Counterfactuals for Design: A Model-Agnostic Method For Design Recommendations

Lyle Regenwetter
Massachusetts Institute of Technology
Cambridge, MA
regenwet@mit.edu

Yazan Abu Obaideh
ProgressSoft
Amman, Jordan
yazan.amer@protonmail.com

Faez Ahmed
Massachusetts Institute of Technology
Cambridge, MA
faez@mit.edu

***Abstract*—We introduce Multi-Objective Counterfactuals for Design (MCD), a novel method for counterfactual optimization in design problems. Counterfactuals are hypothetical situations that can lead to a different decision or choice. In this paper, the authors frame the counterfactual search problem as a design recommendation tool that can help identify modifications to a design, leading to better functional performance. MCD improves upon existing counterfactual search methods by supporting multi-objective queries, which are crucial in design problems, and by decoupling the counterfactual search and sampling processes, thus enhancing efficiency and facilitating objective tradeoff visualization. The paper demonstrates MCD's core functionality using a two-dimensional test case, followed by three case studies of bicycle design that showcase MCD's effectiveness in real-world design problems. In the first case study, MCD excels at recommending modifications to query designs that can significantly enhance functional performance, such as weight savings and improvements to the structural safety factor. The second case study demonstrates that MCD can work with a pre-trained language model to suggest design changes based on a subjective text prompt effectively. Lastly, the authors task MCD with increasing a query design's similarity to a target image and text prompt while simultaneously reducing weight and improving structural performance, demonstrating MCD's performance on a complex multimodal query. Overall, MCD has the potential to provide valuable recommendations for practitioners and design automation researchers looking for answers to their "What if" questions by exploring hypothetical design modifications and their impact on multiple design objectives. The code, test problems, and datasets used in the paper are available to the public at decode.mit.edu/projects/counterfactuals/.**

I. Introduction

Modifying existing designs to generate new ones is an essential aspect of various engineering sectors, such as aerospace, automotive, architecture, pharmaceuticals, consumer goods, and many others. Design modification significantly impacts the performance, efficiency, and safety of engineered systems. Effective methods for design modification can lead to more sustainable and environmentally friendly technologies, better transportation systems, and safer infrastructure. Furthermore, improved design modification methods can enable cost savings and improved efficiency, making products more accessible and affordable for society. However, coming up with good design modifications can be challenging, as it requires navigating huge design spaces and making numerous trade-offs between competing objectives. Often there are too many design attributes and potential modifications to consider. Not surprisingly, designers often struggle with the available choices and may often ask themselves, "What if?".

As a designer, the ability to ask "What if?" questions is crucial in the iterative process of design modification. By exploring hypothetical scenarios, designers can identify opportunities to improve design performance and functionality. However, answering "What if?" questions can be challenging as it requires considering an extensive range of potential modifications and their effects on multiple design objectives. Counterfactuals are a powerful reasoning tool that allows designers to ask such questions by exploring hypothetical design modifications and their impact on multiple design objectives.

A counterfactual is a hypothetical situation that depicts what could have happened if a specific event or action did not occur. It requires envisioning an alternate reality where a different choice or decision was made and analyzing the differences in results. Counterfactuals are often employed in reasoning, decision-making, and causal inference. They aid in comprehending the impact of particular events or actions on outcomes and considering the ramifications of various choices.

Counterfactuals are typically employed to understand how an outcome would change given a different set of actions. This style of counterfactual can be applied to design problems to answer questions like: "How would the performance of this design change if I modified this particular attribute?" There are many tools to predict these 'classic' counterfactuals, such as simulations and predictive models. In this work, we instead consider an 'inverse' counterfactual problem, which states: "What events would have needed to occur to result in this other outcome?" In design contexts, this often equates to the question: "What attributes of my design would I need to change to achieve a particular performance target, design classification, or functional requirement?"

This paper proposes an approach to answer such 'inverse' counterfactual hypotheticals using multi-objective optimization. Our proposed approach, Multi-Objective Counterfactuals for Design (MCD), allows users to input a design and a set of desired attributes, then recommends targeted modifications to the design to achieve these attributes. It identifies these modifications by querying a set of attribute predictors in a directed search procedure dictated by an evolutionary algorithm. We demonstrate how predictors ranging from machine learning regressors to text embedding models can support target attributes ranging from functional performance targets to subjective text requirements.

MCD can be viewed as an AI design assistant that allows users to ask challenging objective and subjective questions about an existing design, such as: "What modifications would it take to make this product 10% lighter?", "What would make my design look like this other concept?", or "How would my design need to change to look more sleek and futuristic?" By enabling designers to interact with AI systems simply and intuitively, counterfactuals open the doors to more successful human-AI collaboration by enhancing and accelerating the design process. A block diagram demonstrating MCD's anticipated usage scenario is shown in Figure 1.

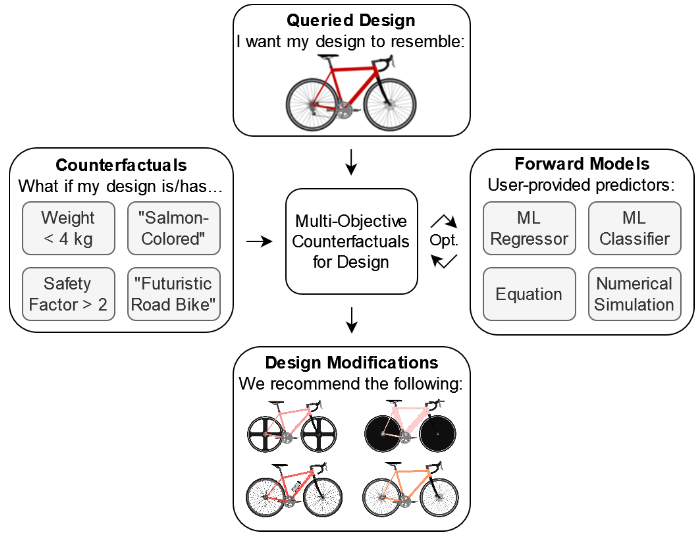


Fig. 1: Multi-Objective Counterfactuals for Design (MCD) is a human-AI collaborative design recommendation tool. Users provide an initial design and a set of counterfactual attributes they would like to achieve. MCD queries a set of attribute predictors to search for a set of diverse modifications to the original design that achieve the counterfactual attributes.

A particularly related body of research to our work is counterfactual explanations, originally developed as a tool to interpret black-box machine learning (ML) models. Counterfactual explanations allow practitioners to understand the behavior of otherwise uninterpretable models by asking questions about counterfactual scenarios. A classic motivating example for counterfactual explanations involves a model that is deciding whether to approve a loan, where the applicant may ask: "What would I need to change for this model to approve my application?" Broadly speaking, these counterfactuals answer a very versatile question: "Hypothetically, what would I need to change about the input to my model for it to predict another outcome?" Many of the common challenges that designers face can be framed as such a question. For example, given a model that predicts the functional performance of a design, a designer can ask how to change the design to achieve some desired functional performance. Despite this, counterfactual explanations have not yet been used in design engineering problems, to the best of our knowledge[1].

