*, pages 4609–4617, 2024.

Junzhe Zhang and Elias Bareinboim. Transfer learning in multi-armed bandit: a causal approach. In *Conference on Autonomous Agents and Multiagent Systems*, pages 1778–1780, 2017.

Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:146–177, 2023.

Li Zhou and Emma Brunskill. Latent contextual bandits and their application to personalized recommendations for new users. In *International Joint Conference on Artificial Intelligence*, pages 3646–3653, 2016.

Appendix A. Related Work

In this paper, we propose to use the transport formula to address the negative transfer in latent contextual bandits under the assumption of distribution shift on the latent context. Also, we assume a high-dimensional proxy variable and employ VAEs to restore the latent context from it. In this section, we relate our contributions to the existing literature.

Transfer Learning in Bandits. Our work is closely aligned with the framework of transfer learning in bandits, which aims to accelerate the convergence to the optimal actions in the target domain by transferring knowledge from an offline dataset collected from related environments. Existing works vary in their assumptions regarding the "differences" between environments. For example, Deshmukh et al. (2017) assume the reward distribution for each arm given the context is different and improves the estimation of the reward distributions for each arm by leveraging the similarities in the context-to-reward mapping. Liu et al. (2018) address potentially different context and action spaces and introduce a method to learn a translation matrix for aligning feature spaces between source and target domains. While they assume a single unknown reward parameter shared between domains, Labille et al. (2021) relax this assumption by considering different unknown reward parameters across domains. Combining with LinUCB, they propose leveraging historical data from similar domains to initialize new arm parameters, using a similarity measure and a decaying weight to prioritize relevant observations. Zhang and Bareinboim (2017) address the presence of unobserved confounders that simultaneously influence both the rewards and actions. Under the assumption of shifts in reward distributions, they propose to derive causal bounds from prior datasets, utilizing these bounds to inform and optimize exploration and exploitation within the target domain. Although these works explore various assumptions regarding domain differences, none of them considers changes in the distribution of the context. The most closely related work to ours is by Cai et al. (2024). Cai et al. (2024) explore transfer learning for contextual bandits under covariate shift, i.e., the distribution of context is different across domains. However, their approach involves partitioning the covariate space into bins, which may not be feasible for high-dimensional covariates such as the proxy in our problem setting. Moreover, their setting does not incorporate an underlying latent context, as is the case in our latent contextual bandit framework.

Latent Contextual Bandits. Latent contextual bandits were initially introduced by Zhou and Brunskill (2016) and later extended by Hong et al. (2020), who employ offline-learned models that provide an initial understanding of the latent states. However, their problem setting differs from ours in two key aspects: (1) their reward function depends on the context, action, and latent state, whereas our reward depends only on the action and latent state; and (2) they do not account for distributional shifts in the context or latent state. More closely related to our setting, Sen et al. (2017) consider a latent confounder associated with the observed context, and the reward depends solely on the latent confounder and action. Unlike our approach, they address the problem in an online setting by factorizing the observed context-reward matrix into two low-dimensional matrices, with one capturing the relationship between contexts and latent confounders and the other between rewards and latent confounders. Although this work defines their problem setting from a causal perspective, they do not apply any theory from causal inference to derive their method.

Bandits and Causal Inference. The connection between bandit problems and causal inference is well-established. Bareinboim et al. (2015) introduce the problem of bandits with unobserved confounders and leverage counterfactual estimation from causal inference to reduce the cumulative regret. Building on this, Forney et al. (2017) further exploits counterfactual reasoning to generate

counterfactual data, integrating it with the observational data and interventional data to learn a good policy. Moreover, there is extensive literature using causal graphs to define the structure of bandits under the name of causal bandits (Lattimore et al., 2016; Lee and Bareinboim, 2018, 2019; Lu et al., 2021; Bilodeau et al., 2022; Yan et al., 2024). While those works exploit knowledge in a single environment, Bellot et al. (2024) propose a general transfer learning framework for causal bandits, using posterior approximations on the causal relations encoded in the selection diagram, grounded in transportability theory. We realize their approach of learning the invariant causal relations aligns with our method, and we position our work as an extension of their framework within the specific context of latent contextual bandits. Further, we relax their assumption of discrete SCMs by considering a high-dimensional proxy variable and a continuous latent context.

