**Proceedings of the ASME 2022
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conferences
IDETC/CIE2022
August 14-17, 2022, St. Louis, MO, USA**

DETC2022-91344

DESIGN TARGET ACHIEVEMENT INDEX: A DIFFERENTIABLE METRIC TO ENHANCE DEEP GENERATIVE MODELS IN MULTI-OBJECTIVE INVERSE DESIGN

Lyle Regenwetter
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: regenwet@mit.edu

Faez Ahmed
Dept. of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: faez@mit.edu

ABSTRACT

Deep Generative Machine Learning Models have been growing in popularity across the design community thanks to their ability to learn and mimic complex data distributions. While early works are promising, further advancement will depend on addressing several critical considerations such as design quality, feasibility, novelty, and targeted inverse design. We propose the Design Target Achievement Index (DTAI), a differentiable, tunable metric that scores a design's ability to achieve designer-specified minimum performance targets. We demonstrate that DTAI can drastically improve the performance of generated designs when directly used as a training loss in Deep Generative Models. We apply the DTAI loss to a Performance-Augmented Diverse GAN (PaDGAN) and demonstrate superior generative performance compared to a set of baseline Deep Generative Models including a Multi-Objective PaDGAN and specialized tabular generation algorithms like the Conditional Tabular GAN (CTGAN). We further enhance PaDGAN with an auxiliary feasibility classifier to encourage feasible designs. To evaluate methods, we propose a comprehensive set of evaluation metrics for generative methods that focus on feasibility, diversity, and satisfaction of design performance targets. Methods are tested on a challenging benchmarking problem: the FRAMED bicycle frame design dataset featuring mixed-datatype parametric data, heavily skewed and multimodal distributions, and ten competing performance objectives.

1 INTRODUCTION

Automatically creating innovative designs that outperform all existing solutions and meet complex real-world engineering constraints is the holy grail of data-driven engineering design. This is an incredibly demanding task and current design automation tools remain insufficient for full autonomy in product design. Recently, Deep Generative Models have emerged as a viable means to bring us toward this overarching design automation goal. Deep Generative Models (DGMs) refer to machine learning algorithms that leverage sequential layers to learn progressively deeper understandings of design representations. Typically, these models are trained to understand the distribution of existing designs in some design space (usually using a dataset of existing designs), then generate new designs by sampling from this learned distribution.

DGMs are typically trained to maximize the statistical similarity between distributions of generated samples and the underlying data distribution. In engineering design, design objectives and constraints make statistical similarity metrics insufficient and sometimes inappropriate. Despite this, an overwhelming majority of research in engineering design continues to optimize and evaluate methods using statistical similarity. We believe the continuation of this practice is rooted in two central challenges. Firstly, appropriate metrics to evaluate DGMs on engineering objectives such as design performance, feasibility, and novelty are poorly established. Secondly, researchers lack effective methods to build these auxiliary objectives into training procedures and instead fall back upon the established structural similarity as the central training mechanism. Addressing these two challenges is the central thrust of this paper.

Contributions: Our key contributions are summarized as follows:

1. We propose a multifaceted set of evaluation metrics for Deep Generative Models in engineering design consisting of our novel DTAI metric, as well as seven other metrics focusing on design and performance space diversity, novelty, feasibility, and target satisfaction.
2. We introduce the Design Target Achievement Index (DTAI), a differentiable scoring metric which allows Deep Generative Models to prioritize, meet, and exceed multi-objective performance targets specified by a designer.
3. We augment a state-of-the-art Performance-Augmented Diverse GAN with a loss based on our DTAI function and feasibility estimator and demonstrate that this framework yields significant performance improvements, such as increasing the average proportion of design targets met by 45% and the proportion of feasible designs by 30% versus state-of-the-art tabular generation methods. To our knowledge, this proposed framework marks the first Deep Generative Method that actively optimizes for overall design performance, diversity, feasibility, and target satisfaction simultaneously.
4. We evaluate several existing Deep Generative Models using our proposed evaluation metrics on a challenging real-world design dataset with ten competing objectives and complicated regions of infeasibility. We demonstrate that our DTAI-augmented method significantly outperforms baseline DGMs in numerous performance metrics.

In the following sections, we discuss the dataset used, the methods tested, the evaluation metrics proposed, and the results of our analysis.

2 REVIEW OF DEEP GENERATIVE MODELS IN ENGINEERING DESIGN

In a recent review of Deep Generative Models (DGMs), Regenwetter *et al.* [1] discuss the application of DGMs across engineering design fields and analyze key limitations in the current state-of-the-art in DGM methodology. The authors suggest that successfully addressing several key challenges will be essential in the continued development of DGMs for engineering design. Four of these challenges are design quality, design novelty, more robust design representation methods, and targeted inverse design. In this section, we briefly summarize the state of the current research, as well as key drivers behind each of these four challenges. For a more detailed review and discussion, we refer the reader to [1].

Design Quality: Design Quality (Performance) is an essential component of the design process. Design quality may be comprised of many diverse (and sometimes competing) objectives. For example, the key measures of a bicycle's quality may include weight, cost efficiency, structural integrity, aesthetics, aerodynamics, and ergonomics. Designs must be high-performing to be competitive, economically viable, and positively impact customers. As Regenwetter *et al.* [1] note, most existing Deep Generative Models fail to account for design performance of any kind, instead solely training to minimize statistical divergence between the distribution of generated samples and the original data distribution. This trend can potentially be attributed to the fact that most DGMs are adapted from disciplines such as computer vision where "realism," of generated samples is the ultimate goal. In general, approaches to incorporate design quality into DGMs fall into three broad categories:

1. Building performance-estimating objectives into the training function of the DGM [2–5] allows the DGM to directly optimize for performance during training. For example, Ahmed & Chen [2,3], train a surrogate model which estimates aerodynamic lift and drag and build this surrogate into the overall loss calculation to generate airfoil designs with high lift/drag ratios. Chen & Ahmed's approach simultaneously builds a diversity score of the generated samples into the training procedure, which can improve design novelty, an objective that, like quality, is similarly overlooked when training solely to minimize statistical divergence.
2. Iteratively training the DGM on datasets augmented with high-perfoming generated designs [6–10] can bias generative models toward higher-performing regions of the design space. For example, Shu *et al.* [8] generate 3D models of aircraft, computationally evaluate their aerodynamic properties, then add high-performing models to their dataset before retraining their model.
3. Fitting surrogate models to link a learned latent design representation with performance parameters [11–15] can allow direct gradient-based optimization of latent vectors to decode into designs. For example, Li *et al.* [14] use a variational autoencoder to learn a latent embedding of phononic crystal designs, then optimize latent variables using a Deep Neural Network by mapping latent vectors to target band gap values.

