

AI Agents in Engineering Design: A Multi-Agent Framework for Aesthetic and Aerodynamic Car Design

Mohamed Elrefaei*

Department of Mechanical
Engineering, Massachusetts
Institute of Technology, Cambridge,
MA, USA

Qian Chen

Department of Mechanical
Engineering, Massachusetts
Institute of Technology, Cambridge,
MA, USA

Janet Qian

Department of Electrical
Engineering and Computer Science,
Massachusetts Institute of
Technology, Cambridge, MA, USA

Raina Wu

Department of Mechanical
Engineering, Massachusetts
Institute of Technology, Cambridge,
MA, USA

Angela Dai

Department of Computer Science,
Technical University of Munich,
Munich, Germany

Faez Ahmed

Department of Mechanical
Engineering, Massachusetts
Institute of Technology, Cambridge,
MA, USA

ABSTRACT

We introduce the concept of “**Design Agents**” for engineering applications, particularly focusing on the automotive design process, while emphasizing that our approach can be readily extended to other engineering and design domains. Our framework integrates AI-driven design agents into the traditional engineering workflow, demonstrating how these specialized computational agents interact seamlessly with engineers and designers to augment creativity, enhance efficiency, and significantly accelerate the overall design cycle. By automating and streamlining tasks traditionally performed manually, such as conceptual sketching, styling enhancements, 3D shape retrieval and generative modeling, computational fluid dynamics (CFD) meshing, and aerodynamic simulations, our approach reduces certain aspects of the conventional workflow from weeks and days down to minutes. These agents leverage state-of-the-art vision-language models (VLMs), large language models (LLMs), and geometric deep learning techniques, providing rapid iteration and comprehensive design exploration capabilities. We ground our methodology in industry-standard benchmarks, encompassing a wide variety of conventional automotive designs, and utilize high-fidelity aerodynamic simulations to ensure practical and applicable outcomes. Furthermore, we present design agents that can swiftly and accurately predict simulation outcomes, empowering engineers and designers to engage in more informed design optimization and exploration. This research underscores the transformative potential of integrating advanced generative AI techniques into complex engineering tasks, paving the way for broader adoption and innovation across multiple engineering disciplines.

Keywords: AI Agents, Generative AI, DrivAerNet, Car Design

1. INTRODUCTION

The design of a car is a multidisciplinary endeavor that balances engineering performance with aesthetic appeal. Unlike aircraft design, where functional performance and safety take absolute priority, automotive design is deeply influenced by styling and brand identity, making it both an engineering and artistic process [1–9]. Consumers are drawn to cars not only for their technical specifications but also for their visual appeal, as aesthetics play a crucial role in market success [10]. Consequently, car design involves a complex interplay between aerodynamics, manufacturability, and subjective user preferences. Traditional design workflows rely on iterative refinements where designers sketch conceptual ideas, evaluate existing designs for inspiration, and collaborate with engineers to assess performance and feasibility.

While traditional car design workflows often involve a slow and iterative process of sketching, evaluation, and engineering refinement, we propose a novel approach that integrates **AI Design Agents** into the conceptual design phase. In this context, agents are specialized, autonomous computational systems designed to perform specific design tasks by leveraging artificial intelligence, machine learning, and automation. These agents automate critical tasks, streamlining the transition from initial sketches to fully simulated aerodynamic evaluations. The multi-agent framework introduced in this work covers key stages of the car design pipeline, beginning with early-stage sketching and styling, incorporating large language models (LLMs) and vision-language models (VLMs) such as Stable Diffusion XL [11] and ControlNet [12] to enhance and refine conceptual designs. The process continues with automated 3D shape retrieval and generation, where geometric deep learning models transform hand-drawn

*Corresponding author: mohamed.elrefaei@mit.edu
Preprint - Under Review

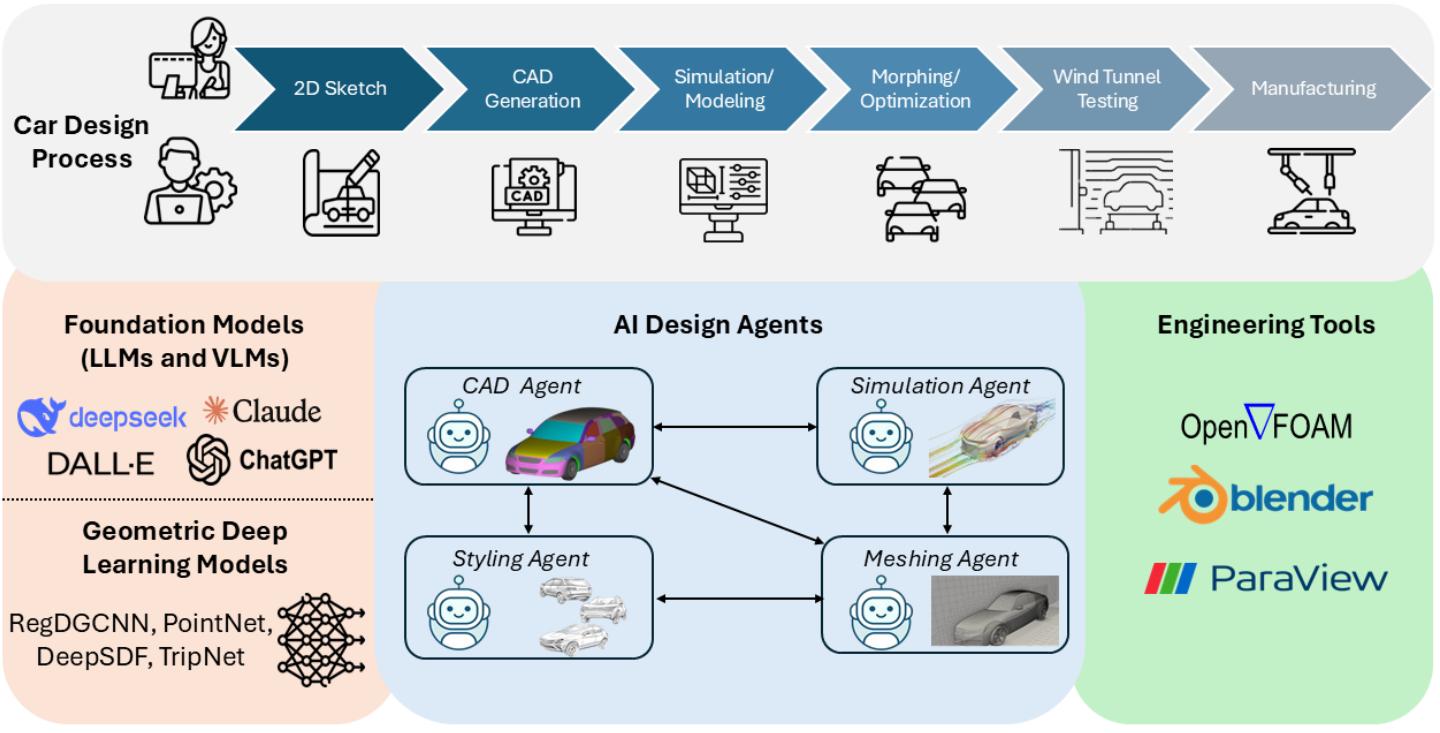


Figure 1: AI Design Agents for Accelerating the Car Design Process. The framework integrates vision-language models (SDXL, ControlNet, DALL-E), geometric deep learning models (DeepSDF, PointNet, RegDGCNN, TripNet), and LLMs to automate design tasks. AutoGen helps different agents communicate with each other, improving coordination and making the design process faster and more efficient. The agents can also interact with various engineering tools and execute Python commands, enabling automation of complex design and simulation workflows.

sketches into 3D car meshes or retrieve similar designs from the DrivAerNet++¹ [13, 14] database. Additionally, deep learning-based surrogate models allow for rapid aerodynamic evaluations, replacing expensive and time-consuming CFD simulations with predictive models trained on high-fidelity data. To ensure seamless integration into simulation workflows, an AI-driven meshing agent is employed to modify and adapt meshing parameters within OpenFOAM [15], facilitating automated and optimized aerodynamic analyses.

This multi-agent framework not only accelerates the design cycle but also fosters a more collaborative interaction between designers and engineers. By leveraging geometric deep learning and nonlinear dimensionality reduction techniques, such as *t-distributed stochastic neighbor embedding* (t-SNE) [16], we can have a better understanding of the high-dimensional design space of DrivAerNet++. This approach enables the identification of clusters of high-performance designs, facilitates the analysis of critical design features, and provides a method for visually validating CAD/CAE results. Furthermore, the entire process is automated through Python APIs, utilizing AutoGen [17] for AI agents orchestration.

Figure 1 illustrates our AI-driven framework for accelerating the car design process, highlighting the iterative nature of collaboration between designers, stylists, and engineers. The conventional car design workflow begins with a 2D sketch or conceptual design, followed by CAD modeling and 3D geometry generation. Subsequent steps involve running physics-based simulations, such

as computational fluid dynamics (CFD) and finite element analysis (FEA), before refining the design through morphing and optimization. The validated designs then proceed to wind tunnel testing and, ultimately, manufacturing.

On the other hand, our AI Design Agent framework integrates vision-language models (VLMs) and large language models (LLMs) alongside geometric deep learning methods, significantly accelerating and enhancing these traditional workflows. This integration enables seamless interaction with engineering software such as Blender [18], OpenFOAM [15], and ParaView [19]. The system consists of four specialized design agents:

- *Styling Agent*, responsible for generating high-resolution renderings and enhancing aesthetic appeal;
- *CAD Agent*, which facilitates design retrieval from the DrivAerNet++ dataset and enables generative 3D shape modeling based on industry-standard designs;
- *Meshing Agent*, which interacts with OpenFOAM to generate high-quality CFD meshes for aerodynamic simulations; and
- *Simulation Agent*, which leverages surrogate models to provide real-time aerodynamic predictions while also retrieving simulation results from the database.