Appendix B. Proofs for Section 3.1

Proposition *3* *For an offline contextual bandit under covariate shift, the causal effect $p(y|z, do(x))$, where $z \in \mathcal{Z}$, $x \in \mathcal{X}$ and $Y \in \mathbb{R}$ is directly transportable. Additionally, the transport formula of $p^*(y|z, do(x))$ can be written as*

$$p^*(y|z, do(x)) = p(y|z, x).$$

Proof Given the selection diagram $\mathcal{D}$ in Fig. 1(*d*), the proof is two-fold. First, we prove the causal effect $p^*(y|z, do(x))$ is directly transportable from the source domain π to the target domain π^*. Second, we prove the transport formula can be written as $p^*(y|z, do(x)) = p(y|z, x)$.

We can rewrite $p^*(y|z, do(x)) = p(y|z, s, do(x))$ by conditioning on the selection node S to denote the distribution in the target domain (Pearl and Bareinboim, 2011). Since we can observe Z, we can use *d-separation* (Pearl, 2009) to prove $(Y \perp\!\!\!\perp S|Z)_{\mathcal{D}_{\overline{X}}}$. This allows us to use the first rule of the do-calculus Pearl (2009). Thus we have the transport formula $p^*(y|z, do(x)) = p(y|z, do(x))$. Therefore, we prove the causal effect $p^*(y|z, do(x))$ is directly transportable from π to π^*.

Following the result above, we can further apply the second rule of the do-calculus and obtain the transport formula $p^*(y|z, s, do(x)) = p(y|z, x)$ based on the fact $(Y \perp\!\!\!\perp X|Z)_{\mathcal{D}_{\underline{X}}}$. Hence, (3) holds.

■

Proposition *4* *For the problem described in Section2.3, the w-specific causal effect $p^*(y|w, do(x))$ is not directly transportable and the transport formula of $p^*(y|w, do(x))$ can be written as*

$$p^*(y|w, do(x)) = \sum_{z \in \mathcal{Z}} p(y|w, z)p^*(z|w).$$

Proof Given the selection diagram $\mathcal{D}$ in Fig. 1(*c*) that encodes the domain discrepancy in the problem outlined in Section 2.3, we can rewrite $p^*(y|w, do(x)) = p(y|w, s, do(x))$. By d-separation, we can write $p(y|w, s, do(x)) = p(y|w, do(x))$ if we have the condition $(Y \perp\!\!\!\perp S|W)_{\mathcal{D}_{\overline{X}}}$. However, this condition is not satisfied as conditioning on the proxy W still leaves the backdoor path $S \to Z \to Y$ open. Hence, we prove $p^*(y|w, do(x)) \neq p(y|w, do(x))$ and the w-specific causal effect is not directly transportable.

Next, we prove $p^*(y|w, do(x)) = \sum_{z\in\mathcal{Z}} p(y|w, z)p^*(z|w)$. We can rewrite $p^*(y|w, do(x)) = p(y|w, s, do(x))$. By basic probability operations, we have

$$p(y|w, s, do(x)) = \sum_{z\in\mathcal{Z}} p(y|w, s, z, do(x))p(z|w, s, do(x)).$$

Since $(S \perp\!\!\!\perp Y|Z)_{\mathcal{D}_{\overline{X}}}$ and $(X \perp\!\!\!\perp Z)_{\mathcal{D}_{\overline{X}}}$, by the first and third rule of the do-calculus, we have $p(y|w, s, do(x)) = \sum_{z\in\mathcal{Z}} p(y|w, z, do(x))p(z|w, s)$. Now, we can also observe that $(S \perp\!\!\!\perp Y|Z)_{\mathcal{D}_{\overline{X}}}$. By the first rule of the do-calculus, we have $P(Y|w, s, do(x)) = \sum_{z\in\mathcal{Z}} p(y|z, do(x))p(z|w, s)$. Further, we have $(X \perp\!\!\!\perp Y|Z)_{\mathcal{D}_{\underline{X}}}$, thus allowing us to apply the second rule of the do-calculus on $p(y|z, do(x))$ and obtain $p(y|z, do(x)) = p(y|z, x)$. Finally, based on the definition of the selection node, we then have the transport formula $p^*(y|w, do(x)) = \sum_{z\in\mathcal{Z}} p(y|w, z)p^*(z|w)$. ■

