BIKED: A DATASET AND MACHINE LEARNING BENCHMARKS FOR DATA-DRIVEN BICYCLE DESIGN

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

Brent Curry  
BikeCAD.ca  
Ottawa, Ontario, Canada  
K1J 6E9  
Email: info@bikecad.ca

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

ABSTRACT

In this paper, we present “BIKED,” a dataset comprised of 4500 individually designed bicycle models sourced from hundreds of designers. We expect BIKED to enable a variety of data-driven design applications for bicycles and generally support the development of data-driven design methods. The dataset is comprised of a variety of design information including assembly images, component images, numerical design parameters, and class labels. In this paper, we first discuss the processing of the dataset and present the various features provided. We then illustrate the scale, variety, and structure of the data using several unsupervised clustering studies. Next, we explore a variety of data-driven applications. We provide baseline classification performance for 10 algorithms trained on differing amounts of training data. We then contrast classification performance of three deep neural networks using parametric data, image data, and a combination of the two. Using one of the trained classification models, we conduct a Shapley Additive Explanations Analysis to better understand the extent to which certain design parameters impact classification predictions. Next, we test bike reconstruction and design synthesis using two Variational Autoencoders (VAEs) trained on images and parametric data. We furthermore contrast the performance of interpolation and extrapolation tasks in the original parameter space and the latent space of a VAE. Finally, we discuss some exciting possibilities for other applications beyond the few actively explored in this paper and summarize overall strengths and weaknesses of the dataset.

1 INTRODUCTION

Transportation has historically been one of the fastest-growing sources of greenhouse gas emissions, currently accounting for as much as 14% of the total [1]. While there are at least 580 million privately owned bicycles (bikes) worldwide [2], further increasing bicycle use has the potential to significantly reduce greenhouse gas emissions. We believe that enabling a bicycle design process that emphasizes customization for targeted use is essential to increase ridership.

Every cyclist has a different “optimum” bicycle. Bicycles have a broad set of use cases that lend themselves towards drastically different designs. Furthermore, the bicycle is a machine that interfaces very precisely with the human body and every human body is unique. Unfortunately, custom designed and fitted bicycles are prohibitively expensive for many users. This cost stems from the fact that custom bicycles are often individually designed by tradespeople with years of experience. These designers typically approach their trade on a bike-by-bike basis, leveraging domain expertise, specialized design tools, and parametric guidelines to design a single bike for a single user. This narrow approach allows designers to learn from their own experience, but makes it difficult to tap into the lessons of other designers’ projects, effectively wasting much of the communal bicycle design experience. Conversely, in an optimal scenario, the lessons from every bicycle designed would improve the collective expertise of the bicycle design community as a whole. In addition, the current status quo in custom bike design is simply
unable to match the massive and growing global bike market, making custom bikes unavailable to most potential cyclists.

We believe data-driven design is a promising approach to help mitigate this issue. Data-driven design is a promising field that harnesses data-driven methods to enhance the tools and capabilities of modern design engineers. Applying data-driven methods to bicycle design will serve to: 1) Accelerate the progression of the bicycle design field by allowing designers to tap into collective expertise. 2) Address the growing rift between availability and demand for custom-fitted bikes by making custom bikes more affordable and accessible. This development would take a step towards addressing global societal needs like reducing carbon emissions and making better fitting bikes available to underprivileged socioeconomic groups.

BIKED is organized and curated specifically for data-driven design. Data is sourced from a rich archive of CAD files from the specialized BikeCAD software, which is primarily used by professional frame designers and bicycle enthusiasts. A thorough sequential data curation process was carried out as detailed in Section 3 to extract parametric data that specifically pertains to model design. As we demonstrate in Section 4, this parametric data captures the rich variations between designs in the dataset and is highly revealing of many parameters of interest like bicycle type.

