
Functional Graphical Models: Structure Enables Offline Data-Driven Optimization

Jakub Grudzien Kuba
BAIR, UC Berkeley

Masatoshi Uehara
Genentech

Pieter Abbeel
BAIR, UC Berkeley

Sergey Levine
BAIR, UC Berkeley

Abstract

While machine learning models are typically trained to solve prediction problems, we might often want to use them for optimization problems. For example, given a dataset of proteins and their corresponding fluorescence levels, we might want to optimize for a new protein with the highest possible fluorescence. This kind of data-driven optimization (DDO) presents a range of challenges beyond those in standard prediction problems, since we need models that successfully predict the performance of new designs that are better than the best designs seen in the training set. It is not clear theoretically when existing approaches can even perform better than the naïve approach that simply selects the best design in the dataset. In this paper, we study how structure can enable sample-efficient data-driven optimization. To formalize the notion of structure, we introduce functional graphical models (FGMs) and show theoretically how they can provide for principled data-driven optimization by decomposing the original high-dimensional optimization problem into smaller sub-problems. This allows us to derive much more practical regret bounds for DDO, and the result implies that DDO with FGMs can achieve nearly optimal designs in situations where naïve approaches fail due to insufficient coverage of the offline data. We further present a data-driven optimization algorithm that infers the FGM structure itself, either over the original input variables or a latent variable representation of the inputs.

1 INTRODUCTION

Machine learning models are typically trained for prediction, but we often want to use such models to solve *optimization* or *decision-making* problems. Imagine we would like to use a dataset of fluorescent proteins and their fluorescence levels to design a new protein with the highest possible fluorescence [3, 34]. Similarly, when provided with examples of hardware accelerators and their performance, an engineer might want to infer a more performant hardware accelerator design [15, 21]. A direct machine learning approach to such problems would be to train a surrogate model that predicts the performance for a given design, and then find its maximizers with some optimization method [35]. In this work, we focus on the challenging yet practical scenario where we can use offline training data to train a surrogate model, but cannot collect additional samples online. This offline data-driven optimization (DDO) approach is often applied in model-based optimization [12, 9, 35, MBO] and offline reinforcement learning [23, 8, RL].

The primary challenge in DDO is insufficient data coverage of optimal examples, which causes the distribution shift in the optimization process: when we optimize the design (*i.e.*, the input to the model), we produce a design very different from the training distribution. Predictions for such designs are likely to be inaccurate, and an optimizer might exploit the model to generate designs that result in the largest (optimistic) errors. Consequently, recent works in MBO and offline RL have focused on incorporating penalties for exploiting data outside of offline data when constructing a surrogate model [22, 35]. However, even with these conservative approaches, it remains unclear whether we can learn the optimal design when the offline training data fails to cover it. The question in this context is: *What enables DDO to be feasible in this challenging offline scenario?*

We make the observation that sample-efficient DDO can be enabled by leveraging structure. Without any structural bias for the surrogate model, even with the aforementioned conservative methods, a naïve ap-

proach in the worst case reduces to choosing the best design observed in the offline data, as we will formalize in Section 3. As the design space becomes large, the probability of an optimal design being actually present in the offline data generally becomes extremely low.

In our work, we introduce the framework of *Functional Graphical Models (FGMs)* as a structure that facilitates sample-efficient DDO. FGMs are designed to describe function’s independence properties, allowing us to decompose every function into subfunctions over partially intersecting cliques. Importantly, FGMs provide an effective way to introduce structural bias for DDO by automatically decomposing the whole optimization problem into smaller subproblems. We demonstrate that DDO with FGMs can mitigate distribution shift, as it only requires the optimal design to be covered by the offline data in a more lenient manner, compared to DDO without any structural bias. Specifically, FGMs enable the learning of the optimal design as long as the variables in each *clique* of the FGM take on optimal values for some data point—it is not required to see any single point with optimal values for *all* cliques jointly. As a result, we can formally demonstrate that DDO with FGMs surpasses the naïve approach.

Our contributions can be summarized as follows. First, we introduce FGMs and show how they give rise to a natural decomposition of the function of interest over its cliques. Second, we demonstrate that FGMs can reduce regret in data-driven optimization significantly. In particular, our results imply that we can learn a near-optimal design as long as it is covered by the offline data within individual cliques of the FGM, rather than the entire space. Lastly, we propose a practical data-driven optimization algorithm that can discover FGMs from the offline data, under Gaussian assumptions, or from a learned latent space in a more general setting. We validate the effectiveness of our algorithm in combination with MBO through numerical experiments that demonstrate the benefit of the discovered FGMs, especially for high-dimensional functions with many cliques.

2 RELATED WORK

Data-driven optimization has connections to several fields, including Bayesian optimization and model-based optimization, as summarized below.

Bayesian Optimization. The idea of using data and machine learning tools for design and optimization has been one of the primary motivations of *Bayesian optimization* (BO) [2, 31]. In BO, one establishes a *prior* belief of the considered function and updates it based on the given dataset, as well as queries for func-

tion evaluations that balance maximization and exploration of the function. However, such an *online* approach is not applicable to problems where additional queries cannot be made or are very cost- and time-consuming (*e.g.*, drug design) that would ideally be solved fully *offline*. Nevertheless, we consider our work to be of interest to the BO community, where it is common to *assume* a decomposition of the objective over a *dependency graph*, which simplifies the problem and enables solving it with fewer queries [14, 30, 11, 39]. In Section 4, we existence of such a decomposition for every function under mild conditions, enabling tractable optimization with finite amounts of offline data.

Probabilistic Graphical Models. Learning FGMs involves uncovering the structure of functions that map inputs x to outputs y . Similarly, the challenge of discovering structural graphs has been extensively studied in the field of probabilistic graphical models [13, 33, 24, PGMs]. FGMs and PGMs differ because PGMs are built upon probabilistic independence, while FGMs are constructed based on the concept of *functional independence*, which we will introduce later in Section 4. In the literature on PGMs, the main focus is on uncovering statistical relationships between variables, whereas for the DDO problems that we study, the main goal of using FGMs is to enable efficient optimization with fewer samples, as we will formalize in Section 5.

Model-Based Optimization. There has been a growing interest in model-based optimization (MBO) and the specific instantiation of MBO from static data, or data-driven optimization (DDO). Recent works have investigated computational design via such *offline* MBO for chemical molecules, proteins, hardware accelerators, and more [9, 3, 20, 21]. MBO and DDO are often used interchangeably, and we employ the term DDO to refer specifically to optimization from static *offline* data. Most of these prior methods adopt a “conservative” (*i.e.*, pessimistic) approach to DDO, employing regularizers to prevent the generation of out-of-distribution designs [35, 27]. Combined with appropriate inductive bias, these methods found application in practical engineering problems, such as hardware accelerator (microchip) design [21]. In our paper, we instead investigate how discovering functional independence structure via our FGM framework can enable efficient optimization from data, and provide to our knowledge the first theoretical analysis of sample efficiency for DDO with such independence properties.

3 PRELIMINARIES

In DDO, the algorithm receives an offline dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$, where the input $\mathbf{x} \in \mathcal{X}$ is drawn from a data distribution $p \in \Delta(\mathcal{X})$, and $y = f(\mathbf{x}) \in \mathbb{R}$ is

the corresponding evaluation of an unknown (black-box) function f at \mathbf{x} . Our goal is to find an input (*i.e.*, design) \mathbf{x} that takes a high value in terms of $f(\mathbf{x})$. In the ideal case, we may want to find an exact $\mathbf{x}^* \triangleq \arg \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$. In our analysis, we study a generalization of this problem, where DDO learns a stochastic distribution $\pi \in \Delta(\mathcal{X})$ that maximizes $J(\pi)$, where $J(\pi)$ is the value defined by $J(\pi) = \mathbb{E}_{\mathbf{x} \sim \pi}[f(\mathbf{x})]$.

In DDO (*e.g.*, MBO), conventional methods typically involve the following steps: (a) obtaining a surrogate model $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ using function approximation techniques like deep neural networks, and (b) optimizing the surrogate model with respect to $\mathbf{x} \in \mathcal{X}$. This is formalized in the following procedure:

$$\hat{f} = \arg \min_{\hat{f} \in \mathcal{F}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\{\hat{f}(\mathbf{x}) - y\}^2], \quad (\text{Step (a)})$$

$$\hat{\pi} = \arg \max_{\pi \in \Pi} \mathbb{E}_{\mathbf{x} \sim \pi} [\hat{f}(\mathbf{x})], \quad (\text{Step (b)})$$

where Π consists of probability distributions over \mathcal{X} , and \mathcal{F} is a regression class that consists of functions mapping \mathcal{X} to \mathbb{R} . When Π is a fully expressive class¹, *i.e.*, $\Pi = \Delta(\mathcal{X})$, Step (b) reduces to

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathcal{X}} \hat{f}(\mathbf{x}).$$

The primary practical challenge in the fully offline DDO setting lies in dealing with insufficient data coverage. To see this, consider the naïve approach: we can set a model $\hat{f}(\mathbf{x})$ to match the observed $f(\mathbf{x})$ values and assign random values to examples outside of the offline data. This approach amounts to constructing a surrogate model \hat{f} in the initial Step (a) without relying on any structural bias. In the discrete case, it is equivalent to setting $\mathcal{F} = \{\mathbf{x} \mapsto \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}) \mid \boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{X}|}\}$, where $\boldsymbol{\phi}(\mathbf{x})$ is a $|\mathcal{X}|$ -dimensional one-hot encoding vector. Maximizing such a model corresponds to choosing \mathbf{x} for which $\theta_{\mathbf{x}}$ is highest. However, if the input space \mathcal{X} is not fully covered by the data, such an optimization process may end up outputting a novel design \mathbf{x} whose ground-truth value $f(\mathbf{x})$ is very low, but its random prediction $\hat{f}(\mathbf{x})$ happens to be high, thus adversarially exploiting the model’s errors. One may want to tackle this problem by, instead of assigning random values to unobserved designs, modeling them *pessimistically*—for example, by setting them to $\hat{f}(\mathbf{x}) = -\infty$. The resulting solution is:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathcal{X}} \hat{f}(\mathbf{x}) = \arg \max_{\mathbf{x} \in \mathcal{D}} \hat{f}(\mathbf{x}), \quad (1)$$

which implies that the performance of the output design is bounded by the performance of the examples

¹When \mathcal{X} is discrete, we can always use this fully expressive class. However, when \mathcal{X} is continuous, one must use a restricted distribution class, *e.g.*, $\Pi = \{\mathcal{N}(\mathbf{x}, \sigma^2); \mathbf{x} \in \mathcal{X}\}$ for small σ^2 .

in the offline data. Hence, in either case, the performance of DDO is expected to be poor when the data distribution $p(\mathbf{x})$ does not cover regions with high output values in terms of $f(\mathbf{x})$. This issue is particularly severe when the space \mathcal{X} is large, which tends to cause $p(\mathbf{x}^*)$ to be low.

