AEROGEN: PARAMETRIC 3D AIRCRAFT GENERATION FROM STRUCTURED DESIGN SPECIFICATIONS

Anonymous authors
Paper under double-blind review

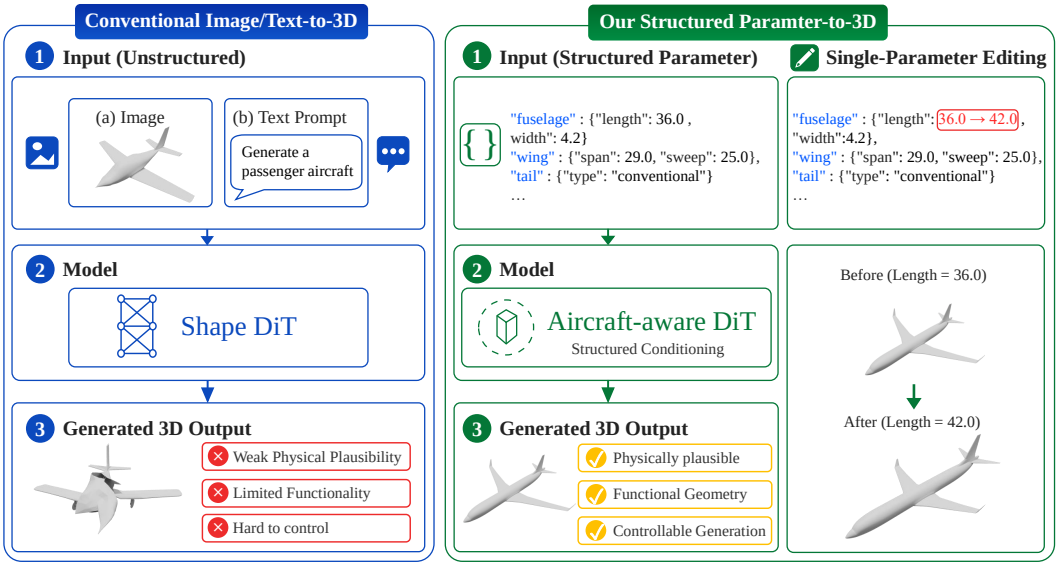

Figure 1: Unlike conventional image- or text-conditioned 3D generation methods, we represent aircraft designs using structured parameters. Image-conditioned approaches often suffer from inconsistent generation quality due to variations in viewpoint, lighting, color and other factors, while text prompts provide coarse semantic guidance. In contrast, our structured Parameter-conditioned framework explicitly parameterizes aircraft structures, imposing stronger geometric constraints and functional consistency, and enabling controllable single-parameter editing.

ABSTRACT

Recent advances in generative models have created new opportunities for design automation in graphics, manufacturing, robotics, and games. However, conceptual designs of aircraft require physical plausibility, functional constraints, and controllable edibility. Such precision remains a bottleneck for common text- or image-conditioned generative models. Motivated by a standard aircraft design workflow, we therefore adopt structured parameters as a complete and interpretable conditioning interface to describe component hierarchies, topological structures, and continuous geometric parameters. We propose a parameter-conditioned aircraft generation framework, where the Parameter Extractor encodes variable-length hierarchical design parameters into conditioning tokens for Diffusion Transformer, while an Attribute Predictor improves the model's sensitivity to aircraft-level parameters by recovering them from generated geometry. Experiments on our built benchmark with 40,348 samples across 74 classes show that our method outperforms image-conditioned baselines in geometric fidelity and shows promising schema-consistent and parameter-sensitive editing.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

In recent years, breakthrough advances in generative artificial intelligence have been driving a paradigm shift across fields. In particular, the maturation of native 3D neural representations Mildenhall et al. (2021); Barron et al. (2021); Kerbl et al. (2023); Yu et al. (2024) and diffusion models Peebles & Xie (2023); Che et al. (2024); Wu et al. (2024) has made it increasingly feasible to translate abstract human engineering semantics directly into high-fidelity 3D geometric entities.

However, commonly used conditioning signals for 3D generation remain limited. Image conditions are sensitive to viewpoint, resolution, lighting, and occlusion, making it difficult to provide stable and accurate guidance, while text conditions often suffer from semantic ambiguity. As illustrated in Fig.1, these issues are particularly critical in aircraft design, where generated geometries should satisfy physical plausibility and functional constraints, and parameter edits should induce controllable and consistent shape changes. Moreover, in aircraft design, the lack of specific aircraft images further makes image-conditioned generation less feasible.

Against this backdrop, we view a Parameter-conditioned aircraft generator as a viable technical pathway to produce editable aircraft. Owing to its hierarchical format, the parameter representation can naturally encode both the tree-like topological structure of aircraft, including fuselage–wing dependencies, and continuous geometric parameters, such as wingspan and sweep angle. Our goal is not to pursue general-purpose state-of-the-art 3D generation, but to study whether structured Parameter specifications provide a more suitable conditioning interface than images or texts for aircraft generation as an engineering-oriented design task.

