Conservative Policy Construction Using Variational Autoencoders for Logged Data With Missing Values

Mahed Abroshan, Kai Hou Yip, Cem Tekin, *Senior Member, IEEE*, and Mihaela van der Schaar, *Fellow, IEEE*

***Abstract*— In high-stakes applications of data-driven decision-making such as healthcare, it is of paramount importance to learn a policy that maximizes the reward while avoiding potentially dangerous actions when there is uncertainty. There are two main challenges usually associated with this problem. First, learning through online exploration is not possible due to the critical nature of such applications. Therefore, we need to resort to observational datasets with no counterfactuals. Second, such datasets are usually imperfect, additionally cursed with missing values in the attributes of features. In this article, we consider the problem of constructing personalized policies using logged data when there are missing values in the attributes of features in both training and test data. The goal is to recommend an action (treatment) when $\tilde{X}$, a degraded version of X with missing values, is observed. We consider three strategies for dealing with missingness. In particular, we introduce the conservative strategy where the policy is designed to safely handle the uncertainty due to missingness. In order to implement this strategy, we need to estimate posterior distribution $p(X|\tilde{X})$ and use a variational autoencoder to achieve this. In particular, our method is based on partial variational autoencoders (PVAEs) that are designed to capture the underlying structure of features with missing values.**

***Index Terms*— Missing values, observational data, policy construction, variational autoencoder.**

I. Introduction

IN MANY real-life applications, the datasets suffer from various forms of imperfection. Missingness in the attributes of the features is one of the most common types of imperfection [1]. In the problem of constructing policies when there are missing values, one can simply use an imputation method to fill out missing attributes and then use one of the many existing approaches for policy recommendation for the complete dataset. However, this does not reflect the uncertainty in the features. Multiple imputations (MIs) [2] can be used instead of single imputation. In order to combine the recommended actions of different imputed instances, one simple idea is to use the mode of actions, and another possibility is to use a stochastic policy where the probability of choosing an action is proportionate to its frequency. In this work, we address this problem in a systematic way. We suggest using a generative model, partial VAEs (PVAEs) [3], to estimate the probability of different imputed features and use these probabilities as the scores of recommended actions for each particular complete feature. An advantage of using VAEs is that they make weak assumptions about the way the data are generated [4], [5]. Also, it has been shown that they are very effective in capturing the latent structure and the correlations among variables in several tasks [3], [6]–[8]. Using posterior probabilities produced by PVAE, we can estimate the action that maximizes the expected reward. However, simply maximizing the expected reward, given that we have uncertainty about the true feature, might be problematic in sensitive applications such as healthcare since the chosen action that maximizes the expected reward may impose poor reward in some of the less likely scenarios, which is not safe. To address this, we suggest using conservative strategy for policy recommendations. With this strategy, we consider all likely scenarios (we can choose how prudent we need to be via a tuning parameter) and recommend an action that maximizes the reward in the worst case scenario (a max-min criterion).

The main factor that differentiates the problem of learning from observational data from supervised learning is that for each feature, the reward is only known for the prescribed action, i.e., we do not know the counterfactuals. Another complicating factor is that the logging policy (also known as propensity score) is usually not random, and hence, we need to deal with the selection bias. In addition to these two issues, in this work, we are considering that features have missing attributes. Note that, as a consequence of this, we not only do not know counterfactuals but also no longer have access to the actual reward for a given action and complete feature. The goal of this work is to address how one can deal with the uncertainty imposed from the missing attributes. Note that there are other sources of uncertainty in the problem that we are leaving for future work. In particular, here, we are using inverse propensity score (IPS) for dealing with selection bias—a method that is known to have high variance (especially when there are not enough samples for actions with low propensity

Manuscript received March 26, 2021; revised September 23, 2021; accepted December 11, 2021. This work was supported by Wave 1 of the UK Research and Innovation (UKRI) Strategic Priorities Fund under the EPSRC Grant EP/T001569/1 and EPSRC Grant EP/W006022/1, particularly the "Health" and "Criminal Justice System" themes within those grants and The Alan Turing Institute. *(Corresponding author: Mahed Abroshan.)*

Mahed Abroshan is with the Alan Turing Institute, London NW1 2DB, U.K. (e-mail: mabroshan@turing.ac.uk).

Kai Hou Yip is with University College London, London WC1E 6BT, U.K. (e-mail: kai.yip.13@ucl.ac.uk).

Cem Tekin is with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: cemtekin@ee.bilkent.edu.tr).

Mihaela van der Schaar is with the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge CB2 1TN, U.K., also with the Alan Turing Institute, London NW1 2DB, U.K., and also with the University of California, Los Angeles, CA 90095 USA (e-mail: mv472@cam.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3136385.

Digital Object Identifier 10.1109/TNNLS.2021.3136385

scores) [9], [10]. Here, we are not considering this uncertainty and implicitly assume that there are enough samples to have a low variance estimate of the propensity score.

In summary, our contribution is as follows. We propose using a max-min criterion (conservative strategy) when there are missing values in the attributes for sensitive applications. We are proposing a new method based on VAEs for handling missing attributes in the counterfactual estimation problem. In one of our methods, we use the VAE to produce a similarity score to determine how much each of the samples should contribute to the estimation of the outcome for the sample in hand. In the other method, we use a conditional VAE setup to directly estimate the reward via the network. We are using the IPS for dealing with selection bias.

Notation: We use capital letters for random variables, lowercase for realization, boldface letters for vectors, and calligraphic letters for denoting sets.

II. Problem Definition and Related Work

The feature $\boldsymbol{x}$ is a d-dimensional vector belonging to the set $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_d$. Here $\mathcal{X}_i$ can be a set of continuous, integer, or categorical variables. Define $\tilde{\mathcal{X}}_i = \mathcal{X}_i \cup \{*\}$, and now, the observed vector with missing attributes $\tilde{\boldsymbol{x}}$ belongs to $\tilde{\mathcal{X}} = \tilde{\mathcal{X}}_1 \times \tilde{\mathcal{X}}_2 \times \cdots \times \tilde{\mathcal{X}}_d$. Define binary vector $\boldsymbol{M}$, which determines the missingness pattern. $M_i = 0$ means that the ith element is observed and $M_i = 1$ otherwise. We assume missing at random (MAR) mechanism for missingness. This means that the probability of a value to be missing may only depend on the observed data (see [11] for exact definition). For each observed covariate $\tilde{\boldsymbol{x}}$, we can recommend an action $a \in \mathcal{A}$, where $\mathcal{A}$ is a finite set (note that we are not restricting actions to be binary). The reward R given action a and true feature $\boldsymbol{x}$ is drawn from an unknown distribution $R \sim \Phi(R|\boldsymbol{x}, a)$. We denote $\mathbb{E}[R|\boldsymbol{x}, a]$ by $\theta(\boldsymbol{x}, a)$. The available datasets are triplets of $(\tilde{\boldsymbol{X}}_i, A_i, R_i)$:

$$\mathcal{D}^n = \{(\tilde{\boldsymbol{X}}_1, A_1, R_1), \ldots, (\tilde{\boldsymbol{X}}_n, A_n, R_n)\} \tag{1}$$

where actions A_i are produced from an unknown logging policy $\pi_0(A|\tilde{\boldsymbol{X}})$ (also called generalized propensity score). Note that we assume that the treatments in the dataset are administered by only observing $\tilde{\boldsymbol{X}}$, and hence, Fig. 1 represents the causal graph that describes the problem. The conditional distribution of these variables is given in (2). With some abuse of notation, the joint distribution is written as $(\boldsymbol{X}, \tilde{\boldsymbol{X}}, A, R) \sim p(\boldsymbol{X}, \tilde{\boldsymbol{X}}, A, R)$

$$\begin{aligned} &\boldsymbol{X} \sim \mu(\boldsymbol{X}), \quad \tilde{\boldsymbol{X}} \sim p(\tilde{\boldsymbol{X}}|\boldsymbol{X}), \\ &A \sim \pi_0(A|\tilde{\boldsymbol{X}}), \quad R \sim \Phi(R|\boldsymbol{X}, A). \end{aligned} \tag{2}$$

We make the two standard assumptions about the logging policy and rewards in the potential outcome framework [12], [13].

1) *Common Support:* $\pi_0(a|\tilde{\boldsymbol{x}}) > 0$ for all $a \in \mathcal{A}$ and $\tilde{\boldsymbol{x}} \in \tilde{\mathcal{X}}$.
2) *Unconfoundedness With Missing Values:* For each feature vector $\boldsymbol{X}$, the set of possible rewards $\{R(a)\}_{a\in\mathcal{A}}$ is statistically independent of the taken action: $\{R(a)\}_{a\in\mathcal{A}} \perp\!\!\!\perp A|\tilde{\boldsymbol{X}}$.

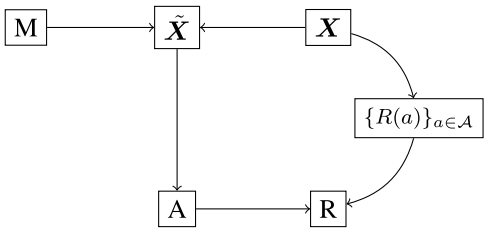

Fig. 1. Causal model for noisy observation problem.