In this paper, we showcase our MCD method and demonstrate that counterfactual search is a simple yet powerful AI-driven design tool that real designers can leverage for a variety of tasks. To do so, we make several key contributions, which we summarize as follows:

1) We introduce Multi-Objective Counterfactuals for Design (MCD), a new method to search for counterfactual design modifications to achieve desired outcomes. We formulate MCD as a multi-objective search problem to minimize the magnitude and extent of the modifications, encourage proximity to the data manifold, and satisfy user-provided multi-modal requirements.
2) We demonstrate that MCD effectively suggests targeted design modifications to improve the functional performance of query designs, illustrating that counterfactual search could be viewed as an effective design recommendation tool.
3) We present the first text and image-based counterfactual search in design using the Contrastive Language-Image Pre-training (CLIP) method. These cross-modal queries were previously not possible with existing counterfactual methods.
4) We demonstrate that MCD can effectively handle multimodal queries, including a mixed-variable text, image, and parametric query, the first example of multimodal queries to a counterfactual search model, to our knowledge.

[1]A search for the term "counterfactual explanations" on the entire ASME digital collection, that includes design venues such as the IDETC conference and the Journal of Mechanical Design, returns zero results on March 10, 2023.

II. Background

Counterfactuals are a useful tool for investigating causality and forecasting the potential outcomes of different actions. Counterfactuals have been extensively used in various fields, including psychology, philosophy, social sciences, and machine learning as they offer a valuable tool for examining causality and understanding the consequences of actions [1]. In psychology, counterfactual thinking has been studied in relation to emotions, such as regret and disappointment. In philosophy, counterfactuals have been used to explore questions of determinism and free will. In social sciences, counterfactual analysis is widely used to evaluate the impact of policies and interventions. Counterfactual explanations are also gaining traction in the field of machine learning as a means to improve the interpretability and fairness of machine learning models.

In this literature review, we discuss three key areas that relate closely to our work — 1) explainability and counterfactuals in machine learning, 2) multi-objective optimization approaches to counterfactuals, and, 3) a multi-modal, zero-shot machine learning model that enables us to capture user requirements.

A. Explainability and Counterfactuals in Machine Learning

Counterfactual explanations are frequently used as a machine learning explainability tool. In machine learning, particularly deep learning, predictions are often mysterious and intractable. To remedy this intractability, a wealth of machine learning 'explainability' tools have been proposed in recent years. One common approach involves determining the sensitivity of the output with respect to the various input parameters (features), a technique known as 'feature importance.' Some popular methods in this category include Local Interpretable Model-Agnostic Explanations (LIME) [2] and Shapley Additive Explanations (SHAP) [3]. In the design automation community, these methods are often used to determine which design parameters have outsized impacts on design performance [4]–[7] or which parameters are important for relationships between products [8]. Another common approach to explainability involves visualizing a model's decisions in some way. This technique lends itself well to data modalities that are easily appreciated visually, such as images, for which saliency maps are a common explainability method [9]–[11].

Counterfactuals were first proposed for machine learning (ML) explainability by Wachter *et al.* [12]. Since then, researchers have proposed a wealth of counterfactual explanation approaches, which Verma *et al.* [1] and Guidotti *et al.* [13] review. Among the popular methods are Diverse Counterfactual Explanations (DiCE) [14], Feasible and Actionable Counterfactual Explanations (FACE) [15], and Multi-Objective Counterfactuals (MOC) [16]. Counterfactuals make for a great explainability tool since they allow users to intuitively understand the ML model's internal decision thresholds (i.e. "Where does my model start predicting a different outcome?"). Much like LIME and SHAP, many counterfactuals take a localized approach, with some even fitting local approximations to the data manifold to guide their explanations [17]. Counterfactuals have most commonly been proposed for tabular data, but have also been applied to images [18] and text [19], among other modalities. In general, good counterfactual explanations should typically demonstrate the following properties:

1) **Validity:** First and foremost, a good counterfactual explanation should result in the desired outcome. Depending on the nature of the problem, this desired outcome may be a class, an inequality, a range, an exact equality, or some combination of the above. For example, if we are querying a model that predicts the mass of a design and we specify a range of 2-3 kg, a proposed counterfactual should have a predicted mass in this range.
2) **Sparsity:** Good counterfactuals should be easy to realize, meaning that they should not change many features of the query. Sparsity refers to the number of features that must be modified to realize a counterfactual.
3) **Proximity:** While the number of modifications needed to realize a counterfactual is an important consideration, the extent of these modifications is also important. In simple terms, we would like counterfactuals to be as similar to the query as possible. This is typically quantified as a distance to the original query.
4) **Manifold Proximity:** In the classic usage of counterfactuals as an ML explainer, a predictive model has been trained on a dataset and is being iteratively queried by the counterfactual model. If queries lie too far from the data manifold on which the predictor was trained, predictions (and by extension counterfactuals) will no longer be accurate. In other use cases where the counterfactual is not explaining a statistical model, manifold proximity may not be desirable.
5) **Actionability:** In many problems, certain input parameters may not be changeable, but will nonetheless play a role in the output of the model. For example, the weight of the rider will play a significant role in the structural loading of a bicycle. However, when designing a bicycle, we can't choose to simply make the rider lighter. A good counterfactual explanation should only modify actionable features. Several works, such as [15], have also proposed more nuanced methods to handle actionability.
6) **Causality:** Features in a dataset may be causally linked, implying that changing one feature may necessitate changing another. In general, establishing causality is difficult. However, in design, we may be aware of causal relationships thanks to our fundamental understanding of the physics relating various input variables. For example, selecting a denser material for a given design may necessitate increasing the weight, provided that the geometry remains unchanged. This has clear ramifications for effective counterfactuals, which should ideally capture and respect any known causal relations in the problem.

A strong counterfactual method should undoubtedly generate high-quality counterfactuals. However, good counterfactual methods should also exhibit several properties that may not be reflected in the strength of individual counterfactual examples themselves:

1) **Diverse Sets:** As emphasized in [14], it may be highly desirable to generate diverse sets of counterfactuals. This gives the user a wealth of options, ideally with different actionable requirements to achieve the query objective.
2) **Model-Agnosticism:** Ideally, the algorithms used for the generation of counterfactual explanations should treat the model as a black box and interact with the model only through its predict function [1]. These "model-agnostic" algorithms allow for wider applicability and code reuse. Notably, model-agnostic approaches do not rely on gradient information from the predictor but may be less sample-efficient than methods that leverage gradients, when available.

Researchers have also adopted counterfactuals in a recommender-system setting. Tran *et al.* [20] review the use of counterfactual explanations in recommendation systems and propose a method to generate counterfactuals for recommender systems. While related, this work differs slightly from our proposed use case, in which the counterfactual-generating model is the recommender. Regenwetter [21] also briefly investigates leveraging Diverse Counterfactual Explanations [14] for design recommendations, citing challenges due to the limitations of single-objective queries.

B. Multi-Objective Counterfactual Explanations

Counterfactual explanations can be viewed as an optimization problem, and can similarly be implemented using an optimization algorithm. Many methods summarize the optimization objective as a weighted sum of the different objectives discussed earlier. However, another approach instead frames the counterfactual search process as a classic multi-objective optimization problem. Dandl *et al.* [16] were the first to formalize this parallel between counterfactual explanations and multi-objective optimization (MOO) in Multi-Objective Counterfactuals (MOC). By handling objectives individually rather than as a single aggregated objective, MOC realizes a key benefit of Multi-Objective Optimization, namely the ability to generate non-dominated sets of counterfactual explanations. Whereas a single-objective approach returns a counterfactual that optimizes for a statically weighted aggregation of objectives, the non-dominated set allows designers to adaptively select counterfactuals based on their specific search priorities, which typically depend on the problem at hand.