Appendix C. Experiment for Negative transfer

We test on both a binary model and a linear Gaussian model to illustrate the negative transfer. Consider a binary model where W, Z, X, and Y are binary variables. Let $P_Z = \text{Bern}(0.9)$ and $P^*_Z = \text{Bern}(0.5)$. We set the optimal action X opposite to Z. A Thompson sampling (Thompson, 1933) agent is trained to learn the w-specific causal effect $p(y|w, do(x))$ by identifying the optimal action conditioned on each $w \in \{0, 1\}$; we refer to this agent as the Context-Thompson Sampling (CTS) agent. The learned $p(y|w, do(x))$ is then directly transferred to the target domain as if it were $p^*(y|w, do(x))$. We refer to this agent as CTS$^-$, and to the one that does not transfer knowledge from the source domain as CTS. The cumulative regrets are presented in Fig. 2(*a*). It can be observed that the CTS$^-$ agent exhibits linearly increasing cumulative regret, while the cumulative regret of the CTS agent trained directly in the target domain increases sub-linearly.

Further, considering the high-dimensional nature of the proxy W, we conducted a simulation where $Z \in \mathbb{R}$, $W \in \mathbb{R}^5$ and $X, Y \in \{0, 1\}$, using LinUCB (Li et al., 2010) as the baseline agent. The distribution of Z is $\mathcal{N}(8, 1)$ and $\mathcal{N}(2, 1)$ in the source domain and the target domain, respectively. The generating process from Z to W follows a linear-Gaussian transformation: $w = a \odot z + b + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$ and $\odot$ is the element-wise multiplication. We set the optimal action to be 1 if $z < 5$, otherwise 0, i.e., $y = \mathbb{1}\{(z \geq 5 \wedge x = 0) \vee (z < 5 \wedge x = 1)\}$. The result, shown in Fig. 2(*b*), demonstrates a similar trend of negative transfer.

Remark 9 *Interestingly, we notice the connection between the binary model and a transfer learning problem of bandits with unobserved confounders proposed by Bareinboim et al. (2015). In their setting, a muti-armed bandit (MAB) agent receives a predilection towards arm selection that is influenced by a hidden confounder, which also affects the reward. The agent then makes decisions based on the predilection. In their work, the counterfactual question "What would have happened had the agent decided to play differently" is asked. By considering the transfer learning setting as defined in Section 2.3 and interpreting the predilection towards the binary action as a proxy for an underlying binary confounder, the problem described in Section 2.3 is established.*

Appendix D. Derivation of Eq. (6)

We start from the marginal distribution of $w = 1$,

$$\begin{aligned} p^*(w=1) &= \sum_{z\in\{0,1\}} p^*(w=1|z)p^*(z) \\ &= p^*(w=1|z=0)p^*(z=0) + p^*(w=1|z=1)P^*(z=1) \\ &= p^*(w=1|z=0)(1-p^*(z=1)) + p^*(w=1|z=1)p^*(z=1). \end{aligned}$$

Rearranging the equation, we obtain an expression for $p^*(z=1)$ as

$$p^*(z=1) = \frac{p^*(w=1) - p^*(w=1|z=0)}{p^*(w=1|z=1) - p^*(w=1|z=0)}.$$

Note, in the assumption of covariate shift, we have $P^*_{W|Z} = P_{W|Z}$, therefore, we have

$$p^*(z=1) = \frac{p^*(w=1) - p(w=1|z=0)}{p(w=1|z=1) - p(w=1|z=0)}.$$

Appendix E. Proof for Corollary. 7

In this section, we provide the proof for Corollary 7. We first state the corollary again for convenience.

Corollary *7 Given the setting in Section. 2.3 and Assumption. 1, let $\boldsymbol{V} = \{X, Y, Z, W\}$ be the set of all variables, θ_1 be the parameter that parameterize the mechanism generating W from Z and $P(\boldsymbol{V})$ be the joint distribution learned from the source domain π by the two decoders in Fig. 3 and $P'(\boldsymbol{V})$ be the one learned by both encoder and decoders in Fig. 3, the expected gradient w.r.t the parameter θ_1 of the log-likelihood under the target domain π^* will be zero for the former one and non-zero for the latter*

$$\mathbb{E}_{\pi^*}\left[\frac{\partial \log P'(\boldsymbol{V})}{\partial \theta_1}\right] \neq \mathbb{E}_{\pi^*}\left[\frac{\partial \log P(\boldsymbol{V})}{\partial \theta_1}\right] = 0. \tag{11}$$

Proof We prove the corollary in two steps. In the first step, we prove the equality in (11). Then, we prove the inequality in (11). We, therefore, derive two lemmas for the two steps, respectively.