In this paper, we propose a parameter-conditioned aircraft generator. We introduce structured parameters as conditioning inputs for Diffusion Transformers (DiTs). By encoding parameters including fuselage length, width, and angular attributes, our method enables geometrically accurate 3D aircraft modeling. To support geometric consistency, topological validity and parameter compliance, we further design an Attribute Predictor that recovers selected geometric attributes from the 3D latent space, thereby providing auxiliary supervision for the Parameter Extractor during training.

To validate our model, we construct a large-scale dataset of paired Parameter–OBJ files and meshes for training, comprising 40,348 aircraft assets. Then we first show that the proposed framework outperforms classic image-based models Team (2025); Xiang et al. (2025a) in terms of geometric fidelity and global structural coherence. Furthermore, we perform qualitative analyses under single-parameter perturbations to demonstrate the combination of the Parameter Extractor and Attribute Predictor can reliably steer the generated geometry according to the specified parameter changes, as shown in Fig. 1.

Contributions. This work introduces a Parameter-based framework for 3D generation and demonstrates its application to real-world aircraft design problems:

- Our method introduces structured parameters as conditioning inputs for 3D generation and specifically applies them to aircraft generation.
- The Parameter Extractor supports inference in variable-length parameter spaces. With the Attribute Predictor, it also enhances controllable single-parameter editing.
- We construct and release a Parameter–OBJ aircraft dataset, providing paired structured design specifications and 3D geometries for evaluating the proposed framework.

2 RELATED WORK

3D Generation. 3D representation and latent-space design are central challenges in 3D generation.DiTs have emerged as scalable generative backbones for 3D generation. Representative systems, including Hunyuan3D-2.1 Team (2025), Trellis Xiang et al. (2025b), and Trellis.2 Xiang et al. (2025a), combine DiT-based or flow-matching models with learned 3D latent representations, achieving strong performance in shape generation, reconstruction, and inference efficiency.

Different from image-conditioned generation, we introduce Parameter Extractor, a structured condition extractor that replaces DINOv3-based Siméoni et al. (2025) image guidance in the DiT back-

Table 1: Comparison of aircraft datasets with AeroJSON-40K. PC denotes point cloud.

Dataset	Quantity	Aircraft Scope	Classes	Modalities
ShapeNet Aircraft Edwards et al. (2021)	4,045	General aircraft	11	Img, mesh
AircraftVerse Cobb et al. (2023)	27,714	UAV	unknown	Param., CAD, mesh, PC
BlendedNet Sung et al. (2025)	999	BWB	1	Param., mesh
AeroJSON-40K	40,348	Manned aircraft + UAV	74	Img, Param., mesh, PC

bones and enables controllable generation of 3D aircraft models with specified geometric attributes and topological features.

Aircraft design. Recent applications of machine learning to engineering design have primarily addressed benchmark development and localized optimization. Within aircraft design, prior work has mostly focused on constructing evaluation datasets Cobb et al. (2023); Liu et al. (2024); Diniz & Fuge (2026); Bonnet et al. (2022) or on predicting and generating isolated components, such as airfoils Wang et al. (2023); Jing et al. (2022); Wang et al. (2025). However, extending these methods to complete aircraft remains challenging, as full-aircraft generation requires coordinated control over complex design parameters and variable topological structures. Only a limited number of recent studies Ghiglino et al. (2026) have begun to explore this problem.

To overcome this difficulty, this paper introduces structured parameters encoding as a highly structured conditional input to accurately guide the generation of aircraft. Unlike previous works that rely on unstructured prompts Wu et al. (2021); Xu et al. (2022); Khan et al. (2024), our approach achieves fine-grained control over the overall aircraft architecture by parsing the parametric features encapsulated within the structured data.

3 Dataset and Data Preparation

3.1 Dataset Construction

As shown in Table 1, we construct AeroJSON-40K of 40,348 aircraft parameter–mesh pairs across 74 categories, where multiple design parameters vary jointly within each category. Compared with prior datasets focusing on airfoils Liu et al. (2024), generic shapes Edwards et al. (2021), UAV designs Cobb et al. (2023), or BWB aerodynamics Sung et al. (2025), our dataset emphasizes diverse full-aircraft geometries with more modalities, including structured parameters, meshes, and point clouds, enabling a benchmark for controllable 3D aircraft generation and evaluation.

To evaluate the effectiveness of the proposed framework, we divide the dataset into two subsets: TypePerturb with 12,839 samples and MainWingCtrl with 27,509 samples. TypePerturb simulates typical aircraft design variations by applying 5%–10% perturbations to all parameters of baseline aircrafts selected from the 74 classes. MainWingCtrl is designed to better assess the supervisory effect of Attribute Predictor, where four main-wing parameters, including span, sweep, dihedral angle and chords are independently varied over a larger range. Each subset is split into training and test sets, with 80% of the data used for training and 20% for testing.

3.2 Data Preparation

Meshes. For the raw meshes, we adopt the data processing pipeline based on the setup proposed in Trellis.2 Xiang et al. (2025a). The meshes are processed by VAE encoders and extracted as sparse structure latent and shape latent. This procedure projects all aircraft samples into a canonical coordinate space, effectively eliminating scale disparities across diverse samples while providing stable supervision signals for the subsequent latent generation phase.