Multi-Objective Counterfactuals (MOC) [16] is a primary inspiration for Multi-Objective Counterfactuals in Design (MCD). However, we have expanded on MCD in several key directions. Chiefly, despite its name, MOC does not inherently support multi-objective queries. Furthermore, MOC does not distinguish between hard and soft constraints, despite the fact that this functionality is ingrained in the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [22] that MOC is built around. MCD addresses these gaps while also decoupling the optimization and sampling steps, and introducing new ways to integrate counterfactuals with a multi-modal, zero-shot machine learning model.

Since the overarching goal of MCD is not to explain predictors, but rather to search for design recommendation counterfactuals, we refer to the problem as 'counterfactual search.' Note that unlike counterfactual explanations, counterfactual search does not require ML predictors and can work with many types of forward models. It also has the additional goals of manifold similarity and meeting multi-objective multi-modal requirements.

C. Cross-Modal Design Recommendations

The multitude of data modalities spanned by design data remains a prominent challenge in data-driven design [23]–[25]. Though a model explained by a counterfactual method may make predictions in one modality, users may instead prefer to query targets in an entirely different modality. We will demonstrate in this paper that MCD can be used in conjunction with rendering pipelines and trained language models to generate counterfactuals for a parametric model using images or even text prompts. In this way, counterfactuals can capture complex and abstract user requirements in a 'zero-shot' fashion, requiring no additional training to understand the context of the prompts. To provide context for this discussion, we will introduce a brief background on relevant subjects in cross-modal learning.

When handling data of modalities like graphs [26], images [27], 3D geometry [28], text [29], [30], and mixed modalities, a common general technique involves mapping datapoints to a vector space. This effectively creates a link from datapoints of the modality to datapoints in the vector space. Two or more modalities can then be linked by creating shared embeddings for the modalities using the same vector space.

Shared text-image embeddings are an example of cross-modal embeddings that have garnered significant attention in recent years [31]. Radford *et al.* [32] propose one of the most widely used models for text-image shared embeddings called Contrastive Language-Image Pretraining (CLIP). CLIP trains a text embedding and image embedding model simultaneously on a dataset of text-image pairs. The models are rewarded for mapping matching pairs to similar embedding vectors and mapping non-matching pairs to dissimilar embedding vectors. In our second and third case studies, we will be leveraging pre-trained CLIP models to query counterfactuals using text prompts. Next, we move on to discuss our methodology.

III. Methodology

In this section, we discuss the construction of the optimization algorithm behind MCD, emphasizing the constraints, objectives, and operators used. We then present our approach for sampling diverse sets of counterfactuals and discuss how we decouple the optimization from the final sampling step. Finally, we demonstrate the capabilities of MCD on a simple 2D problem.

A. Objectives

Optimization algorithms typically seek to find constraint-satisfying solutions that achieve optimal objective scores. We will first discuss how objectives are defined in MCD, then go on to discuss constraints. Broadly, we consider two types of objectives: Objectives related to counterfactual quality and user-specified auxiliary objectives (often used for soft constraints). The former draw on the work of Dandl *et al.* [16], who, among other things, leverage Gower distance [33] and the number of changed features as optimization objectives in MOC.

1) **Gower Distance:** Gower Distance [33] is a metric that indicates the distance between mixed feature data points. Its use as an objective tackles the issue of "proximity" introduced in Sec. II-A. The Gower distance between d-dimensional counterfactual p and query q is given in terms of their feature values p_i and q_i for $i \in [1...d]$, as:

$$f_{pr}(p, q) = \frac{1}{d} \sum_{i=1}^{d} \delta_G(p_i, q_i) \tag{1}$$

$\delta_G(p_i, q_i)$ is a function that depends on feature type and is given as:

$$\delta_G(p,q) = \begin{cases} \frac{1}{\hat{R}_i}|p_i - q_i| & \text{if } p_i \text{ is numerical} \\ \mathbb{1}_{p_i \neq q_i} & \text{if } p_i \text{ is categorical} \end{cases} \quad (2)$$

Here, $\hat{R}_i$ is the range of the feature i observed in the dataset.

2) **Changed Feature Ratio:** This objective calculates the proportion of features that the proposed counterfactual, p, modifies from the query, q. Its use as an objective tackles the issue of "sparsity" introduced in Sec. II-A.

$$f_{sp}(p,q) = \frac{||p-q||_0}{d} = \frac{1}{d}\sum_{i=1}^{d} \mathbb{1}_{p_i \neq q_i} \quad (3)$$

3) **Average Gower Distance:** To measure the "manifold proximity" discussed in Sec. II-A, Dandl *et al.* [16] calculate the average Gower distance to the k nearest observed data points $s^i ... s^k$ from the dataset S:

$$f_{mp}(p,S) = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{d}\sum_{j=1}^{d}\delta_G(p_j, s_j^i) \quad (4)$$

4) **Problem-Specific Objectives:** Just as the user may specify non-negotiable requirements for the model outcome (hard constraints), they may also specify objectives $(f_1(p)...f_M(p))$ that they would like to satisfy, and later specify targets for these objectives during sampling. These auxiliary objectives are directly included as optimization objectives in NSGA-II.

B. Constraints

In a counterfactual search, a variety of optimization constraints may be present. Constraints are considered non-negotiable and always take precedence over objectives. In practice, many optimization algorithms, including the variant of the NSGA-II algorithm driving MCD, prioritize resolving constraint violations before proceeding to the optimization of objectives. MCD considers several types of constraints:

1) **Variable and Constant Features:** Like many counterfactual models, we implement a mechanism to constrain which features are allowed to be modified by a counterfactual, as specified by the user. This addresses the challenge of "actionability" introduced in Sec. II-A. We call the set of actionable features A.
2) **Model Output Constraints:** Users querying a counterfactual method may have requirements for the output of their model. In most counterfactual search approaches, these requirements are treated as non-negotiable hard constraints to satisfy the "validity" property introduced in Sec. II-A. MCD supports such hard requirements, which are handled as constraints in NSGA-II[2], but does not require them. We consider any output with a constraint as belonging to a set B and require that $L_b \leq f_b(p) \leq U_b \forall b \in B$. Instead, we also allow users to specify soft constraints in the form of additional optimization objectives, paired with targets to be used during sampling.
3) **Domain-Specific Constraint Functions:** There are cases in which certain hard constraints are known a priori. MCD can be configured to respect such hard constraints through user-specified black-box constraint functions. Domain-specific constraints can be used for a variety of different purposes, including encoding causality relations into the optimization as discussed in Sec. II-A. We specify these constraint functions as $g_1(p)...g_K(p)$ and, for simplicity, assume they are satisfied for $g_k(p) \geq 0$

[2]By default, we expect queries in the form of inequalities. Since range and equality constraints (or objectives) can be specified using two inequalities, we find this to be an adequately versatile interface for most types of constraints. In rare cases where users need to specify complex constraints, such as disjoint ranges, they can do so by creating a custom constraint function and passing it in as a black box.