Lemma 10 *Given the setting in Section 2.3 and Assumption 1, let $\boldsymbol{V} = \{X, Y, Z, W\}$ be the set of all variables, θ_1 and θ_2 be the parameters that parameterize the mechanism generating W from Z and Y from Z and X. Furthermore, let $P(\boldsymbol{V})$ be the joint distribution learned from the source domain π by the two decoders in Fig. 3. The expected gradient w.r.t. the parameter θ_1 of the log-likelihood under the target domain π^* will be zero,*

$$\mathbb{E}_{\pi^*}\left[\frac{\partial \log P(\boldsymbol{V})}{\partial \theta_1}\right] = 0. \tag{12}$$

Proof Notice when we use two decoders to approximate $P_{\theta_1}(W|Z)$ and $P_{\theta_2}(Y|Z,X)$, the joint $P(\boldsymbol{V})$ can be factorized as

$$P(\boldsymbol{V}) = P(Z)P_{\theta_1}(W|Z)P_{\theta_2}(Y|Z,X)P(X|Z), \tag{13}$$

according to the casual graph in Fig. 1(*a*). With this factorization, the expected gradient can be simplified as

$$
\begin{aligned}
\mathbb{E}_{\pi^*}\left[\frac{\partial \log P(\boldsymbol{V})}{d\theta_1}\right] &= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial \log P(\boldsymbol{V})}{\partial \theta_1}\right] \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial}{\partial \theta_1}\log P(Z)P_{\theta_1}(W|Z)P_{\theta_2}(Y|Z,X)P(X|Z)\right] && \text{by (13)} \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\Bigg[\underbrace{\frac{\partial}{\partial \theta_1}\log P(Z)}_{=0} + \frac{\partial}{\partial \theta_1}\log P_{\theta_1}(W|Z) \\
&\qquad + \underbrace{\frac{\partial}{\partial \theta_1}\log P_{\theta_2}(Y|Z,X)}_{=0} + \underbrace{\frac{\partial}{\partial \theta_1}\log P(X|Z)}_{=0}\Bigg] \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial}{\partial \theta_1}\log_{\theta_1} P(W|Z)\right] \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial}{\partial \theta_1}\log P^*_{\theta_1}(W|Z)\right] && \text{by Assumption 1} \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\Bigg[\underbrace{\frac{\partial}{\partial \theta_1}\log P^*(Z)}_{=0} + \frac{\partial}{\partial \theta}\log P^*_{\theta_1}(W|Z) && \text{by (13)} \\
&\qquad + \underbrace{\frac{\partial}{\partial \theta}\log P^*_{\theta_2}(Y|Z,X)}_{=0} + \underbrace{\frac{\partial}{\partial \theta}\log P^*(X|Z)}_{=0}\Bigg] \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial \log P^*(\boldsymbol{V})}{\partial \theta_1}\right] \\
&= \sum_{\boldsymbol{v}} p^*(\boldsymbol{v})\frac{\partial \log p^*(\boldsymbol{v})}{\partial \theta_1} && \text{expanding the expectation} \\
&= \sum_{\boldsymbol{v}} p^*(\boldsymbol{v})\frac{\partial \log p^*(\boldsymbol{v})}{\partial p^*(\boldsymbol{v})}\frac{\partial p^*(\boldsymbol{v})}{\partial \theta_1} && \text{chain rule} \\
&= \sum_{\boldsymbol{v}} \frac{\partial p^*(\boldsymbol{v})}{\partial \theta_1} \\
&= \frac{\partial}{\partial \theta_1}\underbrace{\sum_{\boldsymbol{v}} p^*(\boldsymbol{v})}_{=1} = 0.
\end{aligned}
$$