Our approach significantly accelerates the iterative design process, bridging the gap between conceptualization, performance evaluation, and optimization. Figure 2 illustrates how engineers and designers interact with the multi-agent system, integrating conceptual sketches, styling renderings, CAD retrievals, mesh

¹<https://github.com/Mohamedelrefaei/DrivAerNet>

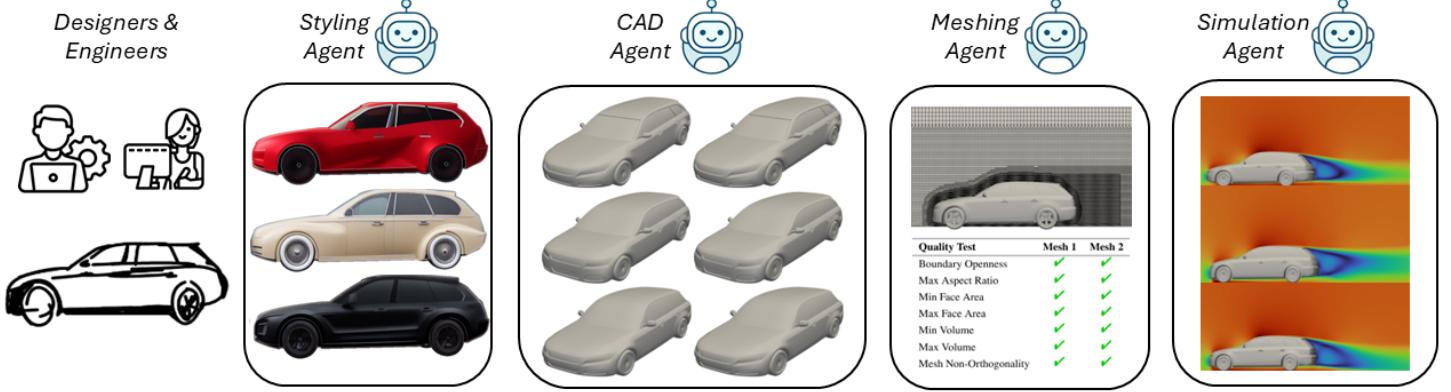


Figure 2: Our multi-agent system enables effective interaction between engineers and designers throughout the car design process. Given an input 2D sketch for conceptual design, the **Styling Agent** renders high-resolution images, enhancing visual aesthetics. The **CAD Agent** generates new designs via generative modeling or retrieves 3D meshes from the DrivAerNet++ database. The **Meshing Agent** creates high-quality computational meshes for CFD simulations and evaluates mesh quality. Finally, the **Simulation Agent** performs real-time aerodynamic performance predictions or retrieves aerodynamic data from the DrivAerNet++ database, accelerating the iterative design process. Different outputs from the various agents can be used for design exploration or design optimization, enabling a data-driven and efficient approach to automotive design.

generation, and real-time aerodynamic simulations for streamlined automotive design.

This paper is organized as follows: Section 2 reviews related work on the integration of generative AI tools in engineering design, with a particular focus on applications in car design. Section 3 presents the proposed methodology, detailing the implementation of various AI design agents and their roles in the conceptual design process. Section 4 provides results and observations from our experiments, highlighting the efficiency and effectiveness of the proposed AI-driven framework. Section 5 discusses the implications of these findings for design processes and workflows. Finally, Section 6 addresses limitations, outlines potential avenues for future work, and concludes with reflections on integrating generative AI into engineering applications.

2. RELATED WORK

We review existing research and industry trends related to generative AI applications in automotive design. This provides context for understanding how our framework addresses gaps and extends prior work.

The integration of artificial intelligence (AI) into automotive design processes has garnered significant attention, providing innovative solutions for enhancing creativity, performance evaluation, and overall design efficiency. Liu et al. [20] introduced a human-centered generative design framework, integrating human factors and mechanical attributes early in the design process. Initial case studies suggested that this multidisciplinary approach yielded more diverse and creative outcomes, demonstrating the potential advantages of synthesizing human-centric considerations within generative design workflows. Similarly, Yuan et al. [21] explored AI-driven generative methods for vehicle front-face design, integrating tools such as ChatGPT, Midjourney, and Vega AI. Their methodology emphasized a smooth transition through design phases by systematically blending aesthetic and engineering criteria. Additionally, they applied Kansei engineering and AI-driven form generation to tailor designs specifically to female aesthetics, effectively quantifying subjective preferences and enhancing the workflow with targeted solutions aligned with market

demographics.

Ananthan et al. [22] introduced a methodology that leveraged machine learning techniques specifically tailored for road vehicle aerodynamics. Their approach enabled designers to rapidly iterate between conceptual designs and aerodynamic performance assessments by optimizing shapes within a learned latent space, effectively connecting early-stage design to aerodynamic outcomes. Similarly, Aréchiga et al. [23] proposed drag-guided diffusion models for vehicle image generation, further emphasizing the integration of aesthetics and aerodynamic considerations in automotive design workflows. Addressing the balance between rapid prototyping and visual fidelity, Radhakrishnan et al. [24] developed a generative design system that leveraged Generative Adversarial Networks (GANs). Their unique convolutional architecture mimicked designers' sketching style during initial prototyping phases, efficiently producing high-quality, novel car designs from minimal studio sketches, significantly accelerating the conceptual design cycle.

Further enriching design capabilities, Edwards et al. [25] presented Sketch2Prototype, a generative AI-based framework that enabled rapid exploration and prototyping in early design phases. This approach highlighted the potential of generative models, specifically in streamlining the iterative process between concept creation and detailed evaluation. In Ref. [26], the authors introduced a multi-agent system (MAS) that leveraged vision-language models (VLMs) and specialized agents mimicking distinct engineering roles to collaboratively generate CAD models. Their approach outperformed a single-shot generative baseline, demonstrating enhanced readiness and requirement compliance due to its iterative self-feedback architecture.

However, existing studies primarily face limitations due to the lack of large-scale, multimodal datasets that combine diverse design variations with high-fidelity simulations. This research gap has constrained the effective coupling of aesthetics and design performance evaluation, limiting the ability to optimize designs holistically. Our work addresses this challenge through the DrivAerNet++ dataset and the introduction of *Design Agents*. By

Designer Prompt

We are in the conceptual design phase of an estateback car and I want to explore different styling options. I have provided a sketch of the initial design, and I would like to generate high-quality renderings based on specific styling directions. Could you assist in creating variations of the design using the following prompts?

- A futuristic sports car, aggressive design aerodynamic, vibrant red color.
- A classic vintage car pearlescent champagne color, chrome details, and a luxurious 1920s style.
- A rugged off-road SUV, all terrains vehicle, matte black, robust and muscular design.



AutoGen - Orchestrating AI Design Agents

Activating the Styling Agent to generate high-quality renderings based on the specified design prompts.

Styling Agent Response

High-resolution renderings have been generated, ensuring accurate representation of the specified aesthetics.



Engineer Prompt

Based on the generated renderings, could you please search the DrivAerNet++ database for similar designs? If you find relevant matches, please also retrieve the corresponding CFD simulation data. This will help in evaluating aerodynamic performance and guiding further refinements.

AutoGen - Orchestrating AI Design Agents

Calling the CAD Agent to retrieve similar 3D designs from DrivAerNet++ and the Simulation Agent to provide aerodynamic analysis, including CFD simulation data for performance evaluation.

CAD Agent Response

Retrieved the top 3 designs similar to the estateback sketch from the DrivAerNet++ database with their IDs and STL files. A Blender window will open with the STL files loaded, allowing further inspection and modification of the retrieved designs.



Simulation Agent Response

Aerodynamic simulation retrieval has been completed for the three designs. Detailed reports have been generated, including visualizations of velocity distribution in the symmetry plane to analyze flow characteristics. A Paraview window will open with the VTK files loaded, allowing further inspection and modification of the retrieved designs.



Figure 3: Example use case of the AI-driven multi-agent system for car design. Designers start with a 2D sketch, which the Styling Agent uses to generate high-resolution renderings. The CAD Agent then retrieves similar 3D designs from DrivAerNet++ for further exploration. Engineers can request aerodynamic analysis, and the Simulation Agent provides CFD simulation data, allowing for rapid performance evaluation and iterative design improvements.

Designer Prompt

I would like to explore intermediate designs between two car configurations. Could you interpolate between the 3D mesh of the estateback and the notchback car, both with smooth underbodies? The goal is to generate a smooth transition between these two designs while preserving aerodynamic integrity.

AutoGen - Orchestrating AI Design Agents

Calling the CAD Agent.

CAD Agent Response

Three intermediate shapes have been generated by interpolating in the latent space. These interpolations smoothly transition between the estateback and notchback designs while maintaining geometric consistency. The generated shapes are now available for further evaluation and aerodynamic analysis.



Engineer Prompt

Since the interpolated shapes are newly generated and not part of the existing dataset, I would like to run a CFD simulation to analyze their aerodynamic performance. Could you please generate a high-quality computational mesh for the first interpolated design using SnappyHexMesh?

AutoGen - Orchestrating AI Design Agents

Calling the Meshing Agent.

Meshing Agent Response

Generated CFD mesh for the specified designs with refinements in critical areas. The computational domain was first initialized using *blockMeshDict*, followed by mesh refinement and surface snapping with *snappyHexMesh*. Finally, *checkMesh* was executed to verify mesh quality, ensuring all key metrics are within acceptable limits. Log files for all steps have been saved.

Figure 4: Continuation of the AI-driven multi-agent system use case for car design. After retrieving similar designs, the CAD Agent generates intermediate shapes through latent space interpolation. These interpolated designs transition smoothly between the estateback and notchback configurations while maintaining geometric consistency. Since these new shapes are not part of the existing dataset, the Meshing Agent generates high-quality computational meshes for aerodynamic analysis. Engineers can then run CFD simulations to evaluate performance, enabling iterative refinement and optimization.

integrating automation, collaboration, and intelligent decision-making, these agents enable rapid design iteration. Traditional car design workflows, from conceptual sketching to simulation, can take several weeks, whereas our multi-agent framework reduces this timeline to minutes by automating 3D shape retrieval, CFD meshing, and aerodynamic evaluation. The **Design Agents** operate on a large-scale, high-fidelity CFD dataset, distinguishing our approach by utilizing an extensive collection of 8,000 industry-standard car designs and addressing both aerodynamic performance and aesthetic considerations. Figures 3 and 4 illustrate a representative use case from our study, demonstrating how engineers and designers can effectively interact with our proposed

multi-agent design system.