C. Formulation as MOO problem

In summary, we express the multi-objective optimization problem in terms of the variables, sets, and functions defined above as follows:

$$\text{minimize: } f_i(p),\ \forall\ i \in \{pr, sp, mp, 1, ..., M\} \quad (5)$$

$$\text{subject to: } f_j(p) - L_j \geq 0,\ U_j - f_j(p) \geq 0,\ \forall\ j \in B,$$

$$g_k(p) \geq 0,\ \forall\ k \in \{1, ..., K\},$$

$$p_l = q_l,\ \forall\ l \notin A$$

D. Algorithm

Any gradient- or non-gradient-based multi-objective optimization method could be used in MCD. To demonstrate our results in this paper, we leverage the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [22] as the backend of MCD. NSGA-II is a multi-objective genetic algorithm that boasts several innovative features, such as non-dominated sorting for elitist selection, crowding distance to encourage diversity, and genetic operators such as tournament selection, simulated binary crossover, and polynomial mutation. We use an implementation of NSGA-II from [34], including the mixed-variable selection, crossover, and mutation functions provided.

The initial population always consists of the query and a set of randomly sampled points from the dataset or the user-specified design space boundaries. In problems with continuous variables, we find that without any precautions to maintain the exact parameter values from the original query, these values tend to get 'lost,' and can never be exactly reconstructed, hurting the sparsity objective of counterfactuals. To allow the algorithm to 'rediscover' the exact parameter values from the query, we introduce a custom operator that randomly reverts individual parameter values back to the query's values with a certain probability.

E. Sampling

Contrary to other counterfactual search approaches, our method decouples the optimization and sampling steps. Conventionally, a user will have to decide on the priorities between various objectives (e.g. proximity, diversity, manifold proximity, etc.) before running the optimization. This is impractical, as these objectives are challenging to select intuitively, and must often be chosen through trial and error. For example, a designer might realize that the generated counterfactuals are much too different from the query to be practically realizable. By avoiding retraining, our method can save significant computational expense and, as we will discuss in Sec. V, enable users to quickly consider counterfactuals from different regions of the objective landscape. We decouple the search and sampling process as follows:

1) Given a query, a set of constraints, and objectives, the optimizer generates a collection of candidate counterfactuals by running NSGA-II.
2) The sampling algorithm collects a set of objective priority weights and optional targets from the user. By collecting these weights after training, MCD allows rapid counterfactual sampling under different objective weights without the need for retraining, unlike other approaches.
3) Each candidate counterfactual is assigned an aggregate quality score, which is calculated as a sum of individual objective scores, weighted by their priority. For any objectives with specified targets, the Design Target Achievement Index [35] is used to quantify target achievement before factoring into the aggregate score. The aggregate score, S of a counterfactual candidate, p, is given in terms of objective priority weights w_{pr}, w_{sp}, w_{mp} by:

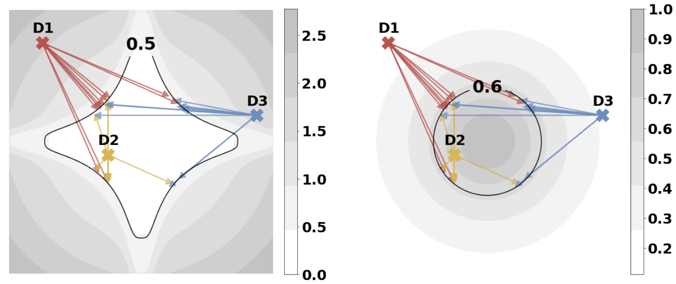
$$S(p) = w_{pr} f_{pr}(p,q) + w_{sp} f_{sp}(p,q) + w_{mp} f_{mp}(p,S) + DTAI(p,t,\alpha,\beta) \tag{6}$$

   Here, $DTAI(p,t,\alpha,\beta)$ is the Design Target Achievement Index of the candidate given auxiliary objective targets, t, priority weights, α, and decay parameters, β [35].
4) A performance-weighted diversity matrix is calculated using a Gower distance-based similarity kernel to evaluate the similarity between counterfactuals. Matrix entries are calculated as a function of aggregate scores and a diversity parameter, w_d as:

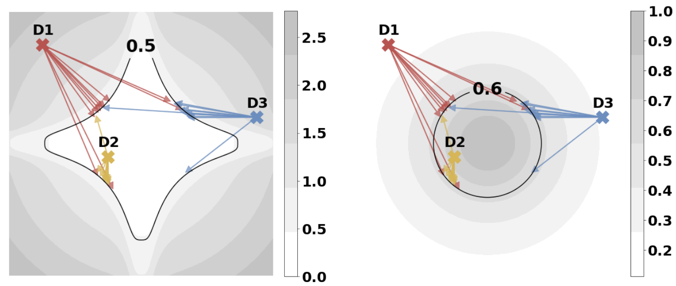
$$D_{i,j} = \delta_G(i,j)\,(S(i)S(j))^{\frac{1}{w_d}} \tag{7}$$

5) A diverse set of high-performing counterfactuals is sampled from this matrix using k-greedy diverse sampling [36].

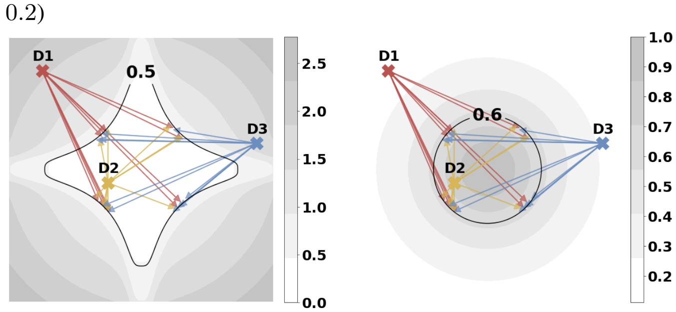
If the user requests only a single counterfactual instead of a diverse set, the candidate with the highest aggregate quality score is returned.



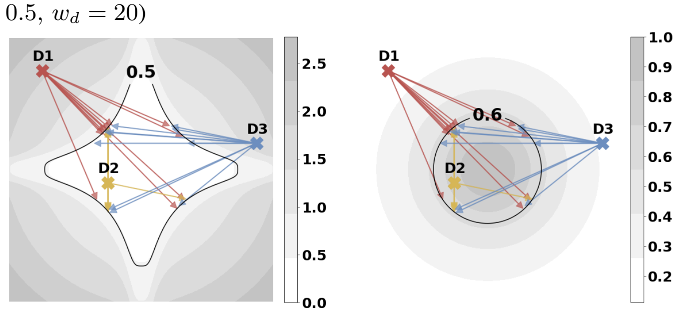
(a) Balanced Sampling ($w_{pr} = 0.5$, $w_{sp} = 0.2$, $w_{mp} = 0.5$, $w_d = 0.2$)



(b) High Prox. Weight ($w_{pr} = 50$, $w_{sp} = 0.2$, $w_{mp} = 0.5$, $w_d = 0.2$)



(c) High Diversity Weight ($w_{pr} = 0.5$, $w_{sp} = 0.2$, $w_{mp} = 0.5$, $w_d = 20$)



(d) High Sparsity Weight ($w_{pr} = 0.5$, $w_{sp} = 20$, $w_{mp} = 0.5$, $w_d = 0.2$)

Fig. 2: Counterfactual sets returned for three query designs under different weightings of counterfactual quality objectives. Performance space constraints are indicated on the plots. Valid counterfactuals must simultaneously meet both constraints.