Rapid advancements in data-driven research fields are often catalyzed by the introduction of quality publicly-available datasets. We hope BIKED will inspire other researchers to use standardized problems and datasets for advancing research in data-driven design applications and algorithm development. To that end, we provide a suite of algorithms (variational autoencoders, convolutional neural networks, ensemble classifiers, etc.) and their performance values for multiple data-driven design problems in Section 5. Section 5.1 demonstrates bicycle classification methods based on parametric or image data. We provide baseline results for 10 classification algorithms and discuss the scalability of their performance with availability of data. We then construct, tune, and train three deep networks to predict class results based on different types of input data provided within BIKED. These deep networks provide a baseline for higher performance models and yield valuable insight into the predictive information contained within the parametric and image data. In addition to maximizing performance, it is crucial for practitioners to understand the patterns identified by machine learning. To demonstrate a case of interpretable machine learning, using one of our classification models, we explore how individual parameters impact class predictions using Shapley Additive Explanations 3 and discuss the design implications of our results.

Next, in Section 5.2 we present applications and baseline results of Variational Autoencoders (VAEs) applied to BIKED data. VAEs support a host of applications including dimensionality reduction, novel data synthesis, interpolation, and extrapolation. We train one VAE on BIKED’s parametric data and one on BIKED’s image data and discuss the performance of each model on the aforementioned tasks. We further explore the concept of bicycle synthesis in Section 5.3 by contrasting several methods for synthesis including random sampling, interpolation, and extrapolation in both the original parameter space and the latent space of the parametric VAE. Finally, in addition to the numerous applications discussed, some other exciting applications that we envision are also presented in Section 6 along with some key strengths and limitations of BIKED.

2 BACKGROUND AND PREVIOUS WORK

In this section, we discuss the background and related work in bicycle design optimization, compare BIKED to other datasets commonly used for data-driven design, and discuss the BikeCAD software and design archive.

2.1 Bicycle Design and Optimization

Bicycle design and optimization is a well-researched field. Countless treatises have explored principles of bicycle design in the centuries since the introduction of the predecessors to the modern bicycle 4, 5. Today, significant research effort is dedicated to improving bicycle aerodynamics 6, 8 and structure 9. Other studies explore practices of bicycle sizing and fitting 10, 11. While some make use of the wide availability of anthropometric data, many are limited by the availability of bicycle models. We expect that a quality dataset of bike models including comprehensive parametric design information will enhance future simulation-based bicycle design studies.

2.2 Comparison to Other Datasets

Research in this domain of data-driven design often taps into 3D model datasets like ShapeNet 12 and Princeton ModelNet 13. These datasets are particularly useful in 3D shape synthesis applications. Though some of this shape synthesis research considers adherence to physical laws, the nature of these datasets tends to emphasize research focused on visual fidelity rather than physical function 14, 17. In contrast, BIKED contains thousands of design parameters that explicitly specify design information ranging from local component geometry to overall design layout. Certain design information can be extracted from 3D models, but the level of detail falls short of the comprehensive numerical design parameters that BIKED provides. In addition, BIKED pulls models from a largely experienced and qualified set of designers, improving the quality of models featured in the dataset.

2.3 BikeCAD

BikeCAD is a parametric computer-aided design (CAD) software optimized for bicycle design. It features a live-updating
model and numerous design menus that help users customize bicycle geometry and features, as shown in Figure 1. The BikeCAD website features an open archive of user-submitted BikeCAD models [18]. Since the archive’s inception in 2011, the archive had accrued approximately 5080 bicycle models as of July 2020. The most recent 4800 at this time point were used in the dataset. Figure 2 shows tile images of several randomly sampled bikes from the dataset. Since designs span many iterations of the BikeCAD software, the advancement of the program over the years has enabled increasingly complex designs in the more recent models on the archive. Each entry contains the BikeCAD file, an image, a rating out of 5, the BikeCAD version the model was made in and a model identification number (ID). Additionally, some models in the archive contain the model ID that they are based on and a few design parameters. Only the BikeCAD files from the archive were used in BIKED. Rating information was not included since most models are unrated and few models have more than a single rating. Additionally, though tracking the model IDs that each model is based on would make for an interesting graph of design progression, many model ID links are no longer functional since their referenced models have been removed from the archive. Nonetheless, tracking this information would be an interesting extension of the dataset but is not currently included.