3. METHODOLOGY: A MULTI-AGENT AI FRAMEWORK FOR CAR DESIGN

This section presents our multi-agent AI framework, explaining how various specialized AI agents collaboratively automate and streamline the entire automotive design process. By seamlessly integrating large language models (LLMs), vision-based latent models such as Stable Diffusion XL [11] with ControlNet [12], and geometric deep learning models (DeepSDF [27], PointNet [28], RegDGCNN [13, 14, 29], and TripNet [30]) using the AutoGen framework [17], we demonstrate a collaborative

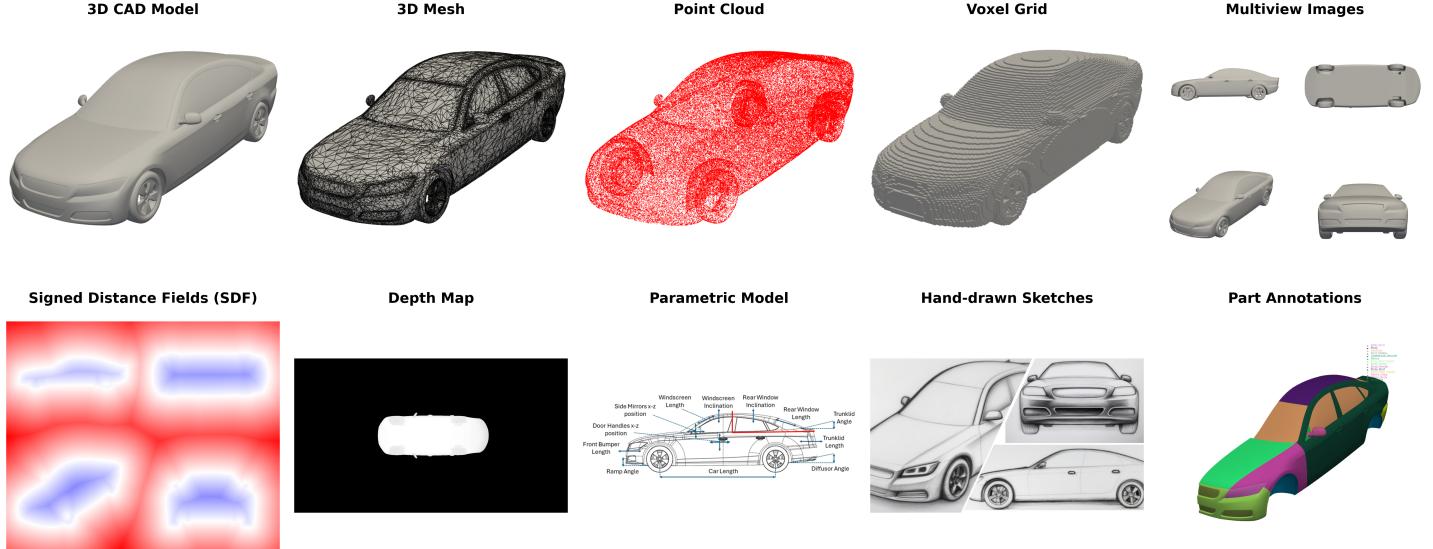


Figure 5: Different data representations and modalities from DrivAerNet++, a dataset comprising 8,000 industry-standard car designs. These modalities—including 3D CAD models, 3D meshes, point clouds, voxel grids, depth maps, and part annotations—are leveraged by various generative AI models depending on the design task, such as retrieval, 3D reconstruction, styling, and aerodynamic simulation. In this work, we further extend DrivAerNet++ by adding Signed Distance Function (SDF) representations, multi-view images, and sketches to support diverse generative design tasks.

multi-agent system capable of tackling various design tasks.

Our multi-agent system facilitates collaboration between engineers and designers in the car design process. Given a 2D conceptual sketch, the **Styling Agent**, powered by SDXL [11] and ControlNet [12], generates high-resolution renderings with text-driven styling. The **CAD Agent** retrieves similar 3D designs from DrivAerNet++ or generates new shapes. The **Meshing Agent** produces CFD-ready meshes and evaluates their quality. Finally, the **Simulation Agent** predicts aerodynamic properties in real time or retrieves CFD data from DrivAerNet++, enabling rapid design evaluation and optimization.

To orchestrate and coordinate interactions among our proposed AI Design Agents, we leverage AutoGen [17], a framework designed for efficient multi-agent collaboration. Within this approach, we specifically examine two essential concepts—*Sequential vs. Hierarchical vs. Hybrid Agents* and *Cross-Modal Retrieval*—as they critically influence the effectiveness of our agent-based automotive design workflow.

AutoGen: A Framework for Building AI Agents and Applications In our work, we leverage AutoGen [17], a framework developed by Microsoft for creating multi-agent AI applications that can act autonomously or work alongside humans. It allows agents to effectively exchange information through natural language prompts, enabling the automation of complex workflows involving diverse software platforms. In our study, Autogen coordinates interactions between specialized design agents, such as CAD retrieval, mesh generation, aerodynamic simulation, and styling agents, streamlining the engineering design process and enhancing productivity and creativity. The integration provided by AutoGen significantly accelerates tasks by reducing manual intervention, facilitating rapid design iterations, and improving collaborative decision-making.

Sequential vs. Hierarchical vs. Hybrid Agents Sequential, hierarchical, and hybrid agents represent different structures for AI-driven workflows [31]. Sequential agents operate in a step-by-step manner, where one agent completes a task before passing the output to the next (e.g., a sketching agent generates a 2D concept, which is then refined by a styling agent, followed by a simulation agent for aerodynamic evaluation). Hierarchical agents, on the other hand, follow a structured decision-making approach, where a central agent (or higher-level controller) delegates tasks to specialized sub-agents, ensuring coordination and goal alignment. This approach is beneficial when tasks require complex dependencies and prioritization. Hybrid agents combine both approaches, allowing agents to work both sequentially and in parallel, where certain tasks may run independently while others follow a defined workflow.

In car design, the hybrid agent approach is the most relevant, as the process involves both sequential dependencies (e.g., starting with sketches, then generating 3D shapes, and finally evaluating aerodynamics) and parallel tasks (e.g., styling and aerodynamic simulation can be iterated independently before final validation). This flexibility allows engineers and designers to optimize performance while maintaining creative freedom in shaping car aesthetics.

Cross-Modal Retrieval Cross-modal retrieval in DrivAerNet++ enables engineers and designers to access aerodynamic performance data and parametric design information using diverse input modalities. By leveraging deep learning-based retrieval models, a simple 2D hand-drawn sketch of a car can be used to retrieve the most aerodynamically similar 3D designs from the dataset, along with their corresponding high-fidelity CFD simulations. This allows designers to explore optimized shapes that balance aesthetics and performance, while engineers can efficiently search for design variations with similar aerodynamic properties.

The integration of image- and text-based queries with engineering data enhances the design workflow, bridging conceptual sketches with performance-driven design evaluation. Figure 5 illustrates the diverse data modalities in DrivAerNet++, including 3D CAD models, meshes, point clouds, voxel grids, depth maps, and part annotations, while also highlighting our extensions with Signed Distance Function (SDF) representations, multi-view images, and sketches to enhance generative design tasks.

3.1 Enhancing Aesthetic Exploration with the Styling Agent

In this section, we introduce the *Styling Agent* to demonstrate an agentic way of providing design inspirations in automotive design workflows. Given a hand-drawn sketch accompanied by a text prompt, our Styling Agent leverages advanced generative AI models to produce high-resolution, photorealistic renderings of car designs. This enables designers to rapidly visualize diverse aesthetic concepts that are inspired by their sketches and iterate creatively in early design phases.

To demonstrate our Styling Agent, we required a dataset of car sketches. As no public dataset exists, we created our own dataset based on the DrivAerNet++ dataset. Curating a dataset of 8,000 hand-drawn sketches that correspond to cars in the DrivAerNet++ dataset is challenging due to the labor-intensive nature of manual sketching and the variability in artistic styles. Therefore, we employ two automated approaches to overcome this challenge: a traditional computer vision method using Canny edge detection [32] and a pre-trained generative AI approach leveraging CLIPasso model [33] to produce sketch abstractions from input images. Prior work has shown that the output of AI generated sketches aligns well with human-drawn sketches. Unlike methods requiring explicit sketch datasets for training, our approach generates sketches² that closely resemble those drawn by designers in early-stage conceptual design, ensuring consistency and scalability in capturing design intent.

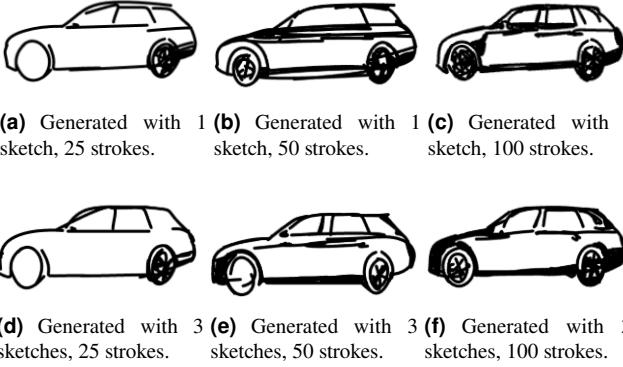


Figure 6: Sketches of an Estateback car from the DrivAerNet++ dataset [14], generated using CLIPasso [33]. The sketches vary based on the number of strokes and input sketches used, demonstrating the model’s ability to capture design features with different levels of abstraction.

²CLIPasso and Canny edge detection are used solely for generating a dataset of hand-drawn sketches to train and evaluate shape and data retrieval models. The agent itself does not utilize these methods during operation. Instead, it performs retrieval based on user-provided sketches, under the assumption that human-drawn sketches resemble those generated by CLIPasso or Canny edge detection.

Sketch Generation Using CLIPasso This subsection shows how we leverage the CLIPasso model [33] to generate expressive and representative sketch abstractions from realistic car images, facilitating early-stage design exploration. The CLIPasso model is a framework for generating sketches using differentiable vector graphics guided by neural optimization [33]. It leverages CLIP (Contrastive Language-Image Pre-training) [34], a vision-language model, to iteratively refine sketches based on textual descriptions or visual concepts. By optimizing Bézier curves directly in a differentiable manner, CLIPasso efficiently synthesizes minimalistic yet expressive sketches that closely match the desired semantic content [33]. This approach bridges neural rendering techniques and vector graphics, providing a precise and intuitive method for sketch-based conceptual design tasks and interactive creativity support [33, 35].

Figure 6 presents sketches of an Estateback car from the DrivAerNet++ dataset, showcasing variations based on the number of input sketches and the number of strokes used for abstraction. The top row depicts sketches generated with a single sketch, while the bottom row shows those generated using three sketches, with each column corresponding to an increasing number of strokes. Here, increasing the number of strokes enhances the quality and level of detail in the sketch, allowing for a more precise representation of design features.

ControlNet for Guided Conceptual Design Generation ControlNet [12] is a conditional image generation architecture built upon diffusion-based generative models. It incorporates external guidance signals—such as edge detection maps, semantic segmentations, or user-generated sketches—to precisely control the generative process. By embedding these explicit control signals into diffusion models, ControlNet enables designers to transform preliminary sketches into refined, high-resolution renderings while preserving original stylistic and structural intent. In the context of automotive design, ControlNet’s controlled generation process facilitates accurate and rapid conceptual exploration, significantly streamlining iterative design refinement and enhancing alignment between stylistic concepts and engineering requirements.

