Learning Generalized Gumbel-max Causal Mechanisms

Anonymous Author(s)
Affiliation
Address
email

Abstract

To perform counterfactual reasoning in Structural Causal Models (SCMs), one needs to know the causal mechanisms, which provide factorizations of conditional distributions into noise sources and deterministic functions mapping realizations of noise to samples. Unfortunately, the causal mechanism is not uniquely identified by data that can be gathered by observing and interacting with the world, so there remains the question of how to choose causal mechanisms. In recent work, Oberst & Sontag (2019) propose Gumbel-max SCMs, which use Gumbel-max reparameterizations as the causal mechanism due to an appealing counterfactual stability property. However, the justification requires appealing to intuition. In this work, we instead argue for choosing a causal mechanism that is best under a quantitative criteria such as minimizing variance when estimating counterfactual treatment effects. To achieve this, we propose a parameterized family of causal mechanisms that generalize Gumbel-max. We show that they can be trained to minimize counterfactual effect variance and other losses on a distribution of queries of interest, yielding lower variance estimates of counterfactual treatment effect than fixed alternatives, also generalizing to queries not seen at training time.

1 Introduction

Pearl [2009] presents a "ladder of causation" that distinguishes three levels of causal concepts: associational (level 1), interventional (level 2), and counterfactual (level 3). Level 1 corresponds to information learnable from passive observation, level 2 corresponds to active intervention, and level 3 corresponds to reasoning about hypothetical interventions given that some other outcome actually occurred (e.g., given that my headache went away after taking an aspirin, what would have happened if I did not take an aspirin?) The three levels are distinct in the sense that it is generally not possible to uniquely determine higher level models from lower level information [Bareinboim et al., 2020]. This means that we cannot uniquely determine a Structural Causal Model (SCM), which allows counterfactual reasoning, even if we can actively intervene in a system (e.g., perform randomized trials of whether to give an aspirin or not).

Nonetheless, we desire to reason about counterfactuals. First, counterfactual reasoning is fundamental to human cognition, e.g., in assigning credit or blame, and as a mechanism for assessing potential alternative past behaviors in order to update policies governing future behavior. Second, counterfactual reasoning is computationally useful, for example allowing us to shift measure from observed policy to an alternative policy in an off-policy manner [Buesing et al., 2018, Oberst and Sontag, 2019].

One approach to dealing with the non-identifiability of level 3 from level 2 is axiomatic, as exemplified by Oberst and Sontag [2019]. There, an intuitive requirement of counterfactuals is presented, and then a causal mechanism is chosen that provably satisfies the requirement. The resulting proposal is *Gumbel-max SCMs* which assume causal mechanisms are governed by the Gumbel-max trick.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In this work, we instead start from the computational perspective and ask what causal mechanism (that is consistent with level 2 observations) yields the most desirable computational behavior? For example, what mechanism leads to the lowest variance estimates of treatment effects in a counterfactual setting? Casting the problem this way gives additional flexibility to choose a causal mechanism that is *specifically-tuned* to a distribution of observations and interventions of interest, and in terms of a loss function that measures the quality of a counterfactual sample.

To motivate the need for specifically-tuned causal mechanisms, we prove limitations of non-tuned mechanisms (including Gumbel-max) and the power of tuned mechanisms by drawing on connections to literature on couplings, optimal transport, and common random numbers. We introduce a continuously parameterized family of causal mechanisms where all members of the family are identical when used in a level 2 context but different when used in a level 3 context. The families contain Gumbel-max, but a wide variety of other mechanisms can be learned by using gradient-based optimization over the family of mechanisms. Empirically we show that the mechanisms can be learned using a variant of Gumbel-softmax relaxation [Maddison et al., 2017, Jang et al., 2017], and that the resulting mechanisms improve over Gumbel-max and other fixed mechanisms. Further, we show that the learned mechanisms generalize, in the sense that we can learn a causal mechanism from a *training set* of observed outcomes and counterfactual queries and have it generalize to a *test set* of observed outcomes and counterfactual queries that were not seen at training time.

2 Background

A coupling between categorical distributions $p(x)$ and $q(y)$ is a joint distribution $\pi(x, y)$ such that

$$\sum_x \pi(x, y) = q(y) \text{ and } \sum_y \pi(x, y) = p(x). \tag{1}$$

The Gumbel-max trick is a method for drawing a sample from a categorical distribution defined by logits $l \in \mathbb{R}^K$. It is based on the fact that, if $\gamma_k \sim \text{Gumbel}(0)$[1] for $k \in \{1, \ldots, K\}$, then

$$\mathbb{P}_\gamma\Big[x = \text{argmax}_k \left[l_k + \gamma_k\right]\Big] = \frac{\exp(l_x)}{\sum_k \exp(l_k)}. \tag{2}$$

Gumbel-max couplings. A Gumbel-max coupling is a joint distribution that couples two categorical distributions with logits $l^{(1)} \in \mathbb{R}^K$ and $l^{(2)} \in \mathbb{R}^K$ using a common realization of Gumbel random variables γ_k [Oberst and Sontag, 2019, Lorberbom et al., 2020]. Formally, it takes the form

$$\pi_{gm}(x, y) = \mathbb{P}_\gamma\Big[x = \text{argmax}_k \left[l_k^{(1)} + \gamma_k\right] \text{ and } y = \text{argmax}_k \left[l_k^{(2)} + \gamma_k\right]\Big]. \tag{3}$$