Since BikeCAD’s primary user base is comprised of frame builders and bicycle enthusiasts, many models are professional-quality designs and are furthermore self selected as showcase models. Note as well that the BikeCAD software promotes quality through realistic default values, comprehensive design and analysis tools such as toe overlap and lean angle analysis, as well as a suite of tools to check the model’s adherence to Union Cycliste Internationale (UCI) standards. All in all, this combination of factors suggests a reasonably high quality for the majority of dataset models.

3 METHODOLOGY

In this section, we discuss the various steps taken to generate and process the dataset. An overall flow diagram of the data creation progression is shown in Figure 3 and the following labeled sections correspond to the processing operations listed.

3.1 Sourcing from Design Archive

In order to process the dataset locally, the most recent 4800 models were downloaded from the design archive in July 2020. The oldest 280 were unused. Original ordering on the design archive was preserved in the subsequent processing of the data.

3.2 File Standardization in BikeCAD

BikeCAD files are XML files structured as a collection of data parameters with corresponding values. Certain parameters have different meanings depending on flags in other parts of the file. This makes a direct comparison between parameters difficult. To ensure that the parameters in the files referred to the same geometric dimensions across different models, BikeCAD’s internal conversion formulas were applied using a custom version of the BikeCAD software. 9 files were corrupt or unprocessable leaving 4791 successfully standardized models.

3.3 Bicycle Assembly and Component Image Exports

In order to remove dimensional labels or backgrounds from the bike models’ corresponding images, we directly edited label visibility and set plain backgrounds in the BikeCAD files, then exported these clean bike images. Next, segmented bike images were exported by editing component visibility in the BikeCAD files. Bikes were then segmented into 5 essential components (frame, saddle, handlebars, wheels, cranks) and two nonessential components (cargo racks, bottles) that only appeared in some models. 4510 bikes were compatible with this process, yielding 5 sets of 4510 component images (essential components), one set of 965 (bottles), and one set of 346 (racks). An example segmentation on one bike model is demonstrated in Figure 4.
3.4 Data Extraction from BikeCAD Files

For organizational purposes and ease of processing, we parsed the individual standardized BikeCAD files and compiled all parameter data in a tabular data structure. In total, 23818 unique parameters (or features) were collected from the files, though many parameters were quite sparse across the model space since certain models had no entries for many parameters.

3.5 Parameter Space Reduction

Lowering the dimensionality of a dataset can help reduce computational cost and memory requirements when applying data-driven models. Given that our original data was 23818 dimensional, dimensionality reduction was critical for efficient data-driven analysis. To reduce the dimensionality of the parameter space, several parameters were removed from consideration. First, parameters with no bearing on bicycle design such as color, text, and positioning of dimensional labels were identified and dropped from the data. Next, parameters defining additional non-standard tubes, water bottles, or tandem features were removed, and any bike with a data value under one of these fields was dropped. This removed some unique designs but drastically reduced the dimensionality of the data by dropping sparsely inhabited parameters. Additionally, each model was manually considered and any designs that were not bikes (scooters, motorcycles, wheelchairs, etc.) were dropped. Finally, any parameters that were empty for every model or for which every model had the same value were dropped. After these steps, 4512 bikes of 4791...
and 1352 parameters of 23813 remained, resulting in a 94.3% reduction in dimensionality.

Some bike models lacked one or more essential components, which in practice implies that the designer turned off the visibility of certain components in the software. For example, numerous models originally had only a frame visible, presumably implying the designer was only intending to design a frame. Since we did not want incomplete models in the dataset, critical components of bike models were used in the processing of the dataset regardless of whether they were hidden or not. In other words, visibility parameters of critical components were dropped from the parameter space. In cases like the standalone bike frame designs, it was unclear if the hidden components like wheels, saddle, and handlebars received any design consideration, but were included nonetheless. As such, some models may include components not intended to be part of a final design. In contrast, nonessential features like bottle holders and fenders were included or excluded in the parameter space based on the visibility in the original model.