JSON files. All the parameters of the aircraft are saved as JSON files, which are exported from OpenVSP. Each aircraft corresponds to a structured JSON file containing a list of geometric components. Each component is defined by its name, semantic type, and continuous design parameters,

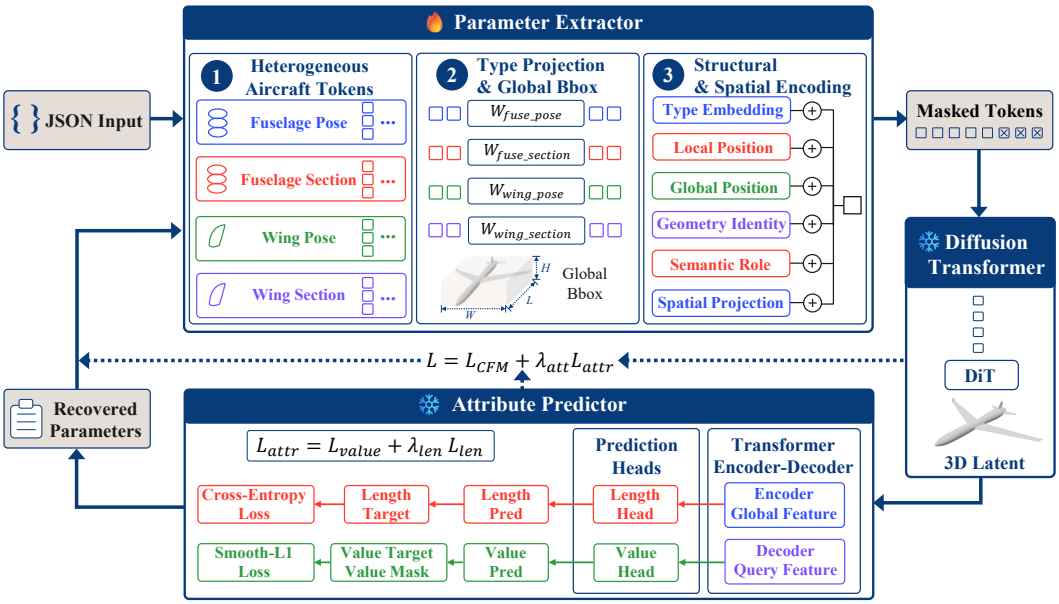

Figure 2: Overview of the proposed Parameter-conditioned aircraft generation framework. Parameter Extractor transforms structured aircraft specifications into masked conditioning tokens via heterogeneous tokenization, type-specific projection, and structural–spatial encoding. During training, an auxiliary Attribute Predictor recovers selected aircraft parameters from the generated 3D latent space, providing value- and length-level supervision for improved parameter controllability.

providing an explicit description of both aircraft topology and geometry. Details of the JSON structure are provided in the Appendix A.

4 METHOD

We formulate Parameter-conditioned aircraft generation as follows. Given structured parameters representation $\mathcal{J}$ describing aircraft components, the goal is to generate a 3D aircraft model $\mathcal{O}$ that is geometrically faithful to the specified design. Unlike image- or text-conditioned methods, our framework uses sturctured parameters as explicit control signals. As shown in Fig. 2, we encode $\mathcal{J}$ into a variable-length sequence of conditioning tokens with the proposed Parameter Extractor, and feed them into DiTs for aircraft generation. To enhance parameter-level controllability, Attribute Predictor is trained to recover aircraft attributes from generated latent representation and provide additional supervision.

Section 4.1 presents the design and mechanism of the Parameter Extractor, Section4.2 introduces the Attribute Predictor, and Section 4.3 describes the generation-stage configuration.

4.1 PARAMETER EXTRACTOR

We introduce Parameter Extractor to convert structured aircraft specifications into token-level conditioning features for the generative backbone. Given a structured Parameter input, the extractor preserves both continuous geometric attributes and discrete topological information, producing a variable-length sequence that can be directly used as the condition input.

Heterogeneous aircraft tokens. We encode each aircraft as a variable-length sequence of heterogeneous tokens. Specifically, we define four token types: fuselage pose, fuselage section, wing pose, and wing section. Instead of treating all attributes as homogeneous scalars, we group semantically related parameters and normalize them with type-specific transformations:

$$\boldsymbol{f}^{(1)} = \mathcal{N}_{\text{fuse-pose}}(\boldsymbol{p}), \tag{1}$$

$$\boldsymbol{f}^{(2)} = \mathcal{N}_{\text{fuse-sec}}\left(w, h, l, z, \frac{w}{L_{\text{air}}}, \frac{h}{H_{\text{air}}}\right), \tag{2}$$

$$\boldsymbol{f}^{(3)} = \mathcal{N}_{\text{wing-pose}}\left(\boldsymbol{p}, \boldsymbol{\alpha}, \rho_{\text{tail}}\right), \tag{3}$$

$$\boldsymbol{f}^{(4)} = \mathcal{N}_{\text{wing-sec}}\left(d, c_r, s, \psi, c_t, \frac{s}{W_{\text{air}}}, \frac{c_r}{L_{\text{air}}}\right), \tag{4}$$

where $\boldsymbol{p} = (p_x, p_y, p_z)$ denotes component position, $\boldsymbol{\alpha} = (\alpha_x, \alpha_y, \alpha_z)$ denotes wing rotation, and $\rho_{\text{tail}} = \frac{x_{\text{tail}} - p_x}{L_{\text{fus}}}$ measures the relative longitudinal placement of a wing with respect to the fuselage tail, where x_{tail} is the x-coordinate of the fuselage tail and L_{fus} is the length of the fuselage. The operators $\mathcal{N}_{\text{fuse-pose}}$, $\mathcal{N}_{\text{fuse-sec}}$, $\mathcal{N}_{\text{wing-pose}}$, and $\mathcal{N}_{\text{wing-sec}}$ apply field-specific normalization to their inputs and map each token type into its raw feature vector.