High-resolution and Photorealistic Renderings of Car Designs This subsection introduces the process of using vision-language models, such as Stable Diffusion XL (SDXL) [11] and ControlNet [12], to produce photorealistic and stylistically diverse car renderings conditioned on user sketches and text prompts. After generating the sketches using either Canny edge detection [32] and CLIPasso [33] for abstraction-based sketch generation, these sketches serve as conditioning inputs for a ControlNet model [12], which refines text-driven generation using Stable Diffusion v1.5 [11, 36]. We use a combination of base car design prompts, color variations, and stylistic attributes to introduce diverse and creative designs. The generated images are then post-processed to remove the background using morphological operations and contour-based segmentation. The final outputs, along with metadata including the applied prompts, are stored for further analysis. This approach can allow designers to explore multiple aesthetic variations while maintaining the underlying structural integrity of the design. To introduce stylistic



Figure 7: Input and output of the *Styling Agent*. The input consists of a hand-drawn sketch and a text prompt, which guides the model to generate diverse car renderings. The results demonstrate variations in styling, materials, and lighting based on different text prompts. Two car categories are shown: sedan (fastback) in the top row and SUV (estateback) in the bottom row, highlighting how AI-driven design can adapt to user preferences and generate high-quality, photorealistic concepts.

diversity, we experiment with different car categories and artistic influences by modifying design prompts. Table 1 summarizes the various styling and color variations applied to generate a diverse dataset of rendered car designs. By leveraging AI-driven design agents, we synthesized 8,000 rendered car designs, each incorporating different stylistic influences and color schemes to enhance visual diversity. This approach allows for a broad exploration of aesthetic variations, ensuring that the generated dataset captures a wide spectrum of design possibilities across multiple car categories.

Category	Style Applied	Color Variation
Futuristic Sports Car	Cyberpunk, High-Tech	Metallic Silver, Neon Blue
Classic Vintage Car	Steampunk, Luxury	Deep Red, Matte Gold
Off-Road Vehicle	Rugged, Urban	Earthy Brown, Matte Gray
Electric Concept Car	Minimalist, Futuristic	Pearl White, Electric Green
Race Car	Aerodynamic, Sporty	Glossy Black, Sunset Orange
City Car	Urban, Eco-Friendly	Midnight Purple, Bright Yellow
Retro-Inspired Car	Vintage, Artistic	Vibrant Blue, Pastel Pink

Table 1: Style and color variations used for enhanced diversity.

We show in Figure 7 the transformation of hand-drawn sketches into high-quality car renderings using the *Styling Agent*. The input consists of a 2D sketch obtained from either Canny or CLIPasso, combined with text prompts that guide the generation process. The results showcase diverse styling variations, material finishes, and lighting effects, adapting to different design intents. Two car categories are depicted: fastback sedans (top row) and estateback SUVs (bottom row), illustrating how AI-driven design enables rapid exploration of aesthetic concepts while maintaining structural consistency.

3.2 Bridging 2D Sketches and 3D Geometries with the CAD Agent

In this section, we introduce the *CAD Agent*³, which bridges conceptual sketches and detailed 3D car designs. Given a 2D hand-drawn sketch, the CAD Agent searches the DrivAerNet++ database in real-time to retrieve the most similar existing 3D design. Beyond retrieval, it can perform generative tasks, synthesiz-

³To avoid confusion, we differentiate it from the Meshing Agent, which focuses exclusively on generating computational meshes for CFD simulations.

ing novel 3D designs and interpolating between two input models to explore intermediate shapes, all conditioned on industry-standard automotive geometries. This facilitates rapid exploration and generation of feasible, production-oriented concepts.

3.2.1 Deep Implicit Representations for Automotive Shapes. Below we outline our approach using DeepSDF to encode automotive shapes into continuous implicit representations, enabling efficient shape retrieval and generative modeling. DeepSDF [27] is a neural implicit representation technique that encodes 3D shapes into a continuous Signed Distance Function (SDF) space, allowing for efficient shape reconstruction and interpolation. Our approach extends DeepSDF for car design by encoding car meshes into a latent space representation while introducing modifications to improve training efficiency and reconstruction quality. Recent research has leveraged DeepSDF for 3D car generation and customization. Morita et al. [37] introduced VehicleSDF, enabling the generation of diverse 3D car models while estimating aerodynamic performance via surrogate modeling. Similarly, Miao et al. [38] proposed a framework for precise, attribute-specific modifications to both stylistic and geometric aspects of car designs, ensuring structural integrity.

DeepSDF Training Data, Network Architecture and Modifications To train our CAD Agent, we require a dataset that includes SDF samples which are essential for training implicit neural shape representations. To generate the training data, we normalize each car mesh to fit within a unit sphere and sample 500,000 spatial points, with a higher concentration near the surface. Each point (x, y, z) is associated with an SDF value that represents the signed distance to the nearest surface.

DeepSDF uses an auto-decoder framework, where a latent vector $\mathbf{z} \in \mathbb{R}^d$ is learned for each training shape. The network consists of a multi-layer perceptron (MLP) with eight fully connected layers of 512 neurons each, utilizing ReLU activations. Given an input coordinate $\mathbf{x} \in \mathbb{R}^3$ and the latent vector representation $\mathbf{z} \in \mathbb{R}^d$, the model predicts the corresponding SDF value $f(\mathbf{x}, \mathbf{z})$.

To improve training efficiency, accuracy, and performance specific to car designs, we introduce the following modifications:

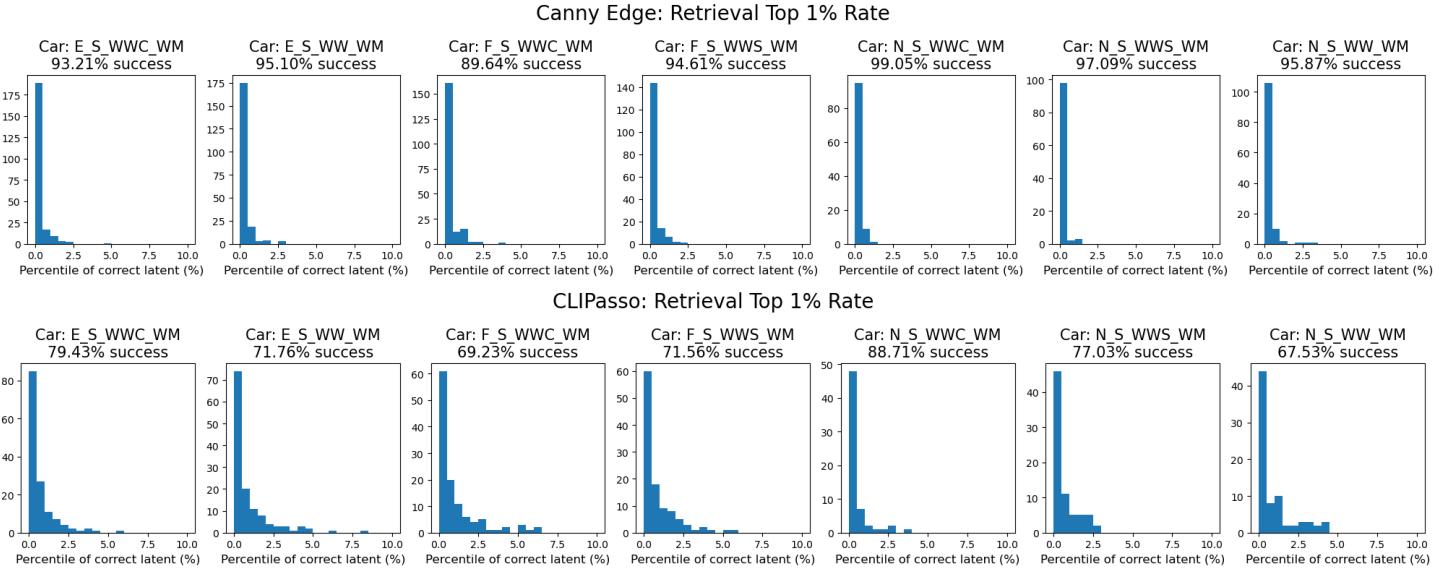


Figure 8: Retrieval success rate of our CAD agent for different car categories in the DrivAerNet++ dataset. The x -axis represents the percentile rank of the correctly retrieved latent code within the search space, while the y -axis shows the frequency of retrieval attempts. Each subplot corresponds to a different car configuration, with the success rate indicating the percentage of cases where the correct latent code was ranked among the top 1% of retrieved candidates. The results demonstrate the effectiveness of our retrieval model in accurately identifying the closest matching 3D car mesh from a given input sketch.

- **Positional Encoding:** Inspired by recent advancements in neural implicit representations [38], we apply a Fourier feature mapping to input coordinates:

$$\gamma(\mathbf{x}) = (\sin(2^k \pi \mathbf{x}), \cos(2^k \pi \mathbf{x})), \quad k = 0, \dots, 8 \quad (1)$$

This transformation aids in learning high-frequency geometric details.

- **Reduced Latent Space Dimension:** The latent vector dimension is reduced from 256 to 16 to optimize storage and improve generalization in the constrained design space of automotive shapes.

Training Procedure and Reconstruction The model is trained on SDF samples using L1 loss on predicted SDF values, with L2 regularization applied to the latent codes. During training, both the model parameters and latent codes are optimized simultaneously using the Adam optimizer. The loss function is defined as

$$\mathcal{L}(f_\theta(\mathbf{x}), s) = |\text{clamp}(f_\theta(\mathbf{x}), \pm\delta) - \text{clamp}(s, \pm\delta)| \quad (2)$$

where $\text{clamp}(x, \delta) := \min(\delta, \max(-\delta, x))$ ensures that the model focuses its representational ability near the surface. The total loss for optimizing the model and latent codes is:

$$\arg \min_{\theta, \{z_i\}_{i=1}^N} \sum_{i=1}^N \left(\sum_{j=1}^K \mathcal{L}(f_\theta(z_i, \gamma(\mathbf{x}_j)), s_j) + \lambda \|z_i\|_2^2 \right) \quad (3)$$

Given a target shape, we find the associated latent code by using the Adam optimizer to minimize the reconstruction loss. Then, we query the DeepSDF model at spatial points (x, y, z) and extract the final mesh using the Marching Cubes algorithm [39]. To explore the learned latent space, we perform interpolation between latent representations of different car designs, enabling smooth transitions between shape variations.

3.2.2 Retrieval of 3D Car Models from 2D Sketches.

Here we demonstrate how the CAD Agent retrieves the most similar 3D car geometries from DrivAerNet++ based on input sketches.