Note that the second assumption can be inferred from our causal graph. This is required for using generalized propensity score $\pi_0(A|\tilde{\boldsymbol{X}})$ as we do in Section IV [14]. The missing data problem frequently arises in machine learning. A motivating example for our model is the following, assume a medical setting where at the time that the treatment was administered to the patient, some of the attributes were missing. This can happen for various reasons, for example, maybe because of the different practices across different hospitals (where some of the attributes are not recorded), lack of certain tests, or maybe emergency situations to name a few. Note that since the treatment was administered only by observing $\tilde{\boldsymbol{X}}$, the causal model of Fig. 1 holds.

A. Related Work

We are considering the problem of off-policy evaluation (also known as offline evaluation in bandit literature). Here, we are using the IPS reweighting method [15], [16] to deal with selection bias. We are using IPS in a deep network model, and from this point of view, our work is mostly related to [17], [18]. Direct method is another method for counterfactual estimation where the goal is to learn a function, mapping pairs of actions and features to rewards [19]. A doubly robust method combines the former two approaches [20], [21]. Note that none of these works consider missing attributes in the feature. Hoiles and van der Schaar [22] provided an off-policy evaluation method based on the regression estimator given in [23] and [24] when there are missing pairs of action and feature in the dataset. However, they do not consider missing attributes in the feature.

The treatment effect estimation problem is another related line of work, where the goal is to find the causal effect of a certain intervention (treatment) on the population or on individuals. The missing value problem is discussed in this framework from early on [14], [25]. The use of generalized propensity scores computed via MIs is suggested in [26]. A summary of several early works can be found in [27]. More recently, in [28], matrix factorization has been proposed for estimating confounders from noisy covariates (also includes covariates with missing values). In [29], a doubly robust-based method is suggested. Parbhoo *et al.* [30] considered missing values only during test time and suggested a method based on the information bottleneck technique. Finally, Mayer *et al.* [31] suggested a new method based on VAEs (adopted for missing values), which learns distribution of the latent confounder and hence assumes a weaker condition than unconfoundedness with missing values, which is harder to justify. In Section V,

we compare our results with several of these recent works. Some of the other notable works in the treatment effect literature are [6] and [32]–[34]. However, they do not consider missing values. Thus, we will compare our results with [6], by imputing the missing values to get complete features and train and use the algorithm on the complete feature.

The problem studied in this work can be considered as an offline version of contextual bandits problem [35]–[37]. There are several works in bandit literature (and more generally in reinforcement learning) that are related to conservative strategy, e.g., [38]–[40]. The goal in bandit literature is to minimize regret, and conservatism in this area means guaranteeing that we are not performing poorly in the process of achieving a low regret (exploration). This is fundamentally different from conservatism in our problem, which is due to uncertainty in the feature.

Our method is based on PVAE introduced in [3]. A few other VAE-based methods are also suggested for the imputation task [4], [41]. Methods based on PVAE have been suggested for other tasks. In [42], they use PVAE for hybrid recommender system, and in [43], they use PVAE for elementwise training data acquisition.

III. Strategies for Finding the Best Action

In this section, we discuss three different strategies for action recommendation when there is uncertainty (in our problem due to missing attributes) in the features. In Section III-A, we address all three of these strategies using two different methodologies.

Assume that for feature $\boldsymbol{x}$, the action that gives the highest expected reward is denoted by $a(\boldsymbol{x})$,

$$a(\boldsymbol{x}) = \arg\max_{a} \theta(\boldsymbol{x}, a). \tag{3}$$

Since we observe $\tilde{\boldsymbol{x}}$, the degraded version of $\boldsymbol{x}$, there is uncertainty in the true value of $a(\boldsymbol{x})$. This uncertainty can be quantified using the Shannon entropy $H(a(\boldsymbol{X})|\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{x}})$. The following proposition presents an expansion for this quantity.

Proposition 1: The uncertainty in the best action $a(\boldsymbol{X})$ when we observe $\tilde{\boldsymbol{X}} \sim p(\tilde{\boldsymbol{X}}|\boldsymbol{X})$ can be expressed as follows:

$$H(a(\boldsymbol{X})|\tilde{\boldsymbol{X}}) = H(a(\boldsymbol{X})) - (I(\boldsymbol{X}; \tilde{\boldsymbol{X}}) - I(\boldsymbol{X}; \tilde{\boldsymbol{X}}|a(\boldsymbol{X})). \tag{4}$$

Proof: See the proof in the Appendix. ■

The first term of the right-hand side, $H(a(\boldsymbol{X}))$, represents the uncertainty in $a(\boldsymbol{X})$ itself. This can be interpreted as the complexity of the function $a(\cdot)$. For example, in the extreme case when there is a single action that is always the best action, then $H(a(\boldsymbol{X})) = 0$. The second term $I(\boldsymbol{X}; \tilde{\boldsymbol{X}})$ is the mutual information between $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$, which represents the quality of the channel between these two variables, and this channel is characterized by the conditional distribution $p(\tilde{\boldsymbol{X}}|\boldsymbol{X})$, i.e., $I(\boldsymbol{X}; \tilde{\boldsymbol{X}})$ shows how much information is passed to $\tilde{\boldsymbol{X}}$ from $\boldsymbol{X}$. The last term is subtracting the amount of information passed to $\tilde{\boldsymbol{X}}$ that is irrelevant to $a(\boldsymbol{X})$. The probability that the best algorithm finds $a(\boldsymbol{X})$ by observing $\tilde{\boldsymbol{X}}$ is given by $(1/2^{H(a(\boldsymbol{X})|\tilde{\boldsymbol{X}})})$. A simple example is given in the Appendix for which we compute these quantities, and we leave further discussion about fundamental limits to future work. We reiterate that in this work, we ignore the uncertainty in the true value of reward, i.e., the uncertainty in the estimation of $\theta(\boldsymbol{x}, a)$. The above analysis holds for any degradation of the input. In particular, in this work, we are considering missingness. The three strategies below can be used to deal with this type of uncertainty.

A. Imputation

The first strategy is to use an imputation algorithm in order to find the most likely feature $\boldsymbol{x}$ given the observed incomplete feature $\tilde{\boldsymbol{x}}$, i.e.,

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\tilde{\boldsymbol{x}}). \tag{5}$$

Then, the recommended action $a_I(\tilde{\boldsymbol{x}})$ can be found by maximizing the reward for $\hat{\boldsymbol{x}}$

$$a_I(\tilde{\boldsymbol{x}}) = a(\hat{\boldsymbol{x}}) = \arg\max_{a} \theta(\hat{\boldsymbol{x}}, a). \tag{6}$$

B. Maximum Expected Reward

The imputation strategy recommends the action only based on one possible complete feature. This does not account for the uncertainty in the true feature. A natural way is to directly maximize the expected reward instead of finding a single potential complete feature. Assuming that attributes are discrete and $|\mathcal{X}|$ is finite (the summation below should be replaced with an integral if this is not the case), the expected reward when $\tilde{\boldsymbol{x}}$ is observed for a given policy like $\pi(A|\tilde{\boldsymbol{X}})$ can be computed

$$\begin{aligned}\mathbb{E}_{\pi}(R(\tilde{\boldsymbol{x}})) &= \sum_{a,\boldsymbol{x}} \theta(\boldsymbol{x}, a)\pi(a|\tilde{\boldsymbol{x}})p(\boldsymbol{x}|\tilde{\boldsymbol{x}}) \\ &= \sum_{a} \pi(a|\tilde{\boldsymbol{x}}) \sum_{\boldsymbol{x}} \theta(\boldsymbol{x}, a)p(\boldsymbol{x}|\tilde{\boldsymbol{x}}).\end{aligned}$$

The policy π that maximizes this expectation is a deterministic policy that recommends $a_M(\tilde{\boldsymbol{x}})$ defined as follows:

$$a_M(\tilde{\boldsymbol{x}}) = \arg\max_{a} \sum_{\boldsymbol{x}} \theta(\boldsymbol{x}, a)p(\boldsymbol{x}|\tilde{\boldsymbol{x}}). \tag{7}$$

MI method is an approximation for this strategy, where we consider several possible complete features and recommend an action that maximizes the average reward over the imputed samples. This method is widely used in the literature (see [27], [31], [44]).

