2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Large Language and Text-to-3D Models for Engineering Design Optimization

Thiago Rios
Honda Research Institute Europe
Offenbach am Main, Germany
thiago.rios@honda-ri.de

Stefan Menzel
Honda Research Institute Europe
Offenbach am Main, Germany
stefan.menzel@honda-ri.de

Bernhard Sendhoff
Honda Research Institute Europe
Offenbach am Main, Germany
bernhard.sendhoff@honda-ri.de

***Abstract*—The current advances in generative artificial intelligence for learning large neural network models with the capability to produce essays, images, music and even 3D assets from text prompts create opportunities for a manifold of disciplines. In the present paper, we study the potential of deep text-to-3D models in the engineering domain and focus on the chances and challenges when integrating and interacting with 3D assets in computational simulation-based design optimization. In contrast to traditional design optimization of 3D geometries that often searches for the optimum designs using numerical representations, *e.g.* B-Spline surfaces, natural language challenges the optimization framework by requiring a different interpretation of variation operators while at the same time may ease and motivate the human user interaction. Here, we propose and realize a fully automated evolutionary design optimization framework using Shap-E, a recently published text-to-3D asset network by OpenAI, in the context of aerodynamic vehicle optimization. For representing text prompts in the evolutionary optimization, we evaluate (a) a bag-of-words approach based on prompt templates and Wordnet samples, and (b) a tokenisation approach based on prompt templates and the byte pair encoding method from GPT4. In our experiments, we show the text-based representations allow the optimizer to find better performing designs. However, it is important to ensure that the designs generated from prompts are within the object class of application, *i.e.* diverse and novel designs need to be realistic. Furthermore, more research is required to develop methods where the strength of text prompt variations and the resulting variations of the 3D designs share causal relations to some degree to improve the optimization.**

***Index Terms*—Large language models, generative AI, text-to-3D, simulation-based optimization, design optimization**

I. Introduction

The recent advances in building foundation models [1], large language models (LLMs) [2], and text-to-image models [3] have a major impact on a variety of fields, such as natural language processing and understanding, text and image generation, and human machine interaction. The maturity and ease of use of these novel models even lead to the adaptation of business models in some domains *e.g.*, text writing, software development, and product design. Although the application of foundation models to engineering has been less discussed compared to other domains, we see great potential in how LLMs, text-to-image and text-to-3D models could be used in industrial engineering. Natural language interfaces between engineers and complex software systems in computational aided engineering (CAE) could improve their usage and make them more accessible for younger engineers or for non-experts in general. Furthermore, *text-to-X* approaches could improve the interaction between engineers and CAE systems by offering new ways for generating and modifying designs (images and 3D objects).

In general, 3D object representations are ubiquitous and rely on a number of parameters that are manipulated so that a certain object is realized. The choice for a particular method depends on the application in the product design process. Traditional object representations are spline curves and surfaces, which are parameterized by control points and knot points. Alternatively, free-form deformations represent modifications of complex objects more efficiently, especially when combined with finite element/volume simulations [4]. Computer aided design (CAD) systems combine boundary representation and constructive solid geometry to generate digital 3D objects, which are typically mapped to high resolution polygonal meshes for engineering simulations (*e.g.*, computational fluid dynamics). Recently, learning-based representations, such as autoencoders (AEs) and variational autoencoders (VAEs) [5], [6], have been utilized in simulation-based optimization problems where the traditional design parameters are exchanged by the learned latent variables.

In computational engineering optimization, text-to-3D generative models enable the exploration of designs through natural language, which is a unique design representation. However, although generating 3D designs from intuitive object descriptions seems promising, the quality of text-to-3D models in the context of design optimization is unknown and needs to be assessed. Besides the meaningfulness of designs for a corresponding input text prompt, the relation between variations in the prompts and resulting designs also affects the performance of a design optimization. Along these lines, text-based domains also create new challenges for handling design constraints and fostering novelty—here, we refer to "novelty" as the capability to generate designs which are different from the training data, yet realistic for a given application. For example, by including the word "car" in all input prompts, the generative model ideally yields car-like geometries with minor variations (Fig. 1). Yet, this condition is often not ensured by the generative model for all possible prompts with the word "car" and, thus, designs with different degrees of realism can be generated from similar prompts.

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

Fig. 1. Examples of car shapes generated using Shap-E based on the prompts *"A car"*, *"A sports car"*, and *"A compact car"*.

Therefore, in the present paper, we discuss the potential of multi-modal large language models as representations of 3D objects for simulation-based engineering design optimization from a practitioners perspective. Firstly, we propose a fully automated computational 3D design optimization framework for vehicle development that integrates the recently published Shap-E [7] as a text-to-3D generative model. Secondly, we analyze our optimization results to identify benefits for interacting with engineering tools through natural language models. For example, replacing standard representations, such as spline curves/surfaces or free form deformations [4], with text prompts has direct implications not only on the quality of the automatically generated models but also potentially improves the interpretation of the characteristics of the optimized design. Although we use automotive applications in this paper, the methods and conclusions equally apply to other design optimization problems such as from aviation, or marine.