To tackle this problem our work explores reasonable structural bias that can be incorporated into surrogate model classes \mathcal{F} , mitigates the problem of insufficient coverage, and can be learned from the offline data.

Notation. We write \mathcal{V} to denote the index set of input variables. If $\mathcal{X} \subseteq \mathbb{R}^d$, then $\mathcal{V} = [d] \triangleq \{1, \dots, d\}$. Thus, $\mathbf{x}_{\mathcal{V}}$ and \mathbf{x} are equivalent. For a subset $\mathcal{S} \subseteq \mathcal{V}$, we denote $\mathbf{x}_{\mathcal{S}}$ as the vector of variables with indexes in \mathcal{S} and $\mathcal{X}_{\mathcal{S}}$ as its domain. We denote the rest of the vector in \mathbf{x} by $\mathbf{x}_{-\mathcal{S}}$. Furthermore, $a \lesssim b$ stands for inequality up to a problem-dependent constant. The notation $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2 , and we use \mathcal{N} to denote $\mathcal{N}(0, I)$ for simplicity.

4 FUNCTIONAL GRAPHICAL MODELS

We present the concept of *functional independence*, which characterizes the structural relationship between inputs and outputs. We then build on this definition to introduce a graphical representation of functional independences between variables, which we refer to as *Functional Graphical Models (FGMs)*. Then, we will show how to apply these concepts to DDO.

4.1 Functional Independence

We define the concept of functional independence, which decomposes the structure of functions mapping from \mathcal{X} to real numbers. This concept can be used as a useful inductive bias when constructing surrogate functions for DDO.

Definition 1 (Functional Independence). *Let $A, B, S \subseteq \mathcal{V}$. We say that \mathbf{x}_A and \mathbf{x}_B are functionally independent given \mathbf{x}_S if, there exist functions $f_{-(B \setminus S)}$ and $f_{-(A \setminus S)}$, such that for every $\mathbf{x} \in \mathcal{X}$,*

$$f(\mathbf{x}) = f_{-(B \setminus S)}(\mathbf{x}_{\mathcal{V} \setminus (B \setminus S)}) + f_{-(A \setminus S)}(\mathbf{x}_{\mathcal{V} \setminus (A \setminus S)}).$$

If $S = \emptyset$ and there exist mutually disjoint supersets \underline{A} of A and \underline{B} of B such that $f_{-B}(\mathbf{x}_{-B}) = f_{\underline{A}}(\mathbf{x}_{\underline{A}})$, and $f_{-A}(\mathbf{x}_{-A}) = f_{\underline{B}}(\mathbf{x}_{\underline{B}})$ the independence is absolute.

Intuitively, functional independence of \mathbf{x}_A and \mathbf{x}_B given \mathbf{x}_S states that, once we fix the value of \mathbf{x}_S , the two sets of variables can be optimized independently. This is a desirable property for DDO because each input subspace is covered denser by the dataset if viewed individually rather than jointly with another subspace.

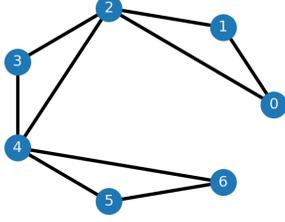


Figure 1: The clique set of this graph is given by $\{\{0, 1, 2\}, \{2, 3, 4\}, \{4, 5, 6\}\}$. Hence, if it is a FGM of $f(\mathbf{x})$, then $f(\mathbf{x}) = f_{0,1,2}(\mathbf{x}_{0,1,2}) + f_{2,3,4}(\mathbf{x}_{2,3,4}) + f_{4,5,6}(\mathbf{x}_{4,5,6})$.

Intuitively, this allows an optimization algorithm to decompose the original problem into two smaller ones with more effective data. We will formalize this intuition in Section 5.

Our first result offers alternative criteria for defining functional independence when \mathcal{X} is contained in an Euclidian space. These properties play a key role in discovering structure in unknown functions, as we will discuss in Section 6.

Lemma 1. *Suppose $f(\mathbf{x})$ is twice-continuously differentiable w.r.t. \mathbf{x} . Let $A, B, S \subseteq \mathcal{V}$. Then, the following statements are equivalent.*

1. \mathbf{x}_A and \mathbf{x}_B are functionally independent given \mathbf{x}_S .
2. For every $\mathbf{x} \in \mathcal{X}$,

$$\frac{\partial f}{\partial \mathbf{x}_{A|S}}(\mathbf{x}) = F_{-(B|S)}(\mathbf{x}_{\mathcal{V} \setminus (B|S)}),$$

for some function $F_{-(B|S)}(\mathbf{x}_{\mathcal{V} \setminus (B|S)})$ that does not take $\mathbf{x}_{(B|S)}$ as input.

3. For every $\mathbf{x} \in \mathcal{X}$,

$$\frac{\partial^2 f}{\partial \mathbf{x}_{A|S} \partial \mathbf{x}_{B|S}}(\mathbf{x}) = 0.$$

Detailed proofs are provided in Appendix B.

In the following subsection, we introduce a graphical representation of functional independence that enables encoding structures within high-dimensional functions and leads to a provably efficient approach to DDO.

4.2 FGMs

We first define functional graphical models (FGMs), a general framework for working with functional independence. An example is shown in Figure 1.

Definition 2 (Functional Graphical Model (FGM)). *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an FGM of a function $f(\mathbf{x})$ if, for any $i, j \in \mathcal{V}$ such that $i \neq j$, $(i, j) \notin \mathcal{E}$, we have that \mathbf{x}_i and \mathbf{x}_j are functionally independent given $\mathbf{x}_{\mathcal{V} \setminus \{i, j\}}$. FGM with the smallest set \mathcal{E} is referred to as the minimal FGM of f .*

Our construction of FGMs is analogous to that of probabilistic graphical models (PGMs), where the absence of an edge indicates probabilistic conditional independence between two random variables [13, 18]. As we prove in Appendix B.1, a result similar to the Hammersley-Clifford theorem in PGMs [5] holds for FGMs, decomposing arbitrary functions as follows.

Theorem 1 (Function Decomposition). *Assume that $\int_{\mathcal{X}} \exp f(\mathbf{x}) d\mathbf{x}$ exists and let \mathcal{G} be any FGM of $f(\mathbf{x})$ and \mathcal{C} be its set of maximal cliques (clique set). Then, there exist functions $\{f_C(\mathbf{x}_C), C \in \mathcal{C}\}$, such that*

$$f(\mathbf{x}) = \sum_{C \in \mathcal{C}} f_C(\mathbf{x}_C).$$

We provide an illustrative example of application of this theorem in Figure 1.

4.3 DDO via FGMs

Consider a DDO algorithm that incorporates FGMs by decomposing its surrogate model, following Theorem 1, over the maximal cliques of the FGM of f . Then, DDO can be instantiated as follows:

$$\hat{\pi}_{\text{FGM}} = \arg \max_{\pi \in \Pi} \mathbb{E}_{\mathbf{x} \sim \pi} [\hat{f}(\mathbf{x})], \quad \hat{f}(\mathbf{x}) = \sum_{C \in \mathcal{C}} \hat{f}_C(\mathbf{x}_C),$$

$$\{\hat{f}_C\}_{C \in \mathcal{C}} = \arg \min_{\{\bar{f}_C \in \mathcal{F}_C\}_{C \in \mathcal{C}}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} \left[\left(\sum_{C \in \mathcal{C}} \bar{f}_C(\mathbf{x}_C) - y \right)^2 \right],$$

where each \mathcal{F}_C is a function class that consists of maps from \mathcal{X}_C to \mathbb{R} .

