Counterfactual Learning with Multioutput Deep Kernels

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we address the challenge of performing counterfactual inference with observational data via Bayesian nonparametric regression adjustment, with a focus on high-dimensional settings featuring multiple actions and multiple correlated outcomes. We present a general class of counterfactual multi-task deep kernels models that estimate causal effects and learn policies proficiently thanks to their sample efficiency gains, while scaling well with high dimensions. In the first part of the work, we rely on Structural Causal Models (SCM) to formally introduce the setup and the problem of identifying counterfactual quantities under observed confounding. We then discuss the benefits of tackling the task of causal effects estimation via stacked coregionalized Gaussian Processes and Deep Kernels. Finally, we demonstrate the use of the proposed methods on simulated experiments that span individual causal effects estimation, off-policy evaluation and optimization.

1 Introduction

In the recent years, there has been a surge of attention towards the use of machine learning methods for causal inference under observational data. The desire for highly personalized decision-making is indeed pervasive in many disciplines such as precision medicine, web advertising and education (Hodson, 2016). However, in these fields, exploration of policies in the real-world is usually very costly and potentially harmful. Thus, in the attempt to answer counterfactual questions about policy interventions, such as “what would have happened if individual i undertook treatment A instead of treatment B?", one cannot typically rely on randomized experimental data. On the other hand, observational data are abundant and more readily accessible at relatively lower costs, although they suffer from sample selection bias issues arising because of confounding factors. Counterfactual quantities can still be identified and estimated in settings with observed (and in some cases unobserved) confounders, provided that some assumptions hold.

This work is aimed especially at tackling the problem of carrying out counterfactual inference using observational data, with a focus on high-dimensional settings characterized by a large number of covariates, multiple discrete actions (or manipulative variables) and multiple correlated outcomes of interest. To this end, we present a general class of counterfactual multi-task Deep Kernel Learning models (CounterDKL) that efficiently adapt to multiple actions and outcomes settings by exploiting existing correlations, while inheriting the appealing scaling properties of DKL (Wilson et al., 2016) under large samples and large number of predictors, and preserving the Bayesian uncertainty quantification properties of Gaussian Processes (GPs). CounterDKL essentially consists of two joint components: i) a deep learning architecture that learns a lower-dimensional representation of high-dimensional input space; ii) a multitask GP (kernel-based) component (Teh et al., 2005; Bonilla et al., 2008; Álvarez et al., 2012; Bohn et al., 2019) placed on the lower-dimensional representations that can learn the posterior joint distribution of the outcomes conditional on inputs and actions, by avoiding parameter proliferation and stability issues that arise from placing a multitask GP kernel directly on the high-dimensional input space.

Studies with multiple outcomes are quite common in applied research, as policy decisions are rarely based on a single outcome, but rather on a profile of different outcomes that might exhibit positive or negative correlation. As an example, in medical contexts, the decision to prescribe a treatment to a specific patient depends on the primary outcomes of interest, as well as on the possible undesirable side effects.

1.1 Related Work

Many significant contributions in non-parametric regression techniques for counterfactual learning have been made in the literature. These include works on heterogeneous (or individual) treatment effects estimation (Shalit et al., 2017; Künzel et al., 2017; Alaa & van der Schaar, 2017; 2018; Yao et al., 2018; Wager & Athey, 2018; Nie et al., 2020), and on the related field of off-policy evaluation and learning (Dudík et al., 2014; Farajtabar et al., 2018; Kallus, 2018; Athey & Wager, 2021), that focus more on the prescriptive, rather than predictive, goal. Among the several contributions, we particularly draw attention to the early work of Hill (2011) that first proposed Bayesian nonparametric methods (specifically Bayesian Additive Regression Trees) for regression adjustment in estimating causal effects, followed by the extension of Hahn et al. (2020); Caron et al. (2022b), and the seminal work of Alaa & van der Schaar (2017; 2018) who first proposed multitask Gaussian Processes for causal effects estimation tasks, but limited to contexts with binary actions and a single (continuous) outcome. All these works highlight the advantages of Bayesian nonparametric models in terms of flexible non-linear function approximation and uncertainty quantification. We add to the above literature through the following main contributions:

- We review the problem of identifying causal effects under observational data through the formalism of *do*-calculus (Pearl, 2009; Bareinboim et al., 2015), which is easily extended to multiple outcomes problems.
- We extend the class of causal multitask GPs (Alaa & van der Schaar, 2017; 2018) to multiple actions-outcomes designs through a stacked coregionalization model, by highlighting advantages in tackling “poor overlap" regions and disadvantages in terms of scalability.
- We introduce the class of CounterDKL for causal inference under multiple actions/outcomes, discussing their benefits over causal multitask GPs in high-dimensional settings.
- We demonstrate the use of CounterDKL on simulated experiments on causal effects estimation, off-policy evaluation (OPE) and learning off-policy (OPL) problems (Dudík et al., 2011; Dudík et al., 2014; Farajtabar et al., 2018; Kallus, 2021), by providing also an accessible `Python` implementation of the models, based on `GPyTorch`[1].

The advantage of multitask GPs and DKL for causal learning lies in their sample efficiency gains. As observational data often feature imbalanced action arms with relatively few instances, sample splitting can result in under and over-fitting of the surfaces of interest. Multitask GPs and DKL both allow for information sharing when learning actions and outcomes correlated tasks, while only DKL guaratees better scalability and also makes the choice of a GP kernel less challenging, as the deep structure can itself learn arbitrarily complex functions (Wilson et al., 2016).

2 Problem Framework

In order to introduce the problems of causal effects identification and estimation, we borrow the notation from *do*-calculus and Structural Causal Models (SCM) (Pearl, 2009), and start by defining the latter:

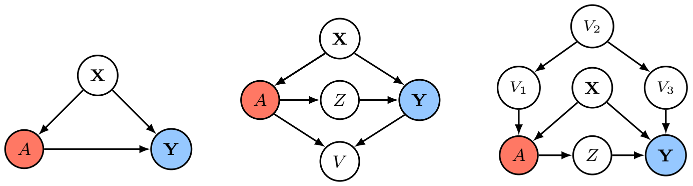
Definition 2.1 (SCM)**.** A Structural Causal Model is a 4-tuple $\langle \mathcal{U}, \mathcal{V}, F, p(u) \rangle$ consisting of (subscript j indicates a random element in the set):

1. $\mathcal{U}$: denoting a set of exogenous variables, defined as variables determined outside of the model.
2. $\mathcal{V}$: denoting a set of endogenous variables, defined as variables determined inside the model.
3. F: a set of functions $f_j \in F$ mapping each element $\varepsilon_j \in \mathcal{U}$ and every parent variables of $V_j \in \mathcal{V}$ — which we denote by $pa(V_j)$ — to the endogenous variables $V_j \in \mathcal{V}$, $f_j : \varepsilon_j \cup pa(V_j) \mapsto V_j$.
4. $p(\varepsilon_j)$: a probability distribution over $\varepsilon_j \in \mathcal{U}$.

A SCM admits an equivalent graphical representation as a causal Directed Acyclic Graph (DAG), where nodes depict variables, i.e. ε_j and V_j, while edges denote the functional causal relationships $f_j \in F$. In the

---

[1] Full code at: **GITHUB TO BE PROVIDED UPON ACCEPTANCE**



(a) (b) (c)

Figure 1: Examples of causal DAGs that satisfy the *backdoor criterion*. Arrows represent causal relationships, and the interest is in the effect of action A on the outcomes $\boldsymbol{Y}$. a) Simple example where covariates $\boldsymbol{X}$ are sufficient to identify the effect $A \to \boldsymbol{Y}$; b) Example where $\boldsymbol{X}$ is sufficient, with a mediator Z and common child V that should never feature in the conditioning set; c) Example where the composite set of covariates $\{V_j\} \cup \boldsymbol{X}$ is sufficient, where j can be either $\{1, 2, 3\}$.

following paragraphs, we examine the problem of identifying and estimating these causal relationships $f_j \in F$, under observational data, relying on SCMs and causal DAGs.