The Wasserstein distance $W_1(p, q; d)$ between two distributions p and q under a given metric d is

$$W_1(p, q; d) = \inf_{\pi \in C(p,q)} \mathbb{E}_{x,y\sim\pi} d(x, y), \tag{4}$$

where $C(p, q)$ is the set of all couplings between p and q. When $d(x, y) = 1_{x\neq y}$, the optimal coupling is known as a *maximal coupling*. A core problem of coupling theory is find a coupling $\pi \in C(p, q)$ in order to estimate $\mathbb{E}_{x\sim p}[h_1(x)] - \mathbb{E}_{y\sim q}[h_2(y)] = \mathbb{E}_{x,y\sim\pi}[h_1(x) - h_2(y)]$ for some real-valued cost functions h_1, h_2, with minimal variance $\text{Var}_{x,y\sim\pi}[h_1(x) - h_2(y)]$. Interesting, whenever $h_1(x), h_2(y)$ are monotone, such a coupling can be attained by $(F_X^{-1}(U), F_Y^{-1}(U)))$, when $F_X(t), F_Y(t)$ are the cumulative distribution functions (CDFs) of X, Y respectively. When h_1, h_2 are not monotone, there are clearly cases where CDF inversion produces suboptimal couplings.

Sampling from the Gumbel-max coupling has interesting properties due to the Gumbel-max trick introducing statistical independence between the max-value and the argmax, cf. Maddison et al. [2014]. Oberst and Sontag [2019] exploit this to sample from the posterior over Gumbel noise

$$\mathbb{E}_{x,y\sim\pi}[h_1(x) - h_2(y)] = \sum_y q(y)\mathbb{E}_{\gamma\sim(G_2|y)}\Big[h_1(\text{argmax}_k[l_k^{(1)} - l_k^{(2)} + \gamma_k]) - h_2(\text{argmax}_k\left[\gamma_k\right])\Big].$$

Here $G_2|y$ is the Gumbel distribution with location $\log l_k^{(2)}$ conditioned on the event that y is the maximal argument. For completeness, the proof appears in the Appendix.

[1] Samples of Gumbel(0) random variables can be generated as $-\log(-\log(u))$, where $u \sim \text{Uniform}([0, 1])$

Structural casual models (SCMs). Causality introduces a framework to ask counterfactual questions. SCMs divide variables into *exogenous* background (or *noise*) variables and *endogenous* variables that are modeled explicitly. Each endogenous variable v_i has a set of parent endogenous variables pa_i, an associated exogenous variable u_i, and a function f_i. Putting these together along with a prior distribution over u_i's defines the *causal mechanism* governing v_i, as $v_i = f_i(pa_i, u_i)$. An SCM for a Markov Decision Process (MDP) consists of uniform random variables $u_t^{(env)}, u_t^{(act)}$ for each timestep t, and a set of transition functions $f_t(a_t, s_t, u_t^{(env)})$ that map states, actions, and noise to next states, and policy functions $h_t(s_t, u_t^{(act)})$ that map states and noise to actions. In our work we explore the recent Gumbel-max SCMs, where the functions f_t are the argmax-functions and the randomness follows the Gumbel probability rule.

The Gumbel-max SCM enjoys a simple intervention mechanism: supposing we intervene by replacing a_t with $\tilde{a}_t$, the intervention amounts to replacing $\mathrm{argmax}_{s_{t+1}}\{p(s_{t+1}|a_t, s_t) + \gamma_t(s_{t+1})\}$ with $\mathrm{argmax}_{s_{t+1}}\{p(s_{t+1}|\tilde{a}_t, s_t) + \gamma_t(s_{t+1})\}$. Analyzing this procedure leads to a key desirable feature of Gumbel-max SCMs, which Oberst and Sontag [2019] refer to as *counterfactual stability*. The property says that if we observe outcome $s_{t+1} = i$ and are considering a counterfactual scenario where the probability of outcome i is increased by a factor of α, then the counterfactual outcome cannot be equal to j unless the probability of j has increased by a factor of more than α. This is desirable because of the intuitive idea that the outcome should only change in a counterfactual scenario if there is a reason for it to change (i.e., some other outcome's relative probability increased more than the observed outcome's).

3 Problem statement: Learning Implicit Couplings

In this section, we formalize the task of selecting causal mechanisms according to some quantitative metric. This is motivated by the observation that, if we only care about our models being consistent with interventional data, then we have flexibility in the choice of causal mechanism.

Let $\boldsymbol{u} \in \mathbb{R}^D$ be a sample from some noise distribution (e.g., we might have an independent Uniform([0, 1]) or Gumbel(0) distribution per dimension) and let $l \in \mathbb{R}^K$ be (conditional) logits vector defining a distribution over K categorical outcomes. We wish to learn a function $f_\theta : \mathbb{R}^D \times \mathbb{R}^K \to \{1, \dots, K\}$ that maps noise and logits to a sampled outcome, and we require that the process produces samples from the distribution defined by the logits $p(k) \propto \exp l_k$ when integrating over $\boldsymbol{u}$ (i.e., we want a reparameterization trick). Define an *implicit coupling* to be a coupling induced by running f_θ with the same noise and two different logit vectors. We are interested learning θ so that the implicit coupling defined in this way has desirable properties.