Our proposed Design Target Achievement Index (DTAI) builds on the first approach by directly aggregating performance estimates into the novel DTAI training loss.

Design Representation: Designs can be digitized using countless representation schemes, and as Regenwetter *et al.* [1] note, many commonly used parameterizations severely limit the usability of generated designs in downstream tasks. Representations like images and voxelizations are easy to train on due to their spatially-structured properties, but generated designs are difficult to evaluate using computational analysis tools like Finite Element Analysis, and even more challenging to physically fabricate. DGMs trained on design representations using interpretable physical dimensions of products are much more viable in downstream tasks, but are significantly more challenging to train, due to the lack of structure in their representation, mixed datatypes, and heavily unpredictable distributions [16]. These challenges of training DGMs on tabular parametric data are noted across domains [17].

Inverse Design: Most existing DGMs in engineering design lack any mechanism to condition design generation toward a specific set of designer-specified performance targets. The recently proposed Performance-conditioned Diverse GAN (PcDGAN) [18] generates designs whose performance *exactly* matches a *single* designer-specified target performance. In this work, however, we consider design problems in which the algorithm attempts to *exceed* a set of *multiple* minimum performance targets, a task which Deep Generative Models have, to our knowledge, not yet addressed. However, many approaches from the multi-objective optimization field consider such minimum performance targets, albeit with some limitations. A common approach sees targets handled as hard constraints which take precedence over any sort of design optimality. This approach is seen in well-known optimization algorithms like NSGA-II [19], which prioritize the resolution of constraint violation before moving on to optimization. While this rigid treatment of design targets can be helpful, hard constraints lack the nuance afforded by softer design objectives, a challenge which we address with our proposed Design Target Achievement Index.

3 EVALUATING GENERATIVE MODELS

Establishing metrics to evaluate DGMs on training objectives beyond structural similarity is essential. In engineering design, we are often given performance targets during design tasks that constitute the minimum performance necessary to meet design goals. In practice, designers implicitly adapt their design process based on these minimum performance targets in ways that are difficult to quantify:

1. If a design is underperforming the performance target in a particular design objective, design iterations should focus on improving performance in this objective
2. If a design is drastically outperforming the performance target in a particular design objective, design iterations should not prioritize the further improvement of this objective
3. Design metrics are typically weighted adaptively to match the relative importance of different design targets.

While we can semantically describe these phenomena, the existing tools used in multi-objective design optimization like hypervolume fail to capture their nuance. Instead, we propose the Design Target Achievement Index, a fast, differentiable scoring method of design performance that addresses each of these concerns by adaptively weighting objectives and specifically rewarding designs that satisfy design targets (Sec. 3.1). We then discuss several other metrics measuring various other aspects of design generation, such as diversity and feasibility.

3.1 Design Target Achievement Index (DTAI)

We propose a novel approach to quantify a design's performance with respect to multiple performance targets in a single metric. Consider a design, i, and let its performance be $p_{i,k}$ with respect to a particular performance target, t_k, for $k \in \{1...T\}$ where T is the number of design objectives (i.e. weight, safety factor, *etc.*). In any given objective, k, we desire our design's performance to exceed the performance target: $p_{i,k} \geq t_k$. In this formulation, we require that design performance and targets be strictly positive and the objectives maximized, however, most design objectives can be trivially reformulated as such. We express the design's performance with respect to each target as the ratio, $r_{i,k}$, between performance and the performance target:

$$r_{i,k} = \frac{p_{i,k}}{t_k} \tag{1}$$

When our design's performance exceeds the target, $r_{i,k} \geq 1$. We propose the following piecewise scoring function to compute an individual target achievement score, $s_{i,k}$ in terms of $r_{i,k}$.

$$s_{i,k} = \begin{cases} \alpha_k(1-r_{i,k}) & r_{i,k} \leq 1 \\ \frac{\alpha_k}{\beta_k}(1-e^{\beta_k(1-r_{i,k})}) & r_{i,k} > 1 \end{cases} \tag{2}$$

This function is parameterized by two tuning factors, α_k and β_k, which are best visualized graphically. α_k adjusts the importance of target k and can be thought of as the key weighting factor used to tune the relative importance between performance targets. β_k is slightly more subtle and reflects the relative importance of further improvement to the objective after the performance target is met. Continuing to optimize some objectives may be more helpful than others and this parameter gives designers the ability to adjust for this nuance. For most design problems, we recommend β_k values around $3-5$. Figures 1 and 2 show the effect of adjusting α_k and β_k.

The individual scores $s_{i,k}$ for each objective are easily summed into an aggregate score, then scaled by the theoretical minimum and maximum values for the sum, s_{min} and s_{max}. This final result is our proposed Design Target Achievement Index, s_{DTAI}.

$$s_{DTAI,i} = \frac{(\sum_{k=1}^{n} s_{i,k}) - s_{min}}{s_{max} - s_{min}} \tag{3}$$

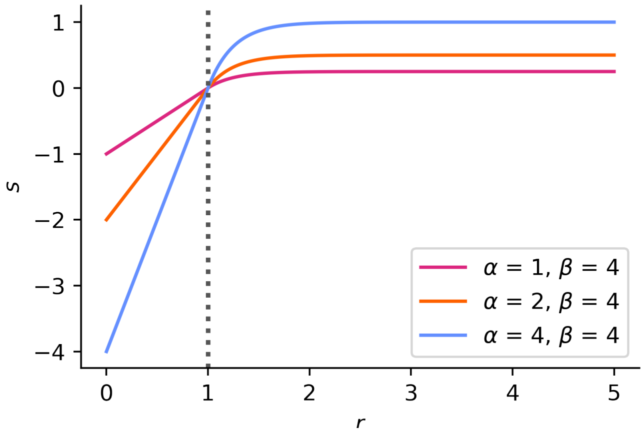

FIGURE 1: Effect of α parameter on Design Target Achievement Index.