Suppose we have access to an observational dataset $\mathbb{D}_i = \{\boldsymbol{X}_i, A_i, \boldsymbol{Y}_i\} \sim p(\cdot)$, with $i \in \{1, ..., N\}$, where $\boldsymbol{X}_i \in \mathcal{X}$ is a set of covariates, $A_i \in \mathcal{A}$ a set of discrete actions (or manipulative variables), and $\mathbf{Y}_i \in \mathbb{R}^M$ a set of M different outcomes. Here we consider continuous type of outcomes for simplicity, but we extend the methods also to discrete outcomes (Milios et al., 2018), as in the experimental Section 5.2. The overarching goal is to identify and estimate the (average) effects of intervening on the manipulative variable A_i, by setting it equal to some value a, on the outcomes $\mathbf{Y}_i$. We denote the main quantity of interest, the joint interventional probability distribution of all the outcomes conditional on $A_i = a$ by using the *do*-operator as $p(\boldsymbol{Y}|do(A = a))$. We assume that the SCM is fully described by the following pair of equations:

$$\begin{aligned} A_i &= f_A\big(pa(A_i), \varepsilon_{i,A}\big) = f_A(\boldsymbol{X_i}, \varepsilon_{i,A}) \\ \mathbf{Y}_i &= \boldsymbol{f}_Y\big(pa(\mathbf{Y}_i), \varepsilon_{i,Y}\big) = \boldsymbol{f}_Y(\boldsymbol{X}_i, A_i, \varepsilon_{i,Y}) \ , \end{aligned} \tag{1}$$

where: $\boldsymbol{f}_Y(\cdot)$ is a multi-valued function if multiple outcomes are considered; $pa(A_i)$ denotes parent variables (or causes) of A_i; $\varepsilon_{i,j}$ are error terms with a distribution $p(\varepsilon_{i,j})$. The equivalent causal DAG is represented in Figure 1(a), although we stress that the methods presented in this work extend to other types of DAGs such as the ones in Figure 1(b) and 1(c). We make two standard assumption for identification of the causal effect $A \to \boldsymbol{Y}$ in this scenario. The first is *unconfoundedness*, stating that there are no unobserved confounders, or equivalently that the set of observed covariates $\boldsymbol{X}_i \in \mathcal{X}$ is causally sufficient, in the sense that conditioning on $\boldsymbol{X}_i$ allows to identify the causal association between A_i and $\mathbf{Y}_i$. Using Pearl's terminology, we equivalently say that $\boldsymbol{X}$ satisfies the *backdoor criterion* (see A.1 in the appendix for a formal definition), in that it "blocks all backdoor paths" from A to all the outcomes $\boldsymbol{Y}$ (Pearl, 2009). The second assumption is *overlap* (or *positivity*). Overlap requires that $p(A_i = a|\boldsymbol{X}_i = x) \in (\alpha, 1-\alpha)$ — where $\alpha \in (0, \frac{1}{2}]$ — i.e. that the observed actions allocation given $\boldsymbol{X}_i = \boldsymbol{x}$ is never deterministic, so we could theoretically observe data points for which $\boldsymbol{X}_i = \boldsymbol{x}$ in each of the discrete arms of $\mathcal{A}$ (A.2 in the appendix). This ensures that we have comparable units in terms of $\boldsymbol{X}_i$ in each action arm, so we can approximate $\boldsymbol{f}_Y(\boldsymbol{X}_i, A_i = a, \varepsilon_{i,Y})$ well enough. Violation of overlap for portions of $\mathcal{X}$ undermines generalization and extrapolation of model's prediction in those regions; thus, one must be careful as to which subpopulation, defined by a common support $\mathcal{X}_{\text{over}} \subseteq \mathcal{X}$, to target to estimate causal effects. Under these two assumptions, the multivariate interventional distribution $p(\boldsymbol{Y}|do(A = a))$ can be recovered via *backdoor adjustment* as described by the theorem below (proof is provided in Appendix A of supplementary materials).

Theorem 2.2 (Backdoor Adjustment, Pearl (2009))**.** *If* $\boldsymbol{X}$ *satisfies the backdoor criterion and the overlap assumption holds as described above, then the causal effect* $A \to \boldsymbol{Y}$ *can be identified as* $p\big(\boldsymbol{Y}|do(A = a)\big) = p\big(\boldsymbol{Y}|A = a\big)$.

Hence, provided that the covariates $\boldsymbol{X}_i$ (or a subset of them) satisfy the backdoor criterion, we can estimate unbiased causal effects with the observed quantities in $\mathbb{D}_i = \{\boldsymbol{X}_i, A_i, \boldsymbol{Y}_i\}$.

3 Counterfactual Learning with Multitask GPs

For ease of exposition, let us consider the simple case depicted by the causal DAG in Figure 1a, with a single continuous outcome $Y_i \in \mathbb{R}$, with $i \in \{1, ..., N\}$. We tackle the problem of estimating $p(Y|do(A = a))$ via nonparametric regression-adjustment (Johansson et al., 2016; Shalit et al., 2017; Künzel et al., 2017; Nie & Wager, 2020; Caron et al., 2022a). In particular, we assume, in line with most of the previous works, additive noise structure[2], such that the outcome functional can be written as:

$$Y_i = f_Y(\boldsymbol{X}_i, A_i) + \varepsilon_{i,Y} \ , \quad \mathbb{E}(\varepsilon_{i,Y}) = 0 \, . \tag{2}$$

There are different ways in which one can derive an estimator for $p(Y|do(A = a))$ and its moments, e.g. $\mathbb{E}(Y|do(A = a))$, from (2) (Künzel et al., 2017; Caron et al., 2022a). Alaa & van der Schaar (2017; 2018) first proposed the use of multitask learning via Gaussian Process regression, in the specific context of conditional average treatment effects estimation, which is defined, assuming binary $A_i \in \{0, 1\}$, as the quantity $\tau(\boldsymbol{x}_i) = \mathbb{E}[Y_i|do(A_i = 1), \boldsymbol{x}_i] - \mathbb{E}[Y_i|do(A_i = 0), \boldsymbol{x}_i]$. The idea behind causal multitask GPs is to view the $D = |\mathcal{A}|$ interventional quantities $Y_i|do(A_i = a_i)$, where D is the number of discrete action arms, as the output from a vector-valued function $\boldsymbol{f}_Y(\cdot) : \mathcal{X} \mapsto \mathbb{R}^D$ (plus noise), modelled with a GP prior:

$$\boldsymbol{f}_Y(\cdot) \sim \mathcal{GP}\Big(\boldsymbol{m}(\cdot), K(\cdot,\cdot)\Big) \ , \tag{3}$$

with mean $\boldsymbol{m}(\boldsymbol{x}_i) \in \mathbb{R}^D$ and covariance/kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathbb{R}^D \times \mathbb{R}^D$, given two P-dimensional input points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$ for units i and j. Given the likelihood function as a multivariate Gaussian $p(\boldsymbol{y}_i|\boldsymbol{f}_Y, \boldsymbol{x}_i, \Sigma) \triangleq \mathcal{N}(\boldsymbol{f}_Y(\boldsymbol{x}_i), \Sigma)$, where $\Sigma \in \mathbb{R}^D \times \mathbb{R}^D$ is the error covariance diagonal matrix with $\{\sigma_d^2\}_{d=1}^D$ on the diagonal and $\boldsymbol{y}_i \in \mathbb{R}^D$ an output point, the posterior predictive distribution for a train set covariate realization $\boldsymbol{x}_i \in \mathcal{X}$, train set outcome realization $\boldsymbol{y}_i \in \mathbb{R}$ and a test set covariate realization $\boldsymbol{x}_j^* \in \mathcal{X}$ is obtained as, assuming zero prior mean $\boldsymbol{m}(\cdot) = \boldsymbol{0}$ for simplicity:

$$\begin{gathered} p\big(\boldsymbol{f}_Y(\boldsymbol{x}_j^*) \mid (\boldsymbol{x}_i, \boldsymbol{y}_i), \boldsymbol{f}_Y, \phi\big) \triangleq \mathcal{N}\big(\boldsymbol{f}_Y^*(\boldsymbol{x}_j^*), K^*(\boldsymbol{x}_j^*, \boldsymbol{x}_j^*)\big) \ , \\ \boldsymbol{f}_Y^*(\boldsymbol{x}_j^*) = K(\boldsymbol{x}_j^*, \boldsymbol{x}_i) H \boldsymbol{y} \ , \quad K^*(\boldsymbol{x}_j^*, \boldsymbol{x}_j^*) = K(\boldsymbol{x}_j^*, \boldsymbol{x}_j^*) - K(\boldsymbol{x}_j^*, \boldsymbol{x}_i) H K^\top(\boldsymbol{x}_j^*, \boldsymbol{x}_i) \ , \\ \text{where} \quad H = \Big[K(\boldsymbol{x}_i, \boldsymbol{x}_i) + \Sigma\Big]^{-1} \ , \end{gathered} \tag{4}$$

and where ϕ denotes the model parameters and $\boldsymbol{f}_Y^*(\boldsymbol{x}_j^*)$ the function evaluated at a test point $\boldsymbol{x}_j^*$. Under zero prior mean $\boldsymbol{m}(\cdot) = \boldsymbol{0}$, the multitask GP in (3) is fully parametrized by its kernel function $K(\cdot, \cdot)$. The structure of the kernel function in a multitask GP is what induces task-relatedness when fitting the multi-valued surface $\boldsymbol{f}_Y(\cdot)$. In particular, we will generally assume a separable kernel structure (Álvarez et al., 2012), so that each single entry in the prior kernel matrix $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ placed on $\boldsymbol{f}_Y(\cdot)$ is of the form $k_{d,d'}(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) k_A(d, d')$, with $d \in \{1, ..., D\}$. Here, $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represents a base kernel (e.g. linear, squared exponential, Matérn, etc.) while $L_{d,d'} = k_A(d, d')$ is the generic entry of the $D \times D$ **coregionalization matrix** L, which contains the parameters governing task-relatedness over the actions A. In the trivial case of $L_{d,d'} = 0$ we have that tasks d and d' are uncorrelated, i.e. actions d and d' are unrelated in the way they affect outcome Y.