To control the tuning of f_θ, we have two degrees of freedom. First, we have a loss function $g_{l^{(1)},l^{(2)}} : \{1, \dots, K\} \times \{1, \dots, K\} \to \mathbb{R}$ that assigns a real-valued loss to a joint outcome $(f_\theta(\boldsymbol{u}, l^{(1)}), f_\theta(\boldsymbol{u}, l^{(2)}))$, perhaps modulated by $l^{(1)}$ and $l^{(2)}$. The loss function is used to specify how desirable a pair of observed and counterfactual outcomes are (e.g., the counterfactual stability criterion disallows certain joint outcomes depending the relationship of $l^{(1)}$ to $l^{(2)}$). Second we have a distribution $\mathcal{D}$ over pairs of logits $(l^{(1)}, l^{(2)})$. This determines the distribution of observed outcomes and counterfactual queries of interest. We can think of $l^{(1)}$ as the logits governing an observed outcome and $l^{(2)}$ as their modification under a hypothetical intervention.

Finally, we have the main objective:

$$\min._\theta \ \mathbb{E}_{(l^1,l^2)\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{u}}\left[g_{l^{(1)},l^{(2)}}(f_\theta(\boldsymbol{u}, l^{(1)}), f_\theta(\boldsymbol{u}, l^{(2)}))\right] \tag{5}$$

$$\text{subject to } \ \mathbb{P}_{\boldsymbol{u}}[f_\theta(\boldsymbol{u}, l) = k] = \frac{\exp l_k}{\sum_{k'} \exp l_{k'}} \qquad \text{for all } l \in \mathbb{R}^K. \tag{6}$$

Relationship to 1-Wasserstein Metric. One may ask how the objective relates to the 1-Wasserstein metric. We could identify $g_{l^{(1)},l^{(2)}}$ with the Wasserstein distance metric and view the joint distribution of $f_\theta(\boldsymbol{u}, l^{(1)}), f_\theta(\boldsymbol{u}, l^{(2)})$ as a coupling between $p \propto \exp l^{(1)}$ and $q \propto \exp l^{(2)}$, which reveals similarities. However, a key difference is that f_θ can be used to generate samples from one side of the coupling (say p) without knowledge of what q will be chosen. Thus, f_θ can be seen as *coupling p to all q simultaneously*. In contrast, the Wasserstein optimization requires knowledge of

both p and q and then computes a coupling specific to that pair. We prove in the next section that optimizing over θ in this way cannot generally achieve the same result as separately computing the Wasserstein for all pairs of distributions, even if f_θ can realize all mappings from noise to samples.

4 Properties of Implicit Couplings

Our development so far raises questions about the relationship of the proposed approach to the Gumbel-max couplings underlying Gumbel-max SCMs and about the relationship of our objective to Wasserstein metrics. In this section we establish the relationships and differences. The main results are that despite being counterfactually stable, Gumbel-max couplings are not even maximal couplings. We then go further and show that any *implicit coupling* is limited in expressivity. This establishes the difference to Wasserstein metrics and optimal transport, which are framed in terms of minimizing over the larger space of all couplings.

4.1 Suboptimality of Gumbel-max Couplings

We know that Gumbel-max couplings are counterfactually stable, but are they maximal couplings? Here we show that the answer is no.

Proposition 1. *The probability that $x = y = i$ in a Gumbel-max coupling is* $\frac{1}{1+\sum_{j\neq i}\max(\frac{p(j)}{p(i)},\frac{q(j)}{q(i)})}$.

The full proof is in the Appendix. The main idea is to express the event $x = y = i$ as a conjunction of inequalities defining the argmax, then simplifying using properties of Gumbel distributions.

Corollary 1. *Gumbel-max couplings are not maximal couplings. In particular, they are suboptimal iff there is an i such that*

$$\max\left(\sum_{j\neq i}\frac{p(j)}{p(i)},\sum_{j\neq i}\frac{q(j)}{q(i)}\right) < \sum_{j\neq i}\max\left(\frac{p(j)}{p(i)},\frac{q(j)}{q(i)}\right). \tag{7}$$

The proof appears in the Appendix. It follows from Prop. 1 and the fact that the probability of $x = y = i$ in a maximal coupling is $\min(p(i), q(i))$.

On the positive side, it is straightforward to show that Gumbel-max couplings are optimal when $p = q$ and when there are only two possible outcomes. We also show that Gumbel-max couplings are within a constant-factor of optimal as maximal couplings.

Corollary 2. *If the Gumbel-max coupling assigns probability α to the event that $x = y$, then the probability that $x = y$ under the maximal coupling is at most 2α.*

The proof appears in the Appendix. It comes from bounding the ratio of the LHS to the RHS in Eq. 7.

4.2 Impossibility of Implicit Maximal Couplings

Since Gumbel-max does not always induce the maximal coupling, we might wonder if some other implicit coupling mechanism could. Here we show that it is impossible. In particular, we show that no fixed implicit coupling is maximal for every pair of distributions over the set $\{1, 2, 3\}$. Thus, there will always be some pair of distributions for which an implicit coupling is non-maximal. A proof of the Proposition appears in the Appendix.