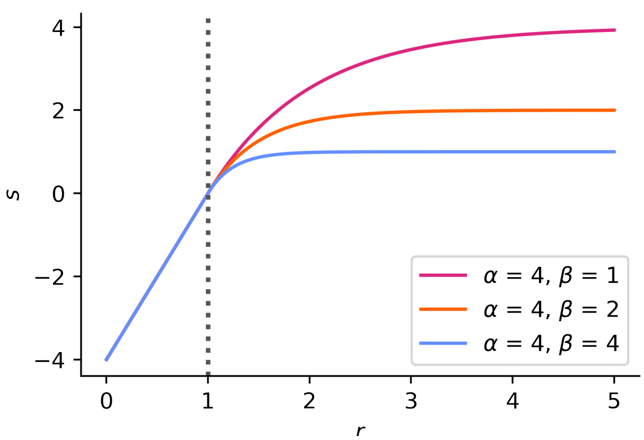

FIGURE 2: Effect of β parameter on Design Target Achievement Index.

s_{min} and s_{max} are easily derived as:

$$s_{min} = -\sum_{k=1}^{T} \alpha_k, \; s_{max} = \sum_{k=1}^{T} \frac{\alpha_k}{\beta_k} \tag{4}$$

The proposed scoring function has numerous benefits that make it desirable both in an evaluation setting and as an objective function for the training of Deep Generative Models:

1. DTAI has a large derivative with respect to an individual objective when it is currently underperforming the corresponding performance target. Conversely, the objective has an exponentially decaying derivative with respect to an individual objective as it further outperforms the objective's performance target.
2. DTAI is differentiable across the entire space of possible performance and constraint values and its derivative is continuous. This allows it to be used directly in the optimization functions of gradient-based generative methods.
3. DTAI is easy to calculate, with computational cost scaling linearly with the number of design objectives.
4. DTAI is bounded between zero and one. The gradient of DTAI is also bounded given a particular set of α and β parameters.
5. The scoring function allows for easy weighting of objectives and modulation of score decay, which enables easy and precise customization by the designer.

We note that this scoring metric is intended for use in targeted inverse design applications, where generating high-performing designs to meet a specific set of performance targets is the overarching goal. This score is poorly suited for quantification of performance space coverage or unconditional design synthesis.

3.2 Hypervolume (HV)

Hypervolume is a useful metric for simultaneously quantifying performance space coverage and overall design optimality of a collection of design candidates. When calculating the hypervolume metric of a set of designs, we consider a T-dimensional space where T is the number of design objectives. The hypervolume is given by the volume of the union of all points within some hypercube spanning one of the designs and a common reference point.

Hypervolume is a frequently used metric in multi-objective optimization and typically aims to quantify a solution set's proximity to the Pareto front [20, 21]. In the context of targeted inverse design (i.e. designing a product for a specific collection of design targets) hypervolume suffers from several limitations. In particular, hypervolume tends to 1) Over-reward further optimization of objectives that have already exceeded performance targets 2) Under-reward focused performance improvements to meet performance targets 3) Ignore the relative importance of different targets. 4) Require non-negligible computational expense. Though implanting design targets as the reference point for hypervolume calculation is a method to ensure that only designs that exceed all performance targets are scored, this method remains rather inflexible since designs that nearly miss performance targets are treated the same as designs that drastically miss them. For this reason, we select a reference point far below the performance targets of each objective, which we discuss further in Sec. **??**.

3.3 Design Diversity (DSD & PSD)

In generative design problems, we may seek to generate a diverse set of design candidates to give a designer a variety of design possibilities. Given a set of generated designs, we can score the diversity of each design by calculating its similarity to the other designs in the set and averaging these values. Mathematically,

$$s_{div,i} = \frac{1}{n-1} \sum_{j \in P} (\phi(y_i, y_j)) \tag{5}$$

Here, $s_{div,i}$ is the diversity score, y is the set of generated designs, and y_i refers to the i^{th} design in the generated set. n is the number of sampling iterations, P is a randomly selected set of n designs from y, and ϕ is the kernel function calculating similarity, in this case, Euclidean Distance. Design diversity can be calculated in the *design space* by calculating the similarity between parametric design vectors, which we call Design Space Diversity (DSD). Design diversity can also be quantified in the *performance space* by calculating the similarity between vectors of design performance values, which we call Performance Space Diversity (PSD).

3.4 Design Novelty (DN)

Quantifying a design's novelty can be important in real-world design tasks where intellectual property is an important concern. We adopt an approach based on [2]. Given a set of designs, we can score the novelty of each design by calculating its similarity to the designs in the original dataset and finding the minimum value (similarity to the most similar original design). Mathematically,

$$s_{DN,i} = \min_j (\phi(y_i, x_j)) \tag{6}$$

Here, $s_{DN,i}$ is the novelty score, y_i refers to the i^{th} design in the generated set, and x_j refers to the j^{th} design in the original dataset. Again, ϕ is the kernel function calculating similarity, for which we use Euclidean Distance.

3.5 Geometric Feasibility Rate (GFR)

Generating physically feasible designs is an important consideration when evaluating generative methods. The Geometric Feasibility Rate is simply the ratio of total designs found to be feasible to the number of designs where feasibility status is known. By leveraging the simulation pipeline of the FRAMED dataset, we have a convenient way to explicitly quantify the Geometric Feasibility Rate of generated designs. Simply stated, feasible designs satisfy a set of predefined feasibility rules provided by FRAMED's authors and furthermore build into a valid 3D model provided.

3.6 Target Success Rate (TSR)

Evaluating a generative method's ability to create designs that satisfy performance targets is critical. While the Design Target Achievement Index (DTAI) proposed is largely affected by a generated design's ability to satisfy performance targets, quantifying the raw fraction of performance targets satisfied is also a helpful reference metric. This ratio is expressed as the fraction of the design targets met or exceeded by any given design, weighted by the importance (as specified by the designer) of the targets.

$$s_{TSR,i} = \sum_{k}^{T} \alpha_k q_{i,k} \tag{7}$$

$$q_{i,k} = \begin{cases} 1 & p_{i,k} \geq t_k \\ 0 & p_{i,k} < t_k \end{cases} \tag{8}$$

Here, T is the number of performance objectives. α_k is the importance of objective k and is the same parameter as the hyperparameter α in DTAI.