The remainder of this paper is outlined as follows: In Section II, we discuss deep learning models for evolutionary design optimization, like generative models for text-to-3D tasks and prompt engineering. Section III details our proposed design optimization framework with a focus on the different approaches for representing text prompts. In Section IV, we demonstrate the application of our framework to a simulation-based design optimization for the minimization of vehicle drag coefficients and discuss the results of our experiments. Section V concludes the paper.

II. Literature review

In this section, we will describe evolutionary design optimization in engineering with learning-based shape representations. Then, we highlight the current state in text-to-3D and text-to-image generative models followed by approaches for prompt engineering.

A. Geometric deep learning for 3D vehicle optimization

The optimization of the shape of 3D objects is an important step in product design. For automotive engineering, the shape of cars is optimized e.g for fuel efficiency, or crash safety. A computational engineering optimization framework typically consists of the shape representation(s), the optimization algorithm for modifying the shape parameters, and simulation tools for determining the design performance. Although engineers typically manipulate a single representation to generate the design, three different types of representation are often utilized during an optimization: A numerical vector with the shape parameters, which are modified during the optimization; a refined polygonal mesh, which is generated from the shape parameters and utilized for the simulations; and a manufacturing oriented representation that is also generated from the shape parameters and is utilized for cost and feasibility evaluations. In this paper, we mostly refer to design representation as the vector of parameters that is subject to optimization. However, the transition between the different representations is of high practical relevance and should not be overlooked in the computational engineering optimization framework.

The introduction of geometric deep learning architectures [5] enabled the development of 3D deep-generative models for engineering tasks. Most of the currently available works focus on learning compact representations of 3D objects for shape generation and performance prediction. Hence, geometric deep-generative models have the potential to accelerate the design process compared to standard representations (*e.g.*, splines), which require experience from the user and often need to be adapted throughout the optimization to balance out design flexibility and computational effort. Furthermore, as the characteristics of the optimum design are limited by the selected representation, *i.e.*, the degrees of freedom to modify the design, learning-based representations generally identify less intuitive design parameters than humans do, which increases the potential to generate novel unique designs.

In [8], Umetani proposes a system for generating 3D car designs and for predicting the corresponding aerodynamic performance. The system is based on a deep autoencoder architecture, where, by manipulating the values of the learned latent representation, the user quickly generates 3D car designs and obtains an estimate of the aerodynamic drag of the shape. Rios *et. al* build upon a 3D point cloud autoencoder [6] and proposed an automated framework for car design optimization based on evolutionary algorithms [9] and multi-task optimization methods [10]. In this case, the autoencoder learns the design features from an existing data set of CAE models and enables the optimization to combine features from arbitrarily different 3D shapes to evolve the designs during he optimization. Similarly, Saha *et. al* evaluated point-based variational autoencoders (VAEs) with respect to their shape-generative and performance prediction capabilities [11], [12]. The authors claim that the regularization of the latent space improves the shape-generative capabilities of the architecture with respect to the regular autoencoder model by learning a smoother latent space. Nevertheless, the regularized representations also hinder the accuracy of surrogate models trained to predict design performance data based on the latent variables.

However, the application of learning-based geometric representations in optimization also raises some challenges in optimization. As observed in any data-driven model, the generative capabilities of DNNs are bounded by the characteristics of the training data, both in terms of quality of the generated models and generalization aspects. Furthermore, the learned design features lack semantic interpretations, which is neglected by automatic optimization frameworks but hinders

the development of cooperative engineering design systems. These challenges are potentially addressed by the recently introduced large language models combined to 3D shape-generative networks, which are trained on significantly larger data sets and allow the user to interpret the designs based on text prompts.

B. Deep generative models for text-to-3D assets

Recently, Point-E [13] and Shap-E [7] have been proposed by OpenAI which offer first capabilities to generate 3D objects from text prompts. Point-E is a generative model that produces 3D point clouds from text prompts. It first generates a single synthetic view using a text-to-image diffusion model, here GLIDE [14], which is followed by a second diffusion model which is conditioned on the generated image to calculate the 3D point cloud. These text-to-image models, such as GLIDE, Dall-E [15], Midjourney and Stable Diffusion [3], rely on diffusion models that are trained on large data sets of annotated images and can generate high-quality images from noisy images. In the second step of Point-E, Nichols *et al.* [13] proposed a novel transformer architecture to include RGB colors of each point and trained the network on (image, 3D) pairs. In a downstream process, they used an upscaler on the point clouds and traditional (limited) meshing methods for rendering purposes.

Shap-E [7] builds upon Point-E and generates 3D meshed objects, which is often required for rendering simulation-based applications in engineering. The network is trained on a similar data set as Point-E, which comprises refined point clouds (16K points) and rendered views of 3D objects from multiple classes that are paired to text prompts. Similar to Point-E, the network relies on diffusion operations in the latent space representations, but learns to generate implicit representations (signed distance functions) of 3D objects, which are utilized in a differentiable implementation of the marching cubes 33 algorithm to generate polygonal meshes.