Therefore, we prove the equality in (11). ■

Lemma 11 *Given the setting in Section 2.3 and Assumption 1, let* $\boldsymbol{V} = \{X, Y, W, Z\}$ *be the set of all variables,* ϕ*,* θ_1*, and* θ_2 *be the parameter that parameterize the posterior* $P(Z|W)$*, the*

mechanism generating W from Z, and the mechanism generating Y from Z and X. Furthermore, let $P(\boldsymbol{V})$ be the joint distribution learned from the source domain π by an autoencoder as shown in Fig. 3. The expected gradient w.r.t. the parameter θ_1 of the log-likelihood under the target domain π^* will be non-zero,

$$\mathbb{E}_{\pi^*}\left[\frac{\partial \log P(\boldsymbol{V})}{\partial \theta_1}\right] \neq 0. \tag{14}$$

Proof Notice when we use an autoencoder to approximate $P_\phi(Z|W)$, $P_{\theta_1}(W|Z)$, and $P_{\theta_2}(Y|Z,X)$, the joint $P(\boldsymbol{V})$ can be factorized as

$$P(\boldsymbol{V}) = P_\phi(Z|W)P_{\theta_1}(W|Z)P_{\theta_2}(Y|Z,X)P(X|Z). \tag{15}$$

The first term on the R.H.S. of (15) is $P_\phi(Z|W)$ instead of $P(Z)$ in (13), since we do not have an observation of $P(Z)$ but use an encoder to estimate Z from W. With this factorization, the expected gradient can be simplified as

$$\begin{aligned}
\mathbb{E}_{\pi^*}\left[\frac{\partial \log P(\boldsymbol{V})}{d\theta_1}\right] &= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial \log P(\boldsymbol{V})}{\partial \theta_1}\right] \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial}{\partial \theta_1}\log P_\phi(Z|W)P_{\theta_1}(W|Z)P_{\theta_2}(Y|Z,X)P(X|Z)\right] \quad \text{by (15)} \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial}{\partial \theta_1}\log \frac{P_{\theta_1}(W|Z)P(Z)}{P(W)}P_{\theta_1}(W|Z)P_{\theta_2}(Y|Z,X)P(X|Z)\right] \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\Bigg[\underbrace{\frac{\partial}{\partial \theta_1}\log P(Z)}_{=0} + 2\frac{\partial}{\partial \theta_1}\log P_{\theta_1}(W|Z) - \frac{\partial}{\partial \theta_1}\log P(W) \\
&\quad + \underbrace{\frac{\partial}{\partial \theta_1}\log P_{\theta_2}(Y|Z,X)}_{=0} + \underbrace{\frac{\partial}{\partial \theta_1}\log P(X)}_{=0}\Bigg] \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[2\frac{\partial}{\partial \theta_1}\log P_{\theta_1}(W|Z) - \frac{\partial}{\partial \theta_1}\log P(W)\right] \\
&= \mathbb{E}_{\boldsymbol{V}\sim P^*}\left[2\frac{\partial}{\partial \theta_1}\log P^*_{\theta_1}(W|Z) - \frac{\partial}{\partial \theta_1}\log P(W)\right] \quad \text{by Assumption 1} \\
&= -\mathbb{E}_{\boldsymbol{V}\sim P^*}\left[\frac{\partial}{\partial \theta_1}\log P(W)\right] \neq 0
\end{aligned}$$

The last equality arises from the last six steps in the proof of Lemma 10, and the inequality arises from the fact that $P(W)$ is affected by the parameter θ_1, which can be better understood from the marginalization of W in the binary model (see Appendix D). Therefore, we prove the inequality in (11). ■

Combining Lemma 10 and Lemma 11, we prove Corollary 7. ■

Appendix F. Details on the data generating mechanisms

In this section, we explain the details of the data-generating mechanisms that we used in the experiments. The code for running all experiments is available at https://github.com/Marksmug/CEVAE-Transportability.

Synthetic binary model. To generate data for both the source domain and the target domain, we use the following data-generating process that is compatible with the causal graph in Fig. 1(*a*): $z \sim \text{Bern}(c)$, $w|z \sim \text{Bern}(0.8z + 0.1(1-z))$, $y = x \oplus z$, where $c \in [0, 1]$ is the parameter to control the distribution of Z and $\oplus$ is the XOR operation. Let c_s and c_t be parameters in the source domain and the target domain, respectively. We consider four settings: (1) $c_s = 0.9$ and $c_t = 0.5$; (2) $c_s = 0.5$ and $c_t = 0.9$; (3) $c_s = 0.9$ and $c_t = 0.1$; (4) $c_s = 0.1$ and $c_t = 0.9$.

