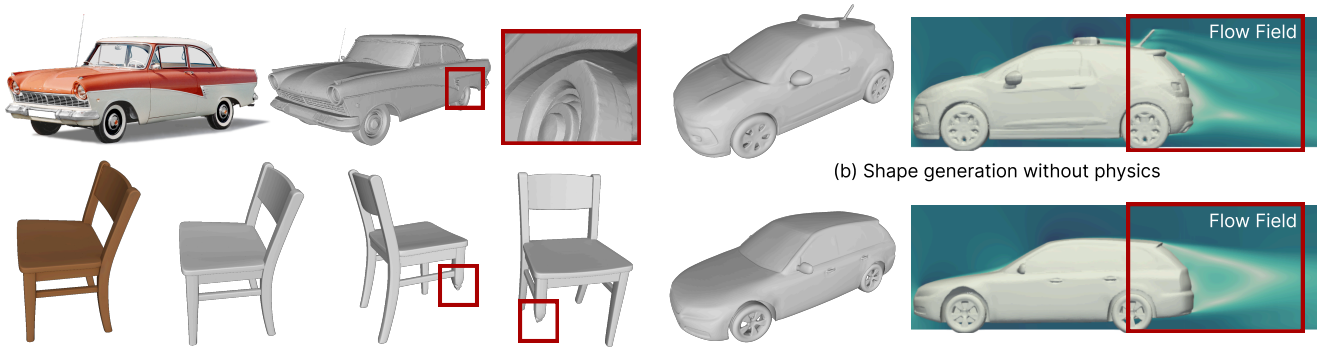
PhysGen: Physically Grounded 3D Shape Generation for Industrial Design

Yingxuan You Chen Zhao Hantao Zhang Ming Xu Pascal Fua
CVLab, EPFL
{yingxuan.you, chen.zhao, hantao.zhang, mingda.xu, pascal.fua}@epfl.ch



(a) Visually plausible but phyisically invalid shape generation
(b) Shape generation without physics
(c) Shape generation with physics (Ours)

Figure 1. Physical knowledge is crucial for realistic and functionally efficient 3D shape generation. Without physics, (a) generated shapes may appear visually plausible yet violate basic physical feasibility, such as car wheels intersecting the body or chairs with broken or unstable legs, and (b) aerodynamic shapes produce wide turbulent wakes, indicating low aerodynamic efficiency. (c) In contrast, physics-guided generation produces shapes with coherent flow and reduced drag, achieving both aesthetic appeal and physical efficiency.

Abstract

Existing generative models for 3D shapes can synthesize high-fidelity and visually plausible shapes. For certain classes of shapes that have undergone an engineering design process, the realism of the shape is tightly coupled with the underlying physical properties, e.g., aerodynamic efficiency for automobiles. Since existing methods lack knowledge of such physics, they are unable to use this knowledge to enhance the realism of shape generation. Motivated by this, we propose a unified physics-based 3D shape generation pipeline, with a focus on industrial design applications. Specifically, we introduce a new flow matching model with explicit physical guidance, consisting of an alternating update process. We iteratively perform a velocity-based update and a physics-based refinement, progressively adjusting the latent code to align with the desired 3D shapes and physical properties. We further strengthen physical validity by incorporating a physics-aware regularization term into the velocity-based update step. To support such physics-guided updates, we build a shape-and-physics variational autoencoder (SP-VAE) that jointly encodes shape and physics information into a unified latent space. The experiments on three benchmarks show that this synergistic formulation improves shape realism beyond mere visual plausibility. Our code and model weights are available at https://github.com/kasvii/PhysGen.

1. Introduction

Generative AI has achieved remarkable success in text [30] and image [46] processing. Meanwhile, 3D content creation has gained increasing attention with applications across a wide range of fields, including virtual reality [28], gaming [7], retail [6], and engineering design [41]. The existing 3D generative models [1, 5, 22, 49] excel in creating apparently realistic objects. However, such realism is only skin-deep. Close examination often reveals inconsistencies or implausibilities, which might be acceptable in some entertainment applications where the objective is merely to produce visually appealing results, but become detrimental when the end goal is engineering-oriented. For example, as illustrated in Fig. 1, the wheels of generated cars [22] touch the body of the car, which makes no sense when designing functional cars. Similarly, the chairs [19] generated from images contain feet with the wrong topology, making them unsuitable to bear the weight of a person.

These failings can be traced to the fact that these methods are trained solely on static datasets [2, 9] of 3D shapes, without regard to the engineering design process that was

used to create them. For instance, automobile and airplane designs [38, 39] are usually optimized for aerodynamic efficiency, which conditions their shape beyond just aesthetics. Such physical awareness is absent from the current 3D shape generation pipelines [1, 19, 22]. To bridge this gap, we demonstrate that physical knowledge can be used to define regularization constraints, which are then used to improve the quality and realism of 3D shape generation.

To this end, we present *PhysGen*, an approach to enforcing physical realism on generated 3D shapes, by using information around the physics of the shape. Specifically, we propose a physics-guided flow matching model, built upon an alternating update algorithm that switches between a velocity-based update and a physics-based refinement. Given the desired physical properties, we further enhance the physical guidance by introducing a physics-aware regularization in the velocity-based update. We update the latent code leveraging gradients derived from the discrepancy between the predicted and target physical values. Notably, 3D generative models typically rely on VAE-learned latent spaces, but existing shape VAEs [5, 49] encode no physics, making physical properties unrecoverable from their latent codes. To enable effective physics-based regularization, we propose a shape-and-physics variational autoencoder (SP-VAE) that embeds both 3D shape and physics information into a unified latent space.

In this paper, we focus on the application of 3D shape generation in automobile design, where physical performance is pivotal to ensuring that generated shapes satisfy real-world engineering requirements. We also showcase additional applications, such as structural optimization [48] under prescribed loads and boundary conditions, which demonstrates the generalization ability of our approach. We evaluate our *PhysGen* across benchmarks [2, 12], including unconditional 3D generation, 3D generation conditioned on a sketch, and 3D generation conditioned on a real single-view image. The results demonstrate that by utilizing physics-based regularization, *PhysGen* produces shapes that are more geometrically plausible and physically efficient than previous approaches. We also conduct comprehensive ablation studies on the key components in our pipeline. In summary, our contributions are as follows:

- We investigate the physical realism in 3D shape generation models and propose a novel physics-guided flow matching model that generates physically efficient and aesthetically pleasing 3D shapes.
- We achieve the physical guidance by alternating between a velocity-based update with physics-aware regularization and a physics-based refinement.
- We build a shape-and-physics variational autoencoder that encodes 3D shape and physics in a unified latent space, enabling the physical guidance in the presented flow matching model.

2. Related Work

2.1. 3D Shape Generation

Inspired by advances in image and video generation, generative models for 3D shape generation [5, 22, 44, 49–51] have recently made rapid progress. One line of work distills knowledge from powerful 2D diffusion models into 3D domains [4, 21, 24, 25, 35, 37, 40], alleviating the scarcity of high-quality 3D data. While effective at producing visually appealing shapes, these methods often exhibit slow convergence and unstable optimization. Another line of work builds native 3D generative models [5, 22, 49, 50]. Methods such as 3DShape2VecSet [49] and Dora [5] commonly adopt a two-stage paradigm, where a 3D VAE first encodes geometry into a latent space, followed by a diffusion model for 3D shape generation. While achieving visually plausible shapes, neither the VAE nor the diffusion models encodes physics information, which limits their usefulness in physics-sensitive domains. To address this, we propose a joint shape-physics VAE that embeds geometry and physics information into a unified latent space, capturing their intrinsic correlations and enabling more physically aware shape generation.

2.2. Physics-Aware Shape Generation

Physics is often incorporated into 3D generation via post-processing optimization [14, 33]. Finite element methods (FEMs) [20, 42, 45] and force density methods (FDMs) [33] can stabilize generated meshes under external forces [14], but they operate solely on explicit meshes. In contrast, generative models encode shape priors implicitly in latent spaces [5, 22, 49], making optimization without shape priors highly sensitive to initialization and prone to failure once shapes drift out of distribution [14]. These limitations motivate physics-aware generation directly in latent space. TripOptimizer [41] applies drag-driven gradient updates to VAE parameters, but often disrupts the model priors learned from the dataset. Recent works [13, 29] inject physical gradients into the diffusion process. However, physics estimates on noisy early samples are unreliable, and the limited steps in later diffusion stages are insufficient for convergence. To address these issues, we propose a physics-guided 3D shape generation method that integrates flow-matching generation and physical guidance into a unified framework. Rather than generating a shape followed by heavy post-optimization, our method alternates between a velocity-based update and a physical refinement, facilitating the convergence toward shapes that are both geometrically plausible and physically valid.