Dreamfusion [16] utilizes a pretrained 2D text-to-image diffusion model to perform text-to-3D synthesis to avoid the need for large scale 3D labelled data. By optimizing a randomly-initialized 3D Neural Radiance Field model (NeRF), which consists of a volumetric raytracer and a multilayer perceptron, and by using their proposed Score Distillation Sampling loss function, the authors generated coherent 3D scenes from text prompts. Magic3D [17] overcomes the slow optimization based on NeRFs and low-resolution images obtained with Dreamfusion by optimizing neural field representations on a coarse level (color, density, and normal fields) and extracting a textured 3D mesh from the density and color fields. Get3D [18] utilizes 3D generative models that synthesize textured meshes for direct usage in 3D rendering engines. They combine a differentiable explicit surface extraction method based on signed distance fields to get a 3D mesh topology and a differentiable rendering technique to learn the texture of the surface. With Gaudi [19], the authors propose a generative model capable of capturing the distribution of complex and realistic 3D scenes that can be rendered from a moving camera. In addition, recently the first attempts have also been made to utilize chatGPT for writing python scripts for blender[1] to generate 3D scenes using text prompts. Nevertheless, apart from Point-E and Shap-E, these text-to-3D models are not openly available and, thus, we utilize Shap-E in our experiments.

C. Prompt engineering and optimization

The interest on prompt engineering [20] and prompt optimization has been increasing with the popularization of *text-to-X* models. In text-to-image applications, prompt engineering allows the user to generate images with higher quality by, *e.g.*, including references to famous artists, style of the image, and negative descriptions to avoid ill-defined images, such as anatomically incorrect hands.

In [21], the authors utilize an interactive evolutionary approach to find optimal prompts in a text-to-image scenario. They create meta prompts to represent spaces of prompts and then use a genetic algorithm to improve the prompt based on feedback interactively provided by the user on resulting image qualities. In [22], the authors apply a multi-objective evolutionary optimization to evolve user prompts for image generation for improved consideration of user preferences. The authors claim that especially the usage of the conditional generative model as a kind of mutation guidance is novel. In [23], the authors build a surrogate model, which has been trained on pairs of 2D vehicle renderings and associated drag coefficients resulting from CFD simulation, and integrate it into a text-to-image model, here Stable Diffusion. As a result, their drag-guided diffusion model is capable to generate vehicle images that are aerodynamically efficient.

III. Methods

To evaluate a text-to-shape deep neural network as a 3D shape-generative model in simulation-based design optimization problems, we propose to adapt an evolutionary design optimization framework that is frequently applied in engineering optimization applications using traditional representations [4], [24], when performance gradient information is unavailable. We adapt the framework by integrating the text-to-3D generative model for mapping the optimization parameters to 3D shapes (Fig. 2).

The framework comprises three main components: An optimizer for directed parameter variation, a text-to-shape model for design generation, and a simulation tool for design performance computation. Based on an initial text prompt, the evolutionary optimizer generates a population of prompts by randomly modifying the initial text. Then, the prompts are read by the shape generation model, which generates corresponding 3D shapes represented as polygonal meshes, which are post-processed and used for the computational simulation. Here, we use OpenFOAM for computational fluid dynamics calculations to determine aerodynamic efficiency. Finally, based on a set of convergence criteria such as convergence of step size adaption or maximum iteration counts, either the optimization loop

[1] https://www.blender.org

stops and yields the best performing shape, or iterates by generating a new population of prompts based on applying evolutionary operators to the individuals with best performance.

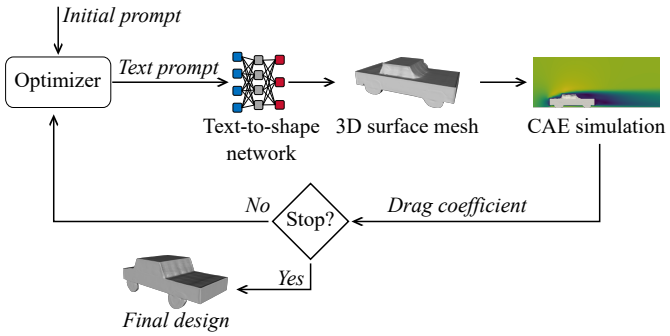

Fig. 2. Workflow of the method utilized in our experiments for optimizing 3D designs using a text-to-shape generative DNN and CAE simulations.

A. Design domain

We consider two approaches for representing the text prompts: Bag-of-words (BoW) and tokenisation. In the bag-of-words approach, we define a prompt template and sets of words which can be utilized to complete the template. Since we target the aerodynamic optimization of car designs, we defined the template as

$$\text{A} < adjective > \text{car in the shape of} < noun >,$$

where the *adjective* and *noun* are sampled from the set of words available in Wordnet [25]. Furthermore, we encoded the words based on the similarity metrics proposed by Wu & Palmer (WUP) [26] with respect to the reference words "fast" and "wing", for the *adjective* and *noun*, respectively, for which we would expect a resulting car with low drag coefficient. Hence, the optimization algorithm yields individuals that encode the distance to the reference words and the prompt is reconstructed by recovering the words in the sets with the most similar distance values. Furthermore, since the sets comprise only words that exist in the English grammar, the generated prompts are human-readable, even though some of the adjective-noun combinations lead to a prompt with counter-intuitive semantic interpretation.

