Data-Driven Car Drag Prediction with Depth and Normal Renderings

Binyang Song*
Assistant Professor
Department of Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA, 24060
Email: binyangs@vt.edu

Chenyang Yuan
Toyota Research Institute, Cambridge, MA
Email: chenyang.yuan@tri.global

Frank Permenter
Toyota Research Institute, Cambridge, MA
Email: frank.permenter@tri.global

Nikos Arechiga
Toyota Research Institute, Los Altos, CA
Email: nikos.arechiga@tri.global

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

ABSTRACT

Generative AI models have made significant progress in automating the creation of 3D shapes, which has the potential to transform car design. In engineering design and optimization, evaluating engineering metrics is crucial. To make generative models performance-aware and enable them to create high-performing designs, surrogate modeling of these metrics is necessary. However, the currently used representations of 3D shapes either require extensive computational resources to learn or suffer from significant information loss, which impairs their effectiveness in surrogate modeling. To address this issue, we propose a new 2D representation of 3D shapes. We develop a surrogate drag model

*Address all correspondence related to ASME style format and figures to this author.

based on this representation to verify its effectiveness in predicting 3D car drag. We construct a diverse dataset of $4,535$ high-quality 3D car meshes labeled by drag coefficients computed from computational fluid dynamics simulations to train our model. Our experiments demonstrate that our model can accurately and efficiently evaluate drag coefficients with an R^2 value above 0.84 for various car categories. Our model is implemented using deep neural networks, making it compatible with recent AI image generation tools (such as Stable Diffusion) and a significant step towards the automatic generation of drag-optimized car designs. Moreover, we demonstrate a case study using the proposed surrogate model to guide a diffusion-based deep generative model for drag-optimized car body synthesis. We have made the dataset and code publicly available at https://decode.mit.edu/projects/dragprediction/.

1 INTRODUCTION

Engineers often need to work with three-dimensional (3D) representations of an object for design, evaluation, and optimization. At the same time, computer vision researchers have developed powerful deep-learning techniques for various 3D tasks [1,2,3,4,5], including automatic generation of novel 3D objects. Applying these techniques to design tasks requires evaluating performance metrics at scale. Traditionally, performance evaluation relies on physical simulation, which is time-consuming and computationally expensive. Data-driven surrogate models provide more scalable alternatives. This paper develops a surrogate model for evaluating the aerodynamic drag of 3D vehicles, aiming toward the eventual performance-guided generation of vehicle designs.

A key challenge in developing a surrogate model is representing shapes in a computationally efficient way that also captures the structure needed to accurately estimate relevant performance metrics. In machine learning, commonly used 3D shape representation methods include voxels, point clouds, and meshes, each affording different advantages and disadvantages. For example, 3D convolutional neural networks (CNNs) are commonly applied to learn structured voxel data [6, 7], while graph neural networks (GNNs) [8, 1] and CNNs generalized to irregular spaces [9, 10, 11] can learn unstructured 3D meshes. In recent years, diffusion models have suc-

cessfully been leveraged for learning point clouds for 3D shape generation [12, 2, 3, 4, 5]. These direct 3D representations are computationally limited to low-resolution shapes, which in turn limits their applications to practical engineering problems.

In addition to direct 3D representations, abstract representations in terms of two-dimensional (2D) renderings have also been explored. Since technologies for recognizing and generating 2D data is older and more mature than that for learning 3D data, several studies employ 2D renderings or point coordinate matrices to represent 3D shapes [13, 14, 15]. Parametric representations are another option to simplify the representation of 3D shapes [16, 17, 18]. These simplified 2D and parametric representations, however, suffer from varying degrees of information loss, and cannot provide sufficient information to reconstruct the corresponding 3D shapes. Accordingly, we propose a new image-based representation of 3D shapes that augments traditional 2D renderings with surface normal and depth information.

We use our representation to train a surrogate model for vehicle drag coefficient prediction, which is a key performance metric that affects not only fuel efficiency but also vehicle aesthetics. As we show, this enables fast and accurate estimation of 3D drag from 2D input. Our contributions are summarized as follows.

1. Dataset: We construct and share a diverse dataset of $4,948$ high-quality car 3D meshes labeled with drag coefficients computed by computational fluid dynamics (CFD) simulations.
2. Representation: We propose to use an efficient 2D image representation of 3D shapes that annotates 2D renderings with depth and surface normal information.
3. Model: We develop a high-performing surrogate model using the proposed representation for car drag coefficient prediction. Leveraging our 2D image representation, we base this model on fusing different pre-trained neural networks using an attention mechanism.
4. Application: We demonstrate a case study where the proposed surrogate model guides a diffusion-based deep generative model for drag-optimized car body generation.

In total, these contributions are a step towards the automatic, performance-aware generation of vehicle body designs. The surrogate model for car drag coefficient prediction also offers an efficient alternative to expensive 3D fluid dynamic simulations. We also hope that our dataset will

facilitate the development of various deep-learning techniques for car body design, evaluation, and optimization.

The remainder of this paper is organized as follows. Section 2 provides a detailed review of the relevant literature. In Section 3, we describe our dataset, our novel representation of 3D shapes, and our surrogate model for drag coefficient prediction. Section 4 reports and discusses the effectiveness of the proposed representation and the performance of the surrogate model, and also summarizes the limitations of our approach.

2 LITERATURE REVIEW

The two research areas most relevant to our contributions are 3D object representation and data-driven prediction of drag coefficients.

2.1 3D Shape Representation and Learning

In machine learning, 3D shapes are commonly represented as voxels, point clouds, or meshes. Different representations are often matched with different learning algorithms since different algorithms are better suited to exploit the advantages of each representation. For example, similar to CNNs that employ 2D kernels to learn visual features from images, 3D CNNs utilize 3D kernels to capture geometric features from structured 3D spatial data in Euclidean spaces. They are a popular option to learn voxels [6, 7] and occupancy grids for 3D shape recognition [19] and generation [20]. Since point clouds and meshes are unstructured, prior studies have explored transforming them into regular voxel grids [6, 21] or other canonicalized formats [22]. However, the sparsity of most 3D data representations makes the computation of the naïve 3D convolutional learning challenging. Researchers have proposed a few approaches to mitigate this issue. For example, multiple-resolution 3D CNNs can learn multi-scale features from multi-level voxels [23], while OctNet [20] represents its volumetric output as an octree with improved resolutions in the later levels. The voting [24] or probing [25] schemes in neural networks have been developed to assign varying amounts of computational effort to different regions of sparse data inputs.

In contrast, a more diverse set of deep learning models have been developed to learn unstructured 3D representations in non-Euclidean spaces (e.g., meshes, manifolds, and point clouds). In-

spired by conventional CNNs, a group of researchers developed a variety of CNN variants to learn irregular representations, including localized spectral CNNs [9], anisotropic CNNs [10], spline-based CNNs [26], geodesic CNNs [11], and others. Beyond that, GNNs have been applied to learn both point clouds and meshes for 3D shape recognition [27] and generation [8, 1]. More recently, diffusion models are becoming an area of active research interest. They have been applied to generate 3D shapes represented by point clouds or similar representations [12, 2, 3, 4, 5]. Due to computational cost, the 3D point clouds or meshes generated by these models still present low resolutions, impairing their applications in engineering domains. Prior studies have also explored simple multi-layer perceptrons (MLPs) for mesh texture editing [28, 29].

Two-dimensional representations have also been explored to represent 3D shapes. A few studies look into representing 3D shapes using 2D images or renderings, which can be processed by standard image learning algorithms [13, 14]. Despite the improved computational efficiency, such methods often suffer from information loss. Alternatively, Achlioptas et al. [15] proposed a representation that uses the point coordinates of a point cloud as a matrix and trains a generative model with the 2D matrix representation. This approach, however, can only work with point clouds that have a fixed number of points. Another set of studies maps 3D shapes to 2D parameter domains, then trains GANs to generate samples in the 2D domains, and finally converts them to 3D meshes [30, 31, 32, 33]. Additionally, implicit representations have also been explored for machine learning tasks. Implicit representations take a latent embedding of a shape and point coordinates as input and assign a value to each point which indicates if this point is inside or outside the shape [34, 35]. These representations are often used for 3D shape generation [36, 37]. A group of other studies exploits parametric representations which seek to convey the control points or other prominent features of 3D shapes for machine learning tasks [16, 17, 18].

In summary, the recognition, evaluation, and generation of 3D shapes using machine learning rely on effective and accurate 3D geometric feature learning. Existing representations of 3D shapes are still greatly limited by their high computational costs, while alternative 2D, implicit, and parametric representations suffer from information loss and may not capture sufficient geometric features for downstream tasks. In this paper, we show that a new representation of 3D shapes

using stacked depth and normal renderings is a promising approach, which helps significantly in the downstream task of predicting the drag coefficients of 3D cars.