3.1 The coregionalization matrix

The most general specification for the coregionalization matrix L, in the context of separable kernels, is devised by the **linear model of coregionalization** (LMC). Under the LMC, we assume that the set of D functions $\{\boldsymbol{f}_{Y_\mathrm{d}}(\cdot)\}_{d=1}^D$ generates from a linear combination of R_q latent functions drawn from Q independent processes $u_q^k(\boldsymbol{x}_i)$, with $k \in \{1, ..., R_q\}$ and $q \in \{1, ..., Q\}$, such that each $f_{Y_d}(\cdot)$ can be expressed as $f_{Y_d}(\boldsymbol{x}_i) = \sum_{q=1}^Q \sum_{k=1}^{R_q} \alpha_{d,q}^k u_q^k(\boldsymbol{x}_i)$. The corresponding covariance matrix for the multi-valued function

---

[2] Technically this makes the setup semi-parametric, rather than fully nonparametric.

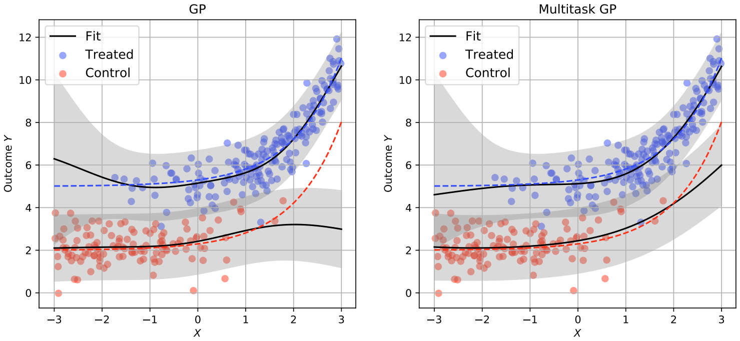


Figure 2: Simple one covariate example, with action space $\mathcal{A} = \{0, 1\}$. Overlap is guaranteed to hold over the whole support $\mathcal{X}$ in the data generating process, i.e. every unit has non-zero probability of being assigned to either $A_i = 1$ or $A_i = 0$, but $p(A_i = 1|X_i)$ is generated as an increasing function of X_i (selection bias). The two underlying counterfactual surfaces $f_{Y_\mathrm{d}}(x_i)$ (dashed lines) display very similar patterns, but GP (left panel) is unable to borrow information from the other arm in poor overlap regions contrary to multitask GP (right panel).

$\boldsymbol{f}_{Y_\mathrm{d}}(\cdot)$, and its single entries, are of the following form:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{q=1}^{Q} B_q \otimes K_q(\boldsymbol{x}_i, \boldsymbol{x}_j)\,, \quad \text{with}$$

$$k\big(f_d(\boldsymbol{x}_i), f_{d'}(\boldsymbol{x}_j)\big) = \sum_{q,q'}^{Q} \sum_{k,k'}^{R_q} \alpha^{i}_{d,q} \alpha^{i'}_{d',q'}\, k_q\big(u^{k}_{q}(\boldsymbol{x}_i), u^{k'}_{q'}(\boldsymbol{x}_j)\big)\,,$$

and where $\otimes$ denotes the Kronecker product, B_q is a matrix of all the $\alpha_{d,q}$ task-relatedness parameters, and $k(\cdot,\cdot), k_q(\cdot,\cdot)$ denote single entries of $K(\cdot,\cdot), K_q(\cdot,\cdot)$ respectively. In our specific case, as in Alaa & van der Schaar (2018), we employ a special case of LMC, named **intrinsic coregionalization model** (ICM) (Bonilla et al., 2008), where the underlying latent process is unique $Q = 1$, so that $K(\boldsymbol{x}_i, \boldsymbol{x}'_i) = B \otimes K_q(\boldsymbol{x}_i, \boldsymbol{x}'_i)$. The ICM specification attempts to avoid severe parameter proliferation in high-dimensional settings with multiple correlated actions $D = |\mathcal{A}|$, while still being capable of capturing task-relatedness through the relatively simple structure of B. However, beside the issue of parameter proliferation when $\mathcal{A}$ features multiple discrete actions, exact GP regression is also known to scale poorly with sample size and cardinality of input space $|\mathcal{X}|$, and direct likelihood maximization methods face issues in over-parametrized models, although some solutions, such as variational methods (Titsias, 2009; Hensman et al., 2013), might be adopted for better scalability.

3.2 Why multitask counterfactual learning?

We know that asymptotically the best approach to estimate the causal quantities $\boldsymbol{f}_Y(\boldsymbol{x}_i)$ would be a “T-Learner" (Künzel et al., 2017; Caron et al., 2022a), which implies splitting the sample and fitting separate models for each arm $A = a$, $\hat{f}(\mathbf{x})_a$. However, in finite sample cases this is rarely the best strategy. By trivially extending the result in Alaa & van der Schaar (2018), we hereby show how increasing action and covariates spaces makes the finite sample problem harder in terms of minimax rates. We consider the Individual Causal Effects (ICE) estimation problem, defined under the SCM specified by (1) and (2) as $\tau_{a,b}(\boldsymbol{x}_i) = f_a(\boldsymbol{x}_i) - f_b(\boldsymbol{x}_i)$ between action a and b. We assume that all the f_a, $\forall a \in \mathcal{A}$ belong to the Hölder ball class of α_a-smooth functions $\mathcal{H}(\alpha_a)$, with $\alpha_a - 1$ bounded derivatives and bounded in sup-norm by a constant $C > 0$. Considering an L^2-norm loss function on $\tau_{a,b}(\boldsymbol{x}_i)$, namely $\mathbb{E}\big[\|\hat{\tau}_{a,b}(\boldsymbol{x}_i) - \tau_{a,b}(\boldsymbol{x}_i)\|^2_{L^2}\big]$, the difficulty of CATE estimation with a nonparametric model $\psi \in \Psi$ can be specified by the optimal minimax rate of convergence, that we define as follows (proof in the Appendix A section).

Corollary 3.1 (Minimax rate)**.** *Assume covariate space is $\mathcal{X} = [0, 1]^P$, and f_a depends on P_a covariates s.t. $P_a \leq P$, $\forall a \in \mathcal{A}$. Define $n_{a,b} < N$ as the subsample identified by action a and b. If both $f_a \in \mathcal{H}(\alpha_a)$ and*

$f_b \in \mathcal{H}(\alpha_b)$*, then CATE optimal minimax rate for a* L^2*-norm loss function is:*

$$\inf_{\hat{\tau}_\psi} \sup_{f_a, f_b} \mathbb{E}\Big[\|\hat{\tau}_{a,b}(\boldsymbol{x}_i) - \tau_{a,b}(\boldsymbol{x}_i)\|^2_{L^2} \Big] \asymp n_{a,b}^{-\left(1+\frac{1}{2}\left(\frac{P_a}{\alpha_a} \vee \frac{P_b}{\alpha_b}\right)\right)^{-1}} \vee \log\left(\frac{P^{P_a+P_b}}{P_a^{P_a} P_b^{P_b}}\right)^{\frac{1}{n_{a,b}}} , \tag{5}$$

where $x \vee y = \max\{x, y\}$ and $\asymp$ is asymptotical equivalence. The first terms on the RHS of 5 relates to the problem of CATE function approximation, while the second term to the degree of sparsity of the CATE. Thus, the optimal minimax rate, which minimizes the loss in the worst case scenario permitted, is asymptotically as complex as the hardest of these two tasks. The "tightness" of this rate depends on the cardinality of the predictor space P and of the subsample defined by action a and b. This implies that, in the presence of multiple discrete actions (where some arms are likely to be very imbalanced), the rate will inevitably grows larger. Thus, in finite samples, the multitask paradigm over actions is well suited to tackle selection bias and particularly estimation in regions with poor overlap, i.e. regions in $\mathcal{X}$ where we mainly observe data points with specific action $A_i = a$ and very few others. Splitting the sample into n_a subgroups and fitting independent models can be very sample inefficient in these settings (5). Multitask GPs can aid extrapolation in such cases of strong sample selection bias, by learning the correlated functions $\{f_{Y_d}(\cdot)\}_{d=1}^{D}$ jointly as $\boldsymbol{f}_Y(\cdot)$. Figure 2 provides a very simple one-covariate example of how multitask learning addresses the issue of extrapolation and prediction in poor overlap regions. Fitting the two surfaces $f_{Y1}(x_i)$ and $f_{Y0}(x_i)$ (dashed lines) through separate GP regressions results in a bad fit out of overlap regions (left panel). Multitask coregionalized GP attempts to fix this problem by embedding the assumption that the two surfaces share similar patterns via joint estimation of $\boldsymbol{f}_{Y_\mathrm{d}}(x_i)$ and their task-relatedness parameters, increasing sample efficiency (right panel). When the two surfaces share minor patterns instead, the optimized parameters should ideally revert back to an independent-tasks GP posterior. The issue of partial overlap might be less severe in scenarios with larger sample size; however, in settings with strong sample selection bias, or settings with multiple discrete actions or action spaces that grow with the sample size, the issue remains relevant.