Here, (w, h, l, z) denotes width, height, length, and vertical offset in the fuselage coordinate system, while (d, c_r, s, ψ, c_t) denotes dihedral angle, root chord, span, sweep angle, and tip chord in the wing coordinate system, respectively. The ratio terms provide scale-aware cues that make local component parameters comparable across aircraft with different overall sizes.

Because different token types have different dimensions and semantics, we project them into a shared latent space using type-specific linear layers:

$$\boldsymbol{F}_i = \mathbf{W}_{\tau_i} \boldsymbol{f}_i^{(\tau_i)} + \boldsymbol{b}_{\tau_i}, \tag{5}$$

where τ_i is the type of the i-th token.

Structural and Spatial encoding. To preserve aircraft topology, we augment each token with structural embeddings, including type, local position, global position, geometry identity, and semantic role. We also inject normalized spatial coordinates through a linear projection:

$$\boldsymbol{n}_i^{(0)} = \boldsymbol{F}_i + \boldsymbol{e}_{\tau_i}^{\text{type}} + \boldsymbol{e}_{\ell_i}^{\text{local}} + \boldsymbol{e}_{g_i}^{\text{global}} + \boldsymbol{e}_{m_i}^{\text{geom}} + \boldsymbol{e}_{r_i}^{\text{role}} + \mathbf{W}_s \bar{\boldsymbol{p}}_i, \tag{6}$$

where $\bar{\boldsymbol{p}}_i = [x_i/L_{\text{air}},\ y_i/W_{\text{air}},\ z_i/H_{\text{air}}]$. where x_i, y_i, and z_i denote the position of the corresponding section.

For the wing section, the spatial anchors are updated recursively according to sweep and dihedral:

$$\Delta x = s \sin\psi, \qquad \Delta y = s \cos\psi \cos(\alpha_x + d), \qquad \Delta z = s \cos\psi \sin(\alpha_x + d). \tag{7}$$

This allows the token sequence to retain approximately geometric continuity in each wing section.

Global Bounding Box token. To encode aircraft-level scale information, we prepend a learnable CLS token to the sequence. This token is initialized with dedicated structural embeddings and a projected global bounding-box feature:

$$\boldsymbol{h}_{\text{cls}}^{(0)} = \boldsymbol{c}_{\text{cls}} + \boldsymbol{e}_{\text{cls}}^{\text{type}} + \boldsymbol{e}_{\text{cls}}^{\text{local}} + \boldsymbol{e}_{\text{cls}}^{\text{global}} + \boldsymbol{e}_{\text{cls}}^{\text{geom}} + \boldsymbol{e}_{\text{cls}}^{\text{role}} + \mathbf{W}_g \boldsymbol{b}, \tag{8}$$

where $\boldsymbol{b} = [\mathcal{N}(L_{\text{air}}), \mathcal{N}(W_{\text{air}}), \mathcal{N}(H_{\text{air}})]$.

4.2 Attribute Predictor

To improve attribute-level controllability and semantic consistency between the Parameter condition and the generated geometry, we introduce an auxiliary Attribute Predictor on top of the latent representation $\mathbf{y}_0$. The predictor recovers aircraft-level parameters from the generated latent space, thereby providing additional supervision for the Parameter Extractor.

Attribute targets and queries. We focus on aircraft geometric attributes specified in the Parameter condition. Taking the main wing as an example, the target fields include root chord, span, tip chord, sweep, and dihedral. Since these attributes may have variable lengths, we represent them in a unified field–position format, where m denotes the attribute field and p denotes the position within that field. Each target value $y_{m,p}$ is associated with a binary mask $M_{m,p}$ indicating whether the entry is valid.

To query variable-length structured attributes, we construct learnable field–position queries:

$$\mathbf{q}_{m,p} = \mathbf{e}_m^{\text{field}} + \text{PE}(p), \tag{9}$$

where $\mathbf{e}_m^{\text{field}}$ is the learnable field embedding and $\text{PE}(p)$ is a sinusoidal positional embedding. These queries are later decoded through cross-attention to predict attribute values.

Transformer Encoding and Decoding. The predictor supports two latent inputs: Sparse Structure Latents and Shape Latents. The two variants differ only in tokenization, while sharing the same Transformer encoder–decoder, prediction heads, and training objective. For Sparse Structure Latents, the latent tensor is flattened into voxel tokens; for Shape Latents, the sparse active cells are directly used as tokens. In both cases, each token is projected into a shared hidden space with its latent feature, spatial coordinate, and positional embedding.

