On applications of epistemic uncertainty in decision-making: From self-driving cars to self-driving labs

Panagiotis Tigas
ptigas@robots.ox.ac.uk
Kellogg College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

July 2023

Abstract

Building agents that make autonomous decisions is challenging but essential in various real-world tasks like autonomous driving, personalized healthcare, and robotics. The world they interact with is complex, governed by unknown, non-stationary and non-linear dynamics, the sensory apparatus can be faulty or noisy, and information about the world can be partial. Uncertainty plays a key role in such cases since it allows for improving the safety of the autonomous agents but also learn quickly with limited data. In this thesis, we develop tools for learning and using models based on principles of Bayesian epistemology, under which an agent infers hypotheses about the world, uses them for acting and updates them based on feedback (i.e. evidence), a process termed Bayesian action-perception loop.

In the first part of the thesis (*pessimism in the face of uncertainty*), we propose a method for using the model uncertainty to improve the robustness and safety of autonomous-driving agents that operate under distribution shifts. Also, we present a benchmark and methodology for assessing sequential decision-making under distribution shifts. In the second part of the thesis (*optimism in the face of uncertainty*), we propose methods for learning personalized treatment-effect models but also causal structures of complex dynamical systems, like the ones found in gene regulatory networks. We show how we can use epistemic uncertainty to bias the acquisition of observations and interventions (also known as experiments or actions) towards informative content that help learn the models as quickly as possible, adapting Bayesian optimal experimental design (BOED) to the context of causal inference and discovery.

Overall, this thesis aims to advance the field of autonomous decision-making by providing practical tools for dealing with uncertainty and improving the efficiency of learning from data.

Acknowledgements

I wish to express my deepest gratitude to my supervisor, Yarin Gal, for his unwavering guidance and for providing the intellectual playground that allowed me to explore my scientific curiosities. His trust and our generous brainstorming sessions were instrumental to my development as a researcher; for that, I am forever grateful.

My sincere appreciation goes to Michael Osborne and Yoshua Bengio for their time and expertise in examining this thesis. I am thankful for their insightful comments and constructive suggestions.

Studying at Oxford was a dream I once considered out of reach. I was fortunate to find mentors who believed in my potential and supported my application. To Piotr Mirowski, Leonidas Palios, Tijl De Bie, and Milad Shokuhi: your help was crucial to this journey, and I cannot thank you enough.

Throughout my DPhil, I had the honor of collaborating closely with Angelos Filos, Andrew Jesson, Yashas Annadani, Zafeirios Fountas, and Noor Sajid. Through countless discussions, late-night calls, long walks, and the shared adrenaline of deadlines and submissions, they shaped my identity as a researcher. I am profoundly grateful for their friendship and for the trust they placed in me as a collaborator.

I also want to thank the members of the Oxford Applied and Theoretical Machine Learning (OATML) group for fostering such a vibrant and intellectually stimulating environment: Joost van Amersfoort, Clare Lyle, Andreas Kirsch, Aidan Gomez, Angelos Filos, Robin Ru, Neil Band, Milad Alizadeh, Lewis Smith, Sebastian Farquhar, Tim Rudner, Andrew Jesson, Jan Brauner, Pascal Notin, Soren Mindermann, Jannik Kossen, Lisa Schut, Mo Razzak, Gunshi Gupta, Kelsey Doerksen, Lorenz Kuhn, Ruben Weitzman, Shresth Malik, Freddie Bickford Smith, Jishnu Mukhoti, and Freddie Kalaitzis. I have never been surrounded by a group so talented and inspiring.

My time at Oxford would not have been the same without the other two-thirds of the "Fish & Chips" secret society: Vitaly Kurin and Alessandro De Palma. You made the journey infinitely more fun, and I feel extremely lucky to have met you both. Also, I would like to thank Kyriacos Shiarlis and Eva Schlindwein for making Oxford feel like home. I will never forget the Palindrome events, the dinners, the jam sessions, and our runs in Aston's Eyot; I am thankful for all those memories.

Beyond Oxford, I am incredibly grateful to have worked with the Geoeffectiveness team at the Frontiers Development Lab (NASA/SETI Institute). Despite the challenges of 2020 and the shift to remote work, this project remains the most fulfilling experience of my professional career. Specifically, I thank Vishal Upendran, Bashi Ferdousi, Teo Bloch, Ashti Bhatt, Ryan McGranaghan, Mark Cheung, and Siddha Ganju for such a fantastic collaboration. I also owe a debt

of gratitude to Katja Hofmann, Cheng Zhang, Sam Devlin, Ida Momennejad, and Jaroslaw Rzepecki for the invaluable lessons learned during my 2021 internship at Microsoft Research.

Finally, I want to thank my "acquired family": Dionysia Mylonaki, Eirini Maliaraki, Kostantinos Pettas, Yannis Baboulias, Dimitris Batsis, Nikos Tsipinakis, Eliza Tsaousi and Natasha Theodorelou, who supported me both emotionally and intellectually throughout this journey. Last but certainly not least, I thank my mum and dad, grandmother, and sister for their unconditional love and belief in me. I love you all.

Haggerston, London
15 July, 2023

Contents

1 Introduction

Figure 1.1: Plato's allegory of the Cave[1].

Forms and Shadows

In Plato's Republic (Plato, 375 BCE), Plato offers an allegory that describes the interaction between the world of ideas (forms) and the world of appearances

[1]Commissioned by the author of the thesis to the artist Iakovos Vais

(shadows). In the allegory, prisoners[2] watch shadows on a wall cast by objects behind them. One prisoner breaks free and realizes that what they perceived as reality was a mere illusion, a shadow of the true world behind them. This allegory illustrates how perceived reality can differ from actual reality, a concept that echoes through the challenge an autonomous agent faces in discerning the true state of the world from the sensory noise of the world it participates in.

This thesis is not about commenting on Plato's idealism, however, it serves as a vehicle to expose the ideas we aim to communicate. The main actors of this story are the *world* and the *agent* and the interaction between them. The agent acts on the world and the world responds with an observation; a process that is called action-perception loop. However, the world we describe here is a world of shadows – the true state (form) is unobservable and only an observation (shadow) can be accessed by the agent, which is a corrupted, noisy, version of the true world. The abstraction of agency can be applied to men, animals, or machines that undergo the processes of learning about their environment to reduce their uncertainty regarding the consequences of their actions and use the acquired knowledge to plan and act.

This brings us to the action-perception or cybernetic loop (Ashby, 1961, Wiener, 1948), a concept central to the understanding of autonomous agents. An autonomous agent, in basic terms, measures a function of the world state through a sensory system, processes this information, and then acts accordingly. The sequence of this interaction is usually towards a goal which is measured via a utility or objective which, in abstract terms, defines the "rationality" (Russell and Norvig, 2002), "intentions" (Dennett, 1971), "purpose" or "telos" (Rosenblueth et al., 1943) of the agent. Such a goal-oriented agent needs to act to maximise an expected value of the utility (Morgenstern and Von Neumann, 1953) while at the same time dealing with the uncertainty and risk that a stochastic, non-predictable and partially observable environment holds. With this thesis, we focus on what ways an agent can **learn** and **act** under *uncertainty* and *unpredictability*.

[2]In Fig. 1.1 prisoners are replaced by robots.

The terms *uncertainty* and *unpredictability* play an important role here, as they signify the challenges of building such autonomous agents – they are adversarial forces which affect *the ability of the agent to predict the future and act optimally.* Of course, resolving such uncertainties is not always possible, but identifying, controlling or even using them during the learning or acting process is important, especially in safety-critical applications like *autonomous driving* or *personalised healthcare.*

From the generative process to the generative model

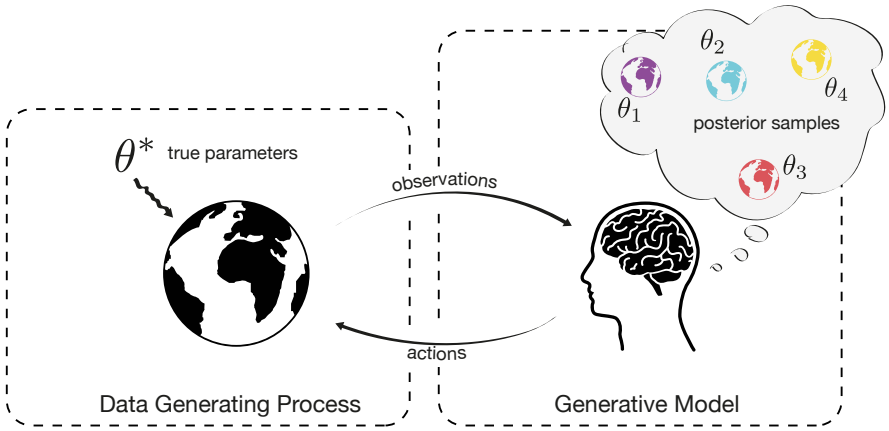


Figure 1.2: The interaction between the data generating process and the generative model.

From the modeling perspective, the world is governed by the *data generating process* (also known as *generative process*) which is the (physical) process that generates interventional and observational data. This is usually hidden from the agent and can only intervene (act) and observe it, often partially (sense).

A *forward* or *generative model* $p(\mathcal{D} \mid \Theta)$ is a model that aims to capture our assumptions about how data was generated. The model is parametrized by θ and the data is denoted by $\mathcal{D}$. The model can be used to generate new data $\mathcal{D}^*$ or to evaluate the likelihood of the data $\mathcal{D}$ given the parameters θ.

A key distinction between the generative model and the generative process is that the process captures the *objective* world where the generative model captures

the *subjective* world – how the agent believes the world works. The goal of the agent is to "match" the generative model with the generative process, thus bridge the objective world with the subjective world model. Having access to a generative model allows for predicting the consequence of the actions and being able to plan. For example, in applications like autonomous driving, a generative model can be used to predict how the car will behave on different actions, and infer a plan that moves the car safely.

Crucially, having access to a generative model allows for the computation of the inference as well – that is, computing a posterior distribution of the latent variables θ given the observations.

Epistemic uncertainty and its applications on self-driving cars and self-driving labs

What we just saw is a hint to what is called the Bayesian viewpoint. According to the Bayesian statistics, an agent maintains a distribution over the hypotheses, called belief and denoted as

$$p(\underbrace{\Theta}_{\text{hypothesis}} \mid \underbrace{\mathcal{D}}_{\text{evidence}}).$$

As a consequence of this, there might be several hypotheses that the Bayesian agent considers which might sometimes disagree with each other.

Let's consider a scientist working in a wet lab. First, she comes up with some hypotheses and then designs some experiments to help reduce the number of hypotheses. Then she enters the wet lab, executes the experiments and analyses the observations to develop new hypotheses. After several cycles, the scientist will have some surviving hypotheses, which will be considered knowledge. Viewing the scientific process as an agent that interacts with a partially observable world, in the face of some evidence, the scientist comes up with various hypotheses. The variety of the hypotheses represents the uncertainty that it has regarding the working mechanism of the object of study. Then it proceeds with careful experimentation

that disambiguates between competing hypotheses, hopefully reducing the uncertainty as a result of the experimental outcomes. The type of uncertainty that is reduced by the acquisition of observation or interventional data is what is called the *epistemic uncertainty*, and it is a key idea of this thesis.

Now let's consider a different scenario. Imagine an autonomous driving agent driving in a novel situation. The car (agent) suddenly sees an obstacle and comes up with various hypotheses regarding future trajectories. Then it uses the hypotheses to devise a plan that is safe in all the possible scenarios it imagined and commits to it, hopefully driving safely around the obstacle.

Those two examples demonstrate different attitudes an agent can have in the face of epistemic uncertainty. Being optimistic in the face of uncertainty results in a reduction of uncertainty by acquiring experiences that target the uncertainty and being pessimistic in the face of uncertainty results in robust and safe behaviour by avoiding plans that can result in outcomes that are catastrophic according to worst-case hypotheses. Our contribution to this thesis is to propose scalable and tractable methods for *safe and robust planning* under high epistemic uncertainty in the setting of *autonomous driving* (part A) and methods that suggest *where and how* to experiment or acquire data to reduce the uncertainty in the setting of personalized healthcare and wet-lab science (part B). The latter part focuses on what is termed self-driving laboratories (Häse et al., 2019, Abolhasani and Kumacheva, 2023) or AI-driven scientific discovery (Jain et al., 2023).

Contributions

The thesis is based on the following published body of work:

1. Filos, Angelos*,**Tigas, Panagiotis***, McAllister, Rowan, Rhinehart, Nick, Levine, Sergey, Gal, Yarin (2020, November). *"Can autonomous vehicles identify, recover from, and adapt to distribution shifts?"*. In International Conference on Machine Learning (pp. 3145-3153). PMLR.
2. Andrey Malinin, Neil Band, Ganshin, Alexander, German Chesnokov, Yarin Gal, Mark J. F. Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila

Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Roginskiy, Denis, Mariya Shmatova, **Panagiotis Tigas**, Boris Yangel. *"Shifts: A dataset of real distributional shift across multiple large-scale tasks"*. In Neural Information Processing Systems 34, Datasets and Benchmarks Track, 2021.

3. Jesson, Andrew*, **Panagiotis Tigas***, Joost van Amersfoort, Andreas Kirsch, Uri Shalit, and Yarin Gal. *"Causal-BALD: Deep Bayesian active learning of outcomes to infer treatment-effects from observational data"*. In Advances in Neural Information Processing Systems 34 (2021): 30465-30478.
4. **Tigas, Panagiotis***, Yashas Annadani*, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. *"Interventions, where and how? experimental design for causal models at scale."*. In Advances in Neural Information Processing Systems 35 (2022)
5. **Tigas, Panagiotis***, Yashas Annadani*, Desi R. Ivanova, Andrew Jesson, Yarin Gal, **Adam Foster, and Stefan Bauer**. *"Differentiable Multi-Target Causal Bayesian Experimental Design."*. In 2023 International Conference on Machine Learning

Projects 1 and 2 fall under the theme of using decision-making under uncertainty with epistemic-uncertainty aware model-based planning. Projects 3,4 and 5, focus on using epistemic uncertainty to learn causal world models using principles from active learning and Bayesian optimal experimental design.

2
Background

A Bayesian route to autonomy

In this thesis, we will use the Bayesian framework as the language of uncertainty. The Bayesian viewpoint allows for using probability to quantify beliefs and uncertainty but also provides us with a formalism for updating our beliefs in the presence of new evidence. Before we discuss the different flavours of uncertainty in Chapter 2, we will give an overview of the Bayesian framework, diagrams for expressing statistical and causal assumptions along with measures of uncertainty like entropy and mutual information.

The Bayesian perspective

A fundamental perspective that Bayesian framework offers is what is called the *subjectivist* view. According to this view, the probability is measuring the degree of belief of an agent regarding the truthfulness of a hypothesis H. It is subjective in the sense that the probability doesn't express a fact about the world or nature but rather the point of view of the agent. Crucially, the Bayesian framework separates the *objective* world from the *subjective* agent. This is an important point that we will revisit several times during the thesis.

Let's assume a hypothesis as an element of a hypothesis space $\mathcal{H}$. For example, in the case of toxicity prediction, the hypothesis space can be the set of all possible models that can predict toxicity. In the Bayesian framework, the modeler first defines a prior belief about the hypotheses in the form of a probability distribution $p(H)$ called the prior. Assuming the presence of some evidence O, we can define the *likelihood* function $p(O \mid h)$ which gives the probability of the outcome assuming a hypothesis h. This captures how much the hypothesis explains the evidence. Having access to the prior and the likelihood, *Bayes theorem* offers a rule for updating an agent's beliefs (distribution over hypotheses) in the presence of some evidence O:

$$\overbrace{p(H \mid O)}^{\text{posterior}} = \frac{\overbrace{p(O \mid H)}^{\text{likelihood}} \overbrace{p(H)}^{\text{prior}}}{\underbrace{\int p(O \mid H) p(H) dH}_{\text{marginal likelihood}}}. \tag{2.1}$$

Entropy, Mutual Information and Relative Entropy

Assuming a distribution over a random variable X, we can define the quantity

$$H(X) = -\mathbb{E}_{x \sim p(X)} \left[\log p(x) \right]. \tag{2.2}$$

This quantity, named **entropy**, quantifies the *"information"* or the *"uncertainty"* of the distribution $P(X)$ (Shannon, 1948). Intuitively, the entropy measures the diversity of the samples of the distribution $p(X)$ and under the Bayesian interpretation of probability, the diversity of the beliefs regarding the true value of X. High entropy implies that the distribution has high uncertainty regarding the random variable X. Furthermore, we can now define the **conditional entropy** as $H(Y \mid X) = -\mathbb{E}_{(x,y) \sim p(X,Y)} [\log \frac{p(x,y)}{p(x)}]$.

Relative entropy or **Kullback-Leibler (KL)** divergence (Kullback and Leibler, 1951) between two probability distributions $p(X)$ and $q(X)$ is defined as

$$\mathrm{D}_{\mathrm{KL}}(p \parallel q) = \mathop{\mathbb{E}}_{x \sim p(X)} \left[\log \frac{p(x)}{q(x)} \right]$$

and measures how a distribution $p(X)$ differs from a reference distribution $q(X)$. As we will see in next section, KL divergence plays an important role in variational

inference as it allows quantify how similar are two distributions. $\mathrm{D}_{\mathrm{KL}}(p \parallel q)$ satisfies Gibbs' inequality $\mathrm{D}_{\mathrm{KL}}(p \parallel q) \geq 0$ and it becomes 0 when $p = q$.

Finally, **mutual information**, defined as $I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$, measures the average reduction of uncertainty about X that results from measuring Y or stated differently, how much information does X contain about Y and vice versa.

Variational Inference

Computing the posterior distribution in Eq. (2.1) involves the computation of the marginal likelihood quantity. Such computation is usually intractable as integrating over non-closed-form continuous (or high-cardinality but discrete) random variables is computationally expensive or impossible, so we appeal to approximate posterior inference. *Variational Inference* (Jordan et al., 1999) turns the intractable inference problem (computing the posterior $p(\theta \mid \mathcal{D})$) into a tractable approximation, where the optimization objective is to minimize the Kullback-Leibler (KL) divergence between the approximate posterior distribution $q_\varphi(\theta)$, parametrized by φ, and the true posterior distribution $p(\theta \mid \mathcal{D})$.

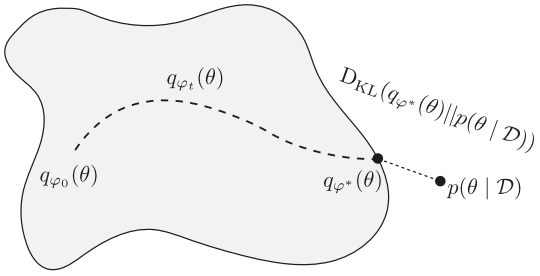


Figure 2.1: Variational Inference turns inference into an optimization problem. The dashed line shows the optimization trajectory of the approximate posterior as it minimizes $\mathrm{D}_{\mathrm{KL}}(q_{\varphi^*}(\theta) \parallel p(\theta \mid \mathcal{D}))$.

However, solving this optimization problem is yet not tractable. If we expand

$\mathrm{D}_{\mathrm{KL}}(q_{\varphi^*}(\theta) \parallel p(\theta \mid \mathcal{D}))$ we see:

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(q_{\varphi^*}(\theta) \parallel p(\theta \mid \mathcal{D})) &= \mathbb{E}_{q_{\varphi^*}(\theta)}[\log q_{\varphi^*}(\theta)] - \mathbb{E}_{q_{\varphi^*}(\theta)}[\log p(\theta \mid \mathcal{D})] && (2.3)\\
&= \mathbb{E}_{q_{\varphi^*}(\theta)}[\log q_{\varphi^*}(\theta)] - \mathbb{E}_{q_{\varphi^*}(\theta)}[\log p(\theta, \mathcal{D})] + \log p(\mathcal{D}) && (2.4)\\
&= -\mathrm{ELBO}(\varphi) + \log p(\mathcal{D}), && (2.5)
\end{aligned}$$

thus, minimizing the KL requires access to the *marginal likelihood* $p(\mathcal{D})$, also known as *model evidence*. In Eq. (2.5), we separate the right-hand side of the equation into quantities that depend on the variational parameters φ and the log model evidence $\log p(\mathcal{D})$. As $\mathrm{D}_{\mathrm{KL}}(q_{\varphi^*}(\theta) \parallel p(\theta \mid \mathcal{D})) \geq 0$ (Kullback and Leibler, 1951), we can see that the log evidence is lower bounded by the *Evidence Lower Bound (ELBO)*: $\mathrm{D}_{\mathrm{KL}}(q_{\varphi^*}(\theta) \parallel p(\theta \mid \mathcal{D})) \geq 0 \Rightarrow -\mathrm{ELBO}(\varphi) + \log p(\mathcal{D}) \geq 0 \Rightarrow \log p(\mathcal{D}) \geq \mathrm{ELBO}(\varphi)$. Thus, by maximizing the ELBO, we minimize the KL divergence between the approximate and true posterior.

$$\begin{aligned}
\varphi^* &= \arg\max_{\phi} \mathbb{E}_{q_\varphi(\theta)}\left[\log \frac{p(\theta, \mathcal{D})}{q_\varphi(\theta)}\right] && (2.6)\\
&= \arg\max_{\phi} \mathbb{E}_{q_\varphi(\theta)}\left[\log \frac{p(\mathcal{D} \mid \theta)p(\theta)}{q_\varphi(\theta)}\right] && (2.7)\\
&= \arg\max_{\phi} \underbrace{\mathbb{E}_{q_\varphi(\theta)}\left[\log p(\mathcal{D} \mid \theta)\right]}_{\text{accuracy}} - \underbrace{\mathrm{D}_{\mathrm{KL}}(q_\varphi(\theta) \parallel p(\theta))}_{\text{complexity}} && (2.8)
\end{aligned}$$

Diagrams of decision-making

Here we introduce diagrammatic ways of representing statistical and causal associations between random variables among other assumptions that can be useful when analysing and modeling decision-making problems.

From statistical to causal modeling

In his book, Reichenbach (1956) states what is known as the **common cause principle**. If two random variables X and Y are statistically dependent, then there exists a random variable Z (confounder) that causally influences X and Y. In the special case where X influences Y (or the other way around), Z coincides with the causal parent.

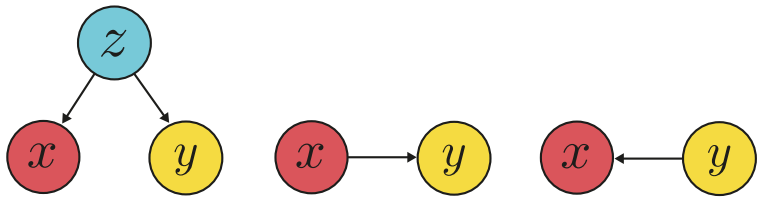


Figure 2.2: Reichenbach common cause principle.

A key distinction brought by this insight is that the statistical dependency between random variables is an *epiphenomenon*, a symptom of causal relationships – the statistical is how causality reveals itself. However, this projection is bringing ambiguity. As stated by Reichenbach, there are many different causal structures that can appear the same – both $X \leftarrow Y$ and $X \rightarrow Y$, both because X and Y to be statistically dependent. The way to break such ambiguities is by conducting experiments, acting in the world or intervening (Pearl, 2009).

Crucially, this fits very naturally in the Bayesian action-perception loop. There are many causal structures (hypotheses) that explain the observation (statistical dependency of random variables) and action (intervention) allows for resolving such ambiguities (causal discovery).

Next we will describe the tools for describing both the statistical and the causal relationships but also the operations required for disambiguating between the causal and the statistical via interventions.

Statistical Modeling

Graphical models are graphical abstractions that allow the modeling of statistical associations between random variables. In this thesis we will focus on *directed acyclic graphs*, also known as Bayesian Networks.

Bayesian Networks

As random variables represent parts of phenomena that interact with each other, it's important to be able to explicitly capture this structure in a graphical, dia-

grammatic way. A formal way to model such structured probabilistic models is Bayesian Networks, also known as belief nets (Pearl, 1985).

Bayesian networks are directed acyclic graphs (DAGs) where the nodes represent the nodes of interest and directed edges represent statistical dependencies among the variables. More formally:

A BayesNet is defined by a directed acyclic graph (DAG) $\mathcal{G}$ in random variables x. The vertices of the graph are the random variables and a set of conditional probability distributions $p(x \mid \mathrm{Pa}_{\mathcal{G}}(x_i))$ where $\mathrm{Pa}_{\mathcal{G}}(x_i)$ returns the parents of x_i in $\mathcal{G}$. The joint distribution over x is then factorized as $p(x) = \prod_i p(x \mid \mathrm{Pa}_{\mathcal{G}}(x_i))$.

The Bayesian network allows for expressing assumptions, efficient factorization of the joint distribution and thus efficient inference from evidence (observations) (Pearl et al., 2000).

Obervational Equivalence / Markov Equivalence Class: Two DAGs are observationally equivalent (or equally they belong to the same *Markov Equivalence Class* (MEC)) if they have the same skeletons and same sets of *v-structures* (2 or more parent nodes direct to the same child node) (Verma and Pearl, 1990). This important property shows also the limit of what is possible with observational data. Observational equivalence tells us that there might be several different Bayesian networks that can explain our observational data. If we would like to recover a causal model that represents causal links between random variables, then we would need to make use of *interventions*.

Causal Modeling

However, Bayesian networks allow also for causal relationships to be captured and allow for additional queries, known as interventions. An intervention is an operation which, in contrast to evidential queries, asks *"How does an action change the outcome?"*. A key distinction between evidential and interventional queries is how we condition the action versus the observation. Given an observation (evidence), the outcome is still influenced by the parents of the observation, thus rare evidence

will not contribute as much as less rare evidence. In contrast, an intervention is a deliberate action that disconnects the statistical properties of the system of study and thus the probability of the intervention becomes 1 – we know it has happened.

Structural Causal Models (SCM)

> *"We ought to regard the present state of the universe as the effect of its antecedent state and as the cause of the state that is to follow. An intelligence knowing all the forces acting in nature at a given instant, as well as the momentary positions of all things in the universe, would be able to comprehend in one single formula the motions of the largest bodies as well as the lightest atoms in the world, provided that its intellect were sufficiently powerful to subject all data to analysis; to it nothing would be uncertain, the future as well as the past would be present to its eyes. The perfection that the human mind has been able to give to astronomy affords but a feeble outline of such an intelligence."*
> *Laplace (1820)*

Laplace's view on laws of nature was that the world deterministically propagates states where the initial states are stochastic. Structural causal models are models that capture the "Laplacian quasi-deterministic conception of causality" (Pearl et al., 2000). To achieve this, it disentangles the deterministic effects (functional relationships) from the sources of stochasticity (exogenous variables).

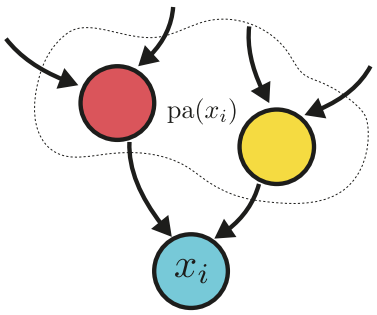


Figure 2.3: Structural Causal Model (SCM).

From the data generative mechanism point of view, the DAG $\mathbf{g}$ on $\mathbf{X_V}$ matches a set of *structural equations*:

$$X_i \coloneqq f_i(X_{\mathrm{pa}_{\mathbf{g}}(i)}, \epsilon_i) \quad \forall i \in \mathbf{V} \tag{2.9}$$

where f_i's are (potentially nonlinear) causal mechanisms that remain invariant when intervening on any variable $X_j \neq X_i$. ϵ_i's are exogenous noise variables with

an arbitrary distribution that are mutually independent, i.e. $\epsilon_i \perp\!\!\!\perp \epsilon_j \forall i \neq j$. (2.9) represents the conditional distributions in a Causal Bayesian Network and can additionally reveal the effect of interventions if the mechanisms are known (Peters et al., 2017, Pearl, 2009). These equations, along with the DAG $\mathbf{g}$, are known as the *structural causal model* (SCM). Though the mechanisms f can be nonparametric in the general case, we assume that there exists a parametric approximation to these mechanisms with parameters $\boldsymbol{\gamma} \in \Gamma$. In the case of linear SCMs, $\boldsymbol{\gamma}$ corresponds to the weights of the edges in E. In the nonlinear case, they could represent the parameters of a nonlinear function that parameterizes the mean of a Gaussian distribution. A common form of (2.9) corresponds to Gaussian additive noise models (ANM)[1]:

$$X_i := f_i(X_{\mathrm{pa}_{\mathbf{g}}(i)}; \boldsymbol{\gamma}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_i^2) \tag{2.10}$$

The ladder of causality

Pearl introduced what is called the ladder of causality to discuss the different levels of questions a causal model can answer:

Level 1 (prediction)

Observational queries can be conducted by sampling the exogenous variables first (exogenous noise) and then propagating the noise via the use of ancestral sampling. The final values of the endogenous random variables correspond to an observational sample.

Level 2 (interventions)

Interventional queries (e.g. $\mathrm{do}(X_i = k)$) can be applied by a process called mutilation which corresponds to the removal of the incoming edges of the intervened node, fixing the value of the random variable x_i to the value of the intervention k and then conducting ancestral sampling, similar to the observational sample case.

[1] ANM's can have noise variables that are non-Gaussian as well, but we restrict our exposition to the Gaussian case.

Level 3 (counterfacturals)

To conduct a counterfactional query, one needs to first solve the SCM system to recover the values of the exogenous random variables corresponding to the observed sample and then apply a do operation with the desired counterfactual query on the same exogenous values.

Rubin's Model of Causality

An alternative to Pearl's Causal Bayesian Networks was developed simultaneously in the 70s by Don Rubin (Rubin, 1974), building on work of Jerzy Spawa-Neyman (Splawa-Neyman et al., 1990). Rubin developed what is called the *Potential Outcomes Model*, also known as *Rubin's Causal Model (RCM)*, according to which the causal effect is defined as a comparison between two states of the world. For example, in answering the question of what is the effect of aspirin on the headache, according to RCM, the causal effect can be estimated by comparing the headache state before and after the intervention, in this case, the admission of aspirin.

For an individual (also called *unit* in RCM literature) x, and a binary treatment, $T \in \{0, 1\}$, we can represent the outcomes of each treatment via a random variable $Y^0(x)$ and $Y^1(x)$, corresponding to the outcomes of treatment $T = 0$ (control) and $T = 1$ (treatment). In our headache example, the patient will be represented with x, taking the aspirin as $T = 1$, not taking the aspirin as $T = 0$ and the corresponding outcomes, $Y^0(x)$ and $Y^1(x)$, the outcome we would observe had the patient received or not received the aspirin. Considering this notation, now we can define the individual-level causal effect for individual $X = x$ as the difference $Y^1(x) - Y^0(x)$. A fundamental limitation of this definition is that we cannot observe the effect of the control $T = 0$ and treatment $T = 1$ at the same time – a limitation known as "the fundamental problem of causal inference" (Holland, 1986). The implication of this is that causal inference is impossible, however under certain assumptions we can estimate the causal effect with the expected difference in potential outcomes for individuals described by $\mathbf{X}$, or the *Conditional Average Treatment Effect (CATE)*: $\tau(\mathbf{x}) \equiv \mathbb{E}[\mathrm{Y}^1 - \mathrm{Y}^0 \mid \mathbf{X} = \mathbf{x}]$ (Abrevaya et al., 2015).

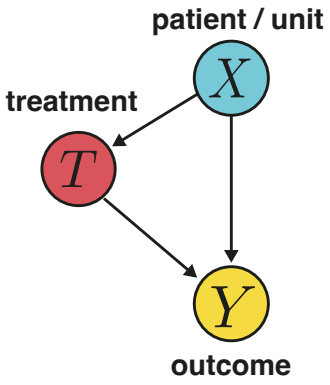


Figure 2.4: Graphical model depicting Potential Outcome framework, also known as Rubin's Causal Model (RCM). The outcome Y is influenced by the patient X and the treatment T.

The CATE is identifiable (i.e we can estimate the causal quantity from statistical quantites) from an observational dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathrm{t}_i, \mathrm{y}_i)\}_{i=1}^{n}$ of samples $(\mathbf{x}_i, \mathrm{t}_i, \mathrm{y}_i)$ from the joint empirical distribution $P_{\mathcal{D}}(\mathbf{X}, \mathrm{T}, \mathrm{Y}^0, \mathrm{Y}^1)$, under the following three assumptions (Rubin, 1974):

Assumption 1. *(Consistency)* $\mathrm{y} = \mathrm{t}\mathrm{y}^{\mathrm{t}} + (1-\mathrm{t})\mathrm{y}^{1-\mathrm{t}}$, *i.e. an individual's observed outcome* y *given assigned treatment* t *is identical to their potential outcome* y^{t}.

Assumption 2. *(Unconfoundedness or Ignorability)* $(\mathrm{Y}^0, \mathrm{Y}^1) \perp\!\!\!\perp \mathrm{T} \mid \mathbf{X}$.

Assumption 3. *(Overlap)* $0 < \pi_{\mathrm{t}}(\mathbf{x}) < 1 : \forall \mathrm{t} \in \mathcal{T}$,

where $\pi_{\mathrm{t}}(\mathbf{x}) \equiv \mathrm{P}(\mathrm{T} = \mathrm{t} \mid \mathbf{X} = \mathbf{x})$ is the **propensity for treatment** for individuals described by covariates $\mathbf{X} = \mathbf{x}$. When these assumptions are satisfied, $\widehat{\tau}(\mathbf{x}) \equiv \mathbb{E}[Y \mid \mathrm{T} = 1, \mathbf{X} = \mathbf{x}] - \mathbb{E}[Y \mid \mathrm{T} = 0, \mathbf{X} = \mathbf{x}]$ is an unbiased estimator of $\tau(\mathbf{x})$ and is identifiable from observational data.

A variety of parametric (Robins et al., 2000, Tian et al., 2014, Shalit et al., 2017) and non-parametric estimators (Hill, 2011, Xie et al., 2012, Alaa and van der Schaar, 2017, Gao and Han, 2020) have been proposed for CATE. Here, we focus on parametric estimators for compactness. Parametric CATE estimators assume that outcomes y are generated according to a likelihood $p_\theta(\mathrm{y} \mid \mathbf{x}, \mathrm{t})$, given measured covariates $\mathbf{x}$, observed treatment t, and model parameters θ. For continuous

outcomes, a Gaussian likelihood can be used: $\mathcal{N}(\mathrm{y} \mid \widehat{\mu}_\theta(\mathbf{x}, \mathrm{t}), \widehat{\sigma}_\theta(\mathbf{x}, \mathrm{t}))$. For discrete outcomes, a Bernoulli likelihood can be used: $\mathrm{Bern}(\mathrm{y} \mid \widehat{\mu}_\theta(\mathbf{x}, \mathrm{t}))$. In both cases, $\widehat{\mu}_\theta(\mathbf{x}, \mathrm{t})$ is a parametric estimator of $\mathbb{E}[Y \mid \mathrm{T} = \mathrm{t}, \mathbf{X} = \mathbf{x}]$, which leads to: $\widehat{\tau}_\theta(\mathbf{x}) \equiv \widehat{\mu}_\theta(\mathbf{x}, 1) - \widehat{\mu}_\theta(\mathbf{x}, 0)$, a parametric CATE estimator.

Uncertainty and its sources

The task of modeling is concerned with approximating a physical process which is usually governed by some equations. We assume that such a process can be written probabilistically as $y \sim p(y \mid x, \theta^*)$. This probabilistic formulation allows for expressing *inherent* stochasticities that might appear in the physical process. However, this viewpoint leaves us with an unsatisfactory sentiment. What is the source of the stochasticities? Laplace in his thought experiment (Marquis de Laplace, 1902) hypothesised a daemon that has access to the true state of the world and as such, from the perspective of Laplace's daemon, there is no stochasticity involved in the process. However, for an inferior agent, with bounded knowledge about the true state of the world, the same phenomenon would appear stochastic. Such an example could be the one of the toss of a coin – we (the bounded agents) cannot predict the outcome however for a Laplacian demon the coin outcome is determined by knowing the initial conditions of the experiment. The limit of predictability of an agent is captured by the notion of **aleatoric uncertainty**.

On the other hand, similar to the process a scientist would follow, a notion of uncertainty can be reduced (up to a limit) by accumulating more evidence that would help them "discover" the hypothesis responsible for the phenomenon. Such type of uncertainty would be called **epistemic uncertainty**[2].

The key implication of this is that a taxonomy of the flavours of uncertainty cannot be complete without considering certain assumptions, or as Der Kiureghian and Ditlevsen (2009) put it, "it is the job of the model builder to make a distinction between aleatoric and epistemic uncertainty".

[2]Which is derived by the Greek word episteme which means scientific.

Under the Bayesian viewpoint, an agent is inferring hypothesis based on some evidence. This is captured by the posterior distribution $p(\theta \mid \mathcal{D})$, where θ is the random variable of the parameters of the models and $\mathcal{D}$ the evidence (dataset). A sample from the posterior induces a distribution over outcomes $p(y \mid x, \theta)$. Crucially, the posterior predictive distribution $p(y \mid x, \mathcal{D}) = \int p(y \mid x, \theta) p(\theta \mid \mathcal{D}) d\theta$ is integrating over the uncertainty induced by the posterior distribution.

We can define the expected *aleatoric* uncertainty as

$$\mathop{\mathbb{E}}_{p(\theta|\mathcal{D})} \left[H[p(y \mid x, \theta)] \right] = \quad - \mathop{\mathbb{E}}_{p(\theta|\mathcal{D}) p(y|x,\theta)} \left[\log p(y \mid x, \theta) \right], \tag{2.11}$$

and the total (or predictive uncertainty)

$$H[p(y \mid x, \mathcal{D})] = \mathop{\mathbb{E}}_{p(y|x,\mathcal{D})} \left[- \log p(y \mid x, \mathcal{D}) \right] \tag{2.12}$$

$$= \mathop{\mathbb{E}}_{p(\theta|\mathcal{D}) p(y|x,\theta)} \left[- \log p(y \mid x, \mathcal{D}) \right]. \tag{2.13}$$

Interestingly, by subtracting *aleatoric uncertainty* from *total uncertainty* we get:

$$H[p(y \mid x, \mathcal{D})] - \mathop{\mathbb{E}}_{p(\theta|\mathcal{D})} \left[H[p(y \mid x, \theta)] \right] \tag{2.14}$$

$$= \mathop{\mathbb{E}}_{p(\theta|\mathcal{D}) p(y|x,\theta)} \left[- \log p(y \mid x, \mathcal{D}) \right] + \mathop{\mathbb{E}}_{p(\theta|\mathcal{D}) p(y|x,\theta)} \left[\log p(y \mid x, \theta) \right] \tag{2.15}$$

$$= \mathop{\mathbb{E}}_{p(\theta|\mathcal{D}) p(y|x,\theta)} \left[\log \frac{p(y \mid x, \theta)}{p(y \mid x, \mathcal{D})} \right] \tag{2.16}$$

$$= I(y; \theta \mid x, \mathcal{D}). \tag{2.17}$$

The takeaway of this is that the mutual information between outcomes and model parameters $I(y; \theta \mid x, \mathcal{D})$ amounts to the total uncertainty minus the expected *aleatoric uncertainty*, which is the *epistemic uncertainty*. As we will see later in Chapter 2, we can use this objective to guide the selection of experiences to add to the dataset in order to reduce the uncertainty – a key idea behind *experimental design*, *active learning* and *exploration*.

On the pathologies of finite data, model misspecification and partial observability

A dataset corresponds to samples from a world (the world can be seen as a function; later we will define the world as a data-generating process). The world can be seen an arbitrarily complex function $f^*(\cdot)$ which maps inputs to outputs via a noisy mapping $\boldsymbol{y} = f^*(\boldsymbol{x}) + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is an independent (to input) noise (Fig. 2.5).

As the world can be arbitrarily complex, a model will need to be sufficiently complex in order to capture data complexity. The complexity of the models can be thought of as the family of functions it can approximate. We notate the family of the functions as $\mathcal{F}$. Thus, a model $f_\theta \in \mathcal{F}$, indexed by θ belongs to the family $\mathcal{F}$. In deep learning, the index θ takes the form of the weights of the network and the family $\mathcal{F}$ (also known as inductive bias) corresponds to the architecture of the network. In a way, the parametrized model becomes a "distilled" version of the dataset. However, a model representing the dataset is not the same as the dataset itself. The model is a simplified representation of the dataset, and will not be able to perfectly reproduce the dataset unless the function representing the world is part of the same function family $\mathcal{F}$ and we have access to infinite many samples.

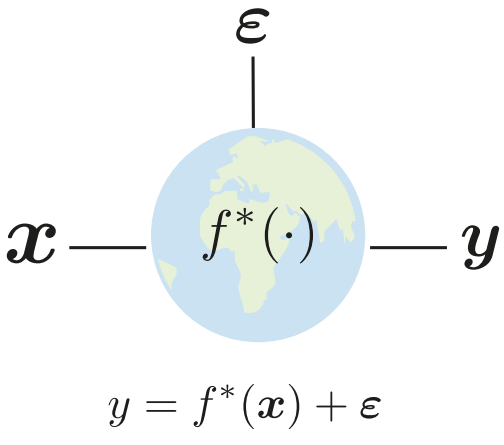


Figure 2.5: The world as a function.

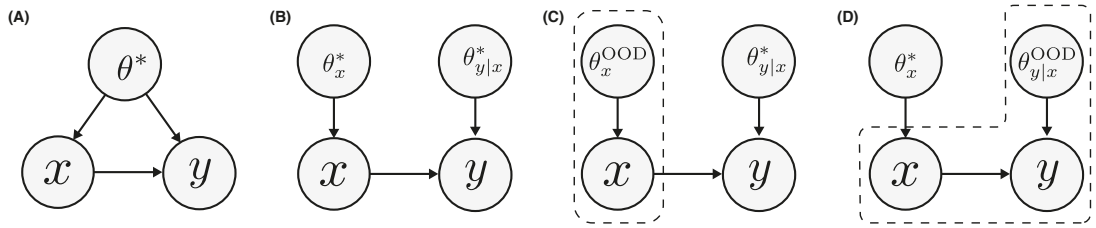
Additionally, the noise variable $\boldsymbol{\varepsilon}$ introduces additional challenges to the modeling effort. In reality, the physical process underlying the phenomenon is hidden

from us, however, even if had access to the true function $f^*(\cdot)$, we wouldn't be able to model the system up to some noise. This noise represents intrinsic (internal to the system) stochasticities or artifacts of the measurement apparatus one is using to sample from the world.

Thus, we can already see three challenges a designer of a model has to face: one is that we rarely have access to infinite many samples (finite samples assumption), another is that our function family $\mathcal{F}$ might not be expressive enough to capture the true function (world) – this is known as model misspecification – and finally, the world is usually partially observable as measurement devices are noisy or the true phenomenon of study contains some stochastic variations. The impact of these limitations is that a decision maker relying on such models will need to express a notion of uncertainty – what is known, what can be known and what cannot be known to be known.

Distribution Shifts

Figure 2.6: Examples of distribution shifts. (A) The data generating process parametrized by θ^* (B) The same data generating process but decomposed θ^* in parameters over inputs and over the conditional distribution $p(y \mid x, \theta^*_{y|x})$ (C) **covariate shift**: the conditional distribution remains the same but the prior over inputs $p(x \mid \theta^{\text{OOD}}_x)$ changes. (D) **Concept Drift**: the distribution over inputs remains the same but the relationship between inputs x and outputs y is different.



Consider the full data-generating process $p(\mathcal{D} \mid \theta^*)$, where θ^* are the true parameters of the world and $\mathcal{D}$ is the random variable representing the dataset. To learn a predictor on the dataset $\mathcal{D}$, statistical learning theory suggests to minimize the empirical risk $f^* = \text{argmin}_{f \in \mathcal{H}} \mathbb{E}_{D_{\text{train}} \sim p(\mathcal{D}|\theta^*)} [R(\mathcal{D}_{\text{train}}, f)]$.

However, in reality, at test time, the data-generating process might be different from the training data-generating distribution $p(\mathcal{D} \mid \theta^{\text{OOD}}) \neq p(\mathcal{D} \mid \theta^*)$, where θ^{OOD} is the parametrization of the test-time data-generating process (OOD here stands for Out of (training) Distribution). In Fig. 2.6 we can see different distribution shifts that a decision-making agent might face. For example, a *covariate shift* might occur when a model predicting the outcome of a drug to patients is tested on patients of a different country than the one the training data generated from. Concept drift on the other hand might occur in situations like autonomous driving, where environmental conditions like temperature might change the friction coefficients and as such how the car will respond to different actions.

Learning models

Deep Neural Networks

Although the field of machine learning has been around for decades, deep learning has only recently emerged as a powerful tool for decision-making. Deep learning is a subfield of machine learning that uses *(deep) neural networks* (Rosenblatt, 1958) to learn complex representations of data. Deep neural networks are composed of multiple layers of nonlinear transformations, which allow them to learn complex, non-linear relationships between inputs and outputs. More formally, a (deep) neural network is a function $f_\theta : \mathbb{R}^n \to \mathbb{R}^m$, parametrized by θ, where θ is a vector of parameters. The function f_θ is a composition of several functions, each of which is a non-linear transformation of the previous function.

More formally, a layer is defined as a function $f(\boldsymbol{x}; \boldsymbol{w}, b) = \alpha(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w} + b)$, where $\boldsymbol{x}$ (input) and $\boldsymbol{w}$ (weights) are vectors and b a scalar (bias). The function $\alpha(\cdot)$ is a non-linear function called activation function. To simplify the notation, we consider the weights vector $\boldsymbol{\theta}$ consisting of $\boldsymbol{w}$ and b. A simple multilayer neural network is a composition of layers $\ldots(f(f(f(\boldsymbol{x}; \boldsymbol{\theta_1}); \boldsymbol{\theta_2}); \boldsymbol{\theta_3})\ldots$.

Although the architecture has evolved to more complicated structures (e.g. CNNs (LeCun et al., 1989, Krizhevsky et al., 2017), Recurrent Neural Networks (Rumel-

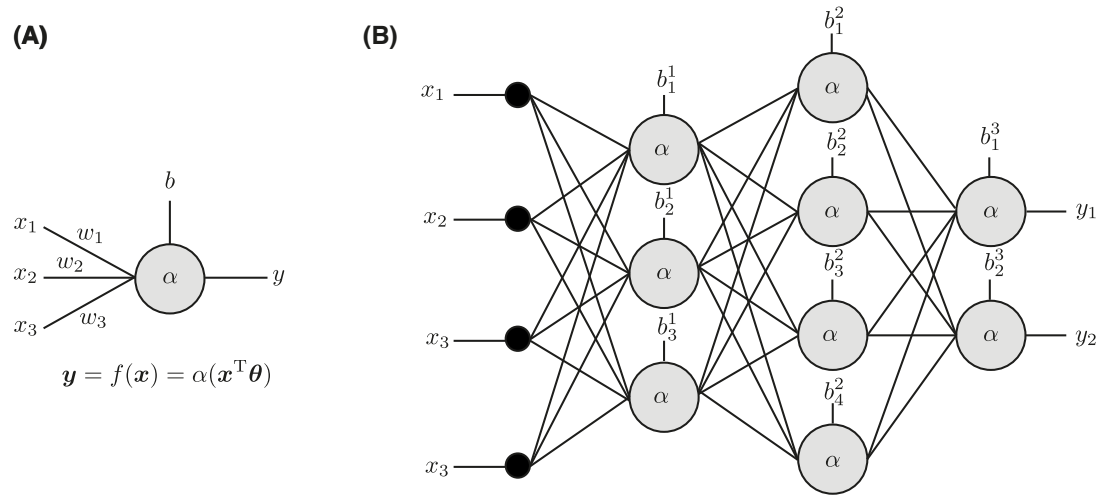


Figure 2.7: (A) A single neuron (perceptron) is a linear mapping $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\theta}$ of the inputs passed through a non-linear function $\alpha(\cdot)$ called the activation function. (B) a multilayer-perceptron (MLP).

hart et al., 1985, Graves and Graves, 2012, Chung et al., 2015), transformers (Vaswani et al., 2017)), the main principles remain the same – deep learning is concerned with composition of non-linear functions parametrized by some weight matrix $\boldsymbol{\theta}$. Without loss of generality, we will assume this abstract depiction of deep neural networks for the rest of the thesis.

Assuming a true function $f^*(\boldsymbol{x})$ (the world) and a dataset

$$\mathcal{D} = \{(\boldsymbol{x_1}, f^*(\boldsymbol{x_1})), \ldots, (\boldsymbol{x_n}, f^*(\boldsymbol{x_n}))\},$$

the goal of training a deep neural network is to learn $f^*(\cdot)$ by minimizing the error between $f(\boldsymbol{x_i}; \boldsymbol{\theta})$ and $f^*(\boldsymbol{x_i})$, by updating the weights $\boldsymbol{\theta}$. The error function, more commonly known as the loss function $\mathcal{L}(\boldsymbol{\theta})$, captures how well the model induced by the weights $\boldsymbol{\theta}$ models the data $\mathcal{D}$. For example, a common loss used in regression is the Mean Squared Error (MSE) $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|}\sum_i^{|\mathcal{D}|}(f^*(\boldsymbol{x_i}) - f(\boldsymbol{x_i}; \boldsymbol{\theta}))^2$.

Bayesian Deep Learning

Bayesian Deep Learning (BDL) is the subfield of Deep Learning that interprets the parameters of Deep Neural Networks as random variables and defines a posterior distribution over the parameters assuming some evidence. More concretely, in the

context of machine learning, evidence can take the form of the dataset $\mathcal{D} = \{X, Y\}$ and the hypothesis the form of some parameters $\boldsymbol{\Theta}$ (e.g. weights of a deep neural network), in which case the posterior is written as $p(\boldsymbol{\theta} \mid \mathcal{D})$. By the use of Bayes rule,

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}}. \tag{2.18}$$

Assuming for a moment that we have access to such posterior model, a prediction can be performed via an integral called the *posterior predictive* or *Bayesian model average (BMA)*:

$$p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) = \int p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta}. \tag{2.19}$$

The main challenge regarding Bayesian Deep Learning is that in both Eq. (2.18) and Eq. (2.19), the integration is intractable, as both the prior and the posterior distributions are over high dimensional random variables and cannot be computed in closed-form.

Eq. (2.19) can be approximated via a simple, but powerful, technique which involves the use of a *Monte Carlo* estimator, turning Eq. (2.19) into an expectation $p(\mathbf{y} \mid \mathbf{x}, \mathcal{D}) \approx \frac{1}{N} \sum_{i=1}^{N} p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_i), \boldsymbol{\theta}_i \sim p(\boldsymbol{\theta} \mid \mathcal{D})$.

However, computing the posterior Eq. (2.18), poses a more significant challenge. Monte Carlo Dropout (Gal and Ghahramani, 2016) connects training with dropouts and conducts forward passes over multiple dropout mask samples with variational inference. Other methods such as Deep Ensembles (Lakshminarayanan et al., 2017), consider the training of an ensemble of models as sampling from a posterior model. More recently, a wave of methods based on deep kernel learning has been proposed van Amersfoort et al. (2021a), Liu et al. (2020) which allows for uncertainty estimation using a single model. Finally, methods such as Lahlou et al. (2021) focus on predicting the epistemic uncertainty directly.