Similarly, in the tokenisation approach, we generate designs by modifying a prompt template:

$$\text{A car in the shape of} < string > .$$

However, we utilize the same byte pair encoding method as in GPT-4 [27] to generate the *string* for completing the prompt template. As the tokens are represented by integers, we directly utilize the values of the tokens as design variables. The main difference with respect to the bag-of-words approach is that the tokenisation generates *strings* with any combination of characters. Hence, although we expect this method to often generate illegible strings, the tokenisation approach allows us to verify the robustness of the network against changes in the prompt and the influence of particular parts of words on the generated designs.

B. Evolutionary optimization

Since gradient information is often challenging to compute from CAE simulation models and in order to cope with multi-modal quality functions, evolutionary algorithms have been frequently applied to complex engineering optimization problems. Apart from being gradient-free, the search mechanisms of evolutionary methods cope with design variables encoded as continuous or binary variables, and can avoid local minima. In our experiments, we utilize the covariance matrix adaptation evolutionary strategy (CMA-ES) [28] to optimize the designs. CMA-ES is an algorithm that has been successfully applied to engineering optimization before and can reach good results even for small population sizes and a limited number of generations. Furthermore, it is relatively robust with regard to the standard settings of the hyperparameters [29].

We set the population size λ to 10, the number of parents μ to 3, and the maximum number of iterations to 100. The population is initialized with the following prompt

$$\text{A } fast \text{ car in the shape of a } wing$$

for the optimizations based on the bag-of-word and with

$$\text{A car in the shape of a } wing$$

for the token representations, respectively. Furthermore, as CMA-ES assumes continuous variables, we generate the token values by approximating the generated parameter values to corresponding nearest integers, and limit the values to available range of tokens ($[0, 32768)$). In both cases, we use a (μ, λ) strategy .

C. Text-to-3D generative model

In our experiments, we utilize a pre-trained version of Shap-E [7] as text-to-3D generative model[2]. The network was trained on an extended version of the data set utilized in [13], which comprises millions of 3D objects from different classes represented as point clouds (16K points) and by renderings. Hence, the utilized version of Shap-E generates objects from different classes and is not specialized on car designs. Also, we utilize the same network hyperparameters as proposed in the literature [7] and vary only the batch size according to the objectives of the experiment.

Since the generative process of Shap-E is probabilistic, i.e., the network generates slightly different shapes from the same prompt, we fixed the seed for random number generation to the same values for all our experiments. Furthermore, we generate 300 designs by feeding the prompt "A car" to Shap-E to compute baseline performance metrics, such as the aerodynamic drag coefficient, which we utilize to evaluate the performance of the optimization framework. Furthermore, as Shap-E generates 3D shapes with different orientations, we re-align the designs assuming that the largest overall dimension

[2]The network architecture and weights are available at https://github.com/openai/shap-e

corresponds to the length of the car (x-axis) and the smallest dimension corresponds to the height (z-axis).

D. Aerodynamics simulation model

The simulation framework utilized in the experiments is a direct adaptation of the available tutorial on aerodynamics simulation of a motorcycle using OpenFOAM[3] [30]. The only modification is on the fluid domain, where the motorcycle shape is replaced by the car geometries generated during the optimization. Further meshing and simulation settings, such as boundary conditions, are kept the same as proposed in the tutorial. We perform the simulations in parallel using 12 processors on a single machine with two CPUs Intel®Xeon®Silver, clocked at 2.10 GHz and 196MB of RAM. The same machine is also equipped with 2 GPUs NVidia®Quadro®RTX 8000 (48 GB each) that are utilized to perform the computations for shape generations using Shap-E.

IV. Experiments and Discussion

In this section, we discuss the experiments performed to evaluate Shap-E as a shape-generative model for evolutionary engineering design optimization. We first discuss the computation of the baseline performance metrics, followed by the optimization cases for each type of representation.

A. Baseline performance metrics

We defined the baseline aerodynamic performance of a car design generated by the network based on the drag coefficient c_d computed for 300 shapes (batch size=300) generated from the prompt "*A car*". To verify the consistency of the simulation model, we also computed the length, height, width and projected frontal area of the generated shapes and verified their relation to the obtained c_d values. For the following analyses, the values of the performance measures are normalized based on the span of values of the baseline set (Eq. 1).

$$c_{d.N} = \frac{c_d}{max(c_{d,baseline}) - min(c_{d,baseline})} \tag{1}$$

By visualizing the obtained distributions of the selected metrics (Fig. 3), we observe that the length of the generated designs is nearly identical for all designs. This effect can be explained by the normalization of the training data, which is a common practice for machine learning applications. Since the projected frontal area typically has the largest impact on the drag coefficient of the designs, we expect the optimization performance to be unaffected by the similarity of the car lengths. Furthermore, we obtained similar distributions for the projected frontal area A_f and the drag coefficient c_d, which is to be expected. By plotting c_d as a function of A_f, we also observe that both are linearly correlated (R-squared of 0.8409), which indicates that the simulation settings are coherent with the physical phenomena.