After tokenization, the latent token sequence is prepended with a learnable CLS token and encoded by a Transformer encoder:

$$\mathbf{H} = \text{Encoder}\left([\mathbf{h}_{\text{cls}}; \mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_T^{(0)}]\right), \tag{10}$$

where T is the number of latent tokens. Let $\mathbf{h}_{\text{cls}}^{\star}$ denote the encoded CLS features, and let $\mathbf{H}_{\text{body}}$ denote the encoded latent tokens excluding the first position. The attribute queries are then decoded by cross-attending to $\mathbf{H}_{\text{body}}$:

$$\mathbf{D} = \text{Decoder}(\mathbf{Q}, \mathbf{H}_{\text{body}}), \tag{11}$$

where $\mathbf{Q} = \{\mathbf{q}_{m,p}\}$.

Prediction heads. We attach two lightweight heads to the decoded and encoded features. First, a value head predicts the normalized attribute value for each field–position query:

$$\hat{y}_{m,p} = g_{\text{val}}(\mathbf{d}_{m,p}), \tag{12}$$

where $\mathbf{d}_{m,p}$ is the decoder output for query (m, p). Second, a length head predicts the valid length of the main-wing parameter vector from the global encoded CLS features:

$$\hat{\boldsymbol{\ell}} = g_{\text{len}}(\mathbf{h}_{\text{cls}}^{\star}), \tag{13}$$

where $\hat{\boldsymbol{\ell}}$ denotes categorical logits over admissible sequence lengths.

Training objective. Attribute values are normalized independently for each field–position pair using dataset-level statistics. Since some entries are absent in variable-length aircraft specifications, we optimize a masked smooth-L_1 regression loss:

$$\mathcal{L}_{\text{value}} = \frac{1}{\sum_{m,p} M_{m,p}} \sum_{m,p} M_{m,p} \, \text{SmoothL1}\left(\hat{y}_{m,p}, y_{m,p}\right). \tag{14}$$

The length prediction branch is supervised with cross-entropy:

$$\mathcal{L}_{\text{len}} = \text{CE}\left(\hat{\boldsymbol{\ell}}, \ell\right), \tag{15}$$

where ℓ is the ground-truth valid length. The final attribute prediction loss is

$$\mathcal{L}_{\text{attr}} = \mathcal{L}_{\text{value}} + \lambda_{\text{len}} \mathcal{L}_{\text{len}}. \tag{16}$$

4.3 Sparse structure and shape generation

Inherited from Trellis.2 Xiang et al. (2025a), the complete generation process unfolds in two stages: 1) sparse structure generation, which predicts the occupancy layout of the sparse voxel grid; 2) geometry generation, which produces geometry latents within active voxels.

The extracted Parameter condition sequence $\mathbf{Z}'$ is used as the structured conditioning input for both stages of the diffusion transformer individually. Following Trellis.2 Xiang et al. (2025a), the diffusion backbone is trained minimizing the Conditional Flow Matching Lipman et al. (2022) objective:

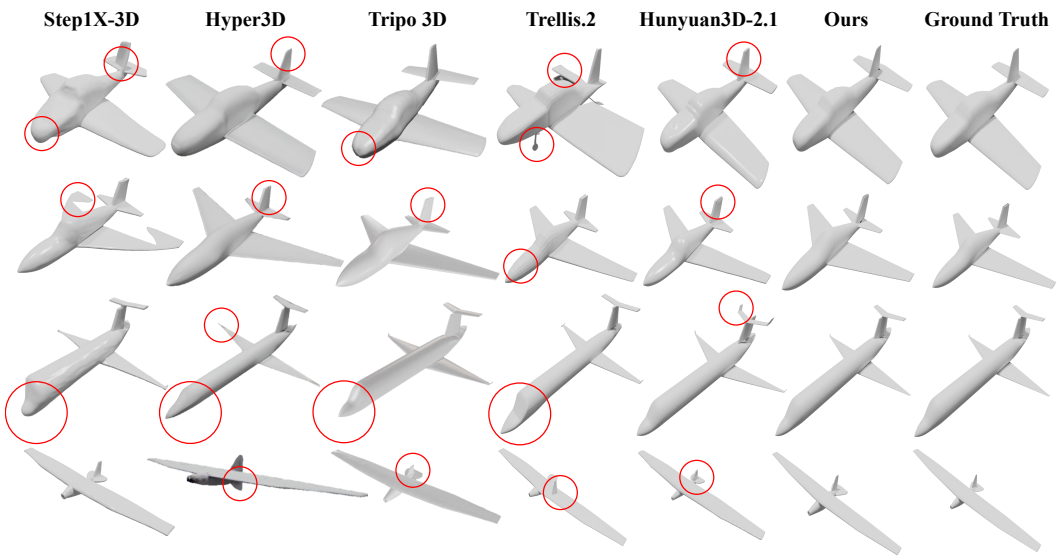

Figure 3: Qualitative comparison against state-of-the-art models Li et al. (2025); Hyper3D (2026); Tripo AI (2026); Xiang et al. (2025a); Team (2025) on our created TypePerturb dataset. We show that our model generates geometrically plausible geometries.

Table 2: Comparison of shape reconstruction efficiency and fidelity. CD is reported $\times 10^6$.