C. Conservative Strategy

In sensitive applications, the strategies presented above may not be acceptable, because in these applications, we have to avoid less likely (but still possible) scenarios for which a very low reward is expected (e.g., death in healthcare application). To achieve this, we suggest a max-min criterion that recommends the action that maximizes the reward in the worst case scenario, which is likely "enough"

$$a_C(\tilde{\boldsymbol{x}}) = \arg\max_{a} \min_{\boldsymbol{x}: p(\boldsymbol{x}|\tilde{\boldsymbol{x}}) > cp(\hat{\boldsymbol{x}}|\tilde{\boldsymbol{x}})} \theta(\boldsymbol{x}, a). \tag{8}$$

Here, the constant $0 \leq c < 1$ determines how prudent we want to be [$\hat{x}$ is defined in (5)]. If we choose $c = 0$, we get

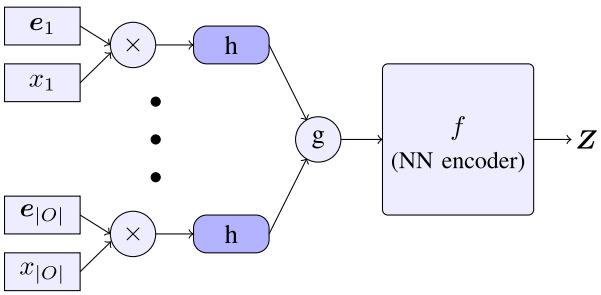

Fig. 2. Encoder of PVAE (PNP setting).

the most conservative policy, where we essentially ignore observed input $\tilde{\boldsymbol{x}}$ and choose the action that has the highest minimum reward for all inputs, while $c \to 1$ is equivalent to the imputation method.

Remark: If we define $R = \int_S p(\boldsymbol{x}|\tilde{\boldsymbol{x}})d\boldsymbol{x}$ where $S = \{\boldsymbol{x} \mid p(\boldsymbol{x}|\tilde{\boldsymbol{x}}) < cp(\hat{\boldsymbol{x}}|\tilde{\boldsymbol{x}})\}$, then R is representing the risk of not considering the true feature in the process of recommending the action. When we choose $c = 0$, we have $R = 0$, and it increases with c. The parameter c then can be thought of as a tuning parameter for this risk. As a proxy, we can model $p(\boldsymbol{x}|\tilde{\boldsymbol{x}})$ with a multivariate Gaussian distribution and can consider $\hat{\boldsymbol{x}}$ to be the center of the distribution. Thus, we can compute this risk for a given c.

IV. Estimation Methods

In this section, we suggest two methods for counterfactual estimation. We show how we can implement the three strategies discussed in Section III using these two methods. In the core of both methods, we use PVAE. In the first method, we train the network using only $\tilde{\boldsymbol{x}}$ as the input, which will produce a similarity score for two features. We will use this similarity score to estimate the reward (we call this method SPVAE). In the second method (called CPVAE), we train a conditional VAE [45] using $\tilde{\boldsymbol{x}}$ incomplete context, and the reward (conditioned on action), and we will use the network for both estimating $p(\boldsymbol{x}|\tilde{\boldsymbol{x}})$ and also estimating the expected rewards $\theta(\boldsymbol{x}, a)$.

A. Partial Variational Autoencoder

We will be using the encoder of partial VAE that was introduced in [3]. In particular, we use the pointnet plus (PNP) setting. The structure of the encoder is represented in Fig. 2. PVAE is designed to deal with the missingness in the input and its structure allows the input dimension to vary. Assume that $x_{i_1}, \ldots, x_{i_{|O|}}$ are the observed attributes of the feature, and each observed attribute x_{i_j} will be multiplied by an embedding vector $\boldsymbol{e}_j$ that will represent the position of the observed attribute. Denote the elementwise multiplication of $\boldsymbol{e}_j$ and $\boldsymbol{x}_{i_j}$ by $\boldsymbol{s}_j = \boldsymbol{x}_{i_j} * \boldsymbol{e}_j$. Now, $\boldsymbol{s}_j$'s will be fed to h, a shared neural net. Then, there is a permutation invariant function g (in our setup, g is a summation similar to [3]) that maps $(h(s_1), \ldots, h(s_{|O|}))$ to $\mathbb{R}^k$ (k is a hyperparameter). Finally, this k-dimensional latent variable will be fed to a fully connected network $f()$. Therefore, we have $\boldsymbol{Z} = f\big(g(h(s_1), \ldots, h(s_{|O|}))\big)$. We refer to [3] for a more detailed discussion about the encoder.

For the decoder of PVAE, we use a fully connected network. Inspired from the decoder in the HI-VAE model [46], we consider the following distributions for different types of variables and map $\boldsymbol{Z}$ to the parameters of an appropriate distribution. This will enable us to handle heterogeneous features of the context. For continuous variables, we have $p(x_i|\boldsymbol{Z}) = \mathcal{N}(\mu_i(\boldsymbol{Z}), \sigma_i(\boldsymbol{Z}))$, where $\mu_i(\boldsymbol{Z})$ and $\sigma_i(\boldsymbol{Z})$ are outputs of the neural network with input $\boldsymbol{Z}$. For categorical attributes, we use one-hot encoding, and the posterior distribution is given by a softmax $p(x_i = j|\boldsymbol{Z}) = (\exp^{-s_j(\boldsymbol{Z})}/\sum_{t=1}^{m} \exp^{-s_t(\boldsymbol{Z})})$, where $s_t(\boldsymbol{Z})$ is the output of the decoder corresponding to the tth category. The loss function is similar to the evidence lower bound (ELBO) used for training of PVAE in [3]

$$\begin{aligned} \log p(\tilde{\boldsymbol{X}}) &\geq \log p(\tilde{\boldsymbol{X}}) - D_{KL}\big(q(\boldsymbol{Z}|\tilde{\boldsymbol{X}})||p(\boldsymbol{Z}|\tilde{\boldsymbol{X}})\big) \\ &= \mathbb{E}_{\boldsymbol{Z}\sim q(\boldsymbol{Z}|\tilde{\boldsymbol{X}})}\big[\log p(\tilde{\boldsymbol{X}}|\boldsymbol{Z})\big] - D_{KL}\big(q(\boldsymbol{Z}|\tilde{\boldsymbol{X}})||p(\boldsymbol{Z})\big). \end{aligned} \tag{9}$$

We consider normal distribution for $p(\boldsymbol{Z}) = \mathcal{N}(0, 1)$. Similar to [3] and [46], we assume that the two following equations hold. The first one states the independence of attributes given the latent variable, i.e.,

$$p(\boldsymbol{x}|\boldsymbol{Z}) = \prod_{i=1}^{d} p(x_i|\boldsymbol{Z}). \tag{10}$$

The second one states that all the information about unobserved attributes in $\tilde{\boldsymbol{x}}$ is encoded into $\boldsymbol{Z}$, i.e., if $\boldsymbol{x}_M$ represents the set of missing attributes, then we have

$$p(\boldsymbol{x}_M|\tilde{\boldsymbol{x}}, \boldsymbol{Z}) = p(\boldsymbol{x}_M|\boldsymbol{Z}). \tag{11}$$

B. SPVAE

We suggest using the following simple estimator for finding $\hat{\theta}(\boldsymbol{x}, a)$:

$$\hat{\theta}(\boldsymbol{x}, a) = \sum_{i=1}^{N} w_i \frac{\mathbb{1}[A_i = a]\, R_i}{\hat{\pi}_0(a|\tilde{\boldsymbol{X}}_i)} \tag{12}$$

where $w_i = (p(\boldsymbol{x}|\tilde{\boldsymbol{X}}_i)/\sum_{j=1}^{N} p(\boldsymbol{x}|\tilde{\boldsymbol{X}}_j))$ are similarity scores corresponding to each of the data samples and $\hat{\pi}_0(a|\tilde{\boldsymbol{X}}_i)$ is the estimation of the propensity score. We explain how to compute $\hat{\pi}_0(a|\tilde{\boldsymbol{X}}_i)$ in Section IV-C. Essentially, w_i shows how much the reward of sample $\tilde{\boldsymbol{X}}_i$ is relevant for estimating the reward for $\boldsymbol{x}$. The IPS term adjusts for the selection bias in the data. For estimating $p(\boldsymbol{x}|\tilde{\boldsymbol{X}}_i)$, we can feed $\tilde{\boldsymbol{X}}_i$ to PVAE, and the output of the network gives the required posterior distribution. Recall that we assumed the Gaussian distribution for the output of the VAE. Using (10) and (11), we have

$$p(\boldsymbol{x}|\tilde{\boldsymbol{X}}_i) = \prod_{j=1}^{d} p(x_j|\boldsymbol{Z}). \tag{13}$$

A variation of SPVAE method that computes $\hat{\theta}(\boldsymbol{x}, a)$ in a slightly different way is proposed in the Supplementary Material.

Remark: Notice that the summation in (12) may become computationally costly. If this is the case, one can randomly choose $M < N$ samples from the dataset and estimate $\hat{\theta}(\boldsymbol{x}, a)$

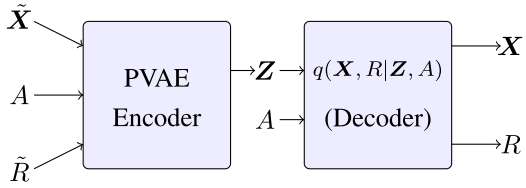

Fig. 3. CPVAE structure.

only using those M samples. The CPVAE method that we propose next will not have this issue since it can estimate the reward with a singe forward pass through a network.