Additionally, as a reference for the initial performance, we generated 50 designs by feeding the prompt

A $fast$ car in the shape of a $wing$

[3]https://www.openfoam.com/

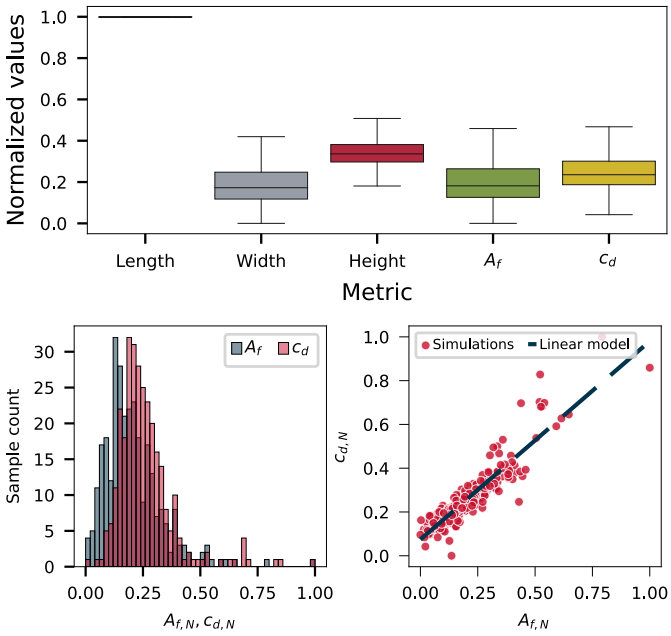

Fig. 3. Visualization of the obtained distributions for the normalized performance measures (top). Correlation between the normalized projected frontal area $|A_f|$ and computed drag coefficient $c_{d,N}$ (bottom).

to Shap-E (Fig. 4) and ran the simulations with the same CFD framework as applied to the baseline shapes. For this set of shapes, we obtained a mean normalized c_d of 0.46 ± 0.08 for a confidence interval of 95%.

Fig. 4. Examples of 3D designs obtained with the prompt "A *fast* car in the shape of *wing*" for computing the initial performance metrics.

B. Prompt-based evolutionary design optimization

In this set of experiments, we utilize the CMA-ES to optimize 3D car shapes by modifying text-prompt templates. For the selected optimization settings (Section III-B), both, the bag-of-words (BoW) and tokenisation approaches yield slow convergence ratios and highly-oscillating population performances throughout the optimization (Fig. 5). However, compared to the tokenisation approach, the BoW-based optimization generated designs with slightly lower c_d, which indicates that the higher degree of freedom of the tokenisation representation cannot be exploited by the optimization.

One of the potential causes for the noisy behavior is the multi-modality of the quality landscape. For the tokenised representation, even though the byte pair encoding method allows to generate a wide range of words from integers, the random variation of the tokens is prone to generate unintelligible expressions (Fig. 6). Therefore, the generated

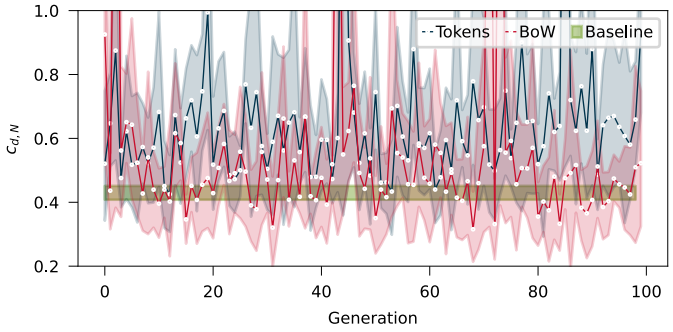

Fig. 5. Baseline and mean of the population c_d value during the optimization. The translucent areas represent a confidence interval of 95% based on the population data for each generation.

shapes are predominately similar to car designs with counter-intuitive modifications and performance values close to the computed baseline. However, the tokens also generate chunks of words with semantic interpretation, from which Shap-E creates designs with mixed features, *e.g.*, a prompt containing the chunk "smoke" yields a cloud-like car design (Fig. 6, right).

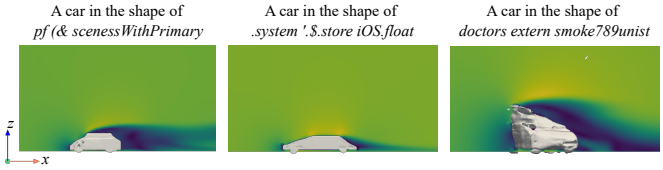

Fig. 6. Examples of 3D shapes generated during the optimization based on the tokenisation approach and the corresponding velocity field obtained in the simulations. Brighter colors indicate higher velocity.

In the BoW approach, the optimization only generates intelligible prompts and, thus, provides a more intuitive relation between the prompts and design properties. We evaluate the prompt-to-shape relation by computing the WUP measure between 300 adjectives and nouns randomly sampled from Wordnet with respect to the word "car", and the Chamfer distance [31] between the shapes generated by feeding only each of the sampled words to Shap-E and the design obtained with the prompt "car". By visualizing the obtained values (Fig. 7), we observe that the samples are clustered around certain WUP values and spread over a wide range of Chamfer distance values.