Now, consider an illustrative simple case where \mathcal{X} is discrete. In this case, natural choices of Π and \mathcal{F}_C would be fully expressive classes: $\Pi = \Delta(\Pi)$, $\mathcal{F}_C = \{\mathbf{x}_C \mapsto \boldsymbol{\theta}_C^\top \boldsymbol{\phi}_C(\mathbf{x}_C) \mid \boldsymbol{\theta}_C \in \mathbb{R}^{|\mathcal{X}_C|}\}$ where $\boldsymbol{\phi}_C$ is a one-hot encoding vector over \mathcal{X}_C . Then, the abovementioned method is equivalent to choosing

$$\arg \max_{\bar{\mathbf{x}} \in \mathcal{X}} \{\boldsymbol{\phi}^\top(\bar{\mathbf{x}}) \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [\boldsymbol{\phi}(\mathbf{x}) \boldsymbol{\phi}(\mathbf{x})^\top]^{-1} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\boldsymbol{\phi}(\mathbf{x}) y]\},$$

$$\boldsymbol{\phi}(\mathbf{x}) = [\boldsymbol{\phi}(\mathbf{x}_1)^\top, \dots, \boldsymbol{\phi}(\mathbf{x}_C)^\top]^\top. \quad (2)$$

Importantly, this method differs from just choosing an optimal \mathbf{x} in the dataset \mathcal{D} , i.e., $\hat{\mathbf{x}}_{\text{sim}}$ in Eq. (1) which corresponds to a method without any inductive bias.

In practice, we must still estimate FGMs from the data and handle the continuous space \mathcal{X} . We will delve into the practical implementation of this in Section 6.

5 FGMs ENABLE SAMPLE-EFFICIENT DATA-DRIVEN OPTIMIZATION

In this section, we present our main result, demonstrating that functions represented by FGMs can be

optimized efficiently with much less stringent data coverage assumptions than those needed for standard data-driven optimization.²

5.1 REGRET GUARANTEES

First, we review the regret guarantee of the naïve approach in **Step (a)** and **Step (b)**. A standard regret guarantee in this setting [4] states that, with probability $1 - \delta$, the following holds:

$$\underbrace{J(\pi^*) - J(\hat{\pi})}_{\text{Regret against } \pi^*} \lesssim \underbrace{\max_{\pi \in \Pi} \max_{\mathbf{x} \in \mathcal{X}} \left| \frac{\bar{\pi}(\mathbf{x})}{p(\mathbf{x})} \right|}_{\text{Coverage (a)}} \times \underbrace{\sqrt{\frac{\log(|\mathcal{F}|/\delta)}{n}}}_{\text{Complexity (a')}} \quad (3)$$

where π^* is an optimal design among Π , i.e., $\pi^* = \arg \max_{\pi \in \Pi} J(\pi)$.

The coverage term (a) measures the discrepancy between a distribution in Π and a data distribution p . The term (a') signifies the size of the function class \mathcal{F} . Note this $|\mathcal{F}|$ could be easily replaced with the covering number when \mathcal{F} is infinite [36].

Our main result shows that DDO with FGMs enjoys a much more favorable regret guarantee:

Theorem 2 (Regret of DDO with FGMs). *Let \mathcal{C} be the set of maximal cliques of an FGM of $f(\mathbf{x})$ and $f(\mathbf{x}) = \sum_{C \in \mathcal{C}} f_C(\mathbf{x}_C)$ be its FGM decomposition. Under the following assumptions $\forall C \in \mathcal{C}$,*

(1) *the function approximator classes are centered*

$$\mathbb{E}_{\mathbf{x} \sim p} [\bar{f}_C(\mathbf{x}_C)] = 0, \forall \bar{f}_C \in \mathcal{F}_C,$$

(2) *the correlations between cliques are well-controlled*

$$\max_{\{\bar{f}_C \in \mathcal{F}_C\}} \text{Corr}[\bar{f}_{C_1}(\mathbf{x}_{C_1}), \bar{f}_{C_2}(\mathbf{x}_{C_2})] \leq \sigma,$$

(3) *the models are well-specified, sp that $f_C \in \mathcal{F}_C$, and*

(4) *$|\bar{f}_C(\mathbf{x}_C)| \leq 1$ for any $\mathbf{x}_C \in \mathcal{X}_C$, $\bar{f}_C \in \mathcal{F}_C$,*

the following result holds with probability $1 - \delta$

$$J(\pi^*) - J(\hat{\pi}_{\text{FGM}}) \lesssim \underbrace{\sqrt{\frac{1}{1 - \sigma}}}_{\text{Corr}(b'')} \underbrace{\max_{C \in \mathcal{C}} \max_{\pi \in \Pi} \max_{\mathbf{x}_C \in \mathcal{X}_C} \left| \frac{\bar{\pi}(\mathbf{x}_C)}{p(\mathbf{x}_C)} \right|}_{\text{Coverage (b)}} \underbrace{\sqrt{\frac{|\mathcal{C}| \sum_C \log(|\mathcal{F}_C|/\delta)}{n}}}_{\text{Complexity (b')}} \quad (4)$$

where $p(\mathbf{x}_C)$, $\bar{\pi}(\mathbf{x}_C)$ denote the marginal distributions of \mathbf{x}_C with respect to $p(\mathbf{x})$ and $\pi(\mathbf{x})$.

While the bound of the naïve approach in Eq. (3) implies that MBO without FGM structure necessitates the data distribution $p(\mathbf{x})$ to cover the optimal

²Note our theory in this section still holds when an observation y has a bounded measurement error ϵ , i.e., $y = f(\mathbf{x}) + \epsilon$.

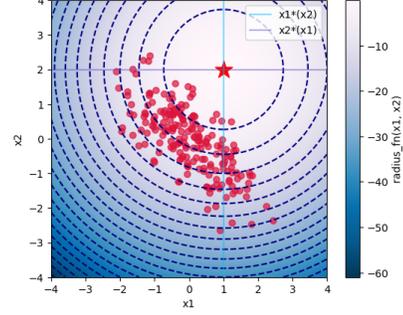


Figure 2: Consider a function $f(\mathbf{x}_1, \mathbf{x}_2) = -(\mathbf{x}_1 - 1)^2 - (\mathbf{x}_2 - 2)^2$. Clearly, the singleton cliques $\{\mathbf{x}_1\}$ and $\{\mathbf{x}_2\}$ are functionally independent, while the data coming from a correlated normal distribution are not statistically independent. While the dataset does not jointly cover the optimal solution $\mathbf{x}^* = (1, 2)$, it does cover individual components $\mathbf{x}_1^* = 1, \mathbf{x}_2^* = 2$, and thus a method that can learn the component functions can compose them into \mathbf{x}^* .

design itself, the FGM factorization only necessitates the data distribution to cover components within each *clique* in the optimal design. This is a significantly less stringent requirement, as the best design in the offline data might be much less effective than the best design achievable by combining favorable settings of the values in each of the cliques, as illustrated in Figure 2, and analyzed deeper below.

Coverage term. The term (b) corresponds to coverage that takes into account the FGM factorization. This term quantifies how a distribution $\pi \in \Pi$ differs from a data distribution p in terms of *marginals* with respect to \mathbf{x}_C . The following lemma demonstrates that term (b) is smaller than the coverage term (a) across the entire space \mathcal{X} :

Lemma 2 (Improvement of Coverage terms). *For any $\pi \in \Delta(\mathcal{X})$, we have*

$$\max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \leq \max_{\mathbf{x} \in \mathcal{X}} \frac{\pi(\mathbf{x})}{p(\mathbf{x})}.$$

Consider the following illustrative example: suppose the space \mathcal{X} is discrete, denoted by $\mathcal{X} = \mathcal{X}_1 \otimes \mathcal{X}_2 \otimes \dots \otimes \mathcal{X}_d$ where $\mathcal{X}_1 = \mathcal{X}_2 = \dots = \mathcal{X}_d$. Now, let's assume that $f(\mathbf{x}) = \sum_C f_C(\mathbf{x}_C)$, and $p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d)$ follows a jointly independent uniform distribution. Term (a) exhibits exponential growth with the dimension d :

$$\max_{\pi \in \Delta(\mathcal{X})} \max_{\mathbf{x} \in \mathcal{X}} \frac{\bar{\pi}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d)}{p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d)} = |\mathcal{X}_1|^d. \quad (4)$$

Therefore, the naïve data-driven optimization procedure suffers from a curse of dimensionality, requiring an exponentially large number of samples to obtain a

near-optimal design as the dimension increases. On the contrary, the term (b) remains reasonably small:

$$\max_{\bar{\pi} \in \Delta(\mathcal{X})} \max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\bar{\pi}(\mathbf{x}_C)}{p(\mathbf{x}_C)} \leq |\mathcal{X}_1|^{\max_C d_C} \quad (5)$$

where d_C represents the dimension of \mathbf{x}_C . Thus, the DDO with FGMs can effectively overcome the curse of dimensionality when $\max_C d_C$ is reasonably small.

Furthermore, in a more extreme example, illustrated in Figure 2, the coverage term (b) is bounded, but the coverage term (a) is infinite. This illustrates that DDO with FGMs can identify the optimal design, when DDO without FGMs cannot do so even with infinite samples. It's worth noting that in this scenario, a discerning reader might express concerns about the potential explosion of the additional term (b''), which could adversely affect the regret of DDO with FGMs. However, as we will discuss in Section 5.2, this term (b'') is moderately constrained.

Statistical complexity term. We can anticipate that (b') is generally much smaller than (a') because the function class for the surrogate model has more structure. Let's examine a specific scenario, where $\mathcal{X}_1 = \dots = \mathcal{X}_d$ and each \mathcal{X}_i is discrete. Suppose that \mathcal{F} is a linear combination of all polynomial basis functions over $[x_1, \dots, x_d]$, i.e., the fully expressive class. In this case, a standard argument based on covering [36] shows that the term (a') would be on the order of $O(\sqrt{|\mathcal{X}_1|^d/n})$, because the number of parameters in the model is $|\mathcal{X}| = |\mathcal{X}_1|^d$. Now, let's consider the term (b'). When we set each \mathcal{F}_C to be a linear combination of all polynomial basis functions over the entire \mathcal{X}_C , its order becomes $O(\sqrt{|\mathcal{C}||\mathcal{X}_1|^{\max_C d_C}/n})$, because the number of parameters in the model is bounded from above by $|\mathcal{C}||\mathcal{X}_1|^{\max_C d_C}$. Again, this implies that DDO with FGMs can effectively overcome the curse of dimensionality when $\max_C d_C$ is reasonably small.