3.3 Multiple Output Designs

Reverting back to setups with multiple correlated outcomes, we introduce a simple extension to the class of counterfactual GPs presented above that involves an extra multitask learning layer over the M outcomes $\boldsymbol{Y}_i$, in addition to the one placed over the D actions A_i. This extended version featuring *stacked coregionalization*, has a GP prior of the following form:

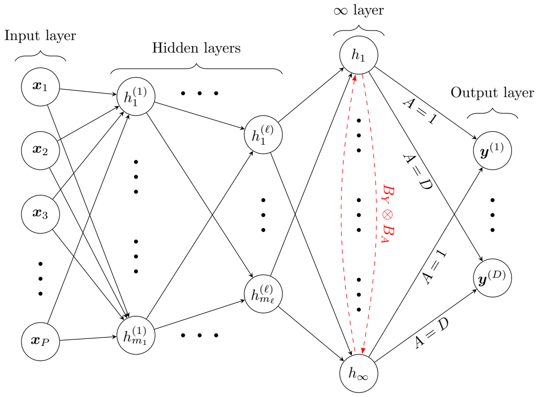
$$\boldsymbol{Y}_i = \boldsymbol{f}_Y(\boldsymbol{x}_i) + \boldsymbol{\varepsilon}_i \ , \quad \mathbb{E}(\boldsymbol{\varepsilon}_i) = \mathbf{0}$$

$$\boldsymbol{f}_Y(\cdot) \sim \mathcal{GP}\big(\mathbf{0}, K_Y(\cdot,\cdot)\big) \ , \quad K_Y(\cdot,\cdot) = B_Y \otimes B_A \otimes K_q(\cdot,\cdot) \ ,$$

where B_Y is the $M \times M$ coregionalization matrix over the outcomes, B_A the $D \times D$ coregionalization matrix over the actions and $K_q(\cdot,\cdot)$ is the base kernel. The vector-valued function $\boldsymbol{f}_Y(\cdot)$ in this case includes all the single-valued functionals $\{f_{d,m}(\cdot)\}_{d,m}^{D,M}$. The extra multitask learning layer over the outcomes $\boldsymbol{Y}_i$ is aimed at increasing sample efficiency by borrowing information among correlated outcomes, as opposed to fitting M separate counterfactual GPs with a single coregionalization layer over A, but it is also conceptually sound, as the quantity of interest is indeed the joint outcomes distribution $p(\boldsymbol{Y}|do(A = a), \boldsymbol{X} = \boldsymbol{x})$, which accounts for and explicitly models correlation between the outcomes, rather than the collection of marginal distributions $\{p(Y_m|do(A = a))\}_{m=1}^{M}$, which leaves correlation unspecified. As we will address in the later section, although the extra layer defined by B_Y allows for higher sample efficiency, it also poses some issues due to parameter proliferation and stability of the optimization problem in high dimensions.

4 Counterfactual Multitask Deep Kernel Learning

Gaussian Processes regressions are known to scale poorly with high dimensions. Their typical computational cost amounts to $\mathcal{O}(n^3)$ for training points and $\mathcal{O}(n^2)$ for test points. Similarly, coregionalized GPs suffer from over-parametrization and instability in the optimization procedure as the number of inputs P and the number of discrete actions D increase. Deep Kernel Learning (DKL) was firstly introduced by Wilson



$$\boldsymbol{h}^{(1)} = g_1(W_1 X + \mathbf{b}_1)\,,$$
$$\dots$$
$$\boldsymbol{h}^{(l)} = g_l(W_l X + \mathbf{b}_l)\,,$$

$$Y = g_\infty(\boldsymbol{h}^{(l)}) + \Xi$$
$$g_\infty(\cdot) \sim \mathcal{GP}(0, \tilde{K}(\cdot,\cdot))\,,$$
$$\tilde{K}(\cdot,\cdot) = B_Y \otimes B_A \otimes K_q(h^{(l)}, h^{(l)'})$$

Figure 3: Counterfactual multitask DKL architecture. The P raw inputs are passed through a deep learning structure with ℓ hidden layers. Multioutput separable kernels (inducing coregionalization over actions A and outcomes $\boldsymbol{Y}$) are then applied to the last Gaussian Process hidden layer, before the M action-specific output layer. Parameters are estimated jointly by minimizing the negative log likelihood.

et al. (2016) with the aim of combining the scalability of Deep Learning methods and the nonparametric Bayesian uncertainty quantification of GPs in tackling prediction tasks in high-dimensional settings. Given a base kernel $k_q(\cdot,\cdot)$ (e.g. linear, squared exponential, etc.), a DKL structure learns a functional $f_Y(\cdot)$ by passing the P inputs $\boldsymbol{X}_i \in \mathcal{X}$ through a deep architecture (a fully-connected feedforward neural network in our case), which maps them to a lower dimensional representation space via non-linear activation functions. The base kernel $k_q(\cdot,\cdot)$ is then applied in this lower dimensional representation space, $k_q(g_\infty(\boldsymbol{x}_i), g_\infty(\boldsymbol{x}_j))$, constituting a final Gaussian Process layer (or an infinite basis functions representation layer). The resulting mathematical object can be described as a kernel being applied to a concatenation of linear and non-linear functions of the inputs, namely $\tilde{K}(\boldsymbol{x}, \boldsymbol{x'}) = K(g_1 \circ \ldots \cdots \circ g_l(\boldsymbol{x}), g_1 \circ \ldots \circ g_l(\boldsymbol{x'}))$ (Bohn et al., 2019). Thus, the DKL architecture is sequentially fully-connected and learnt jointly: the P inputs are passed on to ℓ hidden neural nets layers where the last hidden layer before the GP layer typically maps them to a lower dimensional representation space (with e.g. two hidden units). This is what generates benefits in terms of scalability compared to a classic GP, as the base kernel $k_q(\cdot,\cdot)$ is applied to a lower dimensional representation space, rather than the higher dimensional inputs space directly. Another intrinsic advantage of DKL is that the deep architecture preceeding the GP layer can itself learn arbitrarily complex function, so the choice of a specific GP kernel becomes less cumbersome. For example, Wilson et al. (2016) show that DKL is more robust in recovering step functions, due to weaker smoothness assumptions compared to standard GP kernels.

DKL naturally presents some limitations concerning the more burdensome parameter tuning (e.g., hidden layers and units selection) and the fact that they more easily tend to overfit when overly-parametrized (we refer to Ober et al. (2021) for a more detailed discussion of the issue). The kernel $k(\cdot,\cdot)$ in the last GP layer of a DKL architecture can easily incorporate the separable kernel structure for multitask learning, in the same fashion as the classic GPs presented earlier. In that case, the coregionalization matrix's Kronecker product occurs in the last hidden layer, and features lower dimensional representations instead of the potentially large number of raw inputs, i.e. $K\big(g(\boldsymbol{x}_i), g(\boldsymbol{x}_j)\big) = B_A \otimes K_q\big(g(\boldsymbol{x}_i), g(\boldsymbol{x}_j)\big)$ over the actions $\mathcal{A}$. Similarly, we can induce coregionalization over the M outcomes by adding another coregionalization level as the kernel reads $K\big(g(\boldsymbol{x}_i), g(\boldsymbol{x}_j)\big) = B_Y \otimes B_A \otimes K_q\big(g(\boldsymbol{x}_i), g(\boldsymbol{x}_j)\big)$. Figure 3 graphically depicts a counterfactual multitask Deep Learning architecture, with fully-connected hidden layers, a final (infinite) GP layer and the M action-specific outcomes. The multitask DKL's parameter set comprises the deep neural network's weights W, the base kernel's hyperparameters ϕ (variance, lengthscales, etc.) and the coregionalization matrix B entries, i.e. $\Theta = (W, \phi, B)$ (Appendix D). These parameters are estimated jointly via maximization of the log-marginal likelihood (Wilson et al., 2016; Gardner et al., 2018). In the next section, we will investigate properties of counterfactual multitask GPs and DKL on a variety of simulated experiments.