2.2 Data-Driven Drag Coefficient Evaluation

Performance evaluation of 3D shapes is critical in engineering design and optimization. Among them, drag coefficient prediction is critical for car body design. It is traditionally conducted through simulations by solving the nonlinear Navier-Stokes equations for many iterations, which are time-consuming and computationally expensive. The solution methods are too slow to run in conjunction with a generative design or optimization process, which needs to evaluate a large number of candidate designs. To mitigate this issue, researchers have explored combining differentiable partial differential equations (PDEs) solvers with deep learning models to accelerate the simulation results without sacrificing the simulation accuracy significantly [38]. These differentiable PDE solvers often simulate the problem at a coarse resolution and the neural networks are employed to infer the results at higher resolutions. Such an approach speeds up the simulation process but is still too slow to be implemented during the deep generative process. As an alternative to differentiable PDE solvers, data-driven surrogate modeling is a desirable alternative to the simulation approaches in deep learning, and previous work has explored surrogate models for drag coefficient evaluation.

Parametric representation is commonly used in surrogate modeling of vehicle drag. For instance, Gunpinar et al. [16] represent a car using the coordinates of a set of control points from the 2D car silhouette and trained computational models to predict its drag coefficient in 2D settings. Their model first reduces the dimension of the representation using principal component analysis and then employs regression models or neural networks to learn the low-dimensional representation for drag coefficient prediction. Likewise, Rosset et al. [39] predicted the pressure field along the car silhouette to optimize 2D car designs. Umetani and Bickel [17] employed a parameterization method to represent simplified cars as vectors that indicate the position of control points and projection heights of the surface points. Then, they learned the representation using regression models, neural networks, or the Gaussian process for drag coefficient prediction. These studies reported that the regression or the Gaussian process models achieved higher explanatory power

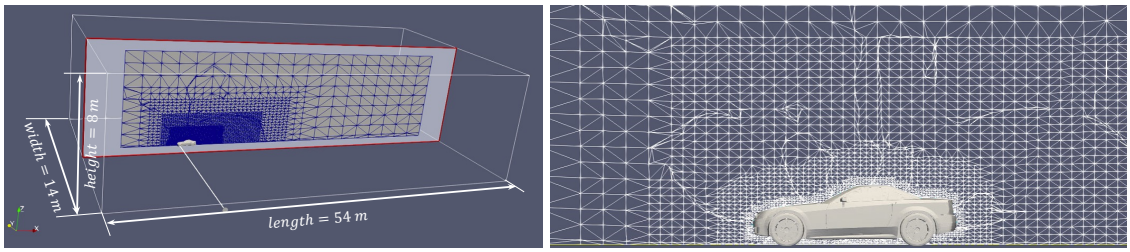
than the neural network models. Badias et al. [18] used locally linear embeddings to parameterize 3D cars and employed dimensionality reduction and interpolation to predict the drag coefficient of a new car. Limited by their parametric representations, these studies attempt to predict the drag coefficients of simplified cars, such as 2D car silhouettes or 3D cars with mirrors, wheels, and other details removed. This simplification may hinder the applications of such models in practical design contexts.

Another set of surrogate models learns 2D or 3D car representations to predict drag coefficients. For example, MeshSDF [40] learns 3D point clouds obtained from an implicit representation using an irregular CNN (i.e., spline-based CNNs [26]), which applies to drag coefficient prediction. Similarly, Baque et al. [41] exploited a geodesic CNN [11] to obtain a latent representation of 3D car meshes for drag coefficient prediction. Another model learns 2D slices of 3D point clouds using regular CNNs [42], while DEBOSH [43] learns meshes using GNNs for the same purpose. Additionally, another class of models obtains the latent representations of 2D [44] or 3D shapes [45] through reconstruction using generative models like variational autoencoders (VAEs) to predict the pressure fields and drag coefficients. Other surrogate models focus on drag prediction of general 3D shapes beyond cars [46, 47, 48]. Due to high computational costs, such models can only work with low-resolution 3D representations or simplified 2D representations. This paper focuses on surrogate modeling using the proposed representation of 3D shapes to circumvent the issues of the reviewed approaches.

3 DATA AND METHOD

In this section, we detail our main contributions in this paper: A high-quality dataset of 3D car meshes and their drag coefficients computed through CFD simulations, a 2D representation generated from 3D car meshes tailored to capturing features important for predicting drag coefficients, and a series of surrogate models trained to predict drag coefficients from the 2D representation as a regression task [1]

[1]The dataset and the surrogate models introduced in this paper can be found: https://decode.mit.edu/projects/dragprediction/.



(A) Computational Domain (B) Refinement regions

Fig. 1. The computational domain for the CFD simulation. (A) The entire simulation domain with larger space behind the car to simulate the wake flow; (B) The meshes closer to the car surface are finer than those in other regions.

3.1 Car Data and CFD Simulation

First, we detail our 3D car dataset and CFD simulations for obtaining drag coefficients from 3D mesh data.

3.1.1 Car Data

The 3D car meshes used in this paper are initially from the ShapeNet V1 dataset [49], which contains $7,497$ 3D car meshes with varying surface qualities. A substantial percentage of the original car meshes from ShapeNet are not watertight, with unsealed areas or holes on the surfaces. We need high-surface-quality car meshes in order to achieve reliable CFD simulation results when computing car drag coefficients. Therefore, we manually checked the surface quality of each car mesh from ShapeNet and selected a subset of $2,474$ high-quality car meshes. Since most of the selected meshes are still imperfect, we further repaired them using the repair module in Autodesk Netfabb Premium. It should be noted that this dataset covers a variety of car configurations, such as pick-up trucks, sedans, sport utility vehicles, wagons, and combat vehicles. The diversity helps our learned surrogate models generalize across all cars.

In addition, we employed two different approaches to augment the original dataset. First, we resized the width of each car using a random coefficient between 0.83 (i.e., $1/1.2$) to 1.2. The resizing augmentation created another $2,474$ cars with slightly different widths and drag coefficients from the original cars, resulting in a dataset of $4,948$ different cars in total. Second, since the car meshes are not perfectly bilaterally symmetric but their drag coefficients are invariant to

bilateral flipping [2], we employed a flipping augmentation to create another $4,948$ cars, which have the same drag coefficients as the cars without this augmentation. After the augmentations, we obtain a dataset of $9,896$ cars. To avoid data leakage, we only treat the $2,474$ unique cars from the original dataset as independent samples when splitting the dataset to train the surrogate model. For every car in any of the training, validation, or test sets, all of its resized and flipped versions belong to the same set.

3.2 CFD Simulation

The drag coefficient of each car is computed by a CFD simulation using OpenFOAM. During mesh preparation, all cars are normalized to have the same length of 3.5 meters to ensure the defined computational domain is suitable for all cars. The computational domain for simulation is then created, serving as a virtual wind tunnel to simulate the airflow around a car, as shown in Figure 1-A. The height, width, and length of the virtual tunnel are 8 meters, 14 meters, and 54 meters, respectively. In order to simulate flow dynamics around the car body more accurately, the computational domain is refined to a smaller mesh size, which becomes coarse away from the car surface, as shown in Figure 1-B. This meshing strategy is applied to all car configurations (e.g., sedans, sports utility vehicles, combat cars, and pick-up trucks) in our dataset. We employed the Realizable K-Omega Shear Stress Transport turbulence model for the CFD simulation. This choice was guided by its suitability for capturing aerodynamics around cars. For reference measurements, we used the car's wheelbase length to define the reference length and the frontal area of the car as the reference area.

On this basis, the inlet velocity and turbulence parameters are set as the inlet conditions, while outlet pressure is specified as the outlet condition. The car surface and road are set to be stationary walls. The sides and top of the computational domain are specified as symmetry boundaries. The steady-state "SimpleFoam" solver and the fluid flow "PotentialFoam" solvers are selected for the simulation. The primary boundary conditions and solver settings are listed in Tables 1 and 2. During the simulation, 300 iterations were conducted for each car, which can achieve the required

[2]Bilateral flipping of an asymmetric car with respect to the simulation domain's symmetry plane results in mirrored pressure and velocity fields. This symmetry ensures that both the original and flipped configurations yield identical drag coefficients upon integration of these fields.

Table 1. Boundary conditions

Tunnel inlet	Velocity inlet, velocity = $40km/h$
Tunnel outlet	Pressure outlet
Tunnel sides	Symmetry
Tunnel top	Symmetry
Tunnel road	No slip wall, with prism layer
Car body	No slip wall

Table 2. Solver settings

Gradient scheme	Linear
Divergence scheme (momentum)	Linear upwind
Divergence scheme (turbulence)	Upwind
Laplacian scheme	Linear
Interpolation scheme	Linear
Pressure solver	GMAG
Velocity solver	Smooth solver
No of Non-orthogonal corrections	2

convergence level and accuracy for concept-level studies. The simulation parameters employed in our study align with those established in prior research, which demonstrated that the simulated drag coefficients were within a 1.1% to 7.0% margin of error when compared to experimental data [50]. Since the drag coefficient outputs may fluctuate during the simulation process, we use the average value from the last 50 iterations as the final output from the simulation.