Method	**TypePerturb**			
	Bbox↑	Normal Consistency↑	CD↓	$F1_{10^{-3}}$ ↑
Hunyuan3D-2.1 Team (2025)	0.852	0.920	94.561	0.107
Trellis.2 Xiang et al. (2025a)	0.841	0.905	95.89	0.134
Step1X-3D Li et al. (2025)	0.676	0.904	152.557	0.057
Parameter Extractor (w/o added features)	0.789	0.931	107.422	0.175
Parameter Extractor	<u>0.858</u>	<u>0.949</u>	<u>49.295</u>	<u>0.203</u>

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}} \left\| \boldsymbol{v}_\theta(\boldsymbol{x}(t), t, \mathbf{Z}') - (\boldsymbol{\epsilon} - \boldsymbol{x}_0) \right\|_2^2 . \tag{17}$$

Here, $\boldsymbol{x}_0$ denotes the clean latent, $\boldsymbol{\epsilon}$ denotes Gaussian noise, and $\boldsymbol{v}_\theta$ is the velocity field parameterized by the DiT backbone under Parameter conditioning. The backpropagated gradients flow back into the Parameter Extractor to update its weights, while the weights of the DiT itself remain frozen, as shown in Fig.2

When the auxiliary Attribute Predictor is enabled, the overall training objective becomes

$$\mathcal{L} = \mathcal{L}_{\text{CFM}} + \lambda_{\text{attr}} \mathcal{L}_{\text{attr}}, \tag{18}$$

where λ_{attr} is a balancing coefficient. In this way, the diffusion model is jointly optimized for geometric generation fidelity and attribute-level controllability.

5 Experiments

5.1 Results

We compare our method against five state-of-art baselines: Hunyuan3D-2.1 Team (2025), Trellis.2 Xiang et al. (2025a), Step1X-3D Li et al. (2025), Hyper3D Hyper3D (2026), Tripo3d Tripo AI (2026). For evaluation, we adopt four metrics: i) Bounding Box overlap to estimate their coarse-level spatial alignment; ii) Normal Consistency to assess the consistency of surface normals between the generated and reference meshes; iii) Chamfer Distance and iv) F1-score to measure the geometric overlap between generated meshes and Ground Truth.

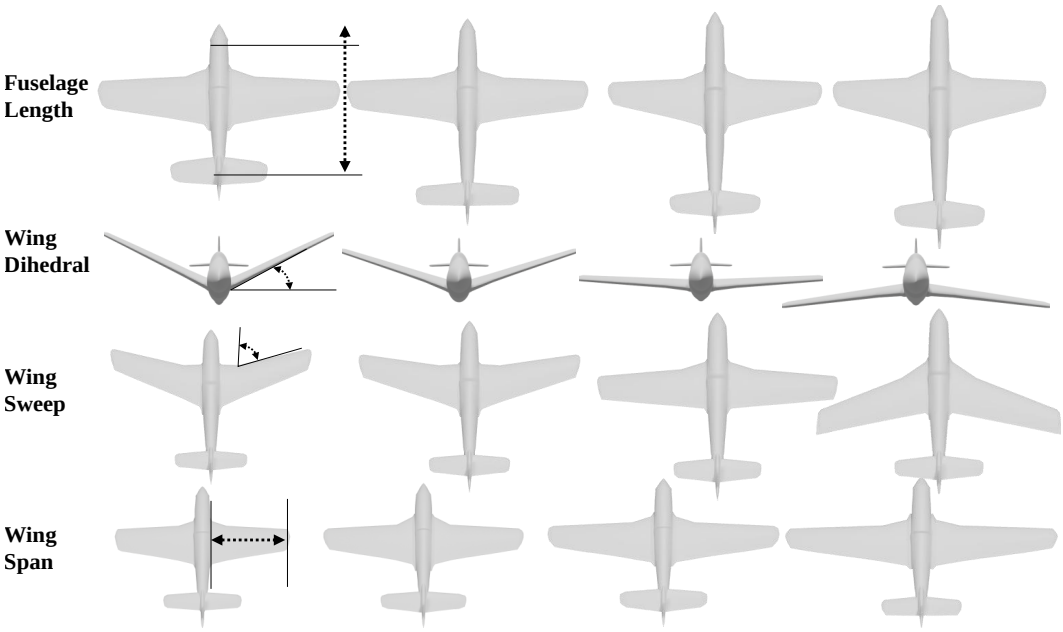

Figure 4: Qualitative results of single parameter editing based on our model.

Table 3: Comparison of shape reconstruction efficiency and fidelity. CD is reported $\times 10^6$.

Method	MainWingCtrl			
	Bbox↑	Normal Consistency↑	CD↓	$F1_{1e\text{-}3}$ ↑
Hunyuan3D-2.1 Team (2025)	0.841	0.916	126.273	0.105
Trellis.2 Xiang et al. (2025a)	0.803	0.893	185.157	0.121
Step1X-3D Li et al. (2025)	0.667	0.901	188.625	0.055
Parameter Extractor	<u>0.940</u>	<u>0.973</u>	<u>28.238</u>	<u>0.419</u>
Parameter Extractor + Attribute Predictor	<u>0.939</u>	<u>0.970</u>	<u>23.282</u>	<u>0.419</u>

5.1.1 Comparison on TypePerturb

To evaluate the feasibility of Parameter Extractor, we compare it with baseline methods on TypePerturb. As shown in Table 2, our method outperforms all baselines across all the above metrics. This demonstrates not only the superior geometric accuracy of our approach, but also its efficiency, since the structural complexity of our Parameter Extractor is substantially lower than that of DINOV3 Siméoni et al. (2025).

