Pareto Gamuts: Exploring Optimal Designs Across Varying Contexts

LIANE MAKATURA, Massachusetts Institute of Technology
MINGHAO GUO, Chinese University of Hong Kong
ADRIANA SCHULZ, University of Washington
JUSTIN SOLOMON and WOJCIECH MATUSIK, Massachusetts Institute of Technology

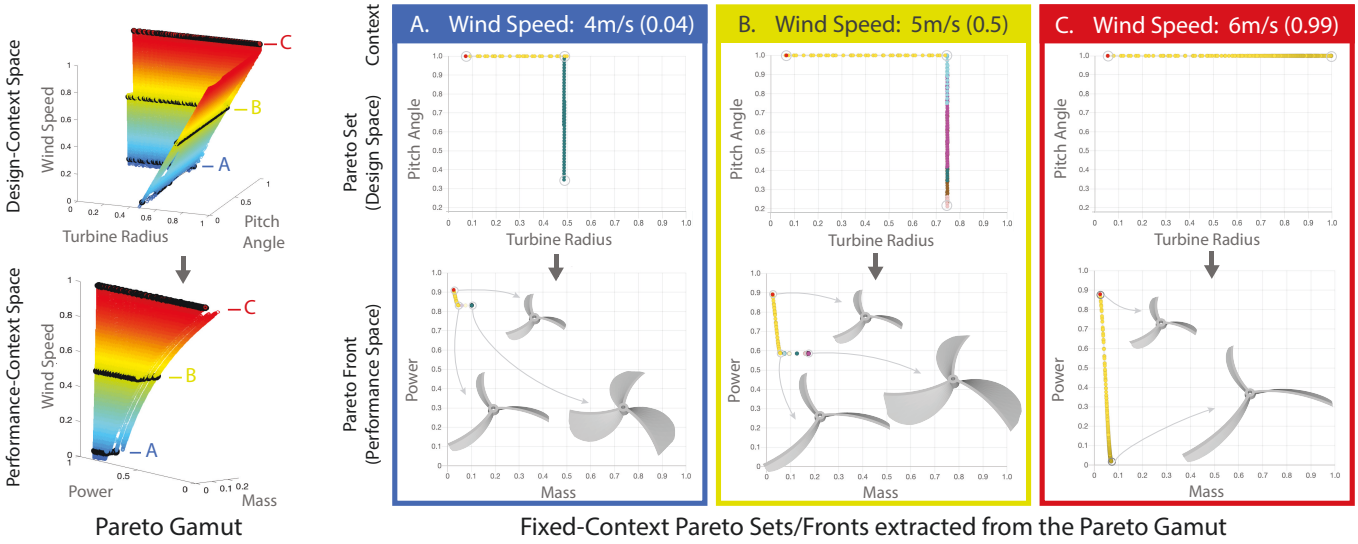


Fig. 1. Manufactured parts are usually optimized for many distinct contexts: here, turbine designs are optimized for mass and power (performance) under various wind speeds (context). Each context yields a distinct *Pareto set*, or collection of designs with optimal performance trade-offs. Existing tools require a separate optimization for each context of interest. As an alternative, we augment the standard multi-objective optimization framework to simultaneously consider design *and* context variables. This allows us to find the *Pareto gamut* (left, colored by context), which captures the Pareto-optimal designs over a range of contexts. From this gamut, we can extract any fixed-context Pareto set (right, top) and its image in performance space (right, bottom).

Manufactured parts are meticulously engineered to perform well with respect to several conflicting metrics, like weight, stress, and cost. The best achievable trade-offs reside on the *Pareto front*, which can be discovered via performance-driven optimization. The objectives that define this Pareto front often incorporate assumptions about the *context* in which a part will be used, including loading conditions, environmental influences, material properties, or regions that must be preserved to interface with a surrounding assembly. Existing multi-objective optimization tools are only equipped to study one context at a time, so engineers must run independent optimizations for each context of interest. However, engineered parts frequently appear in many contexts: wind turbines must perform well in many wind speeds, and a bracket might be optimized several times with its bolt-holes fixed in different locations on each run. In this paper, we formulate a framework for variable-context multi-objective optimization. We introduce the *Pareto gamut*, which captures Pareto fronts over a range of contexts. We develop a global/local optimization algorithm to discover the Pareto gamut directly, rather than discovering a single fixed-context "slice" at a time. To validate our method, we adapt existing multi-objective optimization benchmarks to contextual scenarios. We also demonstrate the practical utility of Pareto gamut exploration for several engineering design problems.

CCS Concepts: • **Applied computing** → **Computer-aided design**; • **Theory of computation** → **Continuous optimization**.

Additional Key Words and Phrases: Computer-aided design, computational design, multi-objective optimization

ACM Reference Format:
Liane Makatura, Minghao Guo, Adriana Schulz, Justin Solomon, and Wojciech Matusik. 2021. Pareto Gamuts: Exploring Optimal Designs Across Varying Contexts. *ACM Trans. Graph.* 40, 4, Article 171 (August 2021), 17 pages. https://doi.org/10.1145/3450626.3459750

Authors' addresses: Liane Makatura, MIT, 77 Massachusetts Avenue, Cambridge, MA 02139, USA; Minghao Guo, Chinese University of Hong Kong; Adriana Schulz, University of Washington; Justin Solomon, MIT; Wojciech Matusik, MIT.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2021 Copyright held by the owner/author(s).
0730-0301/2021/8-ART171
https://doi.org/10.1145/3450626.3459750

1 INTRODUCTION

Performance-driven optimization is a cornerstone for effective engineering design, because it permits the fine-tuning of parts based on their predicted performance in the real world. The desired performance metrics for a given part are often multiple and conflicting:

bicycle components should be lightweight and strong, and wind turbines should be compact with high power output. To achieve good trade-offs between such objectives, engineers typically seek *Pareto-optimal* design solutions, or designs for which it is impossible to improve performance on all metrics at the same time; improving any metric will worsen at least one other. Such Pareto-optimal designs are typically identified via multi-objective optimization schemes.

To use a multi-objective optimization scheme, engineers must parametrize the given part with a set of *design variables*. These specify aspects of the design that are allowed to vary during optimization, e.g. the blade radius and pitch of a wind turbine (Fig. 1). To measure the quality of each design, engineers define a set of relevant *performance metrics*, e.g. the turbine's mass and power output. Performance evaluations (and thus, the optimal designs) are also influenced by external constraints or assumptions derived from the part's intended *context*. For instance, a turbine's power output depends on environmental factors like wind speed and air density. As shown in Fig. 1, the Pareto-optimal designs for low wind speeds have short blades, and are limited to modest power outputs.[1] Fast winds yield longer-bladed designs with objectively better trade-offs (better power for a given mass) compared to slow contexts.

Context parameters typically appear as constants embedded in the performance metrics *a priori*: they are not allowed to vary during optimization because engineers cannot *control* the context. For example, if windspeed were allowed to vary, the Pareto front would exclusively contain turbines designed for and evaluated under the fastest wind speed. Nevertheless, these designs are useless for sites with slow wind; the Pareto set is only relevant if it has been optimized subject to the appropriate fixed context.

However, fixed-context optimization is insufficient in practice because the solutions are predicated on unrealistically precise assumptions. The context represents conditions that are complex, dynamic, and hard-to-measure; thus, a single context rarely captures a problem's true nature. For robust design, one must consider Pareto-optimal designs over several contexts. This is tedious with existing optimization tools, which only provide insight into one context at a time. Industrial design programs like SolidWorks and Altair Inspire include tools to manage ad hoc context exploration, but the process remains inefficient. According to a SolidWorks director of product management, "you never hit run once [for any optimization]. You look at the results and run it again with different assumptions and inputs to evaluate those inputs. Then you run it again. It's an iterative process" [Wasserman 2017].

We address this challenge by developing a novel framework for variable-context optimization, which identifies the Pareto fronts for many context values simultaneously (§1.2). To motivate our approach, we present three broad problem classes that would benefit from variable-context optimization, as shown in Fig. 2.

1.1 Variable-Context Optimization: Motivation

As shown in Fig. 2a, our first class considers parts that must withstand a *range* of contexts due to uncertainty (e.g. fabrication precision) or dynamic environments (e.g. wind). By studying Pareto fronts for many fixed-contexts in the range, engineers can identify

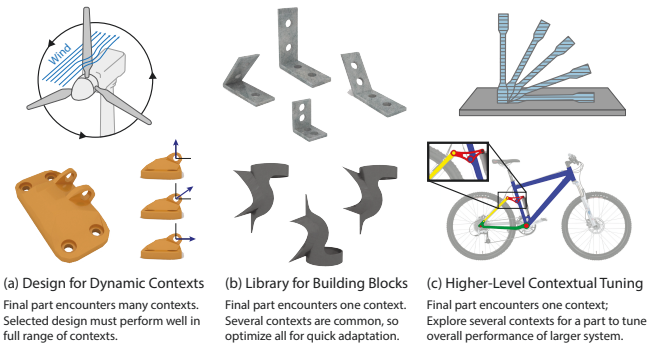


Fig. 2. Multi-context design examples; each column shows a class of problems (context in italic). **(a)** Some parts encounter many contexts: turbines (top) must perform under many *wind speeds* and *temperatures*; mounting brackets (bottom) must withstand a range of loading conditions. **(b)** Manufacturers maintain libraries of basic components for various contexts. We show angle brackets (top) and bicycle lugs (bottom) for different *angle constraints*. **(c)** Contexts are explored to gauge higher-level trade-offs. For 3D-printing (top), *orientation* or *layer height* affect the properties of each individual part, but engineers also set these print parameters to optimize the run overall (capacity for other parts, print time, etc.). Assemblies like bicycles (bottom) require hierarchical design, because the *location of functional interfaces* (e.g., pivot between red rocker and yellow stay) affects the achievable performance of the individual parts and the assembly overall.

universally high-performing designs that compare favorably to the achievable optima in any plausible context. This is more robust than existing methods, where engineers consider optimal designs for some "representative" context given by an expected context distribution or a carefully constructed "worst-case" scenario. Even when executed well, fixed-context approaches can lead to sensitive or over-engineered designs that perform sub-optimally once the context deviates from expectations. However, without access to the achievable fronts for each context, engineers have no efficient way to compare the performance that is sacrificed.

The second class contains "building blocks" like the angle bracket in Fig. 2b, which must assume a specific bending angle (context) for each use case: shelves are mounted with right-angle brackets, but solar panels may require specialized acute angles for optimal power output. Within each context, the bracket's design parameters (e.g., side length, material thickness, bend radius) must be optimized for metrics like mass and compliance. This is expensive with existing tools, so engineers maintain a library of optimal designs for common contexts and compute the solutions for each non-standard request independently. Our approach permits fast adaptation to specialized contexts by providing immediate access to the Pareto-optimal designs for any context in the range.

Our third class considers hierarchical design (Fig. 2c), where contexts are used to enforce high-level decisions during low-level optimizations. This is relevant for complex assemblies like bicycles, which include dozens of sub-assembly components and hundreds of tuneable design choices [Seven Cycles 2020]. It is prohibitively expensive to optimize over the entire system, so engineers must use a hierarchical approach. First, they block out the rough assembly, including the length and adjoining angles of the frame tubes and the

[1] All metrics are formulated for *minimization*, so small values are preferable.

location of pivot points within the rear suspension system [Covill et al. 2014; Spratt 2019]. Each assembly-level decision dictates the *context* of all adjacent parts. Then, each part is optimized for its own set of performance metrics (e.g., minimal mass and compliance), while respecting the context required by the global assembly [Bogomonly 2019; Spratt 2019]. Designers typically explore solutions for many assembly-level configurations, because slight variations can have a drastic effect on the global performance of the bike [Covill et al. 2014; PHeller 2014]. However, each assembly-level change (e.g., pivot position) imposes a new context on the sub-assembly components, which causes a change in the Pareto set for each component and triggers additional optimizations. By contrast, the Pareto gamut provides direct access to the optimized, low-level parts corresponding to any assembly decision within the specified range. We expect this to permit more thorough exploration and a better understanding of each decision's downstream impacts.

1.2 Our Approach

To permit thorough context exploration, we incorporate continuous-valued context variables into the multi-objective optimization formalism. We also provide a global/local algorithm that seeks the Pareto front at every context in a desired range simultaneously, rather than discovering one fixed-context "slice" at a time. This collection of Pareto fronts forms a new object, which we call the *Pareto gamut* (Fig. 1).

Our Pareto gamut discovery algorithm alternates between two steps. First, we use global optimization to uncover a few points on the Pareto gamut. Then, we perform local exploration to uncover Pareto-optimal designs in the neighborhood of each known point. This exploration uses a first-order approximation that takes a single Pareto-optimal point in *some* context and identifies directions in the joint design-context space along which we can walk without leaving the Pareto gamut. By sharing information across contexts and uncovering entire patches of the Pareto gamut at once, our approach amortizes the cost of optimization and permits thorough context exploration at a fraction of the traditional cost.

Our contributions include:

- a formal notion of context parameters and Pareto gamuts,
- a mathematical framework for variable-context optimization,
- a discovery algorithm for the full Pareto gamut, and
- real-world problem classes that benefit from variable-context optimization.

Although our theory extends to any problem size, we focus on examples with a total of at most 4 performance metrics and context variables for computational reasons (discussed in §6 and §7).

2 RELATED WORK

Multi-Objective Optimization. We present a brief overview of multi-objective optimization methods, and refer the reader to Cho et al. [2017] and Marler and Arora [2004] for additional depth. Multi-objective optimization algorithms are classified into three categories, based on the timing of user guidance relative to optimization: before (*a priori*), during (interactive), or after (*a posteriori*) [Van Veldhuizen and Lamont 2000]. Interactive and *a priori* methods allow users to guide the optimization toward "desirable" regions [Koyama et al. 2020; Marler and Arora 2004; Meignan et al. 2015; Ruiz et al. 2019]. These approaches require relatively few samples, but user-guided exploration is biased toward known solutions, which reinforces user expectations and impedes novel insights. By contrast, *a posteriori* methods solicit user preferences only *after* the Pareto front has been discovered. Popular *a posteriori* methods include Normalized Boundary Intersection methods [Das and Dennis 1998], Normalized Normal Constraint methods [Messac et al. 2003], and evolutionary algorithms [Deb and Jain 2014; Deb et al. 2002; Zhang and Xing 2017]. By exposing Pareto-optimal designs before seeking input, *a posteriori* algorithms improve users' understanding of the attainable trade-offs while mitigating known-solution biases.

We develop an *a posteriori* approach over the joint design-context space. This enables full exploration of fixed-context Pareto fronts, while highlighting how the fronts vary when the context changes.

Continuation Methods for Multi-Objective Optimization. The aforementioned algorithms share a common pitfall: each point on the front must be found independently. This is inefficient, as it neglects the fact that Pareto-optimal points are typically clustered together. Via an extension of the classical KKT conditions [Karush 1939; Kuhn and Tucker 1951] (see §3), solutions of multi-objective optimization problems satisfy a constrained system of nonlinear equations with locally continuous solution manifolds [Hillermeier 2001a,b]. This permits the use of path-following methods, or *continuation schemes*, which discover optimal neighborhoods around each known solution.

Rakowska et al. [1991] first applied this approach to bi-objective problems, revealing interpretable design families and explanations for discrete jumps within the Pareto set for specific problems. This work was generalized for k objectives [Hillermeier 2001a; Martín and Schütze 2018], but these papers only consider local expansion from a single known solution. To discover disconnected Pareto sets, recent works embed local continuation schemes in global/local discovery algorithms, as in the interactive method of Schütze et al. [2019] and the *a posteriori* approach of Schulz et al. [2018].