3. Method

Problem Formulation. We aim to generate a 3D shape $\mathcal{S}$, conditioned on physics information A and, optionally, on

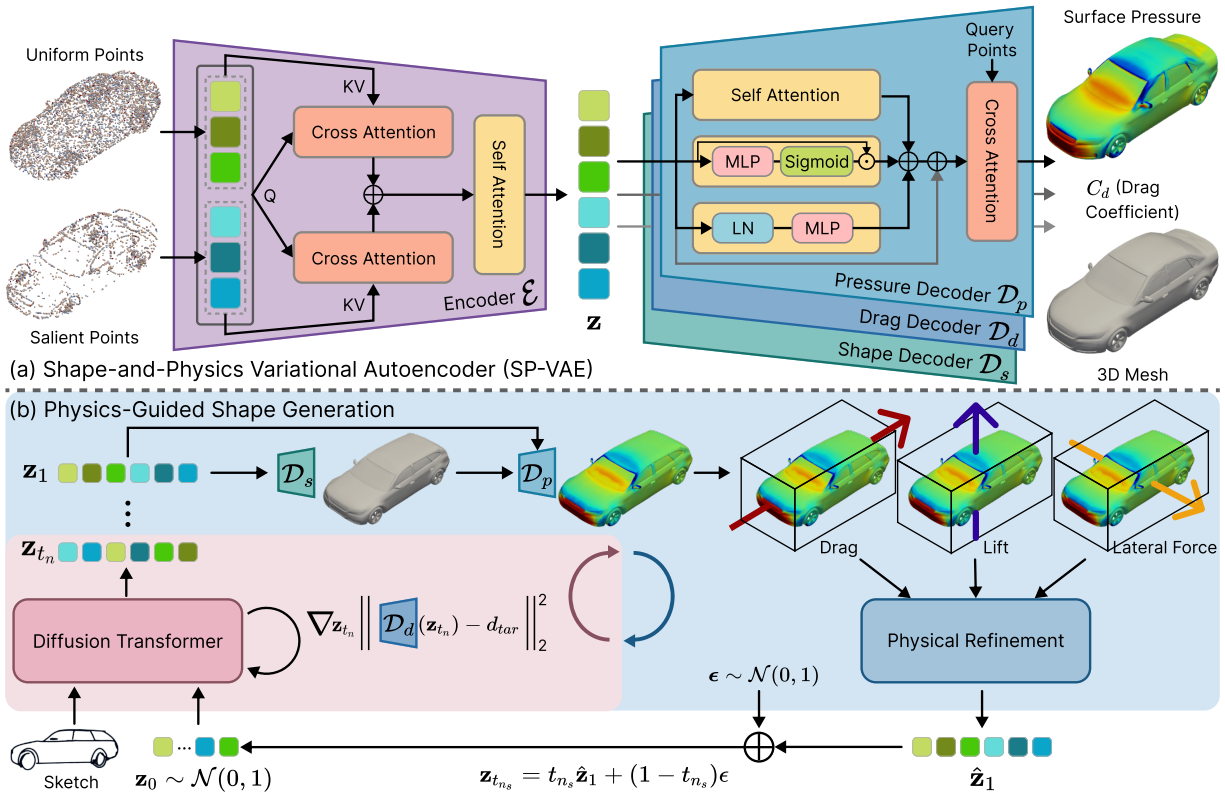


Figure 2. Overview of the proposed framework. (a) The proposed SP-VAE learns a unified latent representation that jointly encodes geometric structure and physical properties. From this shared representation, three decoders reconstruct the 3D shape, surface pressure field, and drag coefficient, respectively. (b) The physics-guided shape generation iteratively bridges flow-matching updates and physical refinements, optionally conditioned on an image, such as a sketch. This alternating strategy updates the latent code to align with the desired 3D shape and physical properties, ensuring both visual plausibility and physical validity.

a 2D image $\mathbf{I}$. In other words, we aim to maximize the posterior distribution $q(\mathcal{S}|A)$ (or $q(\mathcal{S}|\mathbf{I}, A)$ when $\mathbf{I}$ is available). We use a flow matching model [26] to evolve the samples based on physical guidance to ensure that the resulting shapes are plausible from a physics point of view. In this paper, we take A to be aerodynamic properties formulated as $A := \{C_d, \mathcal{P}\}$, where C_d is the dimensionless coefficient of drag and $\mathcal{P} : \mathbb{R}^3 \to \mathbb{R}^+$ is the pressure field measured at surface points. $\mathcal{S}$ is represented by a signed distance function (SDF) $S : \mathbb{R}^3 \to \mathbb{R}$ that evaluates the signed distance of 3D points to the surface.

System Overview. To achieve the mapping from physics information to a 3D shape, we propose a unified framework that incorporates physical awareness into the flow matching generation mechanism, as illustrated in Fig. 2. Specifically, we develop a joint latent space capturing both geometry and physical properties (Sec. 3.1), and a physics-guided flow matching model for 3D shape generation (Sec. 3.2).

3.1. Shape-and-Physics Variational Autoencoder

Recall that we aim to achieve physics-guided 3D shape generation. However, the existing 3D shape VAEs [5, 49] only encode 3D geometry information in the latent space. The lack of physical properties in these representations makes it difficult to generate physically efficient 3D shapes from the latent space. To this end, we develop a Shape-and-Physics Variational Autoencoder (SP-VAE) that jointly encodes shape and physics within a unified latent space.

3.1.1. Shape Encoder and Decoder

We adopt Dora [5] as the baseline architecture for our shape encoder $\mathcal{E}$ and shape decoder $\mathcal{D}_s$. Specifically, uniform surface points $\mathbf{P}_u$ and salient edge points $\mathbf{P}_s$ are first extracted from the mesh as inputs. The encoder applies a dual cross-attention mechanism between $\mathbf{P}_u$ and $\mathbf{P}_s$, aggregates the results via self-attention, and outputs the latent code $\mathbf{z}$. For shape decoding, different from Dora that predicts an occupancy field, we adopt an SDF representation to capture finer geometric details [23]. Several self-attention layers process $\mathbf{z}$, followed by cross-attention with a linear projection that takes query point $\mathbf{x} \in \mathbb{R}^3$ and outputs corresponding SDF value $s = \mathcal{D}_s(\mathbf{x}, \mathbf{z})$. The final 3D mesh is reconstructed via the marching cubes algorithm [27]. Detailed architecture is provided in the supplementary material.

3.1.2. Pressure and Drag Decoders

Pressure Decoder $\mathcal{D}_p$. The pressure decoder predicts a continuous pressure field that allows querying the pressure value $p \in \mathbb{R}$ based on the latent code $\mathbf{z}$ at any 3D spatial point $\mathbf{x} \in \mathbb{R}^3$, formulated as:

$$p = \mathcal{D}_p(\mathbf{x}, \mathbf{z}) \tag{1}$$

As shown in Fig. 2 (top right), given the latent code $\mathbf{z}$, the decoder employs three parallel branches to capture multi-level physical information: a self-attention models global surface dependencies, a channel branch uses a squeeze-excitation [15] mechanism to reweight feature channels, and an MLP refines local representations. The outputs of these branches are fused as $\mathbf{z}_{\text{fused}} = w_1 \cdot \mathbf{z}_{\text{attn}} + w_2 \cdot \mathbf{z}_{\text{channel}} + w_3 \cdot \mathbf{z}_{\text{mlp}}$, where w_1, w_2 and w_3 are learnable weights. Finally, the fused features are decoded through a cross-attention layer that takes 3D spatial point $\mathbf{x}$ as queries and predicts the corresponding pressure value p.

Drag Coefficient Decoder $\mathcal{D}_d$. Unlike the previous decoders that estimate spatial fields, the drag decoder predicts a global drag coefficient $C_d \in \mathbb{R}$ from each latent code. Similar to the pressure decoder, the drag decoder employs the same three-branch feature extraction module, followed by a three-layer MLP that outputs the drag coefficient C_d.

3.1.3. Training Strategy

To achieve stable convergence and promote interaction between geometric and physical information, we adopt a two-stage training strategy consisting of modular pretraining followed by joint fine-tuning.

Stage 1: Independent Training. For the encoder and shape decoder, we initialize from the pretrained weights of Dora [5] and fine-tune them on our dataset. The training objective combines the SDF loss with a Kullback–Leibler (KL) regularization term:

$$\mathcal{L}_{\text{shape}} = \lambda_{\text{sdf}}\mathcal{L}_{\text{sdf}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}, \tag{2}$$

$$\mathcal{L}_{\text{sdf}} = \|s - \hat{s}\|_2^2, \tag{3}$$

where s and $\hat{s}$ denote the predicted and ground-truth SDF values, respectively. With the encoder frozen, the pressure and drag decoders are trained separately from scratch using a composite loss that combines mean absolute error (MAE) and mean squared error (MSE):

$$\mathcal{L}_{\text{press}} = \|p - \hat{p}\|_1 + \|p - \hat{p}\|_2^2, \tag{4}$$

$$\mathcal{L}_{\text{drag}} = \|C_d - \hat{C}_d\|_1 + \|C_d - \hat{C}_d\|_2^2, \tag{5}$$

where $\hat{p}$ and $\hat{C}_d$ represent the ground-truth surface pressure and drag coefficient, respectively.