Summary. Theorem 2 reveals that the original data-driven optimization problem naturally breaks down into subproblems for each clique $C \in \mathcal{C}$. More specifically, this result shows that, while the naïve approach requires the data distribution to cover the optimal design, with FGMs, it is only necessary for the data distribution to cover the optimal values of each clique separately. With non-trivial FGMs, as we see in Eq. (4) and Eq. (5), the difference in the number of samples needed to obtain a near-optimal design between DDO with FGMs and naïve DDO (i.e., DDO without FGMs) can be exponential³. Furthermore, although there could be a potential concern related to the additional term (b''), in certain instances, DDO

with FGMs has the capability to learn the optimal design, whereas the naïve DDO approach cannot achieve this even with an infinite amount of data. The intuition for this is that naïve DDO requires the samples from the data distribution to randomly set *all* of the variables to near-optimal values in at least some of the training points in the offline data, whereas DDO with FGMs only requires some training points for each clique to have near-optimal values, and does not require any single training point to have near-optimal values for *all* cliques. In the naïve case, this can lead to catastrophic sample complexity, essentially reducing the method to selecting the best point in the offline data (i.e., the naïve approach in Eq. 1), whereas with FGMs and relatively small cliques, it is possible to recombine the best values for each clique and find good designs potentially with exponentially fewer samples.

Note that the literature on additive models [10] has previously explored their role in reducing statistical complexity. In contrast, we characterize the coverage term in the context of DDO. It's important to highlight that simply reducing statistical complexity polynomially (i.e., improving the term (a')) is insufficient for overcoming the curse of dimensionality, because the term (a) might still grow exponentially.

5.2 Assumptions on DDO with FGMs

Theorem 2 shows that DDO with FGMs enjoys favorable regret guarantees under a set of assumptions of the DDO problem. In this section, we'll discuss these assumptions in more detail to provide the reader with intuition about our main result. The first and last assumptions are primarily technical. The first one can always be satisfied with negligible error by substituting $f_C(\mathbf{x}^{(i)})$ and $y^{(i)}$ with $f_C(\mathbf{x}^{(i)}) - \mathbb{E}_{\mathbf{x}_C \sim \mathcal{D}}[f(\mathbf{x}_C)]$ and $y^{(i)} - \mathbb{E}_{\mathbf{y} \sim \mathcal{D}}[y]$. The third assumption is standard, and we can easily account for potential misspecification errors in the theorem. The second assumption is substantial. When $\sigma = 1$, our guarantee is essentially void. In essence, this assumption implies that each clique exhibits mild independence, automatically breaking down the entire problem into subproblems. Technically, this enables us to translate the error of $\hat{f}(\mathbf{x}) = \sum_C \hat{f}_C(\mathbf{x}_C)$ for the entire graph into the error of \hat{f}_C for each individual clique.

Readers might have concerns about whether the second assumption could potentially lead to issues, particularly in cases where cliques overlap. In this context, it's important to note that no two cliques overlap completely, nor is one entirely contained within another, since they are elements of the set of maximal cliques of the FGM. Furthermore, this assumption can be significantly relaxed. In the second assumption,

³Note the term (b'') is still 1 in this case. Hence, we can ignore the term (b'') for the comparison.

while the maximum is taken over the entire function class \mathcal{F}_C , we can replace it with a much smaller set consistent with $f(\mathbf{x})$, as we demonstrate in Appendix C. Secondly, in many cases, we can still observe huge improvements in terms of regret, especially when the FGM cliques are much smaller than the whole graph. We illustrate this empirically in Section 7.

5.3 Relation with Pessimism

Most of the common recent approaches to offline DDO can be characterized by *pessimism*, displayed in algorithms through, for example, additional losses that penalize exploring out-of-distribution designs [37, 22, 35]. To facilitate the comparison between the naïve procedure without FGMs and the one with FGMs, in our work, we did not include pessimistic penalties for searching outside of offline data in the algorithms. When we incorporate pessimism into the algorithms, we can formally show that it alleviates distribution shift in that we can obtain a regret bound where $\max_{\pi \in \Pi}$ in the coverage term is replaced with just an optimal design π^* [28, 38]. In this sense, pessimism alleviates the distributional shift problem. However, in the instance we show, even if we replace $\max_{\pi \in \Pi}$ with an optimal design π^* , the coverage term in Eq. (4) for the procedure without FGMs is still $O(|\mathcal{X}_1|^d)$, and the coverage term in Eq. (5) for the procedure with FGMs is still $O(|\mathcal{X}_1|^{\max_C d_C})$. Consequently, pessimism itself cannot resolve distribution shift when the optimal design is not well-covered by the data distribution $p(\mathbf{x})$. On the contrary, FGMs relax the requirement for the data coverage of the whole design space \mathcal{X} into a much milder one of coverage of subspaces induced by individual cliques.

5.4 Relation with Offline RL

The decomposition provided by FGMs inspires thoughts about their connection to offline RL. As we alluded to before, offline RL can be formulated as an instance of DDO with a known FGM. To see this, let us denote the horizon of a given RL problem by H . The goal is to maximize the expected value of the return $R(s_1, a_1, \dots, s_H, a_H)$, which decomposes as $\sum_{t=1}^H r(s_t, a_t)$. Hence, the FGM of the return consists of edges (s_t, a_t) , which also form its cliques. In this specific setting, however, we cannot control all variables directly, since an RL agent can only optimize its actions, which in turn affect the stochastic states. In this paper, we focus our attention on problems where all variables are optimizable, but we see the setting of partially-controllable variables as an exciting avenue for future work.

6 GRAPH DISCOVERY

So far, we’ve shown how functions with known FGMs can be optimized effectively and under much milder coverage conditions than naïve DDO. However, in many cases, we do not know the FGM for a given function. Next, we’ll discuss how to construct FGMs only from offline data practically, provided that the data satisfies a Gaussian distribution assumption:

Assumption 1. *The inputs in the offline data follow the standard Gaussian distribution, $\mathbf{x} \sim \mathcal{N}(0, I)$.*

This assumption may appear strong. However, whitening to remove correlations is a common procedure in machine learning, and in cases where the data distribution has higher order moments, we can use more advanced representation learning methods, such as variational auto-encoders [17, 29, 6, VAE]. Indeed, these methods are designed specifically to acquire latent representations of the data that follow the standard Gaussian distribution.

To derive our graph discovery method, we start with the following lemma, known as the *second-order Stein’s identity* [32, 7]:

Lemma 3. *Let $\bar{\mathbf{x}} \in \mathbb{R}^{d_x}$ and $i \neq j \in [d_x]$. Then*

$$\frac{\partial^2}{\partial \bar{x}_j \partial \bar{x}_i} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}} [f(\bar{\mathbf{x}} + \mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim \mathcal{N}} [x_i x_j f(\bar{\mathbf{x}} + \mathbf{x})].$$

Below, we demonstrate how this lemma can be used to discover FGMs from data in combination with the third criterion (in Lemma 1) for functional independence. Suppose that x_i and x_j are not linked in a FGM of f , such that $(i, j) \notin \mathcal{E}$. By that criterion,

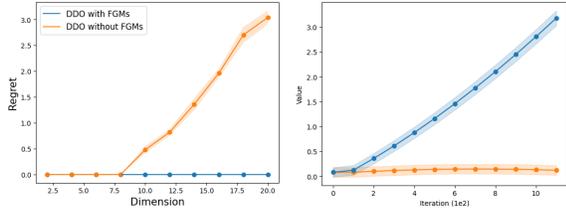
$$\begin{aligned} & \frac{\partial^2}{\partial \bar{x}_j \partial \bar{x}_i} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}} [f(\bar{\mathbf{x}} + \mathbf{x})] \\ &= \mathbb{E}_{\mathbf{x} \sim \mathcal{N}} \left[\frac{\partial^2 f}{\partial \bar{x}_j \partial \bar{x}_i} (\bar{\mathbf{x}} + \mathbf{x}) \right] = \mathbb{E}_{\mathbf{x} \sim \mathcal{N}} [0] = 0. \end{aligned}$$

Letting $\bar{\mathbf{x}} = \mathbf{0}_{d_x}$, for $(i, j) \notin \mathcal{E}$, we have $\mathbb{E}_{\mathbf{x} \sim \mathcal{N}} [x_i x_j f(\mathbf{x})] = 0$.⁴ Hence, to reconstruct edges in the unknown FGM of f , one can take M samples $\{\mathbf{x}^{(k)}, y^{(k)}\}_{k=1}^M$ and estimate

$$\hat{\mathbf{H}}_{ij} = \frac{1}{M} \sum_{k=1}^M x_i^{(k)} x_j^{(k)} y^{(k)}, \quad (6)$$

and infer that an edge is absent if $\hat{\mathbf{H}}_{ij}$ is close to zero. In our experiments, we found that a good implementation of the notion of closeness is via a Gaussian hypothesis test, $|\hat{\mathbf{H}}_{ij}| < c_{\alpha/2}/\sqrt{M}$, where $c_{\alpha/2}$ is the top- $\alpha/2$ quantile of the standard Gaussian distribution.

⁴Note that while it’s uncertain whether the reverse claims hold in general, this graph discovery method remains practical for providing a suitable graph for our downstream tasks, i.e., DDO.