5 Experiments

The fundamental problem of causal inference is that the interventional quantity $p(\boldsymbol{Y}|do(A=a), \boldsymbol{X}_i = \boldsymbol{x}_i)$ is never observable, so we have to resort to simulation to fully evaluate the methods on individual causal effects (ICE) estimation. We evaluate the performance of counterfactual GPs and counterfactual DKL on a data generating process with three different tasks, and on a real-world example combining experimental and observational data. For the first simulated experiment, we construct the DGP such that the *backdoor criterion* holds for $\boldsymbol{X}_i \in \mathcal{X}$. The GPs and DKL implementations in the simulated examples all make use of the KISS-GP approximation to compute the base kernel covariance matrix as $K_q = MK_{U,U}^{\text{deep}}M^\top$ in the GP layer for better scalability (Wilson & Nickisch, 2015; Wilson et al., 2016)[3].

5.1 Simulated Example

We consider a simulated setting with $D=4$ possible actions $\mathcal{A} = \{0,1,2,3\}$ and $M=2$ correlated outcomes $\boldsymbol{Y} = (Y_1, Y_2) \in \mathbb{R}^2$. Actions and outcomes are generated according to a policy $\pi_b(\boldsymbol{x}_i) = p(A_i = a|\boldsymbol{X}_i = \boldsymbol{x}_i)$ and an outcome function $\boldsymbol{f}_Y(\boldsymbol{x}_i)$, both dependent on the covariates $X_i \in \mathcal{X}$. The probabilistic DGP is fully described in the Appendix B of supplementary materials. The models we compare are the following: i) separate standard GP regressions, employed to fit $f_{Y_d}(\cdot)$ distinctly for each outcome and for each action (**GP**); ii) counterfactual multitask GP regression, with coregionalization over A_i only, meaning that we fit two separate models for each outcome, but a unique multi-valued function model for A_i (**CounterGP**); iii) counterfactual multioutput GP regression, a unique model with coregionalization both over A_i and Y_i (**MOGP**); iv) separate DKL regressions with 3 hidden layers of $[50,50,2]$ units, the equivalent of i) but with deep kernel implementation (**DKL**); v) counterfactual multitask DKL regression with 3 hidden layers of $[50,50,2]$ units, the DKL equivalent of ii) (**CounterDKL**); vi) counterfactual multioutput DKL, the DKL equivalent of iii) (**MODKL**). In particular, we consider two slightly different versions of this setup. In the first version we fix the number of covariates to $P=10$ (only 7 of them being relevant for the estimation) and study the behaviour of the estimators with increasing sample size $N \in \{500, 1000, 1500, 2000, 2500\}$. In the second version we fix sample size to $N=1500$ and study the behaviour of the estimators with increasing number of covariates $P \in \{10, 15, 20, 25\}$. Performance of the models is evaluated on the following three related tasks:

- **ICE**: The first is prediction of Individual Causal Effects (ICE), defined as $\mathbb{E}(Y_i|do(A_i=a), \boldsymbol{X}_i = \boldsymbol{x}_i)$, evaluated through RMSE on a 20% test set.
- **OPE**: The second is Off-Policy Evaluation. This involves estimating the *policy value*, defined as the cumulative reward $\mathcal{V}(\pi_e) = \mathbb{E}_{\mathcal{X},\mathcal{A},\mathcal{Y}}\big[\sum_i \pi_e(a_i|x_i)\big(\boldsymbol{Y}_i|do(A_i=a_i)\big)\big]$ originating, in this case, from the random policy $\pi_e \sim \text{Multinom}(.25, .25, .25, .25)$; evaluated through RMSE.
- **OPL**: The last is Off-Policy Learning, that involves finding the optimal mapping $\pi^*: \mathcal{X} \to \mathcal{A}$ in terms of policy value, i.e. $\pi_p^* \in \arg\max_{\pi_p \in \Pi} \mathcal{V}(\pi_p)$. This last task is evaluated through an accuracy metric, which we label Optimal Allocation Rate (OAR), indicating the percentage of units correctly assigned to action $\pi^*(x_i) = a$ that generates the best outcome for them.

Since we are dealing with $M=2$ outcomes, we produce performance measurements on RMSE and optimal allocation rate for both outcomes and then average them, assuming both outcomes are given equal policy importance and live on the same scale. For both versions of the setup, namely increasing N and increasing P, we replicate the experiment $B=100$ times to obtain Monte Carlo averages and 95% confidence intervals for the metrics. Results are depicted in Figure 4. RMSE performance in all models for increasing N and P behaves accordingly to Corollary 3.1. CounterDKL and MODKL perform consistently better than the GP models, as they scale better with an increasing sample size N and increasing number of predictors P. Particularly MOGP's performance deteriorates for issues related to stability of the marginal likelihood maximization and over-parametrization, as we had to omit it from the study of increasing predictors due to failed convergence for $P > 10$. The advantages over standard DKL regression instead are entirely attributable to sample efficiency gains from multitask coregionalization in CounterDKL and MODKL, both in the increasing N

---

[3] All experiments were run on a Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz, 8Gb RAM CPU.

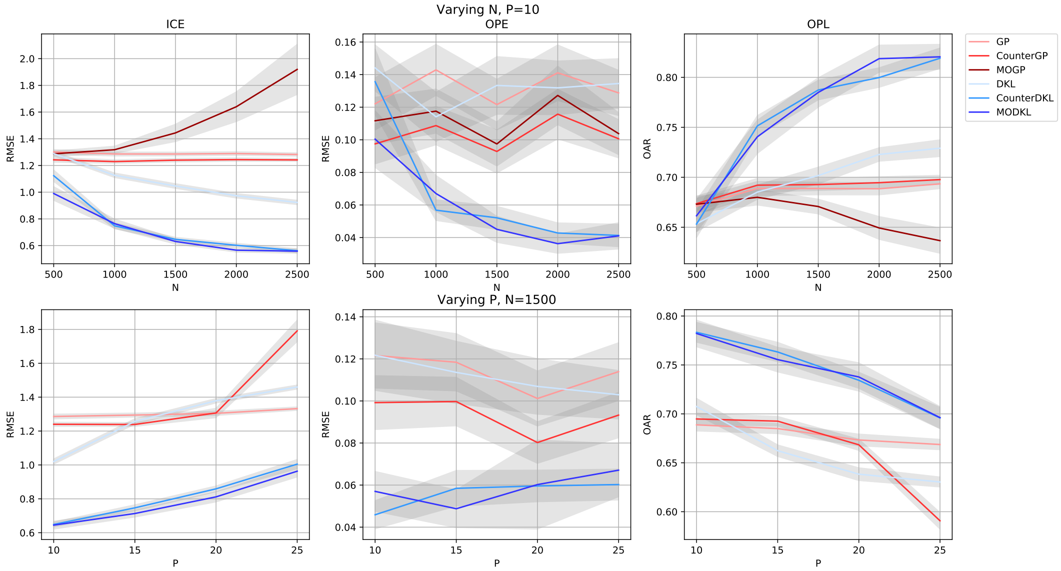


Figure 4: Results on performance of the methods compared, in terms of RMSE or Optimal Allocation Rate (OAR), averaged across $B = 100$ replications for each $N \in \{500, 1000, 1500, 2000, 2500\}$ (first row) and each $P \in \{10, 15, 20, 25\}$ (second row). First column: RMSE evaluated on the individual causal effect (ICE) estimation task (on the test set). Second column: RMSE evaluated on the OPE task. Third column: OAR on the OPL task, defined as percentage of units correctly allocated to the best action among the D ones.

and increasing P studies. In the case of increasing P, we emphasize that as predictor space grows larger the causal DGP becomes relatively sparser (only 7 predictors out of P remain relevant for the estimation), especially in the case of $P = \{20, 25\}$. So in these two cases the batch of DKL models would perhaps achieve better performance from increasing the number of hidden units or hidden layers and adding regularization (dropout, $\ell 1$ or $\ell 2$ regularizers) in the deep architecture part.