Synthetic linear Gaussian model. To generate data for both the source domain and the target domain, we employ the following data-generating process: $z \sim \mathcal{N}(u, 1)$, $w = a \odot z + b + \epsilon$, $y = \mathbb{1}\{(z \geq 5 \wedge x = 0) \vee (z < 5 \wedge x = 1)\}$, where $\odot$ is the element-wise multiplication.

IHDP dataset. We first train a data-generating VAE on the IHDP dataset to generate data compatible with the causal graph in Fig. 1(*a*). The data-generating VAE has an encoder and a decoder that uses a 3-layer MLP with a layer width of 30 and ELU activation. Since the IHDP covariate includes continuous variables, categorical variables, and binary variables, we structure the output layer of the decoder in a way that outputs Gaussian distributions for continuous variables, softmax functions for categorical variables, and a logit for binary variables. We sample 20 000 z from $\mathcal{N}(0, 1)$ and use the decoder of the VAE to generate the corresponding proxy w. The reward y is then generated by the function $y = zx + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$. Actions x are generated by firstly fitting a neural network from a scalar dimension of the measurement to the treatment (action variable) in the original data and then sampling the actions from the probability given by the neural network with latent contexts z as the input. To divide the dataset into the source domain and the target domain, we use rejection sampling to sample two Gaussians from 20 000 z and their corresponding w, x, y. For the target domain, we only have the pair (z, w) and generate y based on the current z and the action x for each step.

MNIST dataset. We first train a GAN on the MNIST dataset to generate data compatible with the causal graph in Fig. 1(*a*). The generator of the data-generating GAN takes a three-dimensional Gaussian noise as the input and uses four transpose convolutional layers with batch normalization and ReLU activation for all but the last layer, which employs a tanh activation. The discriminator takes an image as input and applies four convolutional layers, each with LeakyReLU activation, outputting a flattened probability score for each image, which ranges between 0 and 1, indicating the likelihood of the image being real or fake. When generating data, we sample 100 000 latent values from $\mathcal{N}(0, I)$ and use the generator of the GAN to generate corresponding images as w. The first dimension of the latent value is then chosen as the latent context Z. The reward y and action x are generated by two functions, $y = zx + \epsilon$ and $x = \text{Bern}(\sigma(z + 0.5))$, where $\epsilon \sim \mathcal{N}(0, 1)$ and σ is the sigmoid function. To divide the dataset into the source domain and the target domain, we use rejection sampling to sample two Gaussians from 100 000 z and their corresponding w, x, y. For the target domain, we only use the pair (z, w) and generate y based on current z and action x for each step.

Appendix G. Model details

In this section, we give the details of the architecture of VAEs used in Section 4.2 and their training. The Adam optimizer is used for training in the experiments.

G.1. IHDP

Architecture of VAEs. The VAEs used in this experiment have the same architecture as depicted in Fig. 3, with the difference of two separate neural networks for $p_{\theta_2}(y|z,x)$ that are chosen for different x for each forward process. In total, all VAEs have four neural networks: $q_\phi(z|w)$, $p_{\theta_1}(w|z)$, $p_{\theta_2}(y|z,x=0)$, and $p_{\theta_2}(y|z,x=1)$. Similar to the data-generating VAE, we use a 3-layer MLP with a layer width of 30 for each neural network but a different activation function, ReLU. The latent dimension for all VAEs is set to be one as the data-generating VAE. The output layer for $p_{\theta_1}(w|z)$ follows the same structure as the data-generating VAE, and the two neural networks predicting y output a Gaussian.
Training. For all VAEs, we use a replay buffer $\mathcal{B}$ with capacity 1000 and batch size $M = 32$. We set $\beta = 0.1$ to allow for a more flexible posterior and train all VAEs with frequency list $[1000, 200, 100, 50, 25, 15, 10, 8, 6, 5]$ corresponding to the number of gradient steps $[1, 5, 10, 20, 40, 66, 100, 125, 166, 200]$ with a learning rate of 0.005.

G.2. MNIST.