Bayesian Causal Discovery

A common assumption in causal inference is that causal relations are known qualitatively and can be represented by a DAG. While this qualitative information can be obtained from domain knowledge in some scenarios, it's infeasible in most applications. The goal of causal discovery is to recover the SCM given a dataset $\mathcal{D}$. In general, without further assumptions about the nature of mechanisms f (e.g., linear vs. nonlinear), the true SCM may not be *identifiable* (Peters et al., 2012) from observational data alone. This non-identifiability is because there could be multiple DAGs (and hence multiple factorizations of $P(\mathbf{X_V})$) which explain the data equally well. Such DAGs are said to be *Markov Equivalent*. Interventions can improve identifiability. In addition to identifiability issues, estimating the functional relationships between nodes using finite data is another source of uncertainty. Bayesian parameter estimation over the unknown SCM provides a principled way to quantify these uncertainties and obtain a posterior distribution over the SCM given observational data. An experimenter can then use the knowledge encoded by the posterior to design informative experiments that efficiently *acquire interventional data to resolve unknown edge orientations and functional uncertainty.*

Posterior Distributions over SCMs. The key challenge in obtaining posteriors over SCMs is that the space of DAGs is discrete and superexponential in the number of variables (Peters et al., 2017). However, recent techniques based on variational inference (Annadani et al., 2021, Lorch et al., 2021, Cundy et al., 2021) provide a tractable and scalable way of approximating this posterior. Given a tractable approximation of the posterior over the parameters $\psi \in \Psi$, variational inference optimizes a lower bound on the (log-) evidence:

$$\log p(\mathcal{D}) \geq \mathcal{L}(\psi \in \Psi) = \mathop{\mathbb{E}}_{q_\psi(\boldsymbol{\phi})} [\log p(\mathcal{D} \mid \boldsymbol{\phi})] - \mathrm{D}_{\mathrm{KL}}(q_\psi(\boldsymbol{\phi}) || p(\boldsymbol{\phi}))$$

The key idea in these techniques is the way the variational family Ψ for SCMs, in particular over DAGs, is parameterized. The variational family for the Variational Causal Network (VCN) method (Annadani et al., 2021) is an autoregres-

sive Bernoulli distribution over the adjacency matrix. They further enforce the acyclicity constraint (Zheng et al., 2018) through the prior. BCD-Nets (Cundy et al., 2021) consider a distribution over node orderings through a Boltzmann distribution and perform inference with Gumbel-Sinkhorn (Mena et al., 2018) operator. DiBS (Lorch et al., 2021) considers latent variables over entries of the adjacency matrix and performs inference over these latent variables using SVGD (Liu and Wang, 2016). Finally, Deleu et al. (2022), Nishikawa-Toomey et al. (2022) proposes the use of a GFlowNet (Bengio et al., 2021) whose states are DAGs, and where the dag is created sequentially by adding one edge at a time, checking the acyclicity constraint at each step.

Model-based decision-making under uncertainty.

Models have served complex decision-making for hundreds of years. Modeling physical phenomena via, often approximating, equations has been the study subject of physics. Decision-making with a model takes the form of future planning, imagining different actualizations of the world and deciding accordingly. With the advancements of machine learning, and lately deep learning, models took the form of parametric functions that can be trained to "match" a desired function (or rather, samples from a desired function). These "surrogate" models are distilled versions of the dataset aiming to simulate the true function that generated the data.

Powerful models such as deep neural networks have been used in decision-making in various domains, such as robotics, games, autonomous driving, personalized healthcare, and architecture, among other fields. In this section, we will focus on parametric models such as deep neural networks, and their Bayesian extensions along with a probabilistic or graphical framework for capturing different uncertainties involved in model-based decision making.

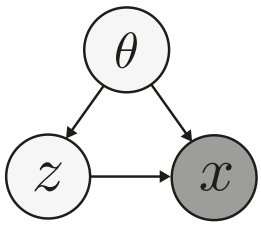


Figure 2.8: Graphical model of a simple inference task. In shaded gray, we depict the observable variable x and by θ (parametrization of the world) and z (the concept) the hidden variables.

Perception as inference

Von Helmholtz (1867) viewed the human perceptual system as an inference engine where the brain is trying to infer the causes of the sensory input. Helmhotz machines (Dayan et al., 1995), variational inference (Jordan et al., 1999) and variational autoencoders (Kingma and Welling, 2014) turn the intractable problem of inference into a tractable approximation. In Fig. 2.8, we describe a simple inference problem where an observation x (e.g. the image of the chair) is caused by some latent cause z (e.g. the presence of the char) and a world parametrization θ (e.g. the laws of physics that turn the presence of a char into an observation). For the moment, we don't care about θ (although we can be Bayesian about θ as well and infer it) but we focus on z. *Perception as inference* suggests that the perceptual system is computing the posterior $p(z \mid x, \theta) = \frac{p(x|z,\theta)p(z|\theta)}{p(x|\theta)}$.

Planning as Inference

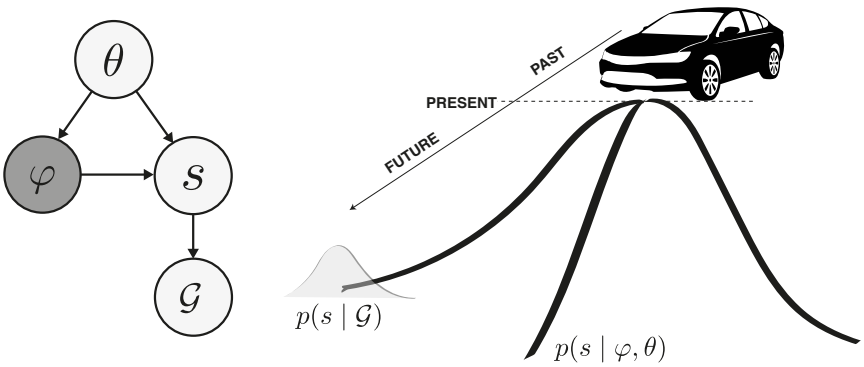


Figure 2.9: Graphical model of a toy planning as inference task.

Perception as inference belongs to passive behaviour. However, we can also include preferences, goals and utilities in the inference problem. In Fig. 2.11 we depict a graphical model of a toy decision-making problem. The observation ϕ takes the form of a sensory signal (e.g. RGB camera, LIDAR) and the state random variable s is representing the physical state of the agent (e.g. position in space). The goal of the agent, expressing desired states, is depicted as $\mathcal{G}$. By connection $s \rightarrow \mathcal{G}$ we represent the relationship "is state s optimal according to some goal definition?" or equally "is state s optimal?". This can then be expressed probabilistically as $p(s \mid \mathcal{G})$. For example, if $\mathcal{G}$ ("desired state of the agent") and s are represented by spatial coordinates, we could define $p(s \mid \mathcal{G} = g)$ as $\mathcal{N}(s; \mu = g, \sigma)$, thus turning the optimality criterion to a likelihood statement. This technique has been explored by (Rhinehart et al., 2020, 2019b) and is the building block of our work in Chapter 3.

Open-loop vs Closed-loop control

Casting the problem of planning as inference allows for uncertainty-aware planning, however, we will require more terminology to describe the bigger picture of action-perception loops. **Closed-loop control**, is selecting the actions as a response to the feedback received from the world (an observation) (Fig. 2.10.A). **Open-loop control** on the other hand, is selecting actions without interacting with the world – without world feedback. We can see planning as inference as an open-loop controller where the plans are selected using "imaginary" experiences. As depicted in Fig. 2.10.B, the agent is receiving feedback from the environment o_t and then imagines a sequence of actions $a_t, a_{t+1}, \ldots$. Then it proceeds with executing the sequence of actions $a_t, a_{t+1}, \ldots$ without waiting for feedback.

Bayesian Decision Theory

Decision making assumes a utility function $U(a)$ that defines an ordering between actions $a \in A$ according some optimality criteria. As the Bayesian framework gave us a methodology for inferring a posterior model $p(\theta \mid \mathcal{D})$ assuming a dataset

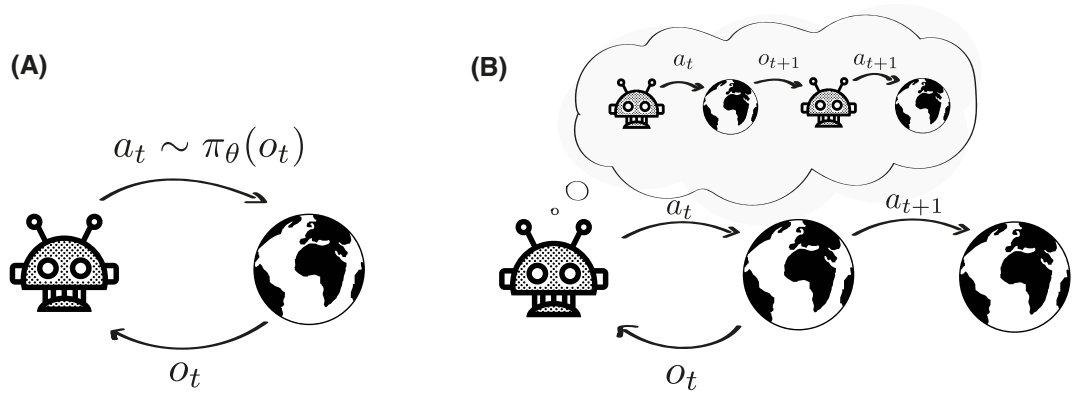


Figure 2.10: Closed-loop vs open-loop control.

$\mathcal{D}$ and a prior $p(\theta)$ over models, Bayesian decision theory follows the principle of maximization of the expected utility where the expectation is taken with respect to different hypotheses sampled from a posterior model $p(\Theta \mid \mathcal{D}$. The Bayes Optimal decision criterion is defined as $a^* = \text{argmax}_{a \in A} \mathbb{E}_{p(\theta|\mathcal{D})}[U(a, \theta)]$. However, aggregating over posterior samples with different aggregators can offer different decision-making properties. A general recipe for *uncertainty-aware* decision-making is to aggregate over the posterior distribution when selecting the optimal action

$$a^* = \underset{a \in A}{\text{argmax}} \oplus_{p(\theta|\mathcal{D})}[U(a, \theta)], \tag{2.20}$$

where $\oplus$ can be different aggregation operators, each inducing a different decision-making behaviour.

Robust agency

(Wald, 1939) formulated decision-making as a game where the decision-making agent is in an adversarial game against the objective world (nature). In this game, the agent plays first by selecting decisions $a \in A$ and world (Nature), in response, selects the worst θ associated with this decision. The agent's objective is then to select $a \in A$ that yields the best payoff. Concretely, Wald's maxmin rule turns Eq. (2.20) into

$$a^* = \underset{a \in A}{\text{argmax}} \min_{p(\theta|\mathcal{D})} [U(a, \theta)]. \tag{2.21}$$

This decision rule represents what is called "pessimism in the face of uncertainty" in decision-making because it assumes the (subjective) worst-case scenario when making a decision. As we will see in Chapter 3, this principle can help the agent be robust under uncertainty that can rise because of distribution shifts and other sources of epistemic uncertainty.

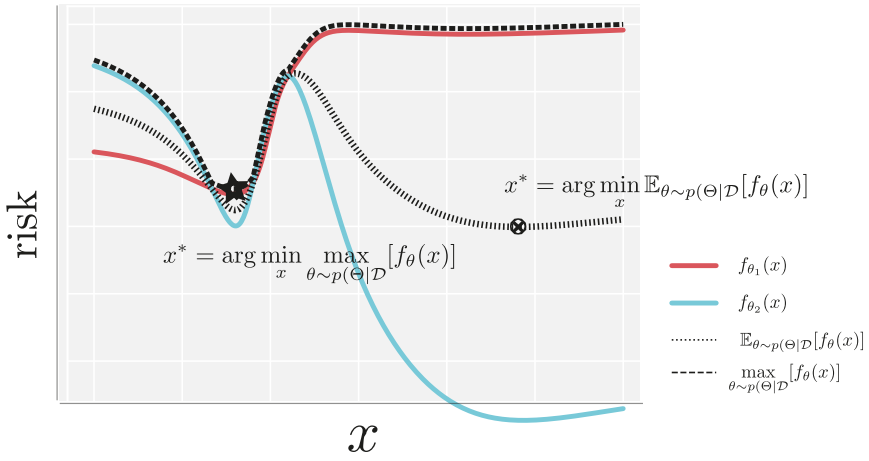


Figure 2.11: Bayesian Decision-Making under different aggregation operations. If we used the Bayes Optimal criterion then we would risk suffering a great loss in the worst-case scenario. However, the min max criterion selects a decision x, which although on expectation is worse than the Bayes optimal decision, there is no risk associated with that decision (the worst-case scenario is minimized).

Epistemic Agency

Episteme is the Greek word for science, knowledge, or understanding. By epistemic agency, we call the ability to act in the world in order to acquire knowledge about the world. The epistemic agency is a key component of empirical sciences and is the ability that allows us to learn about the world and to make decisions based on this knowledge. In this section, we will explore the role of epistemic agency in decision-making, and how it can be used to improve decision-making systems. Box (1980) defines the scientific method as *"... a process of guided learning in which accelerated acquisition of knowledge relevant to some question under investigation is achieved by a hierarchy of iterations in which induction and deduction are used in alternation"*. The scientific method becomes an iterative process where through

hypothetico-deductive reasoning, produces tentative theories of truth (*hypotheses*) and then through feedback (*experiments*) identifies the discrepancies between the theory and the praxis (*action*). We can formulate epistemic agency – the iterative process of producing knowledge – using the field known as *Bayesian Optimal Experimental Design* (Lindley, 1956).

Bayesian Optimal Experimental Design

Bayesian Optimal Experimental Design (BOED) (Lindley, 1956, Chaloner and Verdinelli, 1995) is an information theoretic approach to the problem of selecting the optimal experiment to estimate any parameter θ. For BOED, the *utility* of the experiment ξ is the expected information gain (EIG):

$$\mathrm{U}_{\mathrm{BOED}}(\xi) \triangleq \mathrm{I}(\mathbf{Y};\theta \mid \xi,\mathcal{D}) = \mathop{\mathbb{E}}_{p(\mathbf{y}|\theta,\xi)p(\theta|\mathcal{D})}[\log p(\mathbf{y} \mid \xi,\mathcal{D}) - \log p(\mathbf{y} \mid \theta,\xi,\mathcal{D})] \quad (2.22)$$

$$= \mathop{\mathbb{E}}_{p(\mathbf{y}|\theta,\xi)p(\theta|\mathcal{D})}[\log p(\theta \mid \xi,\mathcal{D}) - \log p(\theta \mid \mathbf{y},\xi,\mathcal{D})] \quad (2.23)$$

, where $\theta \in \Theta$ denotes the parameter we want to learn and Θ the parameterer space of consideration. Interestingly, this is also the mutual information (MI) between the observation $\mathbf{y}$ and θ. The goal of BOED is to select the experiment that maximizes this objective $\xi^* = \mathrm{argmax}_\xi \, \mathrm{U}_{\mathrm{BOED}}(\xi)$.

A common setting, called *static*, *fixed* or *batch* design, is to plan multiple designs $\{\xi_1, \ldots, \xi_\mathcal{B}\}$ in an open loop fashion. In contrast, *adaptive* or *sequential* design is the setting where the designs are planned after the incorporation of the results of previous experimental rounds, adapting in a closed-loop fashion to the feedback. In practice, a mix between batch and adaptive designs is used (i.e. executing T experiments, gathering the response from nature, updating the posterior (adaption step) and proposing T new experiments as a response).

One of the main challenges of computing expected information gain in Eq. (2.22) is computing the expectations which form a nested and doubly intractable quantity (Rainforth et al., 2018b). Crucially, such expectations can be estimated via Monte Carlo methods resulting on tractable approximations, such as Nested Monte Carlo (NMC) (Rainforth et al., 2018a) and Prior Contrastive Estimator (Foster et al., 2019).

Bayesian Active Learning

Active learning is concerned with the acquisition of data from which learning can be maximized (Cohn et al., 1994, Settles, 2009). Usually, active learning assumes the existence of a pool dataset $\mathcal{D}_{\text{pool}}$ for which the labels are expensive to acquire, thus we need to decide which examples from the dataset to label. For example, such a dataset could be medical records for which acquiring labels might imply hiring expensive medical consultants. Bayesian Active Learning is using the Bayesian ingredients for casting learning as an inference problem and the acquisition of informative examples, as the problem of selecting observations that maximizing posterior reduction.

Algorithm 1: Greedy batch BALD algorithm.

Input : acquisition size b, pool dataset $\mathcal{D}_{\text{pool}}$, posterior model $p(\theta \mid \mathcal{D}_{\text{train}})$

$S \leftarrow \emptyset$

for *iteration* $t = 1 \dots b$ **do**

  ▷ `Select example that maximizes the acquisition function`

  $x_t \leftarrow \text{argmax}_{x \in \mathcal{D}_{\text{pool}}} \alpha(A \cup \{x\}, p(\theta \mid \mathcal{D}_{\text{train}}))$

  ▷ `and add to batch`

  $S \leftarrow S \cup x_t$

Output : batch S

A widely adopted acquisition function is Bayesian Active Learning by Disagreement (BALD), introduced by Houlsby et al. (2011), according to which, the example that maximizes the Expected Information Gain (EIG) $\mathrm{I}(\theta; \mathbf{Y} \mid x, \mathcal{D})$ is selected. The key observation BALD objective brought to the Bayesian Active Learning community was that the symmetry of EIG objective $\mathrm{I}(\theta; \mathbf{Y} \mid x, \mathcal{D}) = \mathrm{I}(\mathbf{Y}; \theta \mid x, \mathcal{D})$, allowed for a factorization similar to Eq. (2.22), which turned the estimation of the objective to the tractable version:

$$\alpha_{\text{bald}}(x, p(\Theta \mid \mathcal{D}_{\text{train}})) \tag{2.24}$$

$$= \mathrm{I}(\theta; \mathbf{Y} \mid x, \mathcal{D}) \tag{2.25}$$

$$= \mathop{\mathbb{E}}_{p(\mathbf{y}|x,\theta)p(\theta|\mathcal{D}_{\text{train}})} [\log p(\mathbf{y} \mid x, \mathcal{D}) - \log p(\mathbf{y} \mid x, \theta, \mathcal{D})] \tag{2.26}$$

Often, this active learning is concerned with acquiring examples in batches. In Kirsch et al. (2019) the authors extended BALD to the batch setting, devising a simple greedy algorithm for acquiring batches and by employing the result by Nemhauser et al. (1978) and the fact that *Expected Information Gain* is submodular and non-monotonic function they showed that such an algorithm is $1 - \frac{1}{e}$-optimal.

Part I

Pessimism in the face of uncertainty

3
Robust Imitative Planning

The chapter is based on the following published body of work:

1. Filos, Angelos*,**Tigas, Panagiotis***, McAllister, Rowan, Rhinehart, Nick, Levine, Sergey, Gal, Yarin. *"Can autonomous vehicles identify, recover from, and adapt to distribution shifts?"*. In International Conference on Machine Learning (pp. 3145-3153), PMLR, 2020.
2. Andrey Malinin, Neil Band, Ganshin, Alexander, German Chesnokov, Yarin Gal, Mark J. F. Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Roginskiy, Denis, Mariya Shmatova, **Panagiotis Tigas**, Boris Yangel. *"Shifts: A dataset of real distributional shift across multiple large-scale tasks"*. In Neural Information Processing Systems 34, Datasets and Benchmarks Track, 2021.

Introduction

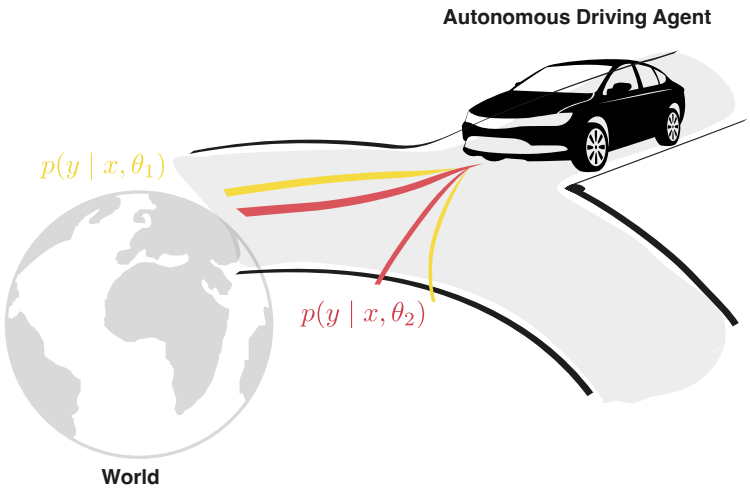


Figure 3.1: Autonomous Driving Agent interacting with the World. In this chapter, the Bayesian agent of consideration is an autonomous driving agent that maintains several hypotheses regarding the future trajectory. The induced trajectories can be catastrophic (off the road) or safe (on the road). We devise different strategies for the agent to drive robustly and safely under the various uncertainties involved.

Autonomous agents hold the promise of systematizing decision-making to reduce catastrophes due to human mistakes. Recent advances in machine learning (ML) enable the deployment of such agents in challenging, real-world, safety-critical domains, such as autonomous driving (AD) in urban areas. However, it has been repeatedly demonstrated that the reliability of ML models degrades radically when they are exposed to novel settings (i.e., *under a shift away from the distribution of observations seen during their training*) due to their failure to generalise, leading to catastrophic outcomes (Sugiyama and Kawanabe, 2012, Amodei et al., 2016, Snoek et al., 2019). The diminishing performance of ML models to out-of-training distribution (OOD) regimes is concerning in life-critical applications, such as AD (Quionero-Candela et al., 2009, Leike et al., 2017).

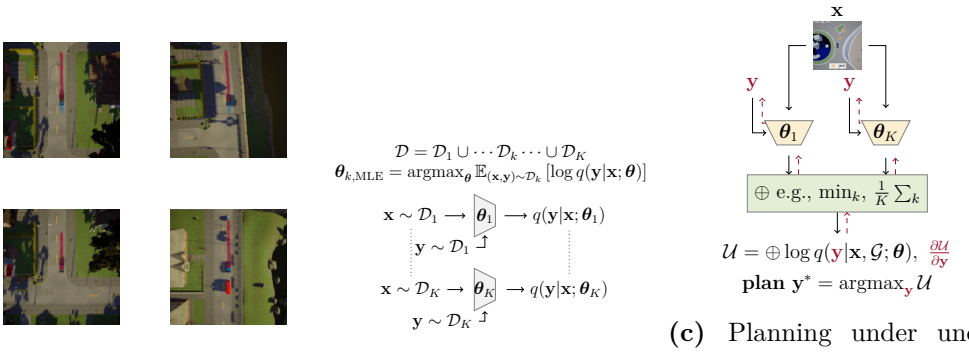
Although there are relatively simple strategies (e.g., stay within the lane boundaries, avoid other cars and pedestrians) that generalise, perception-based, end-to-end approaches, while flexible, they are also susceptible to spurious correlations.

Therefore, they can pick up non-causal features that lead to confusion in OOD scenes (de Haan et al., 2019).

Due to the complexity of the real-world and its ever-changing dynamics, the deployed agents inevitably face novel situations and should be able to cope with them, to at least (a) identify and ideally (b) recover from them, without failing catastrophically. These desiderata are *not* captured by the existing benchmarks (Ros et al., 2019, Codevilla et al., 2019) and as a consequence, are *not* satisfied by the current state-of-the-art methods (Chen et al., 2019, Tang et al., 2019, Rhinehart et al., 2020), which are prone to fail in unpredictable ways when they experience OOD scenarios (depicted in Figure 3.3 and empirically verified in Section 3).

In this paper, we demonstrate the practical importance of OOD detection in AD and its importance for safety. The key contributions are summarised as follows:

1. **Epistemic uncertainty-aware planning:** We present an epistemic uncertainty-aware planning method, called *robust imitative planning* (RIP) for detecting and recovering from distribution shifts. Simple quantification of epistemic uncertainty with deep ensembles enables detection of distribution shifts. By employing Bayesian decision theory and robust control objectives, we show how we can act conservatively in unfamiliar states which often allows us to recover from distribution shifts (didactic example depicted in Figure 3.3).
2. **Uncertainty-driven online adaptation:** Our adaptive, online method, called *adaptive robust imitative planning* (AdaRIP), uses RIP's epistemic uncertainty estimates to efficiently query the expert for feedback which is used to adapt on-the-fly, without compromising safety. Therefore, AdaRIP could be deployed in the real world: it can reason about what it does not know and in these cases ask for human guidance to guarantee current safety and enhance future performance.
3. **Autonomous car novel-scene benchmark:** We introduce an autonomous car novel-scene benchmark, called `CARNOVEL`, to assess the robustness of AD methods to a suite of out-of-distribution tasks. In particular, we evaluate



(a) Demonstrations **(b)** Ensemble training **(c)** Planning under uncertainty

Figure 3.2: The robust imitative planning (RIP) framework. **(a) Expert demonstrations**. We assume access to observations $\mathbf{x}$ and expert state $\mathbf{y}$ pairs, collected either in simulation (Dosovitskiy et al., 2017) or in real-world (Caesar et al., 2019, Sun et al., 2019, Kesten et al., 2019). **(b) Learning algorithm (cf. Section 3)**. We capture epistemic model uncertainty by training an ensemble of density estimators $\{q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_k)\}_{k=1}^K$, via maximum likelihood. Other approximate Bayesian deep learning methods (Gal and Ghahramani, 2016) are also tested. **(c) Planning paradigm (cf. Section 3)**. The epistemic uncertainty is taken into account at planning via the aggregation operator $\oplus$ (e.g., $\min_k$), and the optimal plan $\mathbf{y}^*$ is calculated online with gradient-based optimization *through the learned likelihood models*.

them in terms of their ability to: (a) detect OOD events, measured by the correlation of infractions and model uncertainty; (b) recover from distribution shifts, quantified by the percentage of successful manoeuvres in novel scenes and (c) efficiently adapt to OOD scenarios, provided online supervision.

Problem Setting and Notation

We consider sequential decision-making in safety-critical domains. A method is considered safe when it is accurate, with respect to some metric (cf. Sections 3, 14), and certain.

Assumption 1 (Expert demonstrations)**.** *We assume access to a dataset,* $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^N$*, of time-profiled expert trajectories (i.e., plans),* $\mathbf{y}$*, paired with high-dimensional observations,* $\mathbf{x}$*, of the corresponding scenes. The trajectories are drawn from the expert policy,* $\mathbf{y} \sim \pi_{expert}(\cdot|\mathbf{x})$.

Our goal is to approximate the (i.e., near-optimal) unknown expert policy, π_{expert}, using imitation learning (Widrow and Smith, 1964, Pomerleau, 1989, IL), based

only on the demonstrations, $\mathcal{D}$. For simplicity, we also make the following assumptions, common in the autonomous driving and robotics literature (Rhinehart et al., 2020, Du et al., 2019).

Assumption 2 (Inverse dynamics)**.** *We assume access to an inverse dynamics model (Bellman, 2015, PID controller, $\mathbb{I}$), which performs the low-level control – inverse planning – a_t (i.e., steering, braking and throttling), provided the current and next states (i.e., positions), s_t and s_{t+1}, respectively. Therefore, we can operate directly on state-only trajectories, $\mathbf{y} = (s_1, \dots, s_T)$, where the actions are determined by the local planner, $a_t = \mathbb{I}(s_t, s_{t+1}),\ \forall t = 1, \dots, T-1$.*

Assumption 3 (Global planner)**.** *We assume access to a global navigation system that we can use to specify high-level goal locations $\mathcal{G}$ or/and commands $\mathcal{C}$ (e.g., turn left/right at the intersection, take the second exit).*

Assumption 4 (Perfect localization)**.** *We consider the provided locations (e.g., goal, ego-vehicle positions) as accurate, i.e., filtered by a localization system.*

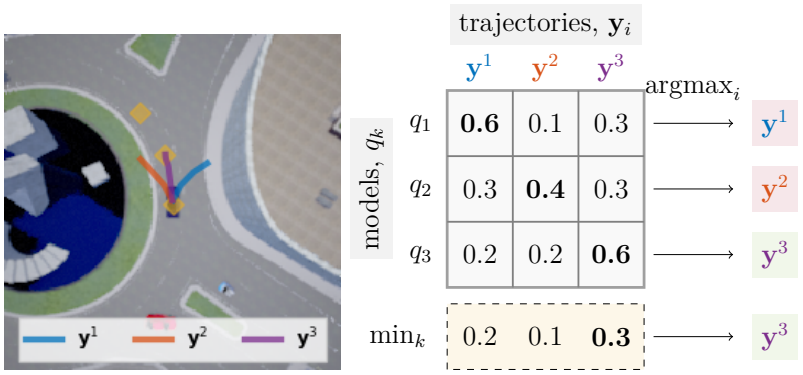
These are benign assumptions for many applications in robotics. If required, these quantities can also be learned from data, and are typically easier to learn than π_{expert}.

Robust Imitative Planning

We seek an imitation learning method that (a) provides a distribution over expert plans; (b) quantifies epistemic uncertainty to allow for detection of OOD observations and (c) enables robustness to distribution shift with an explicit mechanism for recovery. Our method is shown in Figure 3.2. First, we present the model used for imitating the expert.

Bayesian Imitative Model

We perform context-conditioned density estimation of the distribution over future expert trajectories (i.e., plans), using a probabilistic "imitative" model $q(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$,



(a) OOD driving scenario **(b)** Robust imitative planning

Figure 3.3: Didactic example: (a) in a novel, out-of-training distribution (OOD) driving scenario, candidate plans/trajectories $\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3$ are (b) evaluated (row-wise) by an ensemble of expert-likelihood models q_1, q_2, q_3. Under models q_1 and q_2 the best plans are the catastrophic trajectories $\mathbf{y}^1$ and $\mathbf{y}^2$ respectively. Our epistemic uncertainty-aware robust (RIP) planning method aggregates the evaluations of the ensemble and proposes the safe plan $\mathbf{y}^3$. RIP considers the disagreement between the models and avoid overconfident but catastrophic extrapolations in OOD tasks.

trained via maximum likelihood estimation (MLE):

$$\boldsymbol{\theta}_{\text{MLE}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \left[\log q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})\right]. \tag{3.1}$$

Contrary to existing methods in AD (Rhinehart et al., 2020, Chen et al., 2019), we place a prior distribution $p(\boldsymbol{\theta})$ over possible model parameters $\boldsymbol{\theta}$, which induces a distribution over the density models $q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$. After observing data $\mathcal{D}$, the distribution over density models has a posterior $p(\boldsymbol{\theta}|\mathcal{D})$.

Practical implementation. We use an autoregressive neural density estimator (Rhinehart et al., 2018), depicted in Figure 3.2b, as the imitative model, parametrised by learnable parameters $\boldsymbol{\theta}$. The likelihood of a plan $\mathbf{y}$ in context $\mathbf{x}$ to come from an expert (i.e., *imitation prior*) is given by:

$$\begin{aligned} q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) &= \prod_{t=1}^{T} p(s_t|\mathbf{y}_{<t},\mathbf{x};\boldsymbol{\theta}) \\ &= \prod_{t=1}^{T} \mathcal{N}(s_t;\mu(\mathbf{y}_{<t},\mathbf{x};\boldsymbol{\theta}),\Sigma(\mathbf{y}_{<t},\mathbf{x};\boldsymbol{\theta})), \end{aligned} \tag{3.2}$$

where $\mu(\cdot;\boldsymbol{\theta})$ and $\Sigma(\cdot;\boldsymbol{\theta})$ are two heads of a recurrent neural network, with shared torso. We decompose the imitation prior as a telescopic product (cf. Eqn. (3.2)), where conditional densities are assumed normally distributed, and the distribution

parameters are learned (cf. Eqn. (3.1)). Despite the unimodality of normal distributions, the autoregression (i.e., sequential sampling of normal distributions where the future samples depend on the past) allows to model multi-modal distributions (Uria et al., 2016). Although more expressive alternatives exist, such as the mixture of density networks (Bishop, 1994) and normalising flows (Rezende and Mohamed, 2015b), we empirically find Eqn. (3.2) sufficient.

The estimation of the posterior of the model parameters, $p(\boldsymbol{\theta}|\mathcal{D})$, with exact inference is intractable for non-trivial models (Neal, 2012). We use ensembles of deep imitative models as a simple approximation to the posterior $p(\boldsymbol{\theta}|\mathcal{D})$. We consider an ensemble of K components, using $\boldsymbol{\theta}_k$ to refer to the parameters of our k-th model q_k, trained with via maximum likelihood (cf. Eqn. (3.1) and Figure 3.2b). However, any (approximate) inference method to recover the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ would be applicable. To that end, we also try Monte Carlo dropout (Gal and Ghahramani, 2016).

Detecting Distribution Shifts

The log-likelihood of a plan $\log q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ (i.e., *imitation prior*) is a proxy of the quality of a plan $\mathbf{y}$ in context $\mathbf{x}$ under model $\boldsymbol{\theta}$. We detect distribution shifts by looking at the disagreement of the qualities of a plan under models coming from the posterior, $p(\boldsymbol{\theta}|\mathcal{D})$. We use the variance of the imitation prior with respect to the model posterior, i.e.,

$$u(\mathbf{y}) \triangleq \mathrm{Var}_{p(\boldsymbol{\theta}|\mathcal{D})}\left[\log q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})\right] \tag{3.3}$$

to quantify the model disagreement: Plans at in-distribution scenes have low variance, but high variance in OOD scenes. We can efficiently calculate Eqn. (3.3) when we use ensembles, or Monte Carlo, sampling-based methods for $p(\boldsymbol{\theta}|\mathcal{D})$.

Having to commit to a decision, just the detection of distribution shifts via the quantification of epistemic uncertainty is insufficient for recovery. Next, we introduce an epistemic uncertainty-aware planning objective that allows for robustness to distribution shifts.

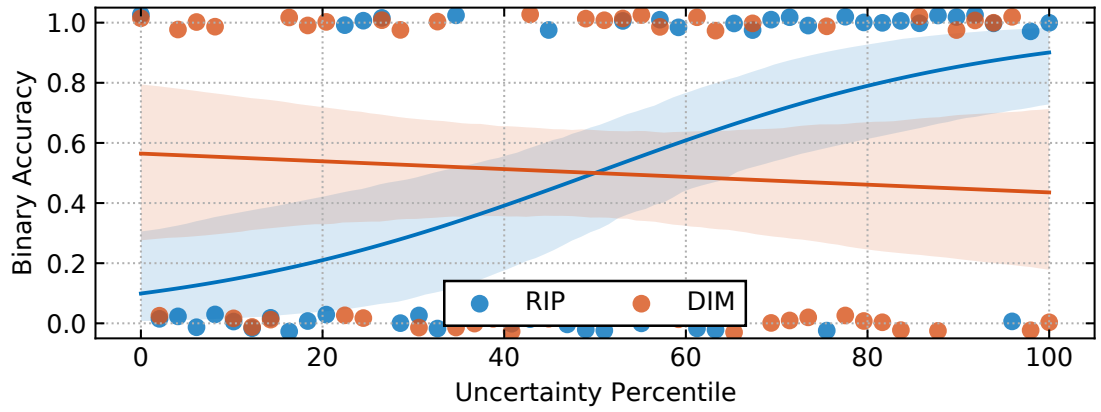


Figure 3.4: Uncertainty estimators as indicators of catastrophes on `CARNOVEL`. We collect 50 scenes for each model that led to a crash, record the uncertainty 4 seconds (Taoka, 1989) before the accident and assert if the uncertainties can be used for detection. RIP's (ours) predictive variance (in blue, cf. Eqn. (3.3)) serves as a useful detector, while DIM's (Rhinehart et al., 2020) negative log-likelihood (in orange) cannot be used for detecting catastrophes.

Planning Under Epistemic Uncertainty

We formulate planning to a goal location $\mathcal{G}$ under epistemic uncertainty, i.e., posterior over model parameters $p(\boldsymbol{\theta}|\mathcal{D})$, as the optimization (Barber, 2012) of the generic objective, which we term *robust imitative planning* (RIP):

$$\begin{aligned}\mathbf{y}^{\mathcal{G}}_{\text{RIP}} &\triangleq \underset{\mathbf{y}}{\text{argmax}} \overbrace{\bigoplus_{\boldsymbol{\theta}\in\text{supp}\left(p(\boldsymbol{\theta}|\mathcal{D})\right)}}^{\text{aggregation operator}} \log \underbrace{p(\mathbf{y}|\mathcal{G},\mathbf{x};\boldsymbol{\theta})}_{\text{imitation posterior}} \\ &\approx \underset{\mathbf{y}}{\text{argmax}} \bigoplus_{\boldsymbol{\theta}\in\text{supp}\left(p(\boldsymbol{\theta}|\mathcal{D})\right)} \log \underbrace{q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})}_{\text{imitation prior}} + \log \underbrace{p(\mathcal{G}|\mathbf{y})}_{\text{goal likelihood}} \quad , \end{aligned} \tag{3.4}$$

assuming the factorization $\log p(\mathbf{y}|\mathcal{G},\mathbf{x}) = \log p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) + \log p(\mathcal{G}|\mathbf{y}) - Z \approx \log q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) + \log p(\mathcal{G}|\mathbf{y}) - Z$, where Z is the normalizing constant $\log p(\mathcal{G}|\mathbf{x})$ (which can be omited in the maximization) and $q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ is the (approximate) imitation prior. The operator $\oplus$ (defined below) is applied on the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ and the goal-likelihood is given, for example, by a Gaussian centered at the final goal location $s^{\mathcal{G}}_T$ and a pre-specified tolerance ϵ, $p(\mathcal{G}|\mathbf{y}) = \mathcal{N}(\mathbf{y}_T; \mathbf{y}^{\mathcal{G}}_T, \epsilon^2 I)$.

Intuitively, we choose the plan $\mathbf{y}^{\mathcal{G}}_{\text{RIP}}$ that maximises the likelihood to have come from an expert demonstrator (i.e., "imitation prior") and is "close" to the

goal $\mathcal{G}$. The model posterior $p(\boldsymbol{\theta}|\mathcal{D})$ represents our belief (uncertainty) about the true expert model, having observed data $\mathcal{D}$ and from prior $p(\boldsymbol{\theta})$ and the aggregation operator $\oplus$ determines our level of awareness to uncertainty under a unified framework.

For example, a deep imitative model (Rhinehart et al., 2020) is a particular instance of the more general family of objectives described by Eqn. (3.4), where the operator $\oplus$ selects a single $\boldsymbol{\theta}_k$ from the posterior (point estimate). However, this approach is oblivious to the epistemic uncertainty and prone to fail in unfamiliar scenes (cf. Section 3).

In contrast, we focus our attention on two aggregation operators due to their favourable properties, which take epistemic uncertainty into account: (a) one inspired by robust control (Wald, 1939) which encourages pessimism in the face of uncertainty and (b) one inspired by Bayesian decision theory, which marginalises the epistemic uncertainty. To comply with utility theory notation, we will define the logarithm of the imitation prior $\log q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$ as the utility of the plan $\mathbf{y}$ under model $\boldsymbol{\theta}$, $U(\mathbf{y},\mathbf{x},\boldsymbol{\theta}) \triangleq \log q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$. Table 3.1 summarises the different operators considered in our experiments. Next, we motivate the used operators.

Worst Case Utility (RIP-WCU)

In the face of (epistemic) uncertainty, robust control (Wald, 1939) suggests to act pessimistically – reason about the *worst case scenario* and optimise it. All models with non-zero posterior probability $p(\boldsymbol{\theta}|\mathcal{D})$ are likely and hence our *robust imitative planning with respect to the worst case utility* (RIP-WCU) objective acts with respect to the most pessimistic model, i.e.,

$$s_{\text{RIP-WCU}} \triangleq \underset{\mathbf{y}}{\operatorname{argmax}} \underset{\boldsymbol{\theta}\in\operatorname{supp}\left(p(\boldsymbol{\theta}|\mathcal{D})\right)}{\min} U(\mathbf{y},\mathbf{x},\boldsymbol{\theta})\,. \tag{3.5}$$

The solution of the $\operatorname{argmax}_{\mathbf{y}} \min_{\boldsymbol{\theta}}$ optimization problem in Eqn. (3.5) is generally not tractable, but our deep ensemble approximation enables us to solve it by evaluating the minimum over a finite number of K models. The maximization over plans, $\mathbf{y}$, is solved with online gradient-based adaptive optimization, specifically

ADAM (Kingma and Ba, 2014). An alternative online planning method with a trajectory library (Liu and Atkeson, 2009) (c.f. Appendix A) is used too but its performance in OOD scenes is noticeably worse than online gradient descent.

Alternative, "softer" robust operators can be used instead of the minimum, including the Conditional Value at Risk (Embrechts et al., 2013, Rajeswaran et al., 2016, CVaR) that employs quantiles. CVaR may be more useful in cases of full support model posterior, where there may be a pessimistic but trivial model, for example, due to misspecification of the prior, $p(\boldsymbol{\theta})$, or due to the approximate inference procedure. Mean-variance optimization (Kahn et al., 2017, Kenton et al., 2019) can be also used, aiming to directly minimise the distribution shift metric, as defined in Eqn. (3.3).

Next, we present a different aggregator for epistemic uncertainty that is not as pessimistic as RIP-WCU and, as found empirically, works sufficiently well too.

Expected Utility (RIP-EU)

In the face of (epistemic) uncertainty, Bayesian decision theory (Barber, 2012) uses the predictive posterior (i.e., model averaging), which weights each model's contribution according to its posterior probability, i.e. Inspired by Bayesian decision theory, we propose to act optimistically using the *expected utility* criterion,

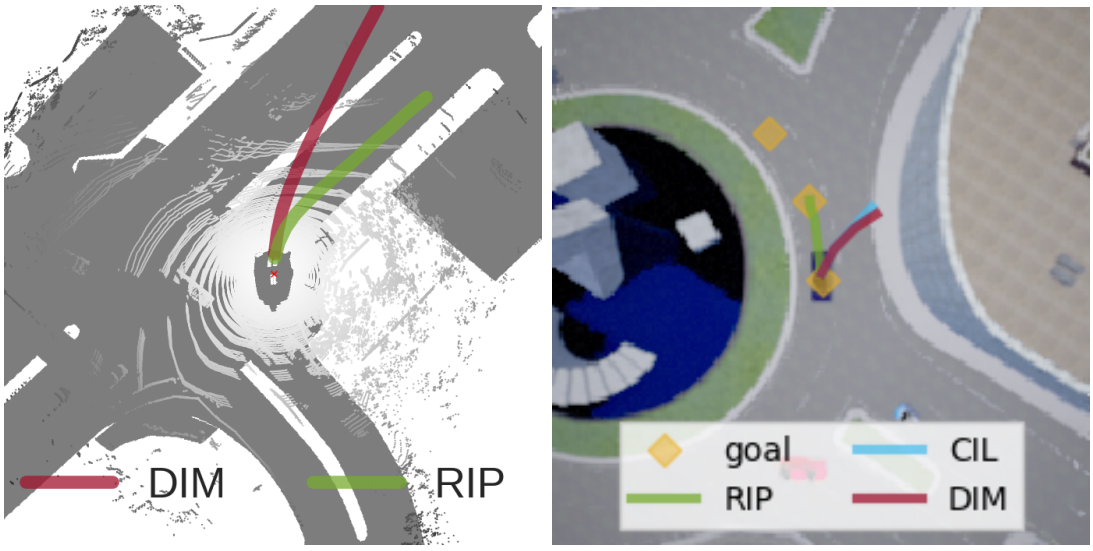
$$s_{\text{RIP-EU}} \triangleq \operatorname*{argmax}_{\mathbf{y}} \int p(\boldsymbol{\theta}|\mathcal{D}) U(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \,. \tag{3.6}$$

Despite the intractability of the exact integration, the ensemble approximation used allows us to efficiently estimate and optimise the objective. We call this method *robust imitative planning with expected utility* (RIP-EU), where the more likely utilities' impacts are up-weighted according to the predictive posterior.

From a multi-objective optimization point of view, we can interpret the log-likelihood, $\log q(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$, as the utility of a task $\boldsymbol{\theta}$, with importance $p(\boldsymbol{\theta}|\mathcal{D})$, given by the posterior density. Then RIP-EU in Eqn. (3.6) gives the Pareto efficient solution (Barber, 2012) for the tasks $\boldsymbol{\theta} \in \operatorname{supp}\Big(p(\boldsymbol{\theta}|\mathcal{D})\Big)$.

Table 3.1: Robust imitative planning (RIP) unified framework. The different aggregation operators applied on the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$, approximated with the deep ensemble (Lakshminarayanan et al., 2017) components $\boldsymbol{\theta}_k$.

Methods	Operator $\oplus$	Interpretation
Imitative Models	$\log q_{k=1}$	Sample
Best Case (RIP-BCU)	$\max_k \log q_k$	Max
Robust Imitative Planning (ours)		
Expected (RIP-EU)	$\sum_k \log q_k$	Geometric Mean
Worst Case (RIP-WCU)	$\min_k \log q_k$	Min



(a) nuScenes **(b)** CARNOVEL

Figure 3.5: RIP-EU (ours) robustness to OOD scenarios, compared to (Codevilla et al., 2018, CIL) and (Rhinehart et al., 2020, DIM).

Benchmarking Robustness to Novelty

We designed our experiments to answer the following questions: **Q1.** Can autonomous driving, imitation-learning, epistemic-uncertainty unaware methods detect distribution shifts? **Q2.** How robust are these methods under distribution shifts, i.e., can they recover? **Q3.** Does RIP's epistemic uncertainty quantification enable identification of novel scenes? **Q4.** Does RIP's explicit mechanism for recovery from distribution shifts lead to improved performance? To that end, we

Table 3.2: We evaluate different autonomous driving prediction methods in terms of their robustness to distribution scene, in the nuScenes ICRA 2020 challenge (Phan-Minh et al., 2019). We use the provided train–val–test splits and report performance on the test (i.e., out-of-sample) scenarios. A "♣" indicates methods that use LIDAR observation, as in (Rhinehart et al., 2019a), and a "♢" methods that use bird-view privileged information, as in (Phan-Minh et al., 2019). A "★" indicates that we used the results from the original paper, otherwise we used our implementation. Standard errors are in gray (via bootstrap sampling). The **outperforming** method is in bold.

	Boston			Singapore		
Methods	$\text{minADE}_1 \downarrow$ (2073 scenes, 50 samples, open-loop planning)	$\text{minADE}_5 \downarrow$	$\text{minFDE}_1 \downarrow$	$\text{minADE}_1 \downarrow$ (1189 scenes, 50 samples, open-loop planning)	$\text{minADE}_5 \downarrow$	$\text{minFDE}_1 \downarrow$
MTP$^{\diamondsuit\bigstar}$ (Cui et al., 2019)	4.13	3.24	9.23	4.13	3.24	9.23
MultiPath$^{\diamondsuit\bigstar}$ (Chai et al., 2019)	3.89	3.34	9.19	3.89	3.34	9.19
CoverNet$^{\diamondsuit\bigstar}$ (Phan-Minh et al., 2019)	3.87	2.41	9.26	3.87	2.41	9.26
DIM$^{\clubsuit}$ (Rhinehart et al., 2020)	3.64±0.05	2.48±0.02	8.22±0.13	3.82±0.04	2.95±0.01	8.91±0.08
RIP-BCU$^{\clubsuit}$ (baseline, cf. Table 3.1)	3.53±0.04	2.37±0.01	7.92±0.09	3.57±0.02	2.70±0.01	8.39±0.03
RIP-EU$^{\clubsuit}$ (ours, cf. Section 3)	3.39±0.03	2.33±0.01	7.62±0.07	3.48±0.01	2.69±0.02	8.19±0.02
RIP-WCU$^{\clubsuit}$ (ours, cf. Section 3)	**3.29**±0.03	**2.28**±0.00	**7.45**±0.05	**3.43**±0.01	**2.66**±0.01	**8.09**±0.04

conduct experiments both on real data, in Section 3, and on simulated scenarios, in Section 3, comparing our method (RIP) against current state-of-the-art driving methods.

nuScenes

We first compare our robust planning objectives (cf. Eqn. (3.5–3.6)) against existing state-of-the-art imitation learning methods in a prediction task (Phan-Minh et al., 2019), based on `nuScenes` (Caesar et al., 2019), the public, real-world, large-scale dataset for autonomous driving. Since we do not have control over the scenes split, we cannot guarantee that the evaluation is under distribution shifts, but only test out-of-sample performance, addressing question **Q4**.

Metrics

For fair comparison with the baselines, we use the metrics from the ICRA 2020 `nuScenes` prediction challenge.

Displacement error. The quality of a plan, $\mathbf{y}$, with respect to the ground truth prediction, $\mathbf{y}^*$ is measured by the average displacement error, i.e.,

$$\text{ADE}(\mathbf{y}) \triangleq \frac{1}{T} \sum_{t=1}^{T} \|s_t - s_t^*\|, \tag{3.7}$$

where $\mathbf{y} = (s_1, \dots, s_T)$. Stochastic models, such as our imitative model, $q(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$, can be evaluated based on their samples, using the minimum (over k samples) ADE (i.e., minADE_k), i.e.,

$$\text{minADE}_k(q) \triangleq \min_{\{\mathbf{y}^i\}_{i=1}^k \sim q(\mathbf{y}|\mathbf{x})} \text{ADE}(\mathbf{y}^i)\,. \tag{3.8}$$

In prior work, Phan-Minh et al. (2019) studied minADE_k for $k > 1$ in order to assess the quality of the generated samples from a model, q. Although we report minADE_k for $k = \{1, 5\}$, we are mostly interested in the decision-making (planning) task, where the driving agent commits to a *single* plan, $k = 1$. We also study the final displacement error (FDE), or equivalently minFDE_1, i.e.,

$$\text{minFDE}_1(\mathbf{y}) \triangleq \|s_T - s_T^*\|\,. \tag{3.9}$$

Baselines

We compare our contribution to state-of-the-art methods in the `nuScenes` dataset: the Multiple-Trajectory Prediction (Cui et al., 2019, MTP), MultiPath (Chai et al., 2019) and CoverNet (Phan-Minh et al., 2019), all of which score a (fixed) set of trajectories, i.e., trajectory library (Liu and Atkeson, 2009). Moreover, we implement the Deep Imitative Model (Rhinehart et al., 2020, DIM) and an optimistic variant of RIP, termed RIP-BCU and described in Table 3.1.

Results

We use the provided train-val-test splits from (Phan-Minh et al., 2019), for towns `Boston` and `Singapore`. For all methods we use $N = 50$ trajectories, and in case of both DIM and RIP, we only optimise the "imitation prior" (cf. Eqn. 3.4), since goals are not provided, running N planning procedures with different random initializations. The performance of the baselines and our methods are reported on Table 3.2. We can affirmatively answer **Q4** since RIP consistently outperforms the current state-of-the-art methods in out-of-sample evaluation. Moreover, **Q2** can be partially answered, since the epistemic-uncertainty-unaware baselines underperformed compared to RIP.

Table 3.3: We evaluate different autonomous driving methods in terms of their robustness to distribution shifts, in our new benchmark, `CARNOVEL`. All methods are trained on CARLA `Town01` using imitation learning on expert demonstrations from the autopilot (Dosovitskiy et al., 2017). A "†" indicates methods that use first-person camera view, as in (Chen et al., 2019), a "♣" methods that use LIDAR observation, as in (Rhinehart et al., 2020) and a "♢" methods that use the ground truth game engine state, as in (Chen et al., 2019). A "★" indicates that we used the reference implementation from the original paper, otherwise we used our implementation. For all the scenes we chose pairs of start-destination locations and ran 10 trials with randomized initial simulator state for each pair. Standard errors are in gray (via bootstrap sampling). The **outperforming** method is in bold.