Stage 2: Joint Fine-Tuning. Finally, all components, including the encoder and three decoders, are jointly optimized under the combined objective:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{shape}}\mathcal{L}_{\text{shape}} + \lambda_{\text{press}}\mathcal{L}_{\text{press}} + \lambda_{\text{drag}}\mathcal{L}_{\text{drag}}, \tag{6}$$

where λ_{shape}, λ_{press}, and λ_{drag} balance the contributions of different tasks. This staged training ensures stable convergence and a physics-informed latent space, forming the basis for physics-guided generation (Sec. 3.2).

3.2. Physics-Guided Shape Generation

Common physics-guided shape generation methods [41, 48] iteratively update the geometry toward target physical objectives. However, without explicit awareness of the underlying shape manifold, these methods struggle to recover once the geometry becomes suboptimal or distorted. In contrast, we integrate the flow-matching 3D shape generation and physics-based guidance into a unified framework, enforcing stable convergence toward geometrically plausible and physically valid results. To achieve this, we propose an alternating generation paradigm, in which we iteratively perform a velocity-based update with physics-based regularization and a physical refinement. Please refer to Alg. 1 for the full algorithm description.

3.2.1. Physics-Regularized Flow Matching

We first train a flow matching model that learns a shape manifold over plausible geometries, which is used for generating high-fidelity 3D shapes, conditionally from images or unconditionally from noise. We adopt rectified flow [26] to formulate the flow matching model, which learns a velocity field that transports noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$ toward data $\mathbf{z}_1$. The forward process is expressed as a linear interpolation to obtain the noisy sample $\mathbf{z}_{t_n}$ in time step $t_n \in [0, 1]$:

$$\mathbf{z}_{t_n} = t_n\,\mathbf{z}_1 + (1 - t_n)\,\boldsymbol{\epsilon}, \tag{7}$$

and the corresponding velocity field is defined as:

$$\mathbf{u}_{t_n} = \frac{d\mathbf{z}_{t_n}}{dt_n} = \mathbf{z}_1 - \boldsymbol{\epsilon}. \tag{8}$$

The model is trained to predict this velocity field from an optional condition $\mathbf{c} = \begin{cases} \mathbf{I}, & \text{if conditional on image} \\ \varnothing, & \text{if unconditional} \end{cases}$. The network architecture is provided in the supplementary material. During the forward update, the reverse step is computed using the predicted velocity $\hat{\mathbf{u}}(\mathbf{z}_{t_n}, t_n, \mathbf{c})$ as:

$$\mathbf{z'}_{t_{n+1}} = \mathbf{z}_{t_n} - (t_{n+1} - t_n)\hat{\mathbf{u}}(\mathbf{z}_{t_n}, t_n, \mathbf{c}). \tag{9}$$

Inspired by classifier guidance [10], we incorporate a physics-based regularization term by using the drag decoder $\mathcal{D}_d$, trained in Sec. 3.1, as a physics-aware estimator during sampling. As shown in Fig. 2 (bottom left), at each time step t_n, $\mathcal{D}_d$ predicts the drag coefficient of $\mathbf{z}_{t_n}$, and the gradient of its deviation from the target d_{tar} guides the update:

$$\mathbf{z}_{t_{n+1}} = \mathbf{z}'_{t_{n+1}} - \lambda_d \nabla_{\mathbf{z}_{t_n}} \left\|\mathcal{D}_d(\mathbf{z}_{t_n}) - d_{\text{tar}}\right\|_2^2. \tag{10}$$

This physics-based regularization softly steers the flow trajectory toward physically plausible regions of the manifold, promoting the generation of physically valid shapes.

Algorithm 1: Physics-Guided Generation

Input: initial noise $\mathbf{z}_0$, shape decoder $\mathcal{D}_s$, drag decoder $\mathcal{D}_d$, pressure decoder $\mathcal{D}_p$, velocity estimator $\hat{\mathbf{u}}(\cdot)$, sampling steps N, refinement steps M, target drag coefficient d_{tar}, weights $\lambda_d, \lambda_x, \lambda_y, \lambda_z$, normals $\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z$, local face area A.

$n_s = 0, t_n \leftarrow \frac{n}{N}$;

for $k = 1$ **to** K **do**

  // Phase 1: Physics-Regularized Flow Matching

  for $n = n_s$ **to** $N-1$ **do**

    $\mathbf{z}'^{k}_{t_{n+1}} \leftarrow \mathbf{z}^k_{t_n} - (t_{n+1} - t_n)\,\hat{\mathbf{u}}(\mathbf{z}^k_{t_n}, t_n)$;

    $\mathbf{z}^k_{t_{n+1}} \leftarrow \mathbf{z}'^{k}_{t_{n+1}} - \lambda_d \nabla_{\mathbf{z}^k_{t_n}} \left\|\mathcal{D}_d(\mathbf{z}^k_{t_n}) - d_{\text{tar}}\right\|_2^2$;

  $p \leftarrow \mathcal{D}_p(\mathbf{z}^k_1)$; *// Pressure Prediction*

  for $m = 1$ **to** M **do** *// Phase 2: Physical Refinement*

    $F_s = \sum_{i=1}^{V} p_i\, \mathbf{n}_{s,i}\, A_i, \quad s \in \{x, y, z\}$;

    $\mathcal{L} \leftarrow \lambda_x \|F_x\|_2 + \lambda_y \|F_y\|_2 + \lambda_z \mathrm{ReLU}(F_z)$;

    $\mathbf{z}^k_{1,m} \leftarrow \mathbf{z}^k_{1,m-1} - \nabla_{\mathbf{z}^k_{1,m-1}} \mathcal{L}$;

  $\hat{\mathbf{z}}^k_1 = \mathbf{z}^k_{1,M}$;

  $n_s = \lfloor 0.75N \rfloor$;

  $\mathbf{z}^{k+1}_{t_{n_s}} = t_{n_s}\hat{\mathbf{z}}^k_1 + (1 - t_{n_s})\epsilon$; *// Re-noise*

Output: $\mathbf{z}^K_1$

3.2.2. Physical Refinement

Given the clean latent $\mathbf{z}^k_1$ from the flow matching model at the k-th iteration, the shape and pressure decoders reconstruct the 3D geometry and its corresponding dense surface pressure. This dense field provides localized physical information that enables fine-grained aerodynamic refinement. We compute directional forces using the surface pressure p, face normals $\mathbf{n}_s$, and face areas A, where $s \in \{x, y, z\}$ denotes drag, lateral, and lift directions, respectively:

$$F_s = \sum_{i=1}^{V} p_i\, \mathbf{n}_{s,i}\, A_i, \qquad s \in \{x, y, z\}. \tag{11}$$

The corresponding physical losses are:

$$\mathcal{L}_x = \|F_x\|_2, \quad \mathcal{L}_y = \|F_y\|_2, \quad \mathcal{L}_z = \mathrm{ReLU}(F_z), \tag{12}$$

$$\mathcal{L} = \lambda_x \mathcal{L}_x + \lambda_y \mathcal{L}_y + \lambda_z \mathcal{L}_z, \tag{13}$$

where $\mathcal{L}_x$ encourages minimal drag, $\mathcal{L}_y$ encourages lateral force symmetry, and $\mathcal{L}_z$ enforces negative lift for traction. The gradients of $\mathcal{L}$ are then backpropagated to the latent code $\mathbf{z}^k_1$, yielding the refined $\hat{\mathbf{z}}^k_1$ for the next iteration.

3.2.3. Alternating Update Strategy

To ensure that generation satisfies both geometric plausibility and physical validity rather than drifting toward either extreme, we adopt an alternating update strategy that couples the velocity-based update in Sec. 3.2.1 with the physical refinement in Sec. 3.2.2. As shown in Alg. 1, at iteration k, the velocity-based update gradually denoises the latent code and encourages lower drag from time step t_{n_s}. Given the sampled latent $\mathbf{z}^k_1$, the physical refinement stage applies direction-aware physical gradients for M steps. The refined latent $\hat{\mathbf{z}}^k_1$ is then re-noised to step t_{n_s}, following Eq.7 to produce $\mathbf{z}^{k+1}_{t_{n_s}}$, which initializes the next velocity-based update phase. After K alternating iterations, the process converges to geometrically and physically plausible shapes.

Table 1. Unified generation vs. post-optimization. "†" denotes a stronger setting (500 steps, learning rate 10^{-3}) than the conservative one (100 steps, 10^{-5}). "O-Acc." denotes overall accuracy.

Model	F-score (0.01)×100↑	CD×1000↓	O-Acc.
Generation w/o phys.	74.03	27.14	60.86
SP-VAE+TripOptimizer [41]	73.93	27.13	60.89
SP-VAE+TripOptimizer† [41]	67.70	32.78	58.75
Ours	**89.65**	**20.99**	**66.48**

4. Experiments

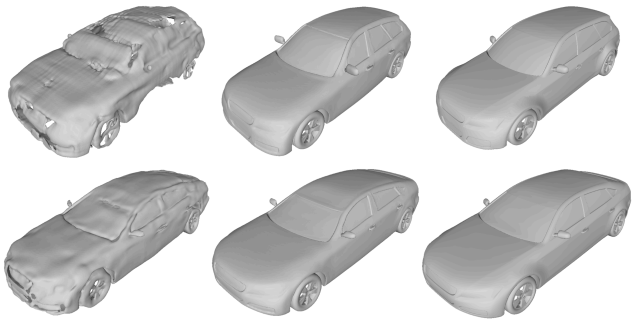
4.1. Experimental Setup

Datasets. We conduct training and evaluation on the DrivAerNet++ [12] dataset, a vehicle aerodynamic benchmark featuring extensive geometric diversity and high-fidelity CFD simulations, including drag coefficients, surface pressure fields, and full 3D flow fields. To evaluate generalization, we test our method on ShapeNet [2] vehicles and further apply it to the structural optimization task [48].