This behavior is explained by the characteristics of the WUP metric, which is based on the depth of the compared terms in the Wordnet taxonomy. Hence, words that belong to the same semantic class but that describe geometrically distinct objects (e.g., "snake" and "frog" are animals, but physically very different) yield a similar WUP value and high variation in the geometric properties (Fig. 8). Furthermore, this characteristic hinders the mapping of the selected design parameters to the performance measure since, by definition, mathematical functions map each sample in the domain to exclusively one element in the codomain.

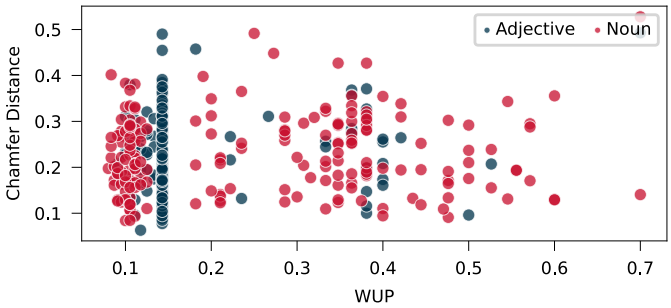

Fig. 7. Visualization of the Chamfer distance as a function of the Wu & Palmer metric for shapes generated with different combinations of adjectives and nouns with respect to designs generated by using "*a car*" as a text prompt.

Fig. 8. Visualization of the geometries generated by similar prompt modifiers. The WUP values of "snake" and "frog" with respect to "car" are 0.35 and 0.36, respectively, but the generated designs are significantly different.

Consequently, the optimization landscape becomes more complex and challenging for the optimization algorithm. In the BoW approach (Fig. 9), we observe that samples with distinct performance values overlap at different points in the design space, and that the landscape lacks a smooth trend for the c_d value, which confirms our hypothesis. Furthermore, the complexity of the landscape also justifies the oscillating behavior of the populations' performance over the generations that was observed in the optimizations.

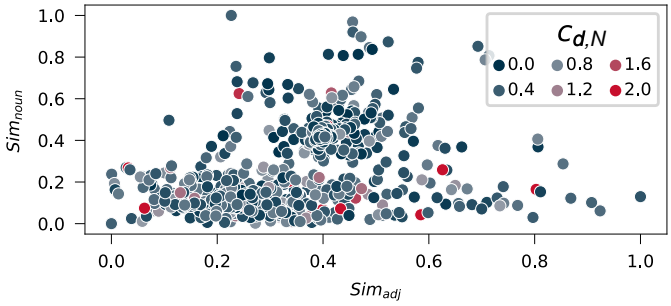

Fig. 9. Visualization of the obtained c_d for the samples generated during the optimization with the BoW approach as a function of the similarity of the adjectives and nouns with respect to the reference words ("*fast*" and "*wing*").

In a second experiment, we changed the evolutionary strategy to $(\mu+\lambda)$ and carried out the optimization using the BoW approach since it seemed to perform better than the tokenisation method. Since the best performing designs are always kept in the population, we expect the mean performance of the population to oscillate less and converge to lower c_d values, even though the algorithm can stagnate at local minima. Indeed the elitist $(\mu + \lambda)$ strategy is able to find designs with

smaller $c_{d,N}$ values (Fig. 10), but the variance throughout the optimization is comparable to the (μ, λ) strategy. This can also be attributed to the lack of correlation between word similarity and geometric distance, as we already discussed.

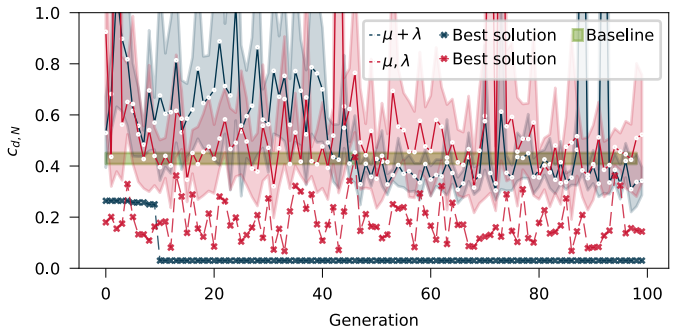

Fig. 10. Mean and minimum normalized $c_{d,N}$ values of the population obtained during the optimizations using the (μ, λ) and $(\mu + \lambda)$ strategies. The translucent areas represent a confidence interval of 95% based on the population data for each generation.

Our interpretation of the results is also supported by the convergence of the design parameters over the generations (Fig. 11). We observe that the variance of the WUP values decreases and stabilizes over the generations, particularly in the $(\mu + \lambda)$ scenario, which is to be expected for the elitist strategy.

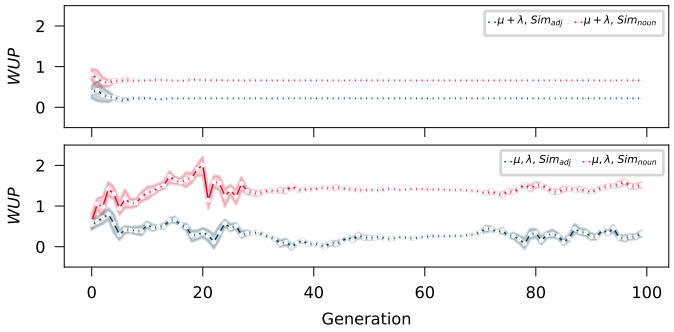

Fig. 11. Mean word similarity metrics obtained over the generations for the noun and adjective in the (μ, λ) and $(\mu+\lambda)$ optimizations. The colored bands indicate a confidence interval of 95% for each metric.