3.3 2D Representation of 3D Shapes

In prior work, voxels, point clouds, and meshes are commonly used to represent 3D shapes. They each require different deep neural networks to learn and rely on intensive computational resources to capture fine-grained, high-resolution 3D features. For car body design, we only focus on the surface of the car and ignore any interior architecture. In this paper, we aim to propose a more information-efficient method to represent 3D shapes like car bodies, which supports learning

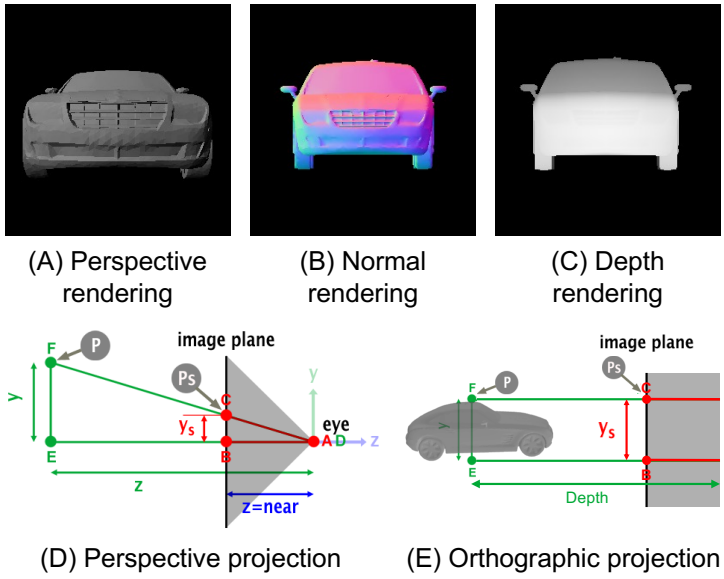


Fig. 2. The proposed 2D representation of 3D shapes. (A) The commonly used perspective rendering of a car; (B) and (C) The normal and depth renderings of the car; (D) Perspective projection used for generating perspective renderings; (E) Orthographic projection used for generating the normal and depth renderings.

3D information more effectively and affordably for drag coefficient prediction.

Since machine learning methods for 2D data learning are more explored than those for 3D data learning, 2D renderings have become an option to represent 3D shapes in many studies. However, the commonly used perspective 2D renderings (Figure 2-A) are generated through perspective projection (Figure 2-D), which causes geometric distortion and information loss for machine learning. Accordingly, we propose a new 2D representation of 3D shapes that consists of two types of renderings, namely the normal rendering (Figure 2-B) and the depth rendering (Figure 2-C), generated through orthographic projection (Figure 2-E). The points facing the cameras are first projected to the image space through a projection defined by Eq. 1. Herein, P_{camera} and P_{world} represent point coordinates (i.e., x, y) in the rendering and real-world space, respectively. $Scale_x$ and $Scale_y$ denote the scaling factors that are determined by the position and angle of the camera and the size of the rendering. Specifically, the pixel values of the normal rendering encode the unit normal vector at each point of the mesh, with the x ($Norm_x$), y ($Norm_y$), and z ($Norm_z$) coordinates mapped to the red ($Color_R$), green ($Color_G$), and blue ($Color_B$) color channels, respectively, as shown by Eq. 2. The pixel values of the depth rendering encode the depth of each

point, i.e., the distance ($Dist$) between the camera and the point, as formulated by Eq. 3.

$$P_{camera} = P_{world} \times \begin{bmatrix} Scale_x & 0 \\ 0 & Scale_y \end{bmatrix}, \tag{1}$$

$$Color_R = Norm_x,\ Color_G = Norm_y, Color_B = Norm_z, \tag{2}$$

$$Color_R = Color_G = Color_B = Dist. \tag{3}$$

According to the definition, the depth and normal renderings capture the point-wise positional and surface information of 3D shapes respectively. In order to capture the geometric features of a car comprehensively, we generate the normal and depth renderings from six orthographic views: front, rear, top, bottom, left, and right. Then, the six single-view renderings are integrated into a single image. With the combined information from all six single-view renderings, the integrated 2D representation conveys 3D geometric information and be potentially converted back to corresponding 3D shapes. Figure 3 describes the process using the depth rendering of a car. Building on the render module of the kaolin python package developed by NVIDIA[3], we develop a differentiable render for 3D to 2D rendering and a separate module for six view integration to produce the 2D representation for each car. The integrated normal and depth renderings are used as the 2D representation of 3D shapes in this paper. We verify the effectiveness of our proposed representation by developing surrogate models to predict car drag coefficients from our 2D representation.

The proposed 2D representation affords greater versatility by encompassing a broader spectrum of vehicle designs (e.g., roadsters and pick-up trucks) that extend beyond the conventional types (i.e., fastback, notchback, and estate back variants) covered by parametric presentations such as DrivAer [51]. The representation not only allows us to circumvent the challenges associated with manually defining and measuring the complex metrics to parameterize cars from comprehensive 3D datasets like ShapeNet, but also offers a task-agnostic car representation that

[3]https://github.com/NVIDIAGameWorks/kaolin.

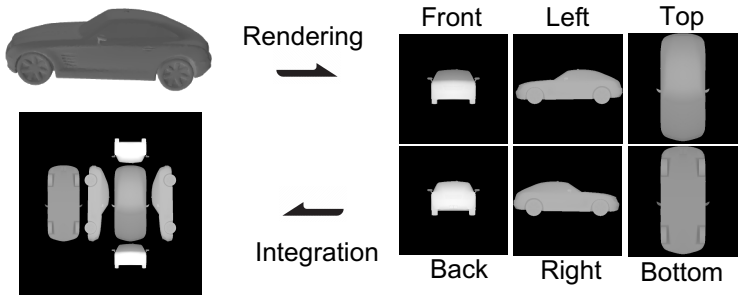


Fig. 3. The conversion process from a 3D mesh to 2D renderings, and to an integrated representation

obviates variable selection and customization for different tasks. This enables easy generalization of the representation to a wider array of car categories, facilitating the prediction of drag coefficients across a more extensive range. Moreover, the representation method also offers higher flexibility to incorporate the variety of nuanced geometric features that influence car aerodynamic performance—features such as mirrors, roof racks, and fine aerodynamic contours on car exteriors, which are not readily encapsulated by simpler parametric representations.

3.4 Surrogate Modeling

Our proposed 2D representation enables us to represent 3D shapes using 2D pixel data. We next develop and compare three surrogate models that take the 2D representation of a car as input and predict its drag coefficient. In this paper, the 2D representations of all cars in our dataset are images with three color channels and a dimension of 384×384, which is the input to all of our surrogate models.

We explore both the CNN-based and transformer-based computer vision models to learn features from the 2D representation of cars. In a set of pilot experiments, we first compare a few different pre-trained CNN-based models, including InceptionV3 [52], ResNet [53], and ResNeXt [54]. In general, they perform similarly after careful hyper-parameter tuning, and ResNeXt is selected in our study because it performs slightly better than the others. The proposed representation integrates six single-view renderings, which exhibit correspondence and convey complementary information for drag coefficient prediction. This characteristic of the representation motivates us to involve attention mechanisms in the surrogate model. Furthermore, since transformer-based

image models can capture the interactions between different image regions through the embedded self-attention mechanism, we also compare the CNN-based models against one transformer-based model, the vision transformer (ViT) [55].

First surrogate model architecture: The first model (Figure 4-A) employs the pre-trained ResNeXt "$101-32\times 8d$" module to embed the image input. The output from the ResNeXt embedding module exhibits a dimension of $12 \times 12 \times 2,048$, which is flattened. Following that, a linear layer with 128 neurons is attached before the output layer. We name this model "ResNeXt" in this paper.

Second surrogate model architecture: The second model (Figure 4-B) applies a self-attention mechanism to enhance the learning of the interactions between different image regions. Specifically, it reshapes the output from the ResNeXt embedding module to $144 \times 2,048$, which is seen as a set of 144 latent features with a dimension of $2,048$. A self-attention mechanism with a latent dimension of 128 is applied to capture the interactions between the image regions. Then, the output from the self-attention mechanism is flattened and projected to a lower dimension (128) through a linear layer as the final embedding to predict the car drag coefficient. This model is referred to as "attn-ResNeXt" hereafter.

Third surrogate model architecture: The third model (Figure 4-C) utilizes a pre-trained ViT module to embed the image input. We compare two different-sized ViT models, including the "$vit-large-patch32-384$" model and the "$vit-base-patch16-224$" model, and achieve slightly better performance from the former. Accordingly, "$vit-large-patch32-384$" was selected for building the third surrogate model. The pooled output from the transformer embedding module is used as the final embedding to predict the car drag coefficient. We call this model "ViT" in this paper.