Evaluation Metrics. We evaluate our method on *physics-aware shape generation* and *aerodynamic property estimation*. For shape generation, we use F-score [5], Chamfer Distance (CD) [5], accuracy [5], and Intersection over Union (IoU) [5] to assess geometric fidelity. For the estimation task, we adopt standard regression metrics including Mean Squared Error (MSE) [3], Mean Absolute Error (MAE) [3], Maximum Absolute Error (Max AE) [3], Relative L_2 Error (Rel L2) [3], and Relative L_1 Error (Rel L1) [3]. To evaluate real-world performance, we conduct high-fidelity CFD simulations in `OpenFOAM` [17] to compute the drag coefficient C_d. Detailed definitions are provided in the supplementary material.

4.2. Physics-Guided Shape Generation

Unified Generation vs. Post-Optimization. Starting from a physically imperfect initial shape, TripOptimizer [41] predicts the drag coefficient of the current shape and iteratively updates the shape by minimizing its discrepancy from the target value. This pipeline treats generation and physics-based refinement as two separate stages, whereas our method integrates them into a unified framework. Table 1 evaluates physics-aware generation accuracy by measuring the similarity between generated shapes and the



(a) SP-VAE+TripOptimizer (b) Ours (c) Ground Truth

Figure 3. Qualitative comparison of post-optimization and our unified generation. SP-VAE + TripOptimizer produces distorted shapes and fails to recover them, whereas our alternating method restores plausible surfaces closer to the ground truth.

Table 2. Shape accuracy under target drag coefficient d_{tar}.

Shape	F-score (0.01) $\times 100 \uparrow$	CD $\times 1000 \downarrow$
w/o target d_{tar}	74.03	27.14
w/ target d_{tar}	89.65 (**21.09%** $\uparrow$)	20.99 (**22.68%** $\uparrow$)

ground truth. Since TripOptimizer is not publicly available, we reproduce its two-stage strategy using our SP-VAE. Under a conservative setting (100 steps, learning rate $= 10^{-5}$), the geometry changes only marginally, whereas a stronger setting (500 steps, learning rate $= 10^{-3}$) causes severe deformations and lower accuracy. As shown in Fig. 3, once the shape is distorted, the two-stage method cannot recover it because it lacks awareness of the shape manifold. In contrast, our alternating strategy jointly promotes shape plausibility and physical efficiency, correcting distortions and producing more accurate shapes.

Shape Generation under Target Drag. As shown in Fig. 4, starting from the generated shape from the sketch image without physical information (gray), refining toward the target drag coefficient (blue) aligns the geometry more closely with the ground-truth shape (red), particularly in the rear, roof and vehicle width, where aerodynamic effects are most sensitive. Quantitatively, Table 2 shows that applying the target drag coefficient improves the F-score (0.01) by 21.09% and reduces the Chamfer Distance (CD) by 22.68% compared to the unguided generation, indicating higher geometric accuracy. These results demonstrate that a target drag coefficient provides additional cues that help alleviate the depth ambiguity inherent in 2D-to-3D generation, leading to improved 3D shape.

Single-View Real Image with Physical information. We evaluate our method on a real single-view image using cross-evaluation, as ground-truth meshes are unavailable. As shown in Fig. 5, without physical information, two shapes generated from the same image but different initial noises (red and orange cars) share similar side views yet differ in front-view width due to depth ambiguity. When

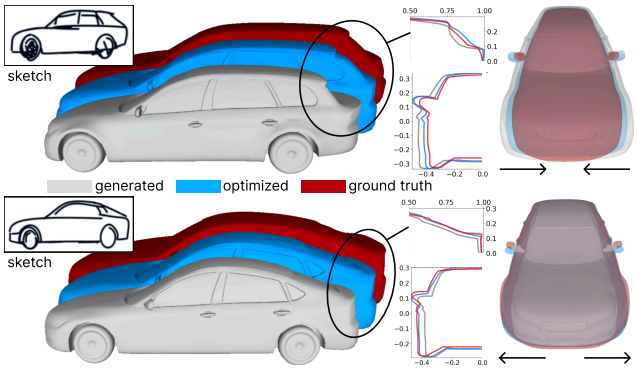


Figure 4. Physical information improves shape generation accuracy. Starting from the physically unguided generation (gray), refining toward the target drag coefficient (blue) aligns the geometry more closely with the ground-truth shape (red).

Table 3. Comparison of shapes from a real image without (w/o phys.) and with (w/ phys.) physical guidance.

Case	Metric	w/o phys.	w/ phys.
Case a	Chamfer Distance ($\times 10^4$) $\downarrow$	20.98	**2.38**
	Normal Consistency $\uparrow$	0.78	**0.90**
	F-score (0.01) $\uparrow$	0.41	**0.85**
Case b	Chamfer Distance ($\times 10^4$) $\downarrow$	8.06	**0.83**
	Normal Consistency $\uparrow$	0.88	**0.96**
	F-score (0.01) $\uparrow$	0.59	**0.98**

Table 4. Comparison on shape reconstruction. "†" indicates fine-tuning on the DrivAerNet++ [11] dataset. "O-", "S-", and "C-" denote overall, sharp, and coarse, respectively.

Model	O-Acc.	O-IoU	S-Acc.	S-IoU	C-Acc.	C-IoU
3DSet2Vector [49]	73.58	51.28	64.61	49.80	83.17	54.32
Hunyuan3D 2.1 [16]	89.43	76.55	87.19	77.57	91.81	74.62
Hi3DGen [47]	91.47	81.52	89.37	82.76	93.67	80.08
Dora [5]	86.49	71.28	84.78	74.16	88.32	66.04
Dora† [5]	95.31	88.61	94.32	89.13	96.37	87.62
Ours	**96.73**	**91.89**	**95.64**	**91.50**	**97.89**	**92.63**

physical guidance, which enforces an approximate target drag coefficient and minimizes directional forces, is applied (blue and green cars), the generated 3D shapes converge to similar front-view widths. Table 3 quantitatively shows reduced inter-shape discrepancy, indicating that physical guidance effectively mitigates depth ambiguity.

4.3. Shape Reconstruction and Physics Estimation

Comparison of Shape Reconstruction. Here, we use our trained encoder and shape decoder in SP-VAE to reconstruct 3D meshes from the input point clouds. The results in Table 4 show that our model outperforms 3DSet2Vector [49], the VAEs of Hunyuan3D 2.1 [16] and Hi3DGen [47], pure Dora [5] model, and its fine-tuned version on the DrivAerNet++ [12] dataset, across all evaluation metrics. The improvements highlight that our physics-informed representa-

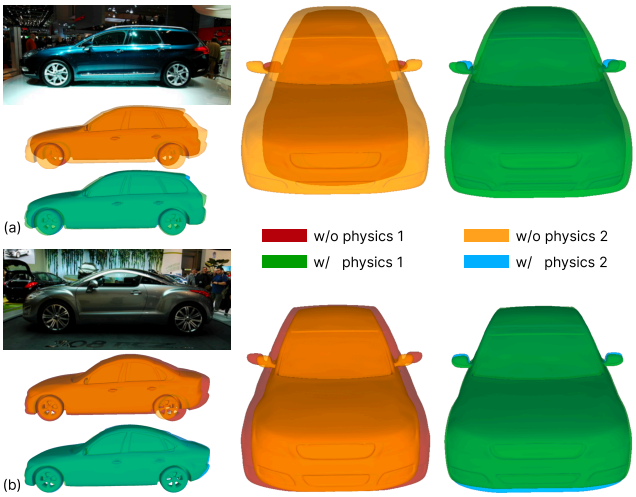


Figure 5. Physical information mitigates depth ambiguity in 3D generation from a real single-view image. Two shapes are first generated from the same image using different initial noises (red and orange cars). When physical guidance is applied, the resulting shapes (blue and green cars) converge to similar front-view widths.

Table 5. Performance comparison on drag coefficient estimation.

Model	MSE↓ ($\times 10^{-5}$)	MAE↓ ($\times 10^{-3}$)	Max AE↓ ($\times 10^{-2}$)
GCNN [18]	17.1	10.43	15.03
RegDGCNN [12]	14.2	9.31	12.79
PointNet [36]	14.9	9.60	12.45
TripNet [3]	9.1	7.17	7.70
Ours	**4.0**	**4.83**	**2.70**

Table 6. Performance comparison on pressure field prediction.