3.6 Other Processing Steps

Following the parameter space reduction, the parameter space consisted of four types of data: continuous, discrete, boolean, and categorical. Continuous and discrete parameters reflect some ordinal significance (i.e. similar numerical values imply actual real-world similarity in that parameter). In contrast, categorical values imply no such ordinal significance even if represented numerically. Most data types in the parameter space could be easily classified (i.e. floating point numbers are continuous). However, integer valued parameters had to be individually considered and manually classified as categorical or discrete. Additionally, certain variables that would ideally be continuous had to instead be treated as categorical. One such example was the bike “size” parameter, since it contained a medley of words, measurements of different units, and measurements with ambiguous units (“xxl”, “623EET”, “Huge”, “52cm”, “26”, etc.). These steps left us with the distribution of variables shown in Table 1. Since many standard machine learning methods do not support categorical data, we applied a commonly used method called one-hot encoding, which converts a categorical variable into a set of n boolean parameters.

Next, missing and unknown values were imputed using mean imputation. Values with a magnitude above a selected threshold were also discarded and then considered to be missing, as they were likely caused by corrupted data or data type issues. A cutoff magnitude of 10,000 was selected since most units were in millimeters or inches, and bikes were unlikely to be designed with component dimensions over 10 meters. This final imputed data is the version of the data used in the discussion and sample applications discussed henceforth.

### Table 1: Assortment of types of parameters present in dataset before and after one-hot encoding

<table>
<thead>
<tr>
<th>Parameter Type</th>
<th>Original</th>
<th>Perc.</th>
<th>Original One-Hot Encoded</th>
<th>Perc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cont. Variables</td>
<td>998</td>
<td>73.6%</td>
<td>998</td>
<td>37.8%</td>
</tr>
<tr>
<td>Booleans</td>
<td>155</td>
<td>11.4%</td>
<td>1593</td>
<td>60.3%</td>
</tr>
<tr>
<td>Categ. Variables</td>
<td>142</td>
<td>10.4%</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Discrete. Variables</td>
<td>49</td>
<td>4.5%</td>
<td>49</td>
<td>1.9%</td>
</tr>
</tbody>
</table>

3.7 Image Regeneration

Because parametric data is difficult to quickly interpret, we developed a method to quickly regenerate images from parametric data, as shown in Figure 5. We use this method to generate corresponding images from the fully processed parameter data. This method also applies to newly generated parametric data, which can assist in visualizing interpolation or novel bike generation results as we demonstrate in Section 5.3. In this process, the one-hot encoding is reversed by taking the most probable category to be the absolute truth. In the original dataset, one-hot vectors are populated with boolean values, but the method also supports probabilistic values to better accommodate generative methods. Next, bike data is inserted into a BikeCAD file template to generate new BikeCAD files from the standardized parametric data. Any fields from the template file that are present in the parametric data are overwritten and any fields that are absent are left at default values. These BikeCAD files are then opened in the software to export corresponding images. Since certain features like component colorings were dropped as described in Section 3.5, this regeneration process yields images with standardized colors as dictated by the template file. We also note that while the process supports probabilistic values, forcing probabilistic distributions into a single concrete class can lead to discontinuities, especially in tasks like interpolation.

4 DATA CHARACTERISTICS

The processed dataset has a number of parameters that can be used as labels for different learning tasks, such as bike style, model year, or individual component types. Note that these labels are neither rigorous nor standardized, since all models are classified under the individual designers’ discretion. Table 2 shows the distribution of bicycle “styles” in the dataset. We observe that road bikes are by far the most prevalent (40.56%) class as labeled by the designers.

To further explore the user defined style labels of the dataset, we explore unsupervised clustering algorithms. Clustering can help identify groups of similar models in a dataset and identify which classes are similar or dissimilar to others. We apply the t-Distributed Stochastic Neighbor Embedding (t-SNE) clustering method [19] to the parameter space of the processed dataset.
FIGURE 5: Process to generate images from parametric data. One-hot encoding is first reversed, taking the class value with highest probability value. Data is then inserted into a template BikeCAD file and finally reprocessed in the software to export images.

TABLE 2: Assortment of bike classes present in the processed dataset. Mountain bikes are abbreviated as "MTB." Bikes classified as “OTHER” are labelled as such by the designer. The “Remaining Label Categories” group contains the remaining 10 explicit categories in descending order of prevalence: BMX, CITY, COMMUTER, CRUISER, HYBRID, TRIALS, CARGO, GRAVEL, TANDEM, CHILDRENS, FAT.