5.2 Real-World Example: Job Training Programs and Unemployment

We demonstrate the efficiency of CounterDKL also on a second experiment taken from Shalit et al. (2017), involving a popular real-world study on a job training program, dating back to LaLonde (1986). The distinctive feature of this dataset is that it combines a randomized and an observational subgroup, where the aim is to estimate the effects of participation on a job training program on earnings and employment. Both the outcome (employment at the end of the program) and treatment (participation to the job program) are binary in this case, thus we use a CounterDKL version for multitask probabilistic classification with discrete outcomes, following the work of (Milios et al., 2018). Given the presence of a randomized subsample, we can exploit it to compute unbiased estimates of the quantities of interest and treat them as ground truth. The two quantities of interest in this case are: i) the Average Treatment Effect on the Treated group (ATT), defined as ATT $= T^{-1} \sum_{i=1}^{T_e} y_i - C^{-1} \sum_{i=1}^{C} y_i$, where T and C are the number of treated and control units in the experimental data; ii) the Policy Risk (Shalit et al., 2017), defined as the average error in allocating the treatment according to the ICE estimates policy rule — namely $\pi(\boldsymbol{x}_i) = 1$ if ICE $= \mathbb{E}(Y_i | do(A_i = 1), \boldsymbol{x}_i) - \mathbb{E}(Y_i | do(A_i = 0), \boldsymbol{x}_i) > 0$ — or $\mathcal{R}_{\text{pol}} = 1 - \big[\mathbb{E}\big(Y | do(A_i = 1), \pi(\boldsymbol{x}_i) = 1\big) p(\pi(\boldsymbol{x}_i) = 1) + \mathbb{E}\big(Y | do(A_i = 0), \pi(\boldsymbol{x}_i) = 0\big) p(\pi(\boldsymbol{x}_i) = 0)\big]$. Notice that we cannot measure performance on ICE directly as this is always unobservable in real-world scenarios; also, we restrict analysis of average causal/treatment effects on the treated group since we are sure that overlap holds there, as all the treated units were part of the randomized experiment subgroup, while the observational subgroup is made only of control units. More details about this experiment can be found in the Appendix C of supplementary materials and in Shalit et al. (2017). We compare the following models: i) GP and CounterGP, as in Alaa & van der Schaar (2017); ii) vanilla PCA plus either GP or CounterGP; iii) vanilla deep AutoEncoder plus either GP or CounterGP; iv) DKL and CounterDKL (ours). Results on performance

Model	**Train MAE**	**Test MAE**	**Train $\mathcal{R}_{\text{pol}}$**	**Test $\mathcal{R}_{\text{pol}}$**	**Runtime (s)**
GP	0.033 ± 0.006	0.036 ± 0.008	0.22 ± 0.02	0.27 ± 0.02	171.3 ± 16.1
CounterGP	0.033 ± 0.006	0.035 ± 0.007	0.24 ± 0.01	0.27 ± 0.02	248.6 ± 6.4
PCA + GP	0.073 ± 0.002	0.074 ± 0.003	0.22 ± 0.01	0.27 ± 0.02	66.3 ± 2.4
PCA + CounterGP	0.074 ± 0.001	0.074 ± 0.001	0.23 ± 0.01	0.26 ± 0.02	126.1 ± 3.9
AutoEnc + GP	0.075 ± 0.004	0.075 ± 0.003	0.21 ± 0.03	0.27 ± 0.02	76.0 ± 3.0
AutoEnc + CounterGP	0.076 ± 0.003	0.076 ± 0.003	0.24 ± 0.02	0.30 ± 0.03	138.7 ± 9.2
DKL	0.029 ± 0.011	0.042 ± 0.015	**0.20 ± 0.01**	**0.21 ± 0.02**	44.8 ± 3.3
CounterDKL	**0.011 ± 0.003**	**0.015 ± 0.005**	**0.22 ± 0.01**	**0.25 ± 0.02**	122.7 ± 7.4

Table 1: Train and test set performance on the Jobs data experiment in terms of Mean Absolute Error (MAE) in estimating ATT, Policy Risk ($\mathcal{R}_{\text{pol}}$) and overall runtime (s), with 10-fold cross-validated 95% intervals. Bold indicates best performance.

are gathered in Table 1, in terms of 70%-30% train and test set Mean Absolute Error (MAE) on ATT, Policy Risk $\mathcal{R}_{\text{pol}}$ and average runtime, accompanied by 10-fold cross-validated 95% error intervals. In this example multitasking is induced only over the binary treatment, as we deal with just a single outcome of interest. As the results depict, by operating jointly via a unique loss function, CounterDKL is significantly more efficient than naively applying dimensionality reduction and fitting a multitask GP on a lower dimensional space as two separate steps. It also displays gains over CounterGP, thanks to its deep component that guarantees better computational time (in terms of runtime) and scalability, and is able learn arbitrarily complex functions while imposing weaker smoothness assumptions than standard GP kernels, even on a low-dimensional covariate space example such as the one presented here (7 covariates).

6 Conclusions

Throughout this work, we considered the problem of counterfactual effects learning using observational data, which is of interest in domains where exploration of policies is costly (healthcare, socio-economic sciences, etc.). We reviewed the class of counterfactual GP regression models, extending it to adjust to multiple actions and outcomes settings, and discussed how multitask learning helps in addressing finite sample selection bias. We then introduced a new class of counterfactual models based on Deep Kernel Learning, whose main advantages lie in their more flexible function approximation capabilities and better scalability. While counterfactual GPs struggle to scale up with sample size, number of predictors and number of actions/outcomes to coregionalize over, DKL capitalizes on these components by learning lower dimensional representations. We stress that the class of DKL methods proposed can be easily expanded to carry out counterfactual learning in other more complex scenarios such as: i) unobserved confounding where identification is still possible (instrumental variables); ii) dynamic settings such as dynamic treatment regimes or RL; iii) non-standard data like images (as DKL can incorporate any type of deep architecture).

References

Ahmed Alaa and Mihaela van der Schaar. Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 129–138, 2018.

Ahmed M. Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task Gaussian Processes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 3427–3435, 2017.

Joshua D. Angrist, Guido W. Imbens, and Donald B. Rubin. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91(434):444–455, 1996.

Susan Athey and Stefan Wager. Policy Learning With Observational Data. *Econometrica*, 89(1):133–161, January 2021.

Heejung Bang and James M. Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

Elias Bareinboim, Andrew Forney, and Judea Pearl. Bandits with unobserved confounders: A causal approach. In *Advances in Neural Information Processing Systems 29*, volume 28, 2015.

Bastian Bohn, Christian Rieger, and Michael Griebel. A representer theorem for deep kernel learning. *J. Mach. Learn. Res.*, 20(1):2302–2333, 2019.

Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task Gaussian Process prediction. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

Alberto Caron, Gianluca Baio, and Ioanna Manolopoulou. Estimating individual treatment effects using non-parametric regression models: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2022a. doi: https://doi.org/10.1111/rssa.12824.

Alberto Caron, Gianluca Baio, and Ioanna Manolopoulou. Shrinkage Bayesian Causal Forests for heterogeneous treatment effects estimation. *Journal of Computational and Graphical Statistics*, pp. 1–13, 2022b. doi: 10.1080/10618600.2022.2067549.

Alberto Caron, Gianluca Baio, and Ioanna Manolopoulou. Interpretable deep causal learning for moderation effects. In *ICML 2022, 2nd Interpretable Machine Learning for Healthcare Workshop*, 2022c. URL `https://arxiv.org/abs/2206.10261`.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 1097–1104, 2011.

Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1447–1456, 2018.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7587–7597, 2018.

P. Richard Hahn, Jared S. Murray, and Carlos M. Carvalho. Bayesian Regression Tree Models for Causal Inference: Regularization, Confounding, and Heterogeneous Effects. *Bayesian Analysis*, 15(3):965 – 1056, 2020.

Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. Deep IV: A flexible approach for counterfactual prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1414–1423, 2017.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, pp. 282–290, 2013.

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

R. Hodson. Precision medicine. *Nature*, 547(7619), 2016.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 652–661, 2016.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 3020–3029, 2016.

Nathan Kallus. Balanced policy evaluation and learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8909–8920, 2018.

Nathan Kallus. More efficient policy learning via optimal retargeting. *Journal of the American Statistical Association*, 116(534):646–658, 2021.

Toru Kitagawa and Aleksey Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.

Sören Künzel, Jasjeet Sekhon, Peter Bickel, and Bin Yu. Meta-learners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116, 06 2017.

Robert J. LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American Economic Review*, 76:604–620, 1986.

Dimitrios Milios, Raffaello Camoriano, Pietro Michiardi, Lorenzo Rosasco, and Maurizio Filippone. Dirichlet-based gaussian processes for large-scale calibrated classification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6008–6018, 2018.

Whitney K. Newey and James L. Powell. Instrumental variable estimation of nonparametric models. *Econometrica*, 71(5):1565–1578, 2003.

X Nie and S Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 09 2020.

Xinkun Nie, Emma Brunskill, and Stefan Wager. Learning when-to-treat policies. *Journal of the American Statistical Association*, 0(ja):1–58, 2020.

Sebastian W. Ober, Carl E. Rasmussen, and Mark van der Wilk. The promises and pitfalls of deep kernel learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161, pp. 1206–1216, 2021.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Min Qian and Susan A. Murphy. Performance guarantees for individualized treatment rules. *Ann. Statist.*, 39(2):1180–1210, 04 2011.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 04 1983.