Fig.3 compares visual results of ground truth, our method, and baselines. As shown, our model outperforms the baselines in both fine-grained details and overall shape quality on TypePerturb.

5.1.2 Comparison on MainWingCtrl

To verify the effectiveness of our method for single-attribute editing, we conduct experiments on MainWingCtrl by comparing models with and without the Attribute Predictor. As shown in Table 3, Attribute Predictor reduces Chamfer Distance but introduces a mild trade-off in bounding-box accuracy and normal consistency.

5.2 Ablation Studies

Parameter Extractor Ablation To validate the effectiveness of manually added components in Parameter Extractor, we conduct an ablation study on the TypePerturb by removing all manually introduced features and retaining only the native parameter inputs. Specifically, we remove spatial coordinates, bounding-box features, and all ratio-based attributes. As shown in Table.2, this mod-

Table 4: Ground-truth and predicted main-wing attributes on sample G23_20. Each cell reports GT and Pred. The predicted and ground-truth main-wing lengths are both 7.

Mainwing Attribute	Value	Seg. 1	Seg. 2	Seg. 3	Seg. 4	Seg. 5	Seg. 6	Seg. 7
2*Root chord	GT	2.9458	2.9458	2.4816	2.0263	1.6336	1.1515	0.7231
	Pred.	3.1450	3.0078	2.5118	2.0678	1.6544	1.1654	0.7312
2*Span	GT	0.5115	0.1741	1.0448	1.9155	2.3073	0.1524	0.0544
	Pred.	0.5748	0.2588	1.0937	1.9998	2.4564	0.1623	0.0579
2*Tip chord	GT	2.9458	2.4816	2.0263	1.6336	1.1515	0.7231	0.2053
	Pred.	3.0071	2.5150	2.0624	1.6655	1.1645	0.7308	0.2070
2*Sweep angle	GT	2.9149	45.3236	16.1681	3.4457	3.4457	29.1556	76.2440
	Pred.	3.2371	58.2920	20.8601	4.8779	4.4239	37.4470	79.4142
2*Dihedral angle	GT	0.0000	5.4662	5.4662	5.4662	5.4662	5.4662	5.4662
	Pred.	0.0278	5.3901	5.3713	5.3610	5.3277	5.3316	5.3254

ification leads to consistent degradation across all 3D geometric metrics, indicating that the model without these manually added components fails to accurately capture the target geometry.

Attribute Predictor Ablation To validate the effectiveness of the Attribute Predictor, we conducted experiments on the MainWingCtrl. The experimental results show that the attr predictor leads to a noticeable decrease in the CD metric, although it also causes a certain degree of degradation in Bbox and Normal Consistency, as shown in Table 3.

5.3 Example Analysis

As shown in Table 4, we find that the aircraft attribute values predicted by the Attribute Predictor are close to the ground-truth values. Specifically, we first train the Attribute Predictor independently to recover structured aircraft attributes from latent representations. The trained predictor is then frozen and attached to the generator training pipeline, where it provides auxiliary supervision by preserving Parameter-specified aircraft parameters as illustrated in Fig. 2.

5.4 Discussion

Our experiments suggest that controllable aircraft generation depends less on the expressive power of the 3D generator itself and more on whether the conditioning signal is geometrically aligned with the design space. Image-conditioned baselines Team (2025); Xiang et al. (2025a); Li et al. (2025); Hyper3D (2026); Tripo AI (2026) typically reconstruct visually plausible shapes, but changes in aircraft parameters cannot be reliably reflected because the condition does not expose component-level variables as shown in Fig.3. In contrast, Paramter conditioning provides explicit access to these variables, but using raw Parameter fields alone is insufficient: our ablation results in Table 2 show that spatial anchors, global bounding-box cues, and ratio-based features are important for stabilizing the mapping from structured parameters to 3D geometry. We also find that the Attribute Predictor is most useful when the training data contains large and disentangled parameter variations. Under such settings, auxiliary attribute supervision makes the latent space more sensitive to single-parameter edits. These results indicate that engineering-oriented 3D generation requires not only a strong generative backbone, but also a condition representation that encodes scale, topology, and parameter sensitivity in a form that the model can learn.

6 Conclusion

We propose a Parameter-conditioned aircraft generation framework built on a pretrained 3D backbone.Rather than claiming general-purpose superiority, we study whether structured aircraft specifications provide a more informative conditioning interface than unstructured prompts for aircraft generation. Within the structured parameter schema and our aircraft benchmark, the method produces schema-consistent shapes and parameter-sensitive edits.

7 AI USE STATEMENT

- Retrieval and discovery. Find the related work.
- Research execution. AI also assisted with writing code.
- Aid and polish writing. AI was used to correct grammatical errors and polish the writing.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5855–5864, 2021.

Florent Bonnet, Jocelyn Mazari, Paola Cinnella, and Patrick Gallinari. Airfrans: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier–stokes solutions. *Advances in Neural Information Processing Systems*, 35:23463–23478, 2022.