3.7 Minimum Target Ratio (MTR)

We may also want to evaluate the degree to which generated designs are meeting or failing performance targets. For any generated design, y_i and objective k consider the ratio $r_{i,k}$ between a designs performance $p_{i,k}$ and the performance target t_k. $s_{MTR,i}$ is defined as the minimum such ratio.

$$s_{MTR,i} = \min_k \left(\frac{p_{i,k}}{t_k} \right) \tag{9}$$

When the MTR is greater than one, it tells us by at least how much the performance in each objective outperforms the performance target. When the MTR is less than one, it tells us how far the design is underperforming the target in its most delinquent objective. Unlike the TSR, the MTR is not weighted by target importance.

4 FRAMED DATASET

For this study, we select the recently-introduced FRAMED dataset [22], which at the time of this work (version 1.0), consists of 4500 community-designed bicycle frame models parameterized over 37 design variables: tube lengths, diameters, and thicknesses, frame material, and frame junction locations. Three sample frame models from the dataset are rendered and shown in Figure 3. The FRAMED dataset provides model weight, and a set of 9 structural performance measures for a total of 10 objectives. These performance measures are derived from simulation results calculated in Finite Element Analysis (FEA) under three loading cases:

1. In-plane loading: Bottom bracket vertical and horizontal displacement, Dropout vertical and horizontal displacement, safety factor
2. Transverse loading: Bottom bracket lateral displacement
3. Eccentric loading: Bottom bracket vertical displacement and rotation, safety factor

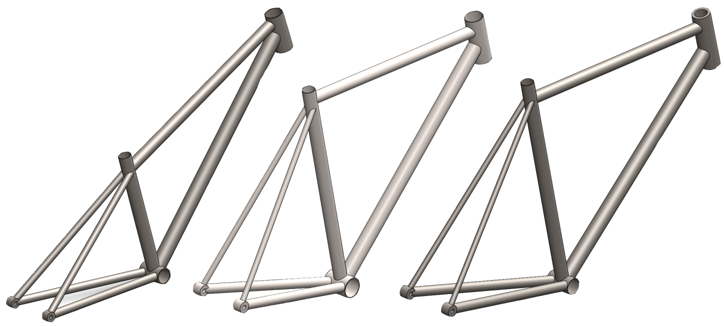

FIGURE 3: Examples of geometrically feasible bicycle frame models from the FRAMED dataset.

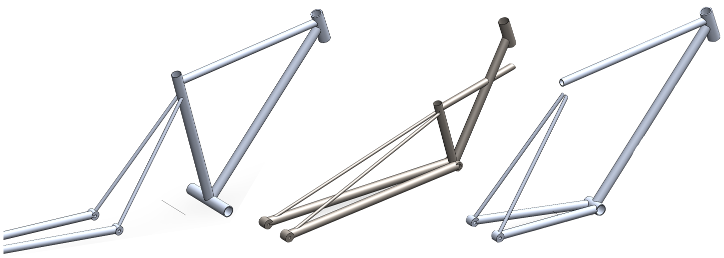

FIGURE 4: Examples of geometrically infeasible bicycle frame models from the FRAMED dataset.

These performance metrics constitute a challenging mixed optimization problem with competing objectives. This bicycle frame optimization task is an active research field with significant commercial investment.

Compared to other datasets frequently used to test and benchmark generative methods, such as the UIUC airfoil database, FRAMED's ten objectives make for a significantly more complex multi-objective optimization problem. Additionally, FRAMED considers the problem of design feasibility by sorting frames into feasible and infeasible designs. Frames are determined to be infeasible through a systematic check of geometric flaws. These flaws include negative tube lengths and thicknesses, triangles with one side longer than the sum of the counterparts, or dimensions that would cause parts not to connect. Some frames tha pass the explicit checks fail to generate into a proper 3D model when built in the FEA simulation software, the reasons for which are highly unpredictable and nearly impossible to exhaustively list. Three frames that build incorrectly in the FEA simulation software are shown in Figure 4. While most of these frames are invalid, their status is uncertain, so they are discarded. FRAMED also presents a challenge for DGM training in that it features mixed datatypes (categorical and continuous), and features data with multimodal and skewed distributions, as seen in Figure 5. Unlike most DGM models used for image data, DGM models for tabular data are less well understood and are generally considered more challenging to train.

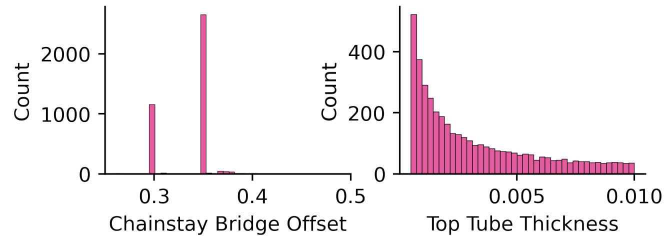

FIGURE 5: Examples of multimodal and skewed parameter distributions in FRAMED dataset.

FRAMED also shares the simulation methodology, code, and 3D CAD model files, which gives us the capability to simulate models we generate while testing the various Deep Generative Models discussed in this work. Critically, this allows us to evaluate and benchmark these generative methods with a variety of evaluation metrics, most of which are based on performance values calculated using FRAMED's simulation methodology.

5 GENERATIVE MODELS

In this section, we present the Deep Generative Models evaluated in this paper, discussing the core functionality and key innovations of each method. We additionally present two trivial baseline cases which we test alongside the Deep Generative Models.

5.1 Baselines

To establish baselines, we consider two trivial "generative methods" against which to benchmark more advanced frameworks. The first of these is a simple random sampling from the dataset. The second of these is a linear interpolation between dataset designs. Mathematically, we generate an interpolant, d_{int} by randomly selecting two designs from the dataset, x_1 and x_2 as well as in interpolation factor, $\gamma \in [0,1]$. Our interpolant is given by:

$$x_{int} = \gamma x_1 + (1-\gamma)x_2 \tag{10}$$

While both of these random sampling methods may be practical methods to select designs, we expect the Deep Generative Models tested to outperform these baselines in most metrics.

5.2 Generative Adversarial Network

Introduced in 2014, the Generative Adversarial Network (GAN) [23] has become a staple across numerous fields, largely due to its unprecedented performance in generating convincing human faces [24]. The GAN consists of a generator network and a discriminator network that play an adversarial game in which the discriminator attempts to distinguish real samples from the dataset from generated samples from the generator, while the generator attempts to fool the discriminator. Though many variations of the GAN have been proposed, we use a standard GAN as a baseline.