C. Propensity Score

For computing $\hat{\pi}_0(A|\tilde{\boldsymbol{X}})$, we first use an MI method to produce multiple complete datasets. Any standard MI method such as [47] or [48] can be used. Then, we fit a standard multinominal logistic regression (LR) model similar to [17] on the completed features. In the test time, we average the propensity score over MIs. This is a classical method that is well studied in the literature [26], [27], [44]. (It is known that averaging the propensity score before performing casual inference gives a better result [49].) A more advanced method for estimating propensity scores with missing values is recently introduced in [29]. We leave exploring effect of using more advanced methods for future work.

D. CPVAE

In this section, we modify PVAE and use it as an end-to-end network to produce an estimation of $\theta(\boldsymbol{x}, a)$. We use conditional VAE and call this method CPVAE. First, the input of the CPVAE is different. During training, the input is a subset of rewards, actions, and observed attributes that we represent with $(\tilde{\boldsymbol{X}}_i, A_i, \tilde{R}_i)$ (by denoting the rewards with $\tilde{R}$, we highlight that they might be missing from the input of the network). The idea is that in the test time, the reward can be treated as a missing attribute of the input, i.e., the input at test time will be the observed attributes and action $(\tilde{\boldsymbol{X}}, A, *)$. The decoder network attempts to reconstruct $(\boldsymbol{X}, R)$, and hence, it will produce an estimation for the reward (see Fig. 3). This method has the advantage that the correlations among different attributes of feature, reward, and action are expected to be captured by the latent variable $\boldsymbol{Z}$. Also, in comparison with SPVAE for producing the estimated rewards, we do not need to compute the summation in (12) and can get the reward with a single forward pass through the network. (Note that it is expected to have a better quality of imputation using outcome in the imputation process [27], [50].)

1) Loss Function: The standard ELBO loss function for CPVAE can be written as follows:

$$\begin{aligned}\log p(\tilde{\boldsymbol{X}}, \tilde{R}|A) &\geq \log p(\tilde{\boldsymbol{X}}, \tilde{R}|A) \\ &\quad - D_{KL}\big(q(\boldsymbol{Z}|\tilde{\boldsymbol{X}}, \tilde{R}, A)||p(\boldsymbol{Z}|\tilde{\boldsymbol{X}}, \tilde{R}, A)\big) \\ &= \mathbb{E}_{\boldsymbol{Z}\sim q(\boldsymbol{Z}|\tilde{\boldsymbol{X}}, \tilde{R}, A)}\big[\log p(\tilde{\boldsymbol{X}}, \tilde{R}|\boldsymbol{Z}, A)\big] \\ &\quad - D_{KL}\big(q(\boldsymbol{Z}|\tilde{\boldsymbol{X}}, \tilde{R}, A)||p(\boldsymbol{Z}|A)\big). \qquad (14)\end{aligned}$$

In order to account for the selection bias in the loss function, we use the IPS technique and write the final loss as

$$\begin{aligned}\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\sum_{a\in\mathcal{A}}\Bigg(&\log p(\tilde{\boldsymbol{X}}_i|\boldsymbol{Z}, A) + \log p(R_i|\boldsymbol{Z}, A) \\ &- D_{KL}\big(q(\boldsymbol{Z}|\tilde{\boldsymbol{X}}, A, R)||p(\boldsymbol{Z}|A)\big)\Bigg)\frac{\mathbb{1}[A_i = a]}{\hat{\pi}_0(a|\tilde{\boldsymbol{X}}_i)}. \qquad (15)\end{aligned}$$

Notice that $\mathbb{E}[\mathcal{L}]$ is equal to the lower bound in (14). The IPS term can also be interpreted as a method to force the autoencoder to learn rare action–feature pairs more carefully by penalizing the loss function.

E. Implementing Strategies

In this section, we explain how to implement three strategies using our two suggested methods.

1) *Imputation:* For SPVAE, when $\tilde{\boldsymbol{x}}$ is observed, we first find the output of PVAE to impute the missing attributes of $\tilde{\boldsymbol{x}}$ (we do not change the values that are not missing). Denote the complete feature vector by $\boldsymbol{x}$. We now use (12) to estimate $\hat{\theta}(\boldsymbol{x}, a)$ for all possible actions a and then recommend the action that maximizes $\hat{\theta}(\boldsymbol{x}, a)$. For CPVAE, we simply feed $\tilde{\boldsymbol{x}}$ along with different actions to the network and choose the action with the highest expected reward.
2) *Maximum Expected Reward:* For SPVAE, we feed $\tilde{\boldsymbol{x}}$ to PVAE and then sample t times from $q(Z|\tilde{\boldsymbol{x}})$ (t is a hyperparameter) to get $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t$. Denote the imputed output of the decoder network of these t latent variables by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$. For all $a \in \mathcal{A}$ and $\boldsymbol{x}_i$, we use (12) to compute $\hat{\theta}(\boldsymbol{x}_i, a)$. Then, we recommend a, which maximizes the average reward of these t inputs. We do similarly for CPVAE, and the estimation of $\hat{\theta}(\boldsymbol{x}_i, a)$ will be done by observing the output of CPVAE.
3) *Conservative:* We need to compute the following expression for all $a \in \mathcal{A}$: $\min_{\boldsymbol{x}: p(\boldsymbol{x}|\tilde{\boldsymbol{x}}) > cp(\hat{\boldsymbol{x}}|\tilde{\boldsymbol{x}})} \theta(\boldsymbol{x}, a)$. First, we pass $\tilde{\boldsymbol{x}}$ through PVAE to get $\hat{x}$ and the posterior distribution $p(\boldsymbol{x}|\tilde{\boldsymbol{x}})$. We produce u samples from the generator model through random sampling, that is, we can randomly sample u times from $P(\boldsymbol{Z})$ to get $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_u$ [recall that $p(\boldsymbol{Z}) = \mathcal{N}(0, 1)$]. Then, output of the decoder gives us u generated samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_u$. Now, using the posterior distribution $p(\boldsymbol{x}|\tilde{\boldsymbol{x}})$, we find samples that satisfy the constraints. Assume that $\mathcal{S}$ is the set of all indices $1 \leq i \leq u$ in which $p(\boldsymbol{x}_i|\tilde{\boldsymbol{x}})$ satisfies the inequality constraint. Then, for SPVAE, we compute $\min_{i\in S} \hat{\theta}(\boldsymbol{x}_i, a)$ using (12) for all $a \in \mathcal{A}$ and recommend a, which maximizes this expression. For CPVAE, for all $a \in \mathcal{A}$, we pass $|\mathcal{S}|$ samples along with actions a and compute the minimum reward for each action. Then, we recommend the action with the highest reward.

V. Experiments

In this section, we evaluate our suggested methods using three experiments. First, we use the MNIST dataset [51] and frame the usual classification task for identifying handwritten digits in a logged bandit setup. Note that for policy recommendation problems, since counterfactuals are not available,

TABLE I
EXPECTED REWARD OF DIFFERENT STRATEGIES FOR MNIST DATA WITH 50% MISSING ATTRIBUTES AND AVERAGE NUMBER OF INSTANCES OF REWARD LESS THAN -7

	R	Num. of inst. $R < -7$
Imputation	-1.21 ± 0.01	2.8 ± 0.83
MER	-1.18 ± 0.01	2.7 ± 0.47
Cons $c = 0.7$	-1.51 ± 0.02	1.3 ± 1.05
Cons $c = 0.1$	-2.10 ± 0.02	0.2 ± 0.04
Cons $c = 0.001$	-2.58 ± 0.01	0.0 ± 0.00

evaluating an algorithm is not directly possible, this is why here, similar to many other works (see, e.g., [17]), we use a classification problem to evaluate our method. We use this dataset to highlight the differences between the three strategies discussed in this article. Second, we use the Infant Health and Development Program (IHDP) dataset [52], which is a widely used dataset in treatment effect literature, to compare the predictive capability of our methods in the presence of missing value with other recent suggested methods. We show that our methods outperform state-of-the-art methods for estimating average treatment effect (ATE) in the presence of missing values. Finally, to further evaluate our method, we use the OhioT1DM dataset [53], a medical dataset that includes blood glucose measurements and insulin doses for numerous type 1 diabetes mellitus patients using the insulin pump therapy.