Proposition 2. *There is no reparameterization algorithm f_θ that takes logits l and a sample $\boldsymbol{u}$ of noise, and deterministically transforms $\boldsymbol{u}$ into a sample from $\exp l$, such that when given any two distributions p and q and using shared noise u, the joint distribution of samples is always a maximal coupling of p and q.*

5 Methods for Learning Implicit Couplings

Here we develop methods for learning implicit couplings, the problem defined in Sec. 3. We use the term *gadget* to refer to a learnable, continuously parameterized family of f_θ. We present two gadgets in this section. Gadget 1 does not fully satisfy the requirements laid out in Sec. 3, but it is simpler and introduces some key ideas, so we present it as a warm-up. Gadget 2 fully satisfies the requirements.

5.1 Gadget 1: Predicting an auxiliary joint

The main idea in Gadget 1 is to learn a mapping $\pi_\theta : \mathbb{R}^K \to \mathbb{R}^{K\times K}$ from categorical distribution $p \in \mathbb{R}^K$ to an auxiliary joint distribution $\pi_\theta(p) \in \mathbb{R}^{K\times K}$. For notational brevity, we use the shorthand $\pi_{x,z}(p)$ to refer to $[\pi_\theta(p)]_{x,z}$. The architecture of the mapping is constrained so that marginalizing out the auxiliary variable always yields a distribution consistent with the given logits, i.e., $\sum_z \pi_{x,z}(p) = p(x)$. We then generate K^2 independent $\gamma_{x,z} \sim \mathrm{Gumbel}(0)$ samples and perform Gumbel-max on the auxiliary joint. We only care about the sample from the non-auxiliary dimension, so one way of doing this is to first maximize out the auxiliary dimension to get $\gamma(p) = \max_z\{\gamma_{x,z} + \log \pi_{x,z}(p)\}$ and then return $\hat{x} = \mathrm{argmax}\{\gamma(p)\}$ as the sample. Here $\hat{x}$ clearly is distributed according to p because this is performing Gumbel-max on a joint distribution with marginals equal to p and then marginalizing out the auxiliary variable.

To create a coupling, we run this process separately for p and q but with shared realizations of the K^2 Gumbels. However, the place where Gadget 1 does not fully satisfy the requirements from Sec. 3 is that we transpose the Gumbels for one of the samples. That is, we sample a coupling as

$$[\gamma_1(p)]_x = \max_z\{\gamma_{x,z} + \log \pi_{x,z}(p)\} \qquad [\gamma_2(q)]_y = \max_z\{\gamma_{z,y} + \log \pi_{y,z}(q)\} \tag{8}$$

$$\hat{x} = \mathrm{argmax}\{\gamma_1(p)\} \qquad \hat{y} = \mathrm{argmax}\{\gamma_2(q)\}. \tag{9}$$

Gadget 1 can still be used to create a coupling, but it is more analogous to antithetical sampling, where the noise source is used differently based on which sample is being drawn. Note that like in antithetical sampling, both processes draw samples from the correct distribution, since transposing a matrix of independent Gumbels does not change the distribution.

Sampling from the posterior. To use the gadget in a counterfactual setting, we need to condition on an outcome $x^{(obs)}$ and then draw a counterfactual sample of y. As in the counterfactual sampling algorithm for Gumbel-max couplings [Oberst and Sontag, 2019], we rely on top-down Gumbel sampling [Maddison et al., 2014], but here we require a slightly more elaborate two-step construction.

The first step is to observe that we can integrate out the auxiliary dimension to get marginals of $\gamma_1(p)$: $[\gamma_1(p)]_x \sim \mathrm{Gumbel}(\log \sum_z \pi_{x,z}(p))$. Since $x^{(obs)}$ was the observed outcome, it is the argmax of these Gumbels. The top-down algorithm can be used to sample $[\gamma(p)]_{x^{(obs)}}$ under the constraint that $x^{(obs)}$ is the argmax, and then the remaining $\gamma(p)$ values can be sampled conditional on the argmax as truncated Gumbels, i.e., $[\gamma_1(p)]_x \sim \mathrm{TruncGumbel}(\log \sum_z \pi_{x,z}(p), [\gamma(p)]_{x^{(obs)}})$.

The second step is to condition on $[\gamma(p)]_x$ for each x and draw $\gamma_{x,z}$ for all z. At this point we know the maximum values for each x but not which joint configuration achieved the argmax. Here we leverage the independence of the max and argmax and begin by drawing the argmax $z^*_x = \mathrm{argmax}_z\, \gamma_{x,z} + \log \pi_{x,z}$. By the Gumbel-max trick, this can just be sampled from a softmax over $\log \pi_{x,:}(p)$. Now, we have the max $[\gamma(p)]_x$ and argmax z^*_x for each x. This determines values of γ_{x,z^*_x}. For all other $z \neq z^*_x$, we can use a second stage of top-down sampling separately for each x to get the needed samples, i.e., $\gamma_{x,z} \sim \mathrm{TruncGumbel}(\log \pi_{x,z}(p), [\gamma_1(p)]_x)$. The result is a posterior sample of the full $K \times K$ matrix of Gumbels that are consistent with the behavior policy. To get a counterfactual sample, transpose the $\gamma_{x,z}$'s and run the q gadget forward to sample y.

5.2 Gadget 2

Gadget 1 is not an implicit coupling as defined in Sec. 3, because it requires Gumbels to be transposed when sampling p versus q. In this section, we present a gadget that is a proper implicit coupling.