5.3 Tabular Generation Algorithms

The task of generating tabular parametric data frequently occurs in non-design-related applications. As such, many algorithms have been proposed to address the tabular generation problem, though, to our knowledge, all have focused on statistical similarity as the key training objective. We test two well-established methods, the Tabular Variational Autoencoder (TVAE) and the Conditional Tabular Generative Adversarial Network (CTGAN) [17] introduced by Xu *et al.* in 2019. CTGAN and TVAE established state-of-the-art performance on tabular data generation for their respective classes of generative methods (GAN and VAE). These methods serve primarily as a benchmark for high-performance methods that do not explicitly address performance-aware design generation.

While we refer the reader to Xu *et al.*'s paper [17] for details, we summarize the key innovations below. CTGAN and TVAE primarily address key challenges of tabular data, such as multimodal or skewed distributions, mixed datatypes (continuous and categorical), and skewed categories, all of which are common in the FRAMED data. CTGAN and TVAE both implement advancements to better handle both continuous and categorical datatypes in tabular data.

To better learn continuous data, the authors implement a method that they call "mode-specific normalization." This approach assumes that continuous variables fall into a Gaussian Mixture distribution and learns a Variational Gaussian Mixture Model. Sampled parameters are then probabilistically assigned to a particular mode and represented in terms of this assigned mode and the corresponding p-value within that mode's Gaussian distribution. This allows CTGAN and TVAE to more reliably learn complex distributions over continuous parameters.

To better learn categorical data, CTGAN trains conditionally using possible values of every categorical parameter in the data as the training condition. This prevents CTGAN from ignoring particular data categories, avoiding mode collapse. TVAE improves performance on categorical data simply by employing mixed activation functions in the final layer of its decoder. In particular, categorical variables are generated using a softmax activation. The combination of advancements for categorical and continuous parameters in tabular data makes CTGAN and TVAE particularly effective in mixed-datatype tabular generation problems.

In our testing, CTGAN and TVAE are trained on the original dataset designs without any one-hot encoding and do not utilize the performance data. We train for a maximum of 2000 epochs.

5.4 Performance-Augmented Diverse GAN (PaDGAN)

Introduced in 2021 by Chen & Ahmed, the Performance-Augmented Diverse GAN (PaDGAN) [2] specializes in performance- and diversity-aware design generation. PaDGAN demonstrated convincing synthesis performance on a variety of synthetic datasets as well as an airfoil design problem, generating a diverse set of samples that significantly exceeds the original dataset in average performance.

We refer the reader to the original paper by Chen & Ahmed for implementation details but summarize the key innovations below. PaDGAN implements an auxiliary training loss based on a Determinantal Point Process (DPP), which calculates a matrix over a batch of designs based on the similarity of designs in the batch and the quality (performance) values of the designs. The DPP loss is then calculated from this DPP matrix using a scaled log determinant. PaDGAN relies either upon a deterministic quality function or a differentiable approximation for the quality known as a surrogate model.

An essential component of PaDGAN is a rapid performance evaluation method, which can be queried during training to evaluate generated design candidates. Our performance objectives have no deterministic relations based on design parameters and simulating designs in the training loop is too costly, so we fit a surrogate model to approximate the ten adjusted performance values and provide this model to PaDGAN to query during training. We select a deep neural network with four hidden layers of 200 neurons and batch-normalization and rectified linear unit activation functions after every hidden layer. This surrogate achieves a coefficient of determination of 0.681 on the entire dataset, implying that the regression fit is moderately strong. While FRAMED's authors present a higher-performing surrogate based on ensembles of individual regressors, including non-differentiable tree-based regressors, PaDGAN requires that the loss calculation be fully differentiable. Identifying higher-performing differentiable surrogates is an area for future work.

PaDGAN inherently optimizes for a single objective, but Chen & Ahmed extend PaDGAN to Multi-Objective PaDGAN (MO-PaDGAN) with a method for combining multiple objectives into a single aggregate score. MO-PaDGAN proposes computing a randomly averaged weight of the different performance values for each sample propagated through the scoring function. The authors' primary motivation for this approach is the exploration of different regions of the design space's "Pareto-front." The Pareto-front is a boundary of the design space consisting of all potentially "optimal" designs where any improvement in one objective must come at the expense of another. Since PaDGAN and MO-PaDGAN require that objectives be maximized, we try taking a simple inverse to convert deflection and weight objectives into maximization problems. We find that MO-PaDGAN trains unstably since it requires that performance scores within a batch be of similar magnitudes. This instability traces back to unbounded individual objective scores since objectives can be arbitrarily close to zero before inversion. To avoid this instability we propose an alternate approach that bounds resultant scores:

$$s_{i,k} = \frac{p_{k,med}}{p_{k,med} + p_{i,k}} \tag{11}$$

Here, $p_{k,med}$ is the median performance in objective k of all dataset designs, while $p_{i,k}$ is the performance of a particular design, i to be scored. Scores can then be weighted using the below function, where w_k are random weights in a given range, say, $[0,1]$.

$$s_{MO,i} = \frac{\sum_k^T s_{i,k} * w_k}{\sum_k^T w_k} \tag{12}$$

5.5 DTAI and Classfier-augmented PaDGAN

While the proposed approach allows us to use MO-PaDGAN on FRAMED, we propose to replace s_{MO} scores with Design Target Achievement Index (DTAI) scores instead. DTAI is better suited for targeted inverse design, spending less time exploring regions of the design space and focusing on direct optimization given a specified set of minimum performance targets.

We further augment PaDGAN with an auxiliary classifier trained to classify feasible bicycle frames. We scale our DTAI score by this classifier's predicted likelihood of the given frame being geometrically valid and use this scaled score in PaDGAN's DPP loss.

$$s_{Tot,i} = s_{DTAI,i} * Q(y_i) \tag{13}$$

Here, Q is a classifier that predicts the likelihood of a design, y_i to be valid. Like the regressor, we are limited to differentiable surrogates and select a neural network that achieves a classification F1 score of 0.71. PaDGAN implements two tuning parameters, γ_0, which adjusts the weight of performance score in the loss function (compared to diversity), and γ_1, which adjusts the weight of the combined performance and diversity loss in the overall training respectively. We set $\gamma_0 = 5$, $\gamma_1 = 0.5$, and train for 50,000 iterations.