(a) Regret (lower-better) (b) Value (higher-better)

Figure 3: Figure (3a): regret (lower-better) of DDO with (blue) and without (orange) FGM, against dimension in the toy quadratic problem. Averaged over 50 runs, showing 95%-confidence intervals. Figure (3b): value (higher-better) of DDO with MLP neural networks, for Gaussian data, with (blue) and without (orange) FGM, against the iteration of gradient ascent. Averaged over 128 designs, showing one-tenth of standard deviation.

7 EXPERIMENTS

In this section, we present numerical experiments to verify our theoretical findings and examine how readily applicable they are in practice. We start with the setting where the FGM is known, and then explore FGM discovery methods with learned latent spaces.

DDO with known FGMs. To empirically validate the regret results from Section 5, we conducted a numerical experiment with a quadratic ground-truth function $f(\mathbf{x}) = x_1x_2 + x_2x_3 + \dots + x_{d-1}x_d + x_dx_1$, where $\forall i; \mathcal{X}_i = \{0, 1\}$, and the approximation $\hat{f}(\mathbf{x})$ is a learned quadratic function. We generate the data from a uniform distribution and compare naïve DDO in Eq. (1) with DDO using FGMs in Eq. (2). Figure 3a shows that, as predicted by Theorem 2, the regret of DDO with FGMs remains near zero even as the problem dimensionality increases, while the regret of naïve DDO explodes. We then ran a more complex version of this experiment, which we visualize in Figure 3b. Here, the data is drawn from the standard Gaussian distribution and the ground-truth function is a random mixture of radial-basis functions (which we describe in more detail below). We fitted 1) a neural network, and 2) a neural network with FGM decomposition (see Appendix C for details), and then optimized the input into each of them with gradient ascent to obtain an estimate of the optimal \mathbf{x} . We plot the value of these solutions (standardized to have zero mean and unit variance), since the regret is not analytically tractable. The results likewise show a large improvement in performance from utilizing FGMs.

DDO with unknown FGMs. Next, we study the effectiveness of our algorithms in several synthetic examples where the FGM is not provided a priori, and where functional independencies must be discovered automatically, including in high-dimensional do-

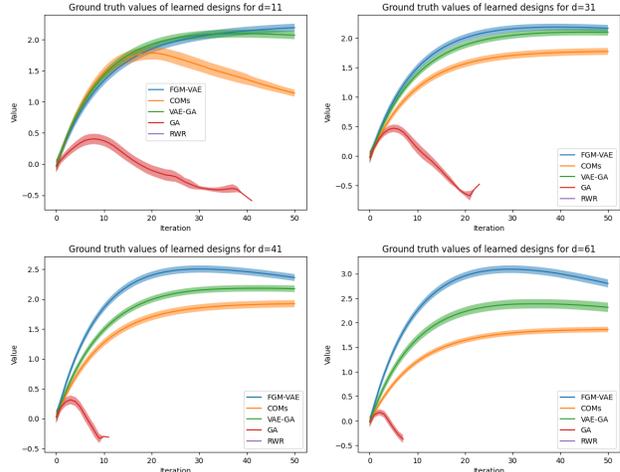


Figure 4: Values of $f(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}} \sim \pi$, along the course of gradient ascent on π , for 11-, 21-, 31-, and 41-dimensional problems (x -axis: iterations, y -axis: design values). The evaluation is over the top-128 designs from a sample of 1024. Lack of curve indicates generation of invalid designs. As the dimensionality grows, the FGM decomposition becomes more circritical.

mains. The data is generated as follows. We construct the ground-truth function $f(\mathbf{x})$ by, first, setting the set of maximal cliques \mathcal{C} of its FGM to be triangles $\{0, 1, 2\}, \{2, 3, 4\}, \{4, 5, 6\}, \dots, \{d-2, d-1, d\}$. Crucially, these cliques overlap, introducing dependencies between their variables. See Figure 1 for an illustration of the case with $d = 7$. The target function $f(\mathbf{x})$ is a mixture of radial-basis functions,

$$f(\mathbf{x}) = \sum_{C \in \mathcal{C}} \mathbf{w}_C \cdot \exp(-\|\mathbf{x}_C - \boldsymbol{\mu}_C\|^2),$$

where the centers $\{\boldsymbol{\mu}_C \mid C \in \hat{\mathcal{C}}\}$ are sampled from a Gaussian distribution and the positive weights $\{\mathbf{w}_C \mid C \in \hat{\mathcal{C}}\}$ are also random. We evaluate $y = f(\mathbf{x})$ on N ground-truth inputs drawn from the standard Gaussian distribution, $\mathbf{x} \sim \mathcal{N}(0_d, I_d)$. We then replace the ground-truth Gaussian inputs \mathbf{x} with observable designs $\mathbf{x}_{\text{obs}} \in \mathbb{R}^{d_{\text{obs}}}$ obtained by *bijectionally* composing a random affine map with the *softplus* nonlinearity. With a slight abuse of notation, the bijection assures that the reparameterization $y = f(\mathbf{x}_{\text{obs}})$ is well-defined. As such, the resulting DDO dataset takes the form $\{\mathbf{x}_{\text{obs}}^{(i)}, y^{(i)}\}_{i=1}^N$, with designs \mathbf{x}_{obs} following a non-Gaussian distribution and laying in a subset of $\mathbb{R}^{d_{\text{obs}}}$.

It is important to note that, since the image of the softplus is the set of positive real numbers, the *valid* values of \mathbf{x}_{obs} (those for which there exists a ground-truth \mathbf{x}) form only a subset of $\mathbb{R}^{d_{\text{obs}}}$. Thus, one way for a DDO method to fail is to generate a design \mathbf{x}_{obs} for which there exists no $\mathbf{x}(\mathbf{x}_{\text{obs}})$.

Since we do not assume access to the function’s FGM, nor Gaussianity of the data, we begin by learning marginally-Gaussian representations $\mathbf{z}(\mathbf{x}_{\text{obs}})$ of the designs. In our experiments, we choose to learn them with *Variational Auto-Encoders* [17, VAE]. Viewing the design values as a function of the Gaussian representations, $y = f(\mathbf{z})$, we then compute the estimator from Eq. (6) to infer the FGM of $f(\mathbf{z})$ and extract its clique set $\hat{\mathcal{C}}$. We then learn a predictive neural network model that decomposes over these cliques, and perform gradient ascent with respect to it over the \mathbf{z} variable. Then, we decode the representation of such a design $\hat{\mathbf{z}}$ with our VAE decoder to propose a new design candidate $\hat{\mathbf{x}}_{\text{obs}}$. We summarize the entire algorithm for DDO with FGMs in Algorithm 1.

Algorithm 1 DDO with FGMs

- 1: **Input:** dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$
- 2: **Preprocess** $\{\mathbf{x}^{(i)}\}_{i=1}^N$ so that they approximately follow the standard Gaussian distribution.
- 3: Approximate the cliques of the FGM of $f(\mathbf{x})$ with Equation (6)
- 4: Learn the functions $\hat{f}_{\mathcal{C}}(\mathbf{x}_{\mathcal{C}})$ by minimizing the mean squared error

$$\mathbb{E}_{(\mathbf{x}, y) \sim p} \left[\left(\sum_{\mathcal{C} \in \hat{\mathcal{C}}} \hat{f}_{\mathcal{C}}(\mathbf{x}_{\mathcal{C}}) - y \right)^2 \right]$$

- 5: $\hat{\pi} = \arg \max_{\pi} \mathbb{E}_{\mathbf{x} \sim \pi} \left[\sum_{\mathcal{C} \in \hat{\mathcal{C}}} \hat{f}_{\mathcal{C}}(\mathbf{x}_{\mathcal{C}}) \right]$.
 - 6: **Invert** the preprocessing step for $\mathbf{x} \sim \hat{\pi}$.
-

We compare this method with naïve gradient ascent (GA), reward-weighted regression [26, RWR]—to see the effectiveness of naïve methods in the tasks we designed; conservative objective models [35, COMs]—to compare to an off-the-shelf state-of-the-art method; and an ablative method that learns VAE representations and performs gradient ascent directly on them (VAE-GA), similarly to [9]—to isolate the benefits of FGM decomposition on top of the learned latent space.

Results in Figure 4 show that incorporating FGMs into DDO leads to better designs compared to baselines, and the gap increases with the dimensionality of the problem, as expected from Theorem 2. It is also worth noting that the naïve baselines—GA and RWR—tend to generate invalid designs quite quickly.

One may wonder if Gaussianity of both the distribution of the ground-truth inputs \mathbf{x} and the VAE prior simplifies the task for our algorithm, since the VAE preprocessing step could learn to invert our non-linear data transformation. This, however, is very unlikely given that there are infinitely many bijective reparameterizations of \mathbf{x} that could explain $f(\mathbf{x})$ perfectly and be also Gaussian (*e.g.*, rotations of \mathbf{x}). Nevertheless, as a sanity check, we re-ran the experiments

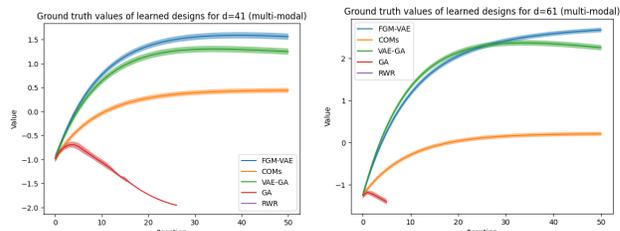


Figure 5: Values of $f(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}} \sim \pi$, along the course of gradient ascent on π , for 41-, and 61-dimensional problems (x -axis: iterations, y -axis: design values). The evaluation is over the top-128 designs from a sample of 1024. Lack of curve indicates generation of invalid designs. The unobserved base distribution generating the data was an even mixture of two Gaussian distributions $\mathcal{N}(-1, I)$ and $\mathcal{N}(1, I)$, with the ground-truth inputs \mathbf{x} sampled from a mixture of Gaussians, rather than the unimodal Gaussian distribution. This way we made sure that the representations learned by the VAE are significantly distinct from the ground-truth inputs, as well as made the multi-modality of the observed designs \mathbf{x}_{obs} more explicit. Figure 5 shows that our findings are preserved under this setting.