F. Showcasing Functionality on 2D Examples

Before showcasing the capabilities of MCD on real design datasets, we will first demonstrate its performance on a simple two-dimensional problem for ease of visualization. We select a challenging two-objective problem and sample synthetic data. We then query three different designs, D1-3, and specify the same challenging constraint criterion for each query, which is only satisfiable in four small disjoint regions of the space. Mathematically, we constrain the performance space values Y_1 and Y_2 such that $0.4 \leq Y_1 \leq 0.6$ and $Y_2 \geq 0.6$. In simple terms, any valid counterfactual must lie near the star-shaped contour on the left and strictly within the circle on the right in the contour plots in Fig. 2. We consider four choices of objective weights:

1) First we examine a fairly "balanced" selection of objective weights ($w_{pr} = 0.5$, $w_{sp} = 0.2$, $w_{mp} = 0.5$, $w_d = 0.2$) in Fig. 2a. In this setting, the sampled counterfactual sets achieve a balance of proximity, diversity, and sparsity.
2) Next, we consider a setting where proximity is prioritized over other objectives ($w_{pr} = 50, w_{sp} = 0.2$, $w_{mp} = .05$, $w_d = 0.2$) in Fig. 2b. In this setting, most counterfactuals in each set are sampled from the mode nearest the queries, though counterfactuals are still diversified within these modes.
3) We next consider a case where diversity is given precedence over other objectives ($w_{pr} = 0.5, w_{sp} = 0.2$, $w_{mp} = 0.5$, $w_d = 20$) in Fig. 2c. In this case, the sampled counterfactual sets are very well distributed across the feasible regions of the space.
4) Finally, we consider the case where sparsity is given the highest priority ($w_{pr} = 0.5, w_{sp} = 20, w_{mp} = 0.5$, $w_d = 0.2$) in Fig. 2d. Many sampled counterfactuals change only one parameter from the query, when possible.

Each of these subsets is sampled from the same set of counterfactual candidates with no re-optimization necessary. Now, having demonstrated MCD's functionality on a simple 2D problem, we move on to a more complex real-world design problem: Bike frame design.

IV. Case Study 1: Design Refinement using Structural Performance Queries

In our first case study, we consider the counterfactual: "What if my design were 30% lighter?" Specifically, we consider a bicycle frame design problem where we are trying to improve the structural properties and reduce the weight of a query design. We use a regression model trained on the FRAMED dataset consisting of Finite Element (FE) simulation results from 4500 community-designed bike frames [**?**], including weight, safety factors, and deflections under various loading conditions. The trained regression model is an AutoGluon tabular AutoML regressor [37] intended to predict the structural performance of bicycle frames accurately.

To illustrate MCD's capabilities, we feed it three variants of the same query. The first has a single objective: finding counterfactuals that reduce the predicted mass of a given design. The second has two competing objectives: Maximize a design's safety factor while minimizing its mass. The third has the same objectives as the second but restricts MCD to only vary a more constrained and actionable set of features. In each example, we query the same design: a steel tube road bike with minor structural inefficiencies. These inefficiencies largely stem from a down tube with insufficient wall thickness, requiring other components to be over-engineered. This bike has a safety factor[3] of 1.24 and a mass of 4.26 kg, so our primary objective is to reduce the mass. Each optimization ran for 100 generations with a population size of 500.

a) Single objective query: In the first variant, MCD was tasked with finding counterfactuals that reduced the mass of the original design from 4.26 kg to under 3 kg. MCD effectively successfully discovered hundreds of valid counterfactuals and sampled a set of five diverse counterfactuals which had, on average, a mass of 2.3 kg, as tabulated in Table I. Although MCD succeeded in its explicitly stated objective, a closer look reveals that it did nothing to remedy the wall thickness issue in the down tube, and as a consequence of weight savings in other parts of the sampled frames, the average safety factor across sampled counterfactuals was an abysmal 0.48. This disregard for secondary objectives is quite characteristic of the many existing single-objective counterfactual search algorithms and illustrates why MCD's novel support of multi-objective queries is so essential for design problems. Our next example showcases how to leverage multi-objective queries to avoid these issues.

TABLE I: Generated counterfactuals for variant 1 (34 columns omitted). Like many single-objective counterfactual engines, MCD tends to achieve single-objective queries at the expense of secondary objectives. MCD's unique support of multi-objective queries remedies this problem.

	Material	Stack (mm)	...	Down Tube Thick. (mm)	Safety Factor	Frame Mass (kg)
Query	Steel	565.6	...	0.52	1.24	4.26
CF 1	Steel	570.8	...	0.52	0.52	1.99
CF 2	Steel	565.6	...	0.52	0.27	1.64
CF 3	Steel	565.6	...	0.52	0.76	2.48
CF 4	Steel	565.6	...	0.52	0.64	2.69
CF 5	Aluminum	522.6	...	0.52	0.22	2.70

b) Bi-objective query: In the second variant, a second objective was introduced: Increase the safety factor to a minimum value of 1.5. Again, MCD successfully discovered numerous counterfactuals, and the diverse 5-bike set sampled this time had an average mass of 2.4 kg and a safety factor of 1.7, as shown in Table II. This time, MCD realized that the bike could be made significantly more weight-efficient by increasing the down tube wall thickness to relieve structural stress on other components to be lightened. However, it also

[3]We use predicted safety factor in FRAMED's in-plane loading scenario [**?**]

changed the material of the bike from steel to aluminum or titanium in four of the five counterfactuals, a modification that would likely carry a significant increase to the cost and may thus be unactionable. In the presence of a cost prediction model, MCD could consider cost as another query objective. However, even without such a model, MCD can be ordered to leave certain design parameters unchanged, as we demonstrate in our final example.

TABLE II: Generated counterfactuals for variant 2 (34 columns omitted). By querying multiple objectives simultaneously, MCD avoided the safety factor issue that occurred in Query 1.

	Material	Stack (mm)	...	Down Tube Thick. (mm)	Safety Factor	Frame Mass (kg)
Query	Steel	565.6	...	0.52	1.24	4.26
CF 1	Aluminum	565.0	...	2.20	1.91	2.81
CF 2	Titanium	561.6	...	2.46	1.82	2.21
CF 3	Aluminum	532.2	...	1.81	1.58	1.75
CF 4	Titanium	563.5	...	3.92	1.60	2.23
CF 5	Steel	565.6	...	2.48	1.65	2.87

c) Bi-objective query with constraints: In the third variant, MCD was no longer allowed to vary frame material. It proceeded to find tens of valid designs through variations in certain tube diameters, lengths, and other structural configurations. From these valid designs, a 5-bike set was sampled that had an average mass of 2.5 kg and an average safety factor of 1.8, as shown in Table III.

TABLE III: Generated Counterfactuals for Query 3 (34 columns omitted). When restricted from modifying frame material, MCD is still able to recommend design modifications that meet the safety factor and mass targets.

	Material	Stack (mm)	...	Down Tube Thick. (mm)	Safety Factor	Frame Mass (kg)
Query	Steel	565.6	...	0.52	1.24	4.26
CF 1	Steel	565.6	...	2.44	2.05	2.93
CF 2	Steel	601.7	...	3.38	2.06	2.31
CF 3	Steel	565.6	...	3.22	1.58	2.71
CF 4	Steel	601.7	...	2.12	1.61	1.87
CF 5	Steel	565.6	...	3.35	1.56	2.82

Through these examples, we have attempted to demonstrate that MCD excels at handling multi-objective performance queries and can be used in such a setting to recommend performance-enhancing design modifications. In our next example, we consider a scenario in which more abstract text queries are provided instead of hard performance constraints.