For verification, we visually inspected the initial and best performing designs obtained in the $(\mu+\lambda)$ scenario (Fig. 12). We observe that the design with lowest normalized c_d value (0.02) is similar to a thin tube, which is potentially caused by the words "rifled" and "riffle" in the prompt. Through visual inspection, the latest and best performing car design has a normalized $c_{d,N}$ of 0.15, which is significantly better than the initial design.

To summarize, we show in our experiments that optimizing 3D designs based on text prompts using automated evolutionary optimization algorithms and CFD simulations is feasible. The proposed framework maps words to a numerical design space, which enables the CMA-ES algorithm to sample solutions, and finally generate a design with lower normalized drag coefficient than the initial shape (from 0.46 to 0.15). Yet, as

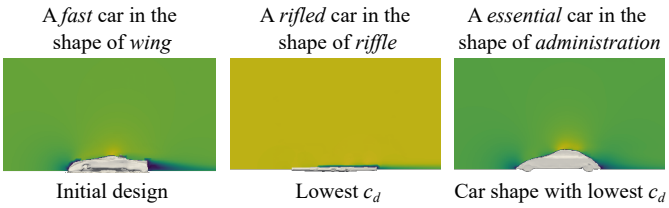

Fig. 12. Initial design, shape with lowest c_d and best performing car design obtained in the optimization with the $(\mu+\lambda)$ strategy. The colors indicate the magnitude of the fluid velocity, where brighter colors indicate higher velocity.

the utilized numerical representation of text prompts is neither canonical nor allows us to map the design performance as a proper mathematical function, the proposed system is prone to generate designs with very distinct geometric properties from similar input values, which reduces the performance of the optimization. Furthermore, other inconsistencies in the geometric representation of the designs, such as mesh quality and orientation of the car shapes (Fig. 13) also increase the noise in the performance values and mislead the search for best performing designs.

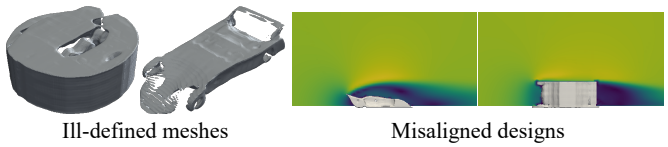

Fig. 13. Examples of generated designs with mesh artifacts (left) and inconsistent orientation with respect to the CFD simulation framework (right).

V. Conclusion and Outlook

In the present paper, we propose an evolutionary design optimization framework, which uses a text-to-3D generative model to map input text prompts to 3D geometries. We utilize two approaches for encoding the text prompts: (a) a bag-of-words approach utilizing Wordnet and (b) a tokenisation approach based on GPT4. The optimization algorithm adapts these prompts for finding 3D shapes with optimal aerodynamic performance, *i.e.* minimum drag coefficients, which are evaluated using a standard CFD tool. The objectives of our studies are to explore the capability of the text-to-3D model embedded in an evolutionary engineering design optimization for (i) generating novel, yet realistic designs, and (ii) for representing a meaningful relation between prompt variations and design variations.

Our results show that the text-to-3D models can be used as an alternative representation in engineering design optimization. Even though searching for designs on a text design space hindered the performance of the optimization, the optimization algorithm still generated designs with lower aerodynamic drag values. Furthermore, many novel possibilities for the interaction with the engineer rise from manipulating 3D designs through text, such as co-exploration of design spaces and qualitative description of design variations. Nevertheless, in the set-up utilized in our experiments and due to the stochastic

nature of the generative model, the optimization also generates geometries unlike cars. This behavior is potentially driven by biases of the network towards specific words. Furthermore, the adaptation of strategy parameters is likely to be severely affected by this complex relation between input variables and drag coefficient, which hinders the performance of the CMA-ES. Thus, other optimization methods, *e.g.*, differential evolution potentially yield better results, as well as the interference in the selection process by an engineer in a collaborative fashion.

Our study is a starting point in the research on using LLMs and text-to-3D models in engineering design optimization. One of the potential future research steps is the introduction of a classifier into the optimization, which identifies non-car like shapes and excludes them from optimization. Another is to augment the semantic distance measure between the different textual descriptions with a measure that relates more to the geometric difference, as well as the exploitation of the interpretability of the representation in the interaction with the engineer.

REFERENCES

[1] R. Bommasani *et al.*, "On the Opportunities and Risks of Foundation Models," 2022. [Online]. Available: https://arxiv.org/abs/2108.07258

[2] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI Blog*, 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," *CoRR*, vol. abs/2112.10752, 2021. [Online]. Available: https://arxiv.org/abs/2112.10752

[4] S. Menzel and B. Sendhoff, "Representing the Change - Free Form Deformation for Evolutionary Design Optimisation," in *Evolutionary Computation in Practice*, T. Yu, D. Davis, C. Baydar, and R. Roy, Eds. Berlin: Springer, 2008, ch. 4, p. 63–86.