Fused surrogate model architecture: Since the surrogate models introduced above can learn from only one of the normal/depth renderings at a time, we further explore if fusing the features of the normal and depth renderings can improve the prediction performance. After fine-tuning the hyperparameters of all three surrogate models, we select the best among the three for this

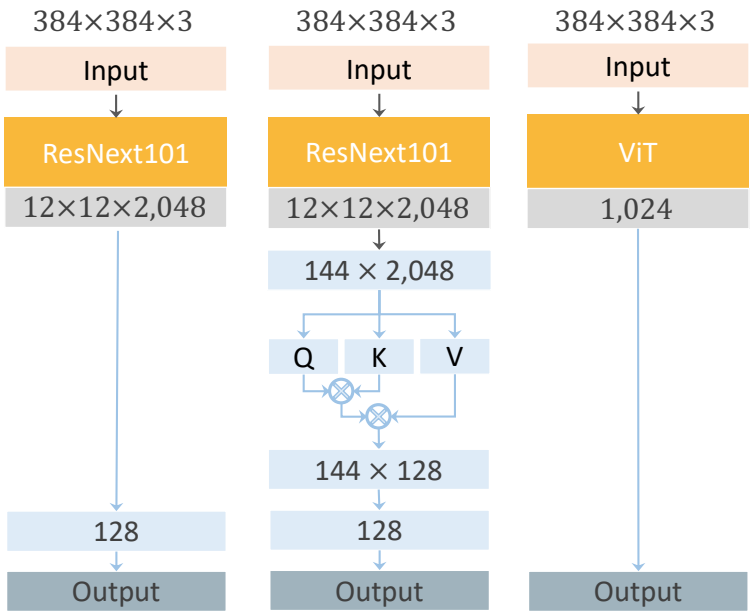


Fig. 4. The architectures of the three surrogate models using different embedding modules or different attention mechanisms

exploration, which is the attn-ResNeXt model in this study. Specifically, we fuse two attn-ResNeXt models respectively pre-trained on the normal and depth renderings using a symmetric cross-attention mechanism, as shown in Figure 5. The cross-attention mechanism is expected to capture the interactions between the regions respectively from the normal and depth renderings. Then, the outputs from the self-attention and cross-attention mechanisms are flattened and projected to a lower dimension (128) through linear layers, which are then concatenated as the final embedding to predict the car drag coefficient. During training, the fused model is initialized with the pre-trained weights from both the normal rendering model and the depth rendering model to transfer the knowledge learned from the single types of renderings to the fused model. This approach has been proven beneficial for avoiding modality failure [56, 57]. We refer to this model as "fused" hereafter.

The hyperparameters of these surrogate models are determined through a set of pilot experiments. In the experiments, all the trainable parameters are unfrozen. The pre-trained ResNeXt and ViT image embedding modules are fine-tuned on our data. We split the entire dataset into the training, validation, and test sets following a ratio of $0.7 : 0.15 : 0.15$. All models are trained on the same training-validation-test split for easy comparisons. We employ different learning rates

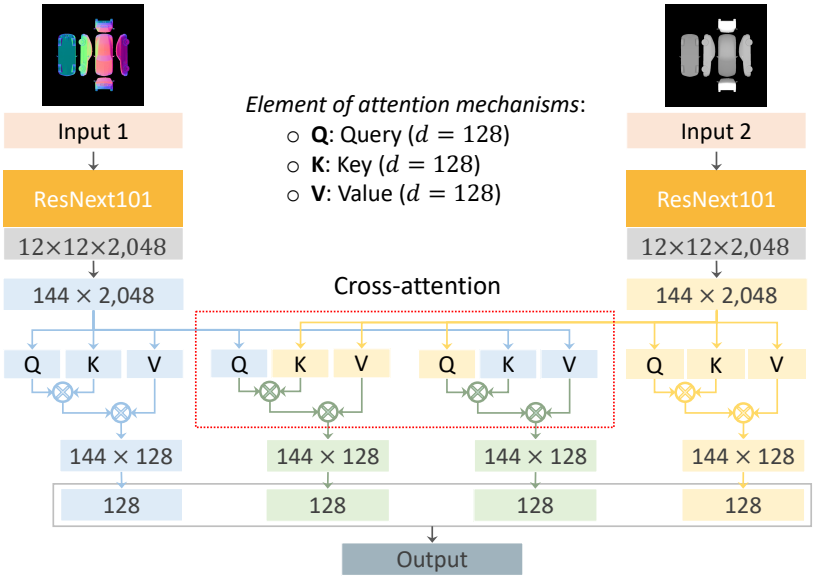


Fig. 5. The surrogate model fusing features of both the normal and depth renderings using a symmetric cross-attention mechanism

ranging from 2×10^{-5} to 8×10^{-5} to train different models with different image inputs. We also apply a decay of 0.96 to schedule the learning rate during the training process. We end the training process if the validation loss does not decrease for 20 consecutive epochs.

4 RESULTS AND DISCUSSION

This section describes our CFD simulation results and compares the performances of different surrogate models based on the proposed 2D representation. To evaluate the models, we report the coefficient of determination (R^2 value) and the mean squared prediction error (MSE). To illustrate sensitivity to initialization, we train each model five times and report the average values of these metrics. We also compare our best surrogate model against two baseline models from prior studies.

4.1 CFD Simulation Results

As described in the last section, our dataset originates from $4,948$ car meshes obtained from ShapeNet. Drag coefficients were successfully simulated for $4,535$ of these meshes using OpenFOAM. To increase the size of the dataset, we flip each car left to right (which leaves the drag coefficient unchanged), giving a total of $4,535 \times 2 = 9,070$ training examples.

The computed drag coefficients range from 0.175 to 0.907. Figure 6 shows their distribution and three sample vehicle images from different drag coefficient regimes. The data is concentrated on the interval $[0.28, 0.65]$.

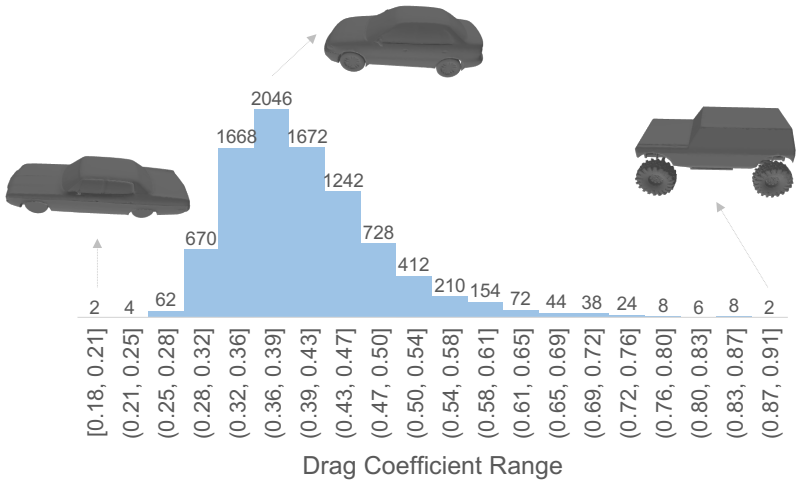


Fig. 6. The distribution of the drag coefficients and three example cars from the lowest, biggest, and highest drag coefficient categories, respectively

4.2 Performance of Surrogate Models

We compare unimodal and fused models using their prediction performance (R^2 value and MSE) on unseen test data.

Performance comparison for different model architectures: We first compare the drag coefficient prediction of six different surrogate models. Each model employs one of the three architectures depicted in Figure 4 and is trained on either depth or surface normal renderings. Figure 7 illustrates the performance of each model. Among the three architectures, the attn-ResNeXt model achieves the highest R^2 values and the lowest MSE values. Normal renderings perform better than depth renderings. The comparison between ResNeXt and attn-ResNeXt suggests that the self-attention mechanism improves the fusion of information from different image regions. Both ResNeXt and attn-ResNeXt outperform the ViT model. A possible reason is that ResNeXt contains far fewer trainable parameters than the ViT model (about 86 million vs about 2 billion), leading

to less overfitting.

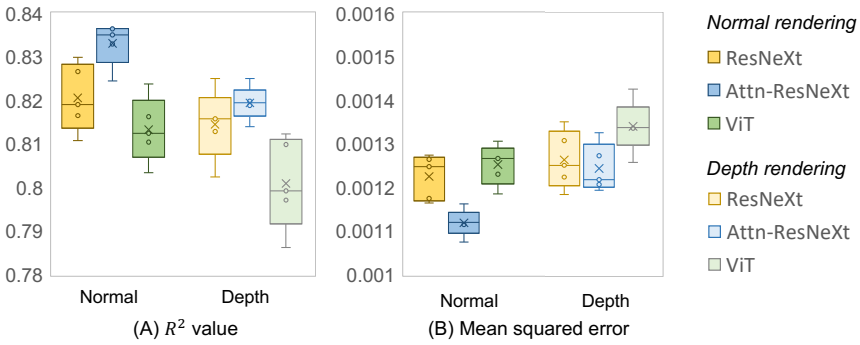


Fig. 7. The performance comparison among the three surrogate models using different rendering inputs. (A) R^2 value; (B) Mean squared error. We observe that Attention enhanced models outperform other models and normal renderings also have a higher predictive power than depth.

Performance of the fused model: We next illustrate that combining the normal and depth information enhances the performance of the surrogate model. We fuse these features using a symmetric cross-attention mechanism as depicted in Figure 5. Moreover, we train this fused model using the *transfer learning* paradigm; that is, we initialize the training of the fused model using the weights of the attn-ResNeXt models respectively pre-trained on the normal and depth renderings. Figure 8 illustrates the superior performance of the fused model, and significantly reduced sensitivity to initialization of the training procedure, as indicated by the reduced variance of the R^2 values and MSE values.