Our global/local discovery algorithm is closest to that of Schulz et al. [2018]. We *generalize* this algorithm and its supporting theory to permit context exploration during the discovery phase. This requires a new data structure and revisions to the global sampling, optimization, and local exploration steps. On the theoretical side, we generalize the perturbative step in Schulz et al. [2018] and the predictor step in Martín and Schütze [2018] by deriving the context's impact on the shape of the Pareto front. Our algorithm is unique in its ability to discover *many* fixed-context fronts simultaneously.

Performance-Driven Design. Optimization is essential for identifying feasible designs that realize some high-level functional specification. Many functional goals have been explored, including balance [Prévost et al. 2013; Wang and Whiting 2016], light reflection [Schwartzburg et al. 2014], acoustic properties [Bharaj et al. 2015; Li et al. 2016], and deformation under a given load [Chen et al. 2014]. Some methods incorporate user feedback in the design loop, coupled with optimizations that guide the user toward viable designs for e.g. functional furniture [Umetani et al. 2012; Yao et al. 2017] or stable-flying multicopters and gliders [Du et al. 2016; Umetani et al. 2014]. Most of these works are restricted to single-objective optimization, and none consider generic context parameters.

There is, however, a growing body of work that considers optimal structural design under a parametrized set of loading conditions. Zhou et al. [2013] optimize the structural integrity of 3D printed structures based on geometry and material alone, with no assumptions about the specific incident load. Panetta et al. [2017] develop methods to minimize the stress concentration of periodic microstructures under all unit loads, while Schumacher et al. [2018] optimize the strength and balance of large-scale 3D-printed objects under unknown external loads. Similarly, Ulu et al. [2017] optimize structures to be as lightweight as possible, while ensuring robustness under any incident force configuration within a set force-magnitude budget. In general, these methods handle uncertainty or variability by computing a *worst-case load*, which accounts for all of the most problematic loading conditions. This is particularly useful if the engineer has little insight about which loading conditions to expect, because it effectively optimizes over the full range of contexts to identify the most critical case(s). However, the conservative nature of this approach can result in designs that are over-engineered (and thus, sub-optimal) with respect to any particular context they may encounter. By contrast, we expose the Pareto fronts for a wide range of plausible contexts, each of may reflect some portion of the "worst-case" aggregate. This provides complementary insight regarding a chosen design's optimality (or lack thereof) relative to the achievable trade-offs for any particular load that may occur.

Design Exploration. We also draw inspiration from methods that *explore* a design space, such as the recent works that seek the gamut of material properties achievable by a fabrication device [Bickel et al. 2010; Dong et al. 2010; Zhu et al. 2017]. These methods probe the design space strategically to achieve a functional goal. Other methods traverse the entire design space so users can efficiently focus on any region of interest [Shugrina et al. 2015]. One such method by Schulz et al. [2017] discusses a special case of context parameters, by interpolating e.g. stress distributions on a mesh subject to distinct loading conditions. However, each load must be selected and precomputed individually on the adaptive grid samples.

Our approach draws on all of these methods to achieve our hybrid goal: to thoroughly explore all contexts in a given range, while only presenting the Pareto-optimal designs for each context.

3 FIXED-CONTEXT PARETO OPTIMALITY

This section reviews the classical theory for fixed-context optimization. Consider a problem parameterized by D design and C context parameters. Any solution can be represented by a design point $\mathbf{x} \in \mathbb{R}^D$ and a context $\mathbf{z} \in \mathbb{R}^C$. For example, a wind turbine might be prescribed by three design variables (radius, height, and pitch angle; $D = 3$) and one context parameter (wind speed; $C = 1$). We control the feasible set of values for each variable using constraint functions $g_k(\mathbf{x}, \mathbf{z}) : \mathbb{R}^D \times \mathbb{R}^C \to \mathbb{R}$ encoding the constraint $g_k(\mathbf{x}, \mathbf{z}) \leq 0$. For the wind turbine, constraints may include an upper bound on the blade radius or a minimal stress rating. We denote our constraints $g_k(\mathbf{x}, \mathbf{z})$ as *active* if $g_k(\mathbf{x}, \mathbf{z}) = 0$ and *inactive* if $g_k(\mathbf{x}, \mathbf{z}) < 0$.

The classical definition of Pareto optimality assumes that the context $\mathbf{z} \in \mathbb{R}^C$ is fixed to a particular value $\mathbf{z} = \mathbf{z}^*$ *a priori*. Since $\mathbf{z}^*$ is effectively a constant, it is typically excluded from the notation. However, we preserve this dependence in anticipation of our variable-context formulation in §4. We begin by formalizing the fixed-context design space, as illustrated in Fig. 3:

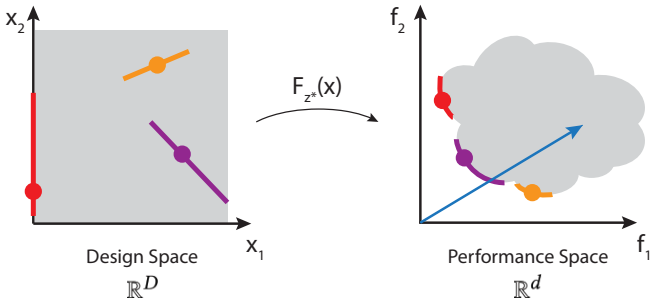


Fig. 3. Fixed-context design space $\mathcal{X}_{\mathbf{z}^*}$ and performance space $F_{\mathbf{z}^*}(\mathcal{X}_{\mathbf{z}^*})$ (gray). The fixed-context Pareto set $\mathcal{P}_{\mathbf{z}^*}$ (red, orange and purple) contains the design points with optimal performance trade-offs; these points map onto the Pareto front $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$. Colors represent contiguous solution manifolds in $\mathcal{X}_{\mathbf{z}^*}$; color is preserved when mapped into performance space. Any ray from the origin (blue) intersects the Pareto front at most once.

Definition 3.1 (Design Space). For fixed context $\mathbf{z}^* \in \mathbb{R}^C$, the feasible *design space* is defined as

$$\mathcal{X}_{\mathbf{z}^*} := \{\mathbf{x} = (x_1, \ldots, x_D) \in \mathbb{R}^D \mid g_k(\mathbf{x}; \mathbf{z}^*) \leq 0, \forall k \in \{0, \ldots, K\}\} .$$

Intuitively, the design space represents the set of valid, manufacturable designs corresponding to the given context $\mathbf{z}^*$.

Each design in this space can be evaluated with respect to a set of d performance metrics $f_i(\mathbf{x}; \mathbf{z}^*) : \mathbb{R}^D \to \mathbb{R}$. For most engineering problems, $d \ll D$. We let $F_{\mathbf{z}^*}(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^d$ be the concatenation of all performance metrics:

$$F_{\mathbf{z}^*}(\mathbf{x}) := (f_1(\mathbf{x}; \mathbf{z}^*), \ldots, f_d(\mathbf{x}; \mathbf{z}^*)).$$

The image of the feasible set $\mathcal{X}_{\mathbf{z}^*}$ under the performance metrics yields the *performance space*:

Definition 3.2 (Performance Space). The fixed-context *performance space* is the image of the design space $\mathcal{X}_{\mathbf{z}^*}$ under the map $F_{\mathbf{z}^*}$:

$$F_{\mathbf{z}^*}(\mathcal{X}_{\mathbf{z}^*}) \subseteq \mathbb{R}^d .$$

Without loss of generality, we assume that small f_i values are preferable. Then, our multi-objective optimization problem can be notated

$$\min_{\mathbf{x}} \{f_i(\mathbf{x}; \mathbf{z}^*)\} \text{ subject to } \mathbf{x} \in \mathcal{X}_{\mathbf{z}^*} . \tag{1}$$

Solutions of Eqn. 1 are *Pareto-optimal* in the following sense:

Definition 3.3 (Dominance). Let $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}_{\mathbf{z}^*}$ be feasible design points. We say $\hat{\mathbf{x}}$ *dominates* $\mathbf{x}$ in context $\mathbf{z}^*$ if $\hat{\mathbf{x}}$ performs at least as well as $\mathbf{x}$ on all metrics:

$$f_i(\hat{\mathbf{x}}; \mathbf{z}^*) \leq f_i(\mathbf{x}; \mathbf{z}^*), \forall i \in \{1, \ldots, d\},$$

and $\hat{\mathbf{x}}$ is strictly better than $\mathbf{x}$ on at least one metric:

$$\exists i \in \{1, \ldots, d\} \text{ s.t. } f_i(\hat{\mathbf{x}}; \mathbf{z}^*) < f_i(\mathbf{x}; \mathbf{z}^*).$$

Definition 3.4 (Pareto Optimality). A point $\mathbf{x} \in \mathcal{X}_{\mathbf{z}^*}$ is *Pareto-optimal* if there does not exist any $\hat{\mathbf{x}} \in \mathcal{X}_{\mathbf{z}^*}$ that dominates $\mathbf{x}$. The set of Pareto-optimal points is the fixed-context *Pareto set* $\mathcal{P}_{\mathbf{z}^*} \subseteq \mathbb{R}^D$; the image $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*}) \subseteq \mathbb{R}^d$ is the fixed-context *Pareto front*.

Assuming nonnegative performance metrics, Pareto optimality can be illustrated as in Fig. 3. For any ray drawn from the origin in direction $\alpha \in \mathbb{R}^d_+$, there exists at most one $t \geq 0$ such that $t\alpha \in F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$. Thus, for minimization, the best designs are those that map closest to the origin along different directions.

Generically, the Pareto front and set are composed of collections of $(d-1)$-dimensional manifolds [Hillermeier 2001a], implying that almost any Pareto-optimal design sits within some Pareto-optimal neighborhood. This insight is exposed via the *dual* formulation of Eqn. 1, as expressed by the KKT conditions [Hillermeier 2001b]:

Proposition 3.5 (KKT conditions). *Let f_i and g_k be continuously differentiable, and assume the vectors*

$$\{\nabla_x g_{k'} \mid k' \text{ is an index of an active constraint}\}$$

are linearly independent. Then, for any $\mathbf{x}^* \in \mathcal{P}_{\mathbf{z}^*}$, there exist dual variables $\boldsymbol{\alpha}^* \in \mathbb{R}^d, \boldsymbol{\beta}^* \in \mathbb{R}^K$ such that:

$$\mathbf{x}^* \in \mathcal{X}_{\mathbf{z}^*}, \tag{2}$$

$$\alpha_i^* \geq 0 \quad \forall i \in \{1, \ldots, d\}, \tag{3}$$

$$\beta_k^* \geq 0 \quad \forall k \in \{1, \ldots, K\}, \tag{4}$$

$$\sum_{i=1}^{d} \alpha_i^* = 1, \tag{5}$$

$$\beta_k^* g_k(\mathbf{x}^*; \mathbf{z}^*) = 0 \quad \forall k \in \{1, \ldots, K\}, \text{ and} \tag{6}$$

$$\left[\sum_{i=1}^{d} \alpha_i^* \nabla_{\mathbf{x}} f_i(\mathbf{x}^*; \mathbf{z}^*)\right] + \left[\sum_{k=1}^{K} \beta_k^* \nabla_{\mathbf{x}} g_k(\mathbf{x}^*; \mathbf{z}^*)\right] = 0. \tag{7}$$

The KKT conditions are necessary (although not sufficient) for Pareto optimality. Thus, to identify Pareto candidates around a known Pareto-optimal point $(\mathbf{x}^*; \mathbf{z}^*)$, we can search for points that locally *preserve* the KKT conditions. In particular, we differentiate the KKT conditions with respect to $\mathbf{x}$ and set this derivative equal to zero, yielding a first-order approximation of the Pareto front about $(\mathbf{x}^*; \mathbf{z}^*)$. Practically, this reveals $(d-1)$ orthogonal directions $\mathbf{x}' \in \mathcal{X}_{\mathbf{z}^*}$ along which we can walk without violating the KKT conditions. This observation is the crux of fixed-context continuation schemes [Hillermeier 2001a; Martín and Schütze 2018; Schulz et al. 2018], and forms the basis of our local exploration as well (§4.3).

4 PARETO GAMUTS

Fixed-context Pareto optimality allows engineers to compute design trade-offs for any *particular* context, e.g., a turbine under fixed wind speed V. To select a turbine that performs well overall, however, engineers must explore designs over a *range* of possible wind speeds.

To facilitate this context exploration, we introduce the *Pareto gamut*, which captures the optimal designs at all contexts in a given range. Intuitively, the Pareto gamut is a collection of fixed-context Pareto fronts that are stacked along an additional axis according to their context value, as illustrated in Fig. 4.

This section formally defines the Pareto gamut (§4.1) and the technical distinction between context and design variables (§4.1.1, §4.2). By accounting for each variable type, we also derive the first-order approximation to the Pareto gamut (§4.3), which exposes KKT-preserving perturbations for design *and* context values.

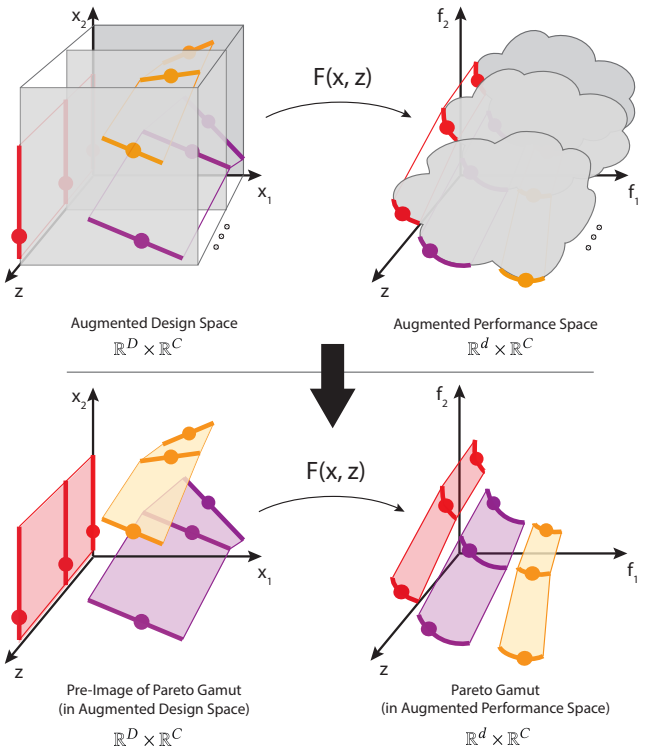


Fig. 4. (**Top**) Augmented design (performance) space is the union of many fixed-context design (performance) spaces, drawn in gray. Each fixed context admits a corresponding Pareto set (front), with contiguous families shown in orange, purple, and red. **(Bottom)** The Pareto gamut $F(\mathcal{P})$ is the union of all Pareto fronts in augmented performance space; the pre-image of the Pareto gamut $\mathcal{P}$ is the union of all Pareto sets in augmented design space.