Gadget 2 again invokes an auxiliary variable and parameterizes a learned joint distribution, but the auxiliary variable z is no longer required to share the same sample space as the variable being sampled. Additionally, instead of performing Gumbel-max on the learned joint directly, we start by drawing a single z independently of p. The gadget is defined as follows:

$$\gamma_i^{(z)} \sim \mathrm{Gumbel}(0) \;\text{ for } i = 1, \ldots, |Z| \qquad \gamma_i^{(x)} \sim \mathrm{Gumbel}(0) \;\text{ for } i = 1, \ldots, |X| \tag{10}$$

$$\hat{z} = \mathrm{argmax}_z(\log \pi(z) + \gamma^{(z)}) \qquad \hat{x} = \mathrm{argmax}_x(\log \pi_\theta(x \mid \hat{z}, p)) + \gamma^{(x)}). \tag{11}$$

To sample a coupling, we re-use all the γ's and run the same process with q instead of p. This means that we additionally get $\hat{y} = \mathrm{argmax}_y(\log \pi_\theta(y \mid \hat{z}, q)) + \gamma^{(x)})$.

Intuitively, we can think of $\hat{z}$ as a latent cluster identity and each cluster being associated with a different learned mapping $\pi_\theta(x \mid \hat{z}, p)$. Since the sampled cluster will be shared between p and q, the learning can choose how to assign clusters so that a Gumbel-max coupling of $\pi_\theta(x \mid \hat{z}, p)$ and $\pi_\theta(y \mid \hat{z}, q)$ produces joint outcomes that are favorable under g.

Architecture for $\pi_\theta(x|z,p)$. Not all choices of $\pi_\theta(x|z,p)$ lead to correct samples. For correctness, we need to enforce the analogous constraint as is Gadget 1, which is that when we integrate out the auxiliary variable, we get samples from the p distribution provided as an input; i.e., $\sum_z \pi(z)\pi_\theta(x|z,p) = p(x)$ for all p. Here we describe how to build an architecture for $\pi_\theta(x|z,p)$ that is guaranteed to satisfy the constraint.

First, we use a neural function approximator $h_\theta : \mathbb{R}^K \to \mathbb{R}_+^{Z \times K}$ that maps logits $l = \log p$ to a nonnegative matrix $A_0 = h_\theta(l)$ of probabilities for each pair (z, x). Next, we iteratively normalize the columns to have marginals $p(x)$ and the rows of A to have marginals $p_\theta(z)$ for T steps (a modified version of the algorithm proposed by Sinkhorn and Knopp [1967]). The last iterate $A = A_T$ always satisfies $\sum_x A_{x,z} = \pi(z)$ but may only approximately satisfy the constraint that $\sum_z \pi(z)\pi_\theta(x|z,p) = p(x)$. To deal with this, we treat A as a proposal and apply a final accept-reject correction, falling back to an independent draw from p if the z-dependent proposal is rejected. The marginals of this process give our expression for $\pi_\theta(x|z,p)$:

$$c_x = \frac{\exp l_x}{\sum_z A_{x,z}}, \qquad c_* = \max_x c_x, \qquad d_z = \sum_x \frac{A_{x,z}}{p_\theta(z)}\, c_x, \tag{12}$$

$$\pi_\theta(x|z,l) = \frac{c_x}{c_*} \cdot \frac{A_{x,z}}{p_\theta(z)} + \left(1 - \frac{d_z}{c_*}\right) p(x). \tag{13}$$

Encoding this expression in the architecture of $\pi_\theta(x|z,l)$ ensures that $\sum_z \pi(z)\pi_\theta(x|z,p) = p(x)$, and thus all choices of θ yield a valid reparameterization trick for all p. See Appendix for a proof. While we could parameterize and learn $\pi(z)$, we have thus far fixed it to the uniform distribution.

Sampling from the Posterior. Given a particular outcome $x \sim p$, we can sample a counterfactual y under some intervention q by first computing the posterior $\pi_\theta(z|x, l^{(1)}) \propto \pi(z)\pi_\theta(x|z, l^{(1)})$ and sampling an auxiliary variable z that is consistent with the observation. Given z, we now have a Gumbel-max coupling between $\pi_\theta(x|z,p)$ and $\pi_\theta(y|z,q)$, so the same top-down algorithm from Oberst and Sontag [2019] can be used to sample posterior Gumbels and a counterfactual outcome y.

5.3 Learning Gadgets

Recall from Sec. 3 that our goal is to learn θ so that the gadgets above produce favorable implicit couplings when measured against dataset $\mathcal{D}$ and cost function g. In both gadgets, the constraint in Eq. 6 is automatically satisfied by the architecture. Thus, we need only concern ourselves with Eq. 5, which is a minimization problem over $\mathcal{L}$ with the following form:

$$\mathcal{L}(\theta) = \mathbb{E}_{(p,q)\sim\mathcal{D}} \mathbb{E}_\gamma \left[g(f_\theta(\gamma, p), f_\theta(\gamma, q)) \right] \tag{14}$$

We would like to use a reparameterization gradient where we sample (p, q) and γ, and then differentiate the inner term with respect to θ. However, the loss is not a smooth function of θ given a realization of γ due to the argmax operations in Eqs. 9, 11. Thus, our learning strategy is to relax these argmax operations into softmaxes, as in Jang et al. [2017], Maddison et al. [2017]. This yields a smoothed $\tilde{f}_\theta \in \Delta^{K-1}$ and a smoothed softmax surrogate loss:

$$\tilde{\mathcal{L}}(\theta) = \mathbb{E}_{(p,q)\sim\mathcal{D}} \mathbb{E}_\gamma \left[\sum_{x,y} [\tilde{f}_\theta(\gamma, p)]_x \cdot [\tilde{f}_\theta(\gamma, q)]_y \cdot g(x, y) \right]. \tag{15}$$

This is differentiable, and we can optimize it using gradient based methods and standard techniques (either explicitly summing over all x, y or taking a sample-based REINFORCE gradient).