Performance in data-sparse regions: Figure 9 illustrates how the accuracy of the model depends on the ground-truth drag coefficient. As indicated, the prediction exhibits increasing deviations in the lowest and highest drag coefficient ranges. One major reason is that we have much fewer car samples with very low or high drag coefficients in our dataset (Figure 6). Accordingly, the model exhibits higher average prediction errors in the lowest and highest drag coefficient ranges, as listed in Table 3.

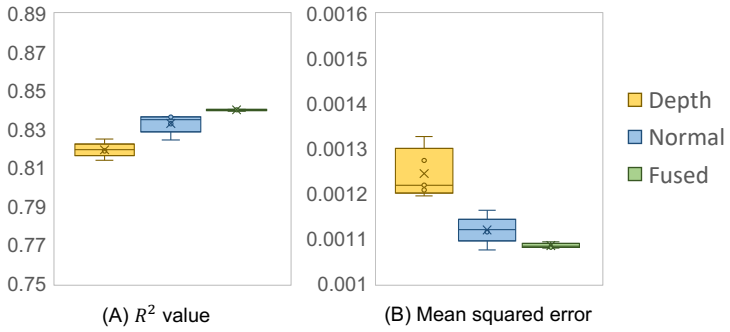


Fig. 8. The performance comparison among the two attn-ResNeXt models respectively using the normal and depth renderings and the fused model using both renderings. (A) R^2 value; (B) Mean squared error. We observed that the proposed fused model outperforms depth and normal models, while also reducing the variance in prediction performance.

Efficiency comparison: Evaluation of the surrogate models is also significantly faster than drag coefficient computation via CFD simulation. Indeed, it takes in total 20 seconds to evaluate the drag coefficients for $1,362$ cars using an NVIDIA RTX A5000 GPU. In comparison, the CFD simulation of a single car takes about 6 minutes on average using a Lambda computer with 12 Intel Xeon(R) E5-1650 CPUs. Finally, the surrogate models are also auto-differentiable and hence more easily incorporated into optimization routines.

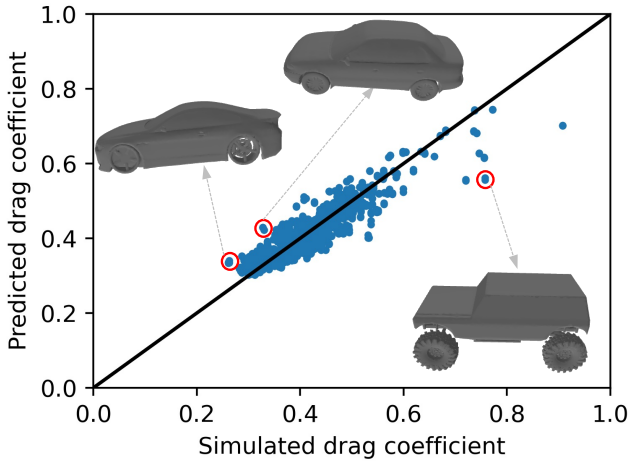


Fig. 9. The comparison between predicted and ground-truth values.

Table 3. The variation of the prediction error with the simulated drag coefficient. (A) R^2 value; (B) Mean squared error. We observe that prediction error increases at the extremes, where there are fewer car models in the dataset.

Drag Coefficient Range	Average Prediction Error
$[0.18, 0.3]$	0.032
$(0.3, 0.4]$	0.021
$(0.4, 0.5]$	0.023
$(0.5, 0.6]$	0.029
$(0.6, 0.7]$	0.021
$(0.7, 0.8]$	0.092
$(0.8, 0.91]$	0.218

4.3 Effectiveness of the Proposed Representation

In this subsection, we verify the effectiveness of the proposed representation by comparing its informativeness with single-view renderings and the perspective renderings. Beyond that, we also compare the performance of our surrogate model with the baseline models from two prior studies. The best surrogate model identified in the last subsection, attn-ResNeXt, is used for the following experiments.

Comparison of single-view renderings with integrated rendering: Compared to the single-view renderings, the integrated rendering is more informative for car drag coefficient evaluation. In this set of experiments, the attn-ResNeXt model takes the single-view normal renderings and the integrated normal renderings as input, respectively. Figure 10 depicts their R^2 and MSE values. The model taking the integrated renderings as input exhibits the highest R^2 value and lowest MSE value compared to all other models taking the single views as input. It is intuitive that the integrated renderings contain the geometric information of a car more comprehensively than any single-view rendering.

Among all single-view renderings, the front, back, left, and right views provide similar amounts of information for drag coefficient evaluation, leading to similar R^2 and MSE values. The bottom view is least informative for this task. In car body design, the streamlined design and the frontal area of a car affect the car's drag coefficient significantly. Every single view contains part of the

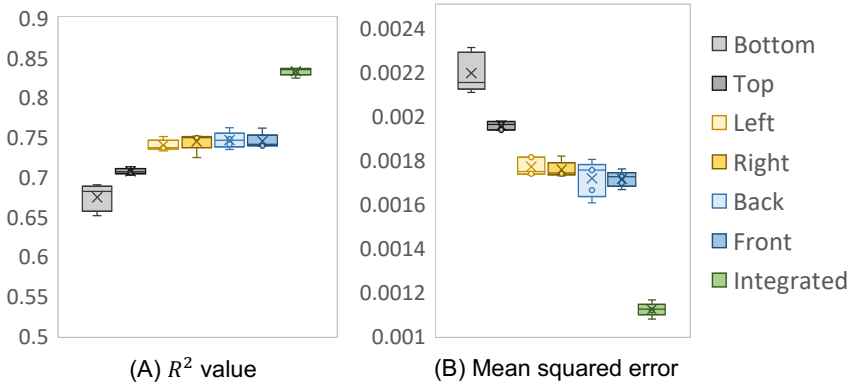


Fig. 10. The performance of the surrogate models using the single-view normal renderings and the integrated normal renderings, respectively. (A) R^2 value; (B) Mean squared error. We observe that the integrated view has the most information, leading to the highest performance. As expected, bottom and top views have the least predictive performance.

information. For example, the front and back views reflect the frontal area and the front or rear part of the streamlined design, while the left and right views show the entire streamlined design from two directions. The top view describes the top half of the streamlined design, which is often more informative than the bottom half depicted by the bottom view. The amount of relevant information conveyed by each view greatly determines the explanatory power of the corresponding model.

The proposed representation is also more informative than the perspective renderings as input for car drag coefficient evaluation. In this set of experiments, the attn-ResNeXt model takes the 2D perspective renderings and the proposed normal and depth renderings as input, respectively. Figure 11 depicts their R^2 and MSE values. The models using the normal renderings and depth renderings achieve significantly higher R^2 values and lower MSE values than the one using the 2D perspective renderings. That is, the normal and depth information conveyed by the proposed representation enables the model to capture more informative features for drag coefficient prediction.

Comparison of normal rendering and depth rendering: Additionally, the normal renderings are more informative than the depth renderings for this task when used separately. Two possible reasons can explain this. First, the normal renderings reflect the surface features directly, while the depth renderings provide the positional information from which the surface features can be inferred in a less straightforward way. Since the aerodynamic performance of a car is determined

by its surface features, this difference probably makes the normal renderings more informative for drag coefficient prediction. Second, the three color channels of the normal renderings store different information regarding the normal vectors along the x, y, and z coordinates, respectively. In comparison, the three color channels of the depth rendering store the same information regarding the distance between the camera and a certain point. The richness of the color channels may also allow the surrogate models to capture more information from the normal renderings.

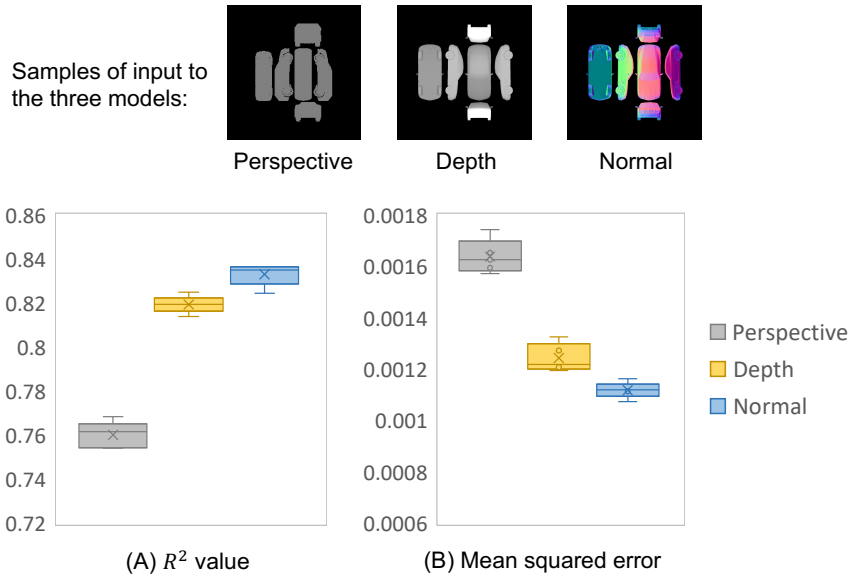


Fig. 11. The performances of the surrogate models using the proposed representation and the commonly used perspective renderings, respectively.