Architecture of VAEs. The VAEs used in this experiment have the same number of neural networks as the ones used in the IHDP dataset. The encoder $q_\phi(z|w)$ has four transpose convolutional layers and one linear layer, all with ELU activation. The decoder $p_{\theta_1}(w|z)$ has one linear layer and four transpose convolutional layers, all with ELU activation except the last transpose convolutional layer. The decoder for predicting y is the same as the two networks used in the IHDP dataset.
Training. For all VAEs, we use a replay buffer $\mathcal{B}$ with capacity 1000 and batch size $M = 32$. We set $\beta = 0.1$ to allow for a more flexible posterior and train all VAEs with frequency list $[1000, 200, 100, 50, 25, 15, 10, 8, 6, 5]$ corresponding to the number of gradient steps $[1, 5, 10, 20, 40, 66, 100, 125, 166, 200]$ with a learning rate of 0.001.

Appendix H. More results for IHDP and MNIST dataset

Motivated by the performance of the causal agent, we further test it in more extreme settings. We truncated the Gaussian distribution of context Z at $z = 0$ to obtain two datasets, with one including all samples with context $z < 0$ and the other one including all samples with context $z \geq 0$. We refer to the former as the negative truncated Gaussian data and the latter as the positive truncated Gaussian data. For the first extreme setting, we use positive truncated Gaussian data as the source domain dataset and negative truncated Gaussian as the target domain dataset. For the second extreme setting, we swap the datasets in the previous setting to create datasets for the source domain and the target domain. In these two cases, there is no overlapping between the source domain and the target domain, and the optimal action is always the opposite in the two domains. We would expect that any knowledge transfer from the source domain is hard to be helpful for the target domain performance. We show the results for both IHDP and MNIST datasets in Fig. 6. The first two columns are the results for the IHDP dataset, with the first column under the first extreme setting and the second column under the second extreme setting. The last two columns are results for the MNIST dataset.

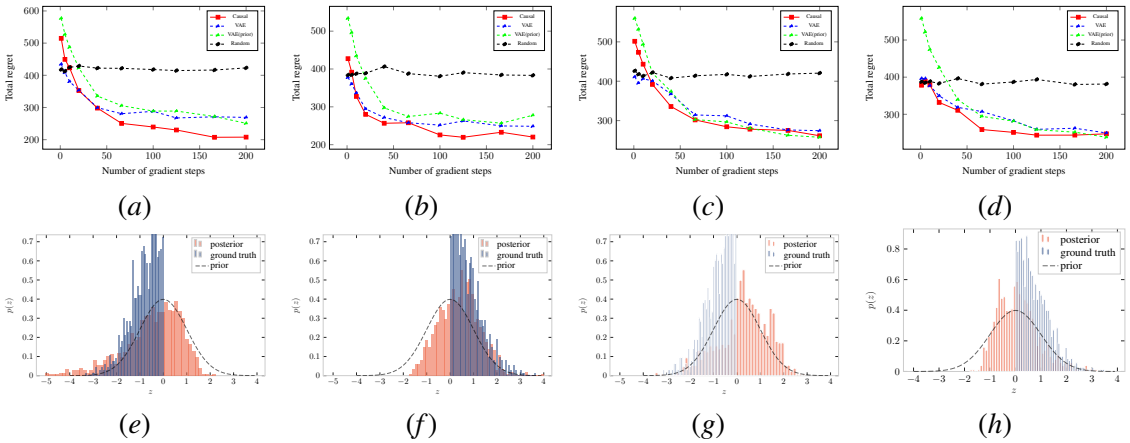


Figure 6: Total cumulative regrets and distribution comparison for IHDP and MNIST dataset in extreme settings.*The first two columns are results for the IHDP dataset, and the last two are for the MNIST dataset. The causal agent shows a constant lower regret compared to baselines, even if there is no overlapping context between the two domains.*

All these results indeed show the negative transfer for VAE (prior) and the causal agent. However, we notice that it takes around 20 gradient steps for the causal agent to catch up with the performance of the VAE agent, while it takes around 40 steps for the VAE (prior) agent to catch up. Moreover, the causal agent almost shows a constant lower regret than the VAE agent after 20 gradient steps. As for the learned posterior, we did not observe a large shift from the prior to the true distribution of Z in the target domain, given that there is no overlapping between P_Z and P_Z^*. Nevertheless, Fig. 6(e) and Fig.6(f) shows a slight shift to P_Z^* in IHDP dataset.