Model	MSE↓ ($\times 10^{-2}$)	MAE↓ ($\times 10^{-1}$)	Rel L2↓ (%)	Rel L1↓ (%)
RegDGCNN [12]	8.29	1.61	27.72	26.21
FigConvNet [8]	<u>4.99</u>	<u>1.22</u>	20.86	21.12
Transolver [43]	7.15	1.41	23.87	22.57
TripNet [3]	5.14	1.25	<u>20.05</u>	20.93
Ours	**4.55**	**1.09**	**20.02**	**17.78**

tions contribute to more high-fidelity shape reconstructions.

Comparison of Physics Estimation. We predict the physical properties from the point cloud, utilizing the pressure decoder and drag decoder. As shown in Tables 5 and 6, our method achieves the best performance in both drag coefficient prediction and surface pressure estimation, whereas the baselines estimate physical quantities solely from shape inputs without leveraging complementary information. These results show that the joint shape-physics latent representation captures the correlation between geometry and aerodynamics, improving both shape reconstruction and physics estimation.

4.4. Ablation Studies

Effectiveness of SP-VAE Training Strategy. As shown in Table 7, joint finetuning consistently improves performance across drag estimation, surface pressure prediction, and shape reconstruction. The most notable gain appears in shape reconstruction, where overall accuracy and IoU improve to 96.73 and 91.89, respectively. These results confirm that joint finetuning fosters mutual reinforcement between geometric and physical representations, leading to a more coherent and expressive latent space.

Effectiveness of Alternating Strategy. We evaluate joint shape quality and surface pressure on the out-of-distribution ShapeNet [2]. As shown in Fig. 6, *physics refinement only* moves the initial imperfect shape toward the physical objectives. However, it introduces geometric artifacts without awareness of the shape manifold. Incorporating *flow-matching updates* restores geometric plausibility, yet the physics objectives remain suboptimal. By iteratively *alternating* between physics refinement and flow-matching updates, the method reconciles physical satisfaction with shape plausibility, achieving both aesthetic quality and aerodynamic performance.

Impact of Drag Coefficient and Full Pressure Field. Fig. 7 compares surface pressure distributions for shapes generated without physical guidance, with drag-only guidance, and with full pressure-field guidance. Without physical guidance, the generated shape exhibits strong pressure over the hood and windshield (red box), unstable flow at the front-roof transition (blue box), and weak low-pressure continuity at the front corner (gray box), resulting in high drag. Drag-only guidance reduces the front pressure peak (red box) and improves flow attachment, but the pressure distribution remains coarse and lacks local smoothness (red and blue boxes). Full pressure-field guidance further suppresses high-pressure regions and yields smoother pressure distributions, thus enhancing aerodynamic performance.

Impact of Physical Decoder Structure. Table 8 analyzes the contribution of the attention, channel, and MLP branches in the proposed physics decoder. Each pair of branches provides complementary benefits, and integrating all three achieves the best performance (4.59 MSE, 1.09 MAE), indicating that multi-dimensional feature fusion effectively enhances physics prediction accuracy.

4.5. Generalization to Structural Optimization

We further extend our physics-guided shape generation framework to structural optimization, where the goal is to minimize compliance under prescribed loads and boundary conditions, following PhysiOpt [48]. As shown in Fig. 8, shapes generated by Hunyuan3D 2.1 [16] often contain thin legs that deform severely under load. PhysiOpt uses a physics simulator for optimization, but without shape-manifold awareness, it often distorts shapes and introduces

Table 7. Ablation study on the training strategy of SP-VAE. "O-", "S-", and "C-" denote overall, sharp, and coarse, respectively.

Model	Drag Estimation			Surface Pressure Estimation				Shape Reconstruction					
	MSE↓ ($\times 10^{-5}$)	MAE↓ ($\times 10^{-3}$)	Max AE↓ ($\times 10^{-2}$)	MSE↓ ($\times 10^{-2}$)	MAE↓ ($\times 10^{-1}$)	Rel L2↓ (%)	Rel L1↓ (%)	O-Acc. (%)	O-IoU (%)	S-Acc. (%)	S-IoU (%)	C-Acc. (%)	C-IoU (%)
Independent Training	4.6	5.14	3.08	4.59	**1.09**	20.12	17.81	95.31	88.61	94.32	89.13	96.37	87.62
Joint Fine-tuning	**4.0**	**4.83**	**2.70**	**4.55**	**1.09**	**20.02**	**17.78**	**96.73**	**91.89**	**95.64**	**91.50**	**97.89**	**92.63**

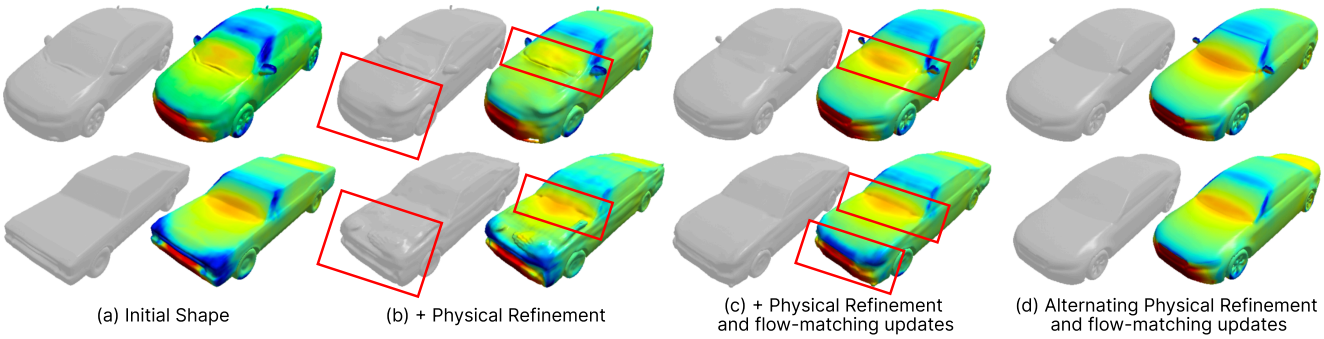


Figure 6. Visualization of generation with physical refinement and flow-matching updates, shown in terms of mesh geometry and surface pressure. Starting from a physically imperfect initialization (a), physical refinement improves physical objectives but introduces distortions (b). Adding flow-matching updates restore geometric plausibility but lead to non-uniform pressure (c). Alternating the two produces refined geometry and more uniform pressure, improving both visual quality and aerodynamic performance (d).

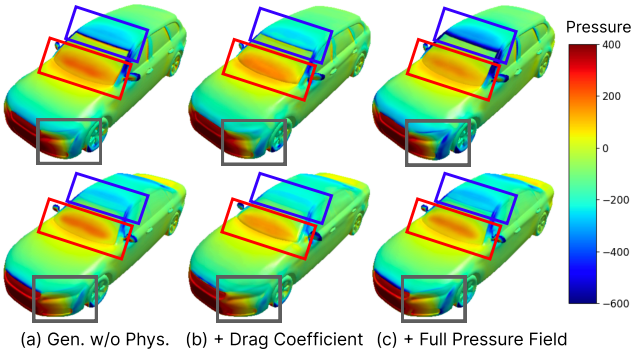


Figure 7. Surface pressure distribution of the initial shape and the generated results under drag coefficient or full pressure field.

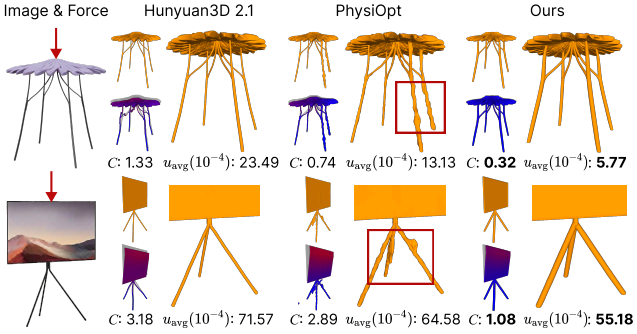


Figure 8. Comparisons for structural optimization. C: compliance (strain energy; lower is stiffer). u_{avg}: mean displacement.

Table 8. Ablation study on the components of physical decoder.

Attn.	Channel	MLP	MSE↓ ($\times 10^{-2}$)	MAE↓ ($\times 10^{-1}$)	Rel L2 ↓ (%)	Rel L1 ↓ (%)
		✓	8.26	1.52	27.44	24.68
	✓		5.43	1.23	22.09	20.07
✓			5.90	1.27	22.84	20.73
✓	✓		5.20	1.21	21.49	19.63
✓		✓	5.16	1.23	21.42	20.04
	✓	✓	5.15	1.21	21.47	19.62
✓	✓	✓	**4.59**	**1.09**	**20.12**	**17.81**

artifacts. In contrast, our method complements physics with shape manifold, improving physical performance and shape quality. See the supplementary material for more details.

5. Conclusion

In this work, we present a flow matching paradigm with explicit physical guidance, including an alternating process between a velocity-based update with physics-aware regularization and a physical refinement. To bridge geometry and physics, we develop SP-VAE, which encodes both in a shared latent space, capturing their correlations and forming the basis for physics-guided generation. Experiments demonstrate that our method produces both physically efficient and aesthetically appealing shapes. It also enhances single-view reconstruction and aerodynamic performance, and generalizes to structural optimization. We hope this work inspires further research on physics-grounded generative modeling for realistic and functional 3D shapes.