Comparison with prior work: Then, we compare our surrogate model with the baseline models from two prior studies. The first study [16] ran 2D CFD simulations with car silhouettes. The second study [17] ran 3D CFD simulations with simplified car designs with certain detailed features (e.g., wheels and mirrors) removed. Moreover, each car in their dataset was only simulated for 10 seconds, which might not return converged and reliable simulation results. That is, their simulations are rough compared to ours. Both baseline models employed parametric car representations to predict the simulated drag coefficients. As shown in Table 4, this study has advantages over the two baseline studies from three perspectives. First, unlike the other two studies targeting at

simplified low-fidelity car designs, this study aims to predict the drag coefficients of high-fidelity car designs. The fidelity of the cars in our dataset makes the associated surrogate model more applicable to the practical car design process. Second, since our dataset covers different types of cars from a wider range of drag coefficients, our surrogate model trained on it is more likely to be generalized to different car categories. Third, our model achieves a lower MSE compared to the first model and a comparable average prediction error with the second model. Since our dataset covers a wider drag coefficient range, the MSE and average error of our model could be lower when it is tested within their drag coefficient ranges, as shown in Figure 9.

Surrogate models trained on datasets of cars created from parametric definitions tend to outperform the models trained on comprehensive datasets like ShapeNet. For instance, the model developed by Jacob et al. [58] presents an R^2 value above 0.9 within the drag coefficient range [0.27, 0.34]. It is important to note that while our surrogate model incorporates high-dimensional 2D renderings, which may seem excessive compared to compact parametric representations, our representation captures a wide array of nuanced geometric features not represented in compact parameterizations. This breadth contributes to the variability in car drag coefficient and is a deliberate choice to enhance the model's generalizability across diverse car designs. We assert that the performance of our surrogate model is competitive when considering its ability to operate over a broad drag coefficient range and represent a variety of vehicle geometries. While a more focused approach on a narrower range of car types with less variability in features could indeed yield higher R-squared values, our objective was to develop a model with broader applicability. Compared to other surrogate models that learn car meshes using graph neural networks or pointNet to predict aerodynamic performance and present higher performance [59, 60], our approach retains the detailed features from the original car models in the ShapeNet dataset instead of omitting components like wheels and mirrors, providing a more comprehensive and high-fidelity representation and more practical predictions.

Discussion: The above comparisons verify the effectiveness of the proposed representation. The proposed representation integrating six single-view renderings contains more comprehensive geometric information than any single-view rendering. Moreover, the proposed representation is

Table 4. The comparison between the best model from this study and two prior studies

	Ours	**Study 1** [16]	**Study 2** [17]
Input to CFD Simulation	3D meshes	2D silhouettes	3D meshes
Input to surrogate models	2D renderings	Parametric	Parametric
Fidelity	Original	Simplified	Simplified
Drag Coefficient Range	$0.17 - 0.85$	$0.21 - 0.51$	$0.2 - 0.6$
Mean Squared Error	8.2×10^{-4}	1.84×10^{-3}	
Average Error	0.024		$0.013 - 0.021$

more informative than the 2D perspective renderings for two reasons. First, the proposed normal and depth renderings convey the geometric information regarding the surface normal and positional features of each point of a 3D shape. Second, the orthographic projection used to generate the proposed representation avoids geometric distortion compared to the perspective projection. These advantages of the proposed representation allow us to reconstruct 3D shapes from them without any learning process, while it is challenging to accurately reconstruct 3D shapes from the 2D perspective renderings without a learning process. Moreover, the proposed representation method is generalizable to broader 3D shape categories whose major geometric information can be captured from the six orthographic views, such as airplanes, ships, bottles, chairs, and so forth.

4.4 Potential Application of the Surrogate Model in Enhancing Deep Generative Models

Providing accurate and fast car drag coefficient evaluation, the surrogate model can be integrated into gradient-based deep generative models of 3D cars. The presence of such a performance evaluation model can make the generation process performance-aware and lead to high-performing 3D car designs. This is a promising direction for future research. To demonstrate the potential applications and validate the effectiveness of the surrogate model, we integrated the pre-trained surrogate model as a guidance module with a pre-trained diffusion model to optimize the drag coefficient during synthesizing new cars, as shown in Figure 12-A. The diffusion model consists of a diffusion process during which Gaussian noise is added to the input image (x_0) until it becomes pure noise (x_T) and a denoising process during which the model removes Gaussian

noise step-by-step to reconstruct the input image ($\widetilde{x}$) [61, 62, 63]. The Gaussian noise parameters are learned by neural networks. Once trained, the denoising module can serve as a generator that takes noise as input to generate new samples. The surrogate model is incorporated as a guidance module following the regressor guidance technique proposed in prior studies [62, 63]. Specifically, the surrogate model ($f(x)$) takes the original noisy image ($\widetilde{x}_t$) sampled from a certain denoising step (t) as input to predict the drag coefficient. Based on the pre-trained weights, the gradients ($\nabla_{\widetilde{x}} f$) of the drag coefficient ($f(\widetilde{x}_t)$) with respect to the noisy image ($\widetilde{x}_t$) can be calculated. Then, the guided diffusion model shifts the mean of the noisy image from μ to $\mu - w \times \nabla_{\widetilde{x}} f \times \Sigma$ to sample the guided image output ($\widetilde{x}_t^C$). Herein, w is the gradient scale factor, while μ and Σ are the mean and variance of the noisy images in this step, respectively. The guided diffusion model was adapted from TopoDiff [62], which readers can refer to for more technical details. This guidance provides a strategy to minimize the drag coefficient during new car design generation.

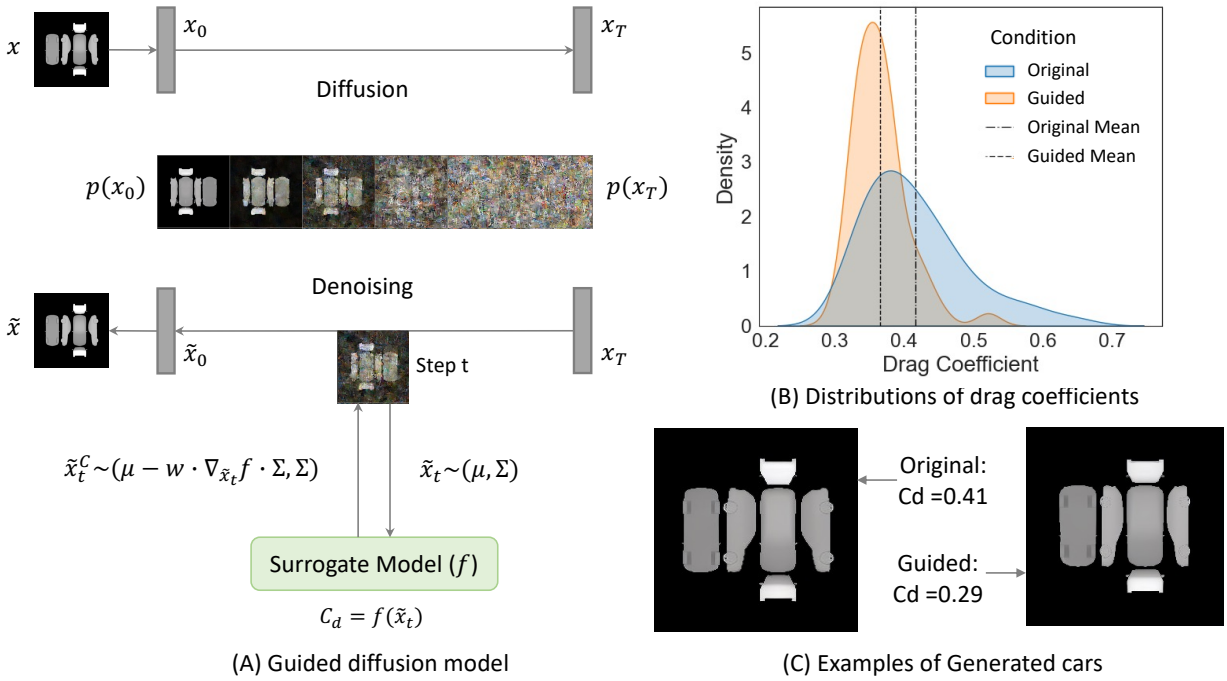


Fig. 12. Guiding a generative model using the surrogate model. (A) The guided diffusion model consists of a pre-trained diffusion model and a pre-trained surrogate model; (B) The drag coefficient distributions of the cars generated by the original and guided diffusion models, respectively; (C) The cars generated by the original and guided diffusion models, respectively.