Donald B. Rubin. Bayesian inference for causal effects: The role of randomization. *Ann. Statist.*, 6(1):34–58, 01 1978.

Phillip J. Schulte, Anastasios A. Tsiatis, Eric B. Laber, and Marie Davidian. Q-and a-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science*, 29(4):640–661, 2014.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, volume 70, pp. 3076–3085, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* A Bradford Book, 2018.

Yee Whye Teh, Matthias Seeger, and Michael I. Jordan. Semiparametric latent factor models. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, volume R5, pp. 333–340, 2005.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 2139–2148, 2016.

Jin Tian and Judea Pearl. A general identification condition for causal effects. In *Eighteenth National Conference on Artificial Intelligence*, pp. 567–573, 2002.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, volume 5, pp. 567–574, 2009.

Masatoshi Uehara, Masahiro Kato, and Shota Yasui. Off-policy evaluation and learning for external validity under a covariate shift. In *Advances in Neural Information Processing Systems 33*, 2020.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (KISS-GP). In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 1775–1784, 2015.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pp. 370–378. PMLR, 2016.

Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. In *Advances in Neural Information Processing Systems 31*, pp. 2633–2643, 2018.

Baqun Zhang, Anastasios A. Tsiatis, Eric B. Laber, and Marie Davidian. A robust method for estimating optimal treatment regimes. *Biometrics*, 68(4):1010–1018, 2012.

Yingqi Zhao, Donglin Zeng, A. John Rush, and Michael R. Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

Xin Zhou, Nicole Mayer-Hamblett, Umer Khan, and Michael R. Kosorok. Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, 112(517): 169–187, 2017.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

A Proofs and Discussion

In this first section of supplementary material we provide assumptions, proofs and brief discussion of the two theoretical results in the main paper (Section 2 theorem and Section 3.2 corollary).

A.1 Theorem 2.2

The *backdoor adjustment* theorem (Pearl, 2009) is a well known and understood results in causal inference. Its extension to multi-action and multi-outcome settings is trivial. We briefly reformulate the necessary assumptions as follows.

Assumption A.1 (Backdoor Criterion)**.** Using the causal graph terminology, we denote by $pa(V_j)$ the parents of a specific random variable V_j and by $de(V_j)$ its descendants. We assume that $pa(A, \boldsymbol{Y}) \subseteq \boldsymbol{X},\ de(A, \boldsymbol{Y}) \not\subset \boldsymbol{X}$. Namely $\boldsymbol{X}$ contains (but does not necessarily coincide with) all the common parent variables of $A \in \mathcal{A}$ and $\boldsymbol{Y} = \{Y_m\}_{m=1}^{M} \in \mathcal{Y}$, for all $m \in \{1, ..., M\}$, and does not contain any common descendant of them.

Assumption A.2 (Overlap)**.** Defining the propensity score as $\pi_a(\boldsymbol{x}_i) = p(A_i = a|\boldsymbol{X}_i = \boldsymbol{x}_i)$, we require that $\pi_a(\boldsymbol{x}_i) \in (\alpha, 1-\alpha)$ for all $i \in \{1, ..., N\}$, where $\alpha \in (0, \frac{1}{2})$. The scalar α represents how tight we require the overlap between units in each arm in terms of the covariates to be.

Proof of Theorem 2.2 Under Assumption 1.1 and 1.2, we are able to identify the effect $A \to \boldsymbol{Y} = \{Y_m\}_{m=1}^{M}$, in the form of the joint interventional distribution $p\big(\boldsymbol{Y}|do(A = a)\big)$, as:

$$\begin{aligned} p\big(\boldsymbol{Y} \mid do(A=a)\big) &= \int_{\mathcal{X}} p\big(\boldsymbol{Y} \mid do(A=a), \boldsymbol{x}\big) p\big(\boldsymbol{x} \mid do(A=a)\big) d\boldsymbol{x} \\ &= \int_{\mathcal{X}} p\big(\boldsymbol{Y} \mid a, \boldsymbol{x}\big) p\big(\boldsymbol{x} \mid do(A=a)\big)\, d\boldsymbol{x} \\ &= \int_{\mathcal{X}} p(\boldsymbol{Y} \mid a, \boldsymbol{x}) p(\boldsymbol{x})\, d\boldsymbol{x} = p(\boldsymbol{Y} \mid a), \end{aligned}$$

The above derivation refers to the marginal interventional distribution, with respect to $\boldsymbol{X}$. Often we are interest in the conditional interventional distribution $p\big(\boldsymbol{Y}|do(A = a), \boldsymbol{X}\big)$ instead (e.g., conditional on patient's characteristics), such as when estimating CATE: $\tau(\boldsymbol{x}) = f_1(\boldsymbol{x}) - f_0(\boldsymbol{x})$. The only additional requirement compared to the original version of backdoor adjustment (Pearl, 2009) is that Assumption 1.1 holds for all the collection of outcomes $\boldsymbol{Y}$.

A.2 Corollary 3.1

Corollary 3.1 is a trivial extension of the result on CATE optimal minimax rate derived in Theorem 1 by Alaa & van der Schaar (2018) to discrete multi-action set domains, where specifically $\{0, 1\} \subset \mathcal{A}$. Thus the proof follows straightforwardly from Alaa & van der Schaar (2018). The main difference is the following.

Proof of Corollary 3.1 Optimal minimax rate in Alaa & van der Schaar (2018) define the hardness of CATE estimation between binary action $a, b \in \mathcal{A}$: $\tau_{a,b}(\boldsymbol{x}_i) = f_a(\boldsymbol{x}_i) - f_b(\boldsymbol{x}_i)$. If we have multi-actions space, it means that, given the whole sample size is N, each pair (strictly more than one pair) of discrete actions a, b defines a subsample (and thus a subpopulation) of $n_{a,b} < N$ units. This implies that the hardness of approximating a function in the Hölder ball class and of performing variable selection is proportional to the smaller subsample $n_{a,b}$, not N, which makes the CATE estimation problem harder the smaller $n_{a,b}$.

This is likely to happens when multi-action scenarios feature infrequently explored action arms. Thus, technically speaking, result in Theorem 1 in Alaa & van der Schaar (2018) is a special case of Corollary 3.1 where $\mathcal{A} = \{0, 1\}$.

B Data Generating Processes

We hereby describe the causal data generating processes in the simulated examples of the paper (Section 3.2 and Section 5.1).

B.1 Section 3.2 one covariate example

For the simple one-covariate example in Section 3.2 (Figure 2), where we discuss the benefits of multitask counterfactual learning, we generated $N = 300$ data points from one, uniformly distributed covariate, $X_i \sim \text{Uniform}(-3, 3)$. Then we generated a binary action variable $A_i \sim \text{Bernoulli}\big(p(A_i = 1|x_i)\big)$, where $p(A_i = 1|x_i) = \Phi\big(0.2 + X_i\big)$ and $\Phi(\cdot)$ is the standard normal cdf. Finally, the two counterfactual outcome surfaces were generated as $f_0(x_i) = 2 + 0.3 \exp X_i$ and $f_1(x_i) = 3 + f_0(x_i)$, with the final outcome being $Y = f_0(x_i) + \tau(x_i)A_i + \varepsilon_i$ where $\tau(x_i) = f_1(x_i) - f_0(x_i)$ is the CATE function and $\varepsilon_i \sim \mathcal{N}(0, 0.75^2)$.

B.2 Section 5.1 experiment

The causal data generating process for the simulated experiment of Section 5.1 is described as follows. The P covariates are generated from a uniform distribution $X_{i,j} \sim \text{Unif}(-3, 3)$ for $j \in \{1, ..., P\}$ and $i \in \{1, ..., N\}$. The action allocation policy is simulated according to a multinomial distribution where the probabilities of being assigned to action $A_i = a$ are generated as a softmax function of the covariates $p(A_i = a|\boldsymbol{X}_i = \boldsymbol{x}_i) = \exp\{X_i\boldsymbol{\beta}_a\} \,/\, \sum_{a\in\mathcal{A}} \exp\{X_i\boldsymbol{\beta}_a\}$, where $\boldsymbol{\beta}_a$ is an action-specific P-dimensional sparse vector of action-specific coefficients defined as follow:

$$
\begin{aligned}
\boldsymbol{\beta}_1 &= \begin{bmatrix}-1 & -0.8 & -0.1 & -0.1 & 0 & ... & 0\end{bmatrix}, \\
\boldsymbol{\beta}_2 &= \begin{bmatrix}0 & 0 & 1 & 0.8 & 0.2 & 0 & ... & 0\end{bmatrix}, \\
\boldsymbol{\beta}_3 &= \begin{bmatrix}1.5 & -0.8 & -0.1 & -0.1 & 0 & ... & 0\end{bmatrix}, \\
\boldsymbol{\beta}_4 &= \begin{bmatrix}-1 & -0.8 & -0.1 & -0.1 & 0 & ... & 0\end{bmatrix}.
\end{aligned}
$$