<table>
<thead>
<tr>
<th>Bike Style</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROAD</td>
<td>1941</td>
<td>40.56</td>
</tr>
<tr>
<td>MTB</td>
<td>708</td>
<td>14.79</td>
</tr>
<tr>
<td>TRACK</td>
<td>479</td>
<td>10.01</td>
</tr>
<tr>
<td>OTHER</td>
<td>338</td>
<td>7.06</td>
</tr>
<tr>
<td>DIRT_JUMP</td>
<td>296</td>
<td>6.18</td>
</tr>
<tr>
<td>TOURING</td>
<td>207</td>
<td>4.32</td>
</tr>
<tr>
<td>CYCLOCROSS</td>
<td>151</td>
<td>3.16</td>
</tr>
<tr>
<td>POLO</td>
<td>129</td>
<td>2.70</td>
</tr>
<tr>
<td>TIMETRIAL</td>
<td>99</td>
<td>2.09</td>
</tr>
<tr>
<td>Remaining Label Categories</td>
<td>438</td>
<td>9.15</td>
</tr>
</tbody>
</table>

SNE is a clustering algorithm that projects samples onto a lower dimensional space while clustering similar models and distancing dissimilar ones. An annotated plot of such a clustering is included in Figure 6.

Next, we consider a more targeted approach by eliminating all parameters except the 50 deemed to be most impactful on bicycle classification. These specific parameters were determined using a SHAP analysis as discussed in Section 5.1. To better visualize the relationship between classes of bikes, we plot only the “median” bike of each class as a representative. We determine this “median” bike by applying min-max normalization to every parameter, taking the median of each normalized parameter over all bikes of that class and finding the nearest model to this median value by euclidean distance. The results of a Principle Component Analysis (PCA) under these conditions are included in Figure 7. In this plot, we can observe that bikes with components and structure more similar to a conventional road bike tend towards the bottom right corner. Bikes with large wheels relative to absolute size tend toward the top. Other patterns and relations can also be discerned.

5 DATA-DRIVEN DESIGN APPLICATIONS

In this section, we demonstrate several sample applications supported by the dataset. In the first, we train ten classification models to predict bicycle “style.” In the second, we train variational autoencoders on BIKED and test them on reconstruction and design synthesis tasks. Finally, we explore methods to interpolate between bike models in the dataset. We selected these applications to provide baseline performance for several classic machine learning tasks and to inspire more advanced research directions using the dataset.

5.1 Bike Classification

For machines to create new designs, it is important for them to identify key characteristics which define a new unseen bicycle design and accurately predict which real-world style of design best describes it. In this application, we explore various methods to address the bicycle style classification task. Training a model to predict bicycle style can be a method to predict best use cases for particular bicycle models.

Classification using parametric data: In the first part of this study, we contrast the performance of numerous classification methods trained on differing amounts of parametric data and ignoring image data. Table 3 shows the results of this study. In total, 10 classification models were tested on 200, 600, 1200, 2000, and 3000 randomly sampled data points from the 4532 processed models. In each test, any model not used for training was used for validation. Neural networks tested have 200 neurons per layer, use Leaky Rectified Linear Unit (ReLu) activation functions [20], and are trained using the adam optimizer [21]. For implementation details of the methods tested, we refer the reader to our provided code. Since bicycle class labels are highly subjective, and many classes overlap or even contain one another, a relatively low classification accuracy is expected. Of the methods tested, the 6-layer neural network was the highest performer, with 67.7% accuracy. Classification accuracy improved at higher quantities of data, especially for more complex models. This would suggest that additional bicycle samples continue to provide valuable information to classification models, and a future expanded dataset with new BikeCAD archive models would likely be valuable.
FIGURE 6: Visualization of the 2-dimensional embedding generated through t-Distributed Stochastic Neighbor embedding of parametric data. Horizontal and vertical axes denote the two embedding dimensions.