6 Related Work

Counterfactual reasoning in high-risk environments requires generating a safe alternative, instead of the observed behavior. Coupling is a natural mathematical concept that binds both the observed

behavior and the alternative choice. Maximal couplings allow reasoning about the how two random variables X, Y can be joined so that $P(X = Y)$ is maximal (cf. Ross and Peköz [2007], Chapter 2).

The variational approach for coupling relates the maximal coupling to the total variation distance $\|p - q\|_{TV} \triangleq \max_{A \subset \{1,\ldots,K\}} |p(A) - q(A)|$ since $\|p - q\|_{TV} = \inf_{\pi \in C(p,q)} \mathbb{P}_\pi[X \neq Y]$. The Wasserstein distance $W(p, q; d)$, in Eq. 4, is a generalization of the variational principle. While it reduces to the maximal coupling when $d(x, y) = 1_{x \neq y}$, it gives rise to a set a various metrics d. The Wassestein distance was mainly applied to discirminative and generative learning, e.g., [Frogner et al., 2015, Arjovsky et al., 2017, Courty et al., 2017]. The Wassestein distance can be extended to the optimal transport setting, when d is any function, which has been used extensively in machine learning, see [Alvarez-Melis et al., 2018, Benamou et al., 2015, Blondel et al., 2018, Courty et al., 2016, Cuturi, 2013, Aude et al., 2016, Kusner et al., 2015, Luise et al., 2018, Peyré et al., 2019].

The maximal coupling of Bernoulli random variables enforces monotonicity. This monotonicity is attained by sampling $u \sim Uniform([0, 1])$ uniformly and setting $X = 1[u \leq p]$ and $Y = 1[u \leq q]$, Fréchet [1951]. More generally, Strassen's theorem asserts that any two random variables satisfy $F_X(t) \leq F_Y(t)$ if and only if they are monotone, i.e., there is a coupling π for which $\mathbb{P}_\pi[Y \leq X] = 1$. The monotone coupling can be realized by using the same uniform random variable U, setting $X = F_X^{-1}(U)$ and $Y = F_Y^{-1}(U)$, Lindvall et al. [1999]. A monotone coupling $\pi \in C(p, q)$ of two marginal probabilities p and q, maximizes the covariance of (X, Y) and consequently minimize the variance of $X - Y$, since $Var_\pi[X - Y] = Var_p[X] + Var_q[\hat{Y}] - 2Cov_\pi(\hat{X}, \hat{Y})$. A coupling that maximize its correlation, or equivalently minimize the variance of its difference, is an optimal coupling. Minimizing the variance of $X - Y$ helps to stabilize their estimation in machine learning applications, e.g., Grathwohl et al. [2017].

Cambanis et al. [1976] give conditions when the coupling $(F_X^{-1}(U), F_Y^{-1}(U))$ is optimal. Let $d(x, y)$ be *supermodular*, i.e., $d(x_1, y_1) + d(x_2, y_2) \geq d(x_1, y_2) + d(x_2, y_1)$ if $x_1 \leq x_2$ and $y_1 \leq y_2$. Then $\sup_{\pi \in C(p,q)} \mathbb{E}_{x,y \sim \pi} d(x, y) = \mathbb{E}_{U \sim \text{Uniform}(0,1)} d(F_X^{-1}(U), F_Y^{-1}(U))$. Specifically, when $d(x, y) = h_1(x)h_2(y)$ and $h_i(F_X^{-1}(u))$ are non-increasing or non-decreasing functions of u, the LHS is the coupling with maximum covariance and the RHS is the coupling achieved by common random numbers paired with a CDF inversion mechanism. See also Glasserman and Yao [1992].

7 Experiments

We evaluate our approach in two settings. First, we build understanding by exploring performance on several datasets of fixed and random logits $\in R^K$. Next, we learn low-variance counterfactual treatment effects in the sepsis management MDP from Oberst and Sontag [2019].

7.1 Optimizing for maximality

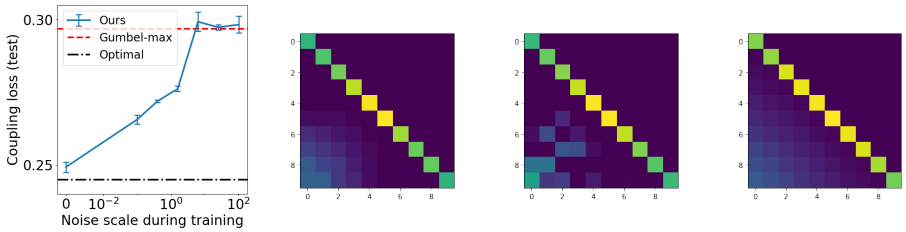
Section 4.2 shows that no implicit coupling is maximal for every pair of p and q. Here we show that we can learn near-maximal couplings if we limit attention to narrower distribution $\mathcal{D}$ of interest.