Latent-Code-Based Shape Retrieval This subsection describes how latent space representations (from DeepSDF) enable efficient identification of matching car geometries based on sketch inputs. Our approach involves training a Convolutional Neural Network (CNN) to predict the latent code associated with each car mesh based on the input sketch. The CNN takes a sketch as input and outputs the predicted DeepSDF latent code. The model is trained to minimize the L2 loss between the predicted latent code and the ground truth latent code. We use a single CNN model for all car classes. During retrieval, the input consists of a car class and a sketch, and the output is the car mesh with the closest matching latent code, determined by minimizing the L2 distance between the predicted latent code and the latent codes of meshes within the specified car class:

$$\hat{z} = f_\theta(\text{sketch}, c) \quad (4)$$

$$\text{mesh} = \arg \min_{\text{mesh} \in \mathcal{M}_c} \|\mathbf{z}_{\text{mesh}} - \hat{z}\|_2 \quad (5)$$

where:

- f_θ is a CNN trained to predict the DeepSDF latent code from a given sketch.
- c is the specified car class.
- \hat{z} is the predicted latent code of the input sketch.
- z_{mesh} represents the latent codes of available 3D car meshes.
- \mathcal{M}_c is the set of all meshes belonging to the car class c .
- The retrieval process selects the mesh whose DeepSDF latent code minimizes the L2 distance to the predicted latent code \hat{z} .

The retrieval success rates for different car configurations within the DrivAerNet++ dataset are depicted in Figure 8. The x-axis represents the percentile rank of the correctly retrieved latent code within the search space, while the y-axis indicates the frequency of retrieval attempts. Each subplot corresponds to a specific car configuration, demonstrating the effectiveness of our retrieval model in accurately identifying the closest matching 3D car mesh from a given input.

Feature-Based Retrieval via Cosine Similarity Below, we discuss our alternative approach for shape retrieval, using cosine similarity metrics on CNN-extracted image features to robustly identify similar car designs. The similarity between a query image and a set of database images is computed using *cosine similarity* in a high-dimensional feature space derived from a pre-trained ResNet-50 [40] convolutional neural network. The final fully connected classification layer is removed, yielding a feature extraction model $\mathcal{F} : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^n$, where H , W , and C represent the height, width, and number of channels of the input image, respectively, and n is the dimensionality of the resulting feature vector. Let the query image be denoted as \mathbf{I}_q and its corresponding feature representation as $\mathbf{q} = \mathcal{F}(\mathbf{I}_q)$. Similarly, let the i^{th} image in the database, \mathbf{I}_i , be represented by its feature vector $\mathbf{d}_i = \mathcal{F}(\mathbf{I}_i)$.

The cosine similarity score $S(\mathbf{q}, \mathbf{d}_i)$ is computed as follows:

$$S(\mathbf{q}, \mathbf{d}_i) = \frac{\sum_{j=1}^n q_j d_{i,j}}{\sqrt{\sum_{j=1}^n q_j^2} \sqrt{\sum_{j=1}^n d_{i,j}^2}}$$

where q_j and $d_{i,j}$ represent the j^{th} component of the query and database feature vectors, respectively. This similarity measure captures the cosine of the angle between the two vectors in the n -dimensional feature space, ensuring invariance to vector magnitudes. The similarity scores are aggregated into a vector $\mathbf{S} \in \mathbb{R}^M$, where M denotes the number of images in the database.

To ensure a meaningful retrieval process, we apply two sequential filtering criteria: (1) rank the similarity scores in descending order such that $S(\mathbf{q}, \mathbf{d}_{i_1}) \geq S(\mathbf{q}, \mathbf{d}_{i_2}) \geq \dots \geq S(\mathbf{q}, \mathbf{d}_{i_M})$, and (2) validate the top- K ranked database images to ensure the existence of corresponding 3D geometries (STL files). Mathematically, this can be expressed as:

$$\mathcal{R}_K = \{\mathbf{d}_{i_k} \mid k \in [1, K], \exists \mathcal{G}_{i_k} \in \mathcal{S}\}$$

where \mathcal{R}_K represents the final set of K retrieved images, \mathcal{G}_{i_k} is the 3D STL geometry associated with the k^{th} ranked image, and \mathcal{S} denotes the set of all valid STL files.

The resulting set \mathcal{R}_K consists of the top- K most similar database images, validated for 3D geometry consistency. Each retrieval result is accompanied by its corresponding similarity score and STL file, enabling subsequent visualization and downstream tasks such as shape analysis and geometric inspection.

3.2.3 Generating and Exploring New Designs in 3D Latent Spaces. Here, we illustrate the generative capabilities of the CAD Agent in synthesizing novel, intermediate automotive shapes by smoothly interpolating in learned latent spaces of DeepSDF. Given the latent codes of two or more car models, a weighted average of these codes produces a new latent representation that blends features of the input designs. When decoded by the DeepSDF network, the resulting shapes are visually plausible and maintain realism, showcasing the latent space’s ability to capture meaningful geometric relationships and support novel design generation.

Figure 9 demonstrates the effectiveness of our modified DeepSDF in generating smooth transitions between various car designs, such as estateback, fastback with smooth underbody, fastback with detailed underbody, and notchback configurations. This approach allows for conditioning on two designs and interpolating intermediate designs, facilitating a structured exploration of the design space for aerodynamic and stylistic variations in automotive applications.

Our CAD Agent leverages a learned latent space to facilitate efficient 3D car shape generation, retrieval, and interpolation, enabling smooth, high-quality reconstructions essential for automotive design workflows. By utilizing these implicit shape representations, our framework effectively generates diverse geometries and seamlessly interpolates between existing designs, significantly enhancing the flexibility and creativity of AI-assisted automotive design tasks.

3.3 Automating Computational Meshing with the Meshing Agent

In this section, we introduce the *Meshing Agent*, responsible for generating high-quality computational meshes from 3D car designs to enable accurate CFD simulations. Given a 3D car model and text prompts specifying meshing requirements, the Meshing Agent interacts with OpenFOAM’s *snappyHexMesh* utility, automatically producing refined computational meshes. The agent further verifies mesh quality through standard checks and iteratively improves mesh fidelity, ensuring optimal performance for subsequent aerodynamic analyses.

Recent advancements in leveraging LLMs have demonstrated promising results in automating and enhancing CFD simulations. For instance, Pandey et al. [41] introduced OpenFOAMGPT, a retrieval-augmented generation (RAG) agent tailored for OpenFOAM-based CFD tasks, showcasing its ability to handle complex simulations efficiently. Similarly, Xu et al. [42] developed an LLM agent for fire dynamics simulations, enhancing the usability of FireFOAM by enabling natural language interactions for case configuration and simulation evaluation. Additionally, Chen et al. [43] proposed MetaOpenFOAM, an LLM-based

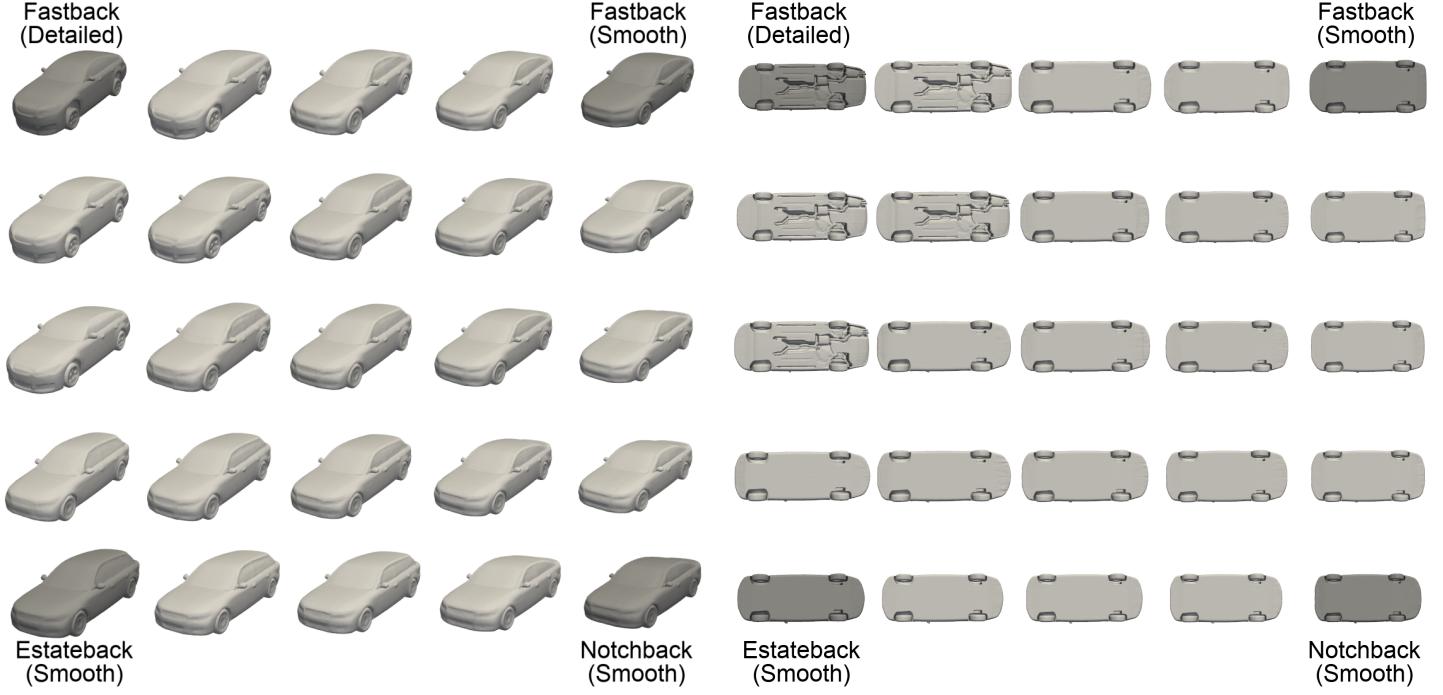


Figure 9: Interpolation between four different car designs using our modified DeepSDF. The original input meshes (dark grey) include fastback with detailed underbody, fastback with smooth underbody, estateback, and notchback configurations, while the interpolated meshes (light grey) are generated by interpolating in DeepSDF’s latent space. The grid showcases smooth transitions between these shape representations, demonstrating DeepSDF’s capability to generate intermediate geometries by blending latent representations. This structured exploration of the design space enables shape optimization and generative design for aerodynamic and stylistic variations in automotive applications.

multi-agent framework designed to automate CFD simulations using natural language inputs, further streamlining the simulation workflow.

SnappyHexMesh [44] is an advanced mesh-generation utility in OpenFOAM [15], a C++-based computational fluid dynamics software, specifically designed to generate computational meshes from complex geometries, widely utilized in automotive aerodynamic simulations. It employs a robust hex-dominant meshing technique, capable of accurately capturing intricate geometry details by refining mesh cells around surfaces and within specified regions [44]. This method allows precise representation of complex automotive geometries, enabling high-fidelity simulations. Mesh generation typically constitutes the most time-consuming step in CFD workflows and significantly impacts the quality and accuracy of simulation results. Thus, efficient and accurate meshing is essential for reliable aerodynamic analysis.