V. Case Study 2: Modifying Designs using Cross-Modal Text Queries

In this case study we examine subjective counterfactuals like: "What if my design looked more 'cyberpunk themed?"' Classically, counterfactual search requires a query in the same data modality as the predictive model. This can be constraining, since it may be more natural in many cases to place queries in a different data modality, especially if that modality is more intuitive for human users. This is often the case for images or text, which are much more easily understood by humans compared to tabular or parametric data. Accordingly, we demonstrate how we can query MCD in a cross-modal setting using text prompts.

A. Methodology: Case Study 2

To enable cross-model queries, we construct an objective evaluation function comprised of several key building blocks:

- To begin, we require a rendered image of a bicycle design. We construct an automated rendering pipeline that works in conjunction with the BikeCAD software to generate an image of a bicycle given a parametric vector.
- We then calculate an embedding for the generated bike image using a pre-trained CLIP model introduced in Sec. II-C that maps the generated bike renders to a vector embedding space.
- Next, we compute the embedding vector for a target text prompt using a pre-trained CLIP text embedding model.
- Finally, we calculate cosine similarity between the two 512-dimensional embedding vectors.

In this case study, this entire objective evaluation pipeline serves as the predictor for counterfactual search. By generating counterfactuals that minimize this cosine similarity objective, the optimizer ensures that generated counterfactuals better match the given text prompts.

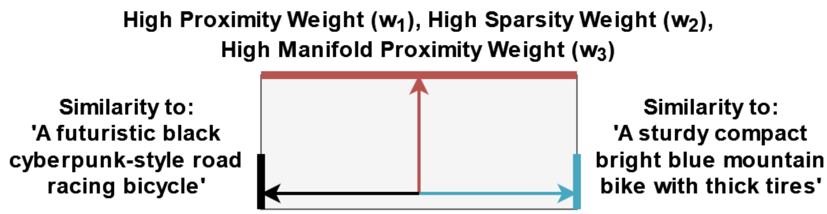
We select a subset of the BIKED [5] dataset's parameter space to consider during optimization and choose a generic red road bike design as a query design. We choose two text prompts as optimization objectives: "A futuristic black cyberpunk-style road racing bicycle" and "A sturdy compact bright blue mountain bike with thick tires." Because the demands of human designers are often difficult to quantify using traditional parametric methods, the first text prompt was selected to be highly subjective. The second prompt is less subjective, offering details about design features, but stops short of explicit design guidelines. In this context, the user is effectively asking questions like: "How would my red road bike design change if I wanted it to look more like a black cyberpunk-style bike?" We optimize for 400 generations with a population size of 100. Next, we perform a series of sampling operations with different objective weights, as shown in II-C. By selecting the optimal bikes at a sweep of different objective weights, we can visualize the best bikes under numerous configurations of objective priorities.

Counterfactual quality objective weights in the i^{th} row were chosen as:

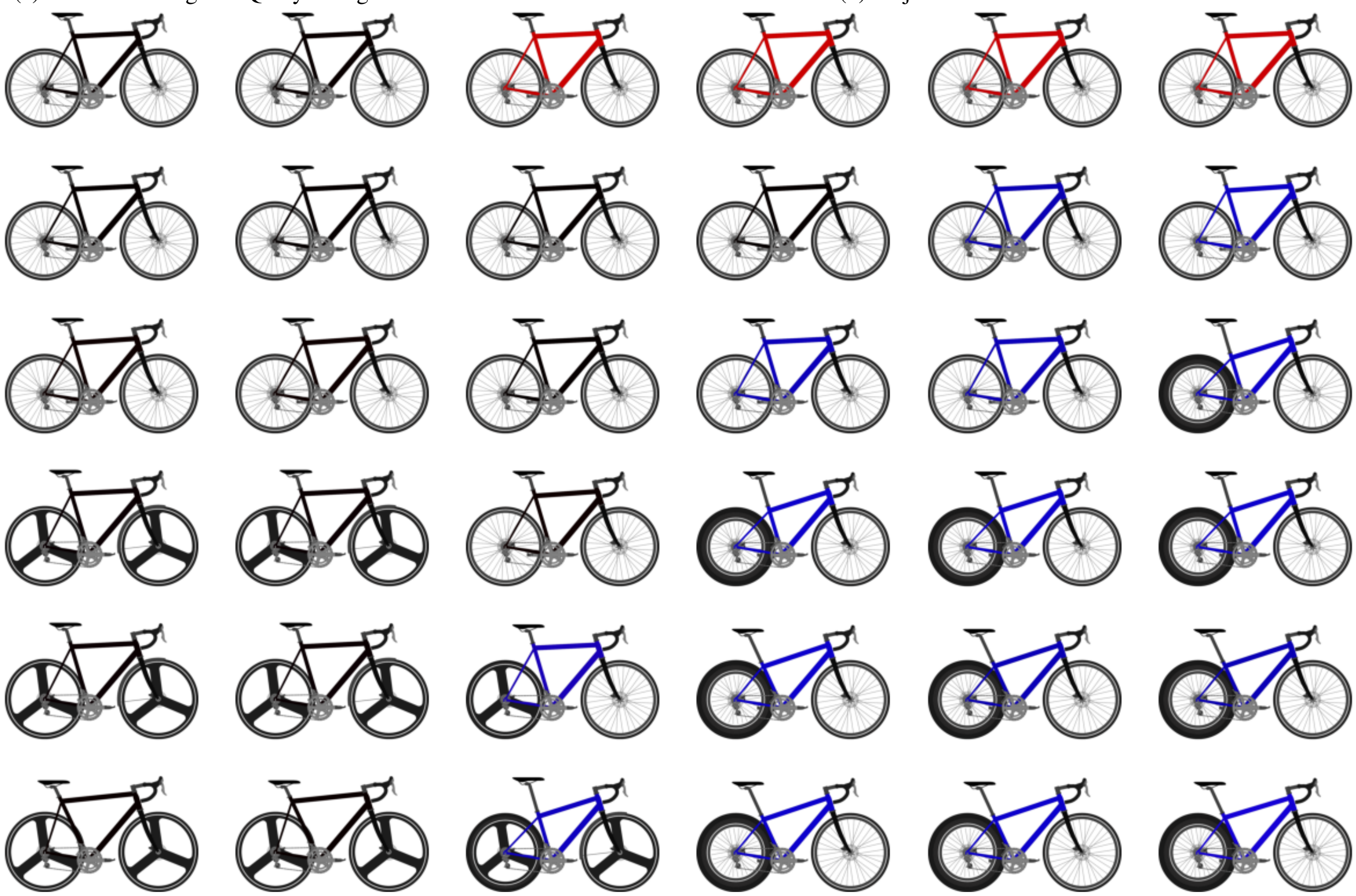
$$w_1 = w_2 = w_3 = \frac{0.2}{2^i} \tag{8}$$

In this way, counterfactuals with better proximity, sparsity, and manifold proximity were prioritized toward the top of the grid, while counterfactuals were given more leeway to deviate

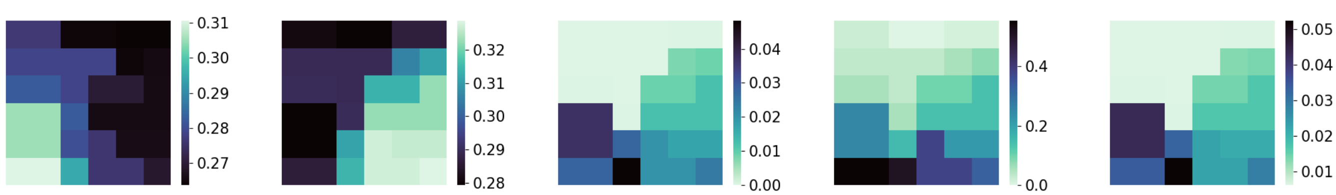
(a) Rendered Image of Query Design



(b) Objective Priorities



(c) Map of Rendered Bicycle Images Corresponding to Generated Counterfactuals



(d) Similarity to: 'A futuristic black cyberpunk-style road racing bicycle'