A. MNIST

In the first experiment, we use the MNIST dataset. The goal of this experiment is to compare different strategies introduced in Section III. Thus, here, we only implement CPVAE using the three strategies. The complete feature has 784 attributes, and each one is a number between 0 and 255. We will erase a fixed percentage of pixels (50% in this experiment) from each image uniformly at random. The goal is to predict the correct label associated with the image, and hence, the set of actions is $\mathcal{A} = \{0, 1, \ldots, 9\}$. The reward is defined as a Gaussian, centered around the difference of the true label (y_i) and the predicted one (i.e., action A_i) $R_i \sim \mathcal{N}(-|y_i - A_i|, 0.1)$. Note that this is different from the standard binary reward defined for classification task (i.e., $R = 1$ if the predicted label is correct and zero otherwise). The reason we choose this reward is to highlight the differences between the three strategies and the necessary compromises that need to be made in the face of uncertainty. For example, assume that we are considering an image that is 0 with probability 0.7 and 8 with probability 0.3. In this scenario, using the reward, we defined that all three strategies are meaningful (i.e., you may choose 2 to avoid low probability), while with the binary reward, all three strategies coincide (all three recommend $a = 0$). The mechanism for assigning actions to images for creating dataset is as follows. For images representing even numbers, $\pi_0(a|\tilde{\boldsymbol{X}}_i) = 1/20$ for $0 \leq a < 5$ and $\pi_0(a|\tilde{\boldsymbol{X}}_i) = 3/20$ for $5 \leq a < 10$. For odd images, $\pi_0(a|\tilde{\boldsymbol{X}}_i) = 3/20$ for $0 \leq a < 5$ and $\pi_0(a|\tilde{\boldsymbol{X}}_i) = 1/20$ for rest of the actions.

In Table I, the average reward of different strategies is reported. We are using the CPVAE method in this experiment. The maximum expected strategy gets the highest reward as expected, followed by the imputation strategy. It can be seen that, as we decrease parameter c, the expected reward decreases. In exchange, the number of instances for which we get a poor reward (here, we considered reward less than -7) is decreasing with c.

In Fig. 4, we show the distribution of recommended action for three conservative strategies where we change tuning parameter c, from top to bottom $c = 0.001$, $c = 0.1$, and $c = 0.7$. For the first figure with $c = 0.001$, it can be seen that the method always chooses action 5, which is the safest action. This action avoids losses more than 5. Since c is too small, images of different digits can pass the condition on (8), and hence, the best action would be 5 (or 4). It can be seen that, as we increase c, fewer images with random digits pass the constraint and, as a result, the distribution of actions spreads over different actions. The details of the experiment setup and some additional experiments are available in the Supplementary Material.

B. IHDP

In this section, we repeat the experiment in [31] on the IHDP dataset. IHDP is a semisynthetic datasets based on the IHDP compiled by Hill [52]. This experiment studies the effects of specialist home visits on future cognitive test scores. The dataset comprises 25 attributes for each instance and 747 instances in total (139 treated, i.e., $a = 1$, and 698 instances with $a = 0$). Following [31], we report the in-sample mean absolute error in the estimation of ATE. ATE denoted by τ is defined as follows:

$$\tau = \mathbb{E}[R(1) - R(0)] = \mathbb{E}[\mathbb{E}[R(1) - R(0)|\tilde{\boldsymbol{X}}]].$$

Since both values of $R(0)$ and $R(1)$ for all $\boldsymbol{X}$'s are known from the dataset, we can calculate the mean absolute error exactly $\Delta = \left|\hat{\tau} - (1/n)\sum_i R(1)_i - R(0)_i\right|$. We consider scenario "B" of [52], where $R(0) \sim \mathcal{N}(\mu_0, 1)$ and $R(1) \sim \mathcal{N}(\mu_1, 1)$. Here, $(\mu_0, \mu_1) = (\exp(X + A)\beta, X\beta - \omega)$, where ω is chosen such that we have an ATE of $\tau = 4$. The missing values are added with missing completely at random (MCAR) mechanism. We compare three missing rates of 10%, 30%, and 50%. We compare our results with several recent methods in Table II. MI is the multiple imputation approach suggested in [44] and [54] with 20 imputations. MF is the matrix factorization method introduced in [28], and MDC.process and MDC.mi are two methods introduced in [31]. They use a VAE to produce a latent space. Then, for finding τ, they fit an estimator on the latent variable. For all three above methods, the results of an OLS estimator and two different doubly robust estimators are reported in [31, Table 1]. Here, we only report the best of the three results for each of the methods for each setting and refer to [31] for the complete table. MIA.GRF is a doubly robust estimator suggested in [29]. Finally, CEVAE, a method introduced in [6], is another baseline we compare with. This method is not designed to work with missing values, and thus, a mean imputation method was performed to get the complete features before applying this method for estimating ATE. For a more detailed explanation about these competitor methods, we refer to [31, Sec. 4]. In Table II, for

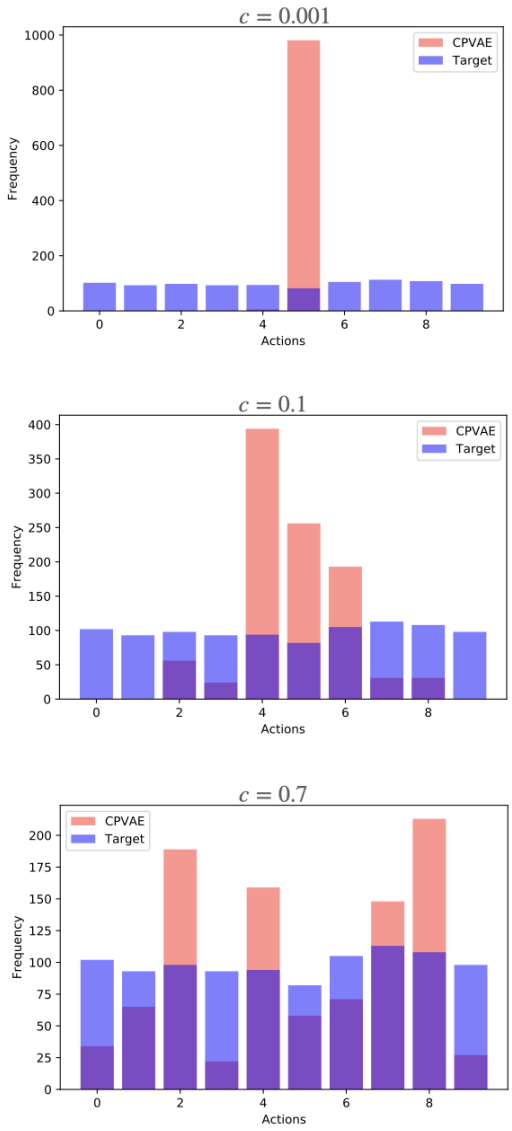

Fig. 4. Distribution of actions recommended by conservative strategy with $c = 0.001$, $c = 0.1$, and $c = 0.7$ from top to bottom.

SPVAE and CPVAE, we use the maximum expected reward strategy with five-time imputations. We can see that both of our methods outperform other methods in 50% missingness and also SPVAE has the best result for 30% (and comparable result in 10%). In the Appendix, we provide a table where we show that our result is not sensitive to the choice of hyperparameters.

C. Type 1 Diabetes OhioT1DM Data

For this experiment, we are using the OhioT1DM dataset [53] that contains continuous glucose monitoring (CGM), insulin dosage, physiological sensor, and self-reported life-event data for six patients with type 1 diabetes for eight weeks. Patients receiving insulin therapy are exposed to the risk of hyperglycemia and hypoglycemia due to underdosing and overdosing. Therefore, it is important that they receive the right dosage of bolus insulin. Note that for evaluating a

TABLE II

MEAN ABSOLUTE ERROR WITH STANDARD ERROR FOR ESTIMATION OF ATE FOR VARIOUS MISSING RATES ON IHDP BENCHMARK DATA

	10%	30%	50%
MI	0.16 ± 0.00	0.30 ± 0.00	0.42 ± 0.01
MIA.grf	0.23 ± 0.01	0.17 ± 0.01	0.19 ± 0.01
MF	0.15 ± 0.01	0.16 ± 0.01	0.20 ± 0.01
CEVAE	0.31 ± 0.01	0.38 ± 0.02	0.38 ± 0.02
MDC.pro	0.15 ± 0.01	0.15 ± 0.02	0.20 ± 0.01
MDC.mi	**0.13 ± 0.02**	0.13 ± 0.01	0.18 ± 0.01
SPVAE	**0.14 ± 0.01**	**0.10 ± 0.01**	**0.11 ± 0.01**
CPVAE	0.18 ± 0.01	0.17 ± 0.01	0.14 ± 0.02