4.1 Formal Definition

Given a bounded range of interest $[z_i^{\min}, z_i^{\max}]$ for each context parameter, the set of context configurations is as follows:

Definition 4.1 (Context Space). The *context space* $\mathcal{Z}$ is given by

$$\mathcal{Z} := \{\mathbf{z} = (z_1, \ldots, z_C) \in \mathbb{R}^C \mid z_i^{\min} \leq z_i \leq z_i^{\max}, \ \forall i \in \{1 \ldots C\}\}.$$

Each $\mathbf{z}^* \in \mathcal{Z}$ yields a distinct Pareto set and front. To organize these objects, we define the following *augmented spaces*:

Definition 4.2 (Augmented Spaces). The *augmented design space* is

$$\mathcal{X} := \{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^D \times \mathbb{R}^C \mid \mathbf{z} \in \mathcal{Z} \text{ and } g_k(\mathbf{x}, \mathbf{z}) \leq 0 \forall k \in \{0, \ldots, K\}\}.$$

The *augmented performance space* is the image of the augmented design space under the map $F(\mathbf{x}, \mathbf{z}) : \mathbb{R}^D \times \mathbb{R}^C \to \mathbb{R}^d \times \mathbb{R}^C$, where $F(\mathbf{x}, \mathbf{z}) := (F_{\mathbf{z}}(\mathbf{x}), \mathbf{z})$.

As shown in Fig. 4, the augmented design (performance) space can be visualized by stacking fixed-context design (performance) spaces along a context axis. Each fixed-context design (performance) space contains a Pareto set (front), as shown by the manifolds in Fig. 4 (top). The *collection* of these Pareto sets (fronts) is our object of study (Fig. 4, bottom). We define this object as follows:

Definition 4.3 (Pareto Gamut). The *Pareto gamut* $F(\mathcal{P})$ is the union of all fixed-context Pareto fronts in augmented performance space:

$$F(\mathcal{P}) = \bigcup_{\mathbf{z}^* \in \mathcal{Z}} F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*}) \quad \subseteq \mathbb{R}^d \times \mathbb{R}^C.$$

The pre-image of the Pareto gamut is denoted $\mathcal{P} \subseteq \mathbb{R}^D \times \mathbb{R}^C$.

The Pareto gamut characterizes the full set of trade-offs at any context configuration, and encodes how these trade-offs change as a function of the context.

4.1.1 Context Parameters: A Novel Entity. The Pareto gamut differs from any fixed-context Pareto front because context parameters are distinct from existing machinery in multi-objective optimization. We validate this with two brief thought experiments to the contrary. If we treated the context as a performance metric, we would unjustly penalize large context values and bias our solutions toward high-performing designs with small context values. Alternatively, if we were to treat context parameters as design variables, we would only be able to identify the *joint* context/design configurations with optimal trade-offs. This yields a subset of the Pareto gamut containing the best-performing designs from *any* context in the permissible range – such as the long-bladed turbines in high wind speeds, which dominate all designs from lower speeds (§1). We examine this subset more closely in §4.2. For now, we note that neither of these context-parameter treatments yield the full Pareto gamut, which captures the entire set of trade-offs under any particular context. Thus, the context parameters must be handled as a novel entity.

In particular, a point $(\mathbf{x}^*, \mathbf{z}^*)$ on the Pareto gamut is (by construction) only necessarily Pareto optimal w.r.t. its fixed context $\mathbf{z}^*$. As in previous work, this implies that context values must be held constant during optimization. However, as discussed in §4.3, the Pareto gamut formalism allows us to relax this constraint during non-optimization steps, to efficiently explore varying contexts.

4.2 Gamut Lower Envelope

The *lower envelope* of the Pareto gamut contains the best achievable trade-offs over *all* contexts (Fig. 5). This offers important intuition, because it is precisely the solution we would get if we were to treat the context parameters as design variables during optimization.

Definition 4.4 (Lower Envelope). The *lower envelope* $F(\mathcal{L})$ of a Pareto gamut $F(\mathcal{P})$ is the set of gamut points that are not dominated by points in another context:

$$F(\mathcal{L}) := \{F(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^d \times \mathbb{R}^C \mid \nexists (\hat{\mathbf{x}}, \hat{\mathbf{z}}) \in \mathcal{P} \text{ that dominates } (\mathbf{x}, \mathbf{z})\}. \tag{8}$$

The *flattened lower envelope* $F^{\perp}(\mathcal{L}) \subseteq \mathbb{R}^d$ is obtained by projecting away the context dimensions of all points in $F(\mathcal{L})$.

We illustrate the lower envelopes in Fig. 5. The flattened lower envelope is particularly useful, as it relates to the Pareto front of a modified problem that treats context parameters as design variables:

Definition 4.5 (Free-Context Problem). The *free-context problem* $H_F(\boldsymbol{y}) : \mathbb{R}^{D+C} \to \mathbb{R}^d$ is a modified version of F that treats all context parameters of F as additional design variables that can be optimized. Namely, the design variables are $\boldsymbol{y} := (x_1, \ldots, x_D, z_1, \ldots, z_C) \in \mathcal{X}$, and each metric is $h_i(\boldsymbol{y}) = f_i(\mathbf{x}, \mathbf{z})$.

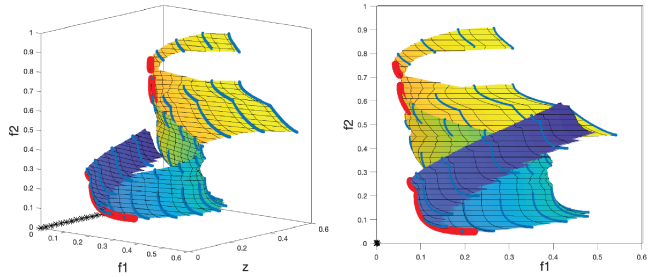


Fig. 5. The Pareto gamut's *lower envelope* (red curve, left) contains the best achievable trade-offs over *all* contexts. By projecting away the context axes, we obtain the *flattened lower envelope* (red curve, right). The Pareto gamut (mesh) is colored by context (z) value. Blue curves indicate fixed-context Pareto fronts, and black asterisks mark the utopia point for each context ($f_1(\mathbf{x}; \mathbf{z}) = f_2(\mathbf{x}; \mathbf{z}) = 0$). The lower envelope spans many contexts.

The definition of Pareto optimality for H_F in the free-context problem is precisely Eqn. 8 in Definition 4.4. Hence, we have the following proposition:

Proposition 4.6 (Lower Envelope Optimality). *For variable-context objective function $F(\mathbf{x}, \mathbf{z}) : \mathbb{R}^D \times \mathbb{R}^C \to \mathbb{R}^d \times \mathbb{R}^C$, we have $H_F(\mathcal{P}) = F^{\perp}(\mathcal{L})$, where $H_F(\mathcal{P})$ is the Pareto front of the free-context problem H_F and $F^{\perp}(\mathcal{L})$ is the flattened lower envelope defined above.*

Proposition 4.6 exposes a direct correlation between the lower envelope of our Pareto gamuts and a related Pareto front that can be obtained via existing multi-objective optimization benchmarks. We use this property in §6.1 to validate our approach against fixed-context multi-objective optimization benchmarks.

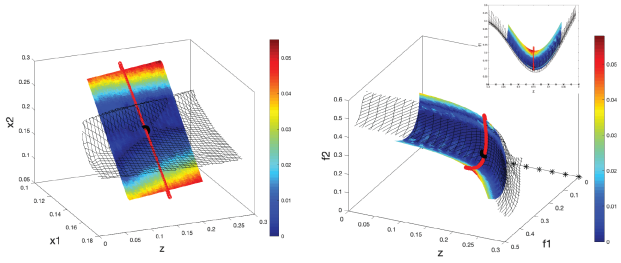
4.3 First-Order Approximation

To discover the Pareto gamut efficiently, we develop a first-order approximation that identifies optimal neighborhoods around known-optimal points. We build on the approach from §3, which yields directions $x' \in \mathcal{X}_{\mathbf{z}^*}$ that preserve the KKT conditions *within* a fixed context $\mathbf{z}^*$. Since the Pareto gamut is a collection of fixed-context fronts, its approximation must also permit us to move *across* contexts without violating the KKT conditions. Thus, we seek generalized directions $(\mathbf{x}', \mathbf{z}') \in \mathcal{X}$ s.t. the neighbor $(\hat{\mathbf{x}}, \hat{\mathbf{z}}) := (\mathbf{x}^*, \mathbf{z}^*) + t \cdot (\mathbf{x}', \mathbf{z}')$ satisfies the KKT conditions w.r.t. its fixed-context $\hat{\mathbf{z}}$.

Intuitively, we expect $(d - 1 + C)$ such directions, as the Pareto gamut can be seen as an extrusion of some fixed-context front (of dimension $d - 1$) through a C-dimensional context space. This provides geometric insight about our directions and those from previous works: the latter form a $(d - 1)$-dimensional subspace of the $(d - 1 + C)$-dimensional space spanned by ours (see Fig. 6).

We formalize this intuition in the following proposition, which constructs a linear system that yields the desired directions $(\mathbf{x}', \mathbf{z}')$:

Proposition 4.7 (Augmented KKT Perturbation). *Suppose $(\mathbf{x}(t), \mathbf{z}(t)) : (-\varepsilon, \varepsilon) \to \mathbb{R}^D \times \mathbb{R}^C$ is composed of Pareto-optimal points, that is, $(\mathbf{x}(t), \mathbf{z}(t)) \in \mathcal{P}_{\mathbf{z}(t)}$ for all $t \in (-\varepsilon, \varepsilon)$. Let f_i and g_k be continuously differentiable for $1 \le i \le d$ and $1 \le k \le K'$, where $K' \le K$ is the number of active constraints at $(\mathbf{x}(0), \mathbf{z}(0))$. Let $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$ be the KKT dual variables at $(\mathbf{x}(t), \mathbf{z}(t))$ (see Proposition*



Augmented Design Space Augmented Performance Space

Fig. 6. Existing continuation schemes identify a $(d-1)$-dimensional approximation (red curve) within a single fixed context of the Pareto gamut. Proposition 4.7 describes a $(d-1+C)$-dimensional approximation (colored mesh) that *contains* the previous work as a subspace, while also encoding how the Pareto set/front change as a function of the context. Our approximation is colored by its error relative to the ground truth (wireframe).

3.5). Then, for any KKT-preserving perturbation v,

$$\underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D_{\mathbf{x}}G & D_{\mathbf{z}}G \\ D_xF^T & D_xG^T & Hx & Hz \end{bmatrix}}_{M \in \mathbb{R}^{(1+K'+D)\times(d+K'+D+C)}} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}'(0) \\ \boldsymbol{\beta}'(0) \\ \mathbf{x}'(0) \\ \mathbf{z}'(0) \end{bmatrix}}_{v \in \mathbb{R}^{(d+K'+D+C)}} = \mathbf{0}\,, \qquad (9)$$

where $G(\mathbf{x},\mathbf{z}) := (g_1(\mathbf{x},\mathbf{z}),\ldots,g_k(\mathbf{x},\mathbf{z}))$,

$$Hx := \sum_{i=1}^{d} \alpha_i(0) H_{xx} f_i(\mathbf{x}(0),\mathbf{z}(0)) + \sum_{k=1}^{K'} \beta_k(0) H_{xx} g_k(\mathbf{x}(0),\mathbf{z}(0))\,, \text{ and}$$

$$Hz := \sum_{i=1}^{d} \alpha_i(0) H_{xz} f_i(\mathbf{x}(0),\mathbf{z}(0)) + \sum_{k=1}^{K'} \beta_k(0) H_{xz} g_k(\mathbf{x}(0),\mathbf{z}(0)).$$

For a function $X(u,v)$, the expression D_uX denotes the Jacobian of X w.r.t. the variables u. Similarly, $H_{uv}X$ denotes the matrix of second derivatives obtained by taking the partials of X w.r.t. u and then v. All D_uX and $H_{uv}X$ in Eqn. 9 are evaluated at $(\mathbf{x}(0),\mathbf{z}(0))$. We provide a constructive proof for Proposition 4.7 in Appendix A and a detailed algorithm for its computation in §5.

Remark 1. *Eqn. 9 is a generalization of the results given by Schulz et al. [2018, Proposition 5.1] and Martín and Schütze [2018, Theorem 3.1]. If* $\mathbf{z}'(t) \equiv \mathbf{0}$ *(i.e., if we do not allow the context to change), we recover the expressions from previous work.*

To verify our result, we analyze the number of exploration directions, v. As shown in Eqn. 9, all directions v reside in the null space of M. The rank of M is at most min(#rows, #cols) = $\min(1+K'+D, d+K'+D+C) = D+1+K'$. Then, the Rank Nullity theorem bounds the number of independent directions v:

$$\begin{aligned} \text{nullity}(M) &= \#\text{columns(M)} - \text{rank}(M) \\ &\geq (d+K'+D+C) - (1+K'+D) \\ &= d-1+C\,. \end{aligned}$$

Thus, by computing the null space of M, we generically obtain a set of at least $d-1+C$ directions along which we can walk in augmented design space without violating the KKT conditions. As discussed, the exact case (nullity$(M) = d-1+C$) is the expected dimensionality for our approximation. Additional directions typically indicate that some design variables (or combinations thereof) have no effect on the performance metrics locally. In this case, it is sufficient to select any subset of $d-1+C$ directions at random, and discard the rest.

We also considered higher-order approximations, but they are prohibitively expensive to compute. We find that affine expansions are sufficiently accurate, even when $\mathcal{P}$ is disconnected or curved.

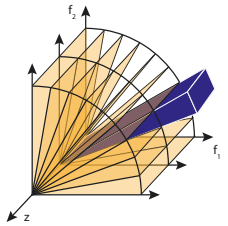
5 PARETO GAMUT DISCOVERY

Given the user's performance objective F, constraints G, and augmented design space $\mathcal{X}$, we compute the Pareto gamut $F(\mathcal{P})$ and its pre-image $\mathcal{P}$. Each iteration of our algorithm (Alg. 1) begins with N_s points sampled from $\mathcal{X}$. Each sample is optimized within its fixed context, such that it satisfies the KKT conditions (Proposition 3.5). We then apply Proposition 4.7 to compute a first-order approximation about each optimized sample. The resulting $(d-1+C)$-dimensional patches are stored and sampled to provide seed points for the next iteration. Discovery continues in this manner until the Pareto gamut converges or the user's computation budget is reached. Below, we detail the necessary data structures and subroutines.

5.1 Data Structure

Our approach discovers continuous manifolds along the Pareto gamut, each of which may span a wide range of performance and context values. We represent each manifold with a simple *patch struct* containing a unique patch ID and information about the patch in continuous *and* discrete forms. For the continuous representation, we store (1) the optimized center point $(\mathbf{x}^*,\mathbf{z}^*) \in \mathcal{X}$ and its image $F(\mathbf{x}^*,\mathbf{z}^*)$, and (2) the set of $d-1+C$ expansion directions that define the locally optimal manifold in $\mathcal{X}$. We also use the patch struct to store a set of discrete points $(\mathbf{x},\mathbf{z}) \in \mathcal{X}$ on this expanded patch, along with the image $F(\mathbf{x},\mathbf{z})$ of each patch point. The resulting patch struct is then appended to a list called the *patch array*.

Although the patch array is a natural way to store this data, it is not well-suited to measure coverage, convergence, or dominance. We address this with an acceleration structure: the *augmented buffer* (see inset). As discussed in §3, any ray $\alpha \in \mathbb{R}^d_+$ from the origin intersects the fixed-context Pareto front $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ at most once. This lets us discretize $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ using (hyper)spherical coordinates: the angle vector $\phi \in [0,\pi/2]^{d-1}$ indicates a performance trade-off (ray direction), and the smallest radius r for which ϕ intersects the feasible performance space indicates the Pareto front candidate.