	`AbnormalTurns`			`BusyTown`		
Methods	Success ↑ (7 × 10 scenes, %)	Infra/km ↓ (×1e−3)	Distance ↑ (m)	Success ↑ (11 × 10 scenes, %)	Infra/km ↓ (×1e−3)	Distance ↑ (m)
CIL♣★ (Codevilla et al., 2018)	65.71±07.37	07.04±05.07	128±020	05.45±06.35	11.49±03.66	217±033
LbC†★ (Chen et al., 2019)	00.00±00.00	05.81±00.58	208±004	20.00±13.48	03.96±00.15	374±016
LbC-GT♢★ (Chen et al., 2019)	02.86±06.39	**03.68**±00.34	217±033	65.45±07.60	02.59±00.02	400±006
DIM♣ (Rhinehart et al., 2020)	74.28±11.26	05.56±04.06	108±017	47.13±14.54	08.47±05.22	175±026
RIP-BCU♣ (baseline, cf. Table 3.1)	68.57±09.03	07.93±03.73	096±017	50.90±20.64	03.74±05.52	175±031
RIP-EU♣ (ours, cf. Section 3)	**84.28**±14.20	07.86±05.70	102±015	**64.54±23.25**	05.86±03.99	170±033
RIP-WCU♣ (ours, cf. Section 3)	**87.14**±14.20	**04.91**±03.60	102±021	**62.72**±05.16	**03.17**±02.04	167±021

	`Hills`			`Roundabouts`		
Methods	Success ↑ (4 × 10 scenes, %)	Infra/km ↓ (×1e−3)	Distance ↑ (m)	Success ↑ (5 × 10 scenes, %)	Infra/km ↓ (×1e−3)	Distance ↑ (m)
CIL♣★ (Codevilla et al., 2018)	60.00±29.34	04.74±03.02	219±034	20.00±00.00	**03.60**±03.23	269±021
LbC†★ (Chen et al., 2019)	50.00±00.00	01.61±00.15	541±101	08.00±10.95	03.70±00.72	323±043
LbC-GT♢★ (Chen et al., 2019)	05.00±11.18	03.36±00.26	312±020	00.00±00.00	06.47±00.99	123±018
DIM♣ (Rhinehart et al., 2020)	70.00±10.54	06.87±04.09	195±012	20.00±09.42	06.19±04.73	240±044
RIP-BCU♣ (baseline, cf. Table 3.1)	75.00±00.00	05.49±04.03	191±013	06.00±09.66	06.78±07.05	251±027
RIP-EU♣ (ours, cf. Section 3)	**97.50**±07.90	**00.26**±00.54	196±013	**38.00**±06.32	05.48±05.56	271±047
RIP-WCU♣ (ours, cf. Section 3)	**87.50**±13.17	**01.83**±01.73	191±006	**42.00**±06.32	04.32±01.91	217±030

Nonetheless, since we do not have full control over train and test splits at the ICRA 2020 challenge and hence we cannot introduce distribution shifts, we are not able to address questions **Q1** and **Q3** with the nuScenes benchmark. To that end, we now introduce a control benchmark based on the CARLA driving simulator (Dosovitskiy et al., 2017).

SHIFTS

Shifts dataset is based on the following paper:

Andrey Malinin, Neil Band, German Chesnokov, Yarin Gal, Mark JF Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Mariya Shmatova, **Panagiotis Tigas** and Boris Yangel. *"Shifts*

Dataset: Real-world distribution shifts in autonomous driving". In NeurIPS 2021 Track Datasets and Benchmarks.

Dataset The dataset for the Vehicle Motion Prediction task was collected by the Yandex Self-Driving Group (SDG) fleet and is the largest vehicle motion prediction dataset released to date, containing 600,000 scenes. These scenes span six locations, three seasons, three times of day, and four weather conditions. Each scene includes information about the state of dynamic objects and an HD map. Each scene is 10 seconds long and is divided into 5 seconds of context features and 5 seconds of ground truth targets for prediction, separated by the time $T = 0$. The goal is to predict the movement trajectory of vehicles at time $T \in (0, 5]$ based on the information available for time $T \in [-5, 0]$. The data contains training, development (`dev`) and evaluation (`eval`) sets. In order to study the effects of distributional shift, we partition the data such that the `dev` and `eval` sets have *in-domain* partitions which match the location and precipitation type of the training set, and *out-of-domain* or *shifted* partitions which do not match the training data along one or more of those axes. As in the other Shifts tasks, we define a canonical partitioning which is used throughout benchmarking.[1] The training set and in-domain partition of the `dev` and `eval` sets are taken from Moscow. Distributionally shifted `dev` data is taken from Skolkovo, Modiin, and Innopolis. Distributionally shifted `eval` data is taken from Tel Aviv and Ann Arbor. We also remove all cases of precipitation from the in-domain sets, while distributionally shifted datasets include precipitation. A full description of the dataset is available in Appendix B.

Metrics Here we consider five different performance metrics — minimum Average Displacement Error (minADE), minimum Final Displacement Error (minFDE), confidence-weighed ADE and FDE, and corrected Negative Log-Likelihood (cNLL). cNLL is a new metric we introduce that is particilarly well-suited for assessing how models handle multi-modal situations. The minimum or weighting is done across

[1] This partitioning is also the one used in the Shifts Challenge: `http://research.yandex.com/shifts`

up to 5 trajectories predicted by the baseline models. See Appendix B for detailed explanations of the metrics.

Baselines We consider two variants of Robust Imitative Planning (RIP) (Filos et al., 2020) as baselines. We use an ensemble of probabilistic models to stochastically generate multiple predictions for a given prediction request. Predictions are aggregated across ensemble members to compute the expected utility (EU). We consider a simple RNN-based behavioral cloning network (RIP-BC) Codevilla et al. (2018) and autoregressive flow–based Deep Imitative Model (RIP-DIM) Rhinehart et al. (2020) as backbone models. We adapt RIP to produce uncertainty estimates at two levels of granularity: per-trajectory and per–prediction request. Finally, we vary the number of ensemble members $K \in \{1, 3, 5\}$ and the uncertainty estimation method between Deep Ensembles (Lakshminarayanan et al., 2017) and Monte-Carlo Dropout (Gal and Ghahramani, 2016). See Appendix B for details on RIP, uncertainty estimation methods, backbone models, experimental setup, and full results. Additional results using Dropout Ensembles are provided in Appendix B.

Table 3.4: Predictive performance of BC & DIM RIP on in-domain, shifted, and full `dev` & `eval` data.

Dataset	Model	cNLL ↓			minADE ↓			weightedADE ↓			minFDE ↓			weightedFDE ↓		
		In	Shifted	Full	In	Shifted	Full	In	Shifted	Full	In	Shifted	Full	In	Shifted	Full
`Dev`	BC, MA, K=1	59.64	98.54	64.29	0.818	0.960	0.835	1.088	1.245	1.107	1.718	2.113	1.765	2.368	2.777	2.417
	BC, MA, K=5	56.86	91.54	61.01	0.765	0.887	0.779	**1.012**	**1.133**	**1.026**	1.617	1.976	1.660	**2.210**	**2.551**	**2.251**
	DIM, MA, K=1	**50.66**	73.00	**53.34**	0.750	0.818	0.758	1.523	1.583	1.530	1.497	1.720	1.524	3.472	3.639	3.492
	DIM, MA, K=5	50.85	**72.45**	53.43	**0.719**	**0.786**	**0.727**	1.399	1.469	1.408	**1.482**	**1.698**	**1.508**	3.202	3.393	3.225
`Eval`	BC, MA, K=1	60.20	98.82	67.93	0.829	1.084	0.880	1.104	1.407	1.164	1.733	2.420	1.870	2.394	3.197	2.555
	BC, MA, K=5	57.75	95.00	65.20	0.777	1.014	0.824	**1.028**	**1.299**	**1.082**	1.636	2.278	1.765	**2.238**	**2.957**	**2.382**
	DIM, MA, K=1	**50.50**	**76.00**	**55.60**	0.759	0.942	0.796	1.551	1.883	1.618	1.511	**1.983**	1.605	3.536	4.376	3.704
	DIM, MA, K=5	51.19	78.85	56.73	**0.728**	**0.918**	**0.766**	1.424	1.754	1.490	**1.493**	2.000	**1.595**	3.256	4.093	3.424

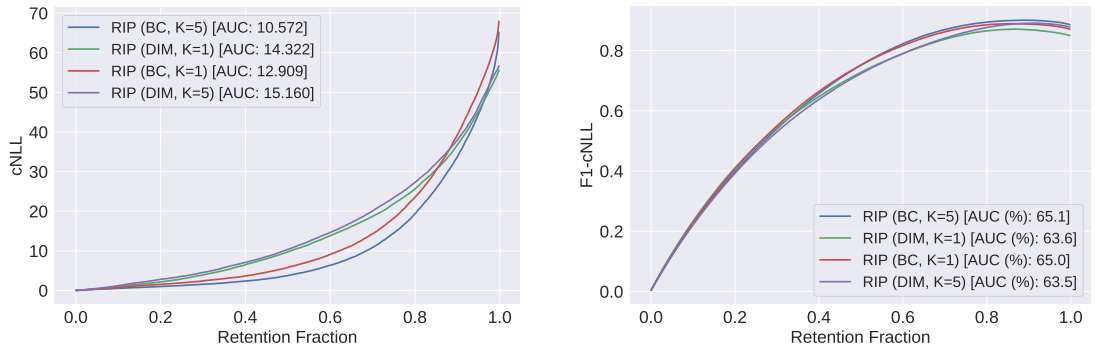
Predictive performance results for the RIP variants are presented in Table 3.4. Performance is assessed on the in-distribution (In), distributionally shifted (Shifted), and combined (Full) `dev` and `eval` datasets. We observe that across all model configurations, performance on the shifted data is worse than that on the in-distribution data. We also observe that RIP-BC consistently outperforms RIP-DIM on the per-trajectory confidence weighted metrics (weightedADE and weightedFDE), and RIP (DIM) outperforms RIP (BC) on minADE and minFDE. This

result might occur if DIM has higher predictive variance. In such a case, DIM might be more effective in modeling multimodality, and therefore would tend to produce at least one high accuracy trajectory on more scenes, improving performance on `min` aggregation metrics. This is supported by DIM models yielding the best cNLL, which is a metric particularly sensitive to correct treatment of multi-modal situations. In contrast, for "obvious" scenes, DIM might then produce unnecessarily complicated trajectories which would be reflected in poor performance on weightedADE.

Table 3.5: Uncertainty and robustness performance for motion prediction. The error metric for computing the area under the F1 curve (F1-AUC) and F1 at 95% retention rate (F1@95%) is **cNLL**.

Data	Ensemble Size (K)	R-AUC cNLL ↓		R-AUC weightedADE ↓		F1-AUC (%) ↑		F1@95% ↑		ROC-AUC (%) ↑	
		RIP-BC	RIP-DIM	RIP-BC	RIP-DIM	RIP-BC	RIP-DIM	RIP-BC	RIP-DIM	RIP-BC	RIP-DIM
`Dev`	1	11.22	12.86	0.268	0.419	65.1	63.8	89.3	87.4	51.0	**51.8**
	5	**9.08**	13.24	**0.236**	0.376	**65.2**	63.7	**90.6**	89.7	49.2	51.4
`Eval`	1	12.91	14.32	0.293	0.458	65.0	63.6	88.4	86.3	**52.8**	51.8
	5	**10.57**	15.16	**0.258**	0.411	**65.1**	63.5	**89.7**	88.9	52.1	50.9

Table 3.5 presents a joint evaluation of the uncertainty quantification and robustness of our baselines. We compute R-AUC with respect to cNLL and weightedADE, and the F1-AUC and F1@95% metrics with respect to the cNLL metric, as detailed in Appendix B. We observe that an ensemble of RIP-BC models outperforms RIP-DIM on these metrics. These results strongly suggest that RIP-BC has more informative uncertainty estimates than RIP-DIM, because RIP-BC achieves better



(a) cNLL error-retention. **(b)** cNLL F1-retention.

Figure 3.6: Retention curves for Vehicle Motion Prediction on full `eval` data.

R-AUC cNLL despite having greater overall error in terms of cNLL (in addition to minADE and minFDE). Figure Fig. 3.6 depicts, for cNLL, error- and F1-retention curves on the full `eval` dataset which reflect the trends observed in Table Table 3.5. Additionally, we find that across model configurations the per–prediction request uncertainty scores do not perform particularly well in detecting distribution shift (ROC-AUC). This may occur due to significant data uncertainty in all cases. Future work on detecting distributional shift on this dataset could, for example, inspect the distribution of log-likelihood scores on the in-distribution and shifted partitions in order to devise a metric for this task, aside from the uncertainty scores U used for the retention analysis.

CARNOVEL

In order to access the robustness of AD methods to novel, OOD driving scenarios, we introduce a benchmark, called `CARNOVEL`. In particular, `CARNOVEL` is built on the CARLA simulator (Dosovitskiy et al., 2017). Offline expert demonstrations[2] from `Town01` are provided for training. Then, the driving agents are evaluated on a suite of OOD navigation tasks, including but not limited to roundabouts, challenging non-right-angled turns and hills, none of which are experienced during training. The `CARNOVEL` tasks are summarised in Appendix A. Next, we introduce metrics that quantify and help us answer questions **Q1**, **Q3**.

Metrics

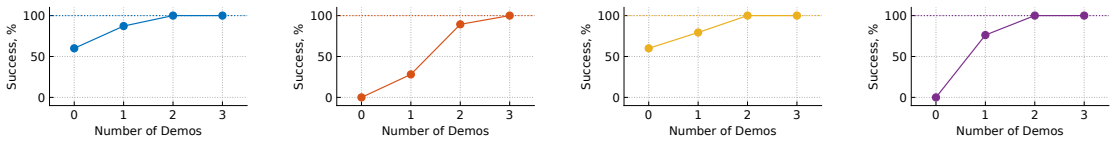
Since we are studying navigation tasks, agents should be able to *reach safely* pre-specified *destinations*. As done also in previous work (Codevilla et al., 2018, Rhinehart et al., 2020, Chen et al., 2019), the **infractions per kilometre** metric (i.e., violations of rules of the road and accidents per driven kilometre) measures how safely the agent navigates. The **success rate** measures the percentage of successful navigations to the destination, without any infraction. However, these standard metrics do not directly reflect the methods' performance under distribution shifts.

[2] using the CARLA rule-based autopilot (Dosovitskiy et al., 2017) without actuator noise.

As a result, we introduce two new metrics for quantifying the performance in out-of-training distribution tasks:

Detection score. The correlation of infractions and model's uncertainty termed *detection score* is used to measure a method's ability to predict the OOD scenes that lead to catastrophic events. As discussed by Michelmore et al. (2018), we look at time windows of 4 seconds (Taoka, 1989, Coley et al., 2009). A method that can detect potential infractions should have high detection score.

Recovery score. The percentage of successful manoeuvres in novel scenes — where the uncertainty-unaware methods fail — is used to quantify a method's ability to recover from distribution shifts. We refer to this metric as *recovery score*. A method that is oblivious to novelty should have 0 recovery score, but positive otherwise.



(a) `AbnormalTurns4` **(b)** `BusyTown2` **(c)** `Hills1` **(d)** `Roundabouts1`

Figure 3.7: Adaptation scores of AdaRIP (cf. Section 3) on `CARNOVEL` tasks that RIP-WCU and RIP-EU (cf. Section 3) do worst. We observe that as the number of online expert demonstrations increases, the success rate improves thanks to online model adaptation.

Baselines

We compare RIP against the current state-of-the-art imitation learning methods in the CARLA benchmark (Codevilla et al., 2018, Rhinehart et al., 2020, Chen et al., 2019). Apart from DIM and RIP-BCU, discussed in Section 3, we also benchmark:

Conditional imitation learning (Codevilla et al., 2018, CIL) is a discriminative behavioural cloning method that conditions its predictions on contextual information (e.g., LIDAR) and high-level commands (e.g., turn left, go straight).

Learning by cheating (Chen et al., 2019, LbC) is a method that builds on CIL and uses (cross-modal) distillation of privileged information (e.g., game state, rich, annotated bird-eye-view observations) to a sensorimotor agent. For reference, we

also evaluate the agent who has uses privileged information directly (i.e., teacher), which we term LbC-GT.

Results

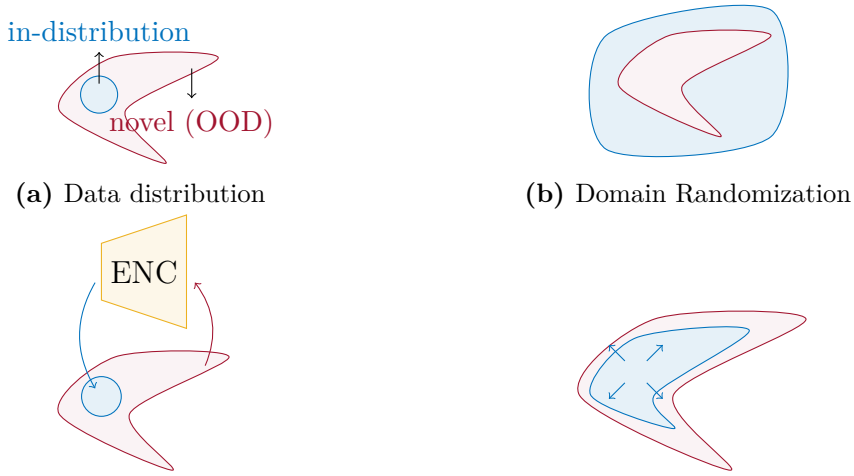
All the methods are trained on offline expert demonstrations from CARLA `Town01`. We perform 10 trials per `CARNOVEL` task with randomised initial simulator state and the results are reported on Table 3.3.

Our robust imitative planning (i.e., RIP-WCU and RIP-EU) consistently outperforms the current state-of-the-art imitation learning-based methods in novel, OOD driving scenarios. In alignment with the experimental results from `nuScenes` (cf. Section 3), we address questions **Q4** and **Q2**, reaching the conclusion that RIP's epistemic uncertainty explicit mechanism for recovery improves its performance under distribution shifts, compared to epistemic uncertainty-unaware methods. As a result, RIP's recovery score (cf. Section 3) is higher than the baselines.

Towards distribution shift detection and answering questions **Q1** and **Q3**, we collect 50 scenes for each method that led to a crash, record the uncertainty 4 seconds (Taoka, 1989) before the accident and assert if the uncertainties can be used for detection. RIP's (ours) predictive variance (cf. Eqn. (3.3)) serves as a useful detector, while DIM's (Rhinehart et al., 2020) negative log-likelihood was unable to detect catastrophes. The results are illustrated on Figure 3.4.

Adaptive Robust Imitative Planning

We empirically observe that the quantification of epistemic uncertainty and its use in the RIP objectives is not always sufficient to recover from shifts away from the training distribution (Chapter 3). However, we can use uncertainty estimates to ask the human driver to take back control or default to a safe policy, avoiding potential infractions. In the former case, the human driver's behaviors can be recorded and used to reduce RIP's epistemic uncertainty via online adaptation. The epistemic uncertainty is reducible and hence it can be eliminated, provided enough demonstrations.



(a) Data distribution **(b)** Domain Randomization

(c) Domain adaptation **(d)** Online adaptation

Figure 3.8: Common approaches to distribution shift, as in (a) there are novel (OOD) points that are outside the support of the training data: (b) domain randomization (e.g., Sadeghi and Levine (2016)) covers the data distribution by *exhaustively* sampling configurations from a simulator; (c) domain adaptation (e.g., McAllister et al. (2019)) projects (or encodes) the (OOD) points to the in-distribution space and (d) online adaptation (e.g., Ross et al. (2011)) progressively expands the in-distribution space by incorporating online, external feedback.

We propose an adaptive variant of RIP, called AdaRIP, which uses the epistemic uncertainty estimates to decide when to query the human driver for feedback, which is used to update its parameters *online*, adapting to arbitrary new driving scenarios. AdaRIP relies on external, online feedback from an expert demonstrator[3], similar to DAgger (Ross et al., 2011) and its variants (Zhang and Cho, 2016, Cronrath et al., 2018). However, unlike this prior work, AdaRIP uses an epistemic uncertainty-aware acquisition mechanism. AdaRIP's pseudocode is given in Algorithm 2.

The uncertainty (i.e., variance) threshold, τ, is calibrated on a validation dataset, such that it matches a pre-specified level of false negatives, using a similar analysis to Figure 3.4.

Benchmarking Adaptation

The goal of this section is to provide experimental evidence for answering the following questions: **Q5.** Can RIP's epistemic-uncertainty estimation be used for

[3] AdaRIP is also compatible with other feedback mechanisms, such as expert preferences (Christiano et al., 2017) or explicit reward functions (de Haan et al., 2019).

Algorithm 2: Adaptive Robust Imitative Planning

Input :

$\mathcal{D}$	Demonstrations	$\mathbb{I}(a_t \mid s_t, s_{t+1})$	Local planner
K	Number of models	$q(\mathbf{y}\mid\mathbf{x};\boldsymbol{\theta})$	Imitative model
$\mathcal{B}$	Data buffer	$p(\mathcal{G}\mid\mathbf{y})$	Goal likelihood
τ	Variance threshold	$p(\boldsymbol{\theta})$	Model prior

`// Approximate model posterior inference, e.g., deep ensemble`

for *model index* $k = 1 \dots K$ **do**
- Bootstrap sample dataset $\mathcal{D}_k \overset{\text{boot}}{\sim} \mathcal{D}$
- Sample model parameters from prior, $\boldsymbol{\theta}_k \sim p(\boldsymbol{\theta})$
- Train ensemble's k-component via maximum likelihood estimation (MLE) `// Eqn. (3.1)` $\boldsymbol{\theta}_k \leftarrow \operatorname{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_k} [\log q(\mathbf{y}\mid\mathbf{x};\boldsymbol{\theta})]$

`// Online planning`

$\mathbf{x}, \mathcal{G} \leftarrow$ `env.reset()`

while *not* ***done*** **do**
- Get robust imitative plan `// Eqn. (3.4)` $\mathbf{y}^* \leftarrow \operatorname{argmax}_{\mathbf{y}} \bigoplus_{\boldsymbol{\theta}} \log q(\mathbf{y}\mid\mathbf{x};\boldsymbol{\theta}) + \log p(\mathcal{G}\mid\mathbf{y})$
- `// Online adaptation`
- Estimate the predictive variance of the $\mathbf{y}^*$ plan's quality under the model posterior `// Eqn. (3.3)` $u(\mathbf{y}^*) = \operatorname{Var}_{p(\boldsymbol{\theta}\mid\mathcal{D})} [\log q(\mathbf{y}^*\mid\mathbf{x};\boldsymbol{\theta})]$
- **if** $u(\mathbf{y}^*) > \tau$ **then**
  - $\mathbf{y}^* \leftarrow$ Query expert at $\mathbf{x}$
  - $\mathcal{B} \leftarrow \mathcal{B} \cup (\mathbf{x}, \mathbf{y}^*)$
  - Update model posterior on $\mathcal{B}$ `// with any few-shot adaptation method`
- $a_t \leftarrow \mathbb{I}(\cdot\mid\mathbf{y}^*)$
- $\mathbf{x}, \mathcal{G},$ `done` $\leftarrow$ `env.step(`a_t`)`

efficiently querying an expert for online feedback (i.e., demonstrations)? **Q6.** Does AdaRIP's online adaptation mechanism improve success rate?

We evaluate AdaRIP on `CARNOVEL` tasks, where the CARLA autopilot (Dosovitskiy et al., 2017) is queried for demonstrations online when the predictive variance (cf. Eqn. (3.3)) exceeds a threshold, chosen according to RIP's detection score, (cf. Figure 3.4). We measure performance according to the:

Adaptation score. The improvement in success rate as a function of number of online expert demonstrations is used to measure a method's ability to adapt efficiently online. We refer to this metric as *adaptation score*. A method that can adapt online should have a positive adaptation score.

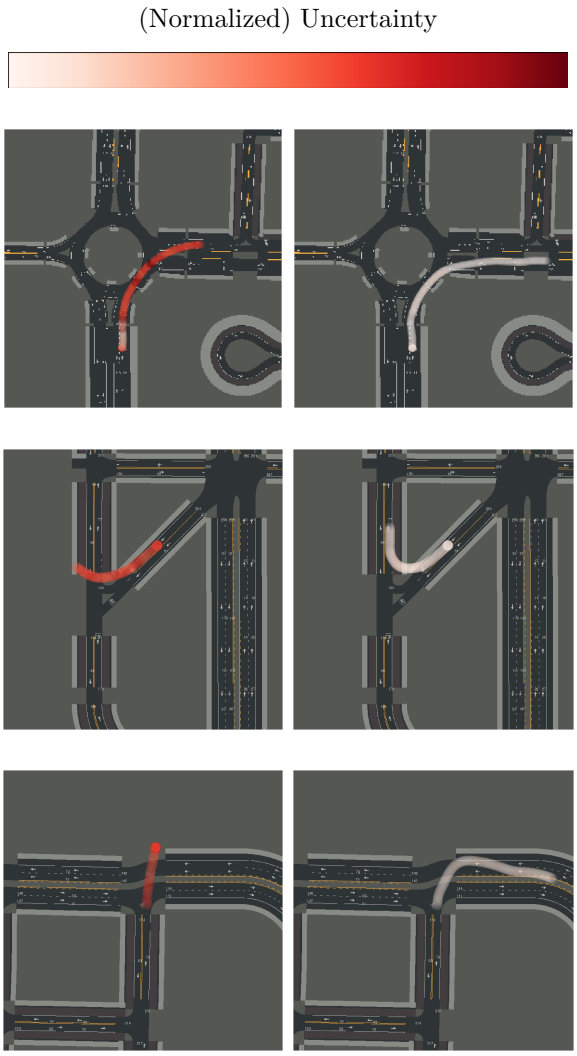
AdaRIP's performance on the most challenging `CARNOVEL` tasks is summarised in Figure 3.7, where, as expected, the success rate improves as the number of online demonstrations increases. Qualitative examples are illustrated in Appendix 14.

Although AdaRIP can adapt to any distribution shift, it is prone to catastrophic forgetting and sample-inefficiency, as many online methods (French, 1999). In this paper, we only demonstrate AdaRIP's efficacy to adapt under distribution shifts and do not address either of these limitations. Future work lies in providing a practical, sample-efficient algorithm to be used in conjunction with the AdaRIP framework. Methods for efficient (e.g., few-shot or zero-shot) and safe adaptation (Finn et al., 2017, Zhou et al., 2019) are orthogonal to AdaRIP and hence any improvement in these fields could be directly used for AdaRIP.

Related Work

Imitation learning. Learning from expert demonstrations (i.e., imitation learning (Widrow and Smith, 1964, Pomerleau, 1989, IL)) is an attractive framework for sequential decision-making in safety-critical domains such as autonomous driving, where trial and error learning has little to no safety guarantees during training. A plethora of expert driving demonstrations has been used for IL (Caesar et al., 2019, Sun et al., 2019, Kesten et al., 2019) since a model mimicking expert demonstrations can simply learn to stay in "safe", expert-like parts of the state space and no explicit reward function need be specified.

On the one hand, behavioural cloning approaches (Liang et al., 2018, Sauer et al., 2018, Li et al., 2018, Codevilla et al., 2018, 2019, Chen et al., 2019) fit command-conditioned discriminative sequential models to expert demonstrations, which are used in deployment to produce expert-like trajectories. On the other hand, Rhinehart et al. (2020) proposed command-*unconditioned* expert trajectory density models which are used for planning trajectories that both satisfy the goal constraints and are likely under the expert model. However, both of these approaches fit point-estimates to their parameters, thus do not quantify their model (*epistemic*) uncertainty, as explained next. This is especially problematic when



(a) RIP **(b)** AdaRIP

Figure 3.9: Examples where the non-adaptive method (a) fails to recover from a distribution shift, despite it being able to detect it. The adaptive method (b) queries the human driver when uncertain (dark red), then uses the online demonstrations for updating its model, resulting into confident (light red, white) and safe trajectories.

estimating what an expert would or would not do in *unfamiliar*, OOD scenes. In contrast, our methods, RIP and AdaRIP, does quantify epistemic uncertainty in order to both improve planning performance and triage situations in which an expert should intervene.

Novelty detection & epistemic uncertainty. A principled means to capture epistemic uncertainty is with Bayesian inference to compute the predictive distribution. However, evaluating the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ with exact inference is intractable for non-trivial models (Neal, 2012). Approximate inference methods (Graves, 2011,

Blundell et al., 2015, Gal and Ghahramani, 2016, Hernández-Lobato and Adams, 2015) have been introduced that can efficiently capture epistemic uncertainty. One approximation for epistemic uncertainty in deep models is model ensembles (Lakshminarayanan et al., 2017, Chua et al., 2018). Prior work by Kahn et al. (2017) and Kenton et al. (2019) use ensembles of deep models to detect and avoid catastrophic actions in navigation tasks, although they can not recover from or adapt to distribution shifts. Our epistemic uncertainty-aware planning objective, RIP, instead, managed to recover from some distribution shifts, as shown experimentally in Section 3.

Coping with distribution shift. Strategies to cope with distribution shift include (a) domain randomization; (b) domain adaptation and (c) online adaptation. *Domain randomization* assumes access to a simulator and *exhaustively* searches for configurations that cover all the data distribution support in order to eliminate OOD scenes, as illustrated in Figure 3.8b. This approach has been successfully used in simple robotic tasks (Sadeghi and Levine, 2016, OpenAI et al., 2018, Akkaya et al., 2019) but it is impractical for use in large, real-world tasks, such as AD. *Domain adaptation* and *bisimulation* (Castro and Precup, 2010), depicted in Figure 3.8c, tackle OOD points by projecting them back to in-distribution points, that are "close" to training points according to some metric. Despite its success in simple visual tasks (McAllister et al., 2019), it has no guarantees under arbitrary distribution shifts. In contrast, *online learning methods* (Cesa-Bianchi and Lugosi, 2006, Ross et al., 2011, Zhang and Cho, 2016, Cronrath et al., 2018) have no-regret guarantees and, provided frequent expert supervision, they asymptotically cover the whole data distribution's support, adaptive to any distribution shift, as shown in Figure 3.8d. In order to continually cope with distribution shift, a learner must receive interactive feedback (Ross et al., 2011), however, the frequency of this costly feedback should be minimised. Our epistemic-uncertainty-aware method, Robust Imitative Planning can cope with some OOD events, thereby reducing the system's dependency on expert feedback, and can use this uncertainty to decide when it cannot cope–when the expert must intervene.

Current benchmarks. We are interested in the control problem, where AD agents get deployed in reactive environments and make sequential decisions. The CARLA Challenge (Ros et al., 2019, Dosovitskiy et al., 2017, Codevilla et al., 2019) is an open-source benchmark for control in AD. It is based on 10 traffic scenarios from the NHTSA pre-crash typology (National Highway Traffic Safety Administration, 2007) to inject challenging driving situations into traffic patterns encountered by AD agents. The methods are only assessed in terms of their generalization to weather conditions, the initial state of the simulation (e.g., the start and goal locations, and the random seed of other agents.) and the traffic density (i.e., empty town, regular traffic and dense traffic).

Despite these challenging scenarios selected in the CARLA Challenge, the agents are allowed to train on the same scenarios in which they evaluated, and so *the robustness to distributional shift is not assessed*. Consequently, both Chen et al. (2019) and Rhinehart et al. (2020) manage to solve the CARLA Challenge with almost 100% success rate, when trained in `Town01` and tested in `Town02`. However, both methods score *almost* 0% when evaluated in `Roundabouts` due to the presence of OOD road morphologies, as discussed in Section 3.

Summary and Conclusions

To summarise, in this paper, we studied autonomous driving agents in out-of-training distribution tasks (i.e. under distribution shifts). We introduced an epistemic uncertainty-aware planning method, called robust imitative planning (RIP), which can detect and recover from distribution shifts, as shown experimentally in a real prediction task, `nuScenes`, and a driving simulator, CARLA. We presented an adaptive variant (AdaRIP) which uses RIP's epistemic uncertainty estimates to efficiently query the expert for online feedback and adapt its model parameters online. We also introduced and open-sourced an autonomous car novel-scene benchmark, termed `CARNOVEL`, to assess the robustness of driving agents to a suite of OOD tasks. Finally, we introduced a dataset called `SHIFTs` to benchmark

the performance of uncertainty-aware autonomous driving agents in OOD settings, based on real-world datasets gathered in various scenes around the world.

Part II

Optimism in the face of uncertainty

4

Active Learning for Treatment Effects from Observational Data

This section is based on the following published work:
Andrew Jesson*, **Panagiotis Tigas***, Joost van Amersfoort, Andreas Kirsch, Uri Shalit, and Yarin Gal. *"Causal-BALD: Deep Bayesian active learning of outcomes to infer treatment-effects from observational data"*. In Advances in Neural Information Processing Systems 34 (2021): 30465-30478.

Abstract

Estimating personalized treatment effects from high-dimensional observational data is essential in situations where experimental designs are infeasible, unethical, or expensive. Existing approaches rely on fitting deep models on outcomes observed for treated and control populations. However, when measuring individual outcomes is costly, as is the case of a tumor biopsy, a sample-efficient strategy for acquiring each result is required. Deep Bayesian active learning provides a framework for efficient data acquisition by selecting points with high uncertainty. However, existing methods bias training data acquisition towards regions of non-overlapping support between the treated and control populations. These are not sample-efficient because the treatment effect is not identifiable in such regions. We in-

troduce causal, Bayesian acquisition functions grounded in information theory that bias data acquisition towards regions with overlapping support to maximize sample efficiency for learning personalized treatment effects. We demonstrate the performance of the proposed acquisition strategies on synthetic and semi-synthetic datasets IHDP and CMNIST and their extensions, which aim to simulate common dataset biases and pathologies.

Introduction

How will a patient's health be affected by taking a medication (Perez, 2019)? How will a user's question be answered by a search recommendation (Noble, 2018)? We can gain insight into these questions by learning about personalized treatment effects. Estimating personalized treatment effects from observational data is essential when experimental designs are infeasible, unethical, or expensive. Observational data represent a population of individuals described by a set of pre-treatment covariates (age, blood pressure, socioeconomic status), an assigned treatment (medication, no medication), and a post-treatment outcome (severity of migraines). An ideal personalized treatment effect is the difference between the post-treatment outcome if the individual receives treatment and the post-treatment outcome if they do not receive treatment. However, it is impossible to observe both outcomes for an individual; therefore, the difference is estimated between populations instead. In the setting of binary treatments, data belong to either the *treatment group* (individuals that received the treatment) or the *control group* (individuals who did not). The personalized treatment effect is the expected difference in outcomes between treated and controlled individuals who share the same (or similar) measured covariates; as an illustration, see the difference between the solid lines in Fig. 4.1 (middle pane).

The use of pre-treatment covariates assembled from high-dimensional, heterogeneous measurements such as medical images and electronic health records is increasing (Sudlow et al., 2015). Deep learning methods have been shown capable of learning personalized treatment effects from such data (Shalit et al., 2017, Shi

et al., 2019, Jesson et al., 2021). However, a problem in deep learning is data efficiency. While modern methods are capable of impressive performance, they need a significant amount of labeled data. Acquiring labeled data can be expensive, requiring specialist knowledge or an invasive procedure to determine the outcome. Therefore, it is desirable to minimize the amount of labeled data needed to obtain a well-performing model. Active learning provides a principled framework to address this concern (Cohn et al., 1996). In active learning for treatment effects (Deng et al., 2011, Sundin et al., 2019, Qin et al., 2021) a model is trained on available labeled data consisting of covariates, assigned treatments, and acquired outcomes. The model predictions decide the most informative examples from data comprised of only covariates and treatment indicators. Outcomes are acquired, e.g., by performing a biopsy for the selected patients, and the model is retrained and evaluated. This process repeats until either a satisfactory performance level is achieved or the labeling budget is exhausted.

At first sight, this might seem simple; however, active learning induces biases that result in divergence between the distribution of the acquired training data and the distribution of the pool set data (Farquhar et al., 2021). In the context of learning causal effects, such bias can have both positive and negative consequences. For example, while random acquisition active learning results in an unbiased sample of the training data, we demonstrate how it can lead to over-allocation of resources to the mode of the data at the expense of learning about underrepresented data. Conversely, while biasing acquisitions toward lower density regions of the pool data can be desirable, it can also lead to outcome acquisition for data with unidentifiable treatment effects, which leads to uninformed, potentially harmful, personalized recommendations.

To see how training data bias can benefit treatment effect estimation, consider one difference between experimental and observational data: the treatment assignment mechanism is unavailable for observational data. In observational data, variables that affect treatment assignment (an untestable condition) may be unobserved. Moreover, the relative proportion of treated to controlled individuals varies

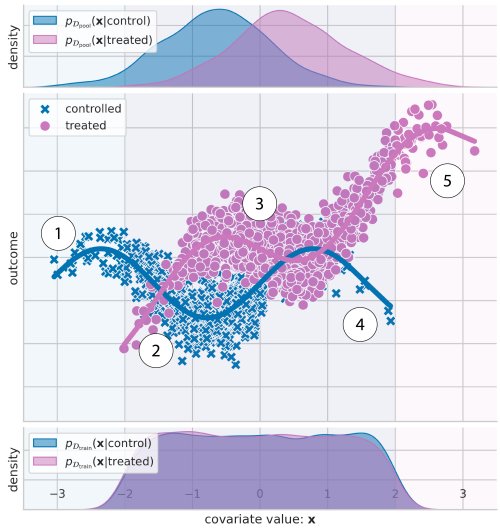


Figure 4.1: Observational data. Top: data density of treatment (right) and control (left) groups. Middle: observed outcome response for treatment (circles) and control (x's) groups. Bottom: data density for active learned training set after a number of acquisition steps.

across different sub-populations of the data. Fig. 4.1 illustrates the latter point, where there are relatively equal proportions of treated and controlled examples for data in region 3. However, the proportions become less balanced as we move to either the left or the right. In extreme cases, say if a group described by some covariate values were systematically excluded from treatment, the treatment effect for that group *cannot be known* (Petersen et al., 2012). Fig. 4.1 illustrates this in region 1, where only controlled examples reside, and in region 5, where only treated cases occur. In the language of causal inference, the necessity of seeing both treated and untreated examples for each sub-population corresponds to satisfying the overlap (or positivity) assumption (see 3). The data available in the pool set limits overlap when treatments cannot be assigned. In this setting, regions 2 and 4 of Fig. 4.1 are very interesting because while either the treated or control group are underrepresented, there may still be sufficient coverage to estimate treatment effects. D'Amour and Franks (2021) have described such regions as having weak overlap. Training data bias towards such regions can benefit treatment effect estimation for underrepresented data by acquiring low-frequency data with sufficient overlap.

We hypothesize that the efficient acquisition of unlabeled data for treatment effect estimation focuses on only exploring regions with sufficient overlap, and

uncertainty should be high for areas with non-overlapping support. The bottom pane of Fig. 4.1 imagines what a resulting training set distribution could look like at an intermediate active learning step. It is not trivial to design such acquisition functions: naively applying active learning acquisition functions results in suboptimal and sample inefficient acquisitions of training examples, as we show below. To this end, we develop epistemic uncertainty-aware methods for active learning of personalized treatment effects from high dimensional observational data. In contrast to previous work that uses only information gain as the acquisition objective, we propose ρBALD and $\mu\rho$BALD as "Causal BALD" objectives because they consider both the information gain and overlap between treated and control groups. We demonstrate the performance of the proposed acquisition strategies using synthetic and semi-synthetic datasets.

Methods

In this section we introduce several acquisition functions, we then analyze how they bias the acquisition of training data, and we show the resulting CATE functions learned from such training data. We are interested in acquisition functions conditioned on realizations of both $\mathbf{x}$ and t:

$$a(\mathcal{D}_{\text{pool}}, p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})) = \underset{\{\mathbf{x}_i, \mathrm{t}_i\}_{i=1}^{b} \subseteq \mathcal{D}_{\text{pool}}}{\operatorname{argmax}} U(\{\mathbf{x}_i, \mathrm{t}_i\}), \tag{4.1}$$

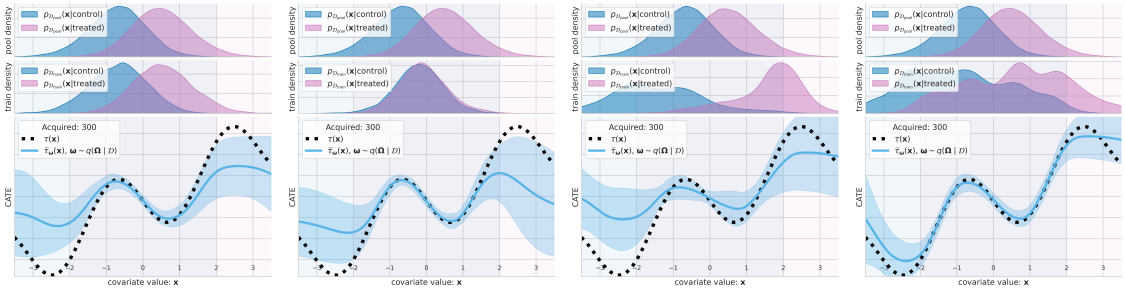
where $U(\{\mathbf{x}_i, \mathrm{t}_i\})$ is a utility function that measures the value of acquiring a batch of data $\{\mathbf{x}_i, \mathrm{t}_i\}$.

In the background section Chapter 2, we specified the three assumptions for Rubin's Causal Model (Rubin, 1974)

Assumption 1. *(Consistency)* $\mathrm{y} = \mathrm{t}\mathrm{y}^{\mathrm{t}} + (1-\mathrm{t})\mathrm{y}^{1-\mathrm{t}}$*, i.e. an individual's observed outcome* y *given assigned treatment* t *is identical to their potential outcome* y^{t}*.*

Assumption 2. *(Unconfoundedness or Ignorability)* $(\mathrm{Y}^0, \mathrm{Y}^1) \perp\!\!\!\perp \mathrm{T} \mid \mathbf{X}$*.*

Assumption 3. *(Overlap)* $0 < \pi_{\mathrm{t}}(\mathbf{x}) < 1 : \forall \mathrm{t} \in \mathcal{T}$*,*



(a) Random **(b)** Propensity **(c)** τBALD **(d)** μBALD

Figure 4.2: Naive acquisition functions How the training set is biased and how this effects the CATE function with a fixed budget of 300 acquired points.

where $\pi_{\mathrm{t}}(\mathbf{x}) \equiv \mathrm{P}(\mathrm{T} = \mathrm{t} \mid \mathbf{X} = \mathbf{x})$ is the **propensity for treatment** for individuals described by covariates $\mathbf{X} = \mathbf{x}$.

When these assumptions are satisfied, $\widehat{\tau}(\mathbf{x}) \equiv \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) - \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0)$, where $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) \equiv E\left[Y \mid \mathrm{T} = 1, \mathbf{X} = \mathbf{x}\right]$ and $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0) \equiv E\left[Y \mid \mathrm{T} = 0, \mathbf{X} = \mathbf{x}\right]$, is an unbiased estimator of $\tau(\mathbf{x}) \equiv E\left[Y^1 \mid \mathbf{X} = \mathbf{x}\right] - E\left[Y^0 \mid \mathbf{X} = \mathbf{x}\right]$ (difference of potential outcomes) and is identifiable from observational data. We make assumptions 1 and 2 (consistency, and unconfoundedness). We relax assumption 3 (overlap) by allowing for its violation over subsets of the support of $\mathcal{D}_{\text{pool}}$. We present all theorems, proofs, and detailed assumptions in Appendix C.

How do naive acquisition functions bias the training data?

To motivate Causal–BALD, we first look at a set of naive acquisition functions. A random acquisition function selects data points uniformly at random from $\mathcal{D}_{\text{pool}}$ and adds them to $\mathcal{D}_{\text{train}}$. In Fig. 4.2a we have acquired 300 such examples from a synthetic dataset and trained a deep-kernel Gaussian process (van Amersfoort et al., 2021b) on those labeled examples. Comparing the top two panels, we see that $\mathcal{D}_{\text{train}}$ (middle) contains an unbiased sample of the data in $\mathcal{D}_{\text{pool}}$ (top). However, in the bottom panel, we see that while the CATE estimator is accurate and confident near the modes of $\mathcal{D}_{\text{pool}}$, it becomes less accurate as we move to lower-density regions. In this way, the random acquisition of data reflects the biases inherent in $\mathcal{D}_{\text{pool}}$ and over-allocates resources to the modes of the distribution. If the mode

were to coincide with a region of non-overlap, the function would most frequently acquire uninformative examples.

Next, we look at using the propensity score to bias data acquisition toward regions where the overlap assumption is satisfied.

Definition 1. *Counterfactual Propensity Acquisition*

$$U_{\pi}(\mathbf{x}, \mathrm{t}) \equiv 1 - \widehat{\pi}_{\mathrm{t}}(\mathbf{x}), \tag{4.2}$$

where $\widehat{\pi}_{\mathrm{t}}(\mathbf{x}) \equiv \mathrm{P}(\mathrm{T} = \mathrm{t} \mid \mathbf{X} = \mathbf{x})$ is the estimator of the **propensity for treatment** for individuals described by covariates $\mathbf{X} = \mathbf{x}$. Intuitively, this function prefers points where the propensity for observing the counterfactual is high. We are considering the setup where $\mathcal{D}_{\text{pool}}$ contains observations of both $\mathbf{X}$ and T, so it is straightforward to train an estimator for the propensity, $\widehat{\pi}_{\mathrm{t}}(\mathbf{x})$. Figure 4.2b shows that while propensity score acquisition matches the treated and control densities in the train set, it still biases data selection towards the modes of $\mathcal{D}_{\text{pool}}$.

The goal of BALD is to acquire data $(\mathbf{x}, \mathrm{t})$ that maximally reduces uncertainty in the model parameters $\mathbf{\Omega}$ used to predict the treatment effect. The most direct way to apply BALD is to use our uncertainty over the predicted treatment effect, expressed using the following information theoretic quantity:

Definition 2. *τBALD*

$$U_{\tau}(\mathbf{x}, \mathrm{t}) \equiv \mathrm{I}(\mathrm{Y}^1 - \mathrm{Y}^0; \mathbf{\Omega} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) \approx \underset{\boldsymbol{\omega} \sim p(\mathbf{\Omega} \mid \mathcal{D}_{\text{train}})}{\mathrm{Var}} \left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) - \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0) \right), \tag{4.3}$$

where $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) \equiv E\left[Y \mid \mathrm{T} = 1, \mathbf{X} = \mathbf{x}\right]$ and $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0) \equiv E\left[Y \mid \mathrm{T} = 0, \mathbf{X} = \mathbf{x}\right]$. Building off the result in (Jesson et al., 2020), we show how the LHS measure about the *unobservable potential outcomes* is estimated by the variance over $\mathbf{\Omega}$ of the *identifiable difference in expected outcomes* in Theorem 1 of the appendix. Alaa and van der Schaar (2018) propose a similar result is for non-parametric models. Intuitively, this measure represents the information gain for $\mathbf{\Omega}$ if we observe the difference in potential outcomes $\mathrm{Y}^1 - \mathrm{Y}^0$ for a given measurement $\mathbf{x}$ and $\mathcal{D}_{\text{train}}$.

However, a fundamental flaw with this measure exists: observing labels for the random variable $\mathrm{Y}^1 - \mathrm{Y}^0$ is impossible. Thus, τBALD represents an irreducible

measure of uncertainty. That is, τBALD will be high if it is uncertain about the label given the unobserved treatment t', regardless of its certainty about the outcome given the factual treatment t, which makes τBALD highest for low-density regions and regions with no overlap. Figure 4.2c illustrates these consequences. We see the acquisition biases the training data away from the modes of the $\mathcal{D}_{\text{pool}}$, where we cannot know the treatment effect (no overlap). In datasets where we have limited overlap, it leads to uninformative acquisitions.

One remedy to the issues of τBALD is to only focus on reducible uncertainty:

Definition 3. *μBALD*

$$U_\mu(\mathbf{x}, \mathrm{t}) \equiv \mathrm{I}(\mathrm{Y}^{\mathrm{t}}; \boldsymbol{\Omega} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) \approx \underset{\boldsymbol{\omega} \sim p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})}{\mathrm{Var}} \left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}) \right), \tag{4.4}$$

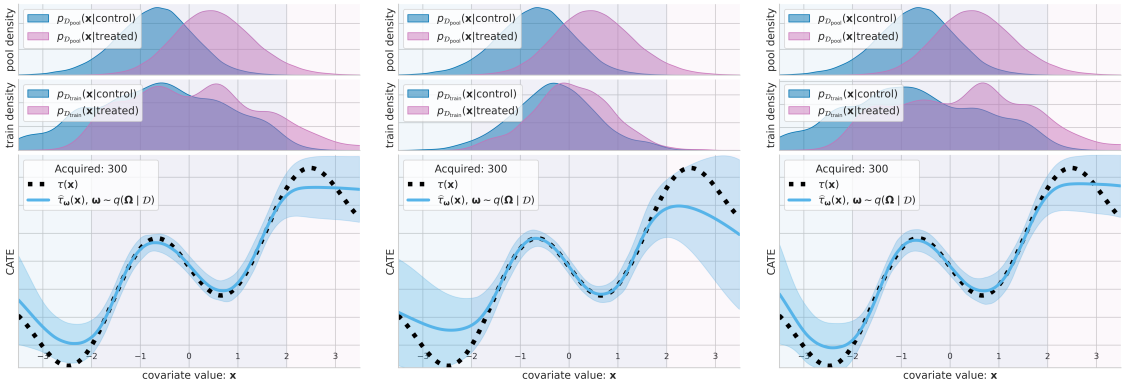
where $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, t) \equiv E\left[Y \mid \mathrm{T} = t, \mathbf{X} = \mathbf{x}\right]$. This measure represents the information gain for the model parameters $\boldsymbol{\Omega}$ if we obtain a label for the observed potential outcome Y^{t} given a data point $(\mathbf{x}, \mathrm{t})$ and $\mathcal{D}_{\text{train}}$. We give proof for these results in Theorem 2 of the appendix.

μBALD only contains observable quantities; however, it does not account for our belief about the counterfactual outcome. As illustrated in Fig. 4.2d, this approach can prefer acquiring $(\mathbf{x}, \mathrm{t})$ when we are also very uncertain about $(\mathbf{x}, \mathrm{t}')$, even if $(\mathbf{x}, \mathrm{t}')$ is not in $\mathcal{D}_{\text{pool}}$. Since we can neither reduce uncertainty over such $(\mathbf{x}, \mathrm{t}')$ nor know the treatment effect, the acquisition function would not be optimally data efficient.

Causal–BALD.

In the previous section, we looked at naive methods that either considered overlap or considered information gain. In this section, we present three measures that account for both factors when choosing a new point to acquire for model training.

A straightforward way to combine knowledge about a data point's information gain and overlap is to simply multiply μBALD(4.4) by the propensity acquisition term (4.2):



(a) $\mu\pi$BALD **(b)** ρBALD **(c)** $\mu\rho$BALD

Figure 4.3: Causal–BALD acquisition functions How the training set is biased and how this affects the CATE function with a fixed budget of 300 acquired points.

Definition 4. *$\mu\pi$BALD*

$$U_{\mu\pi}(\mathbf{x}, \mathrm{t}) \equiv (1 - \widehat{\pi}_{\mathrm{t}}(\mathbf{x})) \underset{\boldsymbol{\omega} \sim p(\boldsymbol{\Omega} | \mathcal{D}_{\text{train}})}{\mathrm{Var}} \left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}) \right), \tag{4.5}$$

where $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, t) \equiv E\left[Y \mid \mathrm{T} = t, \mathbf{X} = \mathbf{x}\right]$ and $\widehat{\pi}_{\mathrm{t}}(\mathbf{x}) \equiv \mathrm{P}(\mathrm{T} = \mathrm{t} \mid \mathbf{X} = \mathbf{x})$. We can see in Fig. 4.3a that the acquisition of training data results in matched sampling that we saw for propensity acquisition in Fig. 4.2b. However, the tails of the overlapping distributions extend further into the low-density regions of the pool set support where the overlap assumption is satisfied.