[5] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric Deep Learning: Going beyond Euclidean Data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[6] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning Representations and Generative Models for 3D Point Clouds," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 40–49. [Online]. Available: http://proceedings.mlr.press/v80/achlioptas18a.html

[7] H. Jun and A. Nichol, "Shap-E: Generating Conditional 3D Implicit Functions," 2023. [Online]. Available: https://arxiv.org/abs/2305.02463

[8] N. Umetani, "Exploring Generative 3D Shapes Using Autoencoder Networks," in *SIGGRAPH Asia 2017 Technical Briefs*, ser. SA '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3145749.3145758

[9] T. Rios, B. van Stein, P. Wollstadt, T. Bäck, B. Sendhoff, and S. Menzel, "Exploiting Local Geometric Features in Vehicle Design Optimization with 3D Point Cloud Autoencoders," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021, pp. 514–521.

[10] T. Rios, B. van Stein, T. Bäck, B. Sendhoff, and S. Menzel, "Multitask Shape Optimization Using a 3-D Point Cloud Autoencoder as Unified Representation," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 206–217, 2022.

[11] S. Saha, S. Menzel, L. L. Minku, X. Yao, B. Sendhoff, and P. Wollstadt, "Quantifying The Generative Capabilities Of Variational Autoencoders For 3D Car Point Clouds," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 1469–1477.

[12] S. Saha, T. Rios, L. L. Minku, B. V. Stein, P. Wollstadt, X. Yao, T. Bäck, B. Sendhoff, and S. Menzel, "Exploiting Generative Models for Performance Predictions of 3D Car Designs," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 1–9.

[13] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," 2022.

[14] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models," in *Proceedings of the 39th International Conference on Machine Learning, PMLR 162*, 2022, pp. 16 784–16 804.

[15] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-Shot Text-to-Image Generation," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8821–8831. [Online]. Available: https://proceedings.mlr.press/v139/ramesh21a.html

[16] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "DreamFusion: Text-to-3D using 2D Diffusion," *arXiv*, 2022. [Online]. Available: https://arxiv.org/abs/2209.14988

[17] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3D: High-Resolution Text-to-3D Content Creation," 2023. [Online]. Available: https://arxiv.org/abs/2211.10440

[18] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, and S. Fidler, "GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images," 2022. [Online]. Available: https://arxiv.org/abs/2209.11163

[19] M. A. Bautista, P. Guo, S. Abnar, W. Talbott, A. Toshev, Z. Chen, L. Dinh, S. Zhai, H. Goh, D. Ulbricht, A. Dehghan, and J. Susskind, "GAUDI: A Neural Architect for Immersive 3D Scene Generation," 2022. [Online]. Available: https://arxiv.org/abs/2207.13751

[20] L. Reynolds and K. McDonnell, "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm," in *CHI EA '21: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–7.

[21] T. Martins, J. M. Cunha, J. Correia, and P. Machado, "Towards the evolution of prompts with metaprompter," in *Artificial Intelligence in Music, Sound, Art and Design*, C. Johnson, N. Rodríguez-Fernández, and S. M. Rebelo, Eds. Cham: Springer Nature Switzerland, 2023, pp. 180–195.

[22] M. Wong, Y.-S. Ong, A. Gupta, K. K. Bali, and C. Chen, "Prompt Evolution for Generative AI: A Classifier-Guided Approach," 2023. [Online]. Available: https://arxiv.org/abs/2305.16347

[23] N. Arechiga, F. Permenter, B. Song, and C. Yuan, "Drag-Guided Diffusion Models for Vehicle Image Generation," 2023. [Online]. Available: https://arxiv.org/abs/2306.09935

[24] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, p. 481–494, 2002. [Online]. Available: documents/Jin_Tec02.pdf

[25] C. Fellbaum, Ed., *Wordnet*, ser. Language, Speech and Communication. Cambridge, MA: Bradford Books, Jun. 2019.

[26] Z. Wu and M. Palmer, "Verb semantics and lexical selection," in *32nd Annual Meeting of the Association for Computational Linguistics*. Las Cruces, New Mexico, USA: Association for Computational Linguistics, Jun. 1994, pp. 133–138. [Online]. Available: https://aclanthology.org/P94-1019

[27] OpenAI, "GPT-4 Technical Report," 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[28] N. Hansen, "The CMA Evolution Strategy: A Tutorial," *CoRR*, vol. abs/1604.00772, 2016. [Online]. Available: http://arxiv.org/abs/1604.00772

[29] T. Bäck and H. Schwefel, "Evolutionary Computation: An Overview," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 20–29.

[30] P. Dubey, M. Y. Pramod, A. S. Kumar, and B. T. Kannan, "Numerical simulation of flow over a racing motorbike using OpenFOAM®," *AIP Conference Proceedings*, vol. 2277, no. 1, 11 2020, 030029. [Online]. Available: https://doi.org/10.1063/5.0025199

[31] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2017, pp. 2463–2471. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.264