For the Pareto gamut, we use (hyper)cylindrical coordinates: the context space $\mathcal{Z} \subset F(\mathcal{X})$ is discretized using C Cartesian coordinates, and the performance space uses $d-1$ (hyper)spherical dimensions. The number of cells N_c along each dimension is user-specified.

When a new patch is found, we insert its patchID and discretized patch points $F(\mathbf{x},\mathbf{z})$ into the augmented buffer. Each point is converted to (hyper)cylindrical coordinates then projected into the

ALGORITHM 1: Pareto gamut discovery

Input: Perf. metrics F, constraints G, augmented design space $\mathcal{X}$
Output: Pareto Gamut $F(\mathcal{P})$ and its pre-image $\mathcal{P}$
$B \leftarrow$ Empty Augmented Buffer
$P \leftarrow []$ ▷ empty Patch Array
$patchID \leftarrow 0$
repeat
  $(\mathbf{x}_0, \mathbf{z}_0^*), \ldots, (\mathbf{x}_{N_s}, \mathbf{z}_{N_s}^*) \leftarrow$ stochasticSampling($\mathcal{X}, P$)
  for $i = 0 : N_s$ **do**
    $(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \leftarrow$ fixedContextOptimization($\mathbf{x}_i, F_{\mathbf{z}_i^*}, G_{\mathbf{z}_i^*}$)
    (*dirs*, *extents*, *degenerate*) $\leftarrow$ localExplorationDirections($\mathbf{x}^*$, $\mathbf{z}_i^*$, $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}^*$, F, G)
    if not *degenerate* **then**
      patchID++
      patchPts $\leftarrow$ patchEvaluation($\mathbf{x}^*, \mathbf{z}_i^*$, *dirs*, *extents*, F, $\mathcal{X}$)
      patchStruct $\leftarrow$ (*patchID*, $\mathbf{x}^*, \mathbf{z}_i^*, F(\mathbf{x}^*, \mathbf{z}_i^*)$, *dirs*, *extents*, *patchPts*)
      P.append(*patchStruct*)
      B.addPatch(*patchID*, *patchPts*)
  end
until *computation budget exceeded or converged (see 5.2)*
$(\mathcal{P}, F(\mathcal{P}))$ = extractParetoGamut(B, P)
return $\mathcal{P}, F(\mathcal{P})$

ALGORITHM 2: fixedContextOptimization

Input: Random sample $\mathbf{x}$, fixed-context perf. metrics $F_{\mathbf{z}^*}$, fixed-context constraints $G_{\mathbf{z}^*}$
Output: Optimized point $\mathbf{x}^*$, KKT dual variables $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$
▷ Get target $t \in F_{\mathbf{z}^*}(\mathcal{X}_{\mathbf{z}^*})$ for scalarization
$p_{\text{in}} \leftarrow F_{\mathbf{z}^*}(\mathbf{x})$
$\alpha \leftarrow \frac{p_{\text{in}}}{||p_{\text{in}}||_1}$ ▷ approx. tradeoff between metrics
$p_{\text{goal}} \leftarrow [\min(1, 2 * (\alpha_i - 1/d)) \text{ for } i = 1 : d]$
$dir \leftarrow \frac{p_{\text{goal}} - p_{\text{in}}}{||p_{\text{goal}} - p_{\text{in}}||_2}$
$\boldsymbol{t} \leftarrow p_{\text{in}} + \delta_{\text{dir}} \cdot ||p_{\text{in}}||_2 \cdot dir$ ▷ δ_{dir} is user-specified

▷ Optimize within fixed context
$f_s(\boldsymbol{y}) \leftarrow \frac{1}{2}||F_{\mathbf{z}^*}(\boldsymbol{y}) - \boldsymbol{t}||^2$ ▷ scalarization function handle
$[\mathbf{x}^*, \boldsymbol{\lambda}] \leftarrow$ constrainedOptimization($f_s, \mathbf{x}, G_{\mathbf{z}^*}$)
while $F_{\mathbf{z}^*}(\mathbf{x}^*)$ *dominates* $\boldsymbol{t}$ **do**
  $\boldsymbol{t} \leftarrow \boldsymbol{t} + \delta \cdot dir$ ▷ push target further; $\mathbf{x}^*$ not optimal
  $f_s(\boldsymbol{y}) \leftarrow \frac{1}{2}||F_{\mathbf{z}^*}(\boldsymbol{y}) - \boldsymbol{t}||^2$ ▷ new scalarization
  $[\mathbf{x}^*, \boldsymbol{\lambda}] \leftarrow$ constrainedOptimization($f_s, \mathbf{x}^*, G_{\mathbf{z}^*}$)
end
$\boldsymbol{\alpha}^* \leftarrow \frac{F_{\mathbf{z}^*}(\mathbf{x}^*) - \boldsymbol{t}}{||F_{\mathbf{z}^*}(\mathbf{x}^*) - \boldsymbol{t}||_1}$
$\boldsymbol{\beta}^* \leftarrow \boldsymbol{\lambda}$ ▷ Lagrange multipliers for $G_{\mathbf{z}^*}$ at $(\mathbf{x}^*; \mathbf{z}^*)$
return $\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$

appropriate buffer cell (blue wedge in the inset). To judge convergence, coverage, and Pareto-optimality, we need only store each point's radius from its fixed-context origin and a reference to its details in the patch array.

5.2 Discovery Algorithm

Our global/local discovery algorithm (Alg. 1) consists of 4 simple steps: (1) randomly select seed points in joint context-design space; (2) push each seed point toward its fixed-context Pareto front; (3) compute a set of KKT-preserving exploration directions about each optimized point; and (4) evaluate the patch spanned by the exploration directions to uncover neighborhoods of Pareto-optimal designs. These steps are repeated until either the Pareto gamut converges or the permissible computation budget is exceeded.

Stochastic Sampling. To avoid local minima, each iteration begins with a user-specified number N_s of random samples $(\mathbf{x}_i, \mathbf{z}_i^*) \in \mathcal{X}$ for $i = 1 \ldots N_s$. In the first iteration, samples are selected uniformly at random from $\mathcal{X}$. Uniform sampling is also used in the final iteration(s) before convergence, to promote a globally robust solution. For all other iterations, each sample is obtained by selecting a patch at random, traversing to a random point on the patch, and computing a new sample $(\mathbf{x}_i, \mathbf{z}_i^*)$ using the perturbative formula from Schulz et al. [2018, §6.2.2]. We clamp all results to ensure $(\mathbf{x}_i, \mathbf{z}_i^*) \in \mathcal{X}$.

Fixed-Context Optimization. Each sample $(\mathbf{x}_i, \mathbf{z}_i^*)$ is optimized toward its fixed-context Pareto front $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ using the scalarization procedure detailed in Alg. 2. This yields an optimal point $(\mathbf{x}_i^*, \mathbf{z}_i^*) \in \mathcal{X}$ and its corresponding KKT dual variables, $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$. We use MATLAB's `fmincon` routine for constrained optimization.

Exploration Directions. For each $(\mathbf{x}_i^*, \mathbf{z}_i^*)$, we then apply Proposition 4.7 to extract $(d-1+C)$ KKT-preserving directions. As shown in Alg. 3, the output of Proposition 4.7 requires some post-processing. Our linear system yields vectors of the form $(\boldsymbol{\alpha}', \boldsymbol{\beta}', \mathbf{x}', \mathbf{z}')^T$, so the directions of interest $(\mathbf{x}', \mathbf{z}')$ may be linearly dependent. Moreover, performance metrics are often disproportionately sensitive to changes in context value; thus, context perturbations appear in many of the directions, and the magnitude of these perturbations overshadows that of the design variables, which reduces movement "within" each fixed context. The Gaussian elimination and extent-finding steps described in Alg. 3 mitigate these issues by eliminating redundancy in the directions and encouraging similarly-scaled movement across and within distinct contexts.

Patch Evaluation. The span of all $(\mathbf{x}', \mathbf{z}')$ yields a (hyper)rectangular patch $A^i \subseteq \mathbb{R}^D \times \mathbb{R}^C$, bounded by the computed *extent* along each direction. To sample $F(A^i)$ for insertion into the buffer, we discretize A^i as a uniform grid (with user-specified dimensions) and evaluate F at each vertex. Any vertices $(\mathbf{x}, \mathbf{z}) \notin \mathcal{X}$ are discarded before evaluation. Because A^i is connected and F is continuous, the true image $F(A^i)$ should be connected. However, our discretized vertices may skip over cells of the augmented buffer due to undersampling. To promote the likelihood that our patch contributes a sample to each cell intersecting $F(A^i)$, we lift a fixed triangulation of the grid on A^i to triangulate our (hyper)cylindrical augmented performance points in design space. We use this triangulation to compute $(d+C)$-dimensional barycentric coordinates for every buffer cell within the boundary of $F(A^i)$. We cannot directly interpolate along our approximation of $F(A^i)$, as this may introduce infeasible points when the gamut is non-convex. Instead, we use the barycentric coordinates to interpolate along A^i; then, we map the interpolated point(s) back to performance space for insertion into the buffer.

ALGORITHM 3: localExplorationDirections

Input: Optimal point $\mathbf{x}^*$, context $\mathbf{z}^*$, KKT dual variables $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*$, perf. metrics $F = (f_1, \ldots, f_d)$, constraints $G_{\text{all}} = (g_1, \ldots, g_K)$
Output: Exploration directions *dirs*, extent to walk along each direction *extents*, whether expansion is *degenerate*

▷ get exploration directions using Proposition 4.7
$G \leftarrow$ getActiveConstraints($\mathbf{x}^*, \mathbf{z}^*, G_{\text{all}}$)
$(D_xG, D_zG, D_xF) \leftarrow \left(\frac{\partial G}{\partial \mathbf{x}}\big|_{(\mathbf{x}^*,\mathbf{z}^*)}, \frac{\partial G}{\partial \mathbf{z}}\big|_{(\mathbf{x}^*,\mathbf{z}^*)}, \frac{\partial F}{\partial \mathbf{x}}\big|_{(\mathbf{x}^*,\mathbf{z}^*)}\right)$
$Hx, Hz \leftarrow$ zeroMatrices
for $i = 1 : d$ **do**
  $Hx \mathrel{+}= \alpha_i^* \cdot \frac{f_i}{\partial \mathbf{x} \partial \mathbf{x}}\big|_{(\mathbf{x}^*,\mathbf{z}^*)}$
  $Hz \mathrel{+}= \alpha_i^* \cdot \frac{f_i}{\partial \mathbf{x} \partial \mathbf{z}}\big|_{(\mathbf{x}^*,\mathbf{z}^*)}$
end
for $k = 1 : K'$ **do**
  $Hx \mathrel{+}= \beta_i^* \cdot \frac{g_k}{\partial \mathbf{x} \partial \mathbf{x}}\big|_{(\mathbf{x}^*,\mathbf{z}^*)}$
  $Hz \mathrel{+}= \beta_i^* \cdot \frac{g_k}{\partial \mathbf{x} \partial \mathbf{z}}\big|_{(\mathbf{x}^*,\mathbf{z}^*)}$
end

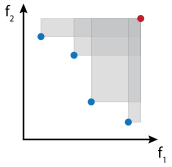
$$M \leftarrow \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D_{\mathbf{x}}G & D_{\mathbf{z}}G \\ D_xF^T & D_xG^T & Hx & Hz \end{bmatrix}$$

$V \leftarrow$ null(M) ▷ each column is a direction $[\boldsymbol{\alpha}', \boldsymbol{\beta}', \mathbf{x}', \mathbf{z}']^T$
if *numCols*(V) $< d + C - 1$ **then return** degenerate=True
if *numCols*(V) $> d + C - 1$ **then**
  $V \leftarrow$ randomly select $d + C - 1$ columns of V

$V \leftarrow V(d + K' + 1 : \text{end}, :)^T$ ▷ $[\mathbf{x}', \mathbf{z}']$ on each row
$V \leftarrow [V(:, D + 1 : \textit{end}), V(:, 1 : D)]$ ▷ $[\mathbf{z}', \mathbf{x}']$ on each row
$V \leftarrow$ sort rows of V in decreasing order of context values
$V \leftarrow$ gaussElimination(V)
if *any row of* V *is zero* **then return** degenerate=True
$dirs \leftarrow [V(:, C + 1 : \textit{end}), V(:, 1 : C)]^T$ ▷ $[\mathbf{x}', \mathbf{z}']$ on each col

▷ Compute extent along each dir $\in \mathcal{X}$ to move unit length in $F(\mathcal{X})$
$DF \leftarrow \frac{\partial F}{\partial (\mathbf{x}, \mathbf{z})}\big|_{(\mathbf{x}^*,\mathbf{z}^*)}$
$Q \leftarrow DF \cdot dirs$ ▷ each col gives perturbation in $F(\mathcal{X})$
extents $\leftarrow (1/||col||$ for each col in columns(Q))
return dirs, extents, degenerate=False

Gamut Convergence. To measure convergence, we must quantify the quality of our Pareto gamut over time. One popular metric for fixed-context optimization is *hypervolume* [Bader and Zitzler 2011]. As shown in the inset, hypervolume (gray) is the Lebesgue measure of the performance space region that is strictly dominated by the candidate Pareto front points (blue), relative to a "worst case" *nadir point* – the upper bound of each f_i (red). As the front improves, hypervolume increases strictly monotonically. Thus, fixed-context methods often declare convergence when the *hypervolume improvement* falls below some threshold δ_h for N_h consecutive iterations.



For the Pareto gamut, we compute a hypervolume vector V^t, where V_i^t is the hypervolume of the i^{th} fixed-context Pareto front after iteration t. We consider our gamut converged if the hypervolume improvement $(V^t - V^{t-j})(V^t - V^{t-j})^T < \delta_h$ for all $j \in [1 \ldots N_h]$.

Pareto Gamut Extraction. Once the gamut has converged, each buffer cell contains the Pareto-dominant point p_ϕ along its respective ray direction ϕ. However, p_ϕ is not necessarily part of the Pareto front: it may be excluded if it is dominated by solutions along different rays. Hence, for each fixed context in the augmented buffer, we extract $F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ according to Definition 3.4.

This set of points (at most one per buffer cell) is often sufficiently dense to capture the Pareto gamut. However, discretization artifacts can produce sparse results in certain scenarios. For example, if the Pareto gamut is asymptotic to the augmented performance axes (as in Fig. 1), the buffer cells can only provide a sparse sampling in this region. To address this without resorting to a fine discretization, we use our patch array to fill gaps in the Pareto gamut after extraction.

In more detail, consider the points in the extracted Pareto gamut that belong to a given patch A^i, and let I be the pre-image of these points. Assuming that the true Pareto front contains a contiguous region of A^i, the boundary of the point set I denotes the maximal region of the patch A^i that could map to the Pareto gamut. Then, all patch samples $(\mathbf{x}, \mathbf{z}) \in A^i$ within this boundary should be eligible for inclusion in the Pareto gamut. A dense sampling of this patch region is contained in the patch array, but the buffer (from which we extract the Pareto gamut) only admits one sample per A^i per cell. Thus, the majority of discretized patch samples are never included in the buffer or considered as candidates for the extracted Pareto gamut. By incorporating these samples directly through the post-processing step described above, we dramatically improve our coverage of the Pareto gamut with minimal additional cost. To remove any dominated points introduced by this process, we run a final dominance check for each fixed context before returning.