Alternatively, we can take an information-theoretic approach to combine knowledge about a data point's information gain and overlap. Let $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})$ be an instance of the random variable $\widehat{\mu}^{\mathrm{t}}_{\boldsymbol{\Omega}} \in \mathbb{R}$ corresponding to the expected outcome conditioned on t. Further, let $\widehat{\tau}_{\boldsymbol{\omega}}(\mathbf{x})$ be an instance of the random variable $\widehat{\tau}_{\boldsymbol{\Omega}} = \widehat{\mu}^{1}_{\boldsymbol{\Omega}} - \widehat{\mu}^{0}_{\boldsymbol{\Omega}}$ corresponding to the CATE. Then,

Definition 5. *ρBALD*

$$U_{\rho}(\mathbf{x}, \mathrm{t}) \equiv \mathrm{I}(\mathrm{Y}^{\mathrm{t}}; \widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) \tag{4.6}$$

$$\gtrapprox \frac{1}{2} \log \left(\frac{\mathrm{Var}_{\boldsymbol{\omega}} \left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}) \right) - 2 \, \mathrm{Cov}_{\boldsymbol{\omega}} (\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}), \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))}{\mathrm{Var}_{\boldsymbol{\omega}} \left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}') \right)} + 1 \right), \tag{4.7}$$

where $\widehat{\tau_{\boldsymbol{\omega}}}(\mathbf{x}) \equiv \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) - \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0)$ and $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, t) \equiv E\left[Y \mid \mathrm{T} = t, \mathbf{X} = \mathbf{x}\right]$. This measure represents the information gain for the CATE $\tau_{\boldsymbol{\Omega}}$ if we observe the outcome

Y for a datapoint (**x**, t) and the data we have trained on $\mathcal{D}_{\text{train}}$. We give proof for this result in Theorem 3.

In contrast to μ-BALD, this measure accounts for overlap in two ways. First, ρ–BALD will be scaled by the inverse of the variance of the expected counterfactual outcome $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}')$. This scaling biases acquisition towards examples for which we know about the counterfactual outcome, so we can assume that overlap is satisfied for observed $(\mathbf{x}, \mathrm{t})$. Second, ρ–BALD is discounted by $\mathrm{Cov}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}), \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))$. This discounting is a concept that we will leave for future discussion.

In Fig. 4.3b we see that ρ–BALD has matched the distributions of the treated and control groups similarly to propensity acquisition in Fig. 4.2b. Further, we see that the CATE estimator is more accurate over the support of the data.

There is, however, a shortcoming of ρ–BALD that may lead to suboptimal data efficiency. Consider two examples in $\mathcal{D}_{\text{pool}}$, $(\mathbf{x}_1, \mathrm{t}_1)$ and $(\mathbf{x}_2, \mathrm{t}_2)$ where $\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_1, \mathrm{t}_1)) = \mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_1, \mathrm{t}_1'))$ and $\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_2, \mathrm{t}_2)) = \mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_2, \mathrm{t}_2'))$: for each point, we are as uncertain about the conditional expectation given the factual treatment as we are uncertain given the counterfactual treatment. Further, let $\mathrm{Cov}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_1, \mathrm{t}_1), \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_1, \mathrm{t}_1')) = \mathrm{Cov}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_2, \mathrm{t}_2), \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_2, \mathrm{t}_2'))$. Finally, let $\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_1, \mathrm{t}_1)) > \mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}_2, \mathrm{t}_2))$: we are more uncertain about the conditional expectation given the factual treatment for data point $(\mathbf{x}_1, \mathrm{t}_1)$ than we are for data point $(\mathbf{x}_2, \mathrm{t}_2)$. Under these three conditions, ρ–BALD would rank these two points equally, and so this method would bias training data to the modes of $\mathcal{D}_{\text{pool}}$ when $(\mathbf{x}_2, \mathrm{t}_2)$ is more frequent than $(\mathbf{x}_1, \mathrm{t}_1)$. In practice, it may be more data-efficient to choose $(\mathbf{x}_1, \mathrm{t}_1)$ over $(\mathbf{x}_2, \mathrm{t}_2)$ as it would more likely be a point as yet unseen by the model.

To combine the positive attributes of μ–BALD and ρ–BALD, while mitigating their shortcomings, we introduce $\mu\rho$BALD.

Definition 6. *$\mu\rho$BALD*

$$U_{\mu\rho}(\mathbf{x}, \mathrm{t}) \equiv \underset{\boldsymbol{\omega}}{\mathrm{Var}}\left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})\right) \frac{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\tau}_{\boldsymbol{\omega}}(\mathbf{x}))}{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))}, \tag{4.8}$$

where $\widehat{\tau_\omega}(\mathbf{x}) \equiv \widehat{\mu}_\omega(\mathbf{x}, 1) - \widehat{\mu}_\omega(\mathbf{x}, 0)$ and $\widehat{\mu}_\omega(\mathbf{x}, t) \equiv E\left[Y \mid \mathrm{T} = t, \mathbf{X} = \mathbf{x}\right]$. Here, we scale Equation 4.7, which has equivalent expression $\frac{\mathrm{Var}_\omega(\widehat{\tau_\omega}(\mathbf{x}))}{\mathrm{Var}_\omega(\widehat{\mu_\omega}(\mathbf{x},\mathrm{t}'))}$ by our measure for μBALD such that in the cases where the ratio may be equal, there is a preference for data points the current model is more uncertain about. We can see in Fig. 4.3c that the training data acquisition is distributed more uniformly over the support of the pool data where the overlap assumption is satisfied. Furthermore, the accuracy of the CATE estimator is highest over that region.

Related Work

Deng et al. (2011) propose the use of Active Learning for recruiting patients to assign treatments that will reduce the uncertainty of an Individual Treatment Effect model. However, their setting is different from ours – we assume that suggesting treatments are too risky or even potentially lethal. Instead, we acquire patients to reveal their outcome (e.g., by having a biopsy). Additionally, although their method uses predictive uncertainty to identify which patients to recruit, it does not disentangle the sources of uncertainty; therefore, it will also recruit patients with high outcome variance. Closer to our proposal is the work from Sundin et al. (2019). They propose using a Gaussian process (GP) to model the individual treatment effect and use the expected information gain over the S-type error rate, defined as the error in predicting the sign of the CATE, as their acquisition function. Although GPs are suitable for quantifying uncertainty, they do not work well on high-dimensional input spaces. In this work, we use Neural network methods to obtain uncertainty: Deep Ensembles (Lakshminarayanan et al., 2017) and DUE (van Amersfoort et al., 2021b), a Deep Kernel Learning GP, both of which work well even on high dimensional inputs. Additionally, the authors assume that noisy observations about the counterfactual treatments are available at training time where we make no such assumptions. We compare to this in our experiment by limiting the access to counterfactual observations ($\boldsymbol{\gamma}$ baseline) and adapting it to Deep Ensembles (Lakshminarayanan et al., 2017) and DUE (van Amersfoort et al., 2021b) (we provide more details about the adaptation in Appendix C). Recent

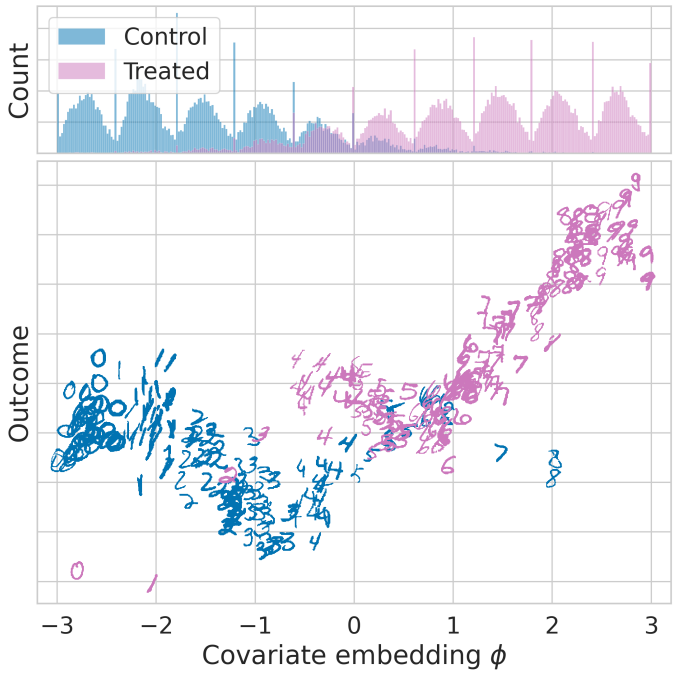


Figure 4.4: Visualizing CMNIST dataset. Model inputs are MNIST digits and assigned treatments. The MNIST digits are high-dimensional proxies for the latent confounding covariate ϕ. Digits are projected onto ϕ by ordering them first by image intensity and then by digit class (0 - 9). Methods must be able to implicitly learn this non-linear mapping in order to predict the conditional expected outcomes.

work by Qin et al. (2021) looks at budgeted heterogeneous effect estimation but does not factor weak or limited overlap into their acquisition function.

Experiments

In this section, we evaluate our acquisition objectives on synthetic and semi-synthetic datasets. Code to reproduce these experiments is available at `https://github.com/OATML/causal-bald`.

Datasets

Starting from the hypothesis that different objectives can target different types of imbalances and degrees overlap, we construct a **synthetic** dataset (Kallus et al., 2019) demonstrating the various biases. We depict this dataset graphically in Fig. 4.1. We use this dataset primarily for illustrative purposes. By design, we have constructed a primary data mode and have regions of weak or no overlap.

Additionally, we study the performance of our acquisition functions on the **IHDP** dataset (Hill, 2011, Shalit et al., 2017), which is a standard benchmark in causal treatment effect literature. Finally, we demonstrate that our method is suitable for high-dimensional, large-sample datasets on **CMNIST** (Jesson et al., 2021), an MNIST (LeCun, 1998) based dataset adapted for causal treatment effect studies. In Fig. 4.4, we see that CMNIST is an adaptation of the synthetic dataset. Model inputs are MNIST digits and assigned treatments, and the response surfaces are generated based on a projection of the digits onto a latent 1-dimensional manifold. The observed digits are high-dimensional proxies for the confounding covariate ϕ. Detailed descriptions of each dataset are available in Appendix C.

Model

Our objectives rely on methods that are capable of modeling uncertainty and handling high-dimensional data modalities. DUE (van Amersfoort et al., 2021b) is an instance of Deep Kernel Learning (Wilson et al., 2016) that uses a deep feature extractor to transform the inputs and defines a Gaussian process (GP) kernel over the extracted feature representation. In particular, DUE uses a variational inducing point approximation (Hensman et al., 2015) and a constrained feature extractor that contains residual connections and spectral normalization to enable reliable uncertainty. Due obtains SotA results on IHDP (van Amersfoort et al., 2021b). In DUE, we distinguish between the model parameters θ and the variational parameters $\boldsymbol{\omega}$, and we are Bayesian only over the $\boldsymbol{\omega}$ parameters. Since DUE is a GP, we obtain a full Gaussian posterior over outcomes from which we can use the mean and covariance directly. When necessary, sampling is very efficient and only requires a single forward pass in the deep model. We describe all hyperparameters in Appendix C.

Baselines

We compare against the following baselines:

Random. This acquisition function selects points uniformly at random.

Table 4.1: Summary of active learning parameters for each dataset.

dataset	warm-up size	acq. size	acq. steps	pool Size	valid. size
Synthetic	10	10	30	10k	1k
IHDP	100	10	38	471	201
CMNIST	250	50	55	35k	15k

Propensity. An acquisition function based on the propensity score (Eq. 4.2). We train a propensity model on the pool data, which we then use to acquire points based on their propensity score. Please note that this is a valid assumption as training a propensity model does not require outcomes.

γ (S-type error rate) (Sundin et al., 2019). This acquisition function is the S-type error rate based method proposed by Sundin et al. (2019). We have adapted the acquisition function to use with Bayesian Deep Neural Networks. The objective is defined as $\mathrm{I}(\boldsymbol{\gamma}; \boldsymbol{\Omega} \mid \mathbf{x}, \mathcal{D}_{\text{train}})$, where $\gamma(x) = \text{probit}^{-1}\left(-\frac{|\mathbf{E}_{p(\tau|\mathbf{x},\mathcal{D}_{\text{train}})}[\tau]|}{\sqrt{\text{Var}(\tau|\mathbf{x},\mathcal{D}_{\text{train}})}}\right)$ and $\text{probit}^{-1}(\cdot)$ is the cumulative distribution function of normal distribution. In contrast to the original formulation, we do not assume access to counterfactual observations at training time.

Experimental Results

For each of the acquisition objectives, dataset, and model we present the mean and standard error of empirical square root of precision in estimation of heterogenous effect (PEHE) [1]. We summarize each active learning setup in Chapter 4. The *warm up size* is the number of examples in the initial pool dataset. *Acquisition size* is the number of examples labeled at each acquisition step. *Acquisition steps* is the number of times we query a batch of labels. *Pool size* is the number of examples in the pool dataset. Finally, *validation size* is the number of examples used for model selection when optimizing the model at each acquisition step.

In Fig. 4.5, we see that epistemic uncertainty aware $\mu\rho$BALD outperforms the baselines, random, propensity, and S-Type error rate ($\boldsymbol{\gamma}$). As analyzed in

[1] $\sqrt{\epsilon_{PEHE}} = \sqrt{\frac{1}{N}\sum_x (\hat{\tau}(x) - \tau(x))^2}$

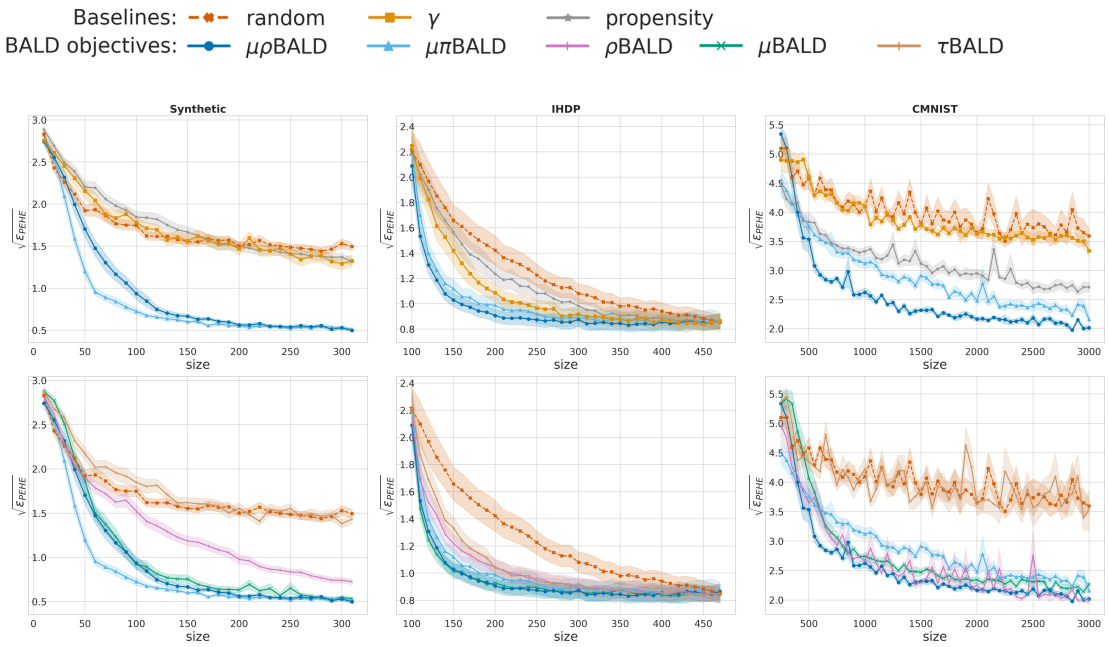


Figure 4.5: $\sqrt{\epsilon_{PEHE}}$ performance (shaded standard error) for DUE models. **(left to right) synthetic** (40 seeds), and **IHDP** (200 seeds). We observe that BALD objectives outperform the **random**, $\boldsymbol{\gamma}$ and **propensity** acquisition functions significantly, suggesting that epistemic uncertainty aware methods that target reducible uncertainty can be more sample efficient.

section 4, we expect this improvement as our acquisition objectives target reducible uncertainty – that is, epistemic uncertainty when there is overlap between treatment and control. Additionally, $\mu\rho$BALDshows superior performance over the other objectives in the high-dimensional dataset CMNIST verifying our qualitative analysis in Figure 4.3c.

Each of (μBALD, ρBALD, $\mu\pi$BALD, and $\mu\rho$BALD) outperforms the baseline methods on these tasks. Of note, the performance ρBALD improves as the dimensionality of the covariates increases. In contrast, the performance of the propensity score-based $\mu\pi$BALD worsens as the dimensionality of the covariates increases. Propensity score estimation is known to be a problem in high-dimensions (DAmour et al., 2021). We see that both μBALD and $\mu\rho$BALD perform consistently as dimensionality increases, with $\mu\rho$BALD showing a statistically significant improvement over μBALD on two of the three tasks. These improvements indicate that $\mu\rho$BALD is more robust for data with high-dimensional covariates than $\mu\pi$BALD ; moreover, $\mu\rho$BALD does not need an additional propensity score model.

Conclusion

We have introduced a new acquisition function for active learning of individual-level causal-treatment effects from high dimensional observational data, based on Bayesian Active Learning by Disagreement (Houlsby et al., 2011). We derive our proposed method from an information-theoretic perspective and compare it with acquisition strategies that either do not consider epistemic uncertainty (i.e., random or propensity-based) or target irreducible uncertainty in the observational setting (i.e., when we do not have access to counterfactual observations). We show that our methods significantly outperform baselines, while also studying the various properties of each of our proposed objectives in both quantitative and qualitative analyses, potentially impacting areas like healthcare where sample efficiency in the acquisition of new examples implies improved safety and reductions in costs.

5

Causal Bayesian Experimental Design

This section is based on the following published work:

Tigas, Panagiotis*, Yashas Annadani*, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. *"Interventions, where and how? experimental design for causal models at scale."*. In Advances in Neural Information Processing Systems 35 (2022)

Abstract

Causal discovery from observational and interventional data is challenging due to limited data and non-identifiability: factors that introduce uncertainty in estimating the underlying structural causal model (SCM). Selecting experiments (interventions) based on the uncertainty arising from both factors can expedite the identification of the SCM. Existing methods in experimental design for causal discovery from limited data either rely on linear assumptions for the SCM or select only the intervention target. This work incorporates recent advances in Bayesian causal discovery into the Bayesian optimal experimental design framework, allowing for active causal discovery of large, nonlinear SCMs while selecting both the interventional target and the value. We demonstrate the performance of the proposed method on synthetic graphs (Erdos-Rènyi, Scale Free) for both linear

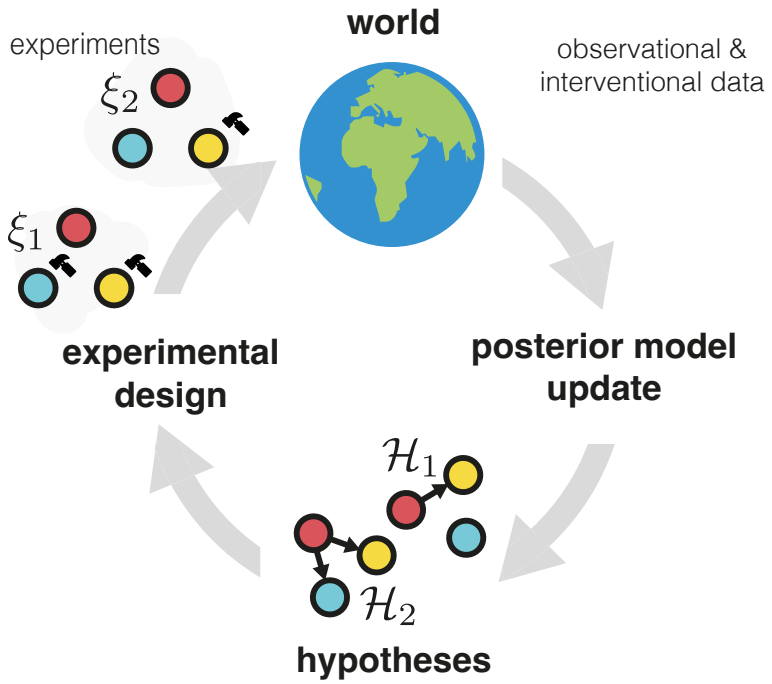


Figure 5.1: Causal Bayesian Experimental Design optimizes experiments that help disambiguate between competing causal hypotheses.

and nonlinear SCMs as well as on the *in-silico* single-cell gene regulatory network dataset, DREAM.

Introduction

What is the structure of the protein-signaling network derived from a single cell? How do different habits influence the presence of disease? Such questions refer to causal effects in complex systems governed by nonlinear, noisy processes. On most occasions, passive observation of such systems is insufficient to uncover the real cause-effect relationship and costly experimentation is required to disambiguate between competing hypotheses. As such, the design of experiments is of significant interest; an efficient experimentation protocol helps reduce the costs involved in experimentation while aiding the process of producing knowledge through the (closed-loop / policy-driven) scientific method (Fig. 5.1).

In the language of causality Pearl (2009), the causal relationships are represented qualitatively by a directed acyclic graph (DAG), where the nodes correspond to different variables of the system of study and the edges represent the flow of

information between the variables. The abstraction of DAGs allows us to represent the space of possible explanations (hypotheses) for the observations at hand. Representing such hypotheses as Bayesian probabilities (beliefs) allows us to formalize the problem of the scientific method as one of Bayesian inference, where the goal is to estimate the posterior distribution $p(\text{DAGs} \mid \text{Observations})$. A posterior distribution over the DAGs allows us to employ information-theoretic acquisition functions that guide experimentation towards the most informative variables for disambiguating between competing hypotheses. Such design procedures belong to the field of *Bayesian Optimal Experimental Design* (Lindley, 1956) for *Causal Discovery* (BOECD) (Tong and Koller, 2001, Murphy, 2001).

In the *Bayesian Optimal Experimental Design* (BOED) (Lindley, 1956) framework, one seeks the experiment that maximizes the *expected information gain* about some parameter(s) of interest. In *causal discovery*, an experiment takes the form of a causal intervention, and the parameters of interest are the DAG structure and the associated Structural Causal Model (SCM) consisting of a set of structural equations representing the functional relationships between the variables in the DAG.

An intervention in a causal model refers to the variable (or target) we manipulate and the value (or strength) at which we set the variable. Hence, the design space in the case of learning causal models is the set of all subsets of the intervention targets and the possibly countably infinite set of intervention values of the chosen targets. The intervention value encapsulates important semantics in many causal inference applications. For instance, in medical applications, an intervention can correspond to the administration of different drugs and the intervention value takes the form of a dosage level for each drug. Even though the appropriate choice of this value is crucial for identifying the underlying causal model, existing work on active causal discovery focuses exclusively on selecting the intervention target (Agrawal et al., 2019, Cho et al., 2016). There, the intervention value is generally some arbitrary fixed value (like 0) which is suboptimal (see Fig. 5.2a). Hence, a holistic treatment of selecting the intervention value and the target in the general case of

nonlinear causal models has been missing. We present a simple yet effective causal Bayesian experimental design method (`CBED` - pronounced "seabed") to acquire optimal intervention targets and values by performing Bayesian optimization.

Additionally, some settings call for the selection of a batch of interventions. The problem of batched interventions is computationally expensive as it requires evaluating all possible combinations of interventions. We extend `CBED` to the batch setting and propose two different batching strategies for tractable, Bayes optimal acquisition of both intervention targets and values. The first strategy `Greedy-CBED` builds up the intervention set greedily. A greedy heuristic is still near-optimal due to submodularity properties of mutual information (Krause and Guestrin, 2012, Agrawal et al., 2019, Kirsch et al., 2019). The second strategy `Soft-CBED` constructs a set of interventions by stochastic sampling from a finite set of candidates, thereby significantly increasing computational efficiency while recovering the causal graph and learning the parameters of the SCM as fast as the greedy strategy. This strategy is well suited for resource-constrained settings.

Throughout this work, we make the following standard assumptions for causal discovery (Peters et al., 2017):

Assumption 1 (Causal Sufficiency)**.** *There are no hidden confounders, and all the random variables of interest are observable.*

Assumption 2 (Finite Samples)**.** *There is a finite number of observational/ interventional samples available.*

Assumption 3 (Nonlinear SCM with Additive Noise)**.** *The structural causal model has nonlinear conditional expectations with additive Gaussian noise.*

Assumption 4 (Single Target)**.** *Each intervention is atomic and applied to a single target of the SCM.*

Additionally, we assume that interventions are planned and executed in batches of size $\mathcal{B}$, with a fixed budget of total interventions given by `Number of Batches` $\times$ $\mathcal{B}$. We also assume that the underlying graph is sparse, as is the case in all the real-world settings (Bengio et al., 2019, Schmidt et al., 2007). Finally, we are interested in recovering the full graph $\mathbf{G}$ with a small number of batches. As with all causal

inference tasks, the assumptions that we make above have to be carefully verified for the application of interest.

We show that our methods, `Greedy-CBED` and `Soft-CBED`, perform better than the state-of-the-art active causal discovery baselines in linear and nonlinear SCM settings. In addition, our approach achieves superior results in the real-world inspired nonlinear dataset, DREAM (Greenfield et al., 2010).

Background

Notation. Let $\mathbf{V} = \{1, \dots, d\}$ be the vertex set of any DAG $\mathbf{g} = (\mathbf{V}, E)$ and $\mathbf{X_V} = \{\mathrm{X}_1, \dots, \mathrm{X}_d\} \subseteq \mathcal{X}$ be the random variables of interest indexed by $\mathbf{V}$. We have an initial observational dataset $\mathcal{D} = \{\mathbf{x_V}^{(i)}\}_{i=1}^{n}$ comprised of instances $\mathbf{x_V} \sim P(\mathrm{X}_1 = \mathrm{x}_1, \dots, \mathrm{X}_d = \mathrm{x}_d) = p(\mathrm{x}_1, \dots, \mathrm{x}_d)$.

Method

The true SCM $\tilde{\boldsymbol{\Theta}} = (\tilde{\mathbf{g}}, \tilde{\boldsymbol{\gamma}})$ over random variables $\mathbf{X_V}$ is a matter of fact, but our belief in $\tilde{\boldsymbol{\theta}}$ is uncertain for many reasons. Primarily, it is only possible to learn the DAG $\tilde{\mathbf{g}}$ up to a Markov equivalence class (MEC) from observational data $\mathcal{D}$. Uncertainty also arises from $\mathcal{D}$ from being a finite sample, which we model by introducing the random variable $\boldsymbol{\Theta}$, of which θ is an outcome. Let $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})$ be an instance of the random variable $\boldsymbol{\theta}$ that is sampled from our posterior over SCMs after observing the dataset $\mathcal{D}$.

We would like to design an experiment to identify an intervention $\xi \coloneqq \{(j, v)\} \coloneqq \mathrm{do}(\mathrm{X}_j = v)$ that maximizes the information gain about $\boldsymbol{\Theta}$ after observing the outcome of the intervention $\mathbf{y} \sim P(\mathrm{X}_1 = \mathrm{x}_1, \dots, \mathrm{X}_d = \mathrm{x}_d \mid \mathrm{do}(\mathrm{X}_j = v) = p(\mathbf{y} \mid \xi)$. Here, $\mathbf{y}$ is an instance of the random variable $\mathbf{Y} \subseteq \mathcal{X}$ distributed according to the distribution specified by the mutilated true graph $\tilde{\mathbf{g}}'$ under intervention. Looking at one intervention at a time, one can formalize BOECD as gain in information about $\boldsymbol{\Theta}$ after observing the outcome of an experiment $\mathbf{y}$. The experiment $\xi \coloneqq \{(j, v)\}$ that maximizes the information gain is the experiment that maximizes the mutual

information between $\boldsymbol{\Theta}$ and $\mathbf{Y}$:

$$\xi^* = \underset{\xi}{\operatorname{argmax}} \{\mathcal{I}(\mathbf{Y}; \boldsymbol{\Theta} \mid \xi, \mathcal{D})\} \tag{5.1}$$

The above objective considers taking `arg max` over not just the discrete set of intervention targets $j \in \mathbf{V}$, but also over the uncountable set of intervention values $v \subset \mathcal{X}_j$. While the existing works in BOECD consider only the design of intervention targets to limit the complexity (Tong and Koller, 2001, Murphy, 2001, Agrawal et al., 2019), our approach tackles both problems. We first outline the methodology for a single design and in Section 5 demonstrate how to extend this single design to a batch setting.

Single Design

To maximize the objective in Equation 5.1, we need to (1) estimate MI for candidate interventions and (2) maximize the estimated MI by optimizing over the domain of intervention value for every candidate interventional target.

Estimating the MI. As mutual information is intractable, there are various ways to estimate it depending on whether we can sample from the posterior and whether the likelihood can be evaluated (Foster et al., 2020, Poole et al., 2019, Houlsby et al., 2011). Since the models we consider allow both posterior sampling and likelihood evaluation, it suffices to obtain an estimator which requires only likelihood evaluation and Monte Carlo approximations of the expectations. To do so, we derive an estimator similar to Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011), which considers MI as a difference of conditional entropies over the outcomes $\mathbf{Y}$:

$$\begin{aligned} \mathrm{U}_{\mathrm{BOED}}(\xi, \mathcal{D}) &\triangleq \mathcal{I}(\mathbf{Y}; \boldsymbol{\Theta} \mid \xi) \\ &= \mathop{\mathbb{E}}_{p(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y},\xi,\mathcal{D})} [\log p(\boldsymbol{\theta} \mid \mathbf{y}, \xi) - \log p(\boldsymbol{\theta} \mid \xi)] \\ &= \mathop{\mathbb{E}}_{p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta},\xi,\mathcal{D})} [\log p(\mathbf{y} \mid \boldsymbol{\theta}, \xi) - \log p(\mathbf{y} \mid \xi)] \end{aligned} \tag{5.2}$$

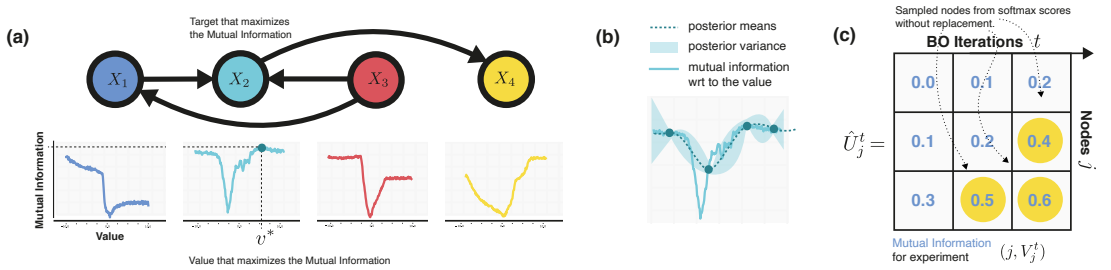


Figure 5.2: **(a)** Each graph shows how the **mutual information** (MI) (y-axis) changes for intervening on that node (plot color matching the node color) with different **values** (x-axis). The SCM in this example is a nonlinear SCM with Additive Gaussian noise. We can see that by intervening on node X_2 with the value v^*, the mutual information gets maximized. **(b)** The posterior distribution of a GP on the Mutual Information function as a response to different intervention values after four Bayesian Optimization (BO) steps. **(c)** For each t iteration of the BO algorithm and each node j, we get a utility function evaluation $\hat{U}_j^t$ (the utility being the MI in our case). Then we sample without replacement proportionally to the scores to prepare a batch (5).

where $\mathrm{H}(\cdot)$ is the entropy. See Appendix D for the derivation. A Monte Carlo estimator of the above equation can be used as an approximation (Appendix D). Eq. (5.2) has an intuitive interpretation. It assigns high mutual information to interventions that the model disagrees the most regarding the outcome. We denote the MI for a single design as $\mathcal{I}(\{(j, v)\}) \coloneqq \mathrm{I}(\mathbf{Y}; \boldsymbol{\Theta} \mid \{(j, v)\}, \mathcal{D})$.

Selecting the Intervention Value. As shown in (5.1), maximizing the objective is achieved not only by selecting the intervention target but also by setting the appropriate intervention value. Selecting intervention targets and values is an intractable problem in general. However, we can make some progress by observing that the number of nodes to intervene on is usually small [1] and hence we can afford to optimize over the intervention target (argmax over discrete and small number of nodes). On the other hand, selecting the value is intractable, even in the case of a few nodes, since it is continuous. For any given target node j, MI is a nonlinear function over $v \in \mathcal{X}_j$ (See Fig 5.2) and hence solving with gradient ascent techniques only yields a local maximum. Given that MI is expensive to evaluate, we treat MI for a given target node j as a black-box function and obtain its maximum using

[1] Bayesian Causal Discovery methods currently scale up to hundreds of nodes, however, we acknowledge that when such methods scale up to thousands and millions of nodes, the problem of selecting intervention targets will become significantly harder.

Bayesian Optimization (BO) (Kushner, 1964, Zhilinskas, 1975, Močkus, 1975). BO seeks to find the maximum of this function $\max_{v \in \mathcal{X}_j} \mathcal{I}(\{(j, v)\})$ over the entire set $\mathcal{X}_j$ with as few evaluations as possible. See appendix 3 for details.

BO typically proceeds by placing a Gaussian Process (GP) (Rasmussen, 2003) prior on the function $\mathcal{I}(\{j, \cdot\})$ and obtain the posterior over this function with the queried points $\mathbf{v}^* = \{v^{(1)*}, \ldots, v^{(T)*}\}$. Let the value of the mutual information queried at each optimization step t be $\hat{\mathrm{U}}_j^t = \mathcal{I}(\{(j, v^{(t)*}\})$. The posterior *predictive* of a point $v^{(t+1)}$ can be obtained in closed form as a Gaussian with mean $\boldsymbol{\mu}_{\boldsymbol{j}}^{(\boldsymbol{t+1})}(v)$ and variance $\boldsymbol{\sigma}_{\boldsymbol{j}}^{(\boldsymbol{t+1})}(v)$. Subscript j signifies the fact that we maintain different μ and σ per intervention target and superscript t represents the BO step. Querying proceeds by having an acquisition function defined on this posterior, which suggests the next point to query. For BO, we use an acquisition function called the Upper Confidence Bound (UCB) (Srinivas et al., 2010) which suggests the next point to query by trading-off *exploration* and *exploitation* with a hyperparameter β_j^t: $v_j^{(t+1)*} = \operatorname{argmax}_v \boldsymbol{\mu}_{\boldsymbol{j}}^{\boldsymbol{t}}(v) + \sqrt{\beta_j^{t+1}} \boldsymbol{\sigma}_{\boldsymbol{j}}^{\boldsymbol{t}}(v)$. We chose UCB because it is a simple and effective acquisition function and it was sufficient for demonstrating the performance of our approach, however, other acquisition functions are possible as well (see Frazier (2018) for a review).

We run GP-UCB independently on every candidate intervention target $j = \{1, \ldots, d\}$ by querying points within a fixed domain $[-k, k] \subset \mathbb{R}$. Note that the domain can be chosen based on the application, for example, if we must constrain dosage levels within a fixed range. Each GP is one-dimensional in our setup; hence a few evaluations of UCB are sufficient to get a good value maxima candidate. Further, GP-UCB for each candidate target is parallelizable, making it efficient. We finally select the design with the highest MI across the candidate intervention targets.

Batch Design

In many applications, it is desirable to select the most informative *set* of interventions instead of a single intervention at a time. Take, for example, a biologist enter-

ing a wet lab with a script of experiments to execute. Batching experiments removes the bottleneck of waiting for an experiment to finish and get analyzed until executing the next one. Given a budget per batch $\mathcal{B}$ which denotes the number of experiments in a batch, the problem of selecting the batch then becomes $\operatorname{argmax}_{\boldsymbol{\Xi}} \mathrm{I}(\mathbf{Y}; \boldsymbol{\Theta} \mid \boldsymbol{\Xi}, \mathcal{D})$, such that $\texttt{cardinality}(\boldsymbol{\Xi}) = \mathcal{B}$, where $\boldsymbol{\Xi}$ is a set of interventions $\bigcup_{i=1}^{\mathcal{B}} (j_i, v_i)$ and $\mathbf{Y}$ denotes the random variable for the outcomes of the interventions of the batch. We denote the MI for a batch design as $\mathcal{I}(\boldsymbol{\Xi}) \coloneqq \mathrm{I}(\mathbf{Y}; \boldsymbol{\Theta} \mid \boldsymbol{\Xi}, \mathcal{D})$.

Greedy Algorithm. Computing the optimal solution $\mathcal{I}(\boldsymbol{\Xi}^*)$ is computationally infeasible. However, as the conditional mutual information is *submodular* and *non-decreasing* (see Appendix 3 for proof), we can derive a simple greedy algorithm (Algorithm 3) that can achieve at least a $(1 - 1/e) \approx 0.64$ approximation of the optimal solution (Krause and Guestrin, 2012, Nemhauser et al., 1978). We denote this strategy as `Greedy-CBED`. Please note, that the greedy algorithm is optimizing the batch design sequentially, i.e. it selects the first design and then it selects the second design conditioned on the first design and so on.

Soft Top-K. Although the greedy algorithm is tractable, it requires $O(\mathcal{B}d)$ instances of GP-UCB. Kirsch et al. (2021) show that a soft top-k selection strategy performs similarly to the greedy algorithm, reducing the computation requirements to $O(d)$ runs of GP-UCB. To achieve this, we construct a finite set of candidate intervention target-value pairs by keeping all the T evaluations of GP-UCB for each node $j = \{1, \ldots, d\}$. Therefore, for d nodes, our candidate set is comprised of $d \times T$ experiments. We score each experiment in this candidate set using the MI estimate. We then sample *without replacement* $\mathcal{B}$ times proportionally to the *softmax* of the MI scores (Algorithm 4). We denote this strategy as `Soft-CBED`.

Comparison with existing active causal discovery methods

We outline how our approach compares with two main existing active causal discovery methods.

Algorithm 3: `Greedy-CBED`

Input : $\mathcal{E}$ environment, N initial observational samples, $\mathcal{B}$ batch Size, d number of nodes

▷ `Initialize set of experiments` $\boldsymbol{\Xi}$ `to empty`

$\boldsymbol{\Xi} \leftarrow \varnothing$
for $n = 1 \dots \mathcal{B}$ **do**
  for $j = 1 \dots d$ **do**
    ▷ `Select optimal intervention value per node` j `using GP-UCB`
    $V_j \leftarrow \operatorname{argmax}_v \mathcal{I}(\boldsymbol{\Xi} \cup \{(j, v)\})$
    $U_j \leftarrow \mathcal{I}(\boldsymbol{\Xi} \cup \{(j, V_j)\})$
  $j^* \leftarrow \operatorname{argmax}_j U_j$
  $v^* \leftarrow V_{j^*}$
  $\boldsymbol{\Xi} \leftarrow \boldsymbol{\Xi} \cup \{(j^*, v^*)\}$
return $\boldsymbol{\Xi}$

Algorithm 4: `Soft-CBED`

Input : $\mathcal{E}$ environment, N initial observational samples, $\mathcal{B}$ batch Size, d number of nodes, ζ softmax temperature

for $j = 1 \dots d$ **do**
  ▷ `Select candidate intervention values per node` j `using GP-UCB`
  Initialize μ_j^0 and σ_j^0
  for $t = 1 \dots T$ **do**
    $V_j^t \leftarrow \operatorname{argmax}_v \mu_j^{t-1}(v) + \sqrt{\beta^t}\sigma_j^{t-1}(v)$
    $\hat{U}_j^t \leftarrow \mathcal{I}(\{(j, V_j^t)\})$
    Update the GP to obtain μ_j^t and σ_j^t
$\{(t_i, j_i)\}_{i \in \{1,\dots,\mathcal{B}\}} \leftarrow \mathcal{B}$ samples *without* replacement $\propto \exp(\hat{U}_j^t / \zeta)$
$\boldsymbol{\Xi} \leftarrow \{(j_i, V_{j_i}^{t_i})\}_{i \in \{1,\dots,\mathcal{B}\}}$
return $\boldsymbol{\Xi}$

ABCD (Agrawal et al., 2019). The estimator of MI used in ABCD is based on weighted importance sampling. However, for the specific choice of the importance sampling weights used in ABCD, their MI estimator ends up with the same approximation as in our method (see Appendix 3). Nevertheless, ABCD does not select intervention values but suboptimally sets them to a fixed value. In addition, our proposed `Soft-CBED` is a faster and more efficient batch strategy, especially when values also have to be acquired. From this perspective, our approach is an extension of ABCD with nonlinear assumptions, value acquisition, and a soft top-k batching strategy.

AIT (Scherrer et al., 2021). AIT is an F-score-based intervention target acquisition strategy. Although it is not a BOECD method, we prove here that it can be viewed as a Monte Carlo estimate of the approximation to MI when the outcomes $\mathbf{Y}$ are Gaussian. Nevertheless, AIT does not select intervention values

like ABCD and does not have a batch strategy.

Related Work

Early efforts of using *Bayesian Optimal Experimental Design for Causal Discovery* (BOECD) can be found in the works of Murphy (2001) and Tong and Koller (2001). However, these approaches deal with simple settings like limiting the graphs to topologically ordered structures, intervening sequentially, linear models, and discrete variables.

In Cho et al. (2016) and Ness et al. (2017), BOECD was applied for learning biological networks structure. BOECD was also explored in Greenewald et al. (2019) under the assumption that undirected edges of the graph always forms a tree. More recently, ABCD framework Agrawal et al. (2019) extended the work of Murphy (2001) and Tong and Koller (2001) in the setting where interventions can be applied in batches with continuous variables. To achieve this, they (approximately) solve the submodular problem of maximizing the batched mutual information between interventions (experiments), outcomes, and observational data, given a DAG. DAG hypotheses are sampled using *DAG-bootstrap* (Friedman et al., 2013). Our work differs from ABCD in a few ways: we work with both linear and nonlinear SCMs by using state-of-the-art posterior models over DAGs Lorch et al. (2021), we apply BO to select the value to intervene with, but we also prepare the batch using *soft*BALD Kirsch et al. (2021) which is significantly faster than the greedy approximation of ABCD method.

In von Kügelgen et al. (2019) the authors proposed the use of *Gaussian Processes* to model the posterior over DAGs and then use BO to identify the value to intervene with, however, this method was not shown to be scalable for larger than bivariate graphs since they rely on multi-dimensional Gaussian Processes for modeling the conditional distributions.

A new body of work has emerged in the field of differentiable causal discovery, where the problem of finding the structure, usually from observational data, is solved with gradient ascent and functional approximators, like neural networks

(Zheng et al., 2018, Ke et al., 2019, Brouillard et al., 2020, Bengio et al., 2019). In recent works (Cundy et al., 2021, Lorch et al., 2021, Annadani et al., 2021), the authors proposed a variational approximation of the posterior over the DAGs which allowed for modeling a distribution rather than a point estimate of the DAG that best explains the observational data $\mathcal{D}$. Such work can be used to replace *DAG-bootstrap* (Friedman et al., 2013), allowing for the modeling of posterior distributions with greater support.

Besides the BOECD-based approaches, a few active causal learning works have been proposed (He and Geng, 2008, Gamella and Heinze-Deml, 2020, Scherrer et al., 2021, Shanmugam et al., 2015, Squires et al., 2020, Kocaoglu et al., 2017a). Active ICP Gamella and Heinze-Deml (2020) uses ICP Peters et al. (2016) for causal learning while using an active policy to select the target, however, this work is not applicable in the setting where the full graph needs to be recovered. In Zhang et al. (2021), the authors propose an active learning method to the problem of identifying the interventions that push a dynamical causal network towards a desired state. A few approaches tackle the problem of actively acquiring interventional data to orient edges of a skeletal graph (Shanmugam et al., 2015, Squires et al., 2020, Kocaoglu et al., 2017a). Closer to our proposal belongs AIT (Scherrer et al., 2021), which uses a neural network-based posterior model over the graphs but evaluates the F-score to select the interventions.

Experiments

We evaluate the performance of our method on synthetic and real-world causal experimental design problems and a range of baselines. We aim to investigate the following aspects empirically: (1) competitiveness of the overall proposed strategies of `Greedy-CBED` and `Soft-CBED` at scale (50 nodes) on synthetic datasets; (2) performance of the value acquisition strategy based on GP-UCB; and (3) performance of the proposed approach on a real-world inspired dataset.

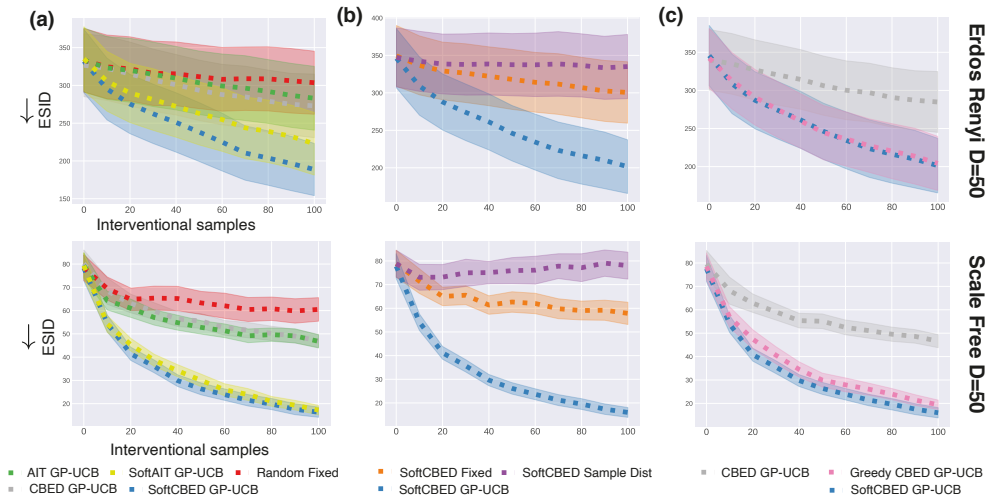


Figure 5.3: Results on the $\mathbb{E}$**-SID** ↓ metric (100 seeds, with standard error of the mean shaded) for 50 variables involving nonlinear functional relationships and additive Gaussian noise. (a) We show that `Soft-CBED` with GP-UCB value selection strategy significantly outperforms the baselines. (b) We isolate the effect of the value selection strategy. We show that intervening with a fixed value and sampling from the support of data both perform worse than having an optimizer like GP-UCB. (c) we compare non-batch (`CBED`) vs batch-based acquisition functions (`Greedy-CBED`, `Soft-CBED`). As we can see, `Soft-CBED` performs as well as `Greedy-CBED`. For all experiments, we use the DiBS (Lorch et al., 2021) posterior model.

Acquisition Functions

We are considering the following acquisition functions:

Random

Random baseline acquires interventional targets at random.

AIT / *soft*AIT

active intervention targeting (AIT) Scherrer et al. (2021) uses an f-score-based acquisition strategy to select the intervention targets. Since the originally proposed approach does not consider a batch setting, we introduce a variant that augments AIT with the proposed soft batching, as described in section 5.

CBED / GreedyCBED / SoftCBED

These are the Monte Carlo estimates of MI, as described in section 5. `CBED` selects a single intervention (target and value) that maximizes the MI and this intervention is applied to the whole batch. In `Greedy-CBED` , the batch is built up in a greedy fashion selecting the target, value pairs one at a time

(Algorithm 3). `Soft-CBED` is sampling (target, value) pairs proportionally to the MI scores to select a batch, as described in section 5 and Algorithm 4.

Value Selection Strategies

Also, we are considering the following value section strategies:

Fixed

This value selection strategy assumes setting the value of the intervention to a fixed value. In the experiments, we fixed the value to 0.

Sample-Dist

This value selection strategy samples from the support of the observational data.

GP-UCB

This strategy uses the proposed GP-UCB Bayesian optimization strategy to select the value that maximizes MI. Although this additional optimization step increases the computational needs, we found that eight Bayesian optimization steps were sufficient.

Tasks

Synthetic Graphs. In this setting, we generate ErdsRényi Erdős and Rényi (1959) (ER) and Scale-Free (SF) graphs (Barabási and Albert, 1999) of size 20 and 50. For linear SCMs, we sample the edge weights γ uniformly at random. For the nonlinear SCM, we parameterize each variable to be a Gaussian whose mean is a nonlinear function of its parents. We model the nonlinear function with a neural network. In all settings, we set noise variance $\sigma^2 = 0.1$. For both types of graphs, we set the expected number of edges per vertex to 1. We provide more details about the experiments in appendix 3.

Single-Cell Protein-Signalling Network. The DREAM family of benchmarks (Greenfield et al., 2010) are designed to evaluate causal discovery algorithms of the regulatory networks of a single cell. A set of ODEs and SDEs generates the dataset, simulating the reverse-engineered networks of single cells. We use

`GeneNetWeaver` (Schaffter et al., 2011) to simulate the steady-state *wind-type* expression and single-gene *knockouts*. Refer to appendix 3 for the exact settings.

Results

For each of the acquisition objectives and datasets, we present the mean and standard error of the expected structural hamming distance $\mathbb{E}$**-SHD**, expected structural interventional distance $\mathbb{E}$**-SID** (Peters and Bühlmann, 2015), area under the receiver operating characteristic curve **AUROC** and area under the precision-recall curve **AUPRC**. In Appendix E we describe the metrics in detail. We evaluate these metrics as a function of the number of acquired interventional samples (or experiments), which helps quantitatively compare different acquisition strategies. Apart from $\mathbb{E}$**-SID**, we relegate results with other metrics to the appendix 3.

On the synthetic graphs (Figure 5.3(a)), we can see that for ER and SF graphs with $50D$ variables and nonlinear functional relationships, the proposed approach based on soft top-k to select a batch with GP-UCB outperforms all the baselines in terms of the $\mathbb{E}$**-SID** metric. On the other hand, AIT alone does not converge to the ground truth graph fast even after combining with the proposed value acquisition, but when further augmented with the proposed soft strategy, the *soft*AIT recovers the ground truth causal graph upto 4 times faster and performs competitively to `Soft-CBED`. We observe similar performance across other metrics as well. In addition, we found this trend to hold for $20D$ variables and linear models. Full results are presented in the appendix 3.

Table 5.1: Performance comparison between different value selection and batch strategies for `CBED`. Experiments are performed using an AMD EPYC 7662 64-Core CPU and Tesla V100 GPU.

Strategy		
Value	**Batch**	**Runtime(s)**
Fixed	Greedy	32.56
	Soft	6.42
GP-UCB	Greedy	284.98
	Soft	24.17

Next, we examine the importance of having a value selection strategy for active causal discovery. We use the MI estimator in Equation 5.2; moreover, we test the proposed GP-UCB with two heuristics - the fixed value strategy and sampling values from the support. As we can see in Figure 5.3(b), selecting the value using GP-UCB clearly benefits the causal discovery process. We expect this finding as

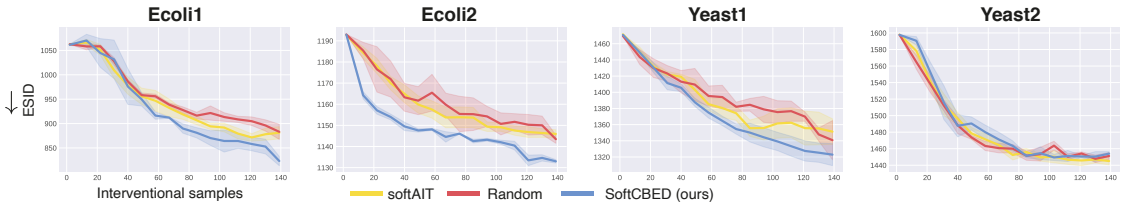


Figure 5.4: Comparison of acquisition functions on DREAM dataset, for 50 dimensions and batch size 10 on $\mathbb{E}$**-SID**$\downarrow$ metric (6 seeds, with standard error of the mean).

the mutual information is not constant with respect to the intervened value. To make this point clear, we demonstrate in the appendix 3 the influence of the value in a simple two variables graph. In addition, we note that naively sampling from the support of the observed dataset performs worse than fixing the value to 0. We hypothesize that this is due to lower epistemic uncertainty in the high density regions of the support, hinting that these regions might be less informative.

In order to further understand how the soft batch strategy compares with other batch selection strategies, we compare the results of `Soft-CBED` with `Greedy-CBED` and `CBED`. We observe (Figure 5.3(c)) that `Greedy-CBED` and `Soft-CBED` give very similar results overall. While `Greedy-CBED` is optimal under certain conditions Kirsch et al. (2019), `Soft-CBED` remains competitive and has the advantage that the batch can be selected in a one-shot manner. This is also evident from the runtime performance of both these batching strategies in Table 5.1. Both these batch selection strategies perform significantly better than selecting one intervention target/value pair, and executing them $\mathcal{B}$ times (`CBED`).