TABLE 3: Classification accuracy (%) results for various classification models using differing quantities of training data points. Classification accuracy values are aggregated over several training iterations for each model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Size</th>
<th>200</th>
<th>600</th>
<th>1200</th>
<th>2000</th>
<th>3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>44.1</td>
<td>44.1</td>
<td>42.5</td>
<td>41.6</td>
<td>40.7</td>
<td></td>
</tr>
<tr>
<td>AdaBoost</td>
<td>48.2</td>
<td>47.2</td>
<td>48.5</td>
<td>47.7</td>
<td>49.2</td>
<td></td>
</tr>
<tr>
<td>K neighbors</td>
<td>47.4</td>
<td>49.1</td>
<td>53.8</td>
<td>55.6</td>
<td>56.4</td>
<td></td>
</tr>
<tr>
<td>Support Vector Clf.</td>
<td>48.1</td>
<td>51.6</td>
<td>56.1</td>
<td>59.3</td>
<td>61.8</td>
<td></td>
</tr>
<tr>
<td>Depth 5 Dec. Tree</td>
<td>52.1</td>
<td>59.2</td>
<td>62.3</td>
<td>64.0</td>
<td>64.8</td>
<td></td>
</tr>
<tr>
<td>5-layer Neural Net</td>
<td>39.3</td>
<td>54.3</td>
<td>59.5</td>
<td>61.8</td>
<td>66.4</td>
<td></td>
</tr>
<tr>
<td>Depth 8 Dec. Tree</td>
<td>50.9</td>
<td>58.3</td>
<td>62.1</td>
<td>64.8</td>
<td>66.5</td>
<td></td>
</tr>
<tr>
<td>Gaussian Process Clf.</td>
<td>10.7</td>
<td>60.4</td>
<td>64.2</td>
<td>65.6</td>
<td>67.0</td>
<td></td>
</tr>
<tr>
<td>Random Forest</td>
<td>59.3</td>
<td><strong>64.1</strong></td>
<td><strong>65.7</strong></td>
<td><strong>66.4</strong></td>
<td><strong>67.7</strong></td>
<td></td>
</tr>
<tr>
<td>6-layer Neural Net</td>
<td>43.5</td>
<td>53.1</td>
<td>61.0</td>
<td>63.0</td>
<td><strong>67.7</strong></td>
<td></td>
</tr>
</tbody>
</table>

Comparison of classification using image and parametric data: The second part of this classification study examines the relative predictive value of images versus parametric data. Three deep neural networks were trained for the bicycle classification task with a train to test ratio of 3:1. The first predicted bicycle classes with 75.5% accuracy using only parametric data. The second predicted bicycle classes with 73.0% accuracy using only image data. The third predicted classes using both parametric and image data with an accuracy of 71.6%. Network diagrams for each network are included in Figure 8. The classification accuracy using only parametric data was superior to the accuracy using only images, suggesting that the parametric data is far richer in meaningful information than the image data. Though we would expect the classifier using both images and parametric data to yield the highest accuracy, it ended up being prone to overfitting and underperformed both of its competitors.

Interpretability analysis to identify design attributes: The final part of this classification study examines the results of a Shapley Additive Explanations analysis. This analysis shows the average impact of a particular feature on the classification probability for each class type. Understanding the classification impact of individual parametric features can help designers appreciate the particular factors that define a particular class of bike. Figure 9 shows the results of the SHAP analysis on the highest per-
forming deep neural network which scored 75.5% classification accuracy. We observe that each of the most significant parameters besides number of cogs is a boolean type. Many directly relate to the style of a particular component. The significance of many of these parameters makes intuitive sense. Handlebar, fork, and dropout style are major distinguishing factors between bike classes and most designers of a particular class of bike will choose to use corresponding classes of components for their design. Humans would probably not immediately consider number of cogs, material, presence of derailleur, or curvature of fork when classifying bikes, but would likely acknowledge their significance. We can also observe that a few features have a particularly large impact on the probability that a bike is classified as a particular type, like track style dropout spacing for track bikes or suspension forks for mountain bikes. Such SHAP Analyses are reasonably consistent between different instances of the same network architecture, but the importance ranks of the top 20 parameters often fluctuate in ranking by 3-5 places.