6 RESULTS

We test our approach on two problem classes. First, we adapt analytic fixed-context benchmarks to validate our algorithm's correctness. Then, we explore engineering design problems to show the utility of our approach and the Pareto gamut more broadly. For all problems, we normalize the parameters and metrics such that $\mathbf{x} \in [0, 1]^D$, $\mathbf{z} \in [0, 1]^C$, and $f_i \in [0, 1]\ \forall i \in \{1 \ldots d\}$. We set $N_c = 200$, $N_s = 10$, $d_{\text{dir}} = 0.3$, $\delta_h = 10^{-3}$, and $N_h = 3$, with nadir point $[1]^d$. All Pareto gamuts are visualized with context along the vertical axis.

6.1 Analytic Results

Since the Pareto gamut is a new object, there are no established benchmarks to verify our discovery algorithm. We build our own test suite by adapting fixed-context benchmarks to a contextual setting. We also derive two methods to verify the correctness of our discovered Pareto gamut: (1) comparison against analytic solutions for the Pareto set/front at any given context $\mathbf{z}^*$, and (2) validation against Proposition 4.6 (Lower Envelope Optimality), which only requires computing a Pareto front rather than a Pareto gamut.

Modified ZDT Functions. The ZDT test suite [Zitzler et al. 2000] contains 5 continuous-domain problems for multi-objective optimization, with carefully designed challenges like non-convexity, multi-modality, and sparse solution density along the front. Each ZDT function has $d = 2$ performance metrics that depend on m design variables $\boldsymbol{y} = (y_1, \ldots, y_m)$. The original problems do not

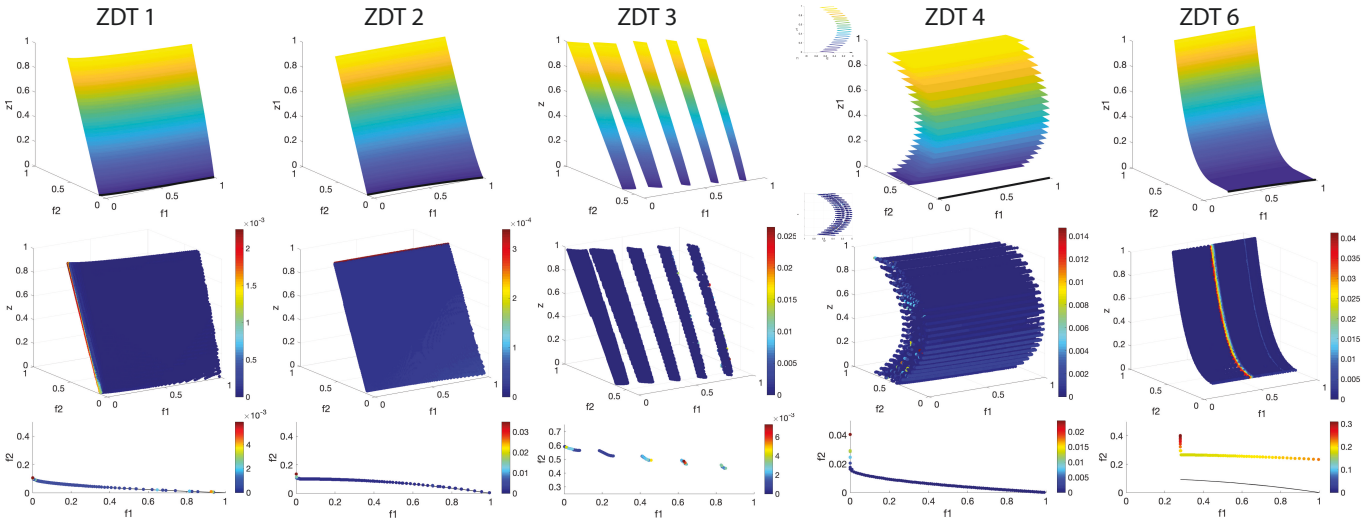


Fig. 7. Our suite of contextual ZDT solutions. **(Top)** Analytic ground truth for the Pareto gamut, with $d = 2$ and $C = 1$. Colored by context value. **(Middle)** Our discovered gamut, colored by distance to the ground truth. **(Bottom)** Flattened lower envelope of our solution (scatter); colored by distance to the ground truth Pareto front of the corresponding free-context problem(black line). All discovered solutions use $N_c = 200$ cells along each dimension of the buffer.

contain context parameters ($C = 0$). To test $C \geq 1$, we treat some of the m design variables as context parameters. We let $m = D + C$ and define $F(\mathbf{x}, \mathbf{z}) : \mathbb{R}^D \times \mathbb{R}^C \rightarrow \mathbb{R}^d \times \mathbb{R}^C$. We preserve the original function definitions with respect to $\hat{\boldsymbol{y}} = (x_1, \ldots, x_D, z_1, \ldots z_C) \in \mathbb{R}^m$. The only difference is our treatment of $\mathbf{z}$, whose values are explored rather than optimized. Fig. 7 (middle) shows strong agreement between our discovered Pareto gamut and the analytic ground truth.

Fig. 7 (bottom) also validates our results against Proposition 4.6 by examining our Pareto gamut's flattened lower envelope. By construction, the corresponding free-context problem $H_F(\boldsymbol{y})$ is the original ZDT problem over m design variables, and its Pareto front $H_F(\mathcal{P})$ is the well-established ZDT solution. As expected, our Pareto gamut's flattened lower envelope matches each established solution well.

Despite strong agreement overall, two discretization artifacts invite discussion. ZDT4 highlights that our discretization must be fine enough to capture the gamut's variation as a function of z. Our method identifies the correct solution manifold, but the Pareto fronts from lower-performing contexts within each cell appear "dominated," so they are left out. The ZDT6 lower envelope discrepancy is because the solution density shrinks exponentially near the context-free front; thus, the fixed-context front at our smallest cell center ($z = 0.0025$) is still far from the ground truth. This is resolved by generating buffer samples along the exact context boundary ($z = 0$).

Fourier Benchmark. The ZDT functions are helpful for validation against an established benchmark. However, the solution $\mathcal{P}$ is a single linear manifold in all 5 cases. We stress test our algorithm with a contextual Fourier benchmark, for which $\mathcal{P}$ is non-linear and disconnected (Fig. 8). Each objective function $f_i(\mathbf{x}, \mathbf{z})$ is

$$f_i(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{D} \sum_{j=1}^{N} \left[\alpha_{ij} \cos(3jx_i + \beta_{ij}^{2z_1}) \right] + \sum_{m=1}^{C} \sum_{n=1}^{N} \left[\gamma_{mn} \cos(3nz_k + \mu_{mn}^{2z_1}) \right],$$

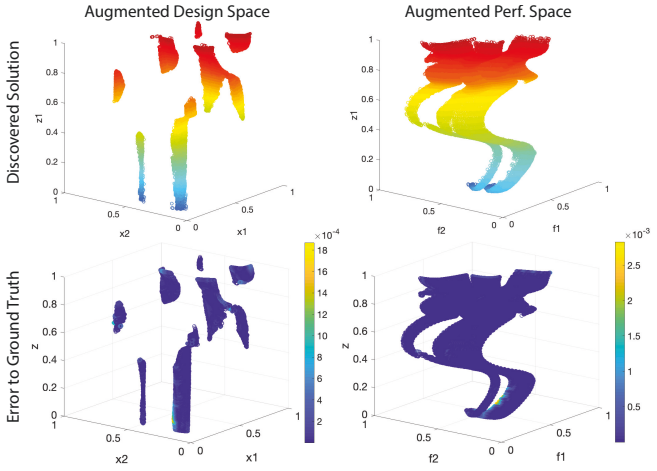


Fig. 8. Fourier benchmark solution. **(Left)** Our discovered Pareto gamut (bottom) and its pre-image (top), colored by context value. **(Right)** Error between our gamut/pre-image and 100 precomputed fixed-context solutions. Strong agreement highlights our algorithm's empirical convergence, even in the case of a highly disconnected, non-linear pre-image $\mathcal{P} \subseteq \mathcal{X}$.

where N is the desired polynomial order, and $\alpha_{ij}, \gamma_{mn} \in [0, 1]$ and $\beta_{ij}, \mu_{mn} \in [0, \pi]$ are randomly chosen coefficients.

6.2 Engineering Design Examples

We now present several engineering examples to demonstrate the unique practical benefits of the Pareto gamut.

Turbine. Fig. 1 illustrates a simple wind turbine with $D = 3$ design variables: radius r (central shaft to blade tip), height h (perpendicular to plane of rotation), and blade pitch angle α. The $d = 2$ performance

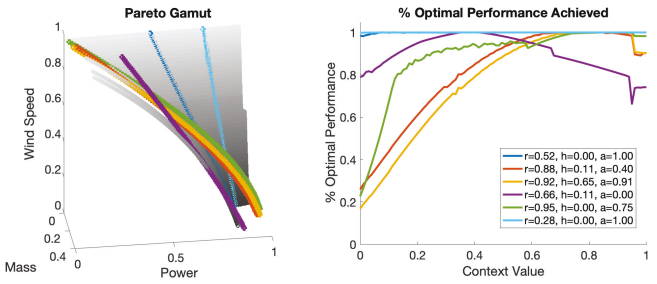


Fig. 9. **(Left)** After computing the Pareto gamut (gray) for turbines over multiple wind speeds, engineers are able to visualize the performance of a particular design (colored line) relative to the achievable optimal performance in any context. **(Right)** Numerical performance curves for each individual design over the context space, providing additional insight about each design's desirability over the context range.

metrics are mass and power output, $P = \frac{1}{2}C_p\rho\pi r^2 v^3$, where ρ is air density and v is wind speed. The coefficient of performance C_p is an empirically measured function of the turbine's geometry and environment, which we approximate using data from Johnson [2006, Fig. 9]. We use v as our context parameter ($C = 1$).

The optimal turbine designs are highly dependent on v (see Fig. 1): small radii are preferable at low wind speeds, but the ceiling on useful radius values increases along with v. Of course, v is dynamic in practice, so final designs must perform well under several contexts.

As shown in Fig. 9, the Pareto gamut can highlight designs that are particularly strong (or weak) in various contexts. For a query design $q \in \mathbb{R}^D$, we compute $F_{\mathbf{z}^*}(q)\ \forall \mathbf{z}^* \in \mathcal{Z}$, and plot these values relative to the Pareto gamut for qualitative inspection. We also quantify the performance "sacrifice" that q incurs at each context, by finding the closest point $q^* \in F_{\mathbf{z}^*}(\mathcal{P}_{\mathbf{z}^*})$ and computing the hypervolume ratio $\text{hv}(F_{\mathbf{z}^*}(q))/\text{hv}(q^*)$. This gives a proxy for q's performance as a percentage of the achievable optima in each context.

This data allows for more informed design choices. For example, the blue designs in Fig. 9 may be most suitable for sites with wide variation in wind speeds, as they are Pareto-optimal under every context. However, the green design befits areas with reliably high wind, as it offers superior power output and perfect optimality at high speeds, plus near-optimality across plausible medium speeds.

Although this discussion of characteristic wind speeds seems reminiscent of the "representative context" approach from §1, the Pareto gamut provides considerably more information. For example, consider a site with wind $v \in [4.4, 5.2]$m/s (normalized value $z = [0.2, 0.6]$). Computing the Pareto front for the average context $z = 0.4$ then selecting the design with maximal power yields the purple solution in Fig. 9. This solution is optimal at the average context, but its performance relative to the achievable optima falls off dramatically to either side. Thus, the turbine may perform sub-optimally a large majority of the time. This sensitivity is only well-exposed by the Pareto gamut, which emphasizes its ability to provide unique insights that were not readily observable with existing methods.

Bicycle Rocker and Seat Stay. As discussed in §1.1, context plays a critical role for hierarchically-designed assemblies such as a full-suspension bicycle (Fig. 2, 10), because high-level assembly decisions (contexts) affect the optimal designs and achievable performance of each individual part comprising the assembly. For example, consider shifting the shared pivot in Fig. 10 upward: this expands the convex hull of the both adjacent parts, altering their achievable trade-offs. However, the higher pivot also produces drastically different suspension characteristics, as shown by the anti-squat curves [Faulkner 2014; Roberts 2020]. Experts agree that "the very first thing that needs to be done is a proper layout of the suspension kinematics to ride properly, then visual and manufacturing concerns come in after" [Benedict 2018]. However, engineers care about the performance of the assembly *and* the individual components, so they must revise the assembly decisions until they admit sufficiently performant components as well. This can lead to long iteration cycles with existing tools, because engineers cannot easily access information about the final parts during the layout phase, as every pivot location would require an independent optimization of the adjacent parts.

Our method naturally facilitates this multi-level exploration, by treating each assembly-level decision as context that is shared by its adjacent components, then computing the Pareto gamut for each part independently. For proof of concept, we consider two pieces of the suspension linkage: the rocker and the seat stay (Fig. 10). Each component has $D = 3$ design variables, $d = 2$ performance metrics (approximate compliance and mass), and $C = 1$ context (shared pivot point). We discover the Pareto gamut for each part independently. Then, as the shared pivot location is adjusted, we can provide instant access to the appropriate fixed-context Pareto front for each affected part. By exposing the downstream implications of each assembly-level choice *in conjunction with* the high-level metrics like anti-squat, the Pareto gamut could permit intuitive, holistic assembly design at a fraction of the traditional overhead.

Alternatively, one might circumvent hierarchical design issues by optimizing over the entire assembly at once, with additional design variables for the pivot locations to be optimized w.r.t. metrics like anti-squat. This global approach has two critical drawbacks. Even in the simple rocker-stay case, the combined problem requires $d \geq 5$ metrics (1 for anti-squat plus 2 each from the stay/rocker), which is considered intractable for *a priori* optimization methods [Schütze et al. 2019]. Engineers could omit or scalarize some objectives to improve tractability, but this tuning process is tedious and biases the results. More importantly, suspension behaviors like anti-squat are highly subjective: experts agree that "there is no set equation for the right amount of anti-squat. All riders have different riding styles. So, some may prefer more anti squat where others prefer less. [...] Different terrain benefits from different designs, too" [Benedict 2018]. Thus, it does not make sense to optimize the anti-squat or, by extension, the pivot positions; they must be explored for each unique scenario.

The Pareto gamut addresses both of these drawbacks by decomposing the complex design problem into intuitive sub-assemblies linked by adjustable constraints (contexts). Since each component is self-contained (as in the traditional hierarchical approach), its optimization is tractable *and* comprehensive, as there is no need to omit or scalarize metrics of interest. Moreover, by optimizing the component for every assembly-level context, the Pareto gamut could permit rapid customization of bicycles with high performance at both the component and assembly levels.