8 CONCLUSION

We showed how DDO can be made feasible and tractable by incorporating graphical factorization structure. We capture this structure with functional graphical models (FGMs), and we prove that a function can decompose over its maximal cliques analogously to how probabilistic graphical models (PGMs) decompose probability distributions. We then showed theoretically that DDO with FGMs admits sample-efficient optimization without requiring that the data distribution to directly cover optimal designs, instead only requiring coverage of optimal values for individual cliques of the FGM. While we did verify our findings experimentally, our experiments serve mainly as numerical verification of our theoretical findings. Enabling data-efficient optimization in practical real-world settings will require additional development of our proposed algorithmic framework, and presents a number of exciting challenges for future work. For example, while our proposed FGM discovery approach is relatively simple and pragmatic, the general problem of discovering such functional independence relationships might relate to other fields that are concerned with recovery structure: discovery of probabilistic independence relationships, causal inference, and structure learning. Incorporating ideas from these fields into the FGM framework might not only constitute an exciting direction for future work, but might also lead to powerful new methods that can discover significantly improved designs from suboptimal datasets.

Acknowledgements

We thank Seohong Park for inspiring discussions on technical aspects of FGMs and James Bowden for inspiring conversations on their applications to scientific problems.

References

- [1] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep.*, 32, 2019.
- [2] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.
- [3] David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, pages 773–782. PMLR, 2019.
- [4] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.
- [5] P Clifford and JM Hammersley. Markov fields on finite graphs and lattices. 1971.
- [6] Bin Dai and David Wipf. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.
- [7] Murat A Erdogdu. Newton-stein method: A second order method for glms via stein’s lemma. *Advances in Neural Information Processing Systems*, 28, 2015.
- [8] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.
- [9] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.
- [10] Trevor Hastie and Robert Tibshirani. Generalized additive models: some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987.
- [11] Trong Nghia Hoang, Quang Minh Hoang, Ruofei Ouyang, and Kian Hsiang Low. Decentralized high-dimensional bayesian optimization with factor graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [12] Warren Hoburg and Pieter Abbeel. Geometric programming for aircraft design optimization. *AIAA Journal*, 52(11):2414–2426, 2014.
- [13] Michael Irwin Jordan. *Learning in graphical models*. 1999.
- [14] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *International conference on machine learning*, pages 295–304. PMLR, 2015.
- [15] Sheng-Chun Kao, Geonhwa Jeong, and Tushar Krishna. Confucius: Autonomous hardware resource assignment for dnn accelerators using reinforcement learning. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 622–636. IEEE, 2020.
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [18] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [19] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.
- [20] Aviral Kumar and Sergey Levine. Model inversion networks for model-based optimization. *Advances in Neural Information Processing Systems*, 33:5126–5137, 2020.
- [21] Aviral Kumar, Amir Yazdanbakhsh, Milad Hashemi, Kevin Swersky, and Sergey Levine. Data-driven offline optimization for architecting hardware accelerators. *arXiv preprint arXiv:2110.11346*, 2021.
- [22] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.
- [23] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

- [24] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [25] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- [26] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750, 2007.
- [27] Han Qi, Yi Su, Aviral Kumar, and Sergey Levine. Data-driven offline decision-making via invariant representation learning. *arXiv preprint arXiv:2211.11349*, 2022.
- [28] Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.
- [29] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.
- [30] Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional bayesian optimization via additive models with overlapping groups. In *International conference on artificial intelligence and statistics*, pages 298–307. PMLR, 2018.
- [31] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.
- [32] Charles Stein, Persi Diaconis, Susan Holmes, and Gesine Reinert. Use of exchangeable pairs in the analysis of simulations. *Lecture Notes-Monograph Series*, pages 1–26, 2004.
- [33] Milan Studeny. *Probabilistic conditional independence structures*. Springer Science & Business Media, 2006.
- [34] Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pages 21658–21676. PMLR, 2022.
- [35] Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pages 10358–10368. PMLR, 2021.
- [36] MJ Wainright. High dimensional statistics. a non-asymptotic point of view, 2019.
- [37] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.
- [38] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.
- [39] Juliusz Ziomek and Haitham Bou-Ammar. Are random decompositions all we need in high dimensional bayesian optimisation? *arXiv preprint arXiv:2301.12844*, 2023.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable] We use deep neural networks in our algorithms which prevents us from analyzing some of the algorithms' properties.
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]

- (b) Complete proofs of all theoretical results. [Yes]
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
- (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
- (a) Citations of the creator If your work uses existing assets. [Not Applicable]
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
 - (d) Information about consent from data providers/curators. [Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
- (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Auxiliary Facts

Theorem 3 (Hammersley-Clifford [5]). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple graph with clique set \mathcal{C} , and let $\mathbf{x} = \mathbf{x}_{\mathcal{V}} \sim p$, where $p(\mathbf{x})$ is a strictly positive probability distribution. If p is Markov with respect to \mathcal{G} , then, $\forall \mathbf{x} \in \mathcal{X}$,*

$$p(\mathbf{x}) = \prod_{C \in \mathcal{C}} \phi_C(\mathbf{x}_C),$$

for some functions $\phi_C(\mathbf{x}_C) > 0$.

Lemma 3. *Let $\bar{\mathbf{x}} \in \mathbb{R}^{d_{\star}}$ and $i \neq j \in [d_{\star}]$. Then*

$$\frac{\partial^2}{\partial \bar{x}_j \partial \bar{x}_i} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[f(\bar{\mathbf{x}} + \mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[x_i x_j f(\bar{\mathbf{x}} + \mathbf{x})].$$

Proof. We start from proving a known result. Namely,

$$\frac{\partial}{\partial \bar{x}_i} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[f(\bar{\mathbf{x}} + \mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[x_i f(\bar{\mathbf{x}} + \mathbf{x})]. \quad (7)$$

To do it, we use integration by parts,

$$\begin{aligned} \frac{\partial}{\partial \bar{x}_i} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[f(\bar{\mathbf{x}} + \mathbf{x})] &= \mathbb{E}_{\mathbf{x} \sim \mathcal{N}} \left[\frac{\partial}{\partial \bar{x}_i} f(\bar{\mathbf{x}} + \mathbf{x}) \right] \\ &= \mathbb{E}_{\mathbf{x} \sim \mathcal{N}} \left[\frac{\partial}{\partial x_i} f(\bar{\mathbf{x}} + \mathbf{x}) \right] \\ &= \mathbb{E}_{\mathbf{x}_{-i} \sim \mathcal{N}} \left[\int_{x_i} dx_i \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-1}{2} x_i^2\right) \frac{\partial}{\partial x_i} f(\bar{\mathbf{x}} + \mathbf{x}) \right] \\ &= \mathbb{E}_{\mathbf{x}_{-i} \sim \mathcal{N}} \left[\left[\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-1}{2} x_i^2\right) f(\bar{\mathbf{x}} + \mathbf{x}) \right]_{-\infty}^{\infty} + \int_{x_i} dx_i \frac{x_i}{\sqrt{2\pi}} \exp\left(\frac{-1}{2} x_i^2\right) f(\bar{\mathbf{x}} + \mathbf{x}) \right] \\ &= \mathbb{E}_{\mathbf{x}_{-i} \sim \mathcal{N}} \left[0 + \int_{x_i} dx_i \frac{x_i}{\sqrt{2\pi}} \exp\left(\frac{-1}{2} x_i^2\right) f(\bar{\mathbf{x}} + \mathbf{x}) \right] \\ &= \mathbb{E}_{\mathbf{x} \sim \mathcal{N}} [x_i f(\bar{\mathbf{x}} + \mathbf{x})]. \end{aligned}$$

Now, applying this result twice,

$$\begin{aligned} \frac{\partial^2}{\partial \bar{x}_j \partial \bar{x}_i} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[f(\bar{\mathbf{x}} + \mathbf{x})] &= \frac{\partial}{\partial \bar{x}_j} \frac{\partial}{\partial \bar{x}_i} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[f(\bar{\mathbf{x}} + \mathbf{x})] \\ &= \frac{\partial}{\partial \bar{x}_j} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[x_i f(\bar{\mathbf{x}} + \mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim \mathcal{N}}[x_i x_j f(\bar{\mathbf{x}} + \mathbf{x})], \end{aligned}$$

as required. □

B Omitted Proofs Of Our Results

B.1 Proof of Lemma 1

Lemma 1. *Suppose $f(\mathbf{x})$ is twice-continuously differentiable w.r.t. \mathbf{x} . Let $A, B, S \subseteq \mathcal{V}$. Then, the following statements are equivalent.*

1. \mathbf{x}_A and \mathbf{x}_B are functionally independent given \mathbf{x}_S .
2. For every $\mathbf{x} \in \mathcal{X}$,

$$\frac{\partial f}{\partial \mathbf{x}_{A \setminus S}}(\mathbf{x}) = F_{-(B \setminus S)}(\mathbf{x}_{\mathcal{V} \setminus (B \setminus S)}),$$

for some function $F_{-(B \setminus S)}(\mathbf{x}_{\mathcal{V} \setminus (B \setminus S)})$ that does not take $\mathbf{x}_{(B \setminus S)}$ as input.