TABLE III

AVERAGE REWARD AND THE PERCENTAGE OF REWARDS LESS THAN -2 OF DIFFERENT METHODS FOR OHIOT1DM DATASET

Method	Average Reward	$R < -2$
LR1	-0.64 ± 0.02	0.112 ± 0.01
LR2	-0.62 ± 0.04	0.101 ± 0.01
RF1	-0.62 ± 0.02	0.110 ± 0.02
RF2	-0.61 ± 0.01	0.108 ± 0.01
GANITE	-0.58 ± 0.09	0.094 ± 0.01
SPVAE-Imp.	-0.58 ± 0.03	0.087 ± 0.01
SPVAE-MER	-0.56 ± 0.03	0.088 ± 0.01
SPVAE-Cons. ($c = 0.4$)	-0.60 ± 0.02	0.081 ± 0.01
CPVAE-Imp.	-0.58 ± 0.03	0.095 ± 0.02
CPVAE-MER	**-0.55 ± 0.02**	0.093 ± 0.01
CPVAE-Cons. ($c = 0.4$)	-0.59 ± 0.02	**0.080 ± 0.01**

recommendation method, we need to have access to counterfactuals that are not available, and hence, it is not possible to directly use the dataset. Here, we follow [55] and first use the dataset to train a simulator using gradient boosting and then use the trained model to produce glucose level for a pair of context (feature) and action (bolus insulin dosage). The corresponding reward will be computed using (16). There are nine attributes for each patient and ten actions uniformly chosen between 0 and 1 (corresponding to normalized insulin dosage). To create missingness, we erase each attribute independently with probability 0.3. We refer to the Appendix for a more detailed explanation of the experiment setting. We compare our method with several baselines, including LR and random forest (RF). In LR1 and RF1, we consider the action as one of the attributes, whereas in LR2 and RF2, we train ten different models corresponding to each of the actions. Many of the more recent competing methods accommodate only two actions and hence cannot be directly used in our setting. Here, we compare with GANITE [34] a GAN-based method that does not have a restriction on the number of actions. For these baseline methods, we first impute missing values using both mean imputation and PVAE (we only report the best performance of the two and use MIs when using PVAE) and then feed the completed feature to the algorithm. As shown in Table III, CPVAE with MER strategy outperforms other methods and the proposed conservative strategy has fewer instances with rewards less than -2

$$R = \begin{cases} \dfrac{x - 80}{10}, & x \leq 90 \text{ (hypoglycemia)} \\ 1, & 90 \leq x \leq 130 \\ \dfrac{180 - x}{50}, & 130 \leq x \text{ (hyperglycemia)} \end{cases} . \quad (16)$$

VI. Conclusion and Future Work

In this work, we proposed using a conservative strategy for dealing with uncertainty due to missingness. We suggested two methods for counterfactual estimation in the presence of missing data using VAEs. Our methods were based on using IPS, which is known to have high variance. One direction for future work is to improve the method we used for estimation of the propensity scores (see, e.g., [29]). We assumed unconfoundedness with missing values, which may not hold in some scenarios. Looking for methods for relaxing this condition is another direction for future work. One direction for future work is to also account for the uncertainty due to the variance of our reward estimation, which could vary for different actions, and this can change our recommendation (we may decide to choose an action with smaller variance). Also, for computing (8), we used random sampling. The minimum in this expression can be approximated using the constrained Bayesian optimization method [56].

Appendix

A. Proposition 1

Proof: First, notice that since $H(a(\boldsymbol{X})|\boldsymbol{X}) = 0$, hence, we have $H(a(\boldsymbol{X}), \boldsymbol{X}, \tilde{\boldsymbol{X}}) = H(\boldsymbol{X}, \tilde{\boldsymbol{X}})$. We also have

$$H(a(\boldsymbol{X}), \boldsymbol{X}, \tilde{\boldsymbol{X}}) = H(\tilde{\boldsymbol{X}}) + H(a(\boldsymbol{X})|\tilde{\boldsymbol{X}}) + H(\boldsymbol{X}|\tilde{\boldsymbol{X}}, a(\boldsymbol{X})).$$

Therefore,

$$H(\tilde{\boldsymbol{X}}, \boldsymbol{X}) = H(\tilde{\boldsymbol{X}}) + H(a(\boldsymbol{X})|\tilde{\boldsymbol{X}}) + H(\boldsymbol{X}|\tilde{\boldsymbol{X}}, a(\boldsymbol{X})).$$

Thus, the following equations hold:

$$\begin{aligned} H(a(\boldsymbol{X})|\tilde{\boldsymbol{X}}) &= H(\tilde{\boldsymbol{X}}, \boldsymbol{X}) - H(\tilde{\boldsymbol{X}}) - H(\boldsymbol{X}|\tilde{\boldsymbol{X}}, a(\boldsymbol{X})) \\ &= H(\boldsymbol{X}|\tilde{\boldsymbol{X}}) - H(\boldsymbol{X}|\tilde{\boldsymbol{X}}, a(\boldsymbol{X})) \\ &= H(\boldsymbol{X}) - I(\boldsymbol{X}; \tilde{\boldsymbol{X}}) - H(\boldsymbol{X}|\tilde{\boldsymbol{X}}, a(\boldsymbol{X})) \\ &= I(\boldsymbol{X}; \tilde{\boldsymbol{X}}, a(\boldsymbol{X})) - I(\boldsymbol{X}; \tilde{\boldsymbol{X}}) \\ &= I(\boldsymbol{X}; a(\boldsymbol{X})) - I(\boldsymbol{X}; \tilde{\boldsymbol{X}}) + I(\boldsymbol{X}; \tilde{\boldsymbol{X}}|a(\boldsymbol{X})) \\ &= H(a(\boldsymbol{X})) - (I(\boldsymbol{X}; \tilde{\boldsymbol{X}}) - I(\boldsymbol{X}; \tilde{\boldsymbol{X}}|a(\boldsymbol{X})). \end{aligned}$$

In the above equations, we used the fact that $H(\tilde{\boldsymbol{X}}, \boldsymbol{X}) = H(\tilde{\boldsymbol{X}}) + H(\boldsymbol{X}|\tilde{\boldsymbol{X}})$ and $H(a(\boldsymbol{X})|\boldsymbol{X}) = 0$. ∎

Example 1: Assume that $\mathcal{X} = \{0, 1\}^4$, and X is uniformly distributed. The channel between $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ is an erasure channel, which erases each bit independently with probability $1/2$. We have $\mathcal{A} = \{a_1, a_2\}$, and the reward is distributed as follows:

$$R|\boldsymbol{x}, a_1 \sim \text{Ber}\left(\frac{x_1 + x_2}{3}\right), \quad R|\boldsymbol{x}, a_2 \sim \text{Ber}\left(\frac{x_3 + x_4}{3} + 0.1\right).$$

Therefore, if $x_1 + x_2 > x_3 + x_4$, $a(\boldsymbol{x}) = a_1$; otherwise, $a(\boldsymbol{x}) = a_2$. For instance, X_i could be the results of four different tests (in which a subset of them will be available) and A is the treatment assigned to the patient. In this setting, we have $H(a(\boldsymbol{X})) = 0.896$ and this is because $x_1 + x_2 > x_3 + x_4$ holds with probability of $5/16$; hence, we have $H(a(\boldsymbol{X})) = h_2(5/16) = 0.896$, where h_2 is the binary entropy function. For computing $I(\boldsymbol{X}; \tilde{\boldsymbol{X}})$, note that since attributes of $\boldsymbol{X}$ are independent and also each attribute will be erased independently, we have

$$I(\boldsymbol{X}; \tilde{\boldsymbol{X}}) = 4(H(\tilde{\boldsymbol{X}}) - H(\tilde{\boldsymbol{X}}|\boldsymbol{X})) = 4(1.5 - 1) = 2.$$

Finally, for computing $I(\boldsymbol{X}; \tilde{\boldsymbol{X}}|a(\boldsymbol{X}))$, note that

$$I(\boldsymbol{X}; \tilde{\boldsymbol{X}}|a(\boldsymbol{X})) = H(\boldsymbol{X}|a(\boldsymbol{X})) - H(\boldsymbol{X}|\tilde{\boldsymbol{X}}, a(\boldsymbol{X})).$$

Now, for the second term, we have

$$\begin{aligned} H(\boldsymbol{X}|a(\boldsymbol{X})) &= \frac{5}{16} H(\boldsymbol{X}|a(\boldsymbol{X}) = a_1) + \frac{11}{16} H(\boldsymbol{X}|a(\boldsymbol{X}) = a_2) \\ &= \frac{5}{16} \log_2\left(\frac{1}{5}\right) + \frac{11}{16} \log_2\left(\frac{1}{11}\right). \end{aligned}$$

The other term can be computed similarly by considering the different cases of $\tilde{\boldsymbol{X}}$ and $a(\boldsymbol{X})$ and using the symmetries for simplifications. Thus, $H(a(\boldsymbol{X})|\tilde{\boldsymbol{X}}) = 0.570$. Note that, the probability that the best algorithm find $a(\boldsymbol{X})$ by observing $\tilde{\boldsymbol{X}}$ is given by

$$\frac{1}{2^{H(a(\boldsymbol{X})|\tilde{\boldsymbol{X}})}}.$$

Thus, in our example, we can hope for guessing $a(\boldsymbol{X})$ correctly on average in 67.3% of cases.