(e) Similarity to: 'A sturdy compact bright blue mountain bike with thick tires'

(f) Counterfactual Proximity (Gower Distance)

(g) Counterfactual Sparsity (Changed Feature Ratio)

(h) Counterfactual Manifold Proximity (Average Gower Distance)

Fig. 3: Visualization of the objective manifold for cross-modal counterfactual selection. Designs sampled from the top of the manifold prioritize proximity, sparsity, and manifold proximity. Designs in the left and right corners prioritize similarity to two respective text prompts. Heatmaps show individual objective scores (lighter is better).

from the query design and data manifold toward the bottom. Diversity weight, w_d, was irrelevant, as only one design was sampled for each combination of objective weights.

Similarly, auxiliary objective weights in the j^{th} column were set through the DTAI objective weighting parameter, α, in terms of the number of columns, n (in our case 6), as:

$$\alpha_1 = 1.5^{n-j},\ \alpha_2 = 1.5^{j-1} \tag{9}$$

These objectives allowed similarity to the first text prompt to take precedence on the left edge of the grid and similarity to the second text prompt to take precedence on the right.

B. Discussion: Case Study 2

As expected, models at the top of the grid are appreciably similar to the red bike; some were essentially indistinguishable. Bikes further down the grid become progressively more visually different, which is corroborated by objective scores, as shown in Figs. 3f-3h.

Bikes in the lower left corner of the grid can be subjectively identified as more similar to "A futuristic black cyberpunk-style road racing bicycle." Among the key modifications are a color change and a shift to tri-spoke wheels, which may be more on-theme for a 'cyberpunk-style' bike. Similarity to the text prompt as evaluated by CLIP agrees, as shown in Fig. 3d.

Likewise, bikes towards the bottom right corner of the grid can be subjectively identified as more similar to "A sturdy compact bright blue mountain bike with thick tires." Bikes in this corner have the slanted down tube which is characteristic of mountain bikes; have the requested color change; and have a thick rear tire. Notably, the model either does not discover a modification to the front tire or does not find that such a modification improves similarity to the prompt. Also, the models maintain the dropped handlebars present on the query, which are characteristic of road bikes. Nevertheless, similarity to the text prompt as evaluated by CLIP was found to be best in this corner, as shown in Fig. 3e.

In this case study we demonstrated that MCD effectively handles multi-objective cross-modal prompts. Next, we move on to consider a challenging multi-modal query case as our final case study.

VI. Case Study 3: Modifying Designs using Multimodal Text, Image, and Parametric Queries

In this final case study, we examine hybrid counterfactuals like: "What if my design were lighter, looked more 'cyberpunk themed,' had better structural properties, and looked like this other design?" Having considered multi-objective cross-modal queries in the previous case study, we now present our most challenging case study. This time, we provide a multi-objective multi-modal query consisting of a target text prompt, image, frame safety factor, and frame mass.

A. Methodology: Case Study 3

To calculate image and text similarity, we leverage the rendering pipeline and pre-trained CLIP model used in case study 2. To calculate structural performance, we use the AutoGluon [37] model trained in [**?**], which was used in case study 1. We again select the same generic red road bike as our query design and select "A futuristic black cyberpunk-style road racing bicycle" as our text prompt. For our target image, we select an image of a Fuji Wendigo 1.1 mountain bike which closely matches the second text description from the previous case study.

Like the last case study, we sample designs in a grid, as shown in Fig. 4 based on a variable objective weighting scheme. We select a spread of DTAI objective weighting parameter (α) values in terms of the i^{th} row and j^{th} column as follows:

$$\begin{gathered}\alpha_{\text{text}} = 2^{n-j},\ \alpha_{\text{image}} = 2^{j} \\ \alpha_{\text{sf}} = 1.5^{n-i-1},\ \alpha_{\text{mass}} = 1.5^{i}\end{gathered} \tag{10}$$

This time, we hold the counterfactual quality objective weights constant at:

$$w_1 = w_2 = w_3 = 0.05 \tag{11}$$

B. Discussion: Case Study 3

As in the previous case study, MCD modifies several components to better match the text query of the models on the left side, including recoloring them black and replacing a regular spoked wheel with a disk wheel. However, due to the proximity, sparsity, and manifold proximity weights being fixed at moderate values, it does not deviate as far from the dataset as some of the most extreme designs in the previous case study.

The bikes on the right side of the grid are visibly more similar to mountain bikes, displaying the characteristic slanted top tube and, in some cases, adding a front suspension to the design. Interestingly, MCD does not generate any blue bikes, indicating that the color of the reference image is not as strongly emphasized as when a color is explicitly stated in a text prompt.

Structural modifications of the bike frame are challenging to appreciate in renderings because the largest drivers of structural performance are tube wall thickness parameters and material, none of which have a visual signature in the rendering. However, Figs. 4g and 4h indicate that bikes at the top and bottom prioritize safety factor and weight, respectively, as intended.

From Fig. 4i, we can see that the bikes toward the top of the grid fall far outside of the data manifold. Unsurprisingly, we see various design infeasibilities in these bikes, such as colliding components. Unless explicitly prevented using constraints, such infeasibilities are typically more common as counterfactuals fall further from the data manifold.

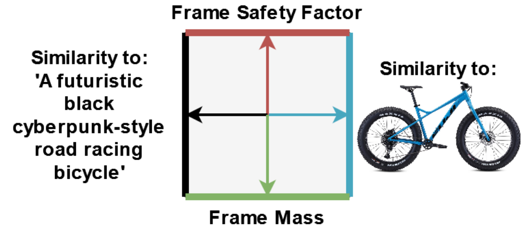
Though some of the generated counterfactuals suffer from infeasibilities, we have demonstrated that MCD can provide meaningful counterfactuals in high-dimensional (i.e., 4 auxiliary objectives and 3 counterfactual quality objectives) and multimodal objective spaces. Next, we proceed to discuss MCD's limitations.

VII. Limitations

MCD makes several key contributions to counterfactual optimization methods for designers, such as incorporating

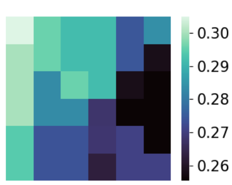
(a) Rendered Image of Query Design

(b) Target Mountain Bike Image

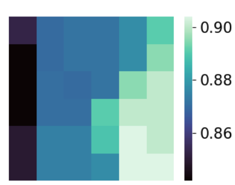


(c) Objective Priorities

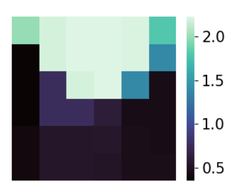
(d) Map of Rendered Bicycle Images Corresponding to Generated Counterfactuals



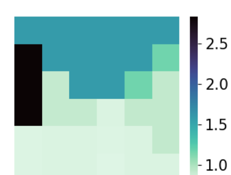
(e) Similarity to: ‘A futuristic black cyberpunk-style road racing bicycle’



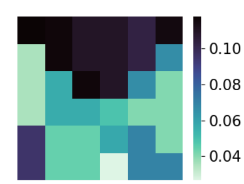
(f) Similarity to Target Mountain Bike Image



(g) Safety Factor



(h) Model Mass (kg)



(i) Counterfactual Manifold Proximity (Average Gower Distance)

Fig. 4: Visualization of the objective manifold for multimodal counterfactual selection. Designs sampled towards the top and bottom of the manifold prioritize safety factor and weight respectively. Designs sampled towards the left and right edges prioritize similarity to a target text prompt and target image, respectively. Heatmaps show individual objective scores (lighter is better). Designs that fall far outside of the data manifold struggle with component overlap and other infeasibility issues.

multiple objectives. However, it also has a few limitations. In model-agnostic configurations, MCD must use a gradient-free optimizer, preventing it from leveraging gradient information, even if some of the predictive models are differentiable. While this gradient-free approach allows MCD to support nondifferentiable predictors and avoid local minima, it potentially makes MCD less sample-efficient than similar gradient-based approaches.