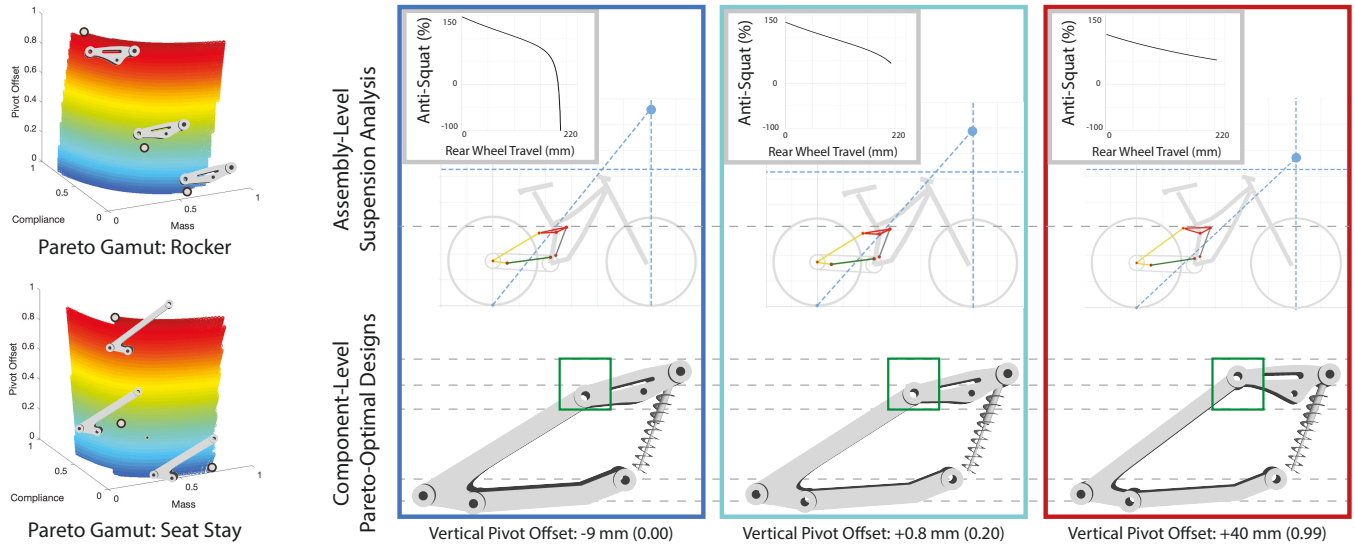


Fig. 10. (**Left**) Pareto gamuts for two parts of a bicycle suspension, over a shared context indicating the vertical position of their mutual pivot point **(Right, green boxes)**. Engineers explore many pivot positions to adjust high-level assembly characteristics like anti-squat (**R,Top**). The part-wise Pareto gamuts offer direct access to the optimized designs for each pivot choice (**R,Bottom**), which provides additional insight about the impact of each assembly-level decision.

Solar Roofing. Context parameters are helpful for solar roof design (Fig. 11), where architects must balance quantifiable metrics like solar output against subjective metrics like the house aesthetics and its location on a land plot. House location also affects the roof's power output, which depends on e.g. the relative position of the sun and any occluders, like trees. We optimize our roof for $d = 2$ performance metrics (power output in the morning and the evening), over a context ($C = 1$) that indicates the building's global orientation. Our design space is a height field of size $D = 100$. We encourage smooth roofs by adding a regularizer to each performance metric.

As shown in Fig. 11, all orientations exhibit similar performance tradeoffs, so none have a quantifiable advantage. However, the Pareto-optimal design that yields a given performance in each context varies dramatically, which affects the aesthetic goals. By efficiently exposing the functionally-optimal designs for all contexts, our Pareto gamut could allow architects to focus on more aesthetic implications that are difficult to quantify. For example, on a property with trees along the east (which occlude morning sun), architects could peruse the evening-focused designs across many orientations to find one that satisfies their aesthetic goals. This generality also means that the *same* Pareto gamut can be used in many distinct situations, without needing to design and optimize a new model for each environment. In particular, the Pareto gamut discussed above could also be used to plan for a lot with trees to the west, by exploring the morning-efficient designs over several contexts.

This example also confirms our algorithm's ability to navigate large design spaces. Although we defer our extended performance comparisons to §6.3, we highlight that for $D = 100$ our approach achieves comparable solution quality with at least 200× fewer function evaluations and 45× shorter runtime than other state-of-the-art multi-objective optimization approaches. We tested this problem with up to $D = 400$ variables, and found similar performance trends in all cases. Thus, the Pareto gamut is ideal for problems with large design spaces and few metrics/contexts, as its complexity depends primarily on d and C. Our discovery and visualization are (largely) agnostic to D, excepting the fact that our implementation uses explicit Jacobian and Hessian matrices. The latter grows quadratically in D, so for $D > 200$ the memory requirements exceed our available RAM and force us to read from disk, increasing the runtime. To alleviate this, we could adapt recent continuation schemes for Pareto front discovery that permit millions of design variables [Ma et al. 2020]. However, even without this improvement, our capacity is large enough to support many engineering problems.

Lamp. The parametric lamp shown in Fig. 12 tests our algorithm on a higher-dimensional Pareto gamut, with $d = 3$ and $C = 1$. The lamp has $D = 21$ design variables controlling the length and orientation of each rod. The performance metrics are mass, instability (L_2 distance between the base center and the lamp's projected center of mass), and illumination of a pre-determined focal point $p \in \mathbb{R}^3$ (average distance of the lamp's bulbs to p). We use the focal point's height p_z as our context, to expose the achievable trade-offs for many height variations, from desk- to floor-lamp.

As expected, the Pareto set includes increasingly taller lamps as p_z increases. Some designs also persist throughout all contexts, such as the lightweight, stable design shown in Fig. 12. Our method improves the design consistency across contexts, because we propagate optimal designs to *all* relevant contexts upon discovery in *some* context, rather than discovering them independently on each front.

This example shows that our method scales in d at least as well as fixed-context approaches. Both cases are subject to the curse of dimensionality, as the potential surface area of each Pareto front (and thus, the size of the point set required for approximation) grows exponentially in d. Fixed-context *a posteriori* methods are considered

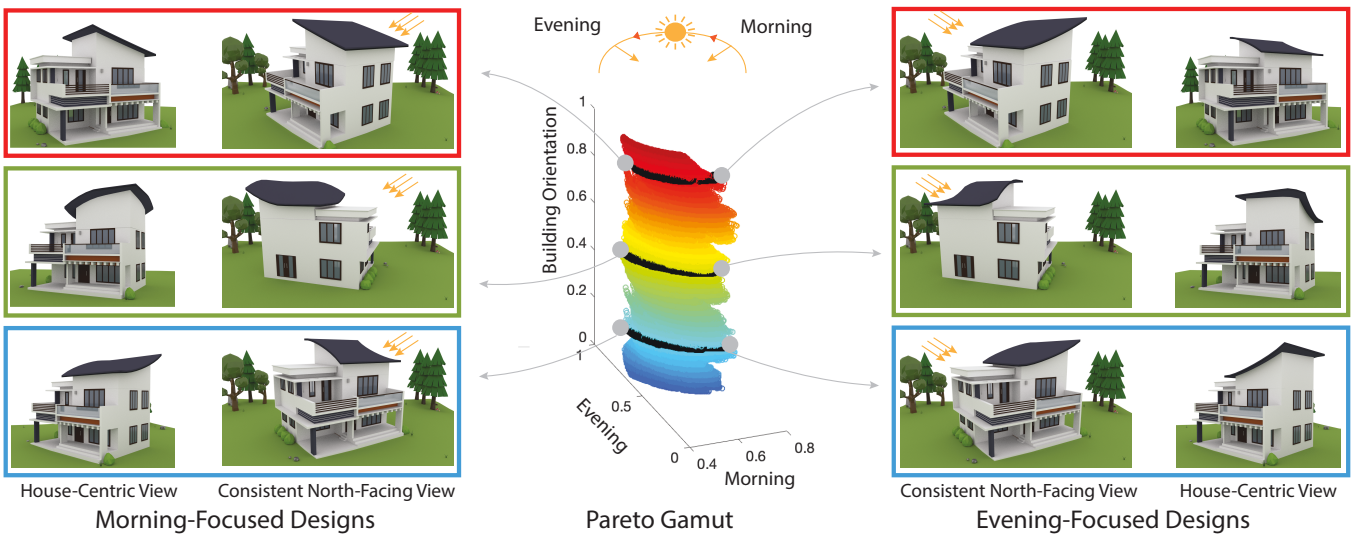


Fig. 11. **(Center)** Pareto gamut for a solar roof, optimized for output at different times of day, over a context that controls the building's orientation. **(Left/Right)** Renderings of the single most extreme point on each end of the highlighted fixed-context fronts. Although the achievable performance of each morning- or evening-focused solution is roughly equivalent, the design that realizes this performance varies dramatically in different contexts.

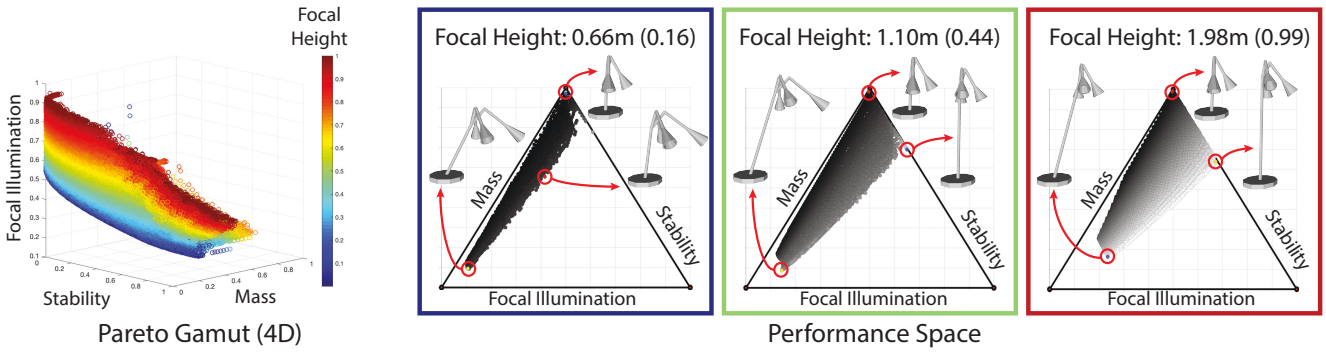


Fig. 12. **(Left)** 4D Pareto gamut for our lamp. Performance values are plotted spatially, and color indicates the context. **(Right)** 3D fixed-context Pareto fronts, embedded in 2D space using barycentric coordinates. High-performing designs w.r.t. each metric are located near the labeled edge. Grayscale value indicates the lamp heights. As expected, higher focal points yield taller lamps (light gray), but the most lightweight, stable lamps (dark gray) persist through all contexts.

intractable for $d > 3$ [Schütze et al. 2019]. Since the Pareto gamut has dimension $d + C$, it may initially seem that engineers will be forced to choose between exploring a third performance metric or exploring a context parameter. Our lamp dismisses this concern by providing a dense gamut for $d + C = 4$, with no algorithmic modifications and a reasonable computational cost (see §6.3).

Bicopter Controller. To demonstrate our algorithm with multiple context parameters ($C = 2$), we optimize the controller for a 2D bicopter with the morphology (bicopter length ℓ and material density ρ) as our context (Fig. 13). We optimize for $d = 2$ common performance metrics: ability to reach a target, and overall energy usage [Tedrake 2020]. Our design space has $D = 32$ variables, denoting the actuation sequence for each of the 2 propellers over 16 time steps.

Fig. 13 demonstrates our successful gamut discovery, even with many highly-dependent design variables over 10,000 fixed contexts. We validate the results intuitively: long, heavy bicopters require more energy-intensive controllers to reach the goal; thus, the trade-offs get worse and the hypervolume decreases. The energy-efficient controllers leave long/heavy bicopters farther from the goal, because they require a higher baseline energy to overcome gravity and our experiment does not consider collision with a ground plane.

This experiment also shows the Pareto gamut's promise for other exciting domains, including co-optimization of robot geometry and controls for various tasks. Typically, at least one element is presumed fixed: e.g., robot morphologies and controllers are optimized for a particular task/terrain [Ha et al. 2018; Zhao et al. 2020], or controllers are optimized to fit a specific robot design [Lee et al. 2020; Spielberg et al. 2017]. By exploring a range of robot morphologies or terrains, our method could indicate e.g. strong "all-purpose" morphologies that admit highly-performant controllers under many conditions.

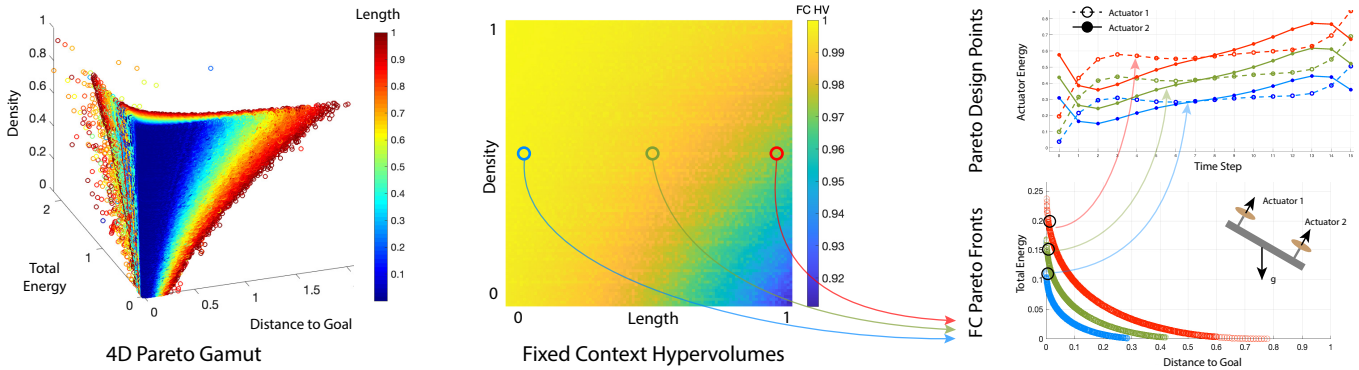


Fig. 13. **Left:** 4D Pareto gamut for the 2-context bicopter controller design problem. **Center:** Hypervolume for each of the 100^2 fixed context Pareto fronts on the gamut. As expected, the hypervolume (i.e., achievable performance) decreases as length or density increase. **Bottom Right:** Several fixed-context Pareto fronts extracted from the gamut; force analysis of the bicopter. **Top Right:** Actuation trajectories for several Pareto-optimal design points. As expected, longer copters require more energy to reach the same distance-to-goal.

6.3 Performance Benchmarks

We compare our algorithm to two state-of-the-art fixed-context approaches: the closely-related approach from Schulz et al. [2018] and a popular evolutionary algorithm, NSGA-II [Deb et al. 2002]. To "approximate" the gamut with these algorithms, we compute Pareto fronts at N evenly-spaced fixed-context values $\mathbf{z} \in \mathcal{Z}$. We report the overall runtime and number of function evaluations for each method, and provide normalized "per-context" costs for fair comparison on each metric. Solution quality is measured by the hypervolume of fixed-context fronts that are either explicitly computed by a fixed-context method or extracted from our gamut, respectively.

As shown in Table 1 (where $N = 5$), all methods yield consistent hypervolumes, which validates the quality of fixed-context fronts extracted from our gamut. Moreover, our method consistently outperforms the others w.r.t. average time and function evaluations per context. The one exception (bicopter, # evals) is due to premature convergence by NSGA, which exhibits notably lower hypervolume. Our method finds the best hypervolume using half as many samples as Schulz et al. due to our generalized first-order approximation, which amortizes the optimization cost over the entire context range.

For further validation, we benchmark the Turbine example using $N = 5, 10, \ldots, 200$ (Fig. 14). To achieve an information density comparable to our Pareto gamut, both previous works require an order of magnitude more function evaluations. Moreover, the cost of fixed-context computation exceeds the cost of our full Pareto gamut once the user seeks $N > 10$ contexts on average, or 35 in the worst case. This offers immediate practical relevance, as engineers already consider this many contexts to ensure robust designs.