We compare our proposed learning method against fixed Gumbel-max couplings and a maximal coupling. We hold fixed a single pair of test distributions p^{test} and q^{test} and then vary distributions that are trained on. Specifically, we train Gadget 2 on pairs $(p^{test} + \rho \cdot \eta_p, q^{test} + \rho \cdot \eta_q)$ where η_p and η_q are vectors of unit variance Gaussian noise, and ρ is a noise scale. When ρ is small, it is possible to learn a coupling that is specific to the region of distributions around (p^{test}, q^{test}) and we can achieve couplings that are nearly maximal. When ρ is large, we are asking f_θ to couple together all pairs of distributions, and we expect to run into the limitations of implicit couplings in general.

Results appear in Fig. 1 (a). Indeed, when noise scales are small, our gadget learns a near-maximal coupling. When the noise scale becomes large, the quality declines. Interestingly, when the noise scale is orders of magnitude larger than the signal, the method automatically reverts back to the quality of Gumbel-max couplings and does not get significantly worse.

7.2 Minimizing variance over random logits

In this experiment, we consider the ability of our learned couplings to reduce variance in a controlled setting, exploring both randomness in p and q and randomness in the reward. In addition, we show



(a) Loss vs noise (b) Optimal (c) Ours (d) Gumbel-max

Figure 1: (a) When the training distribution is focused, we learn a near-maximal coupling. As the distribution becomes more diffuse, the learned coupling reverts to Gumbel-max. (b) Maximal coupling. (c) Our learned coupling in low-noise setting (Gadget 2). (d) Gumbel-max coupling.

Table 1: Comparison of gadgets in controlled setting, shown with standard error.

Reward:	Fixed linear		Random monotonic	Non-monotonic
p and q:	Independent	Mirrored	Fixed inc/dec	Fixed inc/dec
Independent	16.44 ± 0.07	21.42 ± 0.12	2.69 ± 0.37	0.69 ± 0.10
CDF^{-1}	8.06 ± 0.07	12.71 ± 0.13	0.58 ± 0.10	0.71 ± 0.14
Gumbel-max	13.94 ± 0.07	18.91 ± 0.13	2.48 ± 0.36	0.42 ± 0.06
Optimal (LP)	8.06 ± 0.07	12.71 ± 0.13	0.58 ± 0.10	0.09 ± 0.02
Gadget 1	13.96 ± 0.07	16.70 ± 0.13	1.48 ± 0.22	0.23 ± 0.03
Gadget 2	8.61 ± 0.07	13.53 ± 0.14	2.12 ± 0.31	0.18 ± 0.04

that our gadgets achieve a lower variance than both the inverse CDF and Gumbel-max methods under non-monotonic reward functions.

In the first part, we fixed the reward to the identity function and randomized p and q in two ways: (1) independently randomly drawn p and q, (2) p drawn randomly and q set to have the same probabilities as p but assigned in a mirrored order. This tests the abilities of our gadgets to learn to couple arbitrary distributions and to uncover relationships between p and q.

In the second part, we tested our two gadgets under varying monotonic and non-monotonic reward functions where p and q are fixed. In addition, we set p to be increasing and q are in reversed order of p, in order to examine how the gadgets perform when p is very different from q. The monotone function was constructed by sampling a K-length uniform vector followed by a cumulative summation. The non-monotone function was constructed by sampling K-length vector from a normal distribution with locations $\{0, ..., K\}$ followed by the function $R(i) = \sin(30 * i)$. At each trial, the gadgets were trained from scratch under new realization of rewards. In addition to the fixed mechanisms, we also solve the optimal coupling using a linear programming [Bertsimas and Tsitsiklis, 1997] (which may not be achievable by any implicit coupling).

Table 1 shows the results of our experiments. We find that Gadget 2 shows strong performance across all distributions of p and q, outperforming the Gumbel-max and permuted inverse CDF couplings, and approaching the results of the optimal coupling under monotone and non-monotone functions. Gadget 1 outperforms Gumbel-max in the mirrored setting, and also outperforms Gadget 2 in the fixed increasing p / decreasing q setting under monotonic functions.

7.3 MDP counterfactual treatment effects

Following Oberst and Sontag [2019], we used a synthetic environment of sepsis management to minimize the variance of counterfactual treatment effects on an SCM for MDP. We compare both our gadgets with the Gumbel-max and inverse CDF couplings.

The simulator includes four vital signs (blood pressure, oxygen concentration, and glucose levels) with discrete states (low, normal, high), as well as three treatment options (antibiotics, vasopressors, and mechanical ventilation). Our goal is to couple $p(s'|s, a_{doctor})$ and $p(s'|s, a_{intervention})$, i.e., the transition distributions induced by two policies: a behavior policy, which mimics the physician policy,

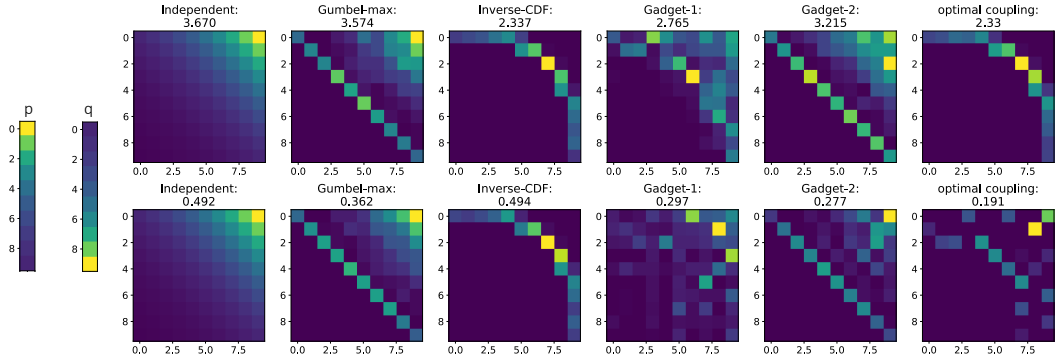


Figure 2: Couplings of each method for the increasing p / decreasing q settings along with the counterfactual effect variance for this specific reward realizations. First row: monotone reward function. Second row: non-monotone reward function.