Our experimental results indicate that the surrogate model, when used as a guidance module, enables the diffusion model to optimize the car drag coefficient during the synthesis of new car designs. To study the effectiveness of the guided diffusion model in optimizing the car drag coefficient, we employed it to generate 50 new cars using the same set of 50 noise inputs and evaluate their drag coefficients using the surrogate model. The drag coefficient distributions of the cars respectively generated by the two models are illustrated in Figure 12-B. The cars generated by the guided model show a significantly lower mean than the cars generated by the original model ($mean_{guided} = 0.37$, $mean_{original} = 0.42$, $p < 0.001$ for pairwise Student's t-test). Figure 12-C showcases two cars, each generated by the original and guided diffusion models respectively using the same noise input. The car at the top, synthesized by the original diffusion model, presents a drag coefficient of 0.37. The car on the bottom, generated by the diffusion model guided by the surrogate model, exhibits a superior drag coefficient of 0.28. Notably, the car design at the bottom exhibits a reduced height, lower ride height, and more streamlined contours. These results underscore the surrogate model's impact in empowering the generative diffusion model to incorporate aerodynamic features into new car designs. It is noteworthy that the surrogate model-guided optimization during the generation process may venture into the adversarial zones of the surrogate model, where its evaluation accuracy is low. Future research should explore the avenues of conformal prediction-based guidance to mitigate this issue.

Moreover, the proposed 2D representation has the potential to promote 3D shape generation, evaluation, and optimization using deep learning models. The 3D representations of shapes are either sparse or redundant in many cases. For example, only surface information is needed to represent a car body design. When it is represented as voxels, all the voxels inside the surface are redundant. The redundancy and sparsity of 3D representations make it highly computationally expensive to learn 3D shapes. With limited computational power, deep learning models struggle to handle high-resolution 3D shapes represented by voxels, meshes, or point clouds. Accordingly, these models do not allow for the generation, evaluation, and optimization of 3D shapes with plenty of geometric details, hindering their applications to real-world problems. Moreover, as AI technologies are more explored to handle 2D data as of now, the proposed 2D representation

enables us to handle 3D shapes with more powerful 2D AI technologies. It is much easier and less expensive to increase the resolution of the 2D representation of 3D shapes than doing that with the 3D representations directly. Therefore, the proposed representation is promising to enable 3D shape generation, evaluation, and optimization at a higher resolution with less computational power needed.

4.5 Limitations and Future Work

While the proposed 2D representation, dataset, and surrogate model are promising, they have limitations and leave room for further improvement.

First, as 2D renderings inherently lack the ability to portray features not visible on the object's surface, the proposed representation is insufficient to model more complex 3D shapes with rich internal structures, such as lattice designs and car interior designs. In this study, some car internal features that cannot be captured by the 2D representation (e.g., wheel well design) may impair the predictive accuracy of the surrogate model. Moreover, although the proposed representation is informative for machine learning, it is less intuitive for human perception compared to 3D representations, such as meshes and point clouds. In future work, we will attempt to refine the 2D representation to enable the incorporation of the internal structures.

Second, the 2D renderings cannot capture detailed information about local surface geometry and the interconnectivity of vertices of 3D shapes represented by meshes. Meanwhile, the resolution of the 2D renderings significantly influences the level of detail that can be captured from 3D shapes. Lower resolution can result in the omission of intricate details present in 3D shapes.

Third, the dataset introduced in this paper is far smaller than the training sets typically used for deep learning models. In particular, the number of samples with high drag coefficients is low. The small dataset may lead to over-fitting during the training process. Additionally, the CFD simulation results in this study have not been validated through experiment measurements. We aim to augment our dataset and verify its reliability by juxtaposing the simulation results with experimental data, and train and test the developed model with a sizable dataset.

Fourth, while we show that the integrated renderings are more informative than the single-view renderings, alternative integration techniques may be more effective. We will explore such

alternatives in future work.

Fifth, the approach proposed in this paper for drag coefficient prediction is a purely data-driven approach, which does not leverage any physics knowledge regarding CFD simulations. The performance of the surrogate model depends on the quality and quantity of the data, and is unlikely to perform well on inputs far from the training set. In future work, we will attempt to incorporate physics into our surrogate model by including the prediction of continuous fields such as pressure and velocity distributions over the car's surface, rather than limiting our output to a single-valued drag coefficient.

Sixth, we only considered the scenario where the airflow is parallel to the car's trajectory in this study. However, in real-world conditions where crosswinds are present, the relative wind angle changes and introduces an additional variable that influences aerodynamic drag. In recognition of this factor, we will consider rotating the car models to various angles to simulate different wind angles relative to the car's direction, which will also effectively augment our dataset.

Lastly, the surrogate model developed in this paper can only make predictions using the proposed 2D representations of cars and does not apply to common car images. A promising future direction is to associate the proposed 2D representations with real images so that the surrogate model can make predictions using easily accessible car images.

5 CONCLUSION

Drag coefficient evaluation is an indispensable element of the aerodynamic design of cars, which has a critical influence on car fuel efficiency. In this paper, we develop a surrogate model that enables accurate, fast, and differentiable drag coefficient evaluation. This surrogate model is built on a new 2D representation of 3D shapes. This representation embeds depth and surface normal information into 2D renderings and combines information from six orthographic views. The results of this study suggest that our proposed representation is more effective and informative than simple 2D perspective renderings for drag coefficient prediction. To train our model, we also assemble a diverse dataset of high-quality 3D car meshes and simulate their drag coefficients using CFD simulations. This dataset, upon public release, can drive the development of other

data-driven design approaches. In total, our contributions facilitate the data-driven design of 3D aerodynamic cars and can be readily combined with generative AI techniques to automate design creation.

ACKNOWLEDGEMENTS

This research was supported in part by the Toyota Research Institute. Additionally, we thank Mr. Hanqi Su for helping us select high-quality car meshes from ShapeNet.

REFERENCES

[1] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y. G., 2018. "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* ***11215 LNCS***, 4, pp. 55–71.

[2] Zhou, L., Du, Y., and Wu, J., 2021. "3D Shape Generation and Completion through Point-Voxel Diffusion". *Proceedings of the IEEE International Conference on Computer Vision*, 4, pp. 5806–5815.

[3] Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., and Kreis, K., 2022. "LION: Latent Point Diffusion Models for 3D Shape Generation".

[4] Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., and Chen, M., 2022. "Point-E: A System for Generating 3D Point Clouds from Complex Prompts".

[5] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M., 2021. "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models".

[6] Prokhorov, D., 2010. "A convolutional learning system for object classification in 3-D lidar data". *IEEE Transactions on Neural Networks,* ***21***(5), 5, pp. 858–863.

[7] Maturana, D., and Scherer, S., 2015. "VoxNet: A 3D Convolutional Neural Network for real-time object recognition". In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 922–928.

[8] Li, X., Xie, C., and Sha, Z., 2022. "A Predictive and Generative Design Approach for Three-Dimensional Mesh Shapes Using Target-Embedding Variational Autoencoder". *Journal of Mechanical Design,* ***144***(11), 11.

[9] Qi, C. R., Su, H., Niessner, M., Dai, A., Yan, M., and Guibas, L. J., 2016. "Volumetric and Multi-View CNNs for Object Classification on 3D Data". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* ***2016-Decem***, 4, pp. 5648–5656.

[10] Wang, C., Pelillo, M., and Siddiqi, K., 2019. "Dominant Set Clustering and Pooling for Multi-View 3D Object Recognition". *British Machine Vision Conference 2017, BMVC 2017*, 6.

[11] Masci, J., Boscaini, D., Bronstein, M. M., and Vandergheynst, P., 2015. "Geodesic convolutional neural networks on Riemannian manifolds". *Proceedings of the IEEE International Conference on Computer Vision,* ***2015-Febru***, 1, pp. 832–840.

[12] Luo, S., and Hu, W., 2021. "Diffusion Probabilistic Models for 3D Point Cloud Generation". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3, pp. 2836–2844.

[13] Ghadai, S., Lee, X. Y., Balu, A., Sarkar, S., and Krishnamurthy, A., 2021. "Multi-resolution 3D CNN for learning multi-scale spatial features in CAD models". *Computer Aided Geometric Design,* ***91***, 11, p. 102038.

[14] Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E., 2015. "Multi-view convolutional neural networks for 3D shape recognition". In Proceedings of the IEEE International Conference on Computer Vision, Vol. 2015 Inter, pp. 945–953.

[15] Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L., 2017. "Learning Representations and Generative Models for 3D Point Clouds". *35th International Conference on Machine Learning, ICML 2018,* ***1***, 7, pp. 67–85.

[16] Gunpinar, E., Coskun, U. C., Ozsipahi, M., and Gunpinar, S., 2019. "A Generative Design and Drag Coefficient Prediction System for Sedan Car Side Silhouettes based on Computational Fluid Dynamics". *Comput. Aided Des.,* ***111***, 6, pp. 65–79.

[17] Umetani, N., and Bickel, B., 2018. "Learning three-dimensional flow for interactive aerodynamic design". *ACM Transactions on Graphics (TOG),* ***37***(4), 7, p. 10.

[18] Badías, A., Curtit, S., González, D., Alfaro, I., Chinesta, F., and Cueto, E., 2019. "An augmented reality platform for interactive aerodynamic design and analysis". *International Journal for Numerical Methods in Engineering,* ***120***(1), 10, pp. 125–138.