Finally, on the DREAM task, we see that our method outperforms *soft*AIT and random baselines on the $\mathbb{E}$**-SID** metric (see Figure 5.4). In these experiments, since the intervention is emulating the gene knockout setting, we only use the fixed value strategy, with a value of 0.0. Although random baseline still remains a competitive choice, in certain settings, `Soft-CBED` objective is significantly better (`Ecoli1`, `Ecoli2` datasets).

Summary and Conclusions

This paper studies the problem of efficiently selecting the Bayes optimal experiments to discover causal models. Our proposed framework simultaneously answers the questions of *where* and *how* to intervene in a batched setting. We present a Bayesian optimization strategy to acquire interventional targets and values. Further, we propose two different batching strategies: one based on greedy selection and the other based on soft top-k selection. The proposed methodology for selecting intervention target-value pairs in a batched setting provides superior performance over the state-of-the-art for causal models up to $50D$ variables. We validate this using synthetic datasets and using real-world inspired datasets of single-cell regulatory networks, showing the potential impact on areas like biology and other experimental sciences.

Differentiable Bayesian Causal Experimental Design

As in prior work (Tigas et al., 2022, Sussex et al., 2021), we are interested in the setting of batch design where we design B experiments at once before collecting experimental data. In other words, we seek a multiset of intervention targets and corresponding states which are jointly maximally informative about the parameters. We denote this multiset as $\xi_{1:B} \coloneqq (I_{1:B}, S^I_{1:B})$. After executing a batch of experiments and collecting experimental outcomes, an experimenter might wish to design a new batch of experiments based on collected data (as summarized by the posterior distribution). Let h_t denote experimental history $(\xi^1, \mathbf{y}^1), \ldots, (\xi^t, \mathbf{y}^t)$ after t batches of acquisition. The BOED objective for this batch setting at any point t is given by the joint mutual information:

$$\begin{aligned} &\mathcal{I}(\mathbf{Y}^t_{1:B}; \boldsymbol{\Theta} \mid \xi^t_{1:B}, h_{t-1}) \\ &\quad = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}|h_{t-1}) \\ p(\mathbf{y}^t_{1:B}|\boldsymbol{\theta}, \xi^t_{1:B})}} \left[\log \frac{p(\mathbf{y}^t_{1:B} \mid \boldsymbol{\theta}, \xi^t_{1:B})}{p(\mathbf{y}^t_{1:B} \mid \xi^t_{1:B}, h_{t-1})} \right] \end{aligned} \tag{5.3}$$

where $\mathbf{Y}^t_{1:B}$ are the random variables corresponding to experimental outcomes for iteration t, $\mathbf{y^t_{1:B}}$ are the instances of these random variables and $\xi^t_{1:B}$ is the corresponding multiset of experimental designs. We drop the superscript t from these variables for simplicity of exposition. Ideally, we wish to maximize the above objective by obtaining the gradients $\nabla_{\xi_{1:B}} \mathrm{I}$ and performing gradient ascent. However, the above objective is doubly intractable (Rainforth et al., 2018a) and approximations are required. This usually leads to a two-stage procedure where the above objective is first estimated with respect to an inference network and then maximized with respect to designs (Foster et al., 2019), which can be typically inefficient (Foster et al., 2020).

Estimators of the Joint Mutual Information

Nested Monte Carlo

Following Huan and Marzouk (2014), Foster et al. (2020, 2021), we consider an estimator that allows for approximating the EIG objective while *simultaneously* optimizing for the experiment ξ that maximizes the objective via gradient-based methods. This estimator, called Nested Monte Carlo (NMC), is based on contrastive estimation of the experimental likelihood and has been extensively used in Bayesian experimental design (Ryan, 2003, Myung et al., 2013). More precisely, assuming some past observational and interventional data $h_{t-1} = \{(\xi^1, \mathbf{y}^1), \dots, (\xi^{t-1}, \mathbf{y}^{t-1})\}$, for every parameter sample from the posterior distribution $\boldsymbol{\theta}_0 \sim p(\boldsymbol{\theta} \mid h_{t-1})$, a set of contrastive samples $\boldsymbol{\theta}_{1:L} \sim p(\boldsymbol{\theta} \mid h_{t-1})$ are considered to obtain a unified objective:

$$\mathcal{U}^t_{\text{NMC}}(\xi_{1:B}) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_{0:L} \mid h_{t-1}) \\ p(\mathbf{y}_{1:B} \mid \boldsymbol{\theta}_0, \xi_{1:B}))}} \left[\log \frac{p(\mathbf{y}_{1:B} \mid \xi_{1:B}, \boldsymbol{\theta}_0)}{\frac{1}{L} \sum_{\ell=1}^{L} p(\mathbf{y}_{1:B} \mid \xi_{1:B}, \boldsymbol{\theta}_\ell))} \right] \tag{5.4}$$

This estimator converges to the true mutual information as $L \to \infty$ (Rainforth et al., 2018a). If the design space is continuous, the optimal *batch* of experiment $\xi^*_{1:B}$ can be found by *directly* maximizing the NMC objective ($\xi^*_{1:B} \leftarrow \operatorname{argmax}_{\xi_{1:B}} \mathcal{U}^t_{\text{NMC}}(\xi_{1:B})$) with gradient-based techniques (Huan and Marzouk, 2014).

The above objective requires estimating the posterior distribution $p(\boldsymbol{\theta} \mid h_{t-1})$ after every acquisition. For causal models, while it is generally hard to estimate this posterior due to DAG space of causal structures being discrete and super-exponential in the number of variables (Tong and Koller, 2001), many approaches exist in the literature (Agrawal et al., 2019, Lorch et al., 2021, Cundy et al., 2021). These approximate posteriors can be nevertheless used for estimating the NMC objective.

Importance Weighted Nested Monte Carlo

To establish an alternative path to estimating the mutual information, we begin by utilizing an observation from Foster et al. (2019) that it is possible to draw the contrastive samples from a distribution other than $p(\boldsymbol{\theta} \mid h_{t-1})$ and obtain an

asymptotically exact estimator, up to a constant C that does not depend on $\xi^t_{1:B}$. Drawing samples from the *original* prior $p(\boldsymbol{\theta})$ gives the estimator

$$\mathcal{I}(\mathbf{Y}^t_{1:B}; \boldsymbol{\Theta} \mid \xi^t_{1:B}, h_{t-1}) - C = \lim_{L\to\infty} \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_0|h_{t-1})p(\boldsymbol{\theta}_{1:L}) \\ p(\mathbf{y}_{1:B}|\boldsymbol{\theta}_0,\xi_{1:B})}} \left[\log \frac{p(\mathbf{y}_{1:B}|\xi_{1:B},\boldsymbol{\theta}_0)}{\frac{1}{L}\sum_{\ell=1}^{L} p(\mathbf{y}_{1:B}|\xi_{1:B},\boldsymbol{\theta}_\ell)p(h_{t-1}|\boldsymbol{\theta}_\ell)}\right].$$

The remaining wrinkle is that we must sample $\boldsymbol{\theta}_0$ from $p(\boldsymbol{\theta}_0|h_{t-1})$. We propose the conceptually simplest approach of applying self-normalized importance sampling (SNIS) to the outer expectation. The resulting objective, based on efficiently re-using samples in a leave-one-out manner, can optimize designs by just sampling parameters from the prior, without having to estimate the posterior:

$$\mathcal{U}^t_{\text{IWNMC}}(\xi_{1:B}) = \mathbb{E}\left[\sum_{m=1}^{L} \omega_m \log \frac{p(\mathbf{y}_{m,1:B}|\boldsymbol{\theta}_m,\xi_{1:B})}{\frac{1}{L-1}\sum_{\ell\neq m} p(\mathbf{y}_{m,1:B}|\boldsymbol{\theta}_\ell,\xi_{1:B})p(h_{t-1}|\boldsymbol{\theta}_\ell)}\right] \tag{5.5}$$

where $\boldsymbol{\theta}_{1:L} \sim p(\boldsymbol{\theta}_{1:L})$ are sampled from the original prior, $\mathbf{y}_{m,1:B} \sim p(\mathbf{y}_{1:B}|\boldsymbol{\theta}_m,\xi_{1:B})$ are all the experimental outcomes in the batch for parameter $\boldsymbol{\theta}_m$ and $\omega_m \propto p(h_{t-1}|\boldsymbol{\theta}_m)$ are self-normalized weights. A full derivation is given in Appendix E.

As IWNMC does not require any posterior estimation but instead relies entirely on the prior, it completely sidesteps the causal discovery process for designing experiments. This is a paradigm change from the NMC estimator which requires causal discovery through the estimation of the posterior. However, we note that using IWNMC with just the prior (Eq. 5.5) as opposed to NMC (Eq. 5.4) comes with trade-offs. IWNMC typically requires a large L to get a good estimate of the EIG. In high dimensions, this can be computationally infeasible. Having a small L on the other hand might result in a failure case if the effective sample size of importance samples becomes 1. We can alleviate this issue if there is some prior information available which could be leveraged to design better proposal distributions. This might consist of knowledge of certain causal mechanisms of the system under study or access to some initial observational data. In such a case,

a proposal distribution which encodes this information (for example with support on graphs which are in the Markov Equivalence Class (MEC) of the observational distribution) can be used instead of the prior. If no prior information is available or a good approximate inference technique is at our disposal, NMC is preferable in high dimensions. Surprisingly, we get good results on variables of size up to 5 with IWNMC from just the prior and up to 40 variables from a proposal distribution which has support on the MEC of observational distribution (see Sec 10).

Optimizing over Targets and States (`DiffCBED`)

While the NMC estimator provides a unified objective to directly optimize over the designs $\xi_{1:B}$, it requires that the design space is continuous so that the gradients $\frac{\partial \mathcal{U}_{\text{NMC}}}{\partial I_{1:B}}$ and $\frac{\partial \mathcal{U}_{\text{NMC}}}{\partial S^I_{1:B}}$ can be computed. However, in the case of designing experiments for causal models, the challenge still remains that optimizing over intervention targets I with gradient-based techniques is not possible because it is a discrete choice.

In order to address this problem, we introduce a *design policy* π_ϕ with learnable parameters ϕ that parameterize a joint distribution over possible intervention targets and corresponding states. Instead of seeking the gradients $\frac{\partial \mathcal{U}_{\text{NMC}}}{\partial I_{1:B}}$ and $\frac{\partial \mathcal{U}_{\text{NMC}}}{\partial S^I_{1:B}}$, the goal now instead is to estimate $\frac{\mathcal{U}_{\text{NMC}}}{\partial \phi}$ so that policy can be updated to be close to optimal. Such a characterization of the design space allows us to use continuous

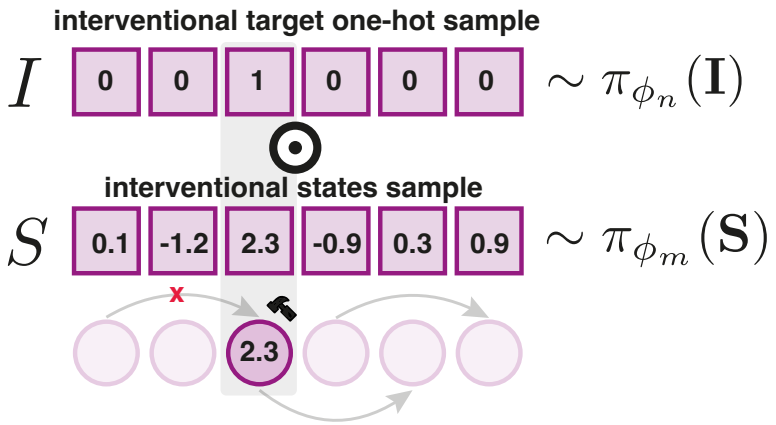


Figure 5.5: A design sample is obtained by first sampling $I_{1:B} \sim \pi_{\phi_n}(\mathbf{I})$, $S_{1:B} \sim \pi_{\phi_m}(\mathbf{S})$ and then setting states to be $S^I_{1:B} = S_{1:B} \odot I_{1:B}$. To obtain hard samples of I, we use the straight-through estimator (Bengio et al., 2013). Illustration for $B = 1$.

relaxations of discrete distributions (Maddison et al., 2016, Jang et al., 2016) to obtain samples of designs and estimate NMC gradients.

Let $\mathbf{I}$ and $\mathbf{S}$ be the random variables that model all possible intervention target combinations and states for a batch design respectively. While there are many possibilities of instantiating the policy in practice, we consider the simplest case where $\pi_\phi(\mathbf{I}, \mathbf{S}) \triangleq \pi_{\phi_n}(\mathbf{I})\pi_{\phi_m}(\mathbf{S})$. Other factorizations, such as $\pi_\phi(\mathbf{I}, \mathbf{S}) \triangleq \pi_{\phi_m}(\mathbf{S} \mid \mathbf{I})\pi_{\phi_n}(\mathbf{I})$ is possible, however, we found that the simple one was sufficient to demonstrate our method.. As the state space is continuous[2], π_{ϕ_m} can be either deterministic (a delta Dirac with $\phi_m \in \mathbb{R}^{B\times d}$) or Gaussian with $\phi_m \in \mathbb{R}^{2\times B\times d}$ parameterizing its mean and log variance. In this work, we found it sufficient to use a deterministic policy over the state space. For the interventional targets, $\phi_n \in \mathbb{R}^{B\times d}$ parameterizes the logits of different relaxed versions of discrete distributions depending on the setting, which we describe below.

Having established the basic structure of a policy, we can support different settings by structuring the policy over designs differently.

Algorithm 5: Differentiable CBED

Input : $\mathcal{E}$ SCM Environment, N Initial observational samples, B Batch Size

$\mathcal{D}_{\text{obs}} \leftarrow \mathcal{E}.\text{sample}(N)$, $\mathcal{D}_{\text{int}} \leftarrow \varnothing$

Train $q(\boldsymbol{\Theta} \mid \mathcal{D}_{\text{obs}}) \approx p(\boldsymbol{\Theta} \mid \mathcal{D}_{\text{int}})$ using appropriate algorithm.

for *batch* $t = 1 \dots T$ *Batches* **do**

  Initialize design policy parameters $\phi = \{\phi_n, \phi_m\}$: trainable logits ϕ_n for the targets; trainable parameters ϕ_m for the states.

  for *update step* $c = 1 \dots C$ **do**

    ▷ `Sample Interventional Targets and States`

    $\{\xi_{1:B}^{(o)}\}_{o=1}^{O} \sim \pi_\phi(\mathbf{I}, \mathbf{S})$

    ▷ `Gradient ascent with straight-through gradient estimator`

    Update $\phi \rightarrow \phi + \alpha \frac{\partial}{\partial \phi} \frac{1}{O} \sum_{o=1}^{O} \left[\mathcal{U}_{\text{NMC}}^{t}(\xi_{1:B}^{(o)})\right]$

  ▷ `Intervene with learned policy`

  $\xi_{1:B} \sim \pi_\phi$

  $\mathcal{D}_{\text{int}} \leftarrow \mathcal{D}_{\text{int}} \cup \mathcal{E}.\text{intervene}(\xi_{1:B})$

  Update the posterior $q(\boldsymbol{\Theta} \mid \mathcal{D}_{\text{obs}} \cup \mathcal{D}_{\text{int}})$.

[2]If the state space is discrete, optimizing π_{ϕ_m} would be similar to π_{ϕ_n} which involves reparameterized gradients.

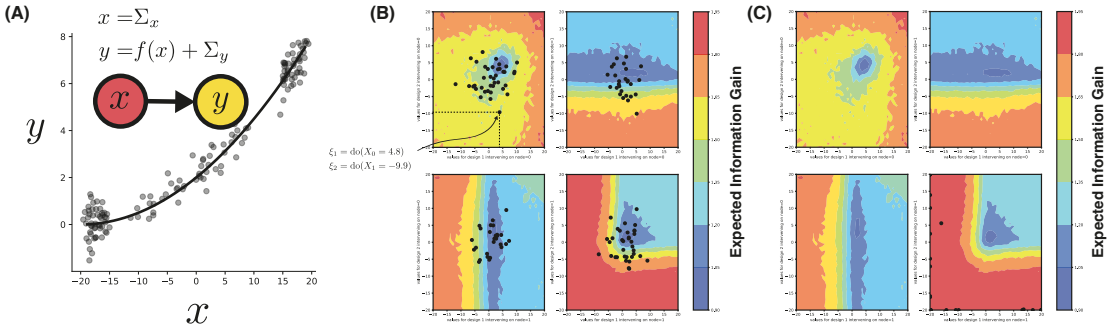


Figure 5.6: Two nodes and two experiments scenario. We assume a ground-truth graph G_{T} of two nodes $X \to Y$. The conditional distribution $p(Y \mid X)$ is shown in **(A)**. The corresponding SCM is $x = \Sigma_x$ and $y = f(x) + \Sigma_y$. The four panels represent the EIG of all possible experiments of batch size two, when intervening on nodes $(0,0), (0,1), (1,0), (1,1)$. Each panel shows how the EIG change on different interventional values. E.g. right top panel shows how EIG changes when applying interventions with values in ranges $[-20, 20]$. We can observe that the algorithm successfully places the designs (samples from the policy) on the high EIG (1.95) area of the plot (• on the plot).

Single Target ($q = 1$)

In this setting, the intervention targets are one-hot vectors, as demonstrated in Fig. 5.5. To sample one-hot vectors in a differentiable manner, we parametrize π_{ϕ_n} as a Gumbel-Softmax distribution (Maddison et al., 2016, Jang et al., 2016) over intervention targets, which is a continuous relaxation of the categorical distribution (in *one-hot* form). Additionally, we use the straight-through (ST) gradient estimator Bengio et al. (2013).

Unconstrained Multi-Target ($q \leq d$)

If instead of a continuous relaxation of the categorical distribution, we parametrise the policy π_{ϕ_n} as a continuous relaxation of the Bernoulli distribution (Binary Concrete) (Maddison et al., 2016), we can now sample multi-target experiments. Notice that since each interventional target sample will have at most d non-zero entries, this policy is suitable for multi-target experiments with an unconstrained number of interventions per experiment.

Constrained Multi-Target ($q = k$)

Finally, when considering a setting where the number of targets per intervention is exactly k. However, this is a significantly more challenging case, since the policy needs to select a subset of k from d nodes. By using a continuous relaxation of subset sampling, as introduced in work by Xie and Ermon (2019), combined with straight-through gradient estimator, we can efficiently optimize the policy to select a subset of nodes to intervene on.

Experiments

We evaluate the performance of our method on synthetic graphs and a range of baselines. We aim to investigate the following aspects empirically: (1) To what extent can we design good experiments without performing intermediate causal discovery/ posterior estimation with IWNMC estimator from the prior? (2) Ability to design good experiments with a proposal distribution with IWNMC (3) the performance of our policy-based design in combination with the differentiable NMC estimator in single-target and multi-target settings, as compared to suitable baselines.

Bivariate Setting

First, we demonstrate the method in a two nodes graph to qualitatively assess what the objective and the optimization method do. Since computing the posterior over graphs and parameters is intractable in the general case, as a first step to study how well we can optimize the EIG objectives, we assume a simple two-nodes SCM. To compute the posterior we enumerate all the graphs of size two and additionally, we parametrize the conditional distributions as Neural Network parametrised Gaussian distribution ($\mathcal{N}(X_i; \mu_{\text{NN}}(X_{\text{pa}(i)}), \sigma_{\text{NN}}(X_{\text{pa}(i)}))$), and we compute the posterior over the parameters of the conditional distributions via Monte-Carlo dropout (Gal and Ghahramani, 2016). We parametrize the intervention targets policy with Gumbel-Softmax and interventional values policy with a Gaussian distribution. The final

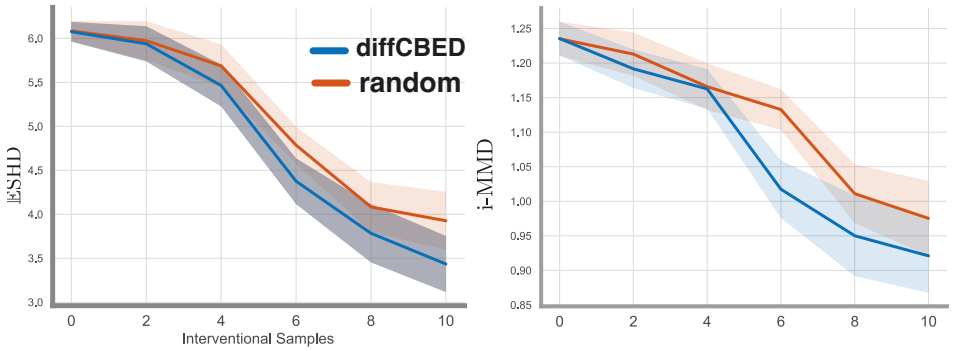


Figure 5.7: We test the designs acquired with IWNMC estimator with just the prior as opposed to the random policy (with random target and state acquisition) on variables of 5 dimensions. Plots correspond to unconstrained multi-target setting with $B = 2$ (shaded area represents 95% confidence intervals - 60 seeds).

policy consists of the logits of the Gumbel-Softmax and the sufficient statistics of the Gaussian distribution. We use Adam optimizer (Kingma and Ba, 2014) to optimize the parameters of the policy. As we can see in Fig.5.6(B-C), the optimizer successfully concentrates the policy on the nodes and values that maximize the EIG objective.

Results

Evaluation metrics

Expected SHD: This metric evaluates the expected structural hamming distance over the graphs sampled from the posterior model and the ground truth graph.

Expected F1-score: This metric evaluates the expected f1-scores over the edges of the adjacency matrices sampled from the posterior model and the ground truth graph.

i-MMD: interventional MMD distance uses the non-parametric distance metric MMD Gretton et al. (2012). Contrary to the graph evaluation metrics, this metric is evaluating the distributions induced by both the graph structure and the conditional distributions. We provide the full definition in Appendix E.

Evaluation of the IWNMC estimator

In this section, we consider optimizing the designs with respect to the IWNMC estimator entirely from the prior, introduced in 9, sidestepping the causal discovery procedure. As noted before, estimating posteriors of causal models is hard, so it is important to understand to what extent IWNMC can be considered a suitable candidate for designing good experiments in the absence of a posterior. For this setting, we sample from the prior distribution over graphs by first sampling an ordering of nodes at random and then sampling edges with probability $p = 0.25$ which adhere to this topological order to give a DAG. We sample the mechanism parameters and noise variances of ANM at random from a Gaussian distribution with mean 0 and variance 1. Figure 5.7 demonstrates results for 5 variable unconstrained multi-target setting with batch size 2. For evaluation, we train DAG Bootstrap (Friedman et al., 2013) with GIES (Hauser and Bühlmann, 2012) on the data acquired from each policy. We can see that we can recover the ground truth SCM faster than a random strategy. This is a surprising, but positive result given that our policy was trained entirely from samples from the prior. We also tested this approach for 10 nodes (results in Appendix E). While this resulted in better performance of the policy as opposed to random in terms of downstream metrics, we observed effective sample size reach 1 indicating that for 10 dimensions or higher, indicating that we might need a better proposal distribution or a posterior estimate.

Baselines

Before we evaluate the IWNMC estimator with a proposal distribution more informative than the prior and the NMC estimator with a posterior estimate of SCM, we present the baselines with which we can also compare the overall performance of our designs.

Single target

Random-Fixed: Uniform random selection of node, fixing the state to a value of 0 (as introduced in Agrawal et al. (2019), Tigas et al. (2022)). **Random-Random**:

Uniform selection of node, uniform selection of state (introduced in Toth et al. (2022)). **SoftCBED**: A stochastic approximation of greedy batch selection as introduced in Tigas et al. (2022).

Multi-Target

Random-Random: Multitarget version of Uniform selection of node, uniform selection of value (introduced in Toth et al. (2022)). **Random-Fixed**: Multitarget version of Uniform selection of node, fixed value to 5 Sussex et al. (2021), as suggested by the authors. **SSGb**: Finite sample baseline from Sussex et al. (2021) with fixed value equal 5. We emphasize that in contrast to our method, the baselines cannot select values, but they either assume a fixed predefined value or select a value at random.

Evaluation in Higher Dimensions

Evaluation of IWNMC with Proposal Distribution

In this experiment, we consider 40 variables, constrained ($q = 5$) multi-target and batch size $B = 2$. Further, we use the same setup as Sussex et al. (2021) to make a fair comparison as well as to construct a proposal distribution. To construct a proposal distribution, we use 800 observational samples to train DAG Bootstrap (Friedman et al., 2013, Agrawal et al., 2019) and augment our posterior samples with samples of dags from the Markov Equivalence Class of the true graph, to make sure that there is support over the graphs from the MEC of the true graph (see Sussex et al. (2021) for details). We then acquire a single batch of experiments from IWNMC estimator for our approach. For the baseline, we acquire a single batch of experiments from the estimator defined in (Sussex et al., 2021).

For random and Sussex et al. (2021) baseline, we set the interventional value to 5, as explained in Sussex et al. (2021). Notice that in contrast to the baselines, our approach doesn't fix the value to 5 but optimizes over a value to perform the intervention with. In Table 5.2 we summarize our results. As we can see, our method outperforms random and SSGb (Sussex et al., 2021) by a great margin,

Table 5.2: Results of multi-target experiments on graphs of size 40 (30 seeds ± s.e.). Similarly to Sussex et al. (2021), we are using posterior samples trained on observational data and re-weighting them with likelihoods.

Method	$\mathbb{E}$SHD ↓	F1 ↑	iMMD ↓
Random	43.78±46.67	0.91±0.08	0.16±0.07
SSGb	15.59±29.66	0.97±0.05	0.10±0.06
diffCBED	0.44±0.21	0.99±0.00	0.07±0.01

indicating that with a good proposal distribution, IWNMC can still be a promising candidate in higher dimensions.

Results with NMC estimator

For the following results, we use DAG-Bootstrap (Agrawal et al., 2019) with 20 components, an approximate posterior method based on GIES causal discovery method (Hauser and Bühlmann, 2012). As GIES is not a differentiable method, once we compute the posterior via the DAG-Bootstrap algorithm, we transfer the weights of the posterior samples (the bootstraps) into JAX (jax) tensors to allow for the gradients to be computed with respect to the experiments.

Single-target synthetic graphs: In this experiment, we test against synthetic graphs of 50 nodes and batch size 5, where the graph is sampled from the class of Erdos-Renyi (a common benchmark in the literature (Tigas et al., 2022, Toth et al., 2022, Scherrer et al., 2021)). In Fig. 5.8 **(A,B,C)** we summarize the results. We observe that our method performs significantly better than the baselines.

20 **nodes, unconstrained** ($q \leq 20$)**, batch size** $B = 2$**:** In this experiment, we want to evaluate the performance of our method as compared with the baselines, on sparse graphs over several acquisitions. Fig. 5.8 **(D,E,F)** summarizes the results of this setting. We observe strong empirical performance as compared to all the baselines.

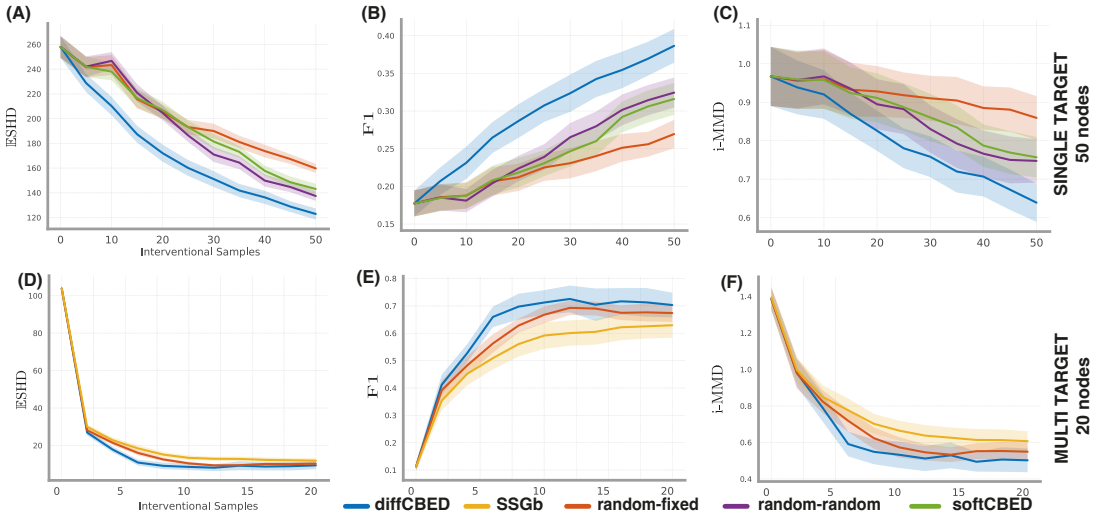


Figure 5.8: **(A,B,C)** Single target-state design setting results for ErdsRényi Erdős and Rényi (1959) graphs with $d = 50$ variables. **(D,E,F)** Multi target-state design setting results for ErdsRényi Erdős and Rényi (1959) graphs with $d = 20$ variables. Each experiment was run with 30 random seeds (shaded area represents 95% CIs)

Related Work

Variational and Amortized Methods. Huan and Marzouk (2014), Foster et al. (2019, 2020), Kleinegesse and Gutmann (2020, 2021) developed a unified framework for estimating the expected information gain (EIG) objective by optimizing variational bounds of the mutual information with gradient-based methods. In Ivanova et al. (2021), Foster et al. (2021), the authors introduced a policy-based method for performing sequential experimentation, amortizing the cost of estimating and optimizing the mutual information objective. More recently, works by Blau et al. (2022) and Lim et al. (2022) used reinforcement learning to train policies for adaptive experimental design. In Jain et al. (2023) the authors suggest the use of GFLowNets (Bengio et al., 2021) to learn a policy for selecting discrete designs and similarly to Ivanova et al. (2021), Foster et al. (2021) amortize the expensive cost of estimating the mutual information objectives.

Experimental Design for Causal Discovery. One of the earliest works of experimental design for causal discovery in a BOED setting was proposed by (Murphy, 2001) and (Tong and Koller, 2001) in the case of discrete variables for

single target acquisition. Since then, a number of works have attempted to address this problem for continuous variables in both the BOED framework (Agrawal et al., 2019, von Kügelgen et al., 2019, Toth et al., 2022, Cho et al., 2016) and other frameworks (Kocaoglu et al., 2017a, Gamella and Heinze-Deml, 2020, Eberhardt et al., 2012, Lindgren et al., 2018, Mokhtarian et al., 2022, Ghassami et al., 2018, Olko et al., 2022, Scherrer et al., 2021). In contrast to the setting studied in this paper, of particular note, are the approaches for experimental design for causal discovery in a non-BOED setting in the presence of cycles (Mokhtarian et al., 2022) and latent variables (Kocaoglu et al., 2017b). Closer to our BOED setting are the approaches of Tigas et al. (2022) and Sussex et al. (2021). Specifically, in (Tigas et al., 2022), the authors introduce a method for selecting single target-state pair with stochastic batch acquisition while Sussex et al. (2021) introduce a method for selecting a batch of multi-target experiments with a greedy strategy, based on a gradient-based approximation to mutual information, without selecting the intervention state. Our presented method in contrast can acquire a batch of multi-target-state pairs.

Discussion

Limitations: A primary limitation of our method is that it needs to estimate a posterior after every acquisition. While the proposed IWNMC estimator presents an interesting alternative, the designs are still non-adaptive. As demonstrated by Foster et al. (2021), Ivanova et al. (2021), a promising and exciting direction is to train a policy to be adaptive and propose new experiments in *real-time*.

Conclusion: We presented a gradient-based method for differentiable Bayesian Optimal Experimental for causal discovery. Our method allows not only for single-target but also various multi-target (constrained and unconstrained) batch acquisition of experiments. While prior work in Causal Bayesian Experimental Design relies on greedy approximations for the selection of a batch Agrawal et al. (2019), Tigas et al. (2022) or black-box methods Toth et al. (2022), Tigas et al. (2022) for

optimizing over interventional states, our method utilizes gradient-based optimization procedures to simultaneously optimize for various design choices. Evaluation on different benchmarks suggests that our method is competitive with baselines.

6
Afterword

This thesis contributed to solving the challenge of decision-making under epistemic uncertainty on a few fronts. First, using the epistemic uncertainty to plan robustly and safely in settings where safety is crucial. We proposed methods for robust planning and developed benchmarks to help the community advance in the safety-first field of self-driving cars. Additionally, departing from the field of self-driving cars, we steered our focus on personalised healthcare and self-driving labs and we introduced solutions to the problem of learning efficient causal models. By making use of Bayesian Optimal Experimental Design, Causal Bayesian Networks and Deep Learning, we proposed methods for efficient causal discovery, aiming for challenging fields like Gene Regulatory Networks discovery.

We began this thesis with Plato's allegory of the cave, and we are closing with a similar remark. The prisoners of the story lived in the world of appearances trying to break free from the "curse" of partial observability and the sensory curtain that separates the agents from the true world. We hope with this thesis, we contributed to helping the prisoners see behind the shadows of causality or at least help them make their journey a little bit safer.

Bibliography

Jax: Autograd and XLA. https://github.com/google/jax.

M. Abolhasani and E. Kumacheva. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis*, pages 1–10, 2023.

J. Abrevaya, Y.-C. Hsu, and R. P. Lieli. Estimating conditional average treatment effects. *Journal of Business & Economic Statistics*, 33(4):485–505, 2015.

A. F. Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

R. Agrawal, C. Squires, K. Yang, K. Shanmugam, and C. Uhler. Abcd-strategy: Budgeted experimental design for targeted causal structure discovery. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3400–3409. PMLR, 2019.

I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al. Solving Rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

A. M. Alaa and M. van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Advances in Neural Information Processing Systems*, pages 3424–3432, 2017.

A. M. Alaa and M. van der Schaar. Bayesian nonparametric causal inference: Information rates and learning algorithms. *IEEE Journal of Selected Topics in Signal Processing*, 12(5):1031–1046, 2018.

D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Y. Annadani, J. Rothfuss, A. Lacoste, N. Scherrer, A. Goyal, Y. Bengio, and S. Bauer. Variational causal networks: Approximate bayesian inference over causal structures. *arXiv preprint arXiv:2106.07635*, 2021.

W. R. Ashby. *An Introduction to Cybernetics*. Chapman & Hall Ltd, 1961.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286 (5439):509–512, 1999.

D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

V. Batagelj and U. Brandes. Efficient generation of large random networks. *Physical Review E*, 71(3):036113, 2005.

R. E. Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 2015.

Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Y. Bengio, T. Deleu, N. Rahaman, R. Ke, S. Lachapelle, O. Bilaniuk, A. Goyal, and C. Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Y. Bengio, S. Lahlou, T. Deleu, E. J. Hu, M. Tiwari, and E. Bengio. Gflownet foundations. *arXiv preprint arXiv:2111.09266*, 2021.

J. Bergstra, D. Yamins, and D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/bergstra13.html.

C. M. Bishop. Mixture density networks. 1994.

T. Blau, E. V. Bonilla, I. Chades, and A. Dezfouli. Optimizing sequential experimental design with deep reinforcement learning. In *International Conference on Machine Learning*, pages 2107–2128. PMLR, 2022.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

G. E. Box. Sampling and bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society: Series A (General)*, 143(4):383–404, 1980.

P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien, and A. Drouin. Differentiable causal discovery from interventional data. *arXiv preprint arXiv:2007.01754*, 2020.

H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

P. S. Castro and D. Precup. Using bisimulation for policy transfer in MDPs. In *AAAI Conference on Artificial Intelligence*, 2010.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. *arXiv preprint arXiv:1912.12294*, 2019.

H. Cho, B. Berger, and J. Peng. Reconstructing causal biological networks through active learning. *PloS one*, 11(3):e0150611, 2016.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, 2017.

K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Neural Information Processing Systems (NeurIPS)*, pages 4754–4765, 2018.

J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. In *International conference on machine learning*, pages 2067–2075. PMLR, 2015.

D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network

learning by exponential linear units (elus). In *International conference on machine learning*, pages 448–456. PMLR, 2016.

F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *International Conference on Computer Vision (ICCV)*, pages 9329–9338, 2019.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine learning*, 15:201–221, 1994.

D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

G. Coley, A. Wesley, N. Reed, and I. Parry. Driver reaction times to familiar, but unexpected events. *TRL Published Project Report*, 2009.

C. Cronrath, E. Jorge, J. Moberg, M. Jirstrand, and B. Lennartson. BAgger: A Bayesian algorithm for safe and query-efficient imitation learning. https://personalrobotics.cs.washington.edu/workshops/mlmp2018/assets/docs/24_CameraReadySubmission_180928_BAgger.pdf, 2018.

H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019.

C. Cundy, A. Grover, and S. Ermon. Bcd nets: Scalable variational approaches for bayesian causal discovery. *Advances in Neural Information Processing Systems*, 34, 2021.

A. D'Amour and A. Franks. Deconfounding scores: Feature representations for causal effect estimation with weak overlap. *arXiv preprint arXiv:2104.05762*, 2021.

P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz Machine. *Neural Computation*, 7(5):889–904, Sept. 1995. ISSN 0899-7667, 1530-888X. doi: 10.1162/neco.1995.7.5.889.

P. de Haan, D. Jayaraman, and S. Levine. Causal confusion in imitation learning. In *Neural Information Processing Systems (NeurIPS)*, pages 11693–11704, 2019.

T. Deleu, A. Góis, C. Emezue, M. Rankawat, S. Lacoste-Julien, S. Bauer, and Y. Bengio. Bayesian structure learning with generative flow networks. *arXiv preprint arXiv:2202.13903*, 2022.

K. Deng, J. Pineau, and S. Murphy. Active learning for personalizing treatment. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 32–39. IEEE, 2011.

D. C. Dennett. Intentional systems. *The Journal of Philosophy*, pages 87–106, 1971.

A. Der Kiureghian and O. Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

Y. Du, T. Lin, and I. Mordatch. Model based planning with energy based models. *arXiv preprint arXiv:1909.06878*, 2019.

A. DAmour, P. Ding, A. Feller, L. Lei, and J. Sekhon. Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*, 221(2):644–654, 2021.

F. Eberhardt, C. Glymour, and R. Scheines. On the number of experiments sufficient

and in the worst case necessary to identify all causal relations among n variables. *arXiv preprint arXiv:1207.1389*, 2012.

P. Embrechts, C. Klüppelberg, and T. Mikosch. *Modelling extremal events: for insurance and finance*, volume 33. Springer Science & Business Media, 2013.

P. Erdős and A. Rényi. On random graphs i. publicationes mathematicae (debrecen). 1959.

S. Farquhar, Y. Gal, and T. Rainforth. On statistical bias in active learning: How and when to fix it. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=JiYq3eqTKY.

A. Filos, P. Tigkas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2020.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pages 1126–1135, 2017.

S. Fort, H. Hu, and B. Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

A. Foster, M. Jankowiak, E. Bingham, P. Horsfall, Y. W. Teh, T. Rainforth, and N. Goodman. Variational bayesian optimal experimental design. *arXiv preprint arXiv:1903.05480*, 2019.

A. Foster, M. Jankowiak, M. OMeara, Y. W. Teh, and T. Rainforth. A unified stochastic gradient approach to designing bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR, 2020.

A. Foster, D. R. Ivanova, I. Malik, and T. Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. *arXiv preprint arXiv:2103.02438*, 2021.

P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

R. M. French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with bayesian networks: A bootstrap approach. *arXiv preprint arXiv:1301.6695*, 2013.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

J. L. Gamella and C. Heinze-Deml. Active invariant causal prediction: Experiment selection through stability. *arXiv preprint arXiv:2006.05690*, 2020.

Z. Gao and Y. Han. Minimax optimal nonparametric estimation of heterogeneous treatment effects. In *Advances in Neural Information Processing Systems*, 2020.

A. Ghassami, S. Salehkaleybar, N. Kiyavash, and E. Bareinboim. Budgeted experiment design for causal structure learning. In *International Conference on Machine Learning*, pages 1724–1733. PMLR, 2018.

H. Gouk, E. Frank, B. Pfahringer, and M. Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

A. Graves. Practical variational inference for neural networks. In *Neural Information Processing Systems (NeurIPS)*, pages 2348–2356, 2011.

A. Graves and A. Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

K. Greenewald, D. Katz, K. Shanmugam, S. Magliacane, M. Kocaoglu, E. Boix Adsera, and G. Bresler. Sample efficient active learning of causal trees. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Greenfield, A. Madar, H. Ostrer, and R. Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10):e13397, 2010.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

F. Häse, L. M. Roch, and A. Aspuru-Guzik. Next-generation experimentation with self-driving laboratories. *Trends in Chemistry*, 1(3):282–291, 2019.

A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Y.-B. He and Z. Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547, 2008.

J. Hensman, A. Matthews, and Z. Ghahramani. Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

J. M. Hernández-Lobato and R. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning (ICML)*, pages 1861–1869, 2015.

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

P. W. Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.

N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *stat*, 1050:24, 2011.

X. Huan and Y. Marzouk. Gradient-based stochastic optimization methods in bayesian experimental design. *International Journal for Uncertainty Quantification*, 4(6), 2014.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

D. R. Ivanova, A. Foster, S. Kleinegesse, M. U. Gutmann, and T. Rainforth. Implicit deep adaptive design: policy-based experimental design without likelihoods. *Advances in Neural Information Processing Systems*, 34:25785–25798, 2021.

M. Jain, T. Deleu, J. Hartford, C.-H. Liu, A. Hernandez-Garcia, and Y. Bengio. Gflownets for ai-driven scientific discovery. *Digital Discovery*, 2(3):557–577, 2023.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

A. Jesson, S. Mindermann, U. Shalit, and Y. Gal. Identifying causal-effect inference failure with uncertainty-aware models. *Advances in Neural Information Processing Systems*, 33, 2020.

A. Jesson, S. Mindermann, Y. Gal, and U. Shalit. Quantifying ignorance in individual-level causal-effect estimates under hidden confounding. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages

4829–4838. PMLR, 2021. URL https://proceedings.mlr.press/v139/jesson21a.html.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.

N. Kallus, X. Mao, and A. Zhou. Interval estimation of individual-level causal effects under unobserved confounding. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2281–2290. PMLR, 2019.

N. R. Ke, O. Bilaniuk, A. Goyal, S. Bauer, H. Larochelle, B. Schölkopf, M. C. Mozer, C. Pal, and Y. Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

Z. Kenton, A. Filos, O. Evans, and Y. Gal. Generalizing from a few environments in safety-critical reinforcement learning. *arXiv preprint arXiv:1907.01475*, 2019.

R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 av dataset 2019, 2019. URL https://level5.lyft.com/dataset/.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014.

A. Kirsch, J. Van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32:7026–7037, 2019.

A. Kirsch, S. Farquhar, and Y. Gal. A simple baseline for batch active learning with stochastic acquisition functions. *arXiv preprint arXiv:2106.12059*, 2021.

S. Kleinegesse and M. Gutmann. Bayesian experimental design for implicit models by mutual information neural estimation. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 5316–5326. PMLR, 2020.

S. Kleinegesse and M. U. Gutmann. Gradient-based bayesian experimental design for implicit models using mutual information lower bounds. *arXiv preprint arXiv:2105.04379*, 2021.

M. Kocaoglu, A. Dimakis, and S. Vishwanath. Cost-optimal learning of causal graphs. In *International Conference on Machine Learning*, pages 1875–1884. PMLR, 2017a.

M. Kocaoglu, K. Shanmugam, and E. Bareinboim. Experimental design for learning causal graphs with latent variables. *Advances in Neural Information Processing Systems*, 30, 2017b.

A. Krause and C. E. Guestrin. Near-optimal nonmyopic value of information in graphical models. *arXiv preprint arXiv:1207.1394*, 2012.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. 1964.

S. Lahlou, M. Jain, H. Nekoei, V. I. Butoi, P. Bertin, J. Rector-Brooks, M. Korablyov, and Y. Bengio. Deup: Direct epistemic uncertainty prediction. *arXiv preprint arXiv:2102.08501*, 2021.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Neural Information Processing Systems (NeurIPS)*, pages 6402–6413, 2017.

P. S. Laplace. Essai philosophique sur les probabilités forming the introduction to his théorie analytique des probabilités. v courcier (1820); repr. 1820.

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

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

J. Leike, M. Martic, V. Krakovna, P. A. Ortega, T. Everitt, A. Lefrancq, L. Orseau, and S. Legg. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

Z. Li, T. Motoyoshi, K. Sasaki, T. Ogata, and S. Sugano. Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability. *arXiv preprint arXiv:1809.11100*, 2018.

X. Liang, T. Wang, L. Yang, and E. Xing. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *European Conference on Computer Vision (ECCV)*, pages 584–599, 2018.

R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

V. Lim, E. Novoseller, J. Ichnowski, H. Huang, and K. Goldberg. Policy-based bayesian experimental design for non-differentiable implicit models. *arXiv preprint arXiv:2203.04272*, 2022.

E. Lindgren, M. Kocaoglu, A. G. Dimakis, and S. Vishwanath. Experimental design for cost-aware learning of causal graphs. *Advances in Neural Information Processing Systems*, 31, 2018.

D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005, 1956.

C. Liu and C. G. Atkeson. Standing balance control using a trajectory library. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3031–3036. IEEE, 2009.

J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems*, pages 2378–2386, 2016.

L. Lorch, J. Rothfuss, B. Schölkopf, and A. Krause. Dibs: Differentiable bayesian structure learning. *arXiv preprint arXiv:2105.11839*, 2021.

C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

P. S. Marquis de Laplace. *A philosophical essay on probabilities*. Wiley, 1902.

R. McAllister, G. Kahn, J. Clune, and S. Levine. Robustness to out-of-distribution inputs via task-aware generative uncertainty. In *International Conference on*

Robotics and Automation (ICRA), pages 2083–2089. IEEE, 2019.

G. Mena, D. Belanger, S. Linderman, and J. Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.

R. Michelmore, M. Kwiatkowska, and Y. Gal. Evaluating uncertainty quantification in end-to-end autonomous driving control. *arXiv preprint arXiv:1811.06817*, 2018.

T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

J. Močkus. On bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pages 400–404. Springer, 1975.

E. Mokhtarian, S. Salehkaleybar, A. Ghassami, and N. Kiyavash. A unified experiment design approach for cyclic and acyclic causal models. *arXiv preprint arXiv:2205.10083*, 2022.

O. Morgenstern and J. Von Neumann. *Theory of Games and Economic Behavior*. Princeton university press, 1953.

K. P. Murphy. Active learning of causal bayes net structure. 2001.

J. I. Myung, D. R. Cavagnaro, and M. A. Pitt. A tutorial on adaptive design optimization. *Journal of mathematical psychology*, 57(3-4):53–67, 2013.

National Highway Traffic Safety Administration. Pre-crash scenario typology for crash avoidance research, 2007. URL https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/pre-crash_scenario_typology-final_pdf_version_5-2-07.pdf.

R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical programming*, 14(1):265–294, 1978.

R. O. Ness, K. Sachs, P. Mallick, and O. Vitek. A bayesian active learning experimental design for inferring signaling networks. In *International Conference on Research in Computational Molecular Biology*, pages 134–156. Springer, 2017.

M. Nishikawa-Toomey, T. Deleu, J. Subramanian, Y. Bengio, and L. Charlin. Bayesian learning of causal structure and mechanisms with gflownets and variational bayes. *arXiv preprint arXiv:2211.02763*, 2022.

S. U. Noble. *Algorithms of oppression: How search engines reinforce racism*. NYU Press, 2018.

M. Olko, M. Zając, A. Nowak, N. Scherrer, Y. Annadani, S. Bauer, Ł. Kuciński, and P. Miłoś. Trust your ∇: Gradient-based intervention targeting for causal discovery. *arXiv preprint arXiv:2211.13715*, 2022.

M. A. OpenAI, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, et al. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.

J. Pearl. Bayesian netwcrks: A model cf self-activated memory for evidential reasoning. In *Proceedings of the 7th conference of the Cognitive Science Society, University of California, Irvine, CA, USA*, pages 15–17, 1985.

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

J. Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2), 2000.

C. C. Perez. *Invisible women: Exposing data bias in a world designed for men*. Random House, 2019.

J. Peters and P. Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural computation*, 27(3):771–799, 2015.

J. Peters, J. Mooij, D. Janzing, and B. Schölkopf. Identifiability of causal graphs using functional models. *arXiv preprint arXiv:1202.3757*, 2012.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012, 2016.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

M. L. Petersen, K. E. Porter, S. Gruber, Y. Wang, and M. J. Van Der Laan. Diagnosing and responding to violations in the positivity assumption. *Statistical methods in medical research*, 21(1):31–54, 2012.

T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. *arXiv preprint arXiv:1911.10298*, 2019.

T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020.

Plato. *Plato: Republic V*. Liverpool University Press, 2 edition, 375 BCE. ISBN 9780856685361. URL http://www.jstor.org/stable/j.ctv1228gz0.

D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Neural Information Processing Systems (NeurIPS)*, pages 305–313, 1989.

B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

T. Qin, T.-Z. Wang, and Z.-H. Zhou. Budgeted heterogeneous treatment effect estimation. In *International Conference on Machine Learning*, pages 8693–8702. PMLR, 2021.

J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. MIT Press, 2009.

T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood. On nesting monte carlo estimators. In *International Conference on Machine Learning*, pages 4267–4276. PMLR, 2018a.

T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood. On Nesting Monte Carlo Estimators. *arXiv:1709.06181 [stat]*, May 2018b.

A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

H. Reichenbach. *The direction of time*, volume 65. Univ of California Press, 1956.

D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015a.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015b.

N. Rhinehart, K. M. Kitani, and P. Vernaza. R2P2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *European Conference on Computer Vision (ECCV)*, pages 772–788, 2018.

N. Rhinehart, R. McAllister, K. Kitani, and S. Levine. PRECOG: Prediction conditioned on goals in visual multi-agent settings. *International Conference on Computer Vision*, 2019a.

N. Rhinehart, R. McAllister, K. Kitani, and S. Levine. PRECOG: PREdiction Conditioned On Goals in Visual Multi-Agent Settings. *arXiv:1905.01296 [cs, stat]*, Sept. 2019b.

N. Rhinehart, R. McAllister, and S. Levine. Deep imitative models for flexible inference, planning, and control. In *International Conference on Learning Representations (ICLR)*, April 2020.

J. M. Robins, M. A. Hernán, and B. Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):551, 2000.

G. Ros, V. Koltun, F. Codevilla, and M. A. Lopez. CARLA challenge, 2019. URL https://carlachallenge.org.

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

A. Rosenblueth, N. Wiener, and J. Bigelow. Behavior, purpose and teleology. *Philosophy of science*, 10(1):18–24, 1943.

S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Artificial Intelligence and Statistics (AISTATS)*, pages 627–635, 2011.

D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

S. Russell and P. Norvig. Artificial intelligence: A modern approach. 2002.

K. J. Ryan. Estimating expected information gains for experimental designs with application to the random fatigue-limit model. *Journal of Computational and Graphical Statistics*, 12(3):585–603, 2003.

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308 (5721):523–529, 2005.

F. Sadeghi and S. Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

A. Sauer, N. Savinov, and A. Geiger. Conditional affordance learning for driving in urban environments. *arXiv preprint arXiv:1806.06498*, 2018.

T. Schaffter, D. Marbach, and D. Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

N. Scherrer, O. Bilaniuk, Y. Annadani, A. Goyal, P. Schwab, B. Schölkopf, M. C. Mozer, Y. Bengio, S. Bauer, and N. R. Ke. Learning neural causal models with active interventions. *arXiv preprint arXiv:2109.02429*, 2021.

M. Schmidt, A. Niculescu-Mizil, K. Murphy, et al. Learning graphical model structure using l1-regularization paths. In *AAAI*, volume 7, pages 1278–1283, 2007.