6 METHODOLOGY

To test each of the methods, we first train the framework on only the valid designs in the full FRAMED dataset. We then generate 250 designs using the trained model, filter out initially infeasible designs, then simulate all remaining designs using the FEA simulation framework proposed in FRAMED. We then evaluate all feasible designs on the evaluation metrics proposed (save for Geometric Feasibility Rate, which is evaluated as the ratio of feasible to infeasible designs). We repeat this process three times for every method and report median scores over the three instantiations.

6.1 Data Preprocessing

FRAMED's raw deflection values are given as absolute deflections, but we simplify to only consider deflection magnitudes. Since we assume the maximization of a set of positive objectives, we invert deflection and weight objectives (excepting PaDGAN when DTAI loss is not used – See Section 5.4). We hereby refer to these modified performance values as the 'adjusted performance values.' We discuss some of the training intricacies below.

6.2 Evaluation Metrics

Hypervolume calculations require a reference point. We select a reference point for hypervolume calculations such that each dimension's coordinate is equal to the FRAMED data's 1^{st} percentile objective score (worse than 99% of FRAMED designs in each metric). Design Target Achievement Index, Target Success Rate, and Minimum Target Ratio all require a set of minimum performance targets. We select these targets to be equal to the 75^{th} percentile objective score (better than 75% of FRAMED designs in each metric). Note that many objectives are difficult to simultaneously optimize since they inherently compete, so there is no guarantee that these minimum performance targets are even possible to simultaneously satisfy. This is a challenging objective, but realistic from an inverse design standpoint, as a designer may often set difficult or even impossible targets and expect a design that comes as close as possible.

7 RESULTS

The overall results of our testing are shown in Table 1. We find that CTGAN and TVAE yield similar results, so we only present CTGAN results for simplicity. Table 2 presents an ablation study analyzing the contributions of the DTAI loss and auxiliary classifier to the proposed method's performance. Figure 6 presents Violin plots of the distribution of Design Target Achievement Index, Target Success Rate, and Minimum Target Ratio over the space of generated designs for baseline methods. A continuous distribution is approximated over the 250 designs generated by each method using a Kernel Density Estimate (KDE).

7.1 Feasibility Performance

GAN and CTGAN perform poorly in generating feasible designs. Many of the infeasible designs generated result from negative tube thicknesses, explaining interpolation's strong performance in feasibility since interpolating between positive thicknesses can never be negative. The difficulty to respect these

TABLE 1: Deep Generative Models scored on the eight proposed evaluation metrics. Models from left to right: Randomly sampled design subsets from the FRAMED dataset (Dataset), Random Interpolation between FRAMED designs (Interpolation), Vanilla GAN (GAN) Conditional Tabular GAN (CTGAN), Proposed PaDGAN with DTAI and auxiliary Geometric Feasibility Classifier (Proposed)

Metric	Dataset	Interpolation	GAN	CTGAN	Proposed
Mean Design Space Diversity (DSD)	**8.60**	**13.28**	**10.08**	**14.12**	**9.48**
Mean Performance Space Diversity (PSD)	**3.80**	**3.40**	**2.71**	**3.95**	**2.71**
Mean Design Novelty (DN)	**0.00**	**7.17**	**8.30**	**8.02**	**9.96**
Geometric Feasibility Rate (GFR) (%)	**100.0**	**100.0**	**65.2**	**65.7**	**95.9**
Hypervolume (HV) *10ˆ-7	**4.81**	**4.00**	**3.04**	**3.54**	**3.82**
Mean Design Target Achievement Index (DTAI)	**0.53**	**0.58**	**0.71**	**0.52**	**0.79**
Mean Target Success Rate (TSR) (%)	**24.0**	**31.3**	**28.6**	**24.4**	**69.8**
Mean Minimum Target Ratio (MTR)	**0.32**	**0.37**	**0.41**	**0.32**	**0.43**

TABLE 2: Ablation Study Contrasting Proposed PaDGAN with DTAI training loss and auxiliary classifier against PaDGAN without DTAI (-DTAI), without the auxiliary classifier (-CLF), and MO-PaDGAN (-DTAI, -CLF).

	Proposed	-DTAI	-CLF	-DTAI, -CLF
Mean Design Space Diversity (DSD)	**9.48**	**6.21**	**5.80**	**7.11**
Mean Performance Space Diversity (PSD)	**2.71**	**2.24**	**2.44**	**2.62**
Mean Design Novelty (DN)	**9.96**	**9.90**	**10.52**	**9.94**
Geometric Feasibility Rate (GFR) (%)	**95.9**	**82.5**	**83.2**	**87.1**
Hypervolume (HV) *10ˆ-7	**3.82**	**3.82**	**3.40**	**2.95**
Mean Design Target Achievement Index (DTAI)	**0.79**	**0.80**	**0.72**	**0.69**
Mean Target Success Rate (TSR) (%)	**69.8**	**67.7**	**51.4**	**48.4**
Mean Minimum Target Ratio (MTR)	**0.43**	**0.51**	**0.46**	**0.42**

feasibility constraints reflects the GAN and CTGAN's difficulty to learn sharp thresholds. In contrast, the auxiliary classier in the proposed method provides the model with a strong gradient signal to avoid infeasible regions and shift the model's learned thresholds.

7.2 Target Achievement Performance

Interpolation, GAN, and CTGAN all perform poorly in Target Success Rate (TSR), each achieving less than 1 in 3 design targets on average per design. In the proposed method, DTAI guides PaDGAN training to specifically achieve design targets, scoring just under 70% design feasibility on average. Its distributions across designs are also significantly more consistent than competing objectives, as seen in Figure 6. Using DTAI as a loss unsurprisingly improves DTAI scores, which themselves are highly reflective of target achievement performance. Interestingly, the standard GAN achieves moderately higher DTAI performance than the dataset, perhaps because it fails to capture less-conventional portions of the design space that may tend towards lower performance. PaDGAN with DTAI also achieves the best average Minimum Target Ratio (MTR), indicating that it is coming closer to targets across all objectives. This improvement in MTR is less pronounced than TSR or DTAI, potentially because MTR doesn't reflect the designer's weighting of objective importance, while the DTAI loss guiding training does. Nonetheless, the improvement of MTR over baselines is still significant.