[19] Garcia-Garcia, A., Gomez-Donoso, F., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M., and Azorin-Lopez, J., 2016. PointNet: A 3D Convolutional Neural Network for real-time object class recognition.

[20] Tatarchenko, M., Dosovitskiy, A., and Brox, T., 2017. "Octree Generating Networks: Efficient Convolutional Architectures for High-Resolution 3D Outputs". In Proceedings of the IEEE International Conference on Computer Vision (ICCV),, pp. 2088–2096.

[21] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J., 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes.

[22] Wang, C., Samari, B., and Siddiqi, K., 2018. "Local Spectral Graph Convolution for Point Set Feature Learning". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* ***11208 LNCS***, 3, pp. 56–71.

[23] Boscaini, D., Masci, J., Rodolà, E., and Bronstein, M., 2016. "Learning shape correspondence with anisotropic convolutional neural networks". *Advances in Neural Information Processing Systems,* ***29***.

[24] Wang, D. Z., and Posner, I., 2015. "Voting for voting in online point cloud object detection". *Robotics: Science and Systems,* ***11***.

[25] Li, Y., Pirk, S., Su, H., Qi, C. R., and Guibas, L. J., 2016. "FPNN: Field Probing Neural Networks for 3D Data". *Advances in Neural Information Processing Systems*, 5, pp. 307–315.

[26] Fey, M., Lenssen, J. E., Weichert, F., and Müller, H., 2017. "SplineCNN: Fast Geometric Deep Learning with Continuous B-Spline Kernels". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 11, pp. 869–877.

[27] Li, J., Chen, B. M., and Lee, G. H., 2018. "SO-Net: Self-Organizing Network for Point Cloud Analysis". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 12, pp. 9397–9406.

[28] Michel, O., Bar-On, R., Liu, R., Benaim, S., and Hanocka, R., 2021. "Text2Mesh: Text-Driven Neural Stylization for Meshes".

[29] Jetchev, N., 2021. "ClipMatrix: Text-controlled Creation of 3D Textured Meshes".

[30] Maron, H., Galun, M., Aigerman, N., Trope, M., Dym, N., Yumer, E., Kim, V. G., and Lipman, Y., 2017. "Convolutional neural networks on surfaces via seamless toric covers". *ACM Transactions on Graphics (TOG),* ***36***(4), 7.

[31] Ben-Hamu, H., Maron, H., Kezurer, I., Avineri, G., and Lipman, Y., 2018. "Multi-chart Generative Surface Modeling". *SIGGRAPH Asia 2018 Technical Papers, SIGGRAPH Asia 2018*, 6.

[32] Saquil, Y., Xu, Q. C., Yang, Y. L., and Hall, P., 2020. "Rank3DGAN: Semantic mesh generation using relative attributes". *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 5586–5594.

[33] Alhaija, H. A., Dirik, A., Knörig, A., Fidler, S., and Shugrina, M., 2022. "XDGAN: Multi-Modal 3D Shape Generation in 2D Space".

[34] Chen, Z., and Zhang, H., 2018. "Learning Implicit Fields for Generative Shape Modeling". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* ***2019-June***, 12, pp. 5932–5941.

[35] Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S., 2019. "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* ***2019-June***, 1, pp. 165–174.

[36] Alwala, K. V., Gupta, A., and Tulsiani, S., 2022. "Pre-train, Self-train, Distill: A simple recipe for Supersizing 3D Reconstruction". pp. 3763–3772.

[37] Liu, Z., Dai, P., Li, R., Qi, X., and Fu, C.-W., 2022. "ISS: Image as Stepping Stone for Text-Guided 3D Shape Generation".

[38] de Avila Belbute-Peres, F., Economon, T. D., and Kolter, J. Z., 2020. "Combining differentiable pde solvers and graph neural networks for fluid flow prediction". In Proceedings of the 37th International Conference on Machine Learning, ICML'20, JMLR.org.

[39] Rosset, N., Cordonnier, G., Duvigneau, R., Bousseau, A., and Rosset Guillaume Cordonnier Regis Duvigneau Adrien Bousseau, N., 2023. "Interactive design of 2D car profiles with aerodynamic feedback". *Computer Graphics Forum,* ***42***(2), 2, pp. 1–11.

[40] Remelli, E., Lukoianov, A., Richter, S. R., Guillard, B., Bagautdinov, T., Baque, P., and Fua, P., 2020. "MeshSDF: Differentiable Iso-Surface Extraction". *Advances in Neural Information Processing Systems,* ***2020-December***, 6.

[41] Baque, P., Remelli, E., Fleuret, F., and Fua, P., 2018. "Geodesic Convolutional Shape Optimization". *35th International Conference on Machine Learning, ICML 2018,* ***2***, 2, pp. 797–809.

[42] Jacob, S. J., Mrosek, M., Othmer, C., and Köstler, H., 2021. "Deep Learning for Real-Time Aerodynamic Evaluations of Arbitrary Vehicle Shapes". *SAE International Journal of Passenger Vehicle Systems,* ***15***(2), 8, pp. 77–90.

[43] Durasov, N., Lukoyanov, A., Donier, J., and Fua, P., 2021. "DEBOSH: Deep Bayesian Shape Optimization".

[44] Thuerey, N., Weissenow, K., Prantl, L., and Hu, X., 2018. "Deep Learning Methods for Reynolds-Averaged Navier-Stokes Simulations of Airfoil Flows". *AIAA Journal,* ***58***(1), 10, pp. 25–36.

[45] Saha, S., Rios, T., Minku, L. L., Stein, B. V., Wollstadt, P., Yao, X., Back, T., Sendhoff, B., and Menzel, S., 2021. "Exploiting Generative Models for Performance Predictions of 3D Car Designs". *2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings*.

[46] Xin, D., Zeng, J., and Xue, K., 2022. "Surrogate drag model of non-spherical fragments based on artificial neural networks". *Powder Technology,* ***404***, 5, p. 117412.

[47] TAO, J., SUN, G., GUO, L., and WANG, X., 2020. "Application of a PCA-DBN-based surrogate model to robust aerodynamic design optimization". *Chinese Journal of Aeronautics,* ***33***(6), 6, pp. 1573–1588.

[48] Sun, G., and Wang, S. "A review of the artificial neural network surrogate modeling in aerodynamic design".

[49] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F., 2015. "ShapeNet: An Information-Rich 3D Model Repository".

[50] Biswas, K., Gadekar, G., and Chalipat, S., 2019. "Development and Prediction of Vehicle Drag Coefficient Using OpenFoam CFD Tool". *SAE Technical Papers,* ***2019-Janua***(January), 1.

[51] Heft, A. I., Indinger, T., and Adams, N. A., 2012. "Introduction of a New Realistic Generic Car Model for Aerodynamic Investigations". *SAE Technical Papers*, 4.

[52] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., 2016. "Rethinking the Inception Architecture for Computer Vision". In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-Decem, pp. 2818–2826.

[53] He, K., Zhang, X., Ren, S., and Sun, J., 2016. "Deep residual learning for image recognition". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* ***2016-Decem***, 12, pp. 770–778.

[54] Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K., 2016. "Aggregated Residual Transformations for Deep Neural Networks". *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017,* ***2017-Janua***, 11, pp. 5987–5995.

[55] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N., 2020. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale".

[56] Du, C., Li, T., Liu, Y., Wen, Z., Hua, T., Wang, Y., and Zhao, H., 2021. "Improving multi-modal learning with uni-modal teachers".

[57] Song, B., Associate, P., Miller, S., and Ahmed, F., 2023. "Attention-Ehanced Multimodal Learning for Conceptual Design Evaluations". *Journal of Mechanical Design*, 1, pp. 1–38.

[58] Jacob, S. J., Mrosek, M., Othmer, C., and Köstler, H., 2022. "Deep Learning for Real-Time Aerodynamic Evaluations of Arbitrary

} Vehicle Shapes". *SAE International Journal of Passenger Vehicle Systems,* ***15***(2), 3, pp. 77–90.

[59] Cunningham, J. D., Simpson, T. W., and Tucker, C. S., 2019. "An Investigation of Surrogate

Models for Efficient Performance-Based Decoding of 3D Point Clouds". *Journal of Mechanical Design,* ***141***(12), 09, p. 121401.

[60] Abbas, A., Rafiee, A., Haase, M., and Malcolm, A., 2022. "Geometrical deep learning for performance prediction of high-speed craft". *Ocean Engineering,* ***258***, p. 111716.

[61] Ho, J., Jain, A., and Abbeel, P., 2020. "Denoising Diffusion Probabilistic Models". *Advances in Neural Information Processing Systems,* ***2020-Decem***, 6.

[62] Mazé, F., and Ahmed, F., 2023. "Diffusion Models Beat GANs on Topology Optimization". *Proceedings of the AAAI Conference on Artificial Intelligence,* ***37***(8), 6, pp. 9108–9116.

[63] Dhariwal, P., and Nichol, A., 2021. "Diffusion Models Beat GANs on Image Synthesis". *Advances in Neural Information Processing Systems,* ***11***, 5, pp. 8780–8794.