B. Settles. Active learning literature survey. 2009.

U. Shalit, F. D. Johansson, and D. Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR, 2017.

K. Shanmugam, M. Kocaoglu, A. G. Dimakis, and S. Vishwanath. Learning causal graphs with small interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

C. Shi, D. Blei, and V. Veitch. Adapting neural networks for the estimation of treatment effects. *Advances in Neural Information Processing Systems*, 32:2507–2517, 2019.

L. Smith and Y. Gal. Understanding measures of uncertainty for adversarial example detection. *Uncertainty in Artificial Intelligence*, 2018.

J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, and Z. Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Neural Information Processing Systems (NeurIPS)*, pages 13969–13980, 2019.

J. Splawa-Neyman, D. M. Dabrowska, and T. P. Speed. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, pages 465–472, 1990.

C. Squires, S. Magliacane, K. Greenewald, D. Katz, M. Kocaoglu, and K. Shanmugam. Active structure learning of causal dags via directed clique trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020.

N. Srinivas, A. Krause, S. Kakade, and M. W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML*, 2010.

C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray, et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *Plos med*, 12(3):e1001779, 2015.

M. Sugiyama and M. Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: An open dataset benchmark. *arXiv preprint arXiv:1912.04838*, 2019.

I. Sundin, P. Schulam, E. Siivola, A. Vehtari, S. Saria, and S. Kaski. Active learning for decision-making from imbalanced observational data. In *International Conference on Machine Learning*, pages 6046–6055. PMLR, 2019.

S. Sussex, A. Krause, and C. Uhler. Near-optimal multi-perturbation experimental design for causal structure learning. *arXiv preprint arXiv:2105.14024*, 2021.

Y. C. Tang, J. Zhang, and R. Salakhutdinov. Worst cases policy gradients. *arXiv preprint arXiv:1911.03618*, 2019.

G. T. Taoka. Brake reaction times of unalerted drivers. *ITE journal*, 59(3):19–21, 1989.

M. Thomas and A. T. Joy. *Elements of information theory*. Wiley-Interscience, 2006.

L. Tian, A. A. Alizadeh, A. J. Gentles, and R. Tibshirani. A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association*, 109(508):1517–1532, 2014.

P. Tigas, Y. Annadani, A. Jesson, B. Schölkopf, Y. Gal, and S. Bauer. Interventions, where and how? experimental design for causal models at scale. *arXiv preprint arXiv:2203.02016*, 2022.

S. Tong and D. Koller. Active learning for structure in bayesian networks. In *International joint conference on artificial intelligence*, volume 17, pages 863–869.

Citeseer, 2001.
C. Toth, L. Lorch, C. Knoll, A. Krause, F. Pernkopf, R. Peharz, and J. von Kügelgen. Active bayesian causal inference. *arXiv preprint arXiv:2206.02063*, 2022.
B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220, 2016.
J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021a.
J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal. Improving deterministic uncertainty estimation in deep learning for classification and regression. *arXiv preprint arXiv:2102.11409*, 2021b.
A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, page 255270, USA, 1990. Elsevier Science Inc. ISBN 0444892648.
H. Von Helmholtz. *Handbuch der physiologischen Optik*, volume 9. Voss, 1867.
J. von Kügelgen, P. K. Rubenstein, B. Schölkopf, and A. Weller. Optimal experimental design via bayesian optimization: active causal structure learning for gaussian process networks. *arXiv preprint arXiv:1910.03962*, 2019.
A. Wald. Contributions to the theory of statistical estimation and testing hypotheses. *The Annals of Mathematical Statistics*, 10(4):299–326, 1939.
B. Widrow and F. W. Smith. Pattern-recognizing control systems, 1964.
N. Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. Technology Press, 1948.
A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.
S. M. Xie and S. Ermon. Reparameterizable subset sampling via continuous relaxations. *arXiv preprint arXiv:1901.10517*, 2019.
Y. Xie, J. E. Brand, and B. Jann. Estimating heterogeneous treatment effects with observational data. *Sociological methodology*, 42(1):314–347, 2012.
J. Zhang and K. Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.
J. Zhang, C. Squires, and C. Uhler. Matching a desired causal state via shift interventions. *Advances in Neural Information Processing Systems*, 34:19923–19934, 2021.
X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. Dags with no tears: Continuous optimization for structure learning. *arXiv preprint arXiv:1803.01422*, 2018.
A. Zhilinskas. Single-step bayesian search method for an extremum of functions of a single variable. *Cybernetics*, 11(1):160–166, 1975.
A. Zhou, E. Jang, D. Kappler, A. Herzog, M. Khansari, P. Wohlhart, Y. Bai, M. Kalakrishnan, S. Levine, and C. Finn. Watch, try, learn: Meta-learning from demonstrations and reward. *arXiv preprint arXiv:1906.03352*, 2019.

Appendices

A

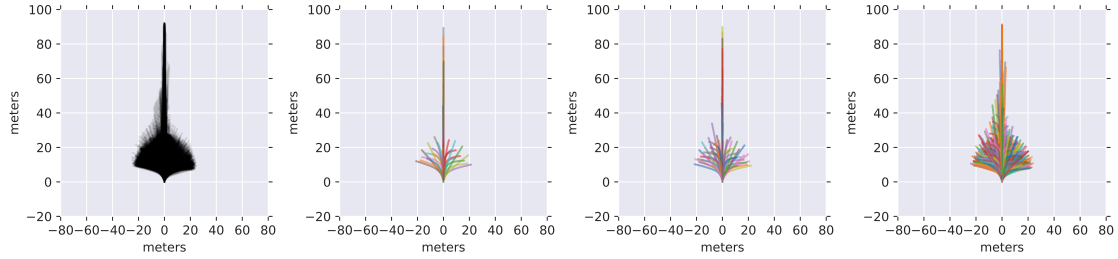
Robust Imitative Planning appendix

Online Planning with a Trajectory Library

In the absence of scalable global optimizers, we search the trajectory space in Eqn. (3.4) by restricting the search space to a trajectory library ((Liu and Atkeson, 2009)), $\mathcal{T}_{\mathbf{Y}}$, a finite set of fixed trajectories. In this work, we perform K-means clustering of the expert plan's from the training distribution and keep 64 of the centroids, as illustrated in Figure A.1. Therefore we efficiently solve a search problem over a discrete space rather than an optimization problem of continuous variables. The modified objective is:

$$\mathbf{y}^{\mathcal{G}}_{\text{RIP}} \approx \underset{\mathbf{y}\in\mathcal{T}_{\mathbf{Y}}}{\text{argmax}} \bigoplus_{\boldsymbol{\theta}\in\text{supp}\left(p(\boldsymbol{\theta}|\mathcal{D})\right)} \log p(\mathbf{y}|\mathcal{G}, \mathbf{x}; \boldsymbol{\theta}) \tag{A.1}$$

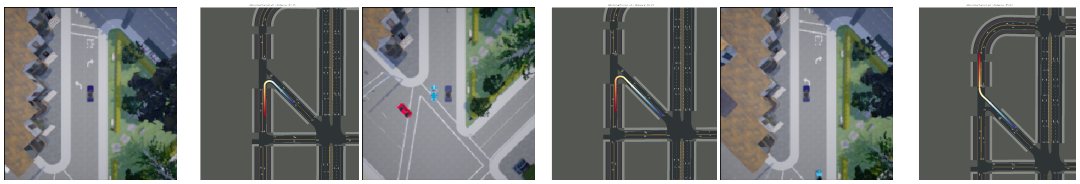
Solving for Eqn. (A.1) results in $\times 20$ improvement in runtime compared to the gradient descent alternative. Although in in-distribution scenes solving Eqn. (A.1) over Eqn. (3.4) does not deteriorate perfomance, in out-of-distribution scenes the trajectory library, $\mathcal{T}_{\mathbf{Y}}$, is not useful. Therefore in the experiments (c.f. Section 3) we used online gradient-descent. Future work lies in developing a hybrid optimization method that takes advantage of the speedup the trajectory library provides without a decrease in performance in out-of-distribution scenarios.



(a) Trajectories **(b)** $K = 64$ **(c)** $K = 128$ **(d)** $K = 1024$

Figure A.1: Our trajectory library from CARLA's autopilot demonstrations, 4 seconds.

CARNOVEL: Suite of Tasks Under Distribution Shift

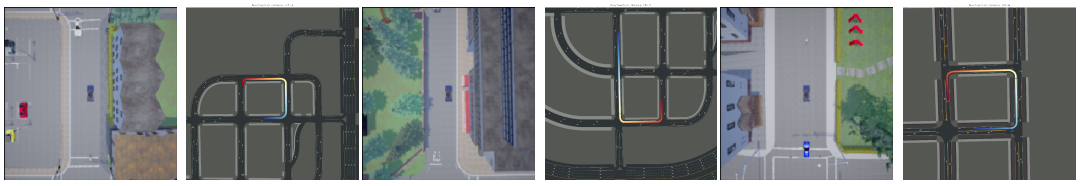


(a) `AbnormalTurns0-v0` **(b)** `AbnormalTurns1-v0` **(c)** `AbnormalTurns2-v0`

(d) `AbnormalTurns3-v0` **(e)** `AbnormalTurns4-v0` **(f)** `AbnormalTurns5-v0`

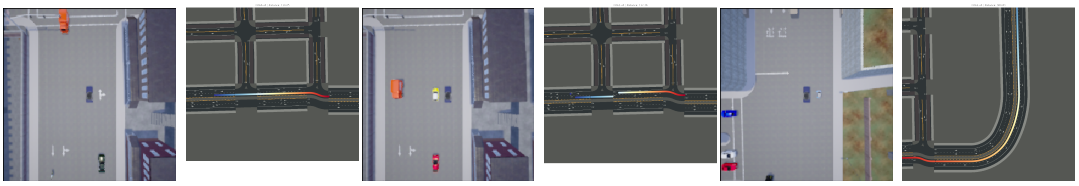
(g) `AbnormalTurns6-v0` **(h)** `BusyTown0-v0` **(i)** `BusyTown1-v0`

(j) `BusyTown2-v0` **(k)** `BusyTown3-v0` **(l)** `BusyTown4-v0`



(m) `BusyTown5-v0` **(n)** `BusyTown6-v0` **(o)** `BusyTown7-v0`

(p) BusyTown8-v0 (q) BusyTown9-v0 (r) BusyTown10-v0



(s) Hills0-v0 (t) Hills1-v0 (u) Hills2-v0

(v) Hills3-v0 (w) Roundabouts0-v0 (x) Roundabouts1-v0

(y) Roundabouts2-v0 (z) Roundabouts3-v0 (aa) Roundabouts4-v0

B

SHIFTs Vehicle Motion Predictions appendix

The current appendix contains a description of the composition, collection, preprocessing and partitioning of the Shifts Vehicle Motion Prediction dataset. Additionally, it contains a description of the metrics used for assessment and an expanded set of experimental results.

Dataset Description

Table B.1: A comparison of various motion prediction datasets. The Shifts Vehicle Motion Prediction dataset is the largest by number of scenes and total size in hours.

Dataset	Scene Length (s)	# Scenes			Total Size (h)	Avg. # Actors
		Train	Dev	Eval		
Argoverse	5	205,942	39,472	78,143	320	50
Lyft	25	134,000	11,000	16,000	1,118	79
Waymo	20	72,347	15,503	15,503	574	-
Shifts	10	500,000	50,000	50,000	1,667	29

Composition The dataset for the Vehicle Motion Prediction task was collected by the Yandex Self-Driving Group (SDG) fleet. This is the largest vehicle motion prediction dataset released to date, containing 600,000 scenes (see Table B.1 for a comparison to other public datasets). The dataset consists of scenes spanning

six locations, three seasons, three times of day, and four weather conditions (cf. Table B.2 and B.3). Each of these conditions is available in the form of tags associated with every scene. Each scene is 10 seconds long and is divided into 5 seconds of context features and 5 seconds of ground truth targets for prediction, separated by the time $T = 0$. The goal of the task is to predict the movement trajectory of vehicles at time $T \in (0, 5]$ based on the information available for time $T \in [-5, 0]$.

Table B.2: The number of scenes in the Vehicle Motion Prediction dataset by location and season.

Location	Train	Dev	Eval
Moscow	450,504	30,505	30,534
Skolkovo	6,283	2,218	2,956
Innopolis	15,086	5,164	5,016
Ann Arbor	19,349	8,290	6,617
Modiin	3,502	2,262	1,555
Tel Aviv	5,276	1,561	3,322

Season	Train	Dev	Eval
Summer	85,698	10,634	10,481
Autumn	126,845	15,290	15,840
Winter	287,457	24,076	23,679
Spring	0	0	0

Each scene includes information about the state of dynamic objects (i.e., vehicles, pedestrians) and an HD map. Each vehicle is described by its position, velocity, linear acceleration, and orientation (yaw, known up to $\pm\pi$). A pedestrian state consists of a position vector and a velocity vector. All state components are represented in a common coordinate frame and sampled at 5Hz frequency by the perception stack running on the Yandex SDG fleet. The HD map includes lane information (e.g., traffic direction, lane priority, speed limit, traffic light association), road boundaries, crosswalks, and traffic light states, which are also sampled at 5Hz. To facilitate easy use of this dataset, we provide utilities to render scene information as a feature map, which can be used as an input to a standard vision model (e.g., a ResNet He et al. ((2016))). Our utilities represent each scene as a birds-eye-view image with each channel corresponding to a particular feature (e.g., a vehicle occupancy map) at a particular timestep. We also provide pre-rendered feature maps for every prediction request (cf. Appendix B) in the dataset, which are used to train the baseline models. The maps are 128×128 pixels in size

with each pixel covering 1 square meter, have 17 channels describing both HD map information and dynamic object states at time $T = 0$, and are centered with respect to the agent for which a prediction is being made. Researchers working with the dataset are free to use these feature maps, use the provided utilities to render another set of feature maps at different (earlier) timesteps, or construct their own scene representations from the raw data.

The ground truth part of a scene contains future states of dynamic objects sampled at 5Hz for a total of 25 state samples. Some objects might not have all 25 states available due to occlusions or imperfections of the on-board perception system.

Table B.3: The number of scenes in the Vehicle Motion Prediction dataset by precipitation and time of day.

Precipitation Type	Train	Dev	Eval
No	432,598	44,799	44,274
Rain	15,618	1,857	1,751
Sleet	15,210	1,082	990
Snow	36,574	2,262	2,985

Sun Phase	Train	Dev	Eval
Astronomical Night	171,867	13,164	13,113
Daylight	299,065	33,879	33,979
Twilight	29,068	2,957	2,908

A number of vehicles in the scene are labeled as *prediction requests*. These are the vehicles that are visible at the most recent time $T = 0$ in the context features part of a scene, and therefore would call for a prediction in a deployed system. For such vehicles we provide not only their future trajectories, but also a number of non–mutually exclusive tags (detailed in Table B.4) describing the associated maneuver in more detail – whether the vehicle is turning, accelerating, slowing down, etc. – for a total of 10 maneuver types. Note that some prediction requests may not have all 25 state samples available. We call prediction requests with fully-observed state *valid* prediction requests and propose to evaluate predictions only on those.

In order to study the effects of distributional shift, as well as assess the robustness and uncertainty estimation of baseline models, we divide the Vehicle Motion Prediction dataset such that there are *in-domain* partitions which match the location and precipitation type of the training set, and *out-of-domain* or *shifted* partitions

Table B.4: Number of actor maneuvers of the respective type.

Maneuver Type	Train	Dev	Eval
Move Left	254,843	25,049	25,820
Move Right	322,231	30,074	30,633
Move Forward	5,032,724	395,467	413,920
Move Back	54,677	4,811	4,891
Acceleration	2,473,750	206,977	215,009
Deceleration	2,050,186	168,550	174,477
Uniform Movement	6,369,920	566,083	573,033
Stopping	441,619	38,411	39,336
Starting	739,143	64,986	65,759
Stationary	4,620,678	433,161	433,576

which do not match the training data along one or more of those axes. Furthermore, we provide a *development* set which acts as a validation set, and an *evaluation* set which acts as the test set. For standardized benchmarking we define a *canonical partitioning* of the full dataset (cf. Figure B.1, Table B.5) as the following. The training, in-domain development, and in-domain evaluation data are taken from Moscow. Distributionally shifted development data is taken from Skolkovo, Modiin, and Innopolis. Distributionally shifted evaluation data is taken from Tel Aviv and Ann Arbor. In addition, we remove all cases of precipitation from the in-domain training, development, and evaluation sets, while distributionally shifted datasets include precipitation. The canonical partitioning is fully described in Figure B.1. This partitioning is also the one used in the Shifts Challenge.

	Moscow	Skolkovo	Modiin	Innopolis	Ann-Arbor	Tel Aviv
No precipitation	train development in evaluation in	development out	development out	development out	evaluation out	evaluation out
Rain	UNUSED	development out	development out	development out	evaluation out	evaluation out
Sleet	UNUSED	UNUSED	UNUSED	UNUSED	evaluation out	evaluation out
Snow	UNUSED	development out	development out	development out	evaluation out	evaluation out

Figure B.1: The canonical partitioning of the Vehicle Motion Prediction dataset.

Table B.5: The number of scenes in the canonical dataset partitioning.

Dataset Partition	In-Distribution	Distributionally Shifted
Train	388,406	-
Development	27,036	9,569
Evaluation	26,865	9,939

Collection Process The Vehicle Motion Prediction data was collected by the perception system running onboard a number of self-driving vehicles equipped with LiDAR sensors, radars, and cameras. This perception system consists of a number of neural network–based detectors followed by an object tracker that fuses detections across sensor modalities and time. The provided HD map for each location has been constructed and validated by cartographers employed by Yandex SDG. The provided dataset was sampled from a much larger dataset collected over a course of 8 months. The sampling procedure was biased towards sampling scenes on which the motion prediction system currently used by the SDC fleet makes mistakes, as well as sampling more scenes from locations where the fleet drives less frequently.

Preprocessing and Cleaning The collected dataset has been cleaned from scenes in which:

- any kind of onboard system failure was detected, as the perception system output can potentially be unreliable in such scenes;
- the perception system has produced outputs that clearly violate physical constraints, such as actors having unrealistic acceleration or colliding with one other.

Format This dataset is provided in protobuf format.

License We release this dataset under the CC BY NC SA 4.0 license.

Task Setup

Vehicle Motion Prediction is a complex task and therefore must be described in detail. We provide a training dataset $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ of time-profiled ground truth trajectories (i.e., plans) $\boldsymbol{y}$ paired with high-dimensional observations (features) $\boldsymbol{x}$ of the corresponding scenes. Each $\boldsymbol{y} = (s_1, \ldots, s_T)$ corresponds to the trajectory of a given vehicle observed through the SDG perception stack. Each state s_t corresponds to the x- and y-displacement of the vehicle at timestep t, s.t. $\boldsymbol{y} \in \mathbb{R}^{T\times 2}$. We consider the performance of models on development and evaluation datasets $\mathcal{D}^j_{\text{dev}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M_j}$. and $\mathcal{D}^j_{\text{eval}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M_j}$. See Figure B.2 for a depiction of the task.

Prediction Requests. There are N (M_j) prediction requests in the training dataset (evaluation datasets), with many requests for each scene corresponding to the many different vehicle trajectories observed. For example, in the canonical partition of the data, there are 388,406 scenes in the training dataset (Moscow, no precipitation), and 5,649,675 valid prediction requests.

Models can be trained to make use of ground truth trajectories that contain occlusions (i.e., prediction requests that are not valid) during training, such as through linear interpolation of missing steps. However, for the baseline methods considered in this work, both training and evaluation are done using only the fully observed ground truth trajectories.

Next, we describe the two levels of uncertainty quantification that we consider for each prediction request in the proposed task: per-trajectory and per–prediction request uncertainty scores.

Per-Trajectory Confidence Scores. Like machine translation, motion prediction is an inherently multimodal task. A motion prediction model can produce a different number of sampled trajectories (plans) D_i for each input $\boldsymbol{x}_i$; in other words, for two inputs $\boldsymbol{x}_i, \boldsymbol{x}_j$ with $i \neq j$, D_i and D_j can differ. As a justification, consider that in a certain context, multiple trajectories may be desirable

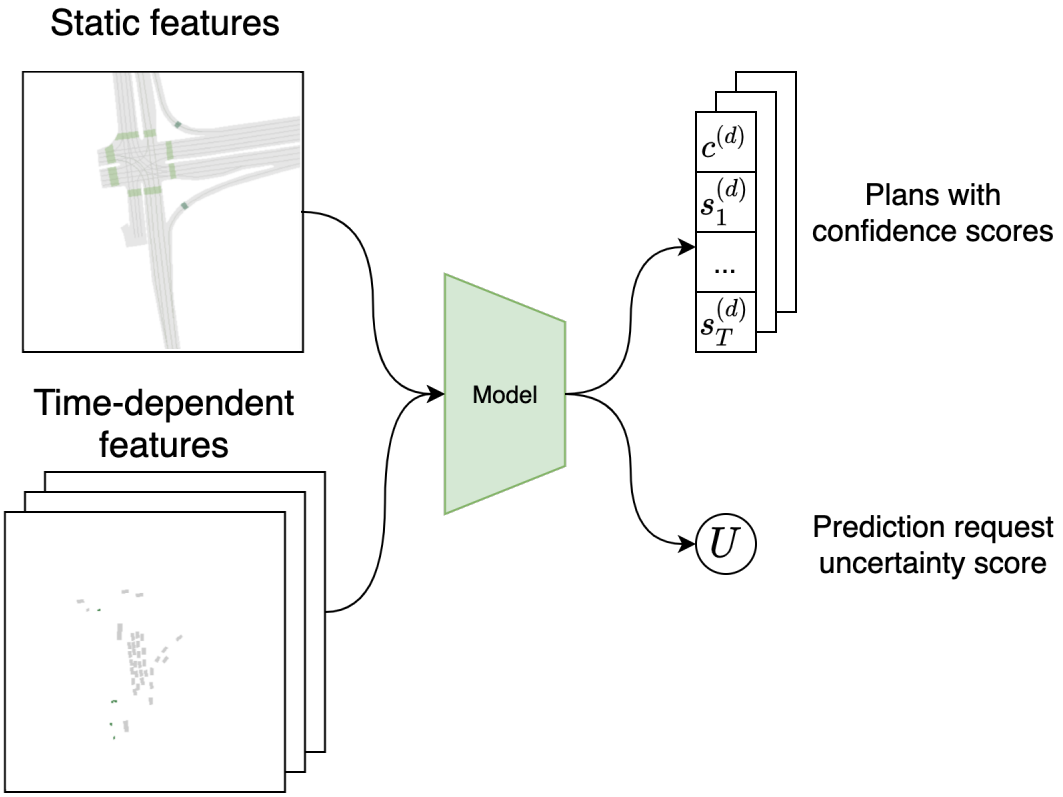


Figure B.2: Diagram of the Vehicle Motion Prediction task. Models take as input a single scene context $\boldsymbol{x}$ composed of static (HD map) and time-dependent input features, and predict trajectories $\{\boldsymbol{y}^{(d)} \mid d \in 1, \ldots, D\}$ with corresponding per-trajectory confidence scores $\{\boldsymbol{c}^{(d)} \mid d \in 1, \ldots, D\}$, as well as a single per–prediction request uncertainty score U.

to capture multimodality (e.g., the vehicle of interest is at a T-junction), and in others a single or fewer trajectories would be sufficient (e.g., the vehicle is clearly proceeding straight). In our task, we expect a stochastic model to accompany its D_i predicted trajectories on a given input $\boldsymbol{x}_i$ with scalar per-trajectory confidence scores $c_i^{(d)}, d \in \{1, \ldots D_i\}$. These provide an ordering of the plausibility of the various trajectories predicted for a given input. The scores must be non-negative and sum to 1 (i.e., form a valid probability distribution).

Per–Prediction Request Uncertainty Score. We also expect models to produce scalar uncertainty estimates corresponding to each prediction request input $\boldsymbol{x}_i$. For example, on evaluation dataset $\mathcal{D}_{\text{eval}}^j$, we have M_j per–prediction request uncertainty scores $\{U_i \mid i \in 1, \ldots, M_j\}$. These correspond to the model's uncer-

tainty in making any trajectory prediction for the agent of interest. In a real-world deployment setting, a self-driving vehicle would associate a high per–prediction request uncertainty score with a scene context that is particularly unfamiliar or high-risk.

Next, we will describe standard motion prediction performance metrics, followed by confidence-aware metrics which reward models with well-calibrated uncertainty.

Performance Metrics

Standard Performance Metrics. We assess the performance of a motion prediction system using several standard metrics.

The average displacement error (ADE) measures the quality of a predicted trajectory $\boldsymbol{y}$ with respect to the ground truth trajectory $\boldsymbol{y}^*$ as

$$\mathrm{ADE}(\boldsymbol{y}) \coloneqq \frac{1}{T}\sum_{t=1}^{T} \|s_t - s_t^*\|_2 \,, \tag{B.1}$$

where $\boldsymbol{y} = (s_1, \ldots, s_T)$. Analogously, the final displacement error

$$\mathrm{FDE}(\boldsymbol{y}) \coloneqq \|s_T - s_T^*\|_2 \,, \tag{B.2}$$

measures the quality at the last timestep.

Stochastic models define a predictive distribution $q(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta})$, and can therefore be evaluated over the D trajectories sampled for a given input $\boldsymbol{x}$. For example, we can measure an aggregated ADE over D samples with

$$\mathrm{aggADE}_D(q) \coloneqq \bigoplus_{\{\boldsymbol{y}\}_{d=1}^{D} \sim q(\boldsymbol{y}|\boldsymbol{x})} \mathrm{ADE}(\boldsymbol{y}^d), \tag{B.3}$$

where $\oplus$ is an aggregation operator, e.g., $\oplus = \min$ recovers the minimum ADE (minADE_D) commonly used in evaluation of stochastic motion prediction models Filos et al. ((2020)), Phan-Minh et al. ((2020)). We consider minimum and mean aggregation of the average displacement error (minADE, avgADE), as well as of the final displacement error (minFDE, avgFDE).

Per-Trajectory Confidence-Aware Metrics. A stochastic model used in practice for motion prediction must ultimately *decide* on a particular predicted

trajectory for a given prediction request. We may make this decision by selecting for evaluation the predicted trajectory with the highest per-trajectory confidence score. In other words, given per-trajectory confidence scores $\{c^{(d)} \mid d \in 1, \ldots, D\}$ we select the top trajectory $y^{(d^*)}, d^* = \arg\max_d c^{(d)}$, and measure the decision quality using *top1* ADE and FDE metrics, e.g.,

$$\text{top1ADE}_D(q) \coloneqq \text{ADE}(\boldsymbol{y}^{(d^*)}). \tag{B.4}$$

We may also wish to assess the quality of the relative weighting of the D trajectories with their corresponding per-trajectory confidence scores $c^{(d)}$. For this the following weighted metric can be considered:

$$\text{weightedADE}_D(q) \coloneqq \sum_{d \in D} c^{(d)} \cdot \text{ADE}(\boldsymbol{y}^{(d)}). \tag{B.5}$$

The top1FDE and weightedFDE metrics follow analogously to the above. Unfortunately, these metrics, while highly intuitive, have a conceptual limitation. Consider the following loss:

$$\mathcal{L}\left(\mathtt{p}(\boldsymbol{y}|\boldsymbol{x}), \{\hat{c}_i^{(1:D)}, \hat{\boldsymbol{y}}^{(1:D)}\}\right) = \mathbb{E}_{\mathtt{p}(\boldsymbol{y}|\boldsymbol{x})}\left[\sum_{d=1}^{D} c^d \text{ADE}(\hat{\boldsymbol{y}}^d, \boldsymbol{y})\right],\ \{\hat{c}_i^{(1:D)}, \hat{\boldsymbol{y}}^{(1:D)}\} = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta}) \tag{B.6}$$

which is the expected weightedADE given a set of trajectories and weights from a model. If we wish to minimize this loss with respect to the predicted trajectories and weights, then:

$$\begin{aligned}
\mathcal{L}_{\min} &= \min_{\{\hat{c}_i^{(1:D)}, \hat{\boldsymbol{y}}^{(1:D)}\}} \left\{ \mathbb{E}_{\mathtt{p}(\boldsymbol{y}|\boldsymbol{x})}\left[\sum_{d=1}^{D} c^d \text{ADE}(\hat{\boldsymbol{y}}^d, \boldsymbol{y})\right] \right\} \\
&= \min_{\{\hat{c}_i^{(1:D)}\}} \left\{ \sum_{d=1}^{D} \hat{c}^d \left(\min_{\{\hat{\boldsymbol{y}}^{(d)}\}} \left\{ \mathbb{E}_{\mathtt{p}(\boldsymbol{y}|\boldsymbol{x})}\left[\text{ADE}(\hat{\boldsymbol{y}}^d, \boldsymbol{y})\right] \right\} \right) \right\} \\
&= \min_{\{\hat{c}_i^{(1:D)}\}} \left\{ \mathbb{E}_{\mathtt{p}(\boldsymbol{y}|\boldsymbol{x})}\left[\text{ADE}(\hat{\boldsymbol{y}}^*, \boldsymbol{y})\right] \sum_{d=1}^{D} \hat{c}^d \right\} \\
&= \mathbb{E}_{\mathtt{p}(\boldsymbol{y}|\boldsymbol{x})}\left[\text{ADE}(\hat{\boldsymbol{y}}^*, \boldsymbol{y})\right]
\end{aligned} \tag{B.7}$$

where $\hat{\boldsymbol{y}}^*$ is the *weighted geometric median*

$$\hat{\boldsymbol{y}}^* = \arg_{\{\hat{\boldsymbol{y}}\}} \min \left\{ \mathbb{E}_{\mathtt{p}(\boldsymbol{y}|\boldsymbol{x})}\left[\text{ADE}(\hat{\boldsymbol{y}}, \boldsymbol{y})\right] \right\} \tag{B.8}$$

Thus, the optimal model would suffer from *mode-collapse* and always yields the weighted geometric median of the modes of the true distribution of trajectories. To put this concretely, at a T-junction, where trajectories can go either left or right, the optimal model will yield a trajectory going straight, which is clearly a fundamentally undesirable behaviour. Mathematically, the problem lies in the additive nature of the metric – each mode can be optimized independently of the others. This can be avoided by instead considering a likelihood based metric, such as the following one:

$$\text{cNLL}(\mathcal{D}) \coloneqq \frac{1}{N}\sum_{n=1}^{N}\left\{-\ln\left[\sum_{d=1}^{D} c^{(d)} \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{y}_{t,i}^{*};\ \boldsymbol{s}_t^{(d)}(\boldsymbol{x}_i;\boldsymbol{\theta}), \boldsymbol{\Sigma}=\mathbf{1})\right]\right\} - T\ln 2\pi \quad \text{(B.9)}$$

Under the following metric, which assumes that each mode is modelled using a Normal distribution of fixed variance, an optimal model would place a Normal over each mode and weight them appropriately. This can be clearly demonstrated using the following numerical example:

$$y \sim \mathtt{p}(y) = 0.5 \cdot \mathcal{N}(x, 10, 1) + 0.5 \cdot \mathcal{N}(x, -10, 1) \quad \text{(B.10)}$$

$$\mathbb{E}_{\mathtt{p}}(y)[\text{wADE}(y, \boldsymbol{s}^{(1:2)} = [10, -10], \boldsymbol{c} = [0.5, 0.5])] = 201.5 \quad \text{(B.11)}$$

$$\mathbb{E}_{\mathtt{p}}(y)[\text{wADE}(y, \boldsymbol{s}^{(1:2)} = [0, 0], \boldsymbol{c} = [0.5, 0.5])] = 101.50$$

$$\mathbb{E}_{\mathtt{p}}(y)[\text{cNLL}(y, \boldsymbol{s}^{(1:2)} = [10, -10], \boldsymbol{c} = [0.5, 0.5])] = 1.09 \quad \text{(B.12)}$$

$$\mathbb{E}_{\mathtt{p}}(y)[\text{cNLL}(y, \boldsymbol{s}^{(1:2)} = [0, 0], \boldsymbol{c} = [0.5, 0.5])] = 50.75 \quad \text{(B.13)}$$

Where we have a bimodal Gaussian mixture distribution with modes at -10, 10. We assume we have a model which predicts the means of two trajectories with equal weight. We have two situations: either the model yields two distinct modes at -10, 10 or a collapsed mode at 0 (the median). We can see that predicting the median will yield a lower weightedADE and correctly predicting two distinct modes will yield the lower cNLL. It is important to highlight that this argument holds *in expectation* and is relevant to situations which contain inherent ambiguity and multi-modality. Note that the offset $T\ln 2\pi$ is used to make assure that the minimal value of this metric is 0, so that it can be used for error-retention and F1-retention plots.

Experimental Setup

Robust Imitative Planning. In detail, we use the following approach for trajectory and confidence score generation.

1) **Trajectory Generation.** Given a scene input $\boldsymbol{x}$, K ensemble members generate G trajectories.[1]

2) **Trajectory Scoring.** We score each of the G trajectories by computing a log probability under each of the K trained likelihood models.

3) **Per-Trajectory Confidence Scores.** We aggregate the $G \cdot K$ resulting log probabilities to G scores using a per-trajectory aggregation operator $\oplus_{\text{trajectory}}$.[2] By aggregating over the log-likelihood estimates sampled from the model posterior (i.e., contributed by each ensemble member), we obtain a robust score for each of the G trajectories Filos et al. ((2020)).

4) **Trajectory Selection.** Among the G trajectories, the RIP ensemble produces the top D trajectories as determined by their corresponding G per-trajectory confidence scores, where D is a hyperparameter.

5) **Per–Prediction Request Uncertainty Score.** We aggregate the D top per-trajectory confidence scores to a single uncertainty score U using the aggregator $\oplus_{\text{pred-req}}$.[3] This value conveys the ensemble's estimated uncertainty for a given scene context and a particular prediction request.

6) **Confidence Reporting.** We obtain scores $c^{(d)}$ by applying a `softmax` to the D top per-trajectory confidence scores. We report these $c^{(d)}$ and U (computed in step **5**) as our final per-trajectory confidence scores and per–prediction request uncertainty score, respectively.

[1] In practice, each ensemble member generates the same number of trajectories Q, s.t. $G = K \cdot Q$.

[2] For example, applying a `min` aggregation is informed by robust control literature Wald ((1939)) in which we aim to optimize for the worst-case scenario, as measured by the log-likelihood of the "most pessimistic" model for a given trajectory.

[3] In practice, this is done by applying the aggregation (e.g., $\oplus_{\text{pred-req}} = \texttt{mean}$) to the confidences $c^{(d)}$, and then *negating* to obtain the uncertainty score U.

To summarize, our implementation of RIP for motion prediction produces D trajectories and corresponding normalized per-trajectory scores $\{c^{(d)} \mid d \in 1, \ldots, D\}$, as well as an aggregated uncertainty score U for the overall prediction request.

Backbone Likelihood Model. We consider two different model classes as ensemble members: a simple behavioral cloning agent with a Gated Recurrent Unit decoder (BC) Cho et al. ((2014)), Codevilla et al. ((2018)) and a Deep Imitative Model (DIM) Rhinehart et al. ((2020)) with an autoregressive flow decoder Rezende and Mohamed ((2015a)), following Filos et al. ((2020)). In both cases, we model the likelihood of a trajectory $\boldsymbol{y}$ in context $\boldsymbol{x}$ to come from an expert (i.e., from the distribution of ground truth trajectories), with learnable parameters $\boldsymbol{\theta}$, as

$$q(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) = \prod_{t=1}^{T} p(s_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}; \boldsymbol{\theta}) \ = \prod_{t=1}^{T} \mathcal{N}(s_t; \mu(\boldsymbol{y}_{<t}, \boldsymbol{x}; \boldsymbol{\theta}), \Sigma(\boldsymbol{y}_{<t}, \boldsymbol{x}; \boldsymbol{\theta})), \quad \text{(B.14)}$$

where $\mu(\cdot; \boldsymbol{\theta})$ and $\Sigma(\cdot; \boldsymbol{\theta})$ are two heads of a recurrent neural network with shared torso. Hence we assume that the conditional densities are normally distributed, and learn those parameters through maximum likelihood estimation. Notably, for the BC model, we found that conditioning on samples $\hat{\boldsymbol{y}}_{<t}$ instead of ground truth values $\boldsymbol{y}_{<t}$ (where usage of ground truth is often referred to as teacher forcing in RNN literature) significantly improved performance across all datasets and metrics.

Uncertainty Estimation Methods. The above ensembling is done using multiple stochastic models trained with different random seeds, as introduced in Deep Ensembles Lakshminarayanan et al. ((2017)). For each ensemble member, we generate Q trajectories. We can also use a Monte Carlo Dropout Gal and Ghahramani ((2016)) approach for each ensemble member, in which we sample new dropout masks *at test time* during each of the Q forward passes (and corresponding trajectory generations). Following Smith and Gal ((2018)) we refer to the combination of this uncertainty estimation method with ensembling as Dropout Ensembles. Previous work has investigated the benefits of Deep Ensembles from a loss landscape perspective Fort et al. ((2019)), and found that Deep Ensembles tend to explore diverse modes in function space, whereas approximate variational methods such as Monte Carlo Dropout explore around a particular mode. Dropout

Ensembles are hence motivated as ensembles of variational methods which aim to consider a diverse set of modes, with local exploration around each mode.

Setup. We report performance of RIP across the two backbone models – Behavioral Cloning (BC) Codevilla et al. ((2018)) and Deep Imitative Model (DIM) Rhinehart et al. ((2020)) – as well as the two uncertainty estimation methods – Deep Ensembles Lakshminarayanan et al. ((2017)) and Dropout Ensembles Gal and Ghahramani ((2016)), Smith and Gal ((2018)). We evaluate RIP on development (`dev`) and evaluation (`eval`) datasets in in-distribution (In), distributionally shifted (Shifted), and combined in-distribution and shifted (Full) settings. With both backbone model classes we vary the number of ensemble members $K \in \{1, 3, 5\}$, train with learning rate 1e-4, use a cosine annealing LR schedule with 1 epoch warmup, and use gradient clipping at 1. We sample $Q = 10$ trajectories from each of the ensemble members. We consider two types of aggregation: "Lower Quartile" in which we compute the mean minus the standard deviation $\mu - \sigma$ of the input scores, and "Model Averaging" (MA) in which we compute the mean μ of the input scores. LQ reflects the intuition to assign a high score to a trajectory when the ensemble members assign it a high score on average, and tend to be certain (have a low standard deviation) in their scoring; MA reflects only the prior intuition. This aggregation strategy (LQ or MA) is used as both the per-trajectory aggregation operator $\oplus_{\text{trajectory}}$ and the per–prediction request aggregation operator $\oplus_{\text{pred-req}}$ (where the latter is followed by negation to obtain an uncertainty, as opposed to a confidence). We fix the RIP ensemble at all K to produce the top $D = 5$ trajectories as ranked by their per-trajectory confidence score.

Additional Results

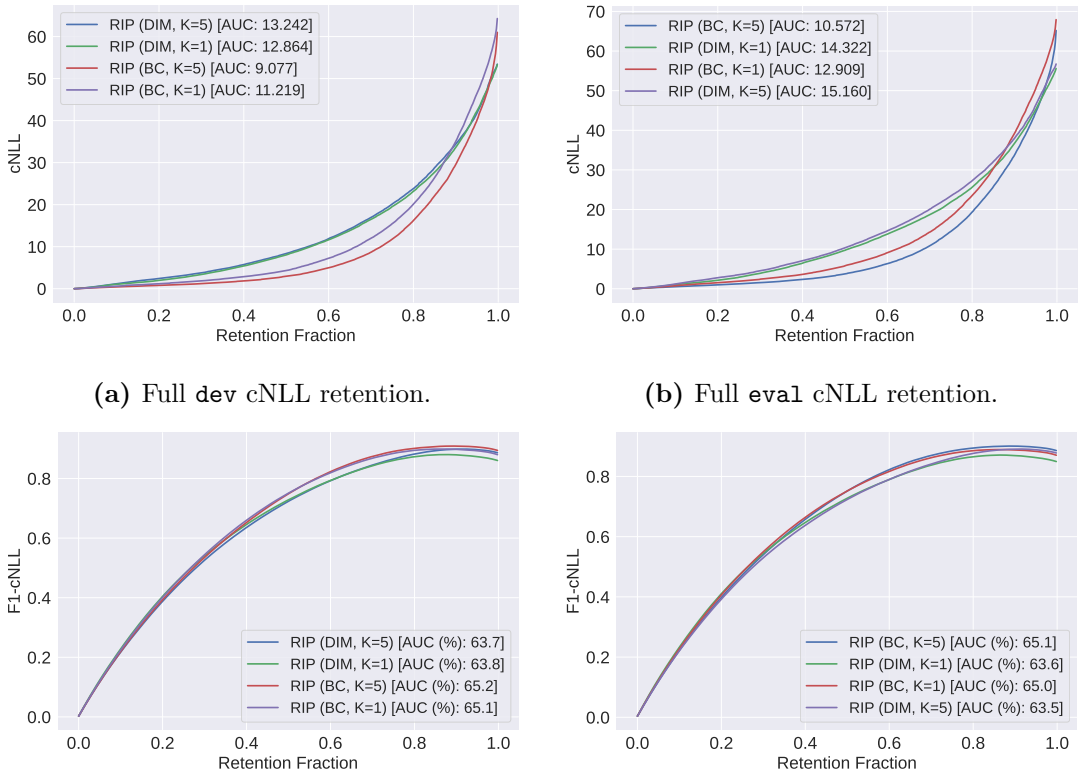
Below, we report predictive performance using standard-metrics, robustness and uncertainty quantification metrics, and retention plots across the RIP variants.

Table B.6: *Predictive performance* of RIP, across model backbones (behavioral cloning (BC) Codevilla et al. ((2018)) and Deep Imitative Model (DIM) Rhinehart et al. ((2020))) and uncertainty estimation methods (Deep Ensembles Lakshminarayanan et al. ((2017)) and Dropout Ensembles Smith and Gal ((2018))). Each section contains losses computed over the in-distribution (In), distributionally shifted (Shifted), and combined (Full) development and evaluation datasets. Altogether, we vary the backbone model, uncertainty estimation method, aggregation strategy (applied for both the per-trajectory aggregation operator $\oplus_{\text{trajectory}}$ and the per–prediction request aggregation operator $\oplus_{\text{pred-req}}$), and the number of ensemble members K. See Appendix B for setup details.

Dataset	Method	Model	minADE ↓			weightedADE ↓			minFDE ↓			weightedFDE ↓			cNLL ↓		
			In	Shifted	Full	In	Shifted	Full	In	Shifted	Full	In	Shifted	Full	In	Shifted	Full
Dev	Deep Ensemble	BC, LQ, K=1	0.818	0.960	0.835	1.088	1.245	1.107	1.718	2.113	1.765	2.368	2.777	2.417	59.64	98.54	64.29
		BC, LQ, K=3	0.780	0.909	0.795	1.040	1.170	1.056	1.638	2.018	1.683	2.254	2.609	2.297	54.78	87.81	58.73
		BC, LQ, K=5	0.766	0.888	0.780	1.017	1.138	1.031	1.618	1.980	1.661	2.214	2.552	2.254	56.45	90.25	60.49
		BC, MA, K=1	0.818	0.960	0.835	1.088	1.245	1.107	1.718	2.113	1.765	2.368	2.777	2.417	59.64	98.54	64.29
		BC, MA, K=3	0.780	0.908	0.795	1.034	1.166	1.050	1.641	2.018	1.686	2.249	2.611	2.292	55.00	88.45	59.00
		BC, MA, K=5	0.765	0.887	0.779	1.012	1.133	1.026	1.617	1.976	1.660	2.210	2.551	2.251	56.86	91.54	61.01
		DIM, LQ, K=1	0.750	0.818	0.758	1.523	1.583	1.530	1.497	1.720	1.524	3.472	3.639	3.492	50.66	73.00	53.34
		DIM, LQ, K=3	0.717	0.787	0.725	1.407	1.470	1.415	1.467	1.687	1.493	3.219	3.397	3.240	48.88	**70.93**	51.52
		DIM, LQ, K=5	0.720	0.787	0.728	1.399	1.470	1.407	1.487	1.704	1.513	3.202	3.397	3.225	51.12	72.87	53.72
		DIM, MA, K=1	0.750	0.818	0.758	1.523	1.583	1.530	1.497	1.720	1.524	3.472	3.639	3.492	50.66	73.00	53.34
		DIM, MA, K=3	0.717	**0.785**	0.725	1.410	1.475	1.418	1.466	**1.685**	1.492	3.226	3.409	3.248	**48.74**	71.30	**51.44**
		DIM, MA, K=5	0.719	0.786	0.727	1.399	1.469	1.408	1.482	1.698	1.508	3.202	3.393	3.225	50.85	72.45	53.43
	Dropout Ensemble	BC, LQ, K=1	0.803	0.908	0.815	1.116	1.236	1.130	1.649	1.952	1.685	2.409	2.718	2.446	55.98	82.49	59.15
		BC, LQ, K=3	0.741	0.853	0.754	1.013	1.132	1.028	1.542	1.873	1.581	2.209	2.545	2.249	53.01	83.93	56.71
		BC, LQ, K=5	0.759	0.878	0.773	**1.008**	1.127	**1.023**	1.605	1.960	1.648	**2.204**	**2.538**	**2.244**	55.58	88.78	59.55
		BC, MA, K=1	0.803	0.908	0.815	1.116	1.236	1.130	1.649	1.952	1.685	2.409	2.718	2.446	55.98	82.49	59.15
		BC, MA, K=3	0.739	0.850	0.752	1.020	1.135	1.033	1.534	1.864	1.574	2.223	2.553	2.263	53.09	83.81	56.76
		BC, MA, K=5	0.757	0.877	0.771	1.010	**1.126**	1.024	1.597	1.952	1.640	2.209	2.539	2.248	55.82	89.57	59.86
		DIM, LQ, K=1	0.750	0.831	0.759	1.498	1.587	1.509	1.510	1.757	1.539	3.432	3.662	3.459	52.57	76.54	55.44
		DIM, LQ, K=3	**0.716**	0.786	0.725	1.412	1.473	1.419	1.466	1.687	1.493	3.234	3.408	3.254	49.69	72.58	52.43
		DIM, LQ, K=5	0.723	0.793	0.731	1.409	1.475	1.417	1.494	1.717	1.521	3.224	3.408	3.246	51.25	73.47	53.91
		DIM, MA, K=1	0.750	0.831	0.759	1.498	1.587	1.509	1.510	1.757	1.539	3.432	3.662	3.459	52.57	76.54	55.44
		DIM, MA, K=3	**0.716**	0.786	**0.724**	1.414	1.479	1.422	**1.465**	**1.685**	**1.491**	3.238	3.420	3.260	49.38	71.86	52.07
		DIM, MA, K=5	0.721	0.793	0.729	1.409	1.474	1.417	1.489	1.717	1.516	3.224	3.405	3.246	50.99	73.64	53.70
Eval	Deep Ensemble	BC, LQ, K=1	0.829	1.084	0.880	1.104	1.407	1.164	1.733	2.420	1.870	2.394	3.197	2.555	60.20	98.82	67.93
		BC, LQ, K=3	0.792	1.026	0.839	1.056	1.326	1.110	1.658	2.297	1.786	2.284	3.005	2.429	55.97	90.54	62.89
		BC, LQ, K=5	0.777	1.015	0.825	1.032	1.303	1.086	1.636	2.283	1.765	2.242	2.964	2.386	57.26	93.92	64.60
		BC, MA, K=1	0.829	1.084	0.880	1.104	1.407	1.164	1.733	2.420	1.870	2.394	3.197	2.555	60.20	98.82	67.93
		BC, MA, K=3	0.792	1.025	0.838	1.050	1.319	1.104	1.661	2.294	1.788	2.278	2.997	2.422	55.94	90.53	62.87
		BC, MA, K=5	0.777	1.014	0.824	1.028	1.299	1.082	1.636	2.278	1.765	2.238	2.957	2.382	57.75	95.00	65.20
		DIM, LQ, K=1	0.759	0.942	0.796	1.551	1.883	1.618	1.511	1.983	1.605	3.536	4.376	3.704	50.50	76.00	55.60
		DIM, LQ, K=3	0.726	0.914	0.764	1.433	1.756	1.498	1.481	1.972	1.579	3.277	4.094	3.440	49.45	76.66	54.89
		DIM, LQ, K=5	0.729	0.921	0.768	1.422	1.757	1.489	1.498	2.007	1.600	3.253	4.098	3.422	51.61	79.71	57.24
		DIM, MA, K=1	0.759	0.942	0.796	1.551	1.883	1.618	1.511	1.983	1.605	3.536	4.376	3.704	50.50	76.00	55.60
		DIM, MA, K=3	0.726	0.912	0.763	1.437	1.759	1.502	1.478	1.967	1.576	3.286	4.101	3.449	49.09	76.07	54.49
		DIM, MA, K=5	0.728	0.918	0.766	1.424	1.754	1.490	1.493	2.000	1.595	3.256	4.093	3.424	51.19	78.85	56.73
	Dropout Ensemble	BC, LQ, K=1	0.812	1.038	0.857	1.128	1.410	1.184	1.664	2.267	1.784	2.430	3.170	2.578	56.57	86.28	62.52
		BC, LQ, K=3	0.751	0.972	0.795	1.029	**1.297**	1.082	1.558	2.154	1.677	2.238	**2.948**	2.380	53.94	86.68	60.49
		BC, LQ, K=5	0.770	1.008	0.817	**1.024**	**1.297**	**1.079**	1.623	2.268	1.752	**2.233**	2.957	**2.378**	56.49	92.77	63.75
		BC, MA, K=1	0.812	1.038	0.857	1.128	1.410	1.184	1.664	2.267	1.784	2.430	3.170	2.578	56.57	86.28	62.52
		BC, MA, K=3	0.749	0.970	0.794	1.036	1.305	1.090	1.551	2.147	1.670	2.253	2.963	2.395	54.07	86.94	60.65
		BC, MA, K=5	0.768	1.004	0.815	1.027	1.299	1.081	1.615	2.253	1.743	2.239	2.958	2.383	56.90	93.27	64.18
		DIM, LQ, K=1	0.739	0.924	0.776	1.478	1.815	1.546	1.474	**1.949**	1.569	3.380	4.239	3.552	49.90	75.31	54.98
		DIM, LQ, K=3	0.722	0.910	0.760	1.431	1.763	1.497	1.470	1.967	1.569	3.266	4.112	3.435	49.30	75.24	54.49
		DIM, LQ, K=5	0.729	0.929	0.769	1.430	1.769	1.497	1.497	2.027	1.603	3.268	4.126	3.440	50.77	80.02	56.63
		DIM, MA, K=1	0.739	0.924	0.776	1.478	1.815	1.546	1.474	**1.949**	1.569	3.380	4.239	3.552	49.90	75.31	54.98
		DIM, MA, K=3	**0.720**	**0.907**	**0.758**	1.432	1.760	1.497	**1.465**	1.960	**1.564**	3.267	4.107	3.435	**48.74**	**74.70**	**53.93**
		DIM, MA, K=5	0.728	0.925	0.767	1.431	1.766	1.498	1.494	2.017	1.599	3.269	4.120	3.439	50.51	79.30	56.28

Table B.7: *Uncertainty and robustness performance* of RIP across the two backbone models (BC and DIM) and uncertainty estimation methods (Deep Ensemble and Dropout Ensemble). The error metric for computing the area under the rejection curve (R-AUC) and area under the F1 curve (F1-AUC) is **cNLL**. We use a threshold of 25 for the F1 metrics, which approximately corresponds to a 1 meter deviation on all trajectories. See Appendix B for setup details.