3. For every $\mathbf{x} \in \mathcal{X}$,

$$\frac{\partial^2 f}{\partial \mathbf{x}_{A \setminus S} \partial \mathbf{x}_{B \setminus S}}(\mathbf{x}) = 0.$$

Proof. 1 \iff 2: Let us, for clarity, write $\underline{A} = A \setminus S$ and $\underline{B} = B \setminus S$. We suppose that 1 holds as in Definition 1,

$$\frac{\partial f}{\partial \mathbf{x}_{\underline{A}}}(\mathbf{x}) = F_{-\underline{B}}(\mathbf{x}_{\mathcal{V} \setminus \underline{B}}). \quad (8)$$

Integrating it with respect to $\mathbf{x}_{\underline{A}}$ gives

$$\begin{aligned} f(\mathbf{x}) &= \int_{\mathbf{x}_{\underline{A}}} F_{-\underline{B}}(\mathbf{x}_{\mathcal{V} \setminus \underline{B}}) d\mathbf{x}_{\underline{A}} + f_{-\underline{A}}(\mathbf{x}_{\mathcal{V} \setminus \underline{A}}) \\ &= f_{-\underline{B}}(\mathbf{x}_{\mathcal{V} \setminus \underline{B}}) + f_{-\underline{A}}(\mathbf{x}_{\mathcal{V} \setminus \underline{A}}), \end{aligned}$$

for some $f_{-\underline{B}}$ and $f_{-\underline{A}}$. This proves statement 2 and allows us to recover Equation (8) for \underline{A} (\underline{B}) by differentiating with respect to $\mathbf{x}_{\underline{A}}$ ($\mathbf{x}_{\underline{B}}$).

1 \iff 3: Suppose that 1 holds as in Definition 1. Since the right-hand side of the equation is not a function of $\mathbf{x}_{\underline{B}}$, differentiating the equation with respect to $\mathbf{x}_{\underline{B}}$ gives 0, which proves statement 3. We recover statement 1 from statement 3 by integrating with respect to $\mathbf{x}_{\underline{B}}$ (or $\mathbf{x}_{\underline{A}}$). \square

B.2 Proof of Theorem 1

Theorem 1 (Function Decomposition). *Assume that $\int_{\mathcal{X}} \exp f(\mathbf{x}) d\mathbf{x}$ exists and let \mathcal{G} be any FGM of $f(\mathbf{x})$ and \mathcal{C} be its set of maximal cliques (clique set). Then, there exist functions $\{f_C(\mathbf{x}_C), C \in \mathcal{C}\}$, such that*

$$f(\mathbf{x}) = \sum_{C \in \mathcal{C}} f_C(\mathbf{x}_C).$$

Proof. Consider a probability distribution defined by

$$e_f(\mathbf{x}) = \frac{\exp f(\mathbf{x})}{Z}, \quad \text{where } Z \triangleq \int_{\mathbf{x}} \exp f(\mathbf{x}) d\mathbf{x}.$$

Then, let i and j be nodes in \mathcal{G} that are not linked: $(i, j) \notin \mathcal{E}$. By Definition 2, it follows that

$$\begin{aligned} e_f(\mathbf{x}) &= \frac{\exp(f_{-i}(\mathbf{x}_{-i}) + f_{-j}(\mathbf{x}_{-j}))}{Z} \\ &= \frac{\exp f_{-i}(\mathbf{x}_{-i})}{\sqrt{Z}} \frac{\exp f_{-j}(\mathbf{x}_{-j})}{\sqrt{Z}}. \end{aligned}$$

That is, measure $e_f(\mathbf{x})$ factorizes over $\{\mathcal{V} \setminus \{i\}, \mathcal{V} \setminus \{j\}\}$. Hence, random variables \mathbf{x}_i and \mathbf{x}_j are (probabilistically) independent given $\mathbf{x}_{\mathcal{V} \setminus \{i,j\}}$ [33]. As the pair (i, j) was arbitrary, this implies that $e_f(\mathbf{x})$ satisfies the *pairwise Markov property* with respect to \mathcal{G} (equivalently, \mathcal{G} is an I-map of $e_f(\mathbf{x})$). As the measure induced by $e_f(\mathbf{x})$ is strictly positive, *Hammersley-Clifford Theorem* [5] implies that $e_f(\mathbf{x})$ factorizes according to \mathcal{G} . That is, for the clique set \mathcal{C} of \mathcal{G} , there exist positive functions $\{u_C(\mathbf{x}_C) \mid C \in \mathcal{C}\}$, such that

$$e_f(\mathbf{x}) = \prod_{C \in \mathcal{C}} u_C(\mathbf{x}_C).$$

Taking the logarithm on both sides, we get

$$f(\mathbf{x}) = \sum_{C \in \mathcal{C}} \log u_C(\mathbf{x}_C) + \log Z \triangleq \sum_{C \in \mathcal{C}} f_C(\mathbf{x}_C) + \log Z.$$

That is, the original function $f(\mathbf{x})$ can be expressed as a sum of functions of cliques of graph \mathcal{G} (up to an input-independent constant). Of course, the constant can be subsumed into the decomposing functions. This lets us conclude the proof of the theorem. \square

B.3 Proof of Theorem 2

Theorem 2 (Regret of DDO with FGMs). *Let \mathcal{C} be the set of maximal cliques of an FGM of $f(\mathbf{x})$ and $f(\mathbf{x}) = \sum_{C \in \mathcal{C}} f_C(\mathbf{x}_C)$ be its FGM decomposition. Under the following assumptions $\forall C \in \mathcal{C}$,*

(1) *the function approximator classes are centered*

$$\mathbb{E}_{\mathbf{x} \sim p}[\bar{f}_C(\mathbf{x}_C)] = 0, \forall \bar{f}_C \in \mathcal{F}_C,$$

(2) *the correlations between cliques are well-controlled*

$$\max_{\{\bar{f}_C \in \mathcal{F}_C\}} \text{Corr}[\bar{f}_{C_1}(\mathbf{x}_{C_1}), \bar{f}_{C_2}(\mathbf{x}_{C_2})] \leq \sigma,$$

(3) *the models are well-specified, sp that $f_C \in \mathcal{F}_C$, and*

(4) *$|\bar{f}_C(\mathbf{x}_C)| \leq 1$ for any $\mathbf{x}_C \in \mathcal{X}_C, \bar{f}_C \in \mathcal{F}_C$,*

the following result holds with probability $1 - \delta$

$$J(\pi^*) - J(\hat{\pi}_{\text{FGM}}) \lesssim \underbrace{\sqrt{\frac{1}{1 - \sigma}}}_{\text{Corr}(b'')} \underbrace{\max_{C \in \mathcal{C}} \max_{\bar{\pi} \in \Pi} \max_{\mathbf{x}_C \in \mathcal{X}_C} \left| \frac{\bar{\pi}(\mathbf{x}_C)}{p(\mathbf{x}_C)} \right|}_{\text{Coverage}(b)} \underbrace{\sqrt{\frac{|\mathcal{C}| \sum_C \log(|\mathcal{F}_C|/\delta)}{n}}}_{\text{Complexity}(b')},$$

where $p(\mathbf{x}_C), \bar{\pi}(\mathbf{x}_C)$ denote the marginal distributions of \mathbf{x}_C with respect to $p(\mathbf{x})$ and $\pi(\mathbf{x})$.

Proof. Following the standard literature of regression [1] and using Assumption (3), by leveraging $f_C \in \mathcal{F}_C$, we can obtain with probability $1 - \delta$,

$$\mathbb{E}_{\mathbf{x} \sim p} \left[(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \right] \leq \text{Stat}, \quad \text{Stat} = D \times \frac{\sum_{C \in \mathcal{C}} \log(|\mathcal{F}_C|/\delta)}{n}. \quad (9)$$

where $\hat{f}(\mathbf{x}) = \sum_C \hat{f}_C(\mathbf{x}_C), f(\mathbf{x}) = \sum_C f_C(\mathbf{x}_C)$, and D is universal constant, noting $\log(|\mathcal{F}|) = \sum_C \log(|\mathcal{F}_C|)$. Note this statement states that the MSE guarantee is ensured for the whole $\hat{f}(\mathbf{x})$.