5.2 Bicycle Synthesis using Variational Autoencoders

In this case study, we explore the applications of BIKED through Variational Autoencoders (VAEs). VAEs encode input data into a low dimensional latent space and decode images from latent space vectors \[^{22,23}\]. In training, the decoder is fed reparametrized latent vectors from the encoder and attempts to reconstruct the input. The decoder can also be used for pure generation tasks by decoding randomly sampled latent vectors. We implement two VAEs, one using image data to generate images and one using parametric data to generate parametric data. Figure 10 shows the architecture of the VAE used for images. While the results of the image VAE are easy to visualize, the performance of the parametric VAE is difficult to evaluate, so images are generated from the parametric data that it generates using the method discussed in Section 3.7. Figure 11 shows the results of the trained VAEs on a reconstruction task. The parametric VAE tends to generate more realistic images, largely due to the image generation process in BikeCAD, however the image VAE tends to reconstruct the input much more accurately. This is largely because the image VAE is directly trained to reconstruct visual appearance while the parametric VAE is only trained to do so indirectly. Critical to this observation is the fact that the parametric VAE knows no weighting of the relative importance (or particularly visual importance) of features. For example, the number of
FIGURE 9: Shapley Additive Explanations analysis of feature significance on model classification result. The top 10 most significant features are shown on the vertical axis and the mean impact to classification probability is shown on the horizontal axis. Since parameters are one-hot encoded, we list boolean values describing whether or not a categorical parameter falls into a particular class as "Parameter: Class." Note that a high significance does not imply a positive correlation.

teeth on the third rear cog is just as important to the VAE as the handlebar type. Weighting for this relative importance could be a future line of inquiry. We also experienced that the parametric VAE sometimes struggles with training issues like posterior collapse, which can be sensitive to initialization and reduce reliability. The VAEs can also generate entirely novel bikes by randomly sampling the latent space and propagating the sampled latent vector through the decoder. The parametric VAE’s generation performance is demonstrated in Figure 12. Generated models are very realistic, but are not as varied in their features as the original dataset.

5.3 Bicycle Synthesis using Design Space Interpolation and Extrapolation

In this subsection, we explore methods for creating new bicycle designs through interpolation and extrapolation of existing dataset models in the design space. Bikes can be directly interpolated or extrapolated in the VAE’s latent space or in the parameter space, yielding full sets of bicycle parameters which can be converted into images for visualization using the process discussed in Section 3.7. Generated images for an interpolation and extrapolation example are shown in Figure 13. Interpolation between two designs in the parameter space typically works well, but extrapolation often becomes unrealistic, generating bikes that are sometimes disconnected, as shown in the bottom two images on the left. Interpolation and extrapolation both work well in the latent space of the VAE, with extrapolation results generally being quite realistic. In the example shown, extrapolating on the road bike in a direction opposite the mountain bike shifts the model to begin resembling a timetrial (triathlon) bike.

FIGURE 10: Model architecture of Variational Autoencoder for images. The encoder (above) converts an image input into a 64-dimensional latent vector in the embedding space. The decoder (below) converts a latent vector into a generated image. The architecture of the parametric VAE is similar, but uses five fully connected layers for the encoder and decoder and no convolution or convolution transpose layers.

6 DISCUSSION

In this section, we discuss other applications that we anticipate and some ideas to expand BIKE. We then discuss key strengths and limitations of the dataset.
6.1 Other Extensions and Applications

We envision many exciting ways to expand upon BIKED using some of the methods and features discussed. For example, using the encoded latent space for interpolation or extrapolation may prove a suitable method for dataset augmentation through the generation of novel bikes. Component-wise bicycle synthesis using segmented bike parts is another interesting concept. We are also anticipating new BikeCAD functionality that will support the generation of 3D CAD files from BikeCAD models, which may provide a means to easily convert original or synthesized models into 3D representations using meshes, point clouds, or voxels.