Dataset	Method	Model	R-AUC ↓			F1-AUC (%) ↑			F1@95% ↑			ROC-AUC (%) ↑
			In	Shifted	Full	In	Shifted	Full	In	Shifted	Full	
Dev	Deep Ensemble	BC, LQ, K=1	11.06	13.91	11.22	64.9	66.7	65.1	89.1	90.2	89.3	51.0
		BC, LQ, K=3	11.26	11.69	11.18	63.4	66.0	63.8	88.5	90.3	88.8	46.7
		BC, LQ, K=5	9.68	10.38	9.62	64.3	66.4	64.6	89.7	91.0	90.0	47.3
		BC, MA, K=1	11.06	13.91	11.22	64.9	66.7	65.1	89.1	90.2	89.3	51.0
		BC, MA, K=3	9.31	10.73	9.31	64.8	66.5	65.0	90.3	91.3	90.6	48.6
		BC, MA, K=5	9.07	10.47	9.08	64.9	66.5	65.2	90.4	91.3	90.6	49.2
		DIM, LQ, K=1	12.54	15.28	12.86	63.6	64.8	63.8	87.2	88.8	87.4	**51.8**
		DIM, LQ, K=3	12.30	14.51	12.57	63.7	64.9	63.8	89.3	89.9	89.3	51.4
		DIM, LQ, K=5	12.87	15.01	13.14	63.5	64.8	63.7	89.7	90.2	89.7	51.4
		DIM, MA, K=1	12.57	15.10	12.86	63.7	64.9	63.8	87.2	88.8	87.4	**51.8**
		DIM, MA, K=3	12.38	14.46	12.64	63.7	64.9	63.8	89.2	89.9	89.3	51.4
		DIM, MA, K=5	12.97	15.10	13.24	63.5	64.8	63.7	89.6	90.2	89.7	51.4
	Dropout Ensemble	BC, LQ, K=1	8.87	10.00	8.87	**65.3**	**67.1**	**65.6**	89.7	90.4	89.9	51.2
		BC, LQ, K=3	**8.11**	**9.53**	**8.14**	64.9	66.5	65.1	90.6	91.3	**90.8**	50.9
		BC, LQ, K=5	8.28	9.60	8.28	65.0	66.6	65.2	90.5	91.3	90.7	50.7
		BC, MA, K=1	8.87	9.99	8.87	**65.3**	**67.1**	**65.6**	89.7	90.4	89.9	51.2
		BC, MA, K=3	8.53	9.79	8.54	64.9	66.5	65.1	**90.7**	**91.4**	**90.8**	50.3
		BC, MA, K=5	8.89	10.23	8.90	64.9	66.5	65.2	90.5	**91.4**	90.7	50.2
		DIM, LQ, K=1	12.57	16.41	13.03	63.8	64.7	63.9	87.6	89.1	87.8	51.5
		DIM, LQ, K=3	12.37	14.91	12.69	63.7	64.8	63.8	89.2	90.0	89.3	51.3
		DIM, LQ, K=5	12.94	15.18	13.22	63.6	64.8	63.7	89.6	90.2	89.7	51.4
		DIM, MA, K=1	12.61	16.30	13.06	63.8	64.8	63.9	87.6	89.1	87.7	51.6
		DIM, MA, K=3	12.49	14.80	12.79	63.6	64.8	63.8	89.2	90.0	89.3	51.4
		DIM, MA, K=5	13.05	15.20	13.33	63.5	64.8	63.7	89.5	90.2	89.6	51.4
Eval	Deep Ensemble	BC, LQ, K=1	11.16	20.84	12.91	64.9	65.5	65.0	88.9	85.6	88.4	52.8
		BC, LQ, K=3	11.31	17.09	12.38	63.4	64.8	63.7	88.4	86.4	88.0	50.9
		BC, LQ, K=5	9.77	15.95	10.88	64.3	65.4	64.5	89.5	87.1	89.1	51.4
		BC, MA, K=1	11.17	20.84	12.91	64.9	65.5	65.0	88.9	85.6	88.4	52.8
		BC, MA, K=3	9.40	16.76	10.73	64.8	65.6	65.0	90.2	87.5	89.7	51.3
		BC, MA, K=5	9.20	16.85	10.57	65.0	65.6	65.1	90.2	87.5	89.7	52.1
		DIM, LQ, K=1	12.78	20.78	14.28	63.5	63.7	63.6	86.9	83.9	86.3	52.0
		DIM, LQ, K=3	12.66	21.40	14.32	63.6	63.9	63.7	89.1	86.0	88.5	51.4
		DIM, LQ, K=5	13.26	22.59	15.05	63.5	63.8	63.6	89.5	86.5	88.9	51.2
		DIM, MA, K=1	12.81	20.83	14.32	63.6	63.8	63.6	86.9	83.9	86.3	51.8
		DIM, MA, K=3	12.74	21.51	14.42	63.6	63.9	63.7	89.1	86.0	88.5	51.1
		DIM, MA, K=5	13.37	22.68	15.16	63.5	63.7	63.5	89.5	86.5	88.9	50.9
	Dropout Ensemble	BC, LQ, K=1	9.06	15.49	10.22	**65.3**	**66.1**	**65.5**	89.5	86.4	89.0	53.7
		BC, LQ, K=3	**8.22**	**14.83**	**9.39**	64.9	65.6	65.1	**90.5**	87.5	**90.0**	53.9
		BC, LQ, K=5	8.39	15.16	9.57	65.0	65.7	65.2	90.4	87.6	89.9	**54.5**
		BC, MA, K=1	9.07	15.50	10.22	**65.3**	**66.1**	**65.5**	89.5	86.4	89.0	53.7
		BC, MA, K=3	8.69	15.90	9.99	64.9	65.6	65.1	**90.5**	**87.7**	**90.0**	53.0
		BC, MA, K=5	9.05	16.69	10.41	65.0	65.6	65.1	90.4	87.6	89.9	53.2
		DIM, LQ, K=1	12.45	20.27	13.92	63.6	63.7	63.6	87.7	84.7	87.1	51.8
		DIM, LQ, K=3	12.63	21.32	14.29	63.7	63.9	63.7	89.1	86.1	88.6	51.3
		DIM, LQ, K=5	13.22	22.78	15.04	63.5	63.8	63.6	89.4	86.3	88.8	51.2
		DIM, MA, K=1	12.51	20.33	14.00	63.6	63.8	63.7	87.7	84.7	87.1	51.5
		DIM, MA, K=3	12.73	21.43	14.40	63.6	63.9	63.7	89.1	86.0	88.5	51.1
		DIM, MA, K=5	13.36	22.85	15.19	63.5	63.8	63.6	89.4	86.3	88.8	50.9



(a) Full `dev` cNLL retention.

(b) Full `eval` cNLL retention.

(c) Full `dev` F1-cNLL retention.

(d) Full `eval` F1-cNLL retention.

Figure B.3: cNLL and F1-cNLL retention curves on the Full (i.e., containing both the in-distribution and distributionally shifted datapoints) `dev` (left column) and `eval` (right column) partitions of the Vehicle Motion Prediction dataset. Top row: retention on cNLL (lower ↓ AUC is better). Bottom row: retention on F1-cNLL (higher ↑ AUC is better). We vary the backbone model and number of ensemble members, fix the Model Averaging (MA) aggregation strategy for the per-trajectory aggregation operator $\oplus_{\text{trajectory}}$ and the per–prediction request aggregation operator $\oplus_{\text{pred-req}}$ (based on results from Table 3.5), and otherwise use the standard RIP settings enumerated in Appendix B.

C

CausalBALD appendix

Theoretical Results

τ-BALD

Theorem 1. *Under the following assumptions:*

1. *Unconfoundedness* $(\mathrm{Y}^0, \mathrm{Y}^1) \perp\!\!\!\perp \mathrm{T} \mid \mathbf{X}$*;*
2. *Consistency* $\mathrm{Y} \mid \mathrm{T} = \mathrm{Y}^{\mathrm{t}}$*;*
3. Y^1 *and* Y^0*, when conditioned on realizations* $\mathbf{x}$ *of the r.v.* $\mathbf{X}$ *and* t *of the r.v.* T*, are independent-normally distributed or joint-normally distributed r.v.s.*
4. $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})$ *is a consistent estimator of* $\mathbb{E}[\mathrm{Y} \mid \mathrm{T} = \mathrm{t}, \mathbf{X} = \mathbf{x}]$

the information gain for $\boldsymbol{\Omega}$ *if we could observe a label for the difference in potential outcomes* $\mathrm{Y}^1 - \mathrm{Y}^0$ *given measured covariates* $\mathbf{x}$*, treatment* t *and a dataset of observations* $\mathcal{D}_{\text{train}} = \{\mathbf{x}_i, \mathrm{t}_i, \mathrm{y}_i\}_{i=1}^n$ *is approximated as*

$$\mathrm{I}(\mathrm{Y}^1 - \mathrm{Y}^0; \boldsymbol{\Omega} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) \approx \underset{\boldsymbol{\omega} \sim p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})}{\mathrm{Var}} (\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) - \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0)) \tag{C.1}$$

Proof.

$$\begin{aligned}
\mathrm{I}(\mathrm{Y}^1 - \mathrm{Y}^0; \boldsymbol{\Omega} \mid \mathbf{x}, \mathcal{D}_{\text{train}}) &= \mathrm{H}(\mathrm{Y}^1 - \mathrm{Y}^0 \mid \mathbf{x}, \mathcal{D}_{\text{train}}) \\
&\quad - \mathop{\mathbb{E}}_{p(\boldsymbol{\Omega}|\mathcal{D}_{\text{train}})} \Big[\mathrm{H}(\mathrm{Y}^1 - \mathrm{Y}^0 \mid \mathbf{x}, \boldsymbol{\omega})\Big] && \text{(C.2a)} \\
&\approx \mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0 \mid \mathbf{x}, \mathcal{D}_{\text{train}}) \\
&\quad - \mathop{\mathbb{E}}_{p(\boldsymbol{\Omega}|\mathcal{D}_{\text{train}})} \Big[\mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0 \mid \mathbf{x}, \boldsymbol{\omega})\Big] && \text{(C.2b)} \\
&= \mathop{\mathrm{Var}}_{p(\boldsymbol{\Omega}|\mathcal{D}_{\text{train}})} \Big(\mathbb{E}[\mathrm{Y}^1 - \mathrm{Y}^0 \mid \mathbf{x}, \boldsymbol{\omega}]\Big) && \text{(C.2c)} \\
&= \mathop{\mathrm{Var}}_{p(\boldsymbol{\Omega}|\mathcal{D}_{\text{train}})} (\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) - \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0)) && \text{(C.2d)}
\end{aligned}$$

In (C.2a) we adapt the result of Houlsby et al. ((2011)) and express the information gain as the mutual information between the observable difference in potential outcomes $\mathrm{Y}^1 - \mathrm{Y}^0$ and the parameters $\boldsymbol{\Omega}$; given observed covariates $\mathbf{x}$, treatment t, and training data $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \mathrm{t}_i, \mathrm{y}_i)\}_{i=1}^{n_{\text{train}}}$. In (C.2b) we apply lemma 1.1 to the r.h.s terms of (C.2a). We then use the result in Jesson et al. ((2020)) and move from (C.2b) to (C.2c) by application of the law of total variance. Finally, under the consistency and unconfoundedness assumptions we express the information gain in terms of the identifiable difference in expected outcomes $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) - \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0)$. □

Lemma 1.1. *Under the following assumptions:*

1. $\mathrm{Y}^1, \mathrm{Y}^0$ *are independent-normally distributed or joint-normally distributed r.v.s;*
2. *With* $A = \mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0)$*: let* $|A - 1| \leq 1$ *and* $A \neq 0$*. That is to say, the predictive variance must be greater than 0 and less than or equal to 2;*

$$\mathrm{H}(\mathrm{Y}^1 - \mathrm{Y}^0) \approx \mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0) \tag{C.3}$$

Proof. By assumption 1, $\mathrm{Y}^1 - \mathrm{Y}^0$ is also a normally distributed random variable. By corollary 1.1,

$$\mathrm{H}(\mathrm{Y}^1 - \mathrm{Y}^0) = \frac{1}{2} + \frac{1}{2}\log(2\pi\,\mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0)) \tag{C.4}$$

So given assumption 2, the first order Taylor polynomial of $\mathrm{H}(\mathrm{Y}^1 - \mathrm{Y}^0)$ is

$$\begin{aligned}\frac{1}{2} + \frac{1}{2}\log(2\pi\,\mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0)) &\approx \frac{1}{2} + \frac{1}{2}(2\pi\,\mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0) - 1) \\ &= \frac{1}{2} + \pi\,\mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0) - \frac{1}{2} \\ &= \pi\,\mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0) \\ &\propto \mathrm{Var}(\mathrm{Y}^1 - \mathrm{Y}^0)\end{aligned} \tag{C.5}$$

□

Corollary 1.1. *The entropy of a normally distributed random variable with variance* σ^2 *is* $\frac{1}{2} + \frac{1}{2}\log(2\pi\sigma^2)$

μ-BALD

Theorem 2. *Under the following assumptions:*

1. *Unconfoundedness* $(\mathrm{Y}^0, \mathrm{Y}^1) \perp\!\!\!\perp \mathrm{T} \mid \mathbf{X}$,
2. *Consistency* $\mathrm{Y} \mid \mathrm{T} = \mathrm{Y}^{\mathrm{t}}$,
3. Y *conditioned on* $\mathbf{x}$ *and* t *is a normally distributed random variable,*
4. $\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})$ *is a consistent estimator of* $\mathbb{E}[\mathrm{Y} \mid \mathrm{T} = \mathrm{t}, \mathbf{X} = \mathbf{x}]$,

the information gain for $\boldsymbol{\Omega}$ *when we observe a label for the potential outcome* Y^{t} *given measured covariates* $\mathbf{x}$*, treatment* t *and a dataset of observations* $\mathcal{D}_{\text{train}} = \{\mathbf{x}_i, \mathrm{t}_i, \mathrm{y}_i\}_{i=1}^n$ *can be approximated as is*

$$\mathrm{I}(\mathrm{Y}^{\mathrm{t}}; \boldsymbol{\Omega} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) \approx \frac{1}{2}\log\left(\frac{\mathrm{Var}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}})}{\mathbb{E}_{\boldsymbol{\omega}}[\mathrm{Var}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \boldsymbol{\omega})]}\right), \tag{C.6}$$

or

$$\mathrm{I}(\mathrm{Y}^{\mathrm{t}}; \boldsymbol{\Omega} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) \approx \underset{\boldsymbol{\omega} \sim p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})}{\mathrm{Var}} \left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})\right). \tag{C.7}$$

Equation (C.6) expresses the information gain as the logarithm of a ratio between predictive and aleatoric uncertainty in the outcome. Whereas, equation (C.6) expresses the information gain as a direct estimate of the epistemic uncertainty.

Proof.

$$\begin{aligned}
\mathrm{I}(\mathrm{Y}^{\mathrm{t}}; \boldsymbol{\Omega} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) &= \mathrm{H}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) - \mathop{\mathbb{E}}_{p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})} [\mathrm{H}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \boldsymbol{\omega})] && \text{(C.8a)} \\
&= \frac{1}{2} \log \left(2\pi \operatorname{Var}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}})\right) \\
&\quad - \mathop{\mathbb{E}}_{p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})} \frac{1}{2} \log \left(2\pi \operatorname{Var}(\mathrm{Y} \mid \boldsymbol{\omega}, \mathbf{x}, \mathrm{t})\right) && \text{(C.8b)} \\
&\geq \frac{1}{2} \log \left(2\pi \operatorname{Var}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}})\right) \\
&\quad - \frac{1}{2} \log \left(2\pi \mathop{\mathbb{E}}_{p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})} \operatorname{Var}(\mathrm{Y} \mid \boldsymbol{\omega}, \mathbf{x}, \mathrm{t})\right) && \text{(C.8c)} \\
&= \frac{1}{2} \log \left(\frac{\operatorname{Var}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}})}{\mathbb{E}_{\boldsymbol{\omega}}[\operatorname{Var}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \boldsymbol{\omega})]}\right) && \text{(C.8d)}
\end{aligned}$$

$$\begin{aligned}
\mathrm{I}(\mathrm{Y}^{\mathrm{t}}; \boldsymbol{\Omega} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) &= \mathrm{H}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) - \mathop{\mathbb{E}}_{p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})} [\mathrm{H}(\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \boldsymbol{\omega})] && \text{(C.9a)} \\
&\approx \operatorname{Var}[\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})} [\operatorname{Var}[\mathrm{Y} \mid \mathbf{x}, \mathrm{t}, \boldsymbol{\omega}]] && \text{(C.9b)} \\
&= \mathop{\operatorname{Var}}_{\boldsymbol{\omega} \sim p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})} (\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})) && \text{(C.9c)}
\end{aligned}$$

In (C.9a) we express the information gain as the mutual information between the observed potential outcome Y^{t} and the parameters $\boldsymbol{\Omega}$; given observed covariates $\mathbf{x}$, treatment t, and training data $\mathcal{D}_{\text{train}}$. By consistency, we can drop the superscript on the potential outcome. In (C.9b) we approximate the r.h.s terms of (C.9a) by application of Lemma 1.1. Finally, we can move from (C.9b) to (C.9c) by application of the law of total variance. □

Note that for discrete or categorical Y*, it is straightforward to evaluate Equation* (C.9a) *directly.*

ρ-BALD

Theorem 3. *Under the following assumptions*

1. $\{\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}) : \mathrm{t} \in \{0, 1\}\}$ *are instances of the independent-normally distributed or joint-normally distributed random variables* $\{\widehat{\mu}^{\mathrm{t}}_{\boldsymbol{\Omega}} = \mathbb{E}[\mathrm{Y} \mid \boldsymbol{\Omega}, \mathrm{T} = \mathrm{t}, \mathbf{x}] : \mathrm{t} \in \{0, 1\}\}$,
2. $\operatorname{Var}_{\boldsymbol{\omega} \sim p(\boldsymbol{\Omega} \mid \mathcal{D}_{\text{train}})}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}')) > 0$.

Let $\widehat{\tau}_{\boldsymbol{\omega}}(\mathbf{x})$ be a realization of the random variable $\widehat{\tau}_{\boldsymbol{\Omega}} = \widehat{\mu}^1_{\boldsymbol{\Omega}} - \widehat{\mu}^0_{\boldsymbol{\Omega}}$. The information gain for $\widehat{\tau}_{\boldsymbol{\Omega}}$ if we observe the label for the potential outcome Y^{t} given measured covariates $\mathbf{x}$, treatment t and a dataset of observations $\mathcal{D}_{\text{train}} = \{\mathbf{x}_i, \mathrm{t}_i, \mathrm{y}_i\}_{i=1}^n$ is approximately

$$\begin{aligned} \mathrm{I}(\mathrm{Y}^{\mathrm{t}}; \widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}_{\text{train}}) &\approx \frac{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\tau}_{\boldsymbol{\omega}}(\mathbf{x}))}{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))}, \\ &= \frac{\mathrm{Var}_{\boldsymbol{\omega}}\left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})\right) - 2\mathrm{Cov}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}), \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))}{\mathrm{Var}_{\boldsymbol{\omega}}\left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}')\right)} + 1, \end{aligned} \tag{C.10}$$

where for binary $\mathrm{T} = t$, $\mathrm{t}' = (1 - \mathrm{t})$.

Proof.

$$\mathrm{I}(\mathrm{Y}^{\mathrm{t}}; \widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}) = \mathrm{H}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}) - \mathrm{H}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathrm{Y}^{\mathrm{t}}, \mathbf{x}, \mathrm{t}, \mathcal{D}) \tag{C.11a}$$

$$= \mathrm{H}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D}) - \mathop{\mathbb{E}}_{\mathrm{y}^{\mathrm{t}} \sim p(\mathrm{Y}^{\mathrm{t}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D})} \mathrm{H}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathrm{y}^{\mathrm{t}}, \mathbf{x}, \mathrm{t}) \tag{C.11b}$$

$$= \frac{1}{2}\log(2\pi\,\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}})) - \mathop{\mathbb{E}}_{\mathrm{y}^{\mathrm{t}} \sim p(\mathrm{Y}^{\mathrm{t}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D})} \left[\frac{1}{2}\log(2\pi\,\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathrm{y}^{\mathrm{t}}))\right] \tag{C.11c}$$

$$\geq \frac{1}{2}\log(2\pi\,\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}})) - \frac{1}{2}\log(2\pi\,\mathbb{E}\left[\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathrm{y}^{\mathrm{t}})\right]) \tag{C.11d}$$

$$= \frac{1}{2}\log\left(\frac{\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}})}{\mathbb{E}\left[\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathrm{y}^{\mathrm{t}})\right]}\right), \tag{C.11e}$$

and we can further expand the fraction to

$$\frac{\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D})}{\mathbb{E}\left[\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathrm{y}^{\mathrm{t}})\right]} = \frac{\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathbf{x}, \mathrm{t}, \mathcal{D})}{\mathrm{Var}_{\boldsymbol{\omega} \sim p(\boldsymbol{\Omega} \mid \mathcal{D})}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))} \tag{C.11f}$$

$$= \frac{\mathrm{Var}_{\boldsymbol{\omega} \sim p(\boldsymbol{\Omega} \mid \mathcal{D})}(\widehat{\tau}_{\boldsymbol{\omega}}(\mathbf{x}) \mid \mathrm{t})}{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))} \tag{C.11g}$$

$$= \frac{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 1) - \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, 0) \mid \mathrm{t})}{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))} \tag{C.11h}$$

$$= \frac{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}) - \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))}{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))} \tag{C.11i}$$

$$= \frac{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})) + \mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}')) - 2\,\mathrm{Cov}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}), \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))}{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))} \tag{C.11j}$$

$$= \frac{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t})) - 2\,\mathrm{Cov}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}), \widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))}{\mathrm{Var}_{\boldsymbol{\omega}}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x}, \mathrm{t}'))} + 1, \tag{C.11k}$$

where (C.11a) by definition of mutual information; (C.11a)-(C.11b) from the result of Houlsby et al. ((2011)); (C.11b)-(C.11c) by Assumption 1. and Corollary 1.1; (C.11c)-(C.11d) by Jensen's inequality; (C.11d)-(C.11e) by the logarithmic quotient identity; (C.11f) by Lemma 3.1; (C.11f)-(C.11g) by definition of the variance. (C.11g)-(C.11h) by definition of $\widehat{\tau}_{\boldsymbol{\omega}}$; (C.11h)-(C.11i) by symmetry of the variance of the difference of two random variables; (C.11i)-(C.11j) by the definition of the variance of the difference of two random variables; and (C.11j)-(C.11k) by cancelling terms. □

Lemma 3.1. *Under the following assumptions*

1. *Consistency* $\mathrm{Y} \mid \mathrm{T} = \mathrm{Y}^{\mathrm{t}}$;

2. *Unconfoundedness* $(\mathrm{Y}^0, \mathrm{Y}^1) \perp\!\!\!\perp \mathrm{T} \mid \mathbf{X}$*;*

$$\mathop{\mathbb{E}}_{\mathrm{y}^\mathrm{t}\sim p(\mathrm{Y}^\mathrm{t}|\mathbf{x},\mathrm{t},\mathcal{D})}\left[\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathrm{y}^\mathrm{t})\right] \approx \mathop{\mathbb{E}}_{\mathrm{y}^\mathrm{t}\sim p(\mathrm{Y}^\mathrm{t}|\mathbf{x},\mathrm{t},\mathcal{D})}\left[\mathop{\mathrm{Var}}_{\boldsymbol{\omega}\sim p(\boldsymbol{\Omega}|\mathcal{D}_{\text{train}})}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x},\mathrm{t}'))\right], \tag{C.12}$$

where for binary $\mathrm{T} = \mathrm{t}$, $\mathrm{t}' = (1-\mathrm{t})$.

Proof.

$$\mathbb{E}_{\mathrm{y}^\mathrm{t}\sim p(\mathrm{Y}^\mathrm{t}|\mathbf{x},\mathrm{t},\mathcal{D})}\left[\mathrm{Var}(\widehat{\tau}_{\boldsymbol{\Omega}} \mid \mathrm{y}^\mathrm{t})\right] = \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\left[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\left(\widehat{\tau}_{\boldsymbol{\omega}} - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\widehat{\tau}_{\boldsymbol{\omega}} \mid \mathrm{y}^\mathrm{t}\right]\right)^2 \mid \mathrm{y}^\mathrm{t}\right]\right], \tag{C.13a}$$

$$= \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\left[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\left(\mathbb{E}[\mathrm{Y}^1 - \mathrm{Y}^0 \mid \mathbf{x},\boldsymbol{\omega}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathbb{E}[\mathrm{Y}^1 - \mathrm{Y}^0 \mid \mathbf{x},\boldsymbol{\omega}] \mid \mathrm{y}^\mathrm{t}\right]\right)^2 \mid \mathrm{y}^\mathrm{t}\right]\right], \tag{C.13b}$$

$$= \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\Bigg[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\Big[\big(\mathbb{E}[\mathrm{Y}^1 \mid \mathbf{x},\boldsymbol{\omega}] - \mathbb{E}[\mathrm{Y}^0 \mid \mathbf{x},\boldsymbol{\omega}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathbb{E}[\mathrm{Y}^1 \mid \mathbf{x},\boldsymbol{\omega}] \mid \mathrm{y}^\mathrm{t}\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathbb{E}[\mathrm{Y}^0 \mid \mathbf{x},\boldsymbol{\omega}] \mid \mathrm{y}^\mathrm{t}\right]\Big)^2 \mid \mathrm{y}^\mathrm{t}\Big]\Bigg], \tag{C.13c}$$

$$= \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\Bigg[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\Big[\Big(\big(\mathbb{E}[\mathrm{Y}^1 \mid \mathbf{x},\boldsymbol{\omega}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathbb{E}[\mathrm{Y}^1 \mid \mathbf{x},\boldsymbol{\omega}] \mid \mathrm{y}^\mathrm{t}\right]\big) - \big(\mathbb{E}[\mathrm{Y}^0 \mid \mathbf{x},\boldsymbol{\omega}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathbb{E}[\mathrm{Y}^0 \mid \mathbf{x},\boldsymbol{\omega}] \mid \mathrm{y}^\mathrm{t}\right]\big)\Big)^2 \mid \mathrm{y}^\mathrm{t}\Big]\Bigg], \tag{C.13d}$$

$$= \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\Bigg[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\Big[\Big(\big(\mathbb{E}[\mathrm{Y}^\mathrm{t} \mid \mathbf{x},\boldsymbol{\omega}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathbb{E}[\mathrm{Y}^\mathrm{t} \mid \mathbf{x},\boldsymbol{\omega}] \mid \mathrm{y}^\mathrm{t}\right]\big) - \big(\mathbb{E}[\mathrm{Y}^{\mathrm{t}'} \mid \mathbf{x},\boldsymbol{\omega}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathbb{E}[\mathrm{Y}^{\mathrm{t}'} \mid \mathbf{x},\boldsymbol{\omega}] \mid \mathrm{y}^\mathrm{t}\right]\big)\Big)^2 \mid \mathrm{y}^\mathrm{t}\Big]\Bigg], \tag{C.13e}$$

$$= \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\Bigg[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega}|\mathrm{y}^\mathrm{t})}\Big[\Big(\big(\mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^\mathrm{t}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega}|\mathrm{y}^\mathrm{t})}\left[\mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^\mathrm{t}]\right]\big) - \big(\mathop{\mathbb{E}}_{p(\mathrm{y}^{\mathrm{t}'}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^{\mathrm{t}'}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega}|\mathrm{y}^\mathrm{t})}\left[\mathop{\mathbb{E}}_{p(\mathrm{y}^{\mathrm{t}'}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^{\mathrm{t}'}]\right]\big)\Big)^2\Big]\Bigg], \tag{C.13f}$$

$$= \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\Bigg[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega}|\mathrm{y}^\mathrm{t})}\Big[\Big(\underbrace{\big(\mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^\mathrm{t}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega}|\mathrm{y}^\mathrm{t})}\left[\mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^\mathrm{t}]\right]\big)}_{\approx 0} - \big(\mathop{\mathbb{E}}_{p(\mathrm{y}^{\mathrm{t}'}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^{\mathrm{t}'}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathop{\mathbb{E}}_{p(\mathrm{y}^{\mathrm{t}'}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^{\mathrm{t}'}]\right]\big)\Big)^2\Big]\Bigg], \tag{C.13g}$$

$$\approx \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\left[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega}|\mathrm{y}^\mathrm{t})}\left[\left(\mathop{\mathbb{E}}_{p(\mathrm{y}^{\mathrm{t}'}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^{\mathrm{t}'}] - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\mathop{\mathbb{E}}_{p(\mathrm{y}^{\mathrm{t}'}|\mathbf{x},\boldsymbol{\omega})}[\mathrm{y}^{\mathrm{t}'}]\right]\right)^2\right]\right], \tag{C.13h}$$

$$= \mathop{\mathbb{E}}_{p(\mathrm{y}^\mathrm{t})}\left[\mathop{\mathbb{E}}_{p(\boldsymbol{\omega}|\mathrm{y}^\mathrm{t})}\left[\left(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x},\mathrm{t}') - \mathop{\mathbb{E}}_{p(\boldsymbol{\omega})}\left[\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x},\mathrm{t}')\right]\right)^2\right]\right], \tag{C.13i}$$

$$= \mathop{\mathbb{E}}_{\mathrm{y}^\mathrm{t}\sim p(\mathrm{Y}^\mathrm{t}|\mathbf{x},\mathrm{t},\mathcal{D})}\left[\mathop{\mathrm{Var}}_{\boldsymbol{\omega}\sim p(\boldsymbol{\Omega}|\mathcal{D}_{\text{train}})}(\widehat{\mu}_{\boldsymbol{\omega}}(\mathbf{x},\mathrm{t}'))\right], \tag{C.13j}$$

where (C.13a) by definition of variance; (C.13a)-(C.13b) by definition of $\widehat{\tau}_{\boldsymbol{\omega}}$; (C.13b)-(C.13c) by linearity of expectations; (C.13c)-(C.13d) by grouping terms; (C.13d)-(C.13e) by symmetry of the square; (C.13e)-(C.13f) by rewriting expectations in terms of densities; (C.13f)-(C.13g)

the observed potential outcome does not have an effect on the expectation of the model for the counterfactual outcome; (C.13g)-(C.13h) we drop the term as an approximation as we cannot estimate here how much the expected outcome is going to change—the conservative assumption is that will not change; (C.13h)-(C.13i) by definition of $\widehat{\mu}_{\boldsymbol{\omega}}$; (C.13i)-(C.13j) by definition of variance; □

Baselines

S-type error Information Gain

In their work, Sundin et al. ((2019)) assume that the underlying model is a Gaussian Process (GP) and also that they have access to the counterfactual outcome. Although GPs are suitable for uncertainty estimation, they do not scale up to high dimensional datasets (e.g. images). We propose to use Deep Ensembles and DUE for alleviating the capabilities issues and we modified the objective to be more suitable for our architecture.

Following the formulation from Houlsby et al. ((2011)), the acquisition strategy becomes

$$\underset{x}{\operatorname{argmax}}\, \mathbb{H}[\gamma|x, D] - \mathbb{E}_{p(\theta|D)}[\gamma|x, \theta],$$

where $\gamma(x) = \text{probit}^{-1}\left(-\frac{|\mathbf{E}_{p(\tau|\mathbf{x},\mathcal{D}_{\text{train}})}[\tau]|}{\sqrt{\text{Var}(\tau|\mathbf{x},\mathcal{D}_{\text{train}})}}\right)$, $\text{probit}^{-1}(\cdot)$ is the cumulative distribution function of normal distribution and $p(\gamma|x, D) = \text{Bernoulli}(\gamma)$. With DUE (Deep Kernel Learning method) and Deep Ensembles (samples from $p(\theta|D)$) we can compute those terms similarly to how we implemented our BALD objectives. Below is an example of how this was implemented in PyTorch:

```
tau_mu = mu1s - mu0s
tau_var = var1s + var0s + 1e-07
gammas = torch.distributions.normal.Normal(0, 1).cdf(
    -tau_mu.abs() / tau_var.sqrt()
)
gamma = gammas.mean(-1)
predictive_entropy = dist.Bernoulli(gamma).entropy()
conditional_entropy = dist.Bernoulli(gammas).entropy().mean(-1)
# it can get negative very small number
```

```
# because of numerical instabilities
scores = (predictive_entropy - conditional_entropy).clamp_min(1e-07)
```

Datasets

Synthetic Data

We modify the synthetic dataset presented by Kallus et al. ((2019)). Our dataset is described by the following structural causal model (SCM):

$$\mathbf{x} \coloneqq N_{\mathbf{x}}, \tag{C.14a}$$

$$\mathrm{t} \coloneqq N_{\mathrm{t}}, \tag{C.14b}$$

$$\mathrm{y} \coloneqq (2\mathrm{t}-1)\mathbf{x} + (2\mathrm{t}-1) - 2\sin(2(2\mathrm{t}-1)\mathbf{x}) + 2(1+0.5\mathbf{x}) + N_{\mathrm{y}}, \tag{C.14c}$$

where $N_{\mathbf{x}} \sim \mathcal{N}(0,1)$, $N_{\mathrm{t}} \sim \mathrm{Bern}(\mathrm{sigmoid}(2\mathbf{x}+0.5))$, and $N_{\mathrm{y}} \sim \mathcal{N}(0,1)$.

Each random realization of the simulated dataset generates 10000 pool set examples, 1000 validation examples, and 1000 test examples. In the experiments, we report results over 40 random realizations. The seeds for the random number generators are i, $i+1$, and $i+2$; $\{i \in [0,1,\ldots,19]\}$, for the training, validation, and test sets, respectively.

IHDP Data.

Infant Health and Development Program (IHDP) is a semi-synthetic dataset ((Hill, 2011, Shalit et al., 2017)) commonly used in literature to study the performance of causal effect estimation methods. The dataset consists of 747 cases, out of which 139 are assigned in treatment group and 608 in control. Each unit is represented by 25 covariates describing different aspects of the infants and their mothers. We report results over 200 random realizations of response surface B described by Hill ((2011)).

CMNIST Data.

Following the setup from Jesson et al. ((2021)), we use a simulated dataset based on MNIST ((LeCun, 1998)). CMNIST is described by the following SCM:

$$\mathbf{x} \coloneqq N_{\mathbf{x}}, \tag{C.15a}$$

$$\phi \coloneqq \left(\text{clip}\left(\frac{\mu_{N_{\mathbf{x}}} - \mu_{\text{c}}}{\sigma_{\text{c}}}; -1.4, 1.4\right) - \text{Min}_{\text{c}}\right) \frac{\text{Max}_{\text{c}} - \text{Min}_{\text{c}}}{1.4 - \text{-}1.4} \tag{C.15b}$$

$$\text{t} \coloneqq N_{\text{t}}, \tag{C.15c}$$

$$\text{y} \coloneqq (2\text{t} - 1)\phi + (2\text{t} - 1) - 2\sin(2(2\text{t} - 1)\phi) + 2(1 + 0.5\phi) + N_{\text{y}}, \tag{C.15d}$$

where N_{t} (swapping x for ϕ), and N_{y} are as described in Appendix C. $N_{\mathbf{x}}$ is a sample of an MNIST image. The sampled image has a corresponding label $c \in [0, \ldots, 9]$. $\mu_{N_{\mathbf{x}}}$ is the average intensity of the sampled image. μ_{c} and σ_{c} are the mean and standard deviation of the average image intensities over all images with label c in the MNIST training set. In other words, $\mu_{\text{c}} = \mathbb{E}[\mu_{N_{\mathbf{x}}} \mid \text{c}]$ and $\sigma_{\text{c}}^2 = \text{Var}[\mu_{N_{\mathbf{x}}} \mid \text{c}]$. To map the high dimensional images $\mathbf{x}$ onto a one-dimensional manifold ϕ with domain $[-3, 3]$ above, we first clip the standardized average image intensity on the range $(-1.4, 1.4)$. Each digit class has its own domain in ϕ, so there is a linear transformation of the clipped value onto the range $[\text{Min}_{\text{c}}, \text{Max}_{\text{c}}]$. Finally, $\text{Min}_{\text{c}} = -2 + \frac{4}{10}\text{c}$, and $\text{Max}_{\text{c}} = -2 + \frac{4}{10}(\text{c} + 1)$.

For each random realization of the dataset, the MNIST training set is split into training ($n = 35000$) and validation ($n = 15000$) subsets using the scikit-learn function train_test_split(). The test set is generated using the MNIST test set ($n = 10000$). The random seeds are $\{i \in [0, 1, \ldots, 19]\}$ for the 10 random realizations generated.

More Results

Model Architectures

For deep ensembles, we use an ensemble of TarNETs Shalit et al. ((2017)). For Due, we append the treatment variable to the features extracted, then define the GP over that input.

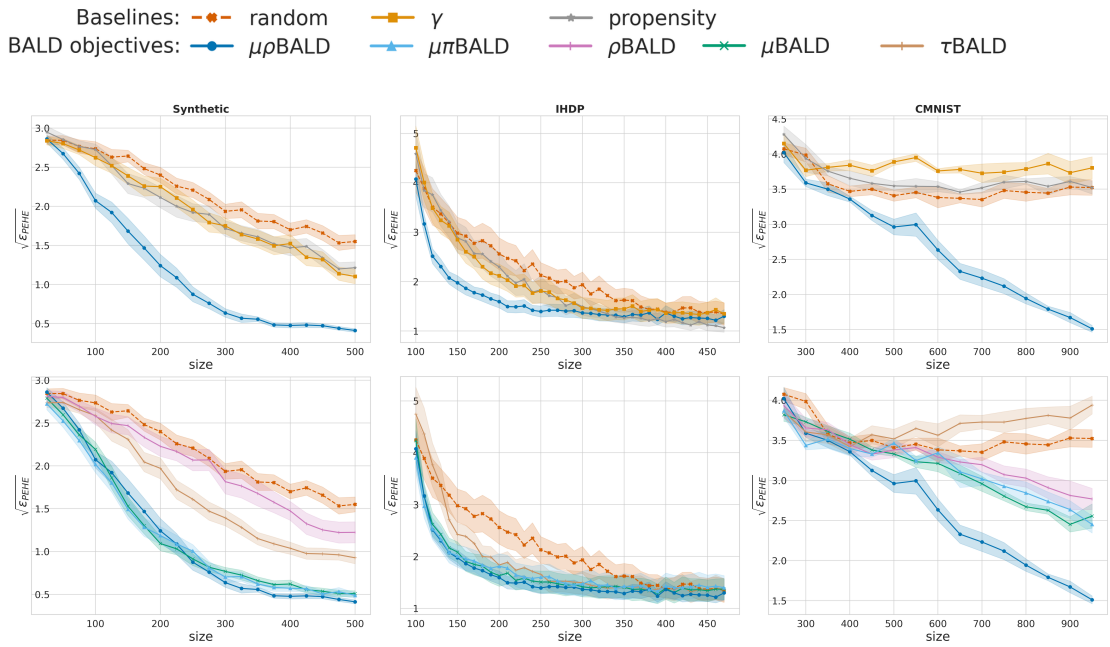


Figure C.1: $\sqrt{\epsilon_{PEHE}}$ performance (shaded standard error) for Deep Ensembles based models. **(left to right) synthetic** (20 seeds), **IHDP** (50 seeds) and *CMNIST* (5 seeds) dataset results, **(top to bottom)** comparison with baselines, comparison between BALD objectives. We observe that BALD objectives outperform the **random**, $\boldsymbol{\gamma}$ and **propensity** acquisition functions significantly, suggesting that epistemic uncertainty aware methods that target reducible uncertainty can be more sample efficient.

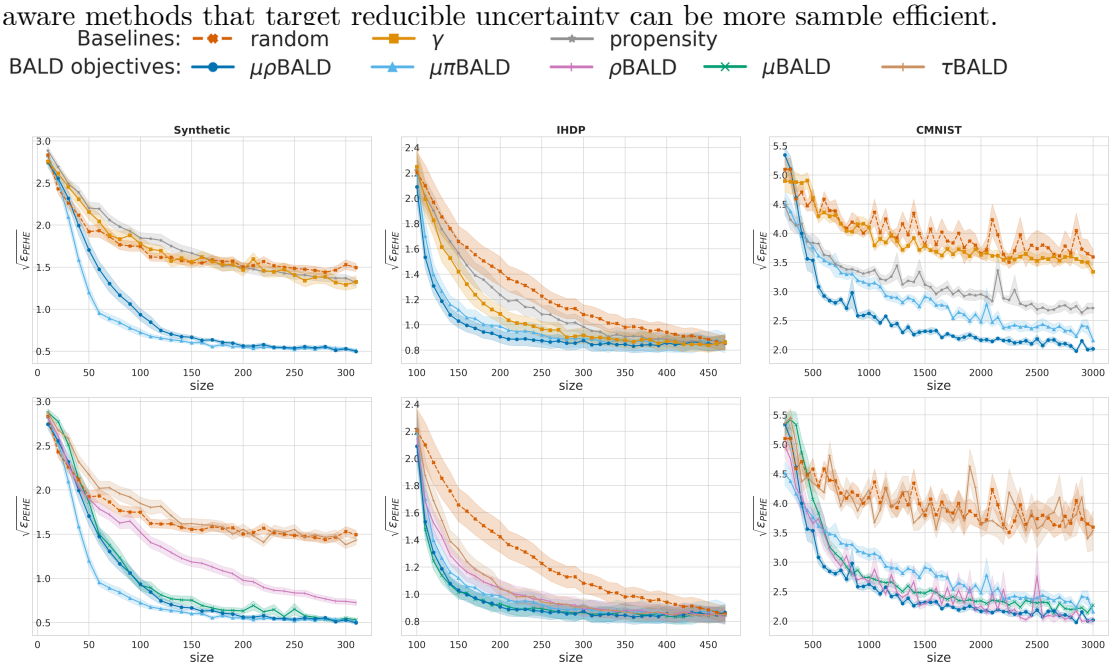


Figure C.2: $\sqrt{\epsilon_{PEHE}}$ performance (shaded standard error) for DUE models. **(left to right) synthetic** (40 seeds), and **IHDP** (200 seeds). We observe that BALD objectives outperform the **random**, $\boldsymbol{\gamma}$ and **propensity** acquisition functions significantly, suggesting that epistemic uncertainty aware methods that target reducible uncertainty can be more sample efficient.

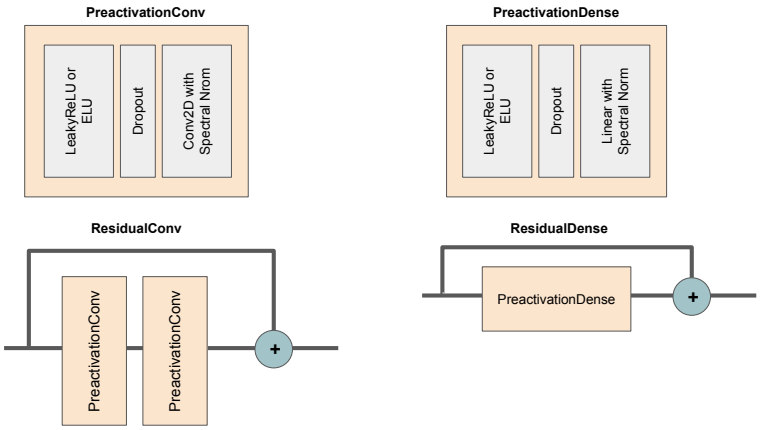


Figure C.3: `PreactivationConv` is a convolution layer with LeakyReLU (or ELU when slope is negative) activation, dropout and spectral norm applied ((Gouk et al., 2021, Miyato et al., 2018)). Similarly, `PreactivationDense` is a dense layer with BatchNorm ((Ioffe and Szegedy, 2015)), LeakyReLU (or ELU when slope is negative) activation and spectral norm applied ((Gouk et al., 2021, Miyato et al., 2018)). `ResidualConv` is the residual convolution layer, defined as `PreactivationConv(PreactivationConv(x)) + SpectralNorm(1x1Conv(x))` and `ResidualDense` are residual dense layers, defined as `PreactivationDense(x)+x)`.

All experiments were trained using Adam optimizer ((Kingma and Ba, 2017)).

Hyper-parameters

We use ray tune ((Liaw et al., 2018)) with the hyperopt Bergstra et al. ((2013)) search algorithm to optimize our network hyper-parameters. The hyper-parameter search spaces are given in Table C.1 and Table C.2. The hyper-paramter optimization objective for each dataset is the expected batch-wise log-likelihood of the validation data for a single dataset realization with random seed 1331. The final hyper-parameters are given in Table C.3 and Table C.4.

Table C.1: Hyper-parameter search space for **Deep Ensemble**

Hyper-parameter	Search Space
hidden units	[100, 200, 400]
network depth	[2, 3, 4]
negative slope	[ReLU Agarap ((2018)), 0.1, 0.2, ELU Clevert et al. ((2016))]
dropout rate	[0.05, 0.1, 0.2, 0.5]
spectral norm	[None, 0.95, 1.5, 3.0]
batch size	[32, 64, 100, 200]
learning rate	[2e-4, 5e-4, 1e-3]

Table C.2: Hyper-parameter search space for **DUE**

Hyper-parameter	Search Space
kernel	[RBF, Matern]
ν (Matern)	[0.5, 1.5, 2.5]
inducing points	[20, 50, 100, 200]
hidden units	[100, 200, 400]
network depth	[2, 3, 4]
negative slope	[ReLU Agarap ((2018)), 0.1, 0.2, ELU Clevert et al. ((2016))]
dropout rate	[0.05, 0.1, 0.2, 0.5]
spectral norm	[None, 0.95, 1.5, 3.0]
batch size	[32, 64, 100, 200]
learning rate	[2e-4, 5e-4, 1e-3]

Table C.3: Training hyper parameters for **Deep Ensemble** experiments

Parameter	**Synthetic**	**IHDP**	**CMNIST**
dim hidden	100	400	100
dropout	0.0	0.15	0.1
depth	4	3	3
spectral norm	12	0.95	24
learning rate	0.001	0.001	0.001
non-linearity	ReLU	ELU	ReLU

Table C.4: Training hyper parameters for **DUE** experiments

Parameter	**Synthetic**	**IHDP**	**CMNIST**
kernel	RBF	Matern ($\nu = 1.5$)	RBF
inducing points	20	100	100
dim hidden	100	200	200
dropout	0.2	0.1	0.05
depth	3	3	2
batch size	200	100	64
spectral norm	0.95	0.95	3.0
learning rate	0.001	0.001	0.001
non-linearity	ReLU	ELU	ELU

D

Causal Bayesian Experimental Design

Theoretical Results

Deriving the Mutual Information over Outcomes

In the following lemma, we derive the mutual information over outcomes given in (5.2).

Lemma 3.2.

$$\begin{aligned} &\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j, v)\}, \mathcal{D}) \\ &= - \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \{(j,v)\}, \mathcal{D})} \left[p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}) \right] \right) \right] \\ &\quad + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}) \right) \right] \right] \end{aligned} \tag{D.1}$$

Proof.

$$\mathrm{I}(\mathbf{Y};\mathbf{\Phi} \mid \{(j,v)\},\mathcal{D}) = \mathrm{H}(\mathbf{Y} \mid \{(j,v)\},\mathcal{D}) - \mathrm{H}(\mathbf{Y} \mid \mathbf{\Phi},\{(j,v)\},\mathcal{D}) \tag{D.2a}$$

$$= \mathrm{H}(\mathbf{Y} \mid \{(j,v)\},\mathcal{D}) - \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D})}\left[\mathrm{H}(\mathbf{Y} \mid \boldsymbol{\phi},\{(j,v)\})\right] \tag{D.2b}$$

$$= - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})}\left[\log\left(p(\mathbf{y} \mid \{(j,v)\},\mathcal{D})\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}|\boldsymbol{\phi},\{(j,v)\})}\left[\log\left(p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right)\right]\right] \tag{D.2c}$$

$$= - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})}\left[\log\left(\int_{\boldsymbol{\phi}} p(\mathbf{y},\boldsymbol{\phi} \mid \{(j,v)\},\mathcal{D}) d\boldsymbol{\phi}\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}|\boldsymbol{\phi},\{(j,v)\})}\left[\log\left(p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right)\right]\right] \tag{D.2d}$$

$$= - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})}\left[\log\left(\int_{\boldsymbol{\phi}} p(\boldsymbol{\phi} \mid \{(j,v)\},\mathcal{D}) p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\},\mathcal{D}) d\boldsymbol{\phi}\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi},|\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}|\boldsymbol{\phi},\{(j,v)\})}\left[\log\left(p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right)\right]\right] \tag{D.2e}$$

$$= - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})}\left[\log\left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\{(j,v)\},\mathcal{D})}\left[p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right]\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}|\boldsymbol{\phi},\{(j,v)\})}\left[\log\left(p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right)\right]\right]. \tag{D.2f}$$

□

Estimating the Mutual Information over Outcomes

For models that allow for evaluation of the experimental outcome density (likelihood), $p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})$, we can use the following estimator for $\mathrm{I}(\mathbf{Y};\mathbf{\Phi} \mid \{(j,v)\},\mathcal{D})$:

$$\widehat{\mathrm{I}}(\mathbf{Y};\mathbf{\Phi} \mid \{(j,v)\},\mathcal{D}) = \widehat{\mathrm{H}}(\mathbf{Y} \mid \{(j,v)\},\mathcal{D}) - \widehat{\mathrm{H}}(\mathbf{Y} \mid \mathbf{\Phi},\{(j,v)\},\mathcal{D}) \tag{D.3}$$

Algorithm 6: Mutual Information Computation

Input : Posterior $q(\boldsymbol{\phi} \mid \mathcal{D}_{\text{obs}} \cup \mathcal{D}_{\text{int}})$, Number of posterior samples c, Number of interventional samples m, Intervention $\{(j,v)\}$. We notate as *int* the interventional and *obs* the observational data.

▷`Sample from the posterior`

1 $\{\widehat{\boldsymbol{\phi}}_i \sim q(\boldsymbol{\phi} \mid \mathcal{D}_{\text{obs}} \cup \mathcal{D}_{\text{int}})\}_{i=1}^{c}$

▷`Sample from mutilated SCMs`

2 $\{\widehat{\mathbf{y}}_{i,j,k} \sim p(\mathbf{y} \mid \widehat{\boldsymbol{\phi}}_i,\{(j,v)\})\}_{k=1}^{m}$

3 **return** $-\frac{1}{c\times m}\sum_{i=1}^{c}\sum_{k=1}^{m}\log\left(\frac{1}{c}\sum_{l=1}^{c} p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l,\{(j,v)\})\right)$ $+\frac{1}{c\times m}\sum_{i=1}^{c}\sum_{k=1}^{m}\log\left(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_i,\{(j,v)\})\right)$

Definition 7. *The Monte Carlo estimator,* $\widehat{\mathrm{H}}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$*, of the marginal entropy of the experimental outcomes,* $\mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$*, is given by:*

$$-\frac{1}{c_o \times m} \sum_{i=1}^{c_o} \sum_{k=1}^{m} \log \left(\frac{1}{c_{in}} \sum_{l=1}^{c_{in}} p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\}) \right), \tag{D.4}$$

where $\widehat{\mathbf{y}}_{i,k} \sim p(\mathbf{y} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\})$ *is one of* m *samples from the density parameterised by the* i*th of* c_o *SCMs* $\widehat{\boldsymbol{\phi}}_i \sim p(\boldsymbol{\phi} \mid \mathcal{D})$ *augmented by intervention* $\{(j,v)\}$*. The likelihood of the sample* $\widehat{\mathbf{y}}_{i,k}$ *is then evaluated under the parameterisation of the* l*th of* c_{in} *additional SCMs* $\widehat{\boldsymbol{\phi}}_l \sim p(\boldsymbol{\phi} \mid \mathcal{D})$ *augmented by intervention* $\{(j,v)\}$*.*

$\widehat{\mathrm{H}}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$ is a consistent but biased estimator of $\mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$ due to the expectation inside of the nonlinear log function. Alternatively, we can look at the following lower bound on $\mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$:

$$\begin{aligned}
\mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) &= \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right] \right) \right], \\
&\leq - \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right) \right] \right],
\end{aligned}$$

by Jensen's inequality. We can then define an unbiased estimator of this lower bound.