Thus A_i is drawn from a multinomial with vector probabilities parameter $\boldsymbol{p}(A_i = a|\boldsymbol{X}_i = \boldsymbol{x}_i)$. The $M = 2$ action-specific correlated counterfactual outcomes $\boldsymbol{Y}_i \mid do(A_i = a)$ instead are generated as

$$
\boldsymbol{Y}_i \mid do(A_i = a) = \boldsymbol{f}_{\boldsymbol{Y}a}(\boldsymbol{X}_i) + \boldsymbol{\varepsilon}_i\,, \quad \boldsymbol{\varepsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\varepsilon_i}), \quad \text{where:}
$$

$$
\begin{aligned}
f_{Y1_1} &= 3 + 0.4X_0X_1 - 0.3X_2^2 + 0.2\exp(X_3) + 0.6\sin(X_4) \\
f_{Y1_2} &= -1 + f_{Y1_1} + 0.1X_5 \\
f_{Y1_3} &= 1 + f_{Y1_1} + 0.3X_5 \\
f_{Y1_4} &= 0.5 + f_{Y1_1} + 0.5X_6
\end{aligned}
$$

$$
\begin{aligned}
f_{Y2_1} &= 1 + 0.2X_0X_1 - 0.2X_2^2 + 0.1\exp(X_3) \\
f_{Y2_2} &= -2 + f_{Y2_1} + 0.2X_5 \\
f_{Y2_3} &= 2 + f_{Y2_1} + 0.4X_5 \\
f_{Y2_4} &= 1 + f_{Y2_1} + 0.5X_6
\end{aligned}
$$

and where $\text{diag}(\Sigma_\varepsilon) = [\sigma_1, ..., \sigma_4]$, with $\sigma_1 = ... = \sigma_4 = 0.5$, and off-diagonal elements are 0. Finally, we briefly describe the main specifications of the methods compared. The GP models (GP, CounterGP and MOGP) all employed a RBF base kernel, while the DKL models employed a three [50, 50, 2] hidden layers feedforward neural network before the GP ∞-layer, which itself employs a RBF base kernel. The multitask and multioutput models (both GPs and DKLs) all make use of the Intrinsic Coregionalization Model (ICM), such that $K(\boldsymbol{x}_i, \boldsymbol{x}_i') = B_Y \otimes B_A \otimes K_q(\boldsymbol{x}_i, \boldsymbol{x}_i')$. All model were optimized through the Adam solver. More details and fully reproducible code on this experiment can be found in the Github repository: **GITHUB TO BE ADDED UPON ACCEPTANCE**.

C The Job Training Data

The Job Training data (LaLonde, 1986) are a popular case study in the causal inference literature. They comprise a portion of data pertaining to a randomized experiment and a portion of observational data. The randomized experiment features 297 treated and 425 control units; The observational subsample is instead made of 2490 control units only. The binary treatment $A_i \in \{0, 1\}$ denotes participation to the job training program. The original outcome Y_i is earnings after the program, which censored continuous ($Y_i = 0$ for unemployed units). However, following Shalit et al. (2017), we construct a binary indicator $Y_i \in \{0, 1\}$ denoting employment status at the end of the job training program as outcome. This gives us the opportunity to demonstrate the use of the methods presented in this paper also on binary/categorical type of outcomes. To this end we use the classification method for GPs proposed in Milios et al. (2018), where class labels are interpreted as coefficients of a degenerate Dirichlet distribution, which makes the GP classification task efficiently faster and more scalable. The 7 covariates $\boldsymbol{X}_i \in \mathcal{X}$ in the study are the following: age, years of schooling, african american ethnicity, hispanic ethnicity, marital status, high school diploma. Given the randomized subsample of the data, we can obtain an unbiased estimate (computed on the randomized units only) for the Average Treatment Effect on the Treated group (ATT) as $\text{ATT} = T^{-1}\sum_{i=1}^{T_e} y_i - C^{-1}\sum_{i=1}^{C} y_i$, where T and C are the number of treated and control units in the experimental data, and treat this as the ground truth for estimating performance of the methods; and also for the policy risk measure $\mathcal{R}_{\text{pol}} = 1 - \big[\mathbb{E}\big(Y|do(A_i = 1), \pi(\boldsymbol{x}_i) = 1\big)p(\pi(\boldsymbol{x}_i) = 1) + \mathbb{E}\big(Y|do(A_i = 0), \pi(\boldsymbol{x}_i) = 0\big)p(\pi(\boldsymbol{x}_i) = 0)\big]$.

A brief overview on the specifications of the models employed follows. All GPs employ RBF base kernel (also DKL's last layer). DKL and CounterDKL deep NN structure features three [10, 5, 2] hidden layers. The AutoEncoder deep structure employed for the “AutoEnc + GP" and “AutoEnc + CounterGP" models similarly learns a 2-dimensional encoded lower-dimensional representation, where the encoder has two [10, 5] hidden layers before the 2-dim representation and the decoder has [5, 10] hidden layers.

D Marginal Likelihood Maximization in Multioutput Deep Kernels

In the multitask deep kernel learning class of models, the parameter space $\Theta = (W, \phi, B)$ is made of the deep neural network's weights W, the base kernel's hyperparameters ϕ (variance, lengthscales, etc.) and the coregionalization matrix B entries. These parameters are learnt jointly by maximizing the log-marginal likelihood $\mathcal{L}$ at the end of the GP layer. Using the chain rule, the derivatives are:

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial K_\phi} \frac{\partial K_\phi}{\partial g(\boldsymbol{x}, W)} \frac{\partial g(\boldsymbol{x}, W)}{\partial W}$$

$$\frac{\partial \mathcal{L}}{\partial \phi} = \frac{\partial \mathcal{L}}{\partial K_\phi} \frac{\partial K_\phi}{\partial \phi}$$

$$\frac{\partial \mathcal{L}}{\partial B} = \frac{\partial \mathcal{L}}{\partial K} \frac{\partial K}{\partial B}$$

where $g(\boldsymbol{x}, W)$ is the function mapping the inputs to the lower representation space parametrized by W, K_ϕ is the base kernel and $K(\cdot) = B \otimes K_\phi(\cdot)$ is the coregionalized kernel.

E Additional Simulated Experiments

Finally, we describe and present results on a few additional simulated examples that we conducted to assess CounterDKL performance compared to some other specifications seen in Section 5.2, on datasets with varying sample size, predictor space and action space dimensions. In particular, following Dudík et al. (2011) and Farajtabar et al. (2018), we make use of some of the popular datasets for classification in the open-source UCI Machine Learning Repository (`https://archive.ics.uci.edu/ml/index.php`), by transforming the classification task in a causal Off-Policy Evaluation task in the following way. Each dataset is equipped with a pair of covariates $\boldsymbol{X}_i$ and classification labels L_i. We view the classification labels L_i as our discrete actions

Data	$\boldsymbol{N}$	$\boldsymbol{P}$	**# actions**
indian	573	10	2
heart	270	13	2
yeast	1484	8	10
contracept	1473	9	3

Table 2: UCI datasets characteristics.

	GP	CounterGP	DKL	CounterDKL
indian	0.390	0.392	0.376	**0.347**
heart	2.553	1.076	0.433	**0.410**
yeast	0.534	0.657	1.3144	**0.081**
contracep	0.339	**0.003**	0.008	0.007

Table 3: OPE absolute regret on UCI datasets. Bold denotes best performance.

$L_i = A_i$, and consequently generate the action-specific outcome Y_{a_i} as function of the covariates as follows:

$$Y_{a_i} = \exp\{\boldsymbol{X_i}\boldsymbol{\beta_a}\} + \varepsilon_i, \quad \text{where } \mathcal{N}(0, 0.5)$$

and $\boldsymbol{\beta_a}$ is a P-dimensional vector of action-specific coefficients, where entries are $\{0.4, 0.2, 0.0\}$ sampled from a Multinomial$(0.6, 0.25, 0.15)$, with replacement. The datasets utilized are summarized in Table 2 in terms of sample size N, number of covariates P and number of actions. We compare GP, CounterGP, DKL and CounterDKL models on an Off-Policy Evaluation task, where we evaluate the uniformly at random generated policy, via the absoulte regret or risk measure, defined as $\mathbb{E}\big[\mid \mathcal{V}(\pi_e) - \hat{\mathcal{V}}(\pi_e) \mid \big]$. All models employ a RBF base kernel, either directly on the inputs or on the lower dimensional layer. Results averaged over $B = 20$ replications of the experiments for each dataset are gathered in Table 3.