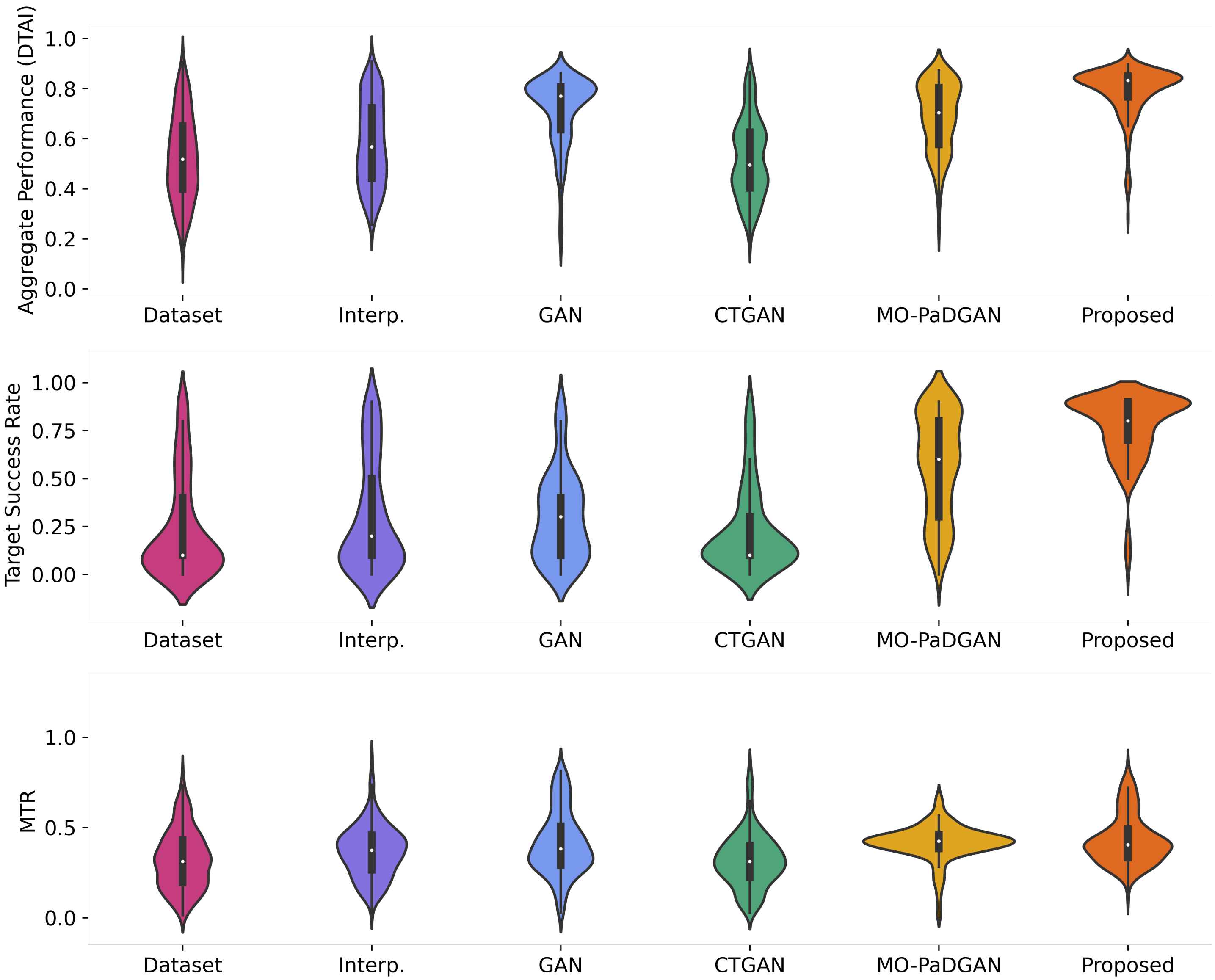

FIGURE 6: We demonstrate violin plots over the set of designs synthesized by each generative method for several metrics. Approximate distributions generated using Kernel Density Estimates are denoted with the colored curves. The black vertical boxes denote the interquartile range of the distribution, while the thin vertical line denotes the 5%-95% confidence interval. The median is denoted with a white point.

PaDGAN with DTAI even improves in Hypervolume scores over GAN and CTGAN. This indicates that despite the emphasis on targeted improvements to objectives, the overall performance of the generated set across all objectives is increased.

7.3 Ablation Study

The ablation study shown in Table 2 contrasts the proposed PaDGAN with DTAI and auxiliary classifier against variants without the classifier (-CLF) and using MO-PaDGAN's random objective weighting instead of DTAI (-DTAI). We note that the standard MO-PaDGAN (-DTAI, -CLF) was arguably the previous state-of-the-art in performance-aware design generation using DGMs. We outperform Mo-PaDGAN in every metric tested, with significant improvements in feasibility (95.9% from 87.1%), DTAI (0.79 from 0.69), Target Success Rate (TSR) (69.8% from 48.4%), and Hypervolume ($3.82E-7$

from $2.95E-7$).

8 LIMITATIONS AND FUTURE WORK

We demonstrated the sweeping improvements that can be achieved by incorporating feasibility and performance into DGM training losses. As discussed in Section 2, an inherent limitation with this approach is that the entire loss calculation procedure must be differentiable, which necessitates differentiable evaluation functions and surrogate models. Developing higher performing surrogates or even rapid differentiable numerical simulations is a promising approach to improve DGM performance.

Since scoring methods requires time-intensive numerical simulation, we were limited in the tuning of the generative methods tested, and acknowledge that results may vary depending on the initialization of methods. While we mitigated this uncertainty by simulating three batches of 250 designs from three instantiations of each method and taking median values, moderate variability in results can be expected. Future work may include testing larger number of samples and more runs to improve the confidence in the results, as well as exploring effects of hyperparameter selections.

This work leaves many avenues for further expansion. Training using only a high-performing subset of bicycle frames from the dataset may improve the performance of the DGMs tested, particularly the vanilla GAN and the tabular generation algorithms, as they factor in no notion of design performance. Testing the DTAI metric with other performance-aware generative frameworks besides PaDGAN would also be valuable. Finally, testing more methods and datasets would yield a more complete perspective. We encourage researchers to test new generative methods using our evaluation metrics on the FRAMED dataset or other datasets of their choice. We also hope to compare and contrast DGMs tested in this work against Multi-Objective-Optimization approaches such as NSGA-II [19].