Definition 8. *The unbiased Monte Carlo estimator,* $\widehat{\mathrm{H}}^*(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$*, of the lower bound on the marginal entropy of the experimental outcomes,*

$$- \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}, \mid \mathcal{D})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right) \right] \right],$$

is given by:

$$-\frac{1}{c_o \times c_{in} \times m} \sum_{i=1}^{c_o} \sum_{k=1}^{m} \sum_{l=1}^{c_{in}} \log \left(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\}) \right), \tag{D.5}$$

Finally, we define our estimator for $\mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D})$.

Definition 9. *The Monte Carlo estimator,* $\widehat{\mathrm{H}}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D})$*, of the entropy of the experimental outcomes conditioned on* $\boldsymbol{\Phi}$*,* $\mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D})$*, is given by:*

$$-\frac{1}{c_o \times m} \sum_{i=1}^{c_o} \sum_{k=1}^{m} \log \left(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_i, \{(j,v)\}) \right), \tag{D.6}$$

where $\widehat{\mathbf{y}}_{i,k} \sim p(\mathbf{y} \mid \widehat{\boldsymbol{\phi}}_i, \{(j,v)\})$ *is one of* m *samples from the density parameterised by the* i*th of* c_o *graphs* $\widehat{\boldsymbol{\phi}}_i \sim p(\boldsymbol{\phi} \mid \mathcal{D})$ *augmented by intervention* $\{(j,v)\}$*.*

Monte Carlo Estimator of the Batch Mutual Information

While Equation 5.2 pertains to MI for a single design, we present here the MI estimator for the batch design.

$$
\begin{aligned}
\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \boldsymbol{\Xi}, \mathcal{D}) &= \sum_{\{(j,v)\} \in \boldsymbol{\Xi}} \mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) \\
&= \sum_{\{(j,v)\} \in \boldsymbol{\Xi}} \mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) - \mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D}) \\
&= - \sum_{\{(j,v)\} \in \boldsymbol{\Xi}} \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right] \right) \right] \\
&\quad + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right) \right] \right]
\end{aligned}
\tag{D.7}
$$

Mutual Information Submodularity and Monotonicity Proofs

Theorem 4. $\mathbf{I}(Y; \omega \mid X)$ *is submodular.*

Proof. The proof follows the structure of ((Kirsch et al., 2019, Appendix A)).

$$\mathbf{I}(Y \cup \{y_1\}; \omega \mid X \cup \{x_1\}) + \mathbf{I}(Y \cup \{y_2\}; \omega \mid X \cup \{x_2\}) \geq$$
$$\mathbf{I}(Y \cup \{y_1, y_2\}; \omega \mid X \cup \{x_1, x_2\}) + \mathbf{I}(Y; \omega \mid X)$$
(conditioning on RVs that are independent of the non-conditioning RVs) $\Leftrightarrow$
$$\mathbf{I}(Y \cup \{y_1\}; \omega \mid X \cup \{x_1, x_2\}) + \mathbf{I}(Y \cup \{y_2\}; \omega \mid X \cup \{x_1, x_2\}) \geq$$
$$\mathbf{I}(Y \cup \{y_1, y_2\}; \omega \mid X \cup \{x_1, x_2\}) + \mathbf{I}(Y; \omega \mid X \cup \{x_1, x_2\})$$
(substituting X $\cup \{x_1, x_2\}$ with X^+) $\Leftrightarrow$
$$\mathbf{I}(Y \cup \{y_1\}; \omega \mid X^+) + \mathbf{I}(Y \cup \{y_2\}; \omega \mid X^+) \geq$$
$$\mathbf{I}(Y \cup \{y_1, y_2\}; \omega \mid X^+) + \mathbf{I}(Y; \omega \mid X^+)$$
(subtract $2 * \mathbf{I}(Y; \omega \mid X^+)$ from both sides
and use the identity $\mathbf{I}(A, B; C) - \mathbf{I}(B; C) = \mathbf{I}(A; C \mid B)$) $\Leftrightarrow$
$$\mathbf{I}(y_1; \omega \mid Y, X^+) + \mathbf{I}(y_2; \omega \mid Y, X^+) \geq \mathbf{I}(y_1, y_2; \omega \mid Y, X^+)$$
$\Leftrightarrow$
$$\mathbf{I}(y_1; \omega \mid Y, X^+) + \mathbf{I}(y_2; \omega \mid Y, X^+) =$$
$$\underbrace{(h(y_1 \mid Y, X^+) + h(y_2 \mid Y, X^+))}_{\geq h(y_1, y_2 \mid Y, X^+) \text{ ((Thomas and Joy, 2006, p.253))}} \quad \underbrace{- (h(y_1 \mid Y, X^+, \omega) + h(y_2 \mid Y, X^+, \omega))}_{= h(y_1, y_2 \mid \omega, Y, X^+) \text{ (because } y_1 \perp\!\!\!\perp y_2 \mid \omega)} \geq$$
$$h(y_1, y_2 \mid Y, X^+) - h(y_1, y_2 \mid \omega, Y, X^+) = \mathbf{I}(y_1, y_2; \omega \mid Y, X^+)$$

□

Theorem 5. $\mathbf{I}(Y; \omega \mid X)$ *is non-decreasing.*

Proof.

$$\mathbf{I}(Y \cup \{y\}; \omega \mid X \cup \{x\}) - \mathbf{I}(Y; \omega \mid X) =$$
(conditioning on RVs that are independent of the non-conditioning RVs)
$$\mathbf{I}(Y \cup \{y\}; \omega \mid X \cup \{x\}) - \mathbf{I}(Y; \omega \mid X \cup \{x\}) =$$
(use the identity $\mathbf{I}(A, B; C) - \mathbf{I}(B; C) = \mathbf{I}(A; C \mid B)$)
$$\mathbf{I}(\{y\}; \omega \mid Y, X \cup \{x\}) \geq 0$$

□

Relation to MI Approximation in ABCD

Here we demonstrate that though ABCD ((Agrawal et al., 2019)) uses an importance weighted estimate of mutual information, for the specific choice of importance weights used in ABCD, the MI estimate turns out to be the same as the one used in this work.

We note that ABCD decomposes the MI as *entropy over the SCM* as opposed to the *entropy over outcomes* used in this work.

Entropy Over SCM

The mutual information in (5.1) can be written as:

$$\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \mathrm{H}(\boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) - \mathrm{H}(\boldsymbol{\Phi} \mid \mathbf{Y}, \{(j,v)\}, \mathcal{D}) \tag{D.8}$$

where $\mathrm{H}(\cdot)$ is the *expected entropy*. As the posterior $p(\mathbf{g}, \boldsymbol{\theta}|\mathcal{D})$ does not change as a result of conditioning on the design choice $\{(j,v)\}$, the first entropy term is constant wrt $\{(j,v)\}$. Hence, selecting the most informative target corresponds to minimising the conditional entropy of the parameters $\boldsymbol{\Phi}$.

$$\begin{aligned} &\mathrm{H}(\boldsymbol{\Phi} \mid \mathbf{Y}, \{(j,v)\}, \mathcal{D}) \\ &= - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathbf{y},\{(j,v)\},\mathcal{D})} \left[\log p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j,v)\}, \mathcal{D}) \right] \right] \end{aligned} \tag{D.9}$$

The above equation cannot be estimated from samples of $q(\boldsymbol{\phi} \mid \mathcal{D}) \approx p(\boldsymbol{\phi} \mid \mathcal{D})$ since the posterior of the SCM would change when the interventional outcome $\mathbf{y}$ is conditioned on. To address this problem, ABCD Agrawal et al. ((2019)) proposes to use weighted importance sampling with weights $w = p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D})$ and use samples from $q(\boldsymbol{\phi} \mid \mathcal{D})$.

Definition 10. *The weighted importance sampling estimate of entropy over SCM* (D.8) *with weights* $w(\boldsymbol{\phi})$ *is given by*

$$\widehat{\mathrm{I}}_{WIS} = \frac{1}{c_o} \sum_{i=1}^{c_o} \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\log w(\widehat{\boldsymbol{\phi}}_i) \right] - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\log \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D},\{(j,v)\})} w(\boldsymbol{\phi}) \right] \right] \tag{D.10}$$

Entropy Over Outcomes.

We can instead consider an alternative factorisation of (5.1) which would not require importance sampling and also compute entropies in the lower dimensional space of experimental outcomes, as given in Equation 5.2.

Definition 11. *The Monte Carlo estimate of entropy over outcomes* (5.2) *is given by*

$$\begin{aligned}\widehat{\mathrm{I}}_{MC} = &\frac{1}{c_o \times m}\sum_{i=1}^{c_o}\sum_{k=1}^{m}\log\left(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_i, \{(j,v)\})\right) \\ &- \frac{1}{c_o \times m}\sum_{i=1}^{c_o}\sum_{k=1}^{m}\log\left(\frac{1}{c_{in}}\sum_{l=1}^{c_{in}} p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\})\right)\end{aligned} \tag{D.11}$$

Relation between Approximations with Entropy over SCM and Entropy over Outcomes

We prove below that for specific choice of importance weights $w(\boldsymbol{\phi}) :== p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D})$ used in ABCD, the MI approximations due to the above two factorizations are the same.. Let $\widehat{\mathrm{I}}_{\mathrm{WIS}}$ (D.10) be the weighted importance sampling estimate of entropy over SCM (D.8) with weights $w(\boldsymbol{\phi})$ and $\widehat{\mathrm{I}}_{\mathrm{MC}}$ (D.11) be the Monte Carlo estimate of entropy over outcomes (5.2). Then, $\widehat{\mathrm{I}}_{\mathrm{WIS}} = \widehat{\mathrm{I}}_{\mathrm{MC}}$ if $w(\boldsymbol{\phi}) = p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D})$.

Proof. Consider the entropy over SCM:

$$\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \mathrm{H}(\boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) - \mathrm{H}(\boldsymbol{\Phi} \mid \mathbf{Y}, \{(j,v)\}, \mathcal{D})$$

$$\begin{aligned}\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = &\mathrm{H}(\boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) \\ &+ \mathop{\mathbb{E}}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}\mid\mathbf{y},\{(j,v)\},\mathcal{D})}\left[\log p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j,v)\}, \mathcal{D})\right]\right]\end{aligned} \tag{D.12}$$

Consider the importance weighted estimate of the above equation with weights $w(\boldsymbol{\phi})$. We can rewrite $p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j,v)\}, \mathcal{D})$ as:

$$p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j,v)\}, \mathcal{D}) = \frac{w(\boldsymbol{\phi})p(\boldsymbol{\phi} \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D},\{(j,v)\})}\left[w(\boldsymbol{\phi})\right]} \tag{D.13}$$

Let $\{\widehat{\boldsymbol{\phi}}_i \sim p(\boldsymbol{\phi} \mid \mathcal{D})\}_{i=1}^{c_o}$, using (D.13) in (D.12),

$$\widehat{\mathrm{I}}_{\mathrm{WIS}}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \mathrm{H}(\boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) + \frac{1}{c_o} \sum_{i=1}^{c_o} \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \log \left[\frac{w(\widehat{\boldsymbol{\phi}}_i) p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\boldsymbol{\phi} \mid \mathcal{D}, \{(j,v)\})} [w(\boldsymbol{\phi})]} \right] \tag{D.14a}$$

Furthermore, using a Monte-Carlo estimate on first term with $\widehat{\boldsymbol{\phi}}_i$, we get

$$\widehat{\mathrm{I}}_{\mathrm{WIS}}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \frac{1}{c_o} \sum_{i=1}^{c_o} \Bigg[- \log p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\}) + \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \log \left(\frac{w(\widehat{\boldsymbol{\phi}}_i) p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\boldsymbol{\phi} \mid \mathcal{D}, \{(j,v)\})} [w(\boldsymbol{\phi})]} \right) \Bigg] \tag{D.14b}$$

Focusing on the second term,

$$\begin{aligned} &\mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \log \left[\frac{w(\widehat{\boldsymbol{\phi}}_i) p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\boldsymbol{\phi} \mid \mathcal{D}, \{(j,v)\})} [w(\boldsymbol{\phi})]} \right] \\ &\quad = \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log w(\widehat{\boldsymbol{\phi}}_i) \right] + \log p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\}) \\ &\quad\quad - \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D}, \{(j,v)\})} w(\boldsymbol{\phi}) \right] \right] \end{aligned} \tag{D.15}$$

Plugging the above result back in (D.14b) and noticing that second term in the above equation cancels with first term in (D.14b), we get:

$$\widehat{\mathrm{I}}_{\mathrm{WIS}} = \frac{1}{c_o} \sum_{i=1}^{c_o} \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log w(\widehat{\boldsymbol{\phi}}_i) \right] - \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D}, \{(j,v)\})} w(\boldsymbol{\phi}) \right] \right] \tag{D.16}$$

$\widehat{\mathrm{I}}_{\mathrm{MC}}$ is given by (D.4)+(D.6). We can notice that $\widehat{\mathrm{I}}_{\mathrm{WIS}} = \widehat{\mathrm{I}}_{\mathrm{MC}}$ if $w(\boldsymbol{\phi}) = p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D})$ and approximating remaining expectations in the above equation with Monte Carlo samples. □

Models

DiBS Hypermarameters

For optimizing DiBS Lorch et al. ((2021)) we used RMSProp with learning rate 0.005. Additionally, per dataset we set the following hyperparameters:

Nodes	Dataset	Graph Prior	Particles Transportation Steps	Number of Particles	Kernel
20	Scale Free	Scale Free	20000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Erds-Rényi	Erds-Rényi	20000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
50	Scale Free	Scale Free	20000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Erds-Rényi	Erds-Rényi	20000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
10	Ecoli1	Erds-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Ecoli2	Erds-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Yeast1	Erds-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Yeast2	Erds-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
50	Ecoli1	Erds-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Ecoli2	Erds-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Yeast1	Erds-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Yeast2	Erds-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)

Table D.1: Settings of DREAM experiments for nodes 10 and 50.

DAG Bootstrap

The DAG bootstrap bootstraps observations and interventions to infer a different causal structure per bootstrap. We used GIES as the causal inference algorithm because of the adaptation of GES on interventional data as well. In our experiments, we used the pcalg R implementation https://github.com/cran/pcalg/blob/master/R/gies.R to discover 100 graphs. Each graph can be seen as a posterior sample from $p(\mathbf{G} \mid \mathcal{D})$. For each of the sampled graphs G_i we compute the appropriate θ_{MLE} under linear Gaussian assumption for the conditional distributions.

Datasets and Experiment details

Synthetic Graphs Experiments

In the synthetic data experiments, we focus on two types of graphs. The Erds-Rényi and Scale Free.

Erdos-Renyi model:

We used `networkx`[1] and method `fast_gnp_random_graph` Batagelj and Brandes ((2005)) to generate graphs based on the Erds-Rényi model. We set expected number of edges per vertex to 1.

Scale Free (Barabasi-Albert) graphs:

[1] https://networkx.org/documentation/networkx-1.10/reference/generated/networkx.generators.random_graphs.fast_gnp_random_graph.html

We used igraph[2] package to generate the graphs. We set the expected number of edges per vertex to 1.

For all the synthetic graph experiments, we used batch size of 10 and number of iterations of 10.

DREAM Experiments

For the DREAM experiments, we used GeneNetWeaver Schaffter et al. ((2011)), a simulator of gene regulatory networks, based on stochastic differential equations. This simulator was used to generate data for *Dialogue for Reverse Engineering Assessments and Methods* (DREAM) Sachs et al. ((2005)) competition with three network inference challenges (`DREAM3`, `DREAM4` and `DREAM5`). We used the `GeneNetWeaver v3.1`[3].

Each experiment is parametrized as an xml file describing the network topology but also the crucial parameters of the stochastic differential equation that `GeneNetWeaver` simulates. In our experiments, we used `Ecoli1`, `Ecoli2`, `Yeast1` and `Yeast2` networks for 10 and 50 nodes.

Each experiment was initialized with 100 observational data. For the observational data, we used the steady state [4] of wild-type experiments. For the interventional data, we used the steady-state of knock-out experiments. Each observational or interventional sample was conducted by running the simulator with a different seed per draw.

	Dataset	Model	Starting Observational Samples	Batch Size	Number of Batches
10 nodes	`Ecoli1`	DiBS non linear	100	5	20
	`Ecoli2`	DiBS non linear	100	5	20
	`Yeast1`	DiBS non linear	100	5	20
	`Yeast2`	DiBS non linear	100	5	20
50 nodes	`Ecoli1`	DiBS non linear	100	20	20
	`Ecoli2`	DiBS non linear	100	20	20
	`Yeast1`	DiBS non linear	100	20	20
	`Yeast2`	DiBS non linear	100	20	20

Table D.2: Settings of DREAM experiments for nodes 10 and 50.

[2] https://igraph.org/python/api/latest/igraph._igraph.GraphBase.html#Barabasi
[3] https://github.com/tschaffter/genenetweaver
[4] Steady state is considered the result of the simulation of the SDE for maximum 2000 steps.

Bayesian Optimisation

Bayesian Optimisation (BO) Kushner ((1964)), Zhilinskas ((1975)), Močkus ((1975)) is a global optimisation technique for optimising black-box functions. More formally, for any function U defined on a set $\mathcal{X}$ which is expensive to evaluate, BO seeks to find the maximum of the function over the entire set $\mathcal{X}$ with as few evaluations as possible.

$$\max_{x \in \mathcal{X}} \mathrm{U}(x)$$

BO typically proceeds by placing a prior on the unknown function and obtaining the posterior over this function with the queried points $\mathbf{x}^* = \{x_1^*, \ldots, x_t^*\}$. A common prior is a Gaussian Process (GP) Rasmussen ((2003)) with mean 0 and covariance function defined by a kernel $k(x, x')$. Let $\mathrm{U}_{\mathbf{x}^*} = [\mathrm{U}(x_1^*), \ldots, \mathrm{U}(x_t^*)]$ denote the vector of function evaluations, $\mathbf{K}$ the kernel matrix with $\mathbf{K}_{i,j} = k(x_i^*, x_j^*)$ and $\mathbf{k}_{t+1} = [k(x_1^*, x_{t+1}), \ldots, k(x_t^*, x_{t+1})]$. The posterior predictive of point x_{t+1} can be obtained in closed form:

$$p(\mathrm{U}) \sim \mathcal{GP}(0, k)$$
$$p(\mathrm{U} \mid \mathbf{x}^*, \mathrm{U}_{\mathbf{x}^*}, x_{t+1}) = \mathcal{N}(\boldsymbol{\mu}(x_{t+1}), \boldsymbol{\sigma}^2(x_{t+1}))$$
$$\boldsymbol{\mu}(x_{t+1}) = \mathbf{k}_{t+1}^T (\mathbf{K} + \mathrm{I})^{-1} \mathrm{U}_{\mathbf{x}^*}$$
$$\boldsymbol{\sigma}^2(x_{t+1}) = k(x_{t+1}, x_{t+1}) - \mathbf{k}_{t+1}^T (\mathbf{K} + \mathrm{I})^{-1} \mathbf{k}_{t+1}$$

For the Gaussian Process, we used the following hyperparameters. Matern kernel, `length scale` 1.0, `length scale bounds` (lower=$1e-5$, upper=$1e5$), `Nu` (smoothness of learned function) 2.5. Also added $1e-6$ to the diagonal of the kernel matrix.

Related Work

Table D.3: Comparison of the proposed experimental design for causal discovery with existing experimental design for causal discovery techniques.

Method	Nonlinear	BOED	Scalable	Continuous	Finite Data	Setting the value
Murphy, Tong and Koller Murphy ((2001)), Tong and Koller ((2001))		✓			✓	
ABCD Agrawal et al. ((2019))		✓	✓	✓	✓	
Active NCM Scherrer et al. ((2021))	✓			✓	✓	
Active ICP Gamella and Heinze-Deml ((2020))	✓			✓	✓	
GP-UCB von Kügelgen et al. ((2019))	✓	✓		✓	✓	✓
Sussex Sussex et al. ((2021))		✓		✓		
Ours	✓	✓	✓	✓	✓	✓

Mutual Information per value for two Variables graph

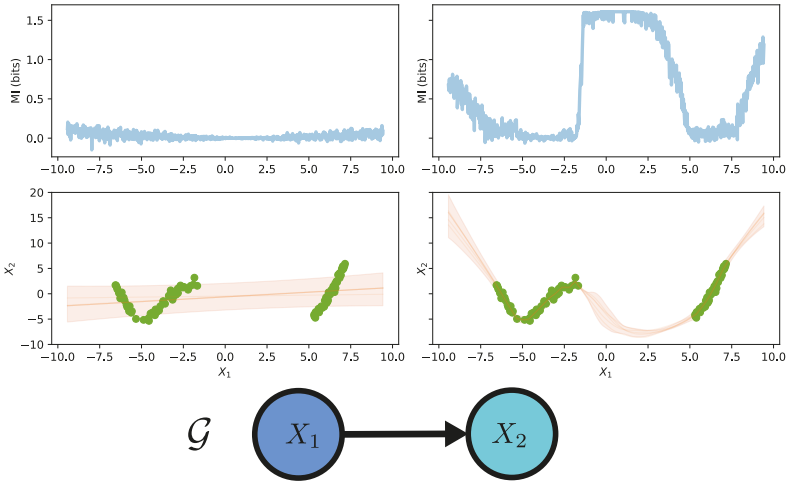


Figure D.1: Estimation of the Mutual Information using two variables model $\mathcal{G}$. In green we represent the interventional data. We train an ensemble of a linear (left plot) and a non-linear (right plot) function approximator (NN) parametrizing a Gaussian Distribution. We can see that in both cases, MI is influenced by the value of intervention $\text{do}(X_1 = x_1)$. In this experiment we used the BALD estimator of the MI.

metrics

$\mathbb{E}$**-SHD**: Defined as the *expected structural hamming distance* between samples from the posterior model over graphs and the true graph $\mathbb{E}\text{-SHD} \coloneqq \mathbb{E}_{\mathbf{g}\sim p(\mathcal{G}|\mathcal{D})}\Big[\text{SHD}(\mathbf{g}, \tilde{\mathbf{g}})\Big]$

$\mathbb{E}$**-SID**: As the SHD is agnostic to the notion of intervention, Peters and Bühlmann ((2015)) proposed the *expected structural interventional distance* ($\mathbb{E}$**-SID**) which quantifies the differences between graphs with respect to the causal inference state-

ments and interventional distributions.

AUROC: The *area under the receiver operating characteristic curve* of the binary classification task of predicting the presence/ absence of all edges.

AUPRC: The *area under the precision-recall curve* of the binary classification task of predicting the presence/ absence of all edges.

Metrics

$\mathbb{E}$**-SHD**: Defined as the *expected structural hamming distance* between samples from the posterior model over graphs and the true graph $\mathbb{E}\text{-SHD} \coloneqq \mathbb{E}_{\mathbf{g}\sim p(\mathcal{G}|\mathcal{D})}\Big[\text{SHD}(\mathbf{g}, \tilde{\mathbf{g}})\Big]$

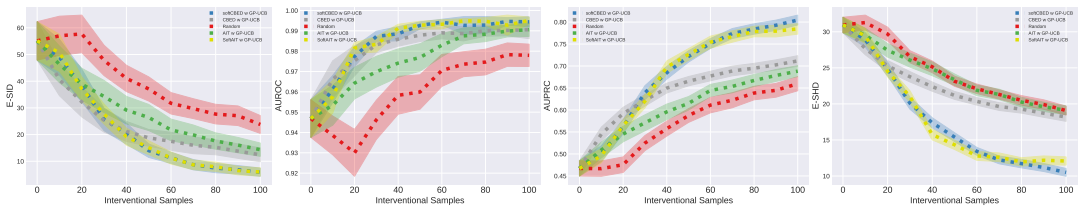
$\mathbb{E}$**-SID**: As the SHD is agnostic to the notion of intervention, Peters and Bühlmann ((2015)) proposed the *expected structural interventional distance* ($\mathbb{E}$**-SID**) which quantifies the differences between graphs with respect to the causal inference statements and interventional distributions.

AUROC: The *area under the receiver operating characteristic curve* of the binary classification task of predicting the presence/ absence of all edges.

AUPRC: The *area under the precision-recall curve* of the binary classification task of predicting the presence/ absence of all edges.

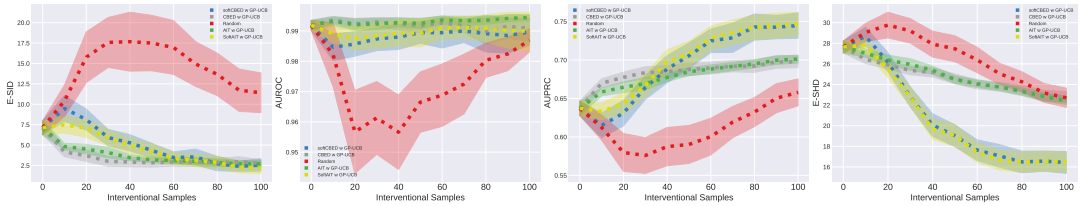
Complete list of Synthetic task results

Unless stated otherwise, for all the synthetic experiments we run 100 seeds, with standard error of the mean shaded.



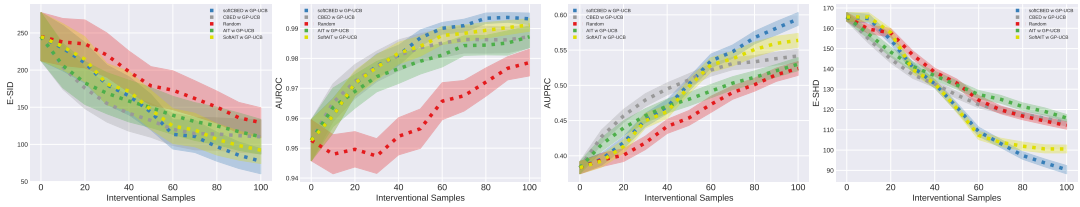
(a) ESID results (↓) **(b)** AUROC results (↑)**(c)** AUPRC results (↑) **(d)** ESHD results (↓)

Figure D.2: Results of ErdsRényi Erdős and Rényi ((1959)) linear SCMs with 20 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.



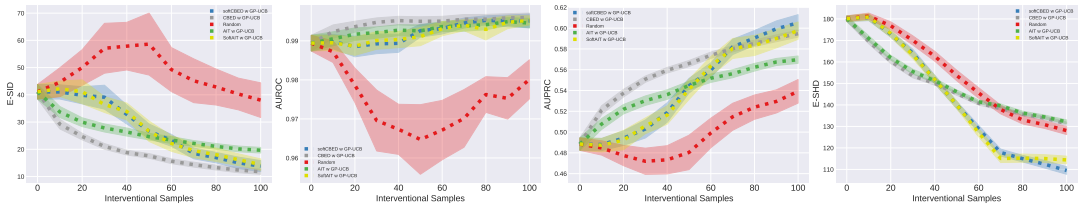
(a) ESID results (↓) **(b)** AUROC results (↑)**(c)** AUPRC results (↑) **(d)** ESHD results (↓)

Figure D.3: Results of scale-free linear SCMs with 20 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.



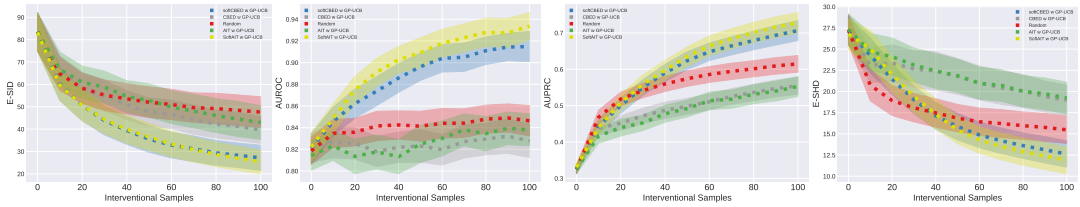
(a) ESID results (↓) **(b)** AUROC results (↑)**(c)** AUPRC results (↑) **(d)** ESHD results (↓)

Figure D.4: Results of ErdsRényi Erdős and Rényi ((1959)) linear SCMs with 50 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.



(a) ESID results (↓) **(b)** AUROC results (↑)**(c)** AUPRC results (↑) **(d)** ESHD results (↓)

Figure D.5: Results of scale-free linear SCMs with 50 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.

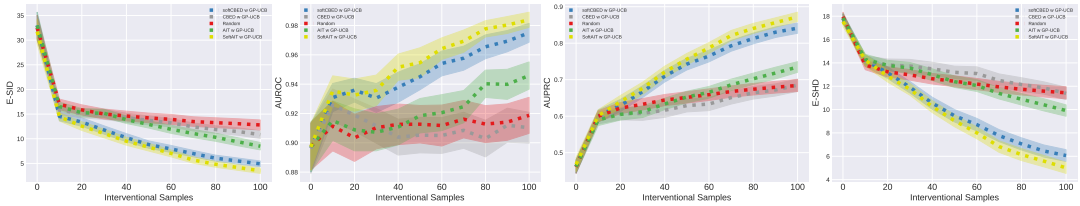


(a) ESID results (↓) **(b)** AUROC results (↑)**(c)** AUPRC results (↑) **(d)** ESHD results (↓)

Figure D.6: Results of ErdsRényi Erdős and Rényi ((1959)) nonlinear SCMs with 20 variables. Experiments were performed with DiBS as the underlying posterior model.

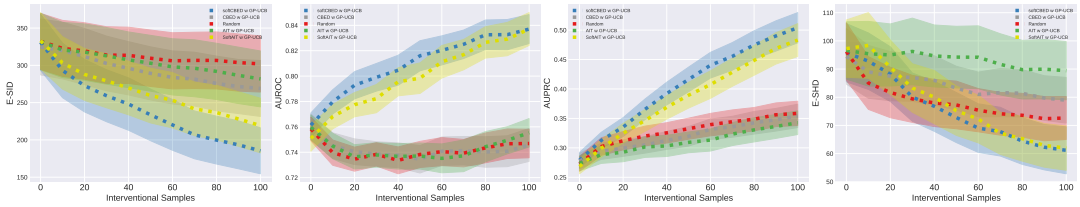
Code Dependencies

We are using the following dependencies.



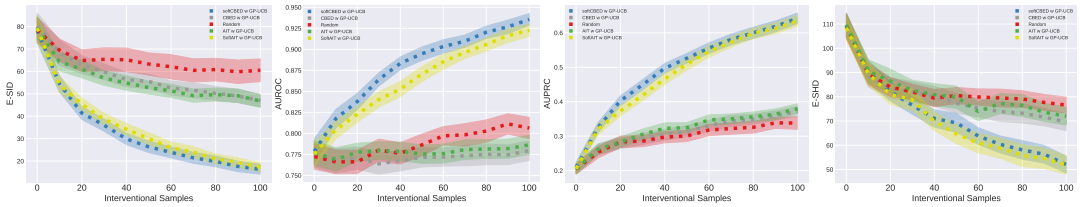
(a) ESID results (↓) **(b)** AUROC results (↑)**(c)** AUPRC results (↑) **(d)** ESHD results (↓)

Figure D.7: Results of scale-free nonlinear SCMs with 20 variables. Experiments were performed with DiBS as the underlying posterior model.



(a) ESID results (↓) **(b)** AUROC results (↑)**(c)** AUPRC results (↑) **(d)** ESHD results (↓)

Figure D.8: Results of ErdsRényi Erdős and Rényi ((1959)) nonlinear SCMs with 50 variables. Experiments were performed with DiBS as the underlying posterior model.



(a) ESID results (↓) **(b)** AUROC results (↑)**(c)** AUPRC results (↑) **(d)** ESHD results (↓)

Figure D.9: Results of scale-free nonlinear SCMs with 50 variables. Experiments were performed with DiBS as the underlying posterior model.

Table D.4: Set-up and dataset details for non-convex, non-linear regression prblem.

Name	URL	License
jaxlib/jax	https://jax.readthedocs.io/en/latest/	Apache
causaldag	https://github.com/FenTechSolutions/CausalDiscoveryToolbox	MIT
pytorch	https://github.com/pytorch/pytorch	BSD
xarray	https://github.com/pydata/xarray	Apache
cdt	https://github.com/FenTechSolutions/CausalDiscoveryToolbox	MIT
bayesian-optimization	https://github.com/fmfn/BayesianOptimization	MIT
pgmpy	https://github.com/pgmpy/pgmpy	MIT
igraph	https://github.com/igraph/igraph	GPL-2.0
numpy	https://github.com/numpy/numpy	BSD
SciPy	https://github.com/scipy/scipy	BSD
scikit-learn	https://github.com/scikit-learn/scikit-learn	BSD
networkx	https://github.com/networkx/networkx	BSD

Computation requirements

Table D.5: Total number of GPU hours (back-of-the-envelope estimation). Experiments are performed on an AMD EPYC 7662 64-core CPU and Tesla V100 GPU.

		Runtime per acq.	**Iterations**	**Seeds**	**Experiments**	**total (hours)**
	greedy ucb (CBED)	284.98	20	100	2	316.64
	greedy fixed (CBED)	32.56	20	100	2	36.17
D=50	soft ucb (CBED)	24.17	20	100	2	26.85
	soft fixed (CBED)	6.42	20	100	2	7.13
		6.42	20	100	2	7.13
	greedy ucb (AIT)	284.98	20	100	2	316.64
	soft ucb (AIT)	24.17	20	100	2	26.85
	greedy ucb (CBED)	113.992	20	100	2	126.65
	greedy fixed (CBED)	13.024	20	100	2	14.47
D=20	soft ucb (CBED)	9.668	20	100	2	10.74
	soft fixed (CBED)	2.568	20	100	2	2.85
	soft sampled (CBED)	2.568	20	100	2	2.85
	greedy ucb (AIT)	113.992	20	100	2	126.65
	soft ucb (AIT)	9.668	20	100	2	10.74
DREAM	soft fixed (CBED)	6.42	20	6	4	0.856
	soft fixed (AIT)	6.42	20	6	4	0.856
					sum	889.31

License

We summarize the licenses on table D.4.

E

Differentiable Causal Bayesian Experimental Design

Derivation of Importance Weighted Nested Monte Carlo Estimator

In this section, we derive the $\mathcal{U}_{\text{IWNMC}}$ (Eq. 5.5) estimator. We derive the estimator for a single design with an experiment denoted by ξ, parameters $\boldsymbol{\theta}$ and experimental outcome random variable $\mathbf{Y}$ and its instance $\mathbf{y}$. Since it is a static design, all the steps of the derivation hold if we replace ξ with $\xi_{1:B}$, $\mathbf{Y}$ with $\mathbf{Y_{1:B}}$ and $\mathbf{y}$ with $\mathbf{y_{1:B}}$. We begin from the variational NMC (VNMC) estimator, introduced by Foster et al. ((2019))

$$\mathcal{I}(\mathbf{Y};\boldsymbol{\Theta} \mid \xi) \leq \mathcal{U}_{\text{VNMC}}(\xi) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_0|h_{t-1}) \\ p(\mathbf{y}|\boldsymbol{\theta}_0,\xi) \\ q(\boldsymbol{\theta}_{1:L}|h_{t-1},\mathbf{y})}} \left[\log \frac{p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_0)}{\frac{1}{L}\sum_{\ell=1}^{L} \frac{p(\mathbf{y}|\xi,\boldsymbol{\theta}_\ell)p(\boldsymbol{\theta}_\ell|h_{t-1})}{q(\boldsymbol{\theta}_{1:L}|h_{t-1},\mathbf{y})}} \right]. \tag{E.1}$$

This can be rewritten as

$$\mathcal{U}_{\text{VNMC}}(\xi) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_0|h_{t-1}) \\ p(\mathbf{y}|\boldsymbol{\theta}_0,\xi) \\ q(\boldsymbol{\theta}_{1:L}|h_{t-1},\mathbf{y})}} \left[\log \frac{p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_0)}{\frac{1}{L}\sum_{\ell=1}^{L} \frac{p(\mathbf{y}|\xi,\boldsymbol{\theta}_\ell)p(\boldsymbol{\theta}_\ell)p(h_{t-1}|\boldsymbol{\theta}_\ell)}{q(\boldsymbol{\theta}_{1:L}|h_{t-1},\mathbf{y})}} + \log p(h_{t-1}) \right] \tag{E.2}$$

and Foster et al. ((2019)) observed that $\log p(h_{t-1})$ is a constant that does not depend on ξ and so can be safely neglected when optimizing over designs. If we take the original prior $p(\boldsymbol{\theta}_\ell)$ as our proposal distribution q, then we arrive at

$$\mathcal{U}_{\text{VNMC-prior}}(\xi) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_0|h_{t-1})p(\boldsymbol{\theta}_{1:L}) \\ p(\mathbf{y}|\boldsymbol{\theta}_0,\xi)}} \left[\log \frac{p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_0)}{\frac{1}{L}\sum_{\ell=1}^{L} p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_\ell)p(h_{t-1} \mid \boldsymbol{\theta}_\ell)} \right] + C \tag{E.3}$$

where $C = \log p(h_{t-1})$. This allows us to sample contrastive samples from any distribution, but does not account for $\boldsymbol{\theta}_0$. If we were to sample $\boldsymbol{\theta}_0$ from $p(\boldsymbol{\theta}_0)$, we can correct using an importance weight

$$\mathcal{U}_{\text{VNMC-prior}}(\xi) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_{0:L}) \\ p(\mathbf{y}|\boldsymbol{\theta}_0,\xi)}} \left[\frac{p(\boldsymbol{\theta}_0 \mid h_{t-1})}{p(\boldsymbol{\theta}_0)} \log \frac{p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_0)}{\frac{1}{L}\sum_{\ell=1}^{L} p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_\ell) p(h_{t-1} \mid \boldsymbol{\theta}_\ell)} \right] + C, \quad \text{(E.4)}$$

but unfortunately, this relies on knowing the density of the posterior or using the fact that $p(\boldsymbol{\theta}_0 \mid h_{t-1})/p(\boldsymbol{\theta}_0) = p(h_{t-1} \mid \boldsymbol{\theta}_0)/p(h_{t-1})$, knowing the marginal likelihood of the data h_{t-1}. Neither of these is usually tractable. Instead, we can use a self-normalized importance sampling approach, which amounts to estimating $p(h_{t-1})$ by a sum over $\boldsymbol{\theta}_{0:L}$, giving the *approximation* IWNMC:

$$\mathcal{U}_{\text{IWNMC}}(\xi) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_{0:L}) \\ p(\mathbf{y}|\boldsymbol{\theta}_0,\xi)}} \left[\frac{p(h_{t-1} \mid \boldsymbol{\theta}_0)}{\frac{1}{L+1}\sum_{k=0}^{L} p(h_{t-1} \mid \boldsymbol{\theta}_k)} \log \frac{p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_0)}{\frac{1}{L}\sum_{\ell=1}^{L} p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_\ell) p(h_{t-1} \mid \boldsymbol{\theta}_\ell)} \right] + C. \quad \text{(E.5)}$$

The form that is given in (5.5) is obtained by first relabelling the $\boldsymbol{\theta}$ samples to start from 1

$$\mathcal{U}_{\text{IWNMC}}(\xi) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_{1:L}) \\ p(\mathbf{y}|\boldsymbol{\theta}_1,\xi)}} \left[\frac{p(h_{t-1} \mid \boldsymbol{\theta}_1)}{\frac{1}{L}\sum_{k=1}^{L} p(h_{t-1} \mid \boldsymbol{\theta}_k)} \log \frac{p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_1)}{\frac{1}{L-1}\sum_{\ell=2}^{L} p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_\ell) p(h_{t-1} \mid \boldsymbol{\theta}_\ell)} \right] + C, \quad \text{(E.6)}$$

noting that the role of $\boldsymbol{\theta}_1$ is arbitrary and can be replaced by any $m \in \{1, \dots, L\}$

$$\mathcal{U}_{\text{IWNMC}}(\xi) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_{1:L}) \\ p(\mathbf{y}|\boldsymbol{\theta}_m,\xi)}} \left[\frac{p(h_{t-1} \mid \boldsymbol{\theta}_m)}{\frac{1}{L}\sum_{k=1}^{L} p(h_{t-1} \mid \boldsymbol{\theta}_k)} \log \frac{p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_m)}{\frac{1}{L-1}\sum_{\ell\neq m}^{L} p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_\ell) p(h_{t-1} \mid \boldsymbol{\theta}_\ell)} \right] + C, \quad \text{(E.7)}$$

and finally taking the mean over m, noting that this does not change the expected value due to linearity

$$\mathcal{U}_{\text{IWNMC}}(\xi) = \mathop{\mathbb{E}}_{\substack{p(\boldsymbol{\theta}_{1:L}) \\ p(\mathbf{y}|\boldsymbol{\theta}_m,\xi)}} \left[\sum_{m=1}^{L} \frac{p(h_{t-1} \mid \boldsymbol{\theta}_m)}{\sum_{k=1}^{L} p(h_{t-1} \mid \boldsymbol{\theta}_k)} \log \frac{p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_m)}{\frac{1}{L-1}\sum_{\ell\neq m}^{L} p(\mathbf{y} \mid \xi, \boldsymbol{\theta}_\ell) p(h_{t-1} \mid \boldsymbol{\theta}_\ell)} \right] + C. \quad \text{(E.8)}$$

We finally drop the constant C as it is independent of ξ and take

$$\omega_m = \frac{p(h_{t-1} \mid \boldsymbol{\theta}_m)}{\sum_{k=1}^{L} p(h_{t-1} \mid \boldsymbol{\theta}_k)}. \quad \text{(E.9)}$$

Expected Information Gain for 6 nodes and batch size 2

Metrics

.

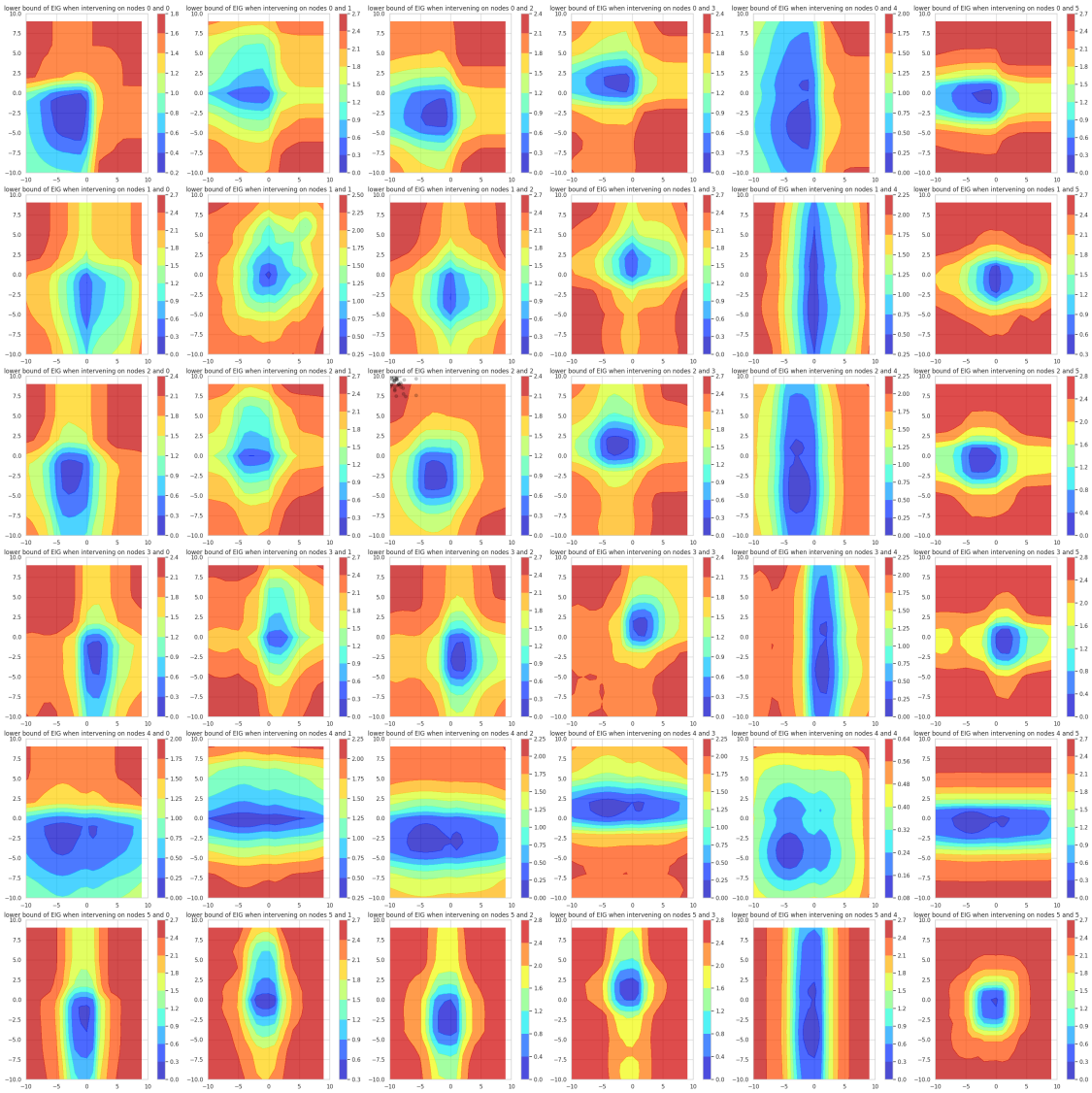


Figure E.1: Here we visualize the Expected Information Gain of batch size two, on two nodes over different interventional values of the range $[-10, 10]$.

$\mathbb{E}$**-SHD**: Defined as the *expected structural hamming distance* between samples from the posterior model over graphs and the true graph $\mathbb{E}\text{-SHD} := \mathbb{E}_{\mathbf{g}\sim p(\mathcal{G}|\mathcal{D})}\Big[\text{SHD}(\mathbf{g}, \tilde{\mathbf{g}})\Big]$

Expected edges F1: The expected F1 score of the binary classification task of predicting the presence/ absence of all edges. The expectation is taken over multiple posterior samples.

i-MMD: Interventional MMD is defined as MMD distance Gretton et al. ((2012)) between the true interventional distribution and the interventional distribution induced by θ and $\mathbf{g}$ (posterior sample). We take an expectation over different

posterior samples, interventional targets and interventional values. For the kernel choice, we use the median heuristic as described in Gretton et al. ((2012)).

DAG Bootstrap

The DAG bootstrap bootstraps observations and interventions to infer a different causal structure per bootstrap. We used GIES as the causal inference algorithm because of the adaptation of GES on interventional data as well. In our experiments, we used the pcalg R implementation `https://github.com/cran/pcalg/blob/master/R/gies.R` to discover 100 graphs. Each graph can be seen as a posterior sample from $p(\mathbf{G} \mid h_{t-1})$. For each of the sampled graphs G_i we compute the appropriate θ_{MLE} under linear Gaussian assumption for the conditional distributions.

Importance Weighted Nested Monte Carlo Full Results

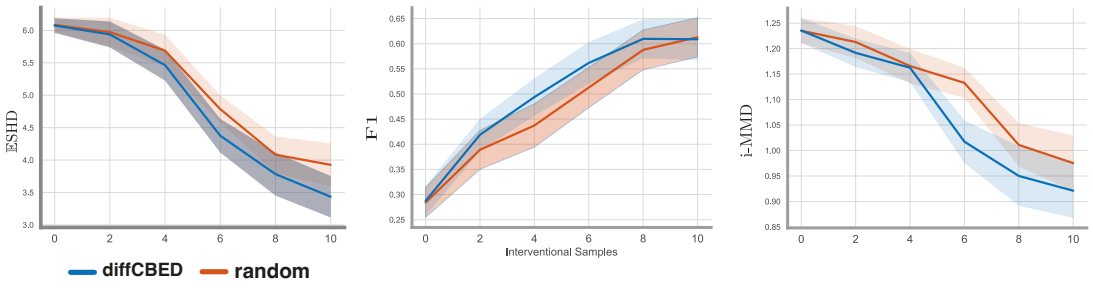


Figure E.2: Multi target-state design setting results for ErdsRényi Erdős and Rényi ((1959)) graphs with $d = 5$ variables. Each experiment was run with 60 random seeds (shaded area represents 95% CIs)

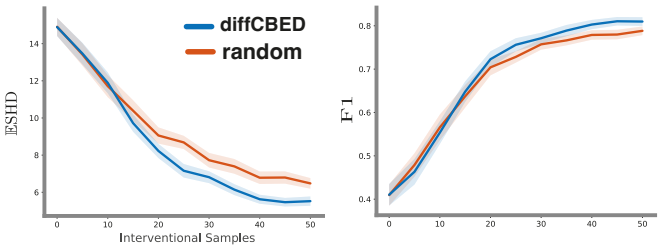


Figure E.3: Single target-state design setting results for ErdsRényi Erdős and Rényi ((1959)) graphs with $d = 10$ variables. Each experiment was run with 60 random seeds (shaded area represents 95% CIs)

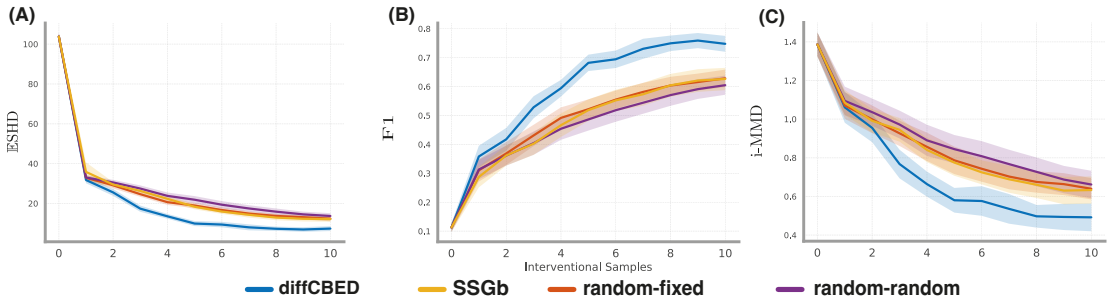


Figure E.4: Multi target-state design setting results for ErdsRényi Erdős and Rényi ((1959)) graphs with $d = 50$ variables. Each experiment was run with 30 random seeds (the shaded area represents 95% CIs). We observe that for batch size 1, the difference between the methods becomes more significant.

20 nodes, unconstrained ($q \leq 20$), batch size $B = 1$:

Datasets and Experiment Details

Synthetic Graphs Experiments

In the synthetic data experiments, we focus on Erds-Rényi graph model. We used `networkx`[1] and method `fast_gnp_random_graph` ((Batagelj and Brandes, 2005)) to generate graphs based on the Erds-Rényi model. We set the expected number of edges per vertex to 1.

Table E.1: Comparison of different BOED for Causal Discovery methods based on their design space assumptions.

	Design Space Assumptions				
	Node Acquisition (Single Target)	Value Acquisition (Single Target)	Node Acquisition (Multi-target)	Value Acquisition (Multi-target)	Batch Acquisition
Murphy ((2001))	✓				
Tong and Koller ((2001))	✓				
Cho et al. ((2016))	✓				
Agrawal et al. ((2019))	✓				✓
Toth et al. ((2022))	✓	✓			
Tigas et al. ((2022))	✓	✓			✓
Sussex et al. ((2021))	✓		✓		✓
Ours	✓	✓	✓	✓	✓

Table summarizing prior work

Optimizer Settings

Table E.2: Table indicating the hyperparameters and optimizer settings for different experimental results.

	Optimization settings			
	Single Target NMC	Multi-Target NMC	Multi-Target IWNMC with prior	Multi-Target IWNMC with proposal
L	30	30	1000	60
Number of outer DAGs N_o	30	30	1000	60
Batch Size	5	2	2	2
Relaxation temperature	$5 \rightarrow .5$	$5 \rightarrow .5$	0.1	$5 \rightarrow .5$
Optimizer	Adam	Adam	Adam	Adam
Learning rate of optimizer	0.1	0.1	0.01	0.1
Number of starting samples (observational)	60	60	2	800
Number of batches	10	10	5	1
Number of DAG Bootstraps	30	30	-	60
Number of training steps per batch	100	100	100	100

[1] https://networkx.org/documentation/networkx-1.10/reference/generated/networkx.generators.random_graphs.fast_gnp_random_graph.html