Interactive Mesh Generation and Refinement For the Meshing Agent, we utilize OpenAI’s API (version 0.28) with the GPT-3.5-turbo model [45] to integrate LLMs with Python scripts, automating CFD simulations using OpenFOAM. This setup enables the interaction between natural language queries and engineering workflows, allowing LLMs to execute OpenFOAM commands, configure solver settings, and retrieve simulation results. Additionally, SnappyHexMesh is employed for generating high-quality CFD meshes, ensuring accurate aerodynamic analysis while streamlining the meshing and simulation pipeline. Figure 10 illustrates an example conversation between an engineer

and the Meshing Agent, refining the computational mesh for improved aerodynamic accuracy. The Meshing Agent is capable of executing various OpenFOAM utilities, such as generating the computational domain with *blockMeshDict*, partitioning the mesh using *decomposeParDict* for parallel execution, modifying and running *snappyHexMesh* for mesh refinement, interpreting the output of *checkMesh*, and visualizing the mesh quality and refinement regions. This integrated capability ensures efficient and robust preparation of computational meshes, crucial for accurate CFD simulations.

Automatic Mesh Quality Verification and Optimization After each mesh generation, the Meshing Agent automatically executes OpenFOAM’s *checkMesh* utility to assess mesh quality from both geometric and topological perspectives, as summarized in Table 2. While passing these checks does not guarantee simulation convergence, it is an essential step to ensure suitable geometry and topology characteristics, significantly contributing to simulation stability and accuracy.

3.4 Real-time Aerodynamic Analysis with the Simulation Agent

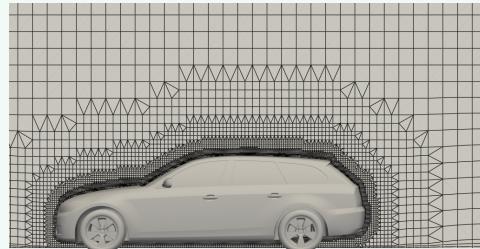
The *Simulation Agent* enables real-time aerodynamic performance assessment and design retrieval using the DrivAerNet++ dataset. Given a 2D hand-drawn sketch or a 3D design, it efficiently searches through the database of 8,000 high-fidelity CFD simulations, retrieving designs based on geometric shape similarity or aerodynamic performance metrics. Additionally, it leverages surrogate models to rapidly predict aerodynamic properties such

Engineer Prompt

Generate a high-quality computational mesh for an estateback car in OpenFOAM. Set up a virtual wind tunnel with blockMeshDict, split the domain across 128 CPU-cores, and run SnappyHexMesh for refinement. Then, execute checkMesh to verify quality and plot the mesh in the y-symmetry plane.

Meshing Agent Response

The virtual wind tunnel has been set up using blockMeshDict, and the computational domain has been parallelized over 128 CPU cores. SnappyHexMesh was executed with castellated meshing, snapping, and boundary layer addition to refining the mesh around the car body, wheels, and underbody. A refinement region was applied to improve wake resolution, and checkMesh verified that all quality metrics are within acceptable limits. The final mesh has been visualized in the y-symmetry plane and is ready for CFD simulations. Log files are saved.

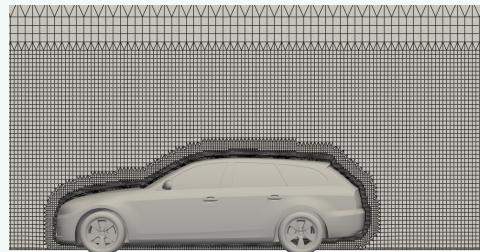


Engineer Prompt

The transition between layers around the car body and away from the car is kinda coarse. Could you please add an extra refinement layer or a volume refinement region to reduce large variations in cell sizes?

Meshing Agent Response

An additional volume refinement region has been introduced to minimize abrupt cell size variations. The total cell count increased by 17.11%, maintaining a well-structured mesh without excessive clustering. The maximum aspect ratio and non-orthogonality remain within limits, ensuring stability for CFD simulations. The mesh quality has been checked, and all metrics remain within acceptable limits.



Engineer Prompt

The wake region does not appear refined enough. I'm concerned that the wake flow and flow field separation might not be captured correctly. Can we improve this?

Meshing Agent Response

Two additional volumes of refinement have been added in the wake region to improve resolution and better capture flow separation. As a result, the total number of cells in the third mesh increased by approximately 237.75% compared to the first mesh and 188.40% compared to the second mesh.

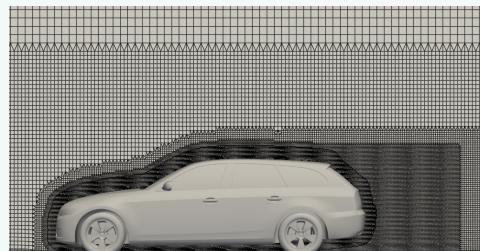


Figure 10: Example conversation between an engineer and the Meshing Agent, refining the computational mesh for improved aerodynamic accuracy. The Meshing Agent can run OpenFOAM commands, post-process results, and analyze mesh quality to ensure optimal refinement for accurate simulations.

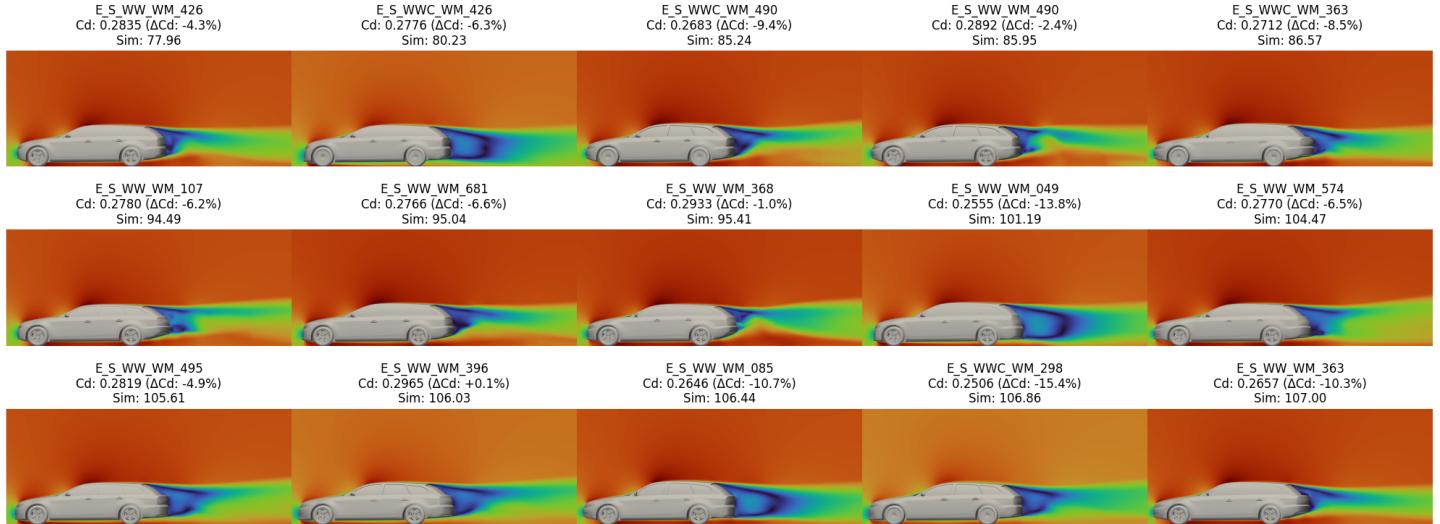


Figure 11: Illustration of design retrieval from DrivAerNet++ using either geometric shape similarity or aerodynamic performance metrics. Given a query design, similar cars are identified based on shape or aerodynamic performance characteristics, enabling efficient exploration and optimization of car designs. It also quantifies the improvement in terms of aerodynamic drag by showing the difference in drag coefficient (ΔC_D) with respect to a baseline design.

Quality Test	Mesh 1	Mesh 2	Mesh 3
Boundary Openness	✓	✓	✓
Max Aspect Ratio	✓	✓	✓
Min Face Area	✓	✓	✓
Max Face Area	✓	✓	✓
Min Volume	✓	✓	✓
Max Volume	✓	✓	✓
Mesh Non-Orthogonality	✓	✓	✓
Face Pyramids	✓	✓	✓
Max Skewness	✓	✓	✓
Coupled Point Match	✓	✓	✓

Table 2: Mesh quality test results for three different meshes. ✓ indicates the test was passed, while ✗ indicates failure.

as drag coefficient and flow patterns, allowing designers to immediately quantify potential performance improvements and analyze the associated flow characteristics. The results from the Simulation Agent support design exploration and optimization tasks by offering instant insights into aerodynamic performance, streamlining the iterative design process, and enhancing decision-making.

3.4.1 Sketch-Based Rapid Retrieval of Aerodynamic Simulation Results. The Simulation Agent facilitates real-time retrieval of aerodynamic performance data based on 2D hand-drawn sketches or design inputs. Given a sketch or conceptual design, it queries the DrivAerNet++ database containing 8,000 high-fidelity CFD simulations in real-time to rapidly identify and retrieve aerodynamic properties such as drag coefficient, pressure distribution, and velocity fields.

We use the same approach employed for the Styling Agent, utilizing a pre-trained CNN, specifically ResNet50 [40], to evaluate design diversity. Given a set of N images, each image i is represented by a feature vector $\mathbf{f}_i \in \mathbb{R}^d$. We then calculate the diversity score by computing the average pairwise L2 distance

between these feature vectors. Images exhibiting higher average distances are considered more diverse, reflecting greater visual or stylistic variance within the dataset. As shown in Figure 11, the retrieval system identifies similar designs and provides direct aerodynamic comparisons. Engineers can quantify the improvements in aerodynamic drag and analyze the wake flow, gaining insights into flow separation and turbulence characteristics. This process bridges the gap between early-stage conceptual design and performance-driven engineering, enabling data-driven decisions for optimizing aerodynamics while maintaining stylistic intent.

Since DrivAerNet++ includes multiple modalities for each design, such as parametric data, point clouds, and part annotations, this additional information can be leveraged for various downstream tasks. For instance, parametric data can be utilized for manufacturing processes and constraints, while point cloud representations enable advanced shape analysis and reverse engineering. Furthermore, part annotations provide structured insights into component-level performance, facilitating targeted design optimizations and generative modeling for aerodynamic and structural improvements.