Acknowledgements

This work was supported in part by the Swiss National Science Foundation.

References

[1] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2, 5, 7, 3

[3] Qian Chen, Mohamed Elrefaie, Angela Dai, and Faez Ahmed. Tripnet: Learning large-scale high-fidelity 3d car aerodynamics with triplane networks. *arXiv preprint arXiv:2503.17400*, 2025. 5, 7

[4] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *International Conference on Computer Vision*, 2023. 2

[5] Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *Conference on Computer Vision and Pattern Recognition*, 2025. 1, 2, 3, 4, 5, 6

[6] Yongwei Chen, Yushi Lan, Shangchen Zhou, Tengfei Wang, and Xingang Pan. Sar3d: Autoregressive 3d object generation and understanding via multi-scale 3d vqvae. In *Conference on Computer Vision and Pattern Recognition*, 2025. 1

[7] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *Conference on Computer Vision and Pattern Recognition*, 2025. 1

[8] Chris Choy, Alexey Kamenev, Jean Kossaifi, Max Rietmann, Jan Kautz, and Kamyar Azizzadenesheli. Factorized implicit global convolution for automotive computational fluid dynamics prediction. *arXiv preprint arXiv:2502.04317*, 2025. 7

[9] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Conference on Computer Vision and Pattern Recognition*, 2023. 1

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 2021. 4

[11] M. Elrefaie, F. Morar, A. Dai, and F. Ahmed. DrivAerNet++: A Large-Scale Multimodal Car Dataset with Computational Fluid Dynamics Simulations and Deep Learning Benchmarks. In *Advances in Neural Information Processing Systems*, 2024. 6, 1, 3

[12] Mohamed Elrefaie, Angela Dai, and Faez Ahmed. Drivaernet: A parametric car dataset for data-driven aerodynamic design and prediction. *Journal of Mechanical Design*, 147 (4), 2025. 2, 5, 6, 7

[13] Giorgio Giannone, Akash Srivastava, Ole Winther, and Faez Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation. *Advances in Neural Information Processing Systems*, 2023. 2

[14] Minghao Guo, Bohan Wang, Pingchuan Ma, Tianyuan Zhang, Crystal Owens, Chuang Gan, Josh Tenenbaum, Kaiming He, and Wojciech Matusik. Physically compatible 3d object modeling from a single image. *Advances in Neural Information Processing Systems*, 2024. 2

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Conference on Computer Vision and Pattern Recognition*, 2018. 4

[16] Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025. 6, 7, 5

[17] Hrvoje Jasak, Aleksandar Jemcov, and Zeljko Tukovic. OpenFOAM: A C++ Library for Complex Physics Simulations. In *International workshop on coupled methods in numerical dynamics*, 2007. 5, 1, 4

[18] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *Conference on Computer Vision and Pattern Recognition*, 2019. 7

[19] Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025. 1, 2

[20] Keith J Lee, Yijiang Huang, and Caitlin T Mueller. A differentiable structural analysis framework for high-performance design optimization. In *Structures*, 2025. 2

[21] Weiyu Li, Rui Chen, Xuelin Chen, and Ping Tan. Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. In *International Conference on Learning Representations*, 2024. 2

[22] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native diffusion and interactive geometry refiner. In *Conference on Computer Vision and Pattern Recognition*, 2025. 1, 2

[23] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 3, 2

[24] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution

text-to-3d content creation. In *Conference on Computer Vision and Pattern Recognition*, 2023. 2

[25] Fangfu Liu, Diankun Wu, Yi Wei, Yongming Rao, and Yueqi Duan. Sherpa3d: Boosting high-fidelity text-to-3d generation via coarse 3d prior. In *Conference on Computer Vision and Pattern Recognition*, 2024. 2

[26] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. 3, 4, 1

[27] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. 3, 2

[28] Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3d creation on pose-free videos at scale. In *Conference on Computer Vision and Pattern Recognition*, 2025. 1

[29] François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization. In *AAAI Conference on Artificial Intelligence*, 2023. 2

[30] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024. 1

[31] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003. 1

[32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 2

[33] Rafael Pastrana, Deniz Oktay, Ryan P Adams, and Sigrid Adriaenssens. Jax fdm: A differentiable solver for inverse form-finding. *arXiv preprint arXiv:2307.12407*, 2023. 2

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision*, 2023. 2

[35] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2017. 7

[37] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *Conference on Computer Vision and Pattern Recognition*, 2024. 2

[38] Lyle Regenwetter, Amin Heyrani Nobari, and Faez Ahmed. Deep generative models in engineering design: A review. *Journal of Mechanical Design*, 2022. 2

[39] Binyang Song, Rui Zhou, and Faez Ahmed. Multi-modal machine learning in engineering design: A review and future directions. *Journal of Computing and Information Science in Engineering*, 2024. 2

[40] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *International Conference on Learning Representations*, 2024. 2

[41] Parsa Vatani, Mohamed Elrefaie, Farhad Nazarpour, and Faez Ahmed. Tripoptimizer: Generative three-dimensional shape optimization and drag prediction using triplane variational autoencoder networks. *Physics of Fluids*, 37(12), 2025. 1, 2, 4, 5

[42] Gaoyuan Wu. A framework for structural shape optimization based on automatic differentiation, the adjoint method and accelerated linear algebra. *Structural and Multidisciplinary Optimization*, 2023. 2

[43] Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. In *International Conference on Machine Learning*, 2024. 7

[44] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *Advances in Neural Information Processing Systems*, 2024. 2

[45] Tianju Xue, Shuheng Liao, Zhengtao Gan, Chanwook Park, Xiaoyu Xie, Wing Kam Liu, and Jian Cao. Jax-fem: A differentiable gpu-accelerated 3d finite element solver for automatic inverse design and mechanistic data science. *Computer Physics Communications*, 2023. 2

[46] Shamim Yazdani, Akansha Singh, Nripsuta Saxena, Zichong Wang, Avash Palikhe, Deng Pan, Umapada Pal, Jie Yang, and Wenbin Zhang. Generative ai in depth: A survey of recent advances, model variants, and real-world applications. *Journal of Big Data*, 2025. 1

[47] Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. In *International Conference on Computer Vision*, 2025. 6

[48] Xiao Zhan, Clément Jambon, Evan Thompson, Kenney Ng, and Mina Konaković Luković. Physiopt: Physics-driven shape optimization for 3d generative models. In *SIGGRAPH Asia*, 2025. 2, 4, 5, 7

[49] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics*, 42(4), 2023. 1, 2, 3, 6

[50] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics*, 2024. 2

[51] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems*, 2023. 2

PhysGen: Physically Grounded 3D Shape Generation for Industrial Design

Supplementary Material

This supplemental material includes the following sections:

(A) Implementation details.
(B) Additional experiments.
(C) Network architectures.
(D) Dataset details.
(E) Evaluation metrics.
(F) CFD simulation setup in OpenFOAM [17].
(G) Generalization to structural optimization.

A. Implementation Details

A.1. SP-VAE

For each 3D mesh used for training, we extract $32,768$ uniform surface points $\mathbf{P}_u$ and $32,768$ salient edge points $\mathbf{P}_s$ using the Sharp Edge Sampling (SES) strategy [5]. As shown in Fig. A, queries $\mathbf{Q}$ for cross-attention are constructed by applying Farthest Point Sampling (FPS) [31] to $\mathbf{P}_u$ and $\mathbf{P}_s$, each downsampled to 1024 points and concatenated into a 2048-point set. We supervise the shape encoder-decoder using both uniformly sampled coarse points in the bounding box and sharp points perturbed around the ground-truth surface, with loss weights $\lambda_{\text{SDF}} = 1$ and $\lambda_{\text{KL}} = 0.001$. The shape encoder-decoder is trained for 1000 epochs on $4\times$H100 GPUs (about 2 days). The pressure decoder is trained using pressure values sampled near the surface for 1500 epochs on $4\times$H100 GPUs (about 21 hours). The drag decoder, which predicts a single global drag coefficient, is trained for 1500 epochs and completes within one day on a single H100 GPU. After individual training, we perform joint fine-tuning for 500 epochs on $4\times$H100 GPUs (about 15 hours) using a combined loss with $\lambda_{\text{shape}} = 10$, $\lambda_{\text{physics}} = 0.1$, and $\lambda_{\text{drag}} = 10$. All experiments use 5819 training samples and 1147 test samples from DrivAerNet++ [11].