Another key limitation stems from the difficulty of genetic algorithms in handling a large number of objectives. Because MCD adds three counterfactual quality objectives to the objective space, it slightly exacerbates the dimensionality issue of multi-objective genetic algorithms. Future work will explore MCD variants that leverage gradient information and many-objective optimization methods to address these limitations.

Additionally, we would like to acknowledge certain limitations with the text-based queries presented in the last two case studies. Though CLIP embeddings can capture more abstract and subjective ideas, they struggle to capture fine-grained technical details of designs. As such, we recommend that users with highly technical constraints specify them parametrically, instead of through text. However, as machine learning models continue to improve, querying counterfactual models for precise technical details through text and images may improve significantly.

VIII. Conclusion

In this paper, we have introduced Multi-Objective Counterfactuals for Design (MCD), a specialized counterfactual optimization method for design tasks. We first discussed previous counterfactual optimization approaches, many stemming from machine learning explainability literature. We then identified key limitations with existing works, particularly their inability to sample multi-objective queries and the inherent coupling of the optimization and sampling process. Next, we demonstrated using 2D examples how MCD solves these two challenges.

We presented a bicycle frame optimization problem and showed how MCD's support of multi-objective queries allows it to recommend meaningful modifications to a query design which improves structural performance. We then identified that although previous counterfactual search models have not supported cross-modal queries, advancements in multi-modal learning reasonably allow counterfactuals to be queried in different data modalities. Next, we showcased how MCD can be queried with text prompts, and illustrated how MCD's decoupling of optimization and sampling allows it to visualize complex objective manifolds without re-optimization. Finally, we asked MCD to generate counterfactuals given a multimodal text, image, and parameter query. By effectively recommending design modifications to match these queries, MCD demonstrated that it can support complex multimodal queries.

All in all, MCD is a valuable tool for designers looking to optimize their designs and design automation researchers looking to interact intuitively with their models. We are excited to release our code and examples at http://decode.mit.edu/projects/counterfactuals/ and anticipate a variety of interesting use cases across the community.

IX. Acknowledgments

We would like to thank Amin Heyrani Nobari for his contributions to the image rendering pipeline that enabled much of the cross-modal work presented. We would also like to thank Tyler Butler for his feedback and edits.

References

[1] S. Verma, V. Boonsanong, M. Hoang, K. E. Hines, J. P. Dickerson, and C. Shah, "Counterfactual explanations and algorithmic recourses for machine learning: A review," *arXiv preprint arXiv:2010.10596*, 2020.

[2] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[3] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[4] J. Joung and H. M. Kim, "Approach for importance–performance analysis of product attributes from online reviews," *Journal of Mechanical Design*, vol. 143, no. 8, 2021.

[5] L. Regenwetter, B. Curry, and F. Ahmed, "Biked: A dataset for computational bicycle design with machine learning benchmarks," *Journal of Mechanical Design*, vol. 144, no. 3, 2022.

[6] R. Rodríguez-Pérez and J. Bajorath, "Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions," *Journal of computer-aided molecular design*, vol. 34, pp. 1013–1026, 2020.

[7] A. Dachowicz, K. Mall, P. Balasubramani, A. Maheshwari, A. K. Raz, J. H. Panchal, and D. A. DeLaurentis, "Mission engineering and design using real-time strategy games: An explainable ai approach," *Journal of Mechanical Design*, vol. 144, no. 2, 2022.

[8] F. Ahmed, Y. Cui, Y. Fu, and W. Chen, "Product Competition Prediction in Engineering Design Using Graph Neural Networks," *ASME Open Journal of Engineering*, vol. 1, 05 2022, 011020. [Online]. Available: https://doi.org/10.1115/1.4054299

[9] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[10] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833.

[11] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134.

[12] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[13] R. Guidotti, "Counterfactual explanations and how to find them: literature review and benchmarking," *Data Mining and Knowledge Discovery*, pp. 1–55, 2022.

[14] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 607–617.

[15] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach, "Face: feasible and actionable counterfactual explanations," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 344–350.

[16] S. Dandl, C. Molnar, M. Binder, and B. Bischl, "Multi-objective counterfactual explanations," in *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I*. Springer, 2020, pp. 448–469.

[17] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini, "Factual and counterfactual explanations for black box decision making," *IEEE Intelligent Systems*, vol. 34, no. 6, pp. 14–23.

[18] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2376–2384.

[19] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata, "Generating counterfactual explanations with natural language," in *ICML Workshop on Human Interpretability in Machine Learning*, 2018, pp. 95–98.

[20] K. H. Tran, A. Ghazimatin, and R. Saha Roy, "Counterfactual explanations for neural recommenders," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1627–1631.

[21] L. Regenwetter, "Data-driven bicycle design using performance-aware deep generative models," Ph.D. dissertation, Massachusetts Institute of Technology, 2022.

[22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[23] L. Regenwetter, A. H. Nobari, and F. Ahmed, "Deep generative models in engineering design: A review," *Journal of Mechanical Design*, vol. 144, no. 7, p. 071704, 2022.

[24] B. Song, R. Zhou, and F. Ahmed, "Multi-modal machine learning in engineering design: A review and future directions," *arXiv preprint arXiv:2302.10909*, 2023.

[25] L. Regenwetter, A. Srivastava, D. Gutfreund, and F. Ahmed, "Beyond statistical similarity: Rethinking metrics for deep generative models in engineering design," *arXiv preprint arXiv:2302.02913*, 2023.

[26] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.

[27] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," *arXiv preprint arXiv:1707.05612*, 2017.

[28] G. Dai, J. Xie, and Y. Fang, "Siamese cnn-bilstm architecture for 3d shape representation learning." in *IJCAI*, 2018, pp. 670–676.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[30] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar *et al.*, "Universal sentence encoder," *arXiv preprint arXiv:1803.11175*, 2018.

[31] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[32] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[33] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, pp. 857–871, 1971.

[34] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.

[35] L. Regenwetter and F. Ahmed, "Design target achievement index: A differentiable metric to enhance deep generative models in multi-objective inverse design," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 86236. American Society of Mechanical Engineers, 2022, p. V03BT03A046.

[36] E. Celis, V. Keswani, D. Straszak, A. Deshpande, T. Kathuria, and N. Vishnoi, "Fair and diverse dpp-based data summarization," in *International Conference on Machine Learning*. PMLR, 2018, pp. 716–725.

[37] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data," *arXiv preprint arXiv:2003.06505*, 2020.