B. Variation of SPVAE

Instead of using propensity score, we can estimate the reward using the following equation, by simply matching the similar actions. For each action $a \in \mathcal{A}$, define $\mathcal{N}_a = \{i : A_i = a\}$ to be the set of all indices with action a

$$\hat{\theta}(\boldsymbol{x}, a) = \sum_{i \in \mathcal{N}_a} w_i' R_i, \quad \text{where} \quad w_i' = \frac{p(\boldsymbol{x}|\tilde{\boldsymbol{X}}_i)}{\sum_{j \in \mathcal{N}_a} p(\boldsymbol{x}|\tilde{\boldsymbol{X}}_j)}. \quad (17)$$

The idea is similar to the original SPVAE, and we estimate the reward with a weighted average of the reward of all instances for which action a was prescribed. The weights measure the similarity between x and $\tilde{\boldsymbol{X}}_i$. Comparing this estimator with (12), we note that the denominator of w_i here is different and also the IPS is missing. Note that we have $\sum_{i=1}^n w_i = 1$ in (12). However, on average, only $\pi_0(a|\tilde{\boldsymbol{X}}_i)$ fraction of samples satisfies $\mathbb{1}[A_i = a]$. One can interpret the propensity score in (12) as a way for compensating this. Basically, we have

$$\mathbb{E}\left[\sum_{i=1}^n w_i \frac{\mathbb{1}[A_i = a]}{\pi_0(a|\tilde{\boldsymbol{X}}_i)}\right] = 1. \quad (18)$$

C. Experiments

In this section, we are outlining details of experiment setting, including hyperparameters and architecture of the neural nets. To implement our experiments, we have used JADE, the U.K. Tier-2 HPC Server specialized for deep learning applications. In particular, all our models are trained with a Nvidia Tesla V100 GPU card, with access to 70-GB memory (though we did not use up the whole memory space). One can also run experiments 1 and 2 on a personal laptop. Also, tensorflow 1.15 is the library used for implementation.

1) MNIST: For our MNIST experiment, we used a reduced MNIST dataset of 5000 data points and performed an 80%–20% train–test split. We used the CPVAE architecture. The encoder followed the PNP-based architecture from [3]. We used the 400-D feature mapping h parameterized by a single fully connected network with ReLU activations and 20-D ID $\mathbf{e_i}$ for each variable. For Gaussian latent variables, we used a 20-D diagonal vector to represent it. The encoder (denoted by function f in this article) is a k-500-200-40, where k is a vector resulted from the concatenation of h and a_{one} is a one-hot encoded action vector. The network f is a fully connected neural network with ReLU activations. The decoder (generator) shares the similar architecture: Z-200-500-D, where Z is a vector resulted from the concatenation of the latent variable and a_{one}, and thus, here, $Z = 30$. Also, D represents the output of the generator model, which should produce pixels and the reward, and hence, $D = 785$. For the conservative strategy, we generated 50 random samples (with the notation of this article $u = 50$). During the training phase, we created artificial missingness to dataset by randomly erasing 50% of the pixels from each image. We used an Adam optimizer with default hyperparameter settings, a learning rate of 0.001, and a batch size (BS) of 8. We trained the network for 20 epochs and repeated the experiment 100 times to get our results.

2) Additional Experiment Results: In Fig. 5, the distribution of the rewards for MNIST experiment for three cases of conservative strategy with $c = 0.001$ and $c = 0.4$ and also imputation strategy is provided. As expected, this illustrates that the imputation strategy has a longer tail in comparison with conservative strategies. Also, $c = 0.001$ has the shortest tail.

3) IHDP: We used both SPVAE and CPVAE on this dataset; for both models, we used 20% of the whole dataset as our training data. Missingness was injected into the dataset by assuming the MCAR mechanism at different levels of missingness, i.e., 10%, 30%, and 50%.

We used 5-D feature mapping h parameterized by a single fully connected network with ReLU activations and 10-D ID $\mathbf{e_i}$ for each variable. The Gaussian latent variable z is set to a 10-D diagonal vector. The inference net is a h-20-20-20 fully connected network with ReLU activations. The generator net is a z-20-20-D (where D is the observed feature dimension of IHDP dataset) fully connected network with ReLU activations. We trained the PVAE using the Adam optimizer with its default hyperparameter settings, a learning rate of 0.001, and a BS of 8. The network was trained for 25 epochs each time and the entire procedure is repeated 100 times for each missingness level.

For CPVAE, we used the same architecture as SPVAE, and the training objective is to reproduce all the attributes with their corresponding rewards given the missing attributes and the action taken. The only difference is that one-hot encoded action was added as the input of encoder and also as an input to the decoder, similar to what we described for MNIST dataset.

4) Hyperparameters: Here, in Table IV, we show that the dependence of our result for IHDP to hyperparameters is insignificant and our method consistently outperforms competitors regardless of choice of hyperparameters. Here, we consider several combinations of latent dimension (LD), BS, and value K in the structure of PVAE, which is the output of the layer that encodes input variables and feeds it to the first encoder.

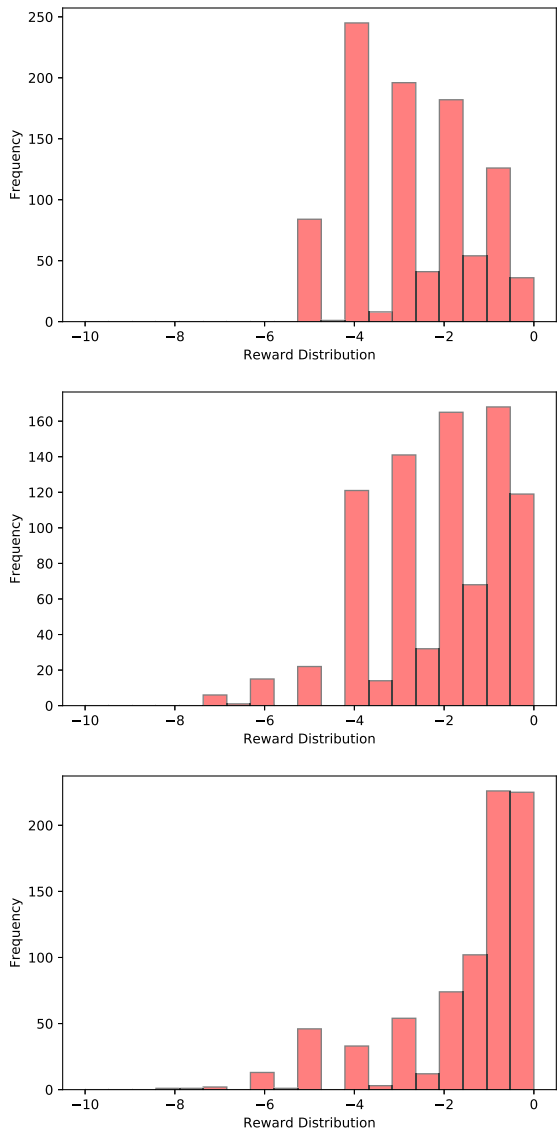

Fig. 5. Distribution of reward for conservative strategies with $c = 0.001$, $c = 0.4$, and imputation strategy (from top to bottom).

5) OhioT1DM: The OhioT1DM dataset contains eight weeks of information about six individuals with Diabetes 1 in a time series format. Since, in the original dataset, only the response to the actual dose that was administered exists, it is not possible to evaluate recommendation methods directly using dataset. Thus, we use a simulator suggested in [55] that is trained on the actual data to estimate the response to a bolus injection. The simulator maps a pair of context and action to the mean of CGM. From CGM, the reward can be calculated according to (16). A gradient boosting regression model with the Huber loss is used to achieve this. In the model,

TABLE IV
ESTIMATED ATE FOR DIFFERENT HYPERPARAMETER SETTINGS WITH $p_{\text{MISS}} = 0.5$. WE CHANGE LATENT VARIABLE DIM., BS, AND VARIABLE K IN THE PVAE ENCODER

Hyperparameters	ATE
LD8,BS8,K20	[0.1135207]
LD10,BS8,K5	[0.11411935]
LD10,BS8,K10	[0.11438434]
LD5,BS8,K10	[0.1149257]
LD5,BS8,K5	[0.11494357]
LD5,BS4,K20	[0.11637331]
LD10,BS2,K10	[0.1164837]
LD10,BS8,K20	[0.11746769]
LD5,BS2,K20	[0.11749744]
LD10,BS2,K20	[0.11766106]
LD8,BS8,K5	[0.11910977]
LD8,BS4,K20	[0.11929849]
LD8,BS4,K10	[0.12004589]
LD5,BS4,K10	[0.12054679]
LD5,BS2,K5	[0.12068875]

100 trees of maximum depth of 5 are used as weak learner. Furthermore, a multivariate Gaussian distribution is fit to approximate patients' features. For producing dataset, we first sample from this distribution to get the features, and then, we choose an action for this feature and then feed the pair of action and feature to the simulator to produce the output. Then, we randomly remove 30% of features and store the triple of feature (with missing values), action, and the simulated reward in the dataset. We have produced 5000 samples for training. The way we produce actions is to train a simple LR model to learn the action that is prescribed in the original dataset and use this model to produce actions. In order to guarantee the second condition of the assumptions in Section II (i.e., $\pi(a|\tilde{\boldsymbol{x}}) > 0$), we choose the action half of times from the action generator (LR model), and for the other half, we randomly choose one of the ten actions. In the test time, we sample from the Gaussian distribution to get the features, then randomly remove 30% of the features, and feed it to our method to get the recommended action. Then, we feed the complete feature and action to the simulator to get the reward. We refer to [55] for more details about the data generation mechanism. We chose the following hyperparameters for the model: $K = 8$, $e_i = 5$, LD is 5, and BS is 8. A two-layer encoder and a two-layer decoder are used, with 10-10-5 and 5-10-10 nodes, respectively.