A.2. Physics-Guided Shape Generation

For flow-based generation, we adopt the rectified flow formulation [26], using 100 sampling steps at inference. In physics-guided shape generation, we incorporate a physics-based regularization term with weight $\lambda_d = 0.03$ for drag guidance during velocity-based updates, while directional weights $\lambda_x = 0.2$, $\lambda_y = 0.1$, $\lambda_z = 0.1$ are applied during the physical refinement phase. For alternating generation, we perform $K = 20$ alternating iterations. In each iteration k, we first apply 20 steps of physical refinement to obtain the refined latent $\hat{\mathbf{z}}_1^k$, then re-noise it back to timestep $t_{n_s} = 0.75$ to produce $\mathbf{z}_{t_{n_s}}^{k+1}$, which initializes the next velocity-based update phase. The full set of iterations takes roughly 210 seconds. Overall, the procedure iteratively alternates between velocity-based updates and physics-based refinement, with each stage performing only a small number of steps (25 steps for velocity updates and 20 steps for physical refinement), gradually converging toward shapes that satisfy both geometric plausibility and physical efficiency.

Table A. Shape generation toward minimizing the drag coefficient. Image-unconditional generation (Unc.) minimizes drag without image, while conditional generation (Cond.) minimizes drag with image conditioning. Average drag coefficients simulated by OpenFOAM indicate aerodynamic performance. (SN: ShapeNet, DAN+: DrivAerNet++.)

Shape	**Average Drag Coefficient**		
	SN (Unc.)	**DAN+** (Unc.)	**DAN+** (Cond.)
w/o minimizing	0.393	0.324	0.334
w/ minimizing	0.304	0.274	0.312
Improvement	**22.70%** ↑	**15.47%** ↑	**6.53%** ↑

B. Additional Experiments

Shape Generation toward Minimizing Drag. Beyond leveraging a known target drag coefficient to improve reconstruction accuracy, our framework can also enhance aerodynamic performance by minimizing the drag coefficient. Table A reports average results over 20 samples per dataset, each simulated using OpenFOAM [17] (see Sec. F for simulation details), covering both the in-distribution DrivAerNet++ [11] dataset and the out-of-distribution ShapeNet [2] car set. Despite never observing ShapeNet geometries during training, our method achieves a substantial 22.7% drag reduction, demonstrating the generalization and the ability to maintain shape plausibility while improving aerodynamic performance. On DrivAerNet++, unconditional physics-guided generation (minimizing drag without image conditioning) reduces drag by 15.47%, whereas the conditional setting (minimizing drag with image conditioning) achieves a 6.53% reduction, as it balances aesthetic alignment with physical efficiency. These findings confirm that alternating prior- and physics-guided generation generalizes robustly to unseen geometries and improves aerodynamic performance while maintaining visual alignment when a conditional image is provided.

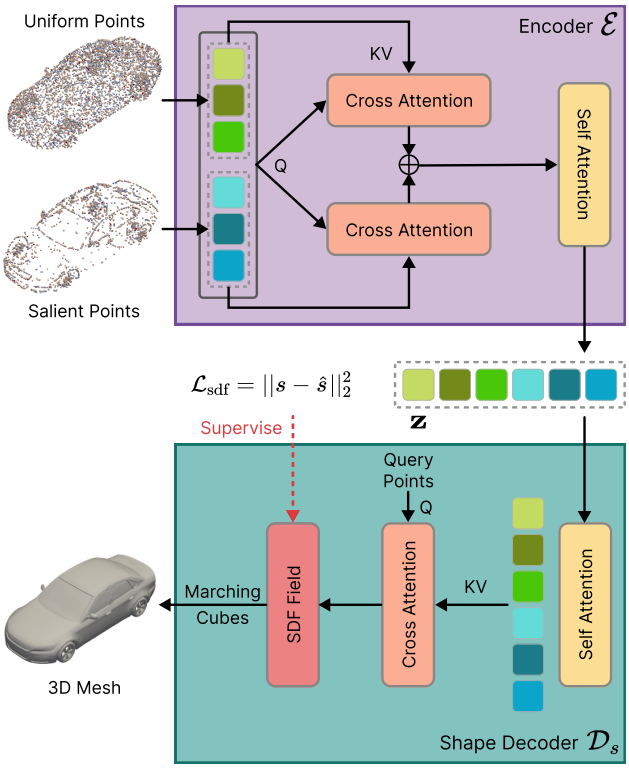


Figure A. Overview of the SP-VAE shape encoder-decoder. The encoder fuses uniform and salient surface points via bidirectional cross-attention and self-attention to produce a latent code. The decoder predicts an SDF field from query points using cross-attention and reconstructs the mesh via marching cubes.

C. Network Architectures

C.1. Shape Encoder and Decoder

We build our shape encoder-decoder architecture upon Dora [5], while extending it to support SDF prediction [23], enabling finer geometric reconstruction than the original occupancy-based representation. As illustrated in Fig. A, the mesh is first extracted into two complementary point sets: (1) uniformly sampled surface points $\mathbf{P}_u$, which capture global coverage of the geometry, and (2) salient edge points $\mathbf{P}_s$, which preserve high-curvature and structurally important regions. These two sets provide separate geometric cues to the encoder. The encoder fuses them via bidirectional cross-attention, letting salient regions inform uniform samples and vice versa. The fused features then pass through self-attention layers to produce the latent code $\mathbf{z}$, capturing both coarse structure and fine details. On the decoding side, we deviate from Dora's occupancy-based formulation and instead predict an SDF field to better preserve high-frequency geometry. The latent code $\mathbf{z}$ is first enriched through several self-attention layers, and a set of 3D query points $\mathbf{x} \in \mathbb{R}^3$ is fed into a linear projection to form the attention queries. Through cross-attention between $\mathbf{x}$ and the latent features, the decoder estimates the corresponding signed distance value $s = \mathcal{D}_s(\mathbf{x}, \mathbf{z})$, effectively conditioning the local geometry on the global shape embedding. The predicted SDF field is then supervised with ground-truth distances sampled around the mesh, and the final mesh is extracted using the marching cubes algorithm [27], yielding a high-quality reconstruction faithful to both global shape and local geometric details.

C.2. Diffusion Transformer Network

We employ a Diffusion Transformer (DiT) [34] within our flow-matching framework to parameterize the velocity field that transports noisy latent codes toward clean representations. As shown in Fig. B, optional conditioning is introduced at the beginning of the network, where $\mathbf{c} = \begin{cases} \mathbf{I}, & \text{if conditional on image,} \\ \varnothing, & \text{if unconditional.} \end{cases}$ For image-conditioned generation, the input image is encoded by DINOv2 [32], and the resulting feature tokens are embedded into a separate conditioning sequence that is injected into every DiT block via cross-attention. The noised latent $\mathbf{z}_{t_n}$ is mapped through a linear projection, while the timestep t_n is encoded using a sinusoidal timestep embedder followed by an MLP. These two embeddings are concatenated to form the token sequence. Each DiT block adopts a pre-norm structure with residual connections and consists of

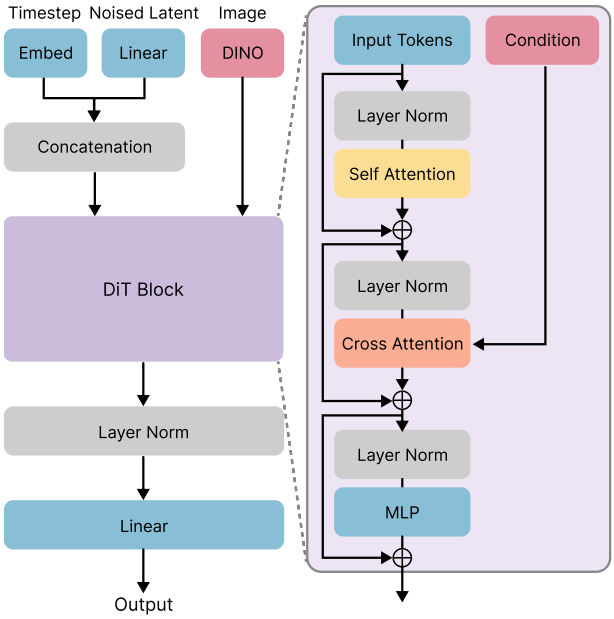


Figure B. Diffusion Transformer (DiT) architecture. Noised latent and timestep embeddings form the input token sequence, while optional DINO-based conditioning is injected via cross-attention in each block. Each DiT block applies self-attention, cross-attention, and an MLP to produce the final velocity prediction.

self-attention over latent tokens, cross-attention with the conditioning tokens $\mathbf{c}$, and a feed-forward network. After all blocks, the final tokens are normalized and projected to produce the velocity field $\hat{\mathbf{u}}(\mathbf{z}_{t_n}, t_n, \mathbf{c})$ required by the flow-matching solver.

D. Dataset details

DrivAerNet++ [11] is a large-scale aerodynamic design dataset comprising $8,000$ high-quality vehicle geometries, each accompanied by high-fidelity CFD simulations, including aerodynamic quantities such as drag coefficients, and both surface pressure and volumetric flow fields. It spans a wide range of automotive body styles, including fastback, notchback, and estateback, and features variations in underbody structure and wheel configurations. Our SP-VAE and flow-based generator are trained on this dataset.

ShapeNet [2] car split is used to evaluate the generalization ability of our method, as it contains vehicle geometries that are not present in the training set. All shapes are uniformly rescaled to match the scale of DrivAerNet++. Although these meshes are physically imperfect yet geometrically reasonable, we use them as initial shapes for our *Physics-Guided Shape Generation* pipeline. By optimizing from these initializations, our method refines the designs into physically efficient and aesthetically pleasing 3D shapes.