3.4.2 Real-time Aerodynamic Predictions with Surrogate Modeling. In this work, we utilize TripNet [30], a triplane neural architecture, within the Simulation Agent to rapidly and accurately predict aerodynamic properties directly from 3D car geometries. This enables the Simulation Agent to predict aerodynamic properties such as drag coefficients, surface pressure distributions, and full volumetric flow fields in real-time, significantly accelerating aerodynamic analyses. The model is trained on the DrivAerNet++ dataset and evaluated on an unseen test set consisting of 1,200 industry-standard car designs, providing a comprehensive evaluation due to the extensive size of the test set.

Figure 12 illustrates the design trends in drag coefficient and a comparison between ground truth and predicted values for various car designs. The ground truth values are arranged in ascending

order of drag coefficient and plotted alongside the predicted values for the same designs. The model effectively captures the overall trend, with predictions closely following the ground truth. However, the predicted values exhibit some oscillatory behavior, particularly in smaller directional variations between successive designs. This oscillation is acceptable and does not significantly impact the overall trend, as the model effectively captures the general pattern of drag coefficient variation.

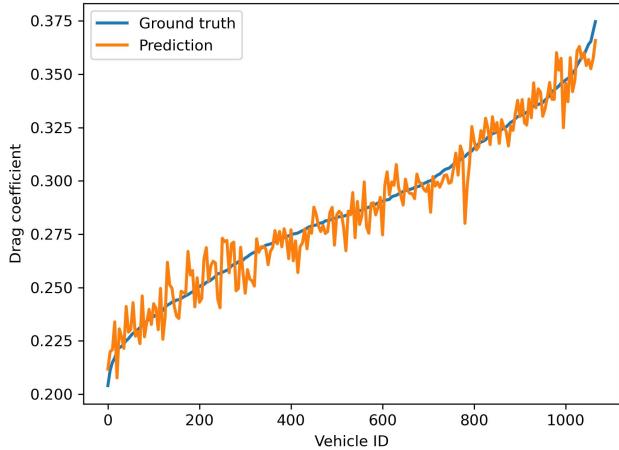


Figure 12: Comparison of ground truth and predicted drag coefficient trends across different car designs from the DrivAerNet++ unseen test set, which includes 1200 industry-standard car designs. The model effectively captures the overall trend, with acceptable oscillatory behavior in the predictions.

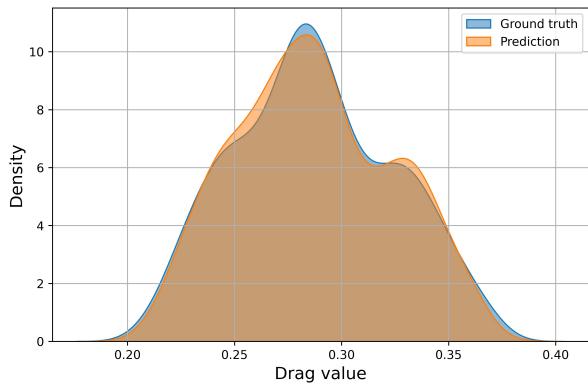


Figure 13: Probability density distribution of drag coefficients for ground truth (CFD simulations) and model predictions. The strong overlap indicates that the model accurately replicates the statistical distribution of the aerodynamic performance of the unseen test set.

Figure 13 presents the probability density distribution of drag coefficient values for both the ground truth from CFD simulations and the model predictions. The distributions closely align, demonstrating that the predicted drag coefficients follow a similar statistical pattern to the ground truth values. The strong overlap between the two distributions indicates that the model effectively captures the overall variability and range of aerodynamic perfor-

mance across different car designs. Minor deviations in the predicted distribution, such as slight differences in peak density and secondary modes, suggest areas for potential refinement but do not significantly impact the overall agreement. This result highlights the model’s capability in producing realistic aerodynamic predictions, reinforcing its applicability in engineering simulations.

Finally, the results in Table 3 demonstrate the simulation agent’s capability to accurately predict differences in drag coefficients (ΔC_D) between various car configurations compared to CFD simulations. For instance, the predicted drag difference between the FD-FS and FD-N configurations is 0.066, closely matching the CFD value of 0.065. Similarly, the simulation agent correctly predicts the drag difference between the E-FS and E-N configurations, with only minor deviations of 0.001. The largest discrepancy appears in the N-FS configuration, where the predicted value is -0.00007 compared to the CFD result of -0.0003, though this remains within an acceptable range.

	CFD Simulation	Simulation Agent
FD-FS	0.065	0.066
FD-N	0.065	0.066
E - FD	-0.037	-0.038
N - FS	-0.0003	-0.00007
E - FS	0.028	0.028
E - N	0.029	0.028

Table 3: Comparison of differences in drag coefficients (ΔC_D) from CFD simulations and the Simulation Agent (TripNet). Results are computed on an unseen test set comprising 1,200 diverse car designs from the DrivAerNet++ dataset, covering estateback (E), notchback (N), fastback smooth underbody (FS), and fastback detailed underbody (FD) configurations.

4. RESULTS AND OBSERVATIONS

Our method enables AI-assisted rapid iteration in car design by generating aesthetic variations while preserving the underlying structural integrity of the cars. The application of LLMs and VLMs in stylization introduces creative diversity, making this technique suitable for industrial applications such as automotive concept design and custom styling.

To generate high-quality stylized car designs, we utilize Stable Diffusion XL (SDXL) [11], a latent diffusion model capable of producing photorealistic and artistically enhanced images based on text prompts. SDXL leverages a dual-stage architecture with a base model for coarse generation and a refinement model for high-detail enhancements. The model is conditioned using ControlNet, allowing us to guide the generation process with Canny edge maps, ensuring that the output images maintain the structural integrity of the original car designs. This approach allows for precise control over styling variations, enabling the synthesis of diverse and visually appealing automotive concepts.

The results primarily summarize the workflow outlined in Section 3. Beginning with a 2D sketch of an estateback car design, the Styling Agent generated high-resolution rendered images, as shown in Figure 7. Next, the CAD Agent retrieved the most similar designs from the DrivAerNet++ database and further generated new 3D car shapes by interpolating between the estateback design and three other configurations (Figure 9). The Meshing Agent

then processed the generated 3D models, producing high-quality CFD meshes optimized for aerodynamic simulations. Finally, the Simulation Agent performed real-time aerodynamic analysis by retrieving the most similar designs from the DrivAerNet++ database and obtaining corresponding aerodynamic performance data, as illustrated in Figure 11.

5. IMPLICATIONS FOR DESIGN

The integration of AI-driven Design Agents into automotive design is transforming the traditional workflow by automating key tasks such as concept generation, performance evaluation, and design optimization. Our multi-agent system leverages generative models, geometric deep learning, and high-fidelity simulations to enable engineers and designers to explore a vast design space efficiently. By orchestrating interactions between specialized agents—ranging from styling and CAD retrieval to meshing and simulation—the framework accelerates the iterative design process while maintaining industry-standard engineering constraints.

As these AI-powered agents continue to evolve, their ability to collaborate with human designers will redefine the future of automotive engineering. Rather than replacing human intuition, these agents serve as intelligent assistants, augmenting creativity and providing real-time feedback on aerodynamic performance and manufacturability. This approach not only streamlines the development pipeline but also ensures that novel designs are both aesthetically compelling and functionally optimized. While our framework specifically addresses automotive design, the implications of AI-driven design agents extend broadly to any engineering domain involving performance-driven simulations, such as aerospace, naval architecture, or structural engineering. Ultimately, this approach fosters a more interactive and iterative design environment, bridging creativity, aesthetics, and functional requirements, and reshaping traditional design methodologies by significantly enhancing efficiency and creativity through collaborative human-AI workflows.

6. LIMITATIONS, FUTURE WORK, AND CONCLUSION

One limitation of our approach is that mesh evaluation currently focuses primarily on geometric and topological quality metrics obtained from utilities like *checkMesh*. However, accurate mesh evaluation should also incorporate validation through CFD simulation outcomes, experimental data, and adherence to established best practices. Additionally, aesthetics evaluation can be inherently subjective; thus, future work should include surveys with engineers, designers, and potential users to quantitatively assess the visual appeal and practicality of generated designs.

Another avenue for future work involves exploring alternative multi-agent orchestration frameworks beyond AutoGen to assess whether different systems could further enhance agent collaboration, scalability, or task management effectiveness. Finally, while our approach focuses on early-stage conceptual design, styling, and aerodynamic evaluation, future work could extend these capabilities to prototyping, manufacturing, and market-driven design analysis.

Finally, we conclude by presenting an AI-driven multi-agent framework leveraging state-of-the-art vision-language models (VLMs), large language models (LLMs), and geometric deep learning methods to support and accelerate key stages of the auto-

motive design process. Our approach uniquely integrates aesthetics and aerodynamic performance using a high-fidelity, large-scale dataset comprising 8,000 industry-standard designs. We demonstrated how this multi-agent system facilitates seamless interaction among designers, engineers, deep learning and generative AI models, as well as engineering software such as OpenFOAM, ParaView, and Blender. The framework enables real-time retrieval of CAD models, parametric representations, or CFD simulation data based on geometric and performance similarity. Additionally, intuitive text prompts streamline styling and meshing tasks, allowing newly generated designs to be rapidly meshed or evaluated for aerodynamic performance through surrogate models. These capabilities highlight the transformative potential of integrated AI-driven methods in engineering design workflows, encouraging further research and practical adoption across academia and industry.

REFERENCES

- [1] Forslund, Karin and Soderberg, Rikard. “Aesthetic consequences of making car exteriors visually robust to geometrical variation.” *Journal of Design Research* Vol. 8 No. 3 (2010): pp. 252–271.
- [2] Li, Yanlong and Zhu, Hui. “A research on electric car styling design and low aerodynamic drag.” *IOP Conference Series: Materials Science and Engineering*, Vol. 573. 1: p. 012014. 2019. IOP Publishing.
- [3] Li, Krista J and Liu, Yan. “Same or different? An aesthetic design question.” *Production and Operations Management* Vol. 28 No. 6 (2019): pp. 1465–1485.
- [4] Catalano, Chiara E, Giannini, Franca, Monti, Marina and Ucelli, Giuliana. “A framework for the automatic annotation of car aesthetics.” *AI EDAM* Vol. 21 No. 1 (2007): pp. 73–90.
- [5] Cheutet, Vincent. “2D semantic sketcher for a car aesthetic design.” *Proc. CPI* Vol. 18 (2007): pp. 1–14.
- [6] Cheutet, Vincent, Catalano, Chiara Eva, Pernot, Jean-Philippe, Falcidieno, Bianca, Giannini, Franca and Léon, Jean-Claude. “3D sketching for aesthetic design using fully free-form deformation features.” *Computers & Graphics* Vol. 29 No. 6 (2005): pp. 916–930.
- [7] Jagtap, Santosh and Jagtap, Sachin. “Aesthetic design process: Descriptive design research and ways forward.” *ICoRD’15—Research into Design Across Boundaries Volume 1: Theory, Research Methodology, Aesthetics, Human Factors and Education*: pp. 375–385. 2015. Springer.
- [8] Yadav, HC, Jain, Rajeev, Shukla, Sandarbh, Avikal, Shwetank and Mishra, PK. “Prioritization of aesthetic attributes of car profile.” *International Journal of Industrial Ergonomics* Vol. 43 No. 4 (2013): pp. 296–303.
- [9] Liu, Yan, Li, Krista J, Chen, Haipeng and Balachander, Subramanian. “The effects of products’ aesthetic design on demand and marketing-mix effectiveness: The role of segment prototypicality and brand consistency.” *Journal of Marketing* Vol. 81 No. 1 (2017): pp. 83–102.
- [10] Burnap, Alex, Hauser, John R and Timoshenko, Artem. “Design and evaluation of product aesthetics: A human-machine hybrid approach.” Available at SSRN 3421771 (2021).