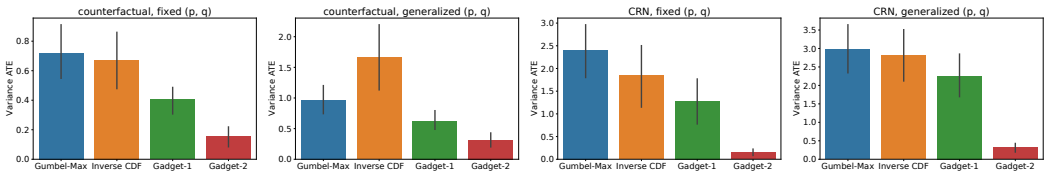


Figure 3: Variance of the treatment effect on the sepsis management simulator.

and a target policy, which is the RL policy. Our counterfactual question is what would have happened to the patient if the RL policy's action had been applied instead of the doctor's.

Using a trained behavior policy, we draw 20,000 patient trajectories from the simulator with a maximum of 20 time steps, where the states are in space of 6 discrete variables (a total of 146 state's categories). Based on them, we learn the parameters of an MDP, and train the target policy over this MDP. Unlike Oberst and Sontag [2019], we focused on coupling single time steps, and set the reward for each category of the discrete variables with a sample from a Gaussian variable. Among all the trajectories and time steps we filtered out all pairs that have less than 4 non-zero probabilities. We conducted the experiment in two settings: joint sampling, where the noise is shared between the two samples, and counterfactual sampling, where we infer the noise $u^{(env)}$ based on the observation (s, a_{doctor}, s'), then sample $(s'|s, a_{intervention})$. In each setting, we tested our gadgets when (p, q) are fixed and when (p, q) are perturbed by a Gaussian sample (generalized). At testing, we compute the treatment effect variance over 2000 samples and average the result across 10 different random seeds. With each trial of the experiment we fixed a pair of (p, q) and set a new random realization of reward. We repeated that process for 5 pairs of (p, q), each with 5 different reward realizations, a total of 15 trials for each experiment settings.

The means and the standard errors of the experiments are shown in Figure 3. Both our gadgets outperformed the fixed sampling mechanisms, while gadget-2 by a significant margin. Details on the implementation of all the experiments are in the Appendix.

8 Discussion

We have presented methods for learning a level 3 causal mechanism that is consistent with observed level 2 information. The framework provides significant flexibility to quantitatively define what makes a causal mechanism desirable, and then it uses learning to find the most desirable mechanism. A limitation is that we have only considered single-step causal processes, but we believe the framework can be extended to multi-step MDPs. In future work, we would also like to investigate qualitative properties of the learned mechanisms as $\mathcal{D}$ and g are varied and find settings that lead to mechanisms that humans find intuitive.

References

David Alvarez-Melis, Tommi Jaakkola, and Stefanie Jegelka. Structured optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pages 1771–1780. PMLR, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. arxiv 2017. *arXiv preprint arXiv:1701.07875*, 30, 2017.

Genevay Aude, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. *arXiv preprint arXiv:1605.08527*, 2016.

Elias Bareinboim, JD Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference. *ACM Special Volume in Honor of Judea Pearl (provisional title)*, 2020.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pages 880–889. PMLR, 2018.

Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *arXiv preprint arXiv:1811.06272*, 2018.

S. Cambanis, G. Simons, and W. Stout. Inequalities for ek(x, y) when the marginals are fixed. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 36:285–294, 1976.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Nicolas Courty, Rémi Flamary, and Mélanie Ducoffe. Learning wasserstein embeddings. *arXiv preprint arXiv:1710.07457*, 2017.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Maurice Fréchet. Sur les tableaux de corrélation dont les marges sont données. *Ann. Univ. Lyon, 3ˆ e serie, Sciences, Sect. A*, 14:53–77, 1951.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a wasserstein loss. *arXiv preprint arXiv:1506.05439*, 2015.

Paul Glasserman and David D Yao. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992.

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

Torgny Lindvall et al. On strassen's theorem on stochastic domination. *Electronic communications in probability*, 4:51–59, 1999.

Guy Lorberbom, Chris J Maddison, Nicolas Heess, Tamir Hazan, and Daniel Tarlow. Direct policy gradients: Direct optimization of policies in discrete action spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of sinkhorn approximation for learning with wasserstein distance. *arXiv preprint arXiv:1805.11897*, 2018.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. A* Sampling. In *Advances in Neural Information Processing Systems 27*, 2014.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*, 2017.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with Gumbel-max structural causal models. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4881–4890. PMLR, 09–15 Jun 2019. URL `http://proceedings.mlr.press/v97/oberst19a.html`.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Sheldon M Ross and Erol A Peköz. *A second course in probability*. www. ProbabilityBookstore. com, 2007.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]