E. Evaluation Metrics

For shape generation quality, we evaluate geometric fidelity using F-score, Chamfer Distance, Accuracy, and IoU.

E.1. Shape Generation

F-score. F-score ($\tau = 0.01$) measures consistency between predicted mesh vertices M and ground-truth vertices G, with threshold $\tau = 0.01$:

$$\text{F-score}(\tau) = 2 \cdot \frac{\text{Precision}(\tau) \cdot \text{Recall}(\tau)}{\text{Precision}(\tau) + \text{Recall}(\tau)}, \tag{A}$$

where

$$\begin{aligned} \text{Precision}(\tau) &= \frac{|\{\, m \in M \mid d(m, G) < \tau \,\}|}{|P|}, \\ \text{Recall}(\tau) &= \frac{|\{\, g \in G \mid d(g, M) < \tau \,\}|}{|G|}. \end{aligned} \tag{B}$$

Here, $d(m, G) = \min_{g \in G} \|m - g\|_2$ denotes the nearest-neighbor distance from point m to the ground-truth surface G. Thus, $d(m, G) < \tau$ indicates that the point m lies within a tolerance τ of the target surface.

Chamfer Distance (CD). CD measures the geometric discrepancy between the predicted point set M and the ground-truth point set G. We use the bidirectional form, defined as:

$$\begin{aligned} \text{CD}(M, G) = &\frac{1}{|M|} \sum_{m \in M} \min_{g \in G} \|m - g\|_2^2 \\ &+ \frac{1}{|G|} \sum_{g \in G} \min_{m \in M} \|g - m\|_2^2. \end{aligned} \tag{C}$$

Accuracy (Coarse, Sharp, Overall). Classification accuracy evaluates how well the predicted SDF-based inside/outside labels match the ground truth at sampled query points. For a point $\mathbf{x}$, the predicted label is $y = \mathbf{1}[\, s(\mathbf{x}) \leq 0\,]$, where $s(\mathbf{x})$ is the predicted SDF. The ground-truth label is $\hat{y} = \mathbf{1}[\, \hat{s}(\mathbf{x}) \leq 0\,]$, where $\hat{s}(\mathbf{x})$ is the ground truth SDF. For each sampling split $k \in \{\text{coarse}, \text{sharp}, \text{overall}\}$:

$$\text{Acc}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{1}[y_i = \hat{y}_i]. \tag{D}$$

where coarse points are uniformly sampled within the bounding box, sharp points are generated by perturbing points around the ground-truth surface, and overall points are the union of the two.

Intersection over Union (IoU). Using the same binary inside–outside labels, IoU quantifies how well the predicted inside region overlaps with the ground-truth inside region. It is computed as the ratio between the number of points correctly classified as inside (intersection) and the number of points labeled as inside by either the prediction or the ground truth (union). A higher IoU indicates closer agreement between the predicted and true shape boundaries. For a given split $k \in \{\text{coarse}, \text{sharp}, \text{overall}\}$, IoU is computed as:

$$\text{IoU}_k = \frac{\sum_{i=1}^{N_k} \hat{y}_i \, y_i}{\sum_{i=1}^{N_k} \mathbf{1}[\hat{y}_i + y_i > 0] + \varepsilon}. \tag{E}$$

E.2. Physical Estimation

For the estimation task, we adopt standard regression metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), Maximum Absolute Error (Max AE), Relative L_2 Error (Rel L2), and Relative L_1 Error (Rel L1). For each sampled point i, let p_i denote the predicted pressure and $\hat{p}_i$ the ground-truth pressure, with N being the total number of evaluated points.

Mean Squared Error (MSE). MSE measures the average squared deviation between predicted and true pressures, emphasizing larger errors more strongly:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (p_i - \hat{p}_i)^2. \tag{F}$$

Mean Absolute Error (MAE). MAE computes the average absolute difference between predicted and ground-truth

pressures, providing a more outlier-robust accuracy measure:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|p_i - \hat{p}_i|. \tag{G}$$

Maximum Absolute Error (Max AE). Max AE Quantifies the worst-case prediction error by identifying the largest absolute pressure deviation across all points:

$$\text{Max AE} = \max_{1\le i\le N}|p_i - \hat{p}_i|. \tag{H}$$

Relative $\mathbf{L_2}$ Error (Rel L2). Rel L2 evaluates the global Euclidean discrepancy normalized by the magnitude of the ground-truth pressure field:

$$\text{Rel L2} = \frac{\|p - \hat{p}\|_2}{\|\hat{p}\|_2}, \tag{I}$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm over all points.

Relative $\mathbf{L_1}$ Error (Rel L1). Rel L1 measures the normalized sum of absolute pressure errors relative to the total absolute ground-truth pressure:

$$\text{Rel L1} = \frac{\|p - \hat{p}\|_1}{\|\hat{p}\|_1}, \tag{J}$$

where $\|\cdot\|_1$ is the ℓ_1 norm, equal to the sum of absolute values over all points.

F. CFD Simulation Setup in OpenFOAM

To evaluate the aerodynamic performance of our generated vehicle geometries, as shown in Table A, we perform high-fidelity CFD simulations using OpenFOAM [17] to compute the drag coefficient, surface pressure distributions, and airflow velocity fields. A uniform inlet freestream velocity of 30 m/s, aligned with the vehicle's longitudinal axis and directed toward the frontal surface, is prescribed to represent standard automotive aerodynamic operating conditions. We employ the steady-state simpleFoam [17] solver together with the k-ω SST turbulence model. Each simulation is run on 32 CPU cores for about 8 hours and proceeds through 2500 iterations to ensure convergence. The final aerodynamic drag coefficient is obtained by averaging the flow fields across the last 500 iterations.

G. Generalization to Structural Optimization

We further evaluate our physics-guided shape generation framework on *structural optimization* by following the setting of PhysiOpt [48]. Given external loads $\mathbf{f}$ and user-specified boundary conditions, we map the latent code $\mathbf{z}$ to an SDF representation, convert it into density-weighted finite elements, and then solve the linear static equilibrium equation:

$$\mathbf{K}(\mathbf{z})\,\mathbf{u} = \mathbf{f}, \tag{K}$$

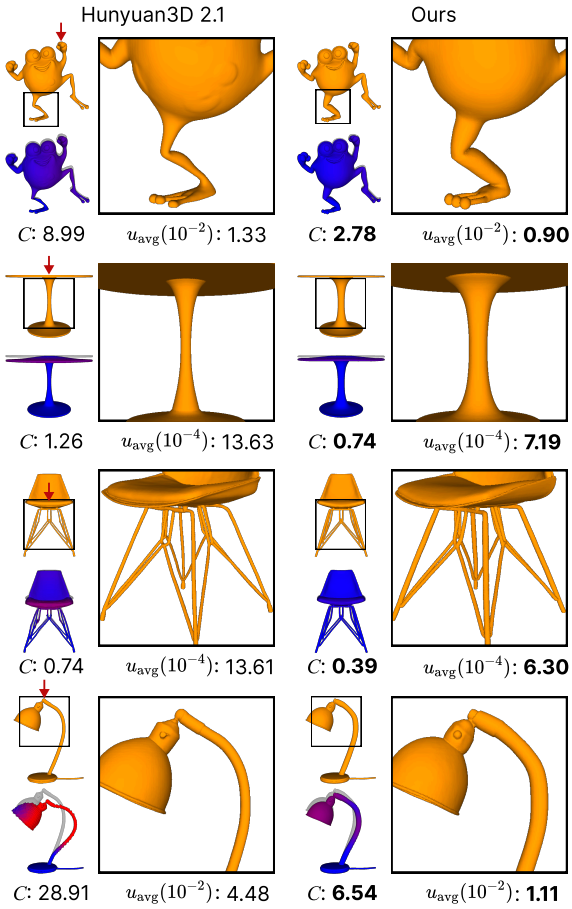


Figure C. Additional results on structural optimization.

where $\mathbf{K}$ is the stiffness matrix and $\mathbf{u}$ denotes the displacement field. As in PhysiOpt, the optimization objective is to reduce the compliance:

$$C = \mathbf{f}^\top \mathbf{u}, \tag{L}$$

which measures the structural deformation, or equivalently the strain energy, under the applied load. Lower compliance indicates a stiffer structure. We also report the average displacement:

$$u_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{u}_i\|_2, \tag{M}$$

where $\mathbf{u}_i$ is the displacement of the i-th FEM node. This formulation follows PhysiOpt's differentiable FEM pipeline, which optimizes shapes directly in latent space under prescribed loads and boundary conditions.

In this task, both PhysiOpt and our method are optimized for 180 steps. Specifically, we perform one velocity-based update after every 5 steps of physical refinement, so that shape plausibility and structural performance are improved jointly throughout the optimization. Fig. C presents additional structural optimization results of our method, where

Hunyuan3D 2.1 [16] generates initial 3D shapes from images without physical awareness. It can be seen that our method improves physical performance while preserving shape quality, yielding structurally stronger and visually appealing shapes.