Now, we have

$$\begin{aligned} & \mathbb{E}_{\mathbf{x} \sim p} [(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2] \\ &= \mathbb{E}_{\mathbf{x} \sim p} \left[\left(\sum_C \{\hat{f}_C(\mathbf{x}_C) - f_C(\mathbf{x}_C)\} \right)^2 \right] \\ &= \sum_C \mathbb{E}_{\mathbf{x} \sim p} \left[\{\hat{f}_C(\mathbf{x}_C) - f_C(\mathbf{x}_C)\}^2 \right] + \sum_{C_1 \neq C_2} \mathbb{E}_{\mathbf{x} \sim p} \left[\{\hat{f}_{C_1}(\mathbf{x}_{C_1}) - f_{C_1}(\mathbf{x}_{C_1})\} \{\hat{f}_{C_2}(\mathbf{x}_{C_2}) - f_{C_2}(\mathbf{x}_{C_2})\} \right]. \end{aligned}$$

Here, using Assumption (1) and (2), note

$$\begin{aligned} & \mathbb{E}_{\mathbf{x} \sim p} \left[\{ \hat{f}_{C_1}(\mathbf{x}_{C_1}) - f_{C_1}(\mathbf{x}_{C_1}) \} \{ \hat{f}_{C_2}(\mathbf{x}_{C_2}) - f_{C_2}(\mathbf{x}_{C_2}) \} \right] \\ & \leq \sigma \times \left\{ \mathbb{E}_{\mathbf{x} \sim p} \left[\{ \hat{f}_{C_1}(\mathbf{x}_{C_1}) - f_{C_1}(\mathbf{x}_{C_1}) \}^2 \right]^{1/2} \right\} \times \left\{ \mathbb{E}_{\mathbf{x} \sim p} \left[\{ \hat{f}_{C_2}(\mathbf{x}_{C_2}) - f_{C_2}(\mathbf{x}_{C_2}) \}^2 \right]^{1/2} \right\}. \end{aligned}$$

Hence, we get

$$\begin{aligned} & \mathbb{E}_{\mathbf{x} \sim p} [\{ \hat{f}(\mathbf{x}) - f(\mathbf{x}) \}^2] \tag{10} \\ & \geq \sum_C \mathbb{E}_{\mathbf{x} \sim p} [\{ \hat{f}_C(\mathbf{x}_C) - f_C(\mathbf{x}_C) \}^2] \end{aligned}$$

$$+ \sigma \times \left\{ \mathbb{E}_{\mathbf{x} \sim p} \left[\{ \hat{f}_{C_1}(\mathbf{x}_{C_1}) - f_{C_1}(\mathbf{x}_{C_1}) \}^2 \right]^{1/2} \right\} \times \left\{ \mathbb{E}_{\mathbf{x} \sim p} \left[\{ \hat{f}_{C_2}(\mathbf{x}_{C_2}) - f_{C_2}(\mathbf{x}_{C_2}) \}^2 \right]^{1/2} \right\} \tag{11}$$

$$\geq \sigma \left\{ \sum_C \mathbb{E}_{\mathbf{x} \sim p} \left[\{ \hat{f}_C(\mathbf{x}_C) - f_C(\mathbf{x}_C) \}^2 \right]^{1/2} \right\}^2 + \{1 - \sigma\} \sum_{C \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim p} [\{ \hat{f}_C(\mathbf{x}_C) - f_C(\mathbf{x}_C) \}^2]. \tag{12}$$

$$\geq \{1 - \sigma\} \sum_{C \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim p} [\{ \hat{f}_C(\mathbf{x}_C) - f_C(\mathbf{x}_C) \}^2]. \tag{13}$$

Therefore, we have

$$\sum_{C \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim p} [\{ \hat{f}_C(\mathbf{x}_C) - f_C(\mathbf{x}_C) \}^2] \leq \frac{1}{1 - \sigma} \times \text{Stat}. \tag{14}$$

Then, for any policy $\pi \in \Pi$, we get

$$\begin{aligned} J(\pi) - J(\hat{\pi}) & \leq J(\pi) - \mathbb{E}_{\mathbf{x} \sim \pi} [\hat{f}(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim \pi} [\hat{f}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim \hat{\pi}} [\hat{f}(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim \hat{\pi}} [\hat{f}(\mathbf{x})] - J(\hat{\pi}) \\ & = J(\pi) - \mathbb{E}_{\mathbf{x} \sim \pi} [\hat{f}(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim \hat{\pi}} [\hat{f}(\mathbf{x})] - J(\hat{\pi}) \\ & = \mathbb{E}_{\mathbf{x} \sim \pi} [f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim \pi} [\hat{f}(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim \hat{\pi}} [\hat{f}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim \hat{\pi}} [f(\mathbf{x})]. \end{aligned}$$

We use the following to upper-bound last terms:

$$|\mathbb{E}_{\mathbf{x} \sim \pi} [f(\mathbf{x}) - \hat{f}(\mathbf{x})]| \leq \sum_{C \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim \pi} [|f_C(\mathbf{x}_C) - \hat{f}_C(\mathbf{x}_C)|] \tag{Triangle inequality}$$

$$\leq \left\{ \max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \right\} \sum_{C \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim p} [|f_C(\mathbf{x}_C) - \hat{f}_C(\mathbf{x}_C)|] \tag{Importance sampling}$$

$$\leq \left\{ \max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \right\} \sum_{C \in \mathcal{C}} \sqrt{\mathbb{E}_{\mathbf{x} \sim p} [\{ f_C(\mathbf{x}_C) - \hat{f}_C(\mathbf{x}_C) \}^2]} \tag{Jensen's inequality}$$

$$\leq \left\{ \max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \right\} \sqrt{|\mathcal{C}|} \sqrt{\sum_{C \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim p} [\{ f_C(\mathbf{x}_C) - \hat{f}_C(\mathbf{x}_C) \}^2]} \tag{CS inequality}$$

$$= \max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \times \sqrt{|\mathcal{C}|} \times \sqrt{\frac{\text{Stat}}{1 - \sigma}}. \tag{Use Ineq. 14}$$

Finally, combining everything, we have

$$\begin{aligned} J(\pi) - J(\hat{\pi}) & \leq 2 \max_{\pi \in \Pi} \max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \times \sqrt{|\mathcal{C}|} \text{Stat} \\ & \lesssim \max_{\pi \in \Pi} \max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \times \sqrt{\frac{|\mathcal{C}| \sum_{C \in \mathcal{C}} \log(|\mathcal{F}_C|/\delta)}{n}} \times \sqrt{\frac{1}{1 - \sigma}}. \end{aligned}$$

□

B.4 Proof of Lemma 2

Lemma 2 (Improvement of Coverage terms). *For any $\pi \in \Delta(\mathcal{X})$, we have*

$$\max_{C \in \mathcal{C}} \max_{\mathbf{x}_C \in \mathcal{X}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \leq \max_{\mathbf{x} \in \mathcal{X}} \frac{\pi(\mathbf{x})}{p(\mathbf{x})}.$$

Proof. Let $\mathbf{x} = [\mathbf{x}_C^\top, \{\mathbf{x}'\}_C^\top]^\top$. Then,

$$\max_{\mathbf{x}'_C} \frac{\pi(\mathbf{x})}{p(\mathbf{x})} = \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)} \times \max_{\mathbf{x}'_C} \frac{\pi(\mathbf{x}'_C | \mathbf{x}_C)}{p(\mathbf{x}'_C | \mathbf{x}_C)} \geq \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)}. \quad (15)$$

Here, we use $\max_{\mathbf{x}'_C} \frac{\pi(\mathbf{x}'_C | \mathbf{x}_C)}{p(\mathbf{x}'_C | \mathbf{x}_C)} \geq 1$. This is because if $\max_{\mathbf{x}'_C} \frac{\pi(\mathbf{x}'_C | \mathbf{x}_C)}{p(\mathbf{x}'_C | \mathbf{x}_C)} < 1$, we would get contradiction:

$$1 = \sum_{\mathbf{x}'_C} \pi(\mathbf{x}'_C | \mathbf{x}_C) \leq \sum_{\mathbf{x}'_C} p(\mathbf{x}'_C | \mathbf{x}_C) \max_{\mathbf{x}'_C} \frac{\pi(\mathbf{x}'_C | \mathbf{x}_C)}{p(\mathbf{x}'_C | \mathbf{x}_C)} < \sum_{\mathbf{x}'_C} p(\mathbf{x}'_C | \mathbf{x}_C) = 1.$$

Finally, using Eq. 15, we obtain

$$\max_{\mathbf{x}} \frac{\pi(\mathbf{x})}{p(\mathbf{x})} = \max_{\mathbf{x}_C} \max_{\mathbf{x}'_C} \frac{\pi(\mathbf{x})}{p(\mathbf{x})} \geq \max_{\mathbf{x}_C} \frac{\pi(\mathbf{x}_C)}{p(\mathbf{x}_C)}.$$

□

C Details of Experiments

We conducted our experiments in a Google Colab with TPU. One can access it via the following link:

<https://colab.research.google.com/drive/1qt4M3C35bvjRHPiPbxE3zPc5zvX6AAU4?usp=sharing>.

In all runs, all neural networks were two-hidden layer MLPs, with 256 hidden units. Each of them was trained with Adam [16] optimization algorithm with batch size of 128. Hyperparameters for COMs were taken from the COMs paper [35] and we did not observe any improvement by tuning them. As hyperparameters for all other methods are a subset of hyperparameters of COMs, we used the same hyperparameters, except that methods that use VAEs [17] have 10 times smaller gradient ascent learning rate. We also used the common choice for temperature $\tau = 0.05$ for reward-weighted regression [26, 25, 19].

To represent a decomposed function $f(\mathbf{x}) = \sum_{C \in \mathcal{C}} f_C(\mathbf{x}_C)$, we use a single neural-network model of the same class as for all other methods. In our experiments, it was a multi-layer perceptron (MLP). The decomposition was implemented by duplicating the input (row) vector \mathbf{x} $|\mathcal{C}|$ times and stacking the copies horizontally. Each copy corresponded to one of the cliques. Then, for each clique we masked out inputs that were not included. That is, we multiplied the stacked copies of \mathbf{x} by the matrix \mathbf{C} such that $C_{ij} = 1$ if \mathbf{x}_j is in the i^{th} clique, and 0 otherwise. That is, we form the following input

$$\begin{bmatrix} \mathbf{C}_{11}\mathbf{x}_1 & \dots & \mathbf{C}_{1d}\mathbf{x}_d \\ & \dots & \\ \mathbf{C}_{|\mathcal{C}|1}\mathbf{x}_1 & \dots & \mathbf{C}_{|\mathcal{C}|d}\mathbf{x}_d \end{bmatrix}$$

We pass the rows to the MLP in parallel and return the mean of the $|\mathcal{C}|$ output values—this is equivalent to the summation from Theorem 1 in terms of representational power but we believe it may be more numerically stable when $|\mathcal{C}| \rightarrow \infty$.