Throughout the testing of DGMs in this work, we evaluated feasibility of roughly 14,000 bicycle frame designs and simulated roughly 10,000 using the same methodology as the FRAMED dataset. While these DGM-generated datapoints should be used with caution as training data for other DGMs to limit potential bias, they may serve as an excellent addition to the FRAMED dataset for supervised learning tasks (and indirectly improve future DGM performance by improving accuracy and generalizability of surrogate models.

9 CONCLUSION

We introduce a novel differentiable scoring metric called Design Target Achievement Index (DTAI) which allows Deep Generative Models to prioritize, meet, and exceed multi-objective performance targets. We augment a Performance-Augmented Diverse GAN with our DTAI objective and demonstrate significantly improved performance in design generation. We then further augment this PaDGAN with an auxiliary classifier to encourage the generation of feasible results. To benchmark our method, we evaluate a variety of Deep Generative Models, including the Multi-Objective PaDGAN, and specialized tabular generation algorithms CTGAN and TVAE. Methods are tested on a challenging bicycle frame design problem with 10 performance objectives. To rigorously evaluate methods for diversity, novelty, constraint satisfaction, overall performance, and feasibility, we propose a comprehensive set of evaluation metrics and score all tested methods on these metrics. The proposed PaDGAN with DTAI loss and auxiliary classifier significantly outperforms baselines in most performance objectives and further outperforms other PaDGAN variants in ablation studies. All in all, this work establishes a novel Deep Generative Framework that actively optimizes performance, diversity, feasibility, and target satisfaction to establish a new state-of-the-art in design generation using Deep Generative Models.

10 ACKNOWLEDGMENTS

We would like to thank Amin Heyrani Nobari for creating the Tensorflow 2.x version of PaDGAN which we modified to generate our results. We also acknowledge MathWorks for supporting this research.

REFERENCES

[1] Regenwetter, L., Nobari, A. H., and Ahmed, F., 2021. "Deep generative models in engineering design: A review". *arXiv preprint arXiv:2110.10863*.

[2] Chen, W., and Ahmed, F., 2021. "Padgan: Learning to generate high-quality novel designs". *Journal of Mechanical Design, **143***(3), p. 031703.

[3] Chen, W., and Ahmed, F., 2021. "Mo-padgan: Reparameterizing engineering designs for augmented multi-objective optimization". *Applied Soft Computing, **113***, p. 107909.

[4] Chen, H., and Liu, X., 2021. "Geometry enhanced generative adversarial networks for random heterogeneous material representation". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21, ASME.

[5] Zhang, H., Yang, L., Li, C., Wu, B., and Wang, W., 2021. "Scaffoldgan: Synthesis of scaffold materials based on generative adversarial networks". *Computer-Aided Design, **138***, p. 103041.

[6] Yoo, S., Lee, S., Kim, S., Hwang, K. H., Park, J. H., and Kang, N., 2021. "Integrating deep learning into cad/cae system: generative design and evaluation of 3d conceptual wheel". *Structural and Multidisciplinary Optimization, **64***(4), pp. 2725–2747.

[7] Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N., 2019.

"Deep generative design: Integration of topology optimization and generative models". *Journal of Mechanical Design,* ***141***(11).

[8] Shu, D., Cunningham, J., Stump, G., Miller, S. W., Yukish, M. A., Simpson, T. W., and Tucker, C. S., 2020. "3d design using generative adversarial networks and physics-based validation". *Journal of Mechanical Design,* ***142***(7), p. 071701.

[9] Fujita, K., Minowa, K., Nomaguchi, Y., Yamasaki, S., and Yaji, K., 2021. "Design concept generation with variational deep embedding over comprehensive optimization". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC-21, ASME.

[10] Dering, M., Cunningham, J., Desai, R., Yukish, M. A., Simpson, T. W., and Tucker, C. S., 2018. "A physics-based virtual environment for enhancing the quality of deep generative designs". In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 51753, American Society of Mechanical Engineers, p. V02AT03A015.

[11] Rawat, S., and Shen, M. H., 2019. Application of adversarial networks for 3d structural topology optimization. Tech. rep., SAE Technical Paper.

[12] Guo, T., Lohan, D. J., Cang, R., Ren, M. Y., and Allison, J. T., 2018. "An indirect design representation for topology optimization using variational autoencoder and style transfer". In 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, p. 0804.

[13] Dong, Y., Li, D., Zhang, C., Wu, C., Wang, H., Xin, M., Cheng, J., and Lin, J., 2020. "Inverse design of two-dimensional graphene/h-bn hybrids by a regressional and conditional gan". *Carbon,* ***169***, pp. 9–16.

[14] Li, X., Ning, S., Liu, Z., Yan, Z., Luo, C., and Zhuang, Z., 2020. "Designing phononic crystal with anticipated band gap through a deep learning based data-driven method". *Computer Methods in Applied Mechanics and Engineering,* ***361***, p. 112737.

[15] Liu, Z., Raju, L., Zhu, D., and Cai, W., 2020. "A hybrid strategy for the discovery and design of photonic structures". *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* ***10***(1), pp. 126–135.

[16] Regenwetter, L., Curry, B., and Ahmed, F., 2021. "BIKED: A Dataset for Computational Bicycle Design With Machine Learning Benchmarks". *Journal of Mechanical Design,* ***144***(3), 10. 031706.

[17] Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K., 2019. "Modeling tabular data using conditional gan". *Advances in Neural Information Processing Systems,* ***32***.

[18] Heyrani Nobari, A., Chen, W., and Ahmed, F., 2021. "Pcdgan: A continuous conditional diverse generative adversarial network for inverse design". In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 606–616.

[19] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002. "A fast and elitist multiobjective genetic algorithm: Nsga-ii". *IEEE transactions on evolutionary computation,* ***6***(2), pp. 182–197.

[20] Bringmann, K., and Friedrich, T., 2013. "Approximation quality of the hypervolume indicator". *Artificial Intelligence,* ***195***, pp. 265–290.

[21] Auger, A., Bader, J., Brockhoff, D., and Zitzler, E., 2009. "Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point". In Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms, pp. 87–102.

[22] Regenwetter, L., Weaver, C., and Ahmed, F., 2022. "Framed: Data-driven structural performance analysis of community-designed bicycle frames". *arXiv preprint arXiv:2201.10459*.

[23] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014. "Generative adversarial nets". *Advances in neural information processing systems,* ***27***.

[24] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T., 2020. "Analyzing and improving the image quality of stylegan". In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8110–8119.