We also envision many exciting applications for the dataset beyond the examples demonstrated above. In particular, we anticipate that bicycle performance quantification can be hugely enhanced using data-driven surrogate modeling. Such surrogate models can rapidly estimate the performance with regard to certain criteria, like the aerodynamic drag or structural rigidity. Using these data-enabled performance quantification methods, we can further envision an all-encompassing algorithm that can design an ideal bicycle for a particular user, given that user’s intended riding conditions and their physical dimensions.

6.2 Strengths and Novelty

The BikeCAD dataset’s emphasis on thorough and comprehensive design information makes it well suited to a multitude of data-driven design tasks. Combined with rich component segmentation data, the inclusion of extensive parametric design information enables data driven tasks that would be impossible using one of the several existing 3D model datasets. Furthermore, as we discussed in Section 2.3, the combination of numerous factors like the demographics of the BikeCAD userbase and the inbuilt software features would indicate that the majority of BIKED models are of good or even professional quality.
6.3 Limitations

As is the case in most datasets, BIKED has several sources of bias to note. Since all bike designs were modeled in BikeCAD, there is a bias towards the default bike models that often serve as the starting point in a user’s design and modeling process. This bias may be noticeable at a high level in overall design characteristics, or at a low level with users leaving certain dimensions at their default values. This is especially true for more niche parameters that have little to no visual impact on the bike model, where novice users may not even understand the parameter’s meaning. This bias causes many parameters to have very uniform values across the vast majority of models, which can then cause issues, such as posterior collapse while training Variational Autoencoders. BIKED also has a slight temporal bias. This can be observed during clustering when models with similar model numbers tend to gravitate to one another. Since model numbers largely correlate to BikeCAD version used, updates to the software that introduced new parameters or updated old default values are likely the root cause. We also reiterate some of the potential biases discussed in Section 3.5. Dropping bikes with particular features like tandem components or extra nonstandard tubes may have introduced biases. Additionally, manually overriding component visibility of critical design components could have unintentionally included components that were not intended to be part of the designer’s vision. We have worked closely with bicycle experts and BikeCAD’s creator to use domain knowledge to alleviate as many sources of bias as possible and plan to continue refining our methods and dataset with the goal of providing a rich and accurate resource to inspire research on data-driven bicycle design.

We also note that, although BikeCAD primarily serves an experienced user base, there is no guarantee of quality or performance of any bike model. Some models are designed to be facetious or highly unrealistic. We would also advise users that the raw dataset may contain offensive or proprietary imagery and language since designers are free to use terminology and graphical designs of their choice.

7 CONCLUSION

In this paper, we have presented BIKED, a dataset for data-driven bicycle design. BIKED features 4791 bicycle models providing full bicycle assembly images, segmented component images, and extensive parametric data to enable various data-driven design approaches. Throughout the paper, we discussed the various processing steps taken to curate the data, and explored some features of the dataset through unsupervised clustering. We then demonstrated several data-driven applications for bicycle design to provide inspiration and baseline performance for researchers interested in using BIKED: First, we trained a variety of classification models to predict bicycle class, including 10 baseline models using parametric data and 3 custom tuned deep neural networks to contrast classification performance using different types of input data. Using one of these classification models, we performed a Shapley Additive Explanations analysis to understand the impact of individual parameters on classification predictions. Next, we trained two Variational Autoencoders to perform a variety of different tasks on both parametric and image data, including model reconstruction, latent space sampling, interpolation, and extrapolation. Finally, we contrasted interpolation and extrapolation performance in the original parameter space and the VAE latent space.

BIKED goes above and beyond existing datasets, providing
extensive design information of various types and promising to enable a slew of novel data-driven design applications. In our discussion, we have documented the process to curate the dataset, explored its features, provided baseline performance values for some common algorithms, and attempted to inspire researchers with sample applications. We hope that BIKED will prove a valuable resource to researchers in the data-driven design community and will support novel and innovative approaches in this rapidly growing field.

8 ACKNOWLEDGEMENTS

We would like to thank Professor Daniel Frey for his consistent input and guidance throughout the project. We would like to thank Kris Vu for assisting with sourcing files from the BikeCAD archive and Amin Heyrani Nobari for assisting with the exporting of component images. Finally, we would like to acknowledge Mathworks for supporting this research.

REFERENCES