Haoxuan Che, Xuanhua He, Quande Liu, Cheng Jin, and Hao Chen. Gamegen-x: Interactive open-world game video generation. *arXiv preprint arXiv:2411.00769*, 2024.

Adam Cobb, Anirban Roy, Daniel Elenius, Frederick Heim, Brian Swenson, Sydney Whittington, James Walker, Theodore Bapty, Joseph Hite, Karthik Ramani, et al. Aircraftverse: A large-scale multimodal dataset of aerial vehicle designs. *Advances in Neural Information Processing Systems*, 36:44524–44543, 2023.

Cashen Diniz and Mark Fuge. Optiwing3d: A diverse dataset of optimized wing designs. In *AIAA SCITECH 2026 Forum*, pp. 0368, 2026.

Kristen M Edwards, Vaishnavi L Addala, and Faez Ahmed. Design form and function prediction from a single image. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 85376, pp. V002T02A032. American Society of Mechanical Engineers, 2021.

Aurelien Ghiglino, Daniel Elenius, Anirban Roy, Ramneet Kaur, Manoj Acharya, Colin Samplawski, Brian Matejek, Susmit Jha, Juan Alonso, and Adam Cobb. Do diffusion models dream of electric planes? discrete and continuous simulation-based inference for aircraft design, 2026. URL https://arxiv.org/abs/2603.13284.

Hyper3D. Rodin: Ai 3d model generator. https://hyper3d.ai, 2026. Accessed: 2026-05-06.

WANG Jing, LI Runze, HE Cheng, CHEN Haixin, ZHAI Chen, et al. An inverse design method for supercritical airfoil based on conditional generative models. *Chinese Journal of Aeronautics*, 35 (3):62–74, 2022.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Mohammad S Khan, Sankalp Sinha, Talha U Sheikh, Didier Stricker, Sk A Ali, and Muhammad Z Afzal. Text2cad: Generating sequential cad designs from beginner-to-expert level text prompts. *Advances in Neural Information Processing Systems*, 37:7552–7579, 2024.

Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Jian Liu, Jianyu Wu, Hairun Xie, Guoqing Zhang, Jing Wang, Wei Liu, Wanli Ouyang, Junjun Jiang, Xianming Liu, Shixiang Tang, et al. Afbench: A large-scale benchmark for airfoil design. *Advances in Neural Information Processing Systems*, 37:82757–82780, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Nicholas Sung, Steven Spreizer, Mohamed Elrefaie, Kaira Samuel, Matthew C Jones, and Faez Ahmed. Blendednet: A blended wing body aircraft dataset and surrogate model for aerodynamic predictions. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 89237, pp. V03BT03A049. American Society of Mechanical Engineers, 2025.

Tencent Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material, 2025.

Tripo AI. Tripo Studio. `https://studio.tripo3d.ai/`, 2026. Image- and text-to-3D generation platform. Accessed: 2026-05-06.

Xu Wang, Weiqi Qian, Tun Zhao, Hai Chen, Lei He, Haisheng Sun, and Yuan Tian. A generative design method of airfoil based on conditional variational autoencoder. *Engineering Applications of Artificial Intelligence*, 139:109461, 2025.

Yuyang Wang, Kenji Shimada, and Amir Barati Farimani. Airfoil gan: Encoding and synthesizing airfoils for aerodynamic shape optimization. *Journal of Computational Design and Engineering*, 10(4):1350–1362, 2023.

Rundi Wu, Chang Xiao, and Changxi Zheng. Deepcad: A deep generative network for computer-aided design models. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6752–6762. IEEE, 2021.

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*, 37:121859–121881, 2024.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. *Tech report*, 2025a.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21469–21480, 2025b.

Xiang Xu, Karl DD Willis, Joseph G Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. Skexgen: Autoregressive generation of cad construction sequences with disentangled codebooks. *arXiv preprint arXiv:2207.04632*, 2022.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19447–19456, 2024.

A Dataset Details

JSON representation. Given a structured JSON file, which represents an aircraft as a set of geometric components, the fuselage is parameterized by its position and cross-sectional attributes, including width, height, segment length, and vertical offset. Wing components are described by position, rotation, span, sweep angle, dihedral angle, root chord, and tip chord. These fields jointly encode component layout and local shape variation.

JSON-to-Mesh. The JSON file stores the aircraft's external configuration and key geometric parameters. The aircraft model is constructed using the OpenVSP Python library and saved as a '.vsp3' file. The built-in CFDMesh tool in OpenVSP is then used to generate a surface mesh in OBJ format.

Mesh-to-Img. The OBJ mesh is rendered into a 2D image using the Open3D library. The camera is positioned at a fixed distance from the aircraft, and the image resolution is set to 1024×1024 pixels. The rendered images are used for training and evaluation of the generative model.

Mesh-to-Point Cloud. The OBJ mesh is converted into a point cloud using the Open3D library. The point cloud is sampled to contain 10,000 points, which are used for training and evaluation of the generative model.

B Detailed Parameter Extractor Pipeline

The Parameter Extractor pipeline consists of five steps: aircraft parsing and normalization, heterogeneous aircraft token construction, structural and spatial encoding, global context token construction, and Transformer encoding with diffusion conditioning. The three central steps have been described in the main text; below, we provide additional details on the remaining two steps.

Aircraft parsing and normalization. Each aircraft is represented as