To approximate the Pareto gamut and its pre-image, one may also consider interpolating fixed-context solutions. If the Pareto sets and fronts vary smoothly w.r.t. $\mathbf{z}$, as in the Turbine, this is plausible. However, sufficient smoothness in both spaces is not guaranteed: the performance oscillations for ZDT4 would require many Pareto fronts to avoid aliasing, and the solar roof's drastic design changes would require many sets. This is difficult to gauge a priori, which casts doubt on results interpolated from small N. Thus, our Pareto gamut offers higher confidence for a comparable, if not lower, cost.

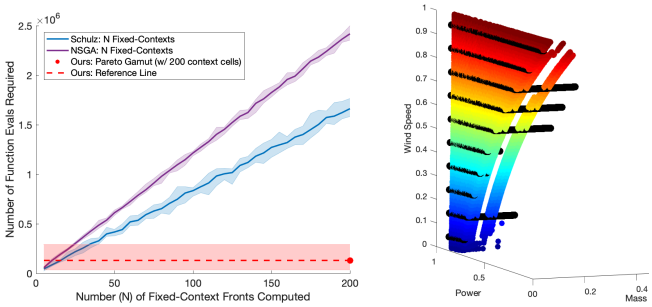


Fig. 14. Comparison on Turbine example. **(Left)** Function evaluations required to compute N evenly-spaced fixed-context Pareto fronts with prior work (blue, purple) vs. our Pareto gamut with $N = N_c = 200$ context cells (red dot). Each experiment was run 10 times. Solid lines indicate the mean value, and shaded areas indicate the min/max. The dashed red line emphasizes the critical point ($N = 10$, evals $\approx 10^5$) at which computing the full Pareto gamut becomes more efficient than repeated fixed-context computations with either method. **(Right)** Pareto gamut density achieved by our method (colored by context value) vs. Schulz et al. [2018] (black curves), assuming $\sim 10^5$ function evaluations. In agreement with the critical point at left, Schulz et al. [2018] computes 10 fixed-context fronts with this budget.

7 DISCUSSION AND LIMITATIONS

The Pareto gamut offers many new possibilities, but it is not without its challenges. For example, the performance metric normalization can be problematic if the context drastically alters the achievable performance range. For the turbine over a physically common wind range of $v \in [2, 20]$m/s, the achievable power outputs span 6 orders of magnitude. Since power scales cubically with the context (v), the Pareto fronts for low wind speeds occupy a small fraction of the available performance space – and thus, a small fraction of the buffer cells. This condensed front falsely implies that there are no tradeoffs at low wind speeds; in reality, the tradeoffs have been lost within our discretization. To address this, we limit wind speeds to $v \in [4, 6]$m/s so the discrepancy in achievable values is not detrimentally large. It would be worthwhile to explore this further, to increase the context

Table 1. Performance comparison illustrating our method's amortized advantage (gray columns) over two state-of-the-art fixed-context approaches: Schulz et al. [2018] and NSGA-II [Deb et al. 2002]. We compare one run of our algorithm (yielding the Pareto gamut) to 5 runs of Schulz/NSGA (each yielding a single fixed-context front). **Our algorithm provides ≥ 40× denser coverage of the context space without sacrificing proportional efficiency/effectiveness**, as evidenced by our gamut's comparable hypervolume at each fixed-context front, and the superior efficiency wrt function evaluations and time per context. We use an official Python benchmark [Blank and Deb 2020] for NSGA-II, and a MATLAB implementation for Schulz et al. and Ours. All experiments were run on an Intel Core i7 CPU. For $d + C < 4$, reported results are averaged over 10 experimental runs; for $d + C = 4$, we report 1 representative run.

	Turbine ($D = 3$, $d = 2$, $C = 1$)								
	Fixed Context Hypervolume					#Function Evals		Time (s)	
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9	Total	Per z†	Total	Per z†
NSGA	0.18	0.27	0.40	0.58	0.81	61,000	12,200	8.75	1.75
Schulz	0.18	0.27	0.40	0.58	0.81	44,538	8,908	37.9	7.58
Ours	0.16	0.26	0.37	0.68	0.83	124,713	**624**	52.1	**0.26**

	BikeRocker ($D = 3$, $d = 2$, $C = 1$)								
	Fixed Context Hypervolume					#Function Evals		Time (s)	
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9	Total	Per z†	Total	Per z†
NSGA	0.77	0.74	0.72	0.69	0.66	63,000	12,600	13.5	2.70
Schulz	0.76	0.74	0.70	0.68	0.66	291,268	58,253	34.4	6.88
Ours	0.77	0.74	0.71	0.69	0.66	1,877,917	**9,390**	489.4	**2.45**

	Lamp ($D = 21$, $d = 3$, $C = 1$)								
	Fixed Context Hypervolume					#Function Evals		Time (s)	
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9	Total	Per z†	Total	Per z†
NSGA	0.58	0.58	0.56	0.51	0.44	5,000,000*	1,000,000*	1,298.1	259.6
Schulz	0.57	0.58	0.58	0.58	0.57	25,744,440	5,148,888	20,160.5	4,032.1
Ours	0.57	0.60	0.60	0.59	0.56	683,774	**3,419**	28,305.8	**141.5**

	Solar Roof ($D = 100$, $d = 2$, $C = 1$)								
	Fixed Context Hypervolume					#Function Evals		Time (s)	
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9	Total	Per z†	Total	Per z†
NSGA	0.27	0.25	0.23	0.27	0.25	5,000,000*	1,000,000*	2,046.2	409.23
Schulz	0.30	0.31	0.31	0.31	0.30	1,013,233	202,647	1,427.3	285.5
Ours	0.30	0.31	0.31	0.30	0.30	273,632	**1,368.2**	1,225.9	**6.1**

	Bicopter with Constant Density $\rho = 0.5$ ($D = 32$, $d = 2$, $C = 1$)								
	Fixed Context Hypervolume					#Function Evals		Time (s)	
	z=0.1	z=0.3	z=0.5	z=0.7	z=0.9	Total	Per z†	Total	Per z†
NSGA	0.89	0.89	0.86	0.87	0.75	43,150	**8,630**	49.1	9.82
Schulz	0.98	0.97	0.96	0.96	0.95	231,030	46,206	956.5	191.3
Ours	0.99	0.98	0.98	0.97	0.96	5,678,502	28,392	868.5	**4.3**

	Bicopter with Density as 2^{nd} Context ($D = 32$, $d = 2$, $C = 2$)								
	FC HV for $\mathbf{z} = (\ell, \rho) = (\ell, 0.5)$					#Function Evals		Time (s)	
	ℓ=0.1	ℓ=0.3	ℓ=0.5	ℓ=0.7	ℓ=0.9	Total	Per z†	Total	Per z†
NSGA	0.89	0.89	0.86	0.87	0.75	43,150	8,630	49.1	9.82
Schulz	0.98	0.97	0.96	0.96	0.95	231,030	46,206	956.5	191.3
Ours	0.99	0.99	0.98	0.98	0.97	30,782,146	**3,078**††	67,629	**6.76**††

† Divide Total by 5 for NSGA-II and Schulz (number of fixed contexts computed); divide by 200 for Ours (number of buffer cells along context dimension).
†† When $C = 2$, divide Total by 100^2 (number of buffer cells along context dimensions) for Ours. * NSGA-II is capped at 1,000,000 function evaluations per fixed context.

parameter range while preserving the integrity of each fixed-context Pareto front and its local change as a function of the context.

The feasible design space may also change in response to the context. For instance, as the angle of an L-bracket shrinks (see Fig. 2), the maximum material thickness must be reduced. In this paper, we restrict each design variable to the largest range that is admissible over *all* contexts. This assumption is made for simplicity: our theory permits linear and non-linear constraints on the design variables, so the user could go beyond box constraints if desired. However, codifying the permissible range for each context can be challenging. Future work could improve our pipeline by automatically discovering the feasible design space across varying contexts.

As discussed in §6.2, full Pareto gamut discovery is likely only feasible for low-dimensional augmented performance spaces (d+$C \leq 4$). To permit $d + C > 4$, it would be interesting to build an interactive optimization method for Pareto gamut discovery and traversal. Our first-order approximation offers a strong foundation for interactive methods, as it can generate feasible search directions about any known-optimal point on the fly. Then, a set of optimally-perturbed samples could be presented to the user for selection. Alternatively, the user could specify a desired design or context perturbation $\boldsymbol{u} \in \mathcal{X}$ (e.g. longer blades, lower wind speed); by projecting u onto our exploration subspace, we could find perturbations $w \in \mathcal{X}$ that satisfy user guidance while preserving optimality. We could also support user-guided *performance* perturbations (e.g. better power) by considering a similar approach with $DF \cdot v$. Schütze et al. [2019] demonstrate this interactive approach for fixed-context optimization, by building on the first-order approximation of Martín and Schütze [2018]. Our first-order approximation could be integrated into a similar framework, to provide novel features such as (1) built-in sensitivity analysis w.r.t. environment parameters, uncertain measurements, etc. and (2) an efficient way to find an optimal design $\hat{\mathbf{x}}$ in context $\hat{\mathbf{z}}$ given a known-optimal solution in *some* context $(\mathbf{x}^*, \mathbf{z}^*)$.

Finally, future work could expand on our results to make Pareto gamut exploration more practical and accessible. Our current examples are implemented with hand-designed meshes and analytic performance metrics that are differentiated symbolically using MATLAB. However, our method could easily be extended to more complicated models and metrics by interfacing with recent differentiable engineering frameworks, such as XCAD [Hafner et al. 2019]. Another useful direction would build upon our visualization prototypes to provide an intuitive user interface for Pareto gamut exploration.

8 CONCLUSION

To discover effective engineering solutions, it is critical to consider potential designs' performance w.r.t. multiple metrics *and* contexts. Existing optimization schemes only provide insight into one context at a time, causing context exploration to be tedious and time-consuming. We address this by incorporating variable contexts into the multi-objective optimization process, by formalizing the notion of a Pareto gamut and developing an algorithm that discovers the Pareto gamut directly. Our method yields dense coverage of the context range, with high-quality fixed-context Pareto fronts and relatively little additional computational cost. As evidenced by our examples, the Pareto gamut is well-poised to improve the functional design process by providing unprecedentedly thorough information and laying the groundwork for a more holistic design approach. Thus, our work serves as a critical first step toward comprehensive design optimization for practical engineering pursuits.

ACKNOWLEDGMENTS

The authors thank Harrison Wang for his early contributions; Hannes Hergeth for his insights and visualization tool; Paul Zhang, David Palmer, and Ed Chien for their mathematical insights; Tim Erps, Mike Foshey, Andy Spielberg, Beichen Li, and James Minor for engineering examples; and Cartemere for the bike illustration in Fig. 2.

This material is based upon work supported by the National Science Foundation (NSF) (Grant No. IIS-1955697), the NSF Graduate Research Fellowship (Grant No. 1122374), the Intelligence Advanced Research Projects Agency (Grant No. 2019-19020100001), and the Defense Advanced Research Projects Agency (Grant No. FA8750-20-C-0075). The MIT Geometric Data Processing group acknowledges the generous support of Army Research Office grant W911NF2010168, of Air Force Office of Scientific Research award FA9550-19-1-031, of NSF grant IIS-1838071, from the CSAIL Systems that Learn program, from the MIT–IBM Watson AI Laboratory, from the Toyota–CSAIL Joint Research Center, from a gift from Adobe Systems, from an MIT.nano Immersion Lab/NCSOFT Gaming Program seed grant, and from the Skoltech–MIT Next Generation Program.

REFERENCES

Johannes Bader and Eckart Zitzler. 2011. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation* 19, 1 (2011), 45–76. https://doi.org/10.1162/EVCO_a_00009

Tyler Benedict. 2018. *Suspension Tech: What is Anti-Squat?* https://bikerumor.com/2018/09/13/suspension-tech-what-is-anti-squat/

Gaurav Bharaj, David I. W. Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. 2015. Computational Design of Metallophone Contact Sounds. *ACM Trans. Graph.* 34, 6, Article 223 (Oct. 2015), 13 pages. https://doi.org/10.1145/2816795.2818108

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and Fabrication of Materials with Desired Deformation Behavior. *ACM Transactions on Graphics* 29, 4, Article 63 (July 2010), 10 pages.

Julian Blank and Kalyanmoy Deb. 2020. pymoo: Multi-objective Optimization in Python. arXiv:cs.NE/2002.04504

Michael Bogomonly. 2019. *Partner Spotlight: Using Generative Design With Onshape and ParaMatters.* https://www.onshape.com/cad-blog/partner-spotlight-using-generative-design-with-onshape-and-paramatters

Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. 2014. An Asymptotic Numerical Method for Inverse Elastic Shape Design. *ACM Trans. Graph.* 33, 4, Article 95 (July 2014), 11 pages. https://doi.org/10.1145/2601097.2601189

Jin-Hee Cho, Yating Wang, Ing-Ray Chen, Kevin S. Chan, and Ananthram Swami. 2017. A Survey on Modeling and Optimizing Multi-Objective Systems. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1867–1901. https://doi.org/10.1109/comst.2017.2698366

Derek Covill, Steven Begg, Eddy Elton, Mark Milne, Richard Morris, and Tim Katz. 2014. Parametric Finite Element Analysis of Bicycle Frame Geometries. *Procedia Engineering* 72 (2014), 441 – 446. https://doi.org/10.1016/j.proeng.2014.06.077 The Engineering of Sport 10.

Indraneel Das and J. E. Dennis. 1998. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization* 8, 3 (1998), 631–657.

Kalyanmoy Deb and Himanshu Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part 1: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (Aug. 2014), 577–601. https://doi.org/10.1109/TEVC.2013.2281535

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Trans. Evol. Comp* 6, 2 (April 2002), 182–197.

Yue Dong, Jiaping Wang, Fabio Pellacini, Xin Tong, and Baining Guo. 2010. Fabricating Spatially-varying Subsurface Scattering. *ACM Transactions on Graphics* 29, 4, Article 62 (July 2010), 10 pages.

Tao Du, Adriana Schulz, Bo Zhu, Bernd Bickel, and Wojciech Matusik. 2016. Computational Multicopter Design. *ACM Trans. Graph.* 35, 6, Article 227 (Nov. 2016), 10 pages. https://doi.org/10.1145/2980179.2982427

Matt Faulkner. 2014. *Suspension Linkage Kinematics: The Basics of Anti-Squat and Pedal Kickback.* https://www.ridingfeelsgood.com/suspension-linkage-kinematics-basics-anti-squat-pedal-kickback/

Sehoon Ha, Stelian Coros, Alexander Alspach, Joohyung Kim, and Katsu Yamane. 2018. Computational co-optimization of design parameters and motion trajectories for robotic systems. *The International Journal of Robotics Research* 37, 13-14 (2018), 1521–1536. https://doi.org/10.1177/0278364918771172 arXiv:https://doi.org/10.1177/0278364918771172

Christian Hafner, Christian Schumacher, Espen Knoop, Thomas Auzinger, Bernd Bickel, and Moritz Bächer. 2019. X-CAD: Optimizing CAD Models with Extended Finite Elements. *ACM Trans. Graph.* 38, 6, Article 157 (Nov. 2019), 15 pages. https://doi.org/10.1145/3355089.3356576

C. Hillermeier. 2001a. Generalized Homotopy Approach to Multiobjective Optimization. *Journal of Optimization Theory and Applications* 110, 3 (01 Sep 2001), 557–583.