REFERENCES

[1] R. J. Little *et al.*, "The prevention and treatment of missing data in clinical trials," *New England J. Med.*, vol. 367, no. 14, pp. 1355–1360, 2012.

[2] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*, vol. 81. Hoboken, NJ, USA: Wiley, 2004.

[3] C. Ma, S. Tschiatschek, K. Palla, J. M. Hernández-Lobato, S. Nowozin, and C. Zhang, "EDDI: Efficient dynamic discovery of high-value information with partial VAE," 2018, *arXiv:1809.11142*.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–14.

[5] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," 2014, *arXiv:1401.4082*.

[6] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal effect inference with deep latent-variable models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6446–6456.

[7] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," 2015, *arXiv:1502.04623*.

[8] D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised learning of 3D structure from images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4996–5004.

[9] E. L. Ionides, "Truncated importance sampling," *J. Comput. Graph. Statist.*, vol. 17, no. 2, pp. 295–311, Jun. 2008.

[10] A. Swaminathan and T. Joachims, "Batch learning from logged bandit feedback through counterfactual risk minimization," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1731–1755, 2015.

[11] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[12] D. B. Rubin, "Estimating causal effects of treatments in randomized and nonrandomized studies," *J. Educ. Psychol.*, vol. 66, no. 5, p. 688, 1974.

[13] D. B. Rubin, "Causal inference using potential outcomes: Design, modeling, decisions," *J. Amer. Stat. Assoc.*, vol. 100, no. 469, pp. 322–331, 2005.

[14] P. R. Rosenbaum and D. B. Rubin, "Reducing bias in observational studies using subclassification on the propensity score," *J. Amer. Stat. Assoc.*, vol. 79, no. 387, pp. 516–524, Sep. 1984.

[15] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," *J. Amer. Stat. Assoc.*, vol. 47, no. 260, pp. 663–685, 1952.

[16] P. R. Rosenbaum and D. B. Rubin, "The central role of the propensity score in observational studies for causal effects," *Biometrika*, vol. 70, no. 1, pp. 41–55, 1983.

[17] O. Atan, W. R. Zame, Q. Feng, and M. van der Schaar, "Constructing effective personalized policies using counterfactual inference from biased data sets with many features," *Mach. Learn.*, vol. 108, no. 6, pp. 945–970, Jun. 2019.

[18] T. Joachims, A. Swaminathan, and M. de Rijke, "Deep learning with logged bandit feedback," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–12.

[19] R. Prentice, "Use of the logistic model in retrospective studies," *Biometrics*, vol. 32, pp. 599–606, 1976.

[20] J. M. Robins, A. Rotnitzky, and L. P. Zhao, "Estimation of regression coefficients when some regressors are not always observed," *J. Amer. Stat. Assoc.*, vol. 89, no. 427, pp. 846–866, Sep. 1994.

[21] M. Dudík, J. Langford, and L. Li, "Doubly robust policy evaluation and learning," 2011, *arXiv:1103.4601*.

[22] W. Hoiles and M. van der Schaar, "Bounded off-policy evaluation with missing data for course recommendation and curriculum design," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1596–1604.

[23] L. Li, R. Munos, and C. Szepesvári, "Toward minimax off-policy value estimation," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, 2015, pp. 608–616.

[24] J. Yoon, C. Davtyan, and M. van der Schaar, "Discovery and clinical decision support for personalized healthcare," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 4, pp. 1133–1145, Jul. 2017.

[25] R. J. Little and D. B. Rubin, *Statistical Analysis With Missing Data*, vol. 793. Hoboken, NJ, USA: Wileys, 1987.

[26] R. B. D'Agostino, Jr., and D. B. Rubin, "Estimating and using propensity scores with partially missing data," *J. Amer. Stat. Assoc.*, vol. 95, no. 451, pp. 749–759, Sep. 2000.

[27] J. Hill, "Reducing bias in treatment effect estimation in observational studies suffering from missing data," Columbia Univ. Inst. Social Econ. Res. Policy, Work. Paper 04-01, 2004.

[28] N. Kallus, X. Mao, and M. Udell, "Causal inference with noisy and missing covariates via matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6921–6932.

[29] I. Mayer, E. Sverdrup, T. Gauss, J.-D. Moyer, S. Wager, and J. Josse, "Doubly robust treatment effect estimation with missing attributes," *Ann. Appl. Statist.*, vol. 14, no. 3, pp. 1409–1431, Sep. 2020.

[30] S. Parbhoo, M. Wieser, A. Wieczorek, and V. Roth, "Information bottleneck for estimating treatment effects with systematically missing covariates," *Entropy*, vol. 22, no. 4, p. 389, Mar. 2020.

[31] I. Mayer, J. Josse, F. Raimundo, and J.-P. Vert, "MissDeepCausal: Causal inference from incomplete data using deep latent variable models," 2020, *arXiv:2002.10837*.

[32] U. Shalit, F. D. Johansson, and D. Sontag, "Estimating individual treatment effect: Generalization bounds and algorithms," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 3076–3085.

[33] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests," *J. Amer. Stat. Assoc.*, vol. 113, no. 523, pp. 1228–1242, Jul. 2018.

[34] J. Yoon, J. Jordon, and M. van der Schaar, "GANITE: Estimation of individualized treatment effects using generative adversarial nets," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–22.
[35] O. Atan, C. Tekin, and M. van der Schaar, "Global bandits," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 5798–5811, Dec. 2018.
[36] Y.-X. Wang, A. Agarwal, and M. Dudik, "Optimal and adaptive off-policy evaluation in contextual bandits," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3589–3597.
[37] A. Swaminathan *et al.*, "Off-policy evaluation for slate recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3632–3642.
[38] Y. Wu, R. Shariff, T. Lattimore, and C. Szepesvári, "Conservative bandits," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1254–1262.
[39] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, 2015.
[40] K. Srinivasan, B. Eysenbach, S. Ha, J. Tan, and C. Finn, "Learning to be safe: Deep RL with a safety critic," 2020, *arXiv:2010.14603*.
[41] P.-A. Mattei and J. Frellsen, "MIWAE: Deep generative modelling and imputation of incomplete data sets," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4413–4423.
[42] C. Ma, W. Gong, J. M. Hernández-Lobato, N. Koenigstein, S. Nowozin, and C. Zhang, "Partial VAE for hybrid recommender system," in *Proc. NIPS Workshop Bayesian Deep Learn.*, 2018, pp. 1–7.
[43] W. Gong, S. Tschiatschek, S. Nowozin, R. E. Turner, J. M. Hernández-Lobato, and C. Zhang, "Icebreaker: Element-wise efficient information acquisition with a Bayesian deep latent Gaussian model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14791–14802.
[44] A. Mattei, "Estimating and using propensity score in presence of missing background data: An application to assess the impact of childbearing on wellbeing," *Stat. Methods Appl.*, vol. 18, no. 2, pp. 257–273, Jul. 2009.
[45] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.
[46] A. Nazabal, P. M. Olmos, Z. Ghahramani, and I. Valera, "Handling incomplete heterogeneous data using VAEs," 2018, *arXiv:1807.03653*.
[47] S. V. Buuren and K. Groothuis-Oudshoorn, "MICE: Multivariate imputation by chained equations in R," *J. Stat. Softw.*, vol. 45, no. 3, pp. 1–68, 2011.
[48] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing data imputation using generative adversarial nets," 2018, *arXiv:1806.02920*.
[49] R. Mitra and J. P. Reiter, "A comparison of two methods of estimating propensity scores after multiple imputation," *Stat. Methods Med. Res.*, vol. 25, no. 1, pp. 188–204, Feb. 2016.
[50] K. G. M. Moons, R. A. R. T. Donders, T. Stijnen, and F. E. Harrell, "Using the outcome for imputation of missing predictor values was preferred," *J. Clin. Epidemiol.*, vol. 59, no. 10, pp. 1092–1101, Oct. 2006.
[51] Y. LeCun. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[52] J. L. Hill, "Bayesian nonparametric modeling for causal inference," *J. Comput. Graph. Statist.*, vol. 20, no. 1, pp. 217–240, Jan. 2011.
[53] C. Marling and R. C. Bunescu, "The OhioT1DM dataset for blood glucose level prediction," in *Proc. KHD@ IJCAI*, 2018, pp. 1–4.
[54] S. Seaman and I. White, "Inverse probability weighting with missing predictors of treatment assignment or missingness," *Commun. Statist.-Theory Methods*, vol. 43, no. 16, pp. 3499–3515, Aug. 2014.
[55] E. Turgay, C. Bulucu, and C. Tekin, "Exploiting relevance for online decision-making in high-dimensions," *IEEE Trans. Signal Process.*, vol. 69, pp. 1438–1451, 2021.
[56] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints," in *Proc. ICML*, 2014, pp. 937–945.