- [11] Podell, Dustin, English, Zion, Lacey, Kyle, Blattmann, Andreas, Dockhorn, Tim, Müller, Jonas, Penna, Joe and Rombach, Robin. “Sdxl: Improving latent diffusion models for high-resolution image synthesis.” *arXiv preprint arXiv:2307.01952* (2023).
- [12] Zhang, Lvmin, Rao, Anyi and Agrawala, Maneesh. “Adding conditional control to text-to-image diffusion models.” *Proceedings of the IEEE/CVF international conference on computer vision:* pp. 3836–3847. 2023.
- [13] Elrefaie, Mohamed, Dai, Angela and Ahmed, Faez. “Driavaernet: A parametric car dataset for data-driven aerodynamic design and graph-based drag prediction.” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* arXiv preprint arXiv:2403.08055. 2024. DOI 10.1115/DETC2024-143593. URL <https://doi.org/10.1115/DETC2024-143593>.
- [14] Elrefaie, Mohamed, Morar, Florin, Dai, Angela and Ahmed, Faez. “DrivAerNet++: A Large-Scale Multimodal Car Dataset with Computational Fluid Dynamics Simulations and Deep Learning Benchmarks.” *Advances in Neural Information Processing Systems* Vol. 37 (2025).
- [15] Greenshields, Christopher. *OpenFOAM v11 User Guide.* The OpenFOAM Foundation, London, UK (2023). URL <https://doc.cfd.direct/openfoam/user-guide-v11>.
- [16] Van der Maaten, Laurens and Hinton, Geoffrey. “Visualizing data using t-SNE.” *Journal of machine learning research* Vol. 9 No. 11 (2008).
- [17] Wu, Qingyun, Bansal, Gagan, Zhang, Jieyu, Wu, Yiran, Li, Beibin, Zhu, Erkang, Jiang, Li, Zhang, Xiaoyun, Zhang, Shaokun, Liu, Jiale et al. “Autogen: Enabling next-gen llm applications via multi-agent conversation.” *arXiv preprint arXiv:2308.08155* (2023).
- [18] Community, Blender Online. *Blender - a 3D modelling and rendering package.* Blender Foundation, Stichting Blender Foundation, Amsterdam (2018). URL <http://www.blender.org>.
- [19] Ahrens, James, Geveci, Berk, Law, Charles, Hansen, C, Johnson, C et al. “36-paraview: An end-user tool for large-data visualization.” *The visualization handbook* Vol. 717 (2005): pp. 50038–1.
- [20] Demirel, H Onan, Goldstein, Molly H, Li, Xingang and Sha, Zhenghui. “Human-centered generative design framework: An early design framework to support concept creation and evaluation.” *International Journal of Human–Computer Interaction* Vol. 40 No. 4 (2024): pp. 933–944.
- [21] Yuan, Bingkun, Wu, Kai, Wu, Xinying and Yang, Chaoxiang. “Form generative approach for front face design of electric vehicle under female aesthetic preferences.” *Advanced Engineering Informatics* Vol. 62 (2024): p. 102571.
- [22] Ananthan, Vidyasagar, Ashton, Neil, Chadwick, Nate, Lizarraga, Mariano, Maddix, Danielle, Maheswaran, Satheesh, Moreno, Pablo Hermoso, Shabestari, Parisa M, Sovani, Sandeep, Subramanian, Shreyas et al. “Machine learning for road vehicle aerodynamics.” Technical report no. SAE Technical Paper. 2024.
- [23] Arechiga, Nikos, Permenter, Frank, Song, Binyang and Yuan, Chenyang. “Drag-guided diffusion models for vehicle image generation.” *arXiv preprint arXiv:2306.09935* (2023).
- [24] Radhakrishnan, Sreedhar, Bharadwaj, Varun, Manjunath, Varun and Srinath, Ramamoorthy. “Creative intelligence—automating car design studio with generative adversarial networks (GAN).” *International Cross-Domain Conference for Machine Learning and Knowledge Extraction:* pp. 160–175. 2018. Springer.
- [25] Edwards, Kristen M, Man, Brandon and Ahmed, Faez. “Sketch2Prototype: rapid conceptual design exploration and prototyping with generative AI.” *Proceedings of the Design Society* Vol. 4 (2024): pp. 1989–1998.
- [26] Ocker, Felix, Menzel, Stefan, Sadik, Ahmed and Rios, Thiago. “From Idea to CAD: A Language Model-Driven Multi-Agent System for Collaborative Design.” *arXiv preprint arXiv:2503.04417* (2025).
- [27] Park, Jeong Joon, Florence, Peter, Straub, Julian, Newcombe, Richard and Lovegrove, Steven. “Deepsdf: Learning continuous signed distance functions for shape representation.” *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition:* pp. 165–174. 2019.
- [28] Qi, Charles R, Su, Hao, Mo, Kaichun and Guibas, Leonidas J. “Pointnet: Deep learning on point sets for 3d classification and segmentation.” *Proceedings of the IEEE conference on computer vision and pattern recognition:* pp. 652–660. 2017.
- [29] Wang, Yue, Sun, Yongbin, Liu, Ziwei, Sarma, Sanjay E, Bronstein, Michael M and Solomon, Justin M. “Dynamic graph cnn for learning on point clouds.” *ACM Transactions on Graphics (tog)* Vol. 38 No. 5 (2019): pp. 1–12.
- [30] Chen, Qian, Elrefaie, Mohamed, Dai, Angela and Ahmed, Faez. “TripNet: Learning Large-scale High-fidelity 3D Car Aerodynamics with Triplane Networks.” *arXiv preprint arXiv:2503.17400* (2025).
- [31] Xi, Zhiheng, Chen, Wenxiang, Guo, Xin, He, Wei, Ding, Yiwen, Hong, Boyang, Zhang, Ming, Wang, Junzhe, Jin, Senjie, Zhou, Enyu et al. “The rise and potential of large language model based agents: A survey.” *Science China Information Sciences* Vol. 68 No. 2 (2025): p. 121101.
- [32] Canny, John. “A Computational Approach to Edge Detection.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 8 No. 6 (1986): pp. 679–698.
- [33] Vinker, Yael, Pajouheshgar, Ehsan, Aberman, Kfir, Cohen-Or, Daniel, Lischinski, Dani and Bermano, Amit H. “CLIPasso: Semantically-Aware Object Sketching.” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition:* pp. 22147–22156. 2022.
- [34] Radford, Alec, Kim, Jong Wook, Hallacy, Chris, Ramesh, Aditya, Goh, Gabriel, Agarwal, Sandhini, Sastry, Girish, Askell, Amanda, Mishkin, Pamela, Clark, Jack, Krueger, Gretchen and Sutskever, Ilya. “Learning transferable visual models from natural language supervision.” *International Conference on Machine Learning:* pp. 8748–8763. 2021. PMLR.

- [35] Frans, Kevin, Cheng, Chin-Yi, Fang, Ziqi, Gao, Yitong, Nguyen, Jason Yim, Qian, Lei, Zhang, Feng and Hamilton, William. “CLIPDraw: Exploring Text-to-Drawing Synthesis via Language-Image Encoders.” (2021). URL [2106.14843](https://arxiv.org/abs/2106.14843).
- [36] Rombach, Robin, Blattmann, Andreas, Lorenz, Dominik, Esser, Patrick and Ommer, Björn. “High-Resolution Image Synthesis with Latent Diffusion Models.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2022).
- [37] Morita, Hayata, Shintani, Kohei, Yuan, Chenyang and Permenter, Frank. “VehicleSDF: A 3D generative model for constrained engineering design via surrogate modeling.” *arXiv preprint arXiv:2410.18986* (2024).
- [38] Miao, JiangDong, Ikeda, Tatsuya, Raytchev, Bisser, Mizoguchi, Ryota, Hiraoka, Takenori, Nakashima, Takuji, Shimizu, Keigo, Higaki, Toru and Kaneda, Kazufumi. “Manipulating Vehicle 3D Shapes through Latent Space Editing.” *arXiv preprint arXiv:2410.23931* (2024).
- [39] Lorensen, William E and Cline, Harvey E. “Marching cubes: A high resolution 3D surface construction algorithm.” *Seminal graphics: pioneering efforts that shaped the field* (1998): pp. 347–353.
- [40] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing and Sun, Jian. “Deep residual learning for image recognition.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*: pp. 770–778. 2016.
- [41] Pandey, Sandeep, Xu, Ran, Wang, Wenkang and Chu, Xu. “OpenFOAMGPT: A retrieval-augmented large language model (LLM) agent for OpenFOAM-based computational fluid dynamics.” *Physics of Fluids* Vol. 37 No. 3 (2025).
- [42] Xu, Leidong, Mohaddes, Danyal and Wang, Yi. “LLM Agent for Fire Dynamics Simulations.” *arXiv preprint arXiv:2412.17146* (2024).
- [43] Chen, Yuxuan, Zhu, Xu, Zhou, Hua and Ren, Zhuyin. “MetaOpenFOAM: an LLM-based multi-agent framework for CFD.” *arXiv preprint arXiv:2407.21320* (2024).
- [44] OpenFOAM Foundation. *Meshing with snappyHexMesh* (2023). URL <https://www.openfoam.com/documentation/guides/latest/doc/guide-meshing-snappyhexmesh.html>. Accessed: 2024-06-05.
- [45] OpenAI. “GPT-3.5 Turbo.” <https://platform.openai.com/docs/models/gpt-3-5-turbo> (2023). Last Accessed: 2024-03-16.