Claus Hillermeier. 2001b. *Nonlinear multiobjective optimization: a generalized homotopy approach.* Vol. 135. Springer Science & Business Media.

Gary Johnson. 2006. *Wind Energy Systems (electronic edition).* http://www.ece.k-state.edu/people/faculty/gjohnson/files/Windbook.pdf Original work published 1994.

William Karush. 1939. *Minima of Functions of Several Variables with Inequalities as Side Constraints.* Master's thesis. Dept. of Mathematics, University of Chicago, Chicago, Illinois.

Yuki Koyama, Issei Sato, and Masataka Goto. 2020. Sequential Gallery for Interactive Visual Design Optimization. *ACM Trans. Graph.* 39, 4, Article 88 (July 2020), 12 pages. https://doi.org/10.1145/3386569.3392444

H. W. Kuhn and A. W. Tucker. 1951. Nonlinear Programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability.* University of California Press, Berkeley, Calif., 481–492. https://projecteuclid.org/euclid.bsmsp/1200500249

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. 2020. Learning quadrupedal locomotion over challenging terrain. *Science Robotics* 5, 47 (2020). https://doi.org/10.1126/scirobotics.abc5986 arXiv:https://robotics.sciencemag.org/content/5/47/eabc5986.full.pdf

Dingzeyu Li, David I.W. Levin, Wojciech Matusik, and Changxi Zheng. 2016. Acoustic Voxels: Computational Optimization of Modular Acoustic Filters. *ACM Trans. Graph.* 35, 4 (2016). https://doi.org/10.1145/2897824.2925960

Pingchuan Ma, Tao Du, and Wojciech Matusik. 2020. Efficient Continuous Pareto Exploration in Multi-Task Learning. In *Proceedings of the 37th International Conference on Machine Learning.*

R. T. Marler and J. S. Arora. 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 26, 6 (01 Apr 2004), 369–395. https://doi.org/10.1007/s00158-003-0368-6

Adanay Martín and Oliver Schütze. 2018. Pareto Tracer: a predictor-corrector method for multi-objective optimization problems. *Engineering Optimization* 50, 3 (2018), 516–536.

David Meignan, Sigrid Knust, Jean-Marc Frayret, Gilles Pesant, and Nicolas Gaud. 2015. A Review and Taxonomy of Interactive Optimization Methods in Operations Research. *ACM Trans. Interact. Intell. Syst.* 5, 3, Article Article 17 (Sept. 2015), 43 pages. https://doi.org/10.1145/2808234

A. Messac, A. Ismail-Yahaya, and C.A. Mattson. 2003. The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization* 25, 2 (01 Jul 2003), 86–98.

Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case Stress Relief for Microstructures. *ACM Trans. Graph.* 36, 4, Article 122 (July 2017), 16 pages. https://doi.org/10.1145/3072959.3073649

PHeller. 2014. *Horst-Link Pivot Placement and Pedaling Efficiency (Norco ART).* https://www.pinkbike.com/u/PHeller/blog/horst-link-pivot-placement-and-pedaling-efficiency-norco-art.html

Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. 2013. Make It Stand: Balancing Shapes for 3D Fabrication. *ACM Trans. Graph.* 32, 4, Article 81 (July 2013), 10 pages. https://doi.org/10.1145/2461912.2461957

J. Rakowska, R.T. Haftka, and L.T. Watson. 1991. Tracing the efficient curve for multi-objective control-structure optimization. *Computing Systems in Engineering* 2, 5 (1991), 461 – 471. https://doi.org/10.1016/0956-0521(91)90049-B Computational Structures Technology 3- Part 2.

Dan Roberts. 2020. *Enginerding: What Is Anti-Squat & How Does It Actually Affect Mountain Bike Performance?* https://www.pinkbike.com/news/definitions-what-is-anti-squat.html

Ana B. Ruiz, Francisco Ruiz, Kaisa Miettinen, Laura Delgado-Antequera, and Vesa Ojalehto. 2019. NAUTILUS Navigator: free search interactive multiobjective optimization without trading-off. *Journal of Global Optimization* 74, 2 (01 Jun 2019), 213–231. https://doi.org/10.1007/s10898-019-00765-2

Adriana Schulz, Harrison Wang, Eitan Grinspun, Justin Solomon, and Wojciech Matusik. 2018. Interactive Exploration of Design Trade-offs. *ACM Trans. Graph.* 37, 4, Article 131 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201385

Adriana Schulz, Jie Xu, Bo Zhu, Changxi Zheng, Eitan Grinspun, and Wojciech Matusik. 2017. Interactive Design Space Exploration and Optimization for CAD Models. *ACM Transactions on Graphics* 36, 4 (jul 2017).

Christian Schumacher, Jonas Zehnder, and Moritz Bächer. 2018. Set-in-Stone: Worst-Case Optimization of Structures Weak in Tension. *ACM Trans. Graph.* 37, 6, Article 252 (Dec. 2018), 13 pages. https://doi.org/10.1145/3272127.3275085

Oliver Schütze, Oliver Cuate, Adanay Martín, Sebastian Peitz, and Michael Dellnitz. 2019. Pareto Explorer: a global/local exploration tool for many-objective optimization problems. *Engineering Optimization* 0, 0 (2019), 1–24.

Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. 2014. High-contrast Computational Caustic Design. *ACM Trans. Graph.* 33, 4, Article 74 (July 2014), 11 pages. https://doi.org/10.1145/2601097.2601200

Seven Cycles. 2020. *Seven Design Philosophy – Understanding Chainstays.* https://www.sevencycles.com/chainstay-philosophy.php

Maria Shugrina, Ariel Shamir, and Wojciech Matusik. 2015. Fab Forms: Customizable Objects for Fabrication with Validity and Geometry Caching. *ACM Trans. Graph.* 34, 4, Article Article 100 (July 2015), 12 pages. https://doi.org/10.1145/2766994

A. Spielberg, B. Araki, C. Sung, R. Tedrake, and D. Rus. 2017. Functional co-optimization of articulated robots. In *2017 IEEE International Conference on Robotics and Automation (ICRA).* 5035–5042.

Ed Spratt. 2019. *The Tech Behind the New Atherton Bikes.* https://www.pinkbike.com/news/the-tech-behind-the-new-atherton-bikes.html

Russ Tedrake. 2020. *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832).* Downloaded on May 19, 2020 from http://underactuated.mit.edu/.

Erva Ulu, James Mccann, and Levent Burak Kara. 2017. Lightweight Structure Design under Force Location Uncertainty. *ACM Trans. Graph.* 36, 4, Article 158 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073626

Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra. 2012. Guided Exploration of Physically Valid Shapes for Furniture Design. *ACM Trans. Graph.* 31, 4, Article 86 (July 2012), 11 pages. https://doi.org/10.1145/2185520.2185582

Nobuyuki Umetani, Yuki Koyama, Ryan Schmidt, and Takeo Igarashi. 2014. Pteromys: Interactive Design and Optimization of Free-formed Free-flight Model Airplanes. *ACM Trans. Graph.* 33, 4, Article 65 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601129

David A. Van Veldhuizen and Gary B. Lamont. 2000. Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evol. Comput.* 8, 2 (June 2000), 125–147. https://doi.org/10.1162/106365600568158

Lingfeng Wang and Emily Whiting. 2016. Buoyancy Optimization for Computational Fabrication. *Computer Graphics Forum (Proceedings of Eurographics)* 35, 2 (2016).

Shawn Wasserman. 2017. Topology Optimization Comes to SOLIDWORKS. https://www.engineersrule.com/topology-optimization-comes-solidworks/

Jiaxian Yao, Danny M. Kaufman, Yotam Gingold, and Maneesh Agrawala. 2017. Interactive Design and Stability Analysis of Decorative Joinery for Furniture. *ACM Trans. Graph.* 36, 2, Article 20 (March 2017), 16 pages. https://doi.org/10.1145/3054740

J. Zhang and L. Xing. 2017. A Survey of Multiobjective Evolutionary Algorithms. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC),* Vol. 1. 93–100.

Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2020. RoboGrammar: Graph Grammar for Terrain-Optimized Robot Design. *ACM Trans. Graph.* 39, 6, Article 188 (Nov. 2020), 16 pages. https://doi.org/10.1145/3414685.3417831

Qingnan Zhou, Julian Panetta, and Denis Zorin. 2013. Worst-Case Structural Analysis. *ACM Trans. Graph.* 32, 4, Article 137 (July 2013), 12 pages. https://doi.org/10.1145/2461912.2461967

Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization with Microstructures. *ACM Transactions on Graphics* 36, 5, Article 164 (July 2017), 16 pages.

Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.* 8, 2 (June 2000), 173–195. https://doi.org/10.1162/106365600568202

A PROOF OF FIRST-ORDER APPROXIMATION

Suppose we construct a curve $\gamma(t) := (\mathbf{x}(t), \mathbf{z}(t))$, constrained to the Pareto gamut for $t \in (-\varepsilon, \varepsilon)$ as in Proposition 4.7. At a given point $(\mathbf{x}^*, \mathbf{z}^*)$ along the curve, the number of active constraints g_k is $K' \leq K$. Without loss of generality, we can permute the constraints g_k such that the first $K' \leq K$ are the active ones. Because our objectives are continuous, all (in)active constraints remain (in)active within our (potentially restricted) ε-ball of interest. Thus, inactive constraints g_k have no impact on Eqn. 7 (KKT stationarity), because Eqn. 6 (KKT complementary slackness) ensures that $\beta_k = 0$ for all relevant ε. We reformulate our KKT conditions from Proposition 3.5, in order to (1) explicitly ignore the inactive constraints (which have no effect), (2) reflect our parametrization on t, and (3) require that the context $\mathbf{z}$ remains within its predefined feasible range:

$$(\mathbf{x}^*, \mathbf{z}^*) \in \mathcal{X} \tag{10}$$

$$\alpha_i(t) \geq 0 \quad \forall i \in \{1, \ldots, d\}, t \in (-\epsilon, \epsilon) \tag{11}$$

$$\beta_k(t) \geq 0 \quad \forall k \in \{1, \ldots, K'\}, t \in (-\epsilon, \epsilon) \tag{12}$$

$$\sum_{i=1}^{d} \alpha_i(t) = 1 \quad \forall t \in (-\epsilon, \epsilon) \tag{13}$$

$$\beta_k(t) g_k(\mathbf{x}(t), \mathbf{z}(t)) = 0 \quad \forall k \in \{1, \ldots, K'\}, t \in (-\epsilon, \epsilon) \tag{14}$$

$$\left[\sum_{i=1}^{d} \alpha_i(t) \nabla_{\mathbf{x}} f_i(\mathbf{x}(t), \mathbf{z}(t))\right] + \left[\sum_{k=1}^{K'} \beta_k(t) \nabla_{\mathbf{x}} g_k(\mathbf{x}(t), \mathbf{z}(t))\right] = 0 \quad \forall t \in (-\epsilon, \epsilon) \tag{15}$$

Denote with $h(t)$ the last line of the modified KKT conditions (Eqn. 15). We recall that at $t = 0$, we have $(\mathbf{x}(0), \mathbf{z}(0)) = (\mathbf{x}^*, \mathbf{z}^*)$, which is Pareto optimal by construction. Thus, there generically exist KKT dual variables $\boldsymbol{\alpha}(0) = \boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}(0) = \boldsymbol{\beta}^*$ such that $h(0) = 0$. We wish to find a first order approximation about $t = 0$ that preserves this condition locally, i.e. $h(t) = 0$ for $t \in (-\epsilon, \epsilon)$. Thus, we want to enforce $h'(t) = 0$ for $t \in (-\epsilon, \epsilon)$. Differentiating $h(t)$ and evaluating at $t = 0$ yields the following relation:

$$\begin{aligned}
0 \equiv h'(0) &= D_{\mathbf{x}} F^T(\mathbf{x}^*, \mathbf{z}^*) \boldsymbol{\alpha}'(0) + D_{\mathbf{x}} G^T(\mathbf{x}^*, \mathbf{z}^*) \boldsymbol{\beta}'(0) \\
&\quad + \left[\sum_{i=1}^{d} \alpha_i^* H_{xx} f_i(\mathbf{x}^*, \mathbf{z}^*) + \sum_{k=1}^{K'} \beta_k^* H_{xx} g_k(\mathbf{x}^*, \mathbf{z}^*)\right] \mathbf{x}'(0) \\
&\quad + \left[\sum_{i=1}^{d} \alpha_i^* H_{xz} f_i(\mathbf{x}^*, \mathbf{z}^*) + \sum_{k=1}^{K'} \beta_k^* H_{xz} g_k(\mathbf{x}^*, \mathbf{z}^*)\right] \mathbf{z}'(0) \\
&=: D_x F^T \cdot \boldsymbol{\alpha}'(0) + D_x G^T \cdot \boldsymbol{\beta}'(0) + Hx \cdot \mathbf{x}'(0) + Hz \cdot \mathbf{z}'(0) \,.
\end{aligned}$$

We also consider the additional constraints given by Eqn. 13 and Eqn. 14. The former simply requires that $\left(\sum \alpha_i = 1\right) \implies \left(\sum \alpha_i' = 0\right)$. For the latter, we differentiate with respect to t, and simplify:

$$\begin{aligned}
0 &\equiv \beta_k'(t) g_k(\mathbf{x}(t), \mathbf{z}(t)) \\
&\quad + \beta_k(t) \left[\nabla_{\mathbf{x}} g_k(\mathbf{x}(t), \mathbf{z}(t))^T \cdot \mathbf{x}'(t) + \nabla_{\mathbf{z}} g_k(\mathbf{x}(t), \mathbf{z}(t))^T \cdot \mathbf{z}'(t)\right] \qquad (16) \\
&\equiv \nabla_{\mathbf{x}} g_k(\mathbf{x}(t), \mathbf{z}(t))^T \cdot \mathbf{x}'(t) + \nabla_{\mathbf{z}} g_k(\mathbf{x}(t), \mathbf{z}(t))^T \cdot \mathbf{z}'(t) \qquad (17)
\end{aligned}$$

This simplification (from Eqn. 16 to Eqn. 17) follows because all constraints g_k with $k \in \{1, \ldots, K'\}$ are active for all $t \in (-\epsilon, \epsilon)$. Thus, $g_k(\mathbf{x}(t), \mathbf{z}(t)) = 0$ (causing the first term to vanish), and $\beta_k(t) \neq 0$. Stacking expression Eqn. 17 for all K' constraints g_k into a single matrix, and evaluating at 0, we have

$$\mathbf{0} = [D_{\mathbf{x}} G, D_{\mathbf{z}} G] \cdot \begin{bmatrix} \mathbf{x}'(0) \\ \mathbf{z}'(0) \end{bmatrix}$$

Combining all of the constraints above, it must be the case that

$$\begin{aligned}
\mathbf{0} &\equiv \underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & D_{\mathbf{x}} G & D_{\mathbf{z}} G \\ D_x F^T & D_x G^T & Hx & Hz \end{bmatrix}}_{\in \mathbb{R}^{(1+K'+D) \times (d+K'+D+C)}} \begin{bmatrix} \boldsymbol{\alpha}'(0) \\ \boldsymbol{\beta}'(0) \\ \mathbf{x}'(0) \\ \mathbf{z}'(0) \end{bmatrix} \qquad (18) \\
&=: Mv \qquad (19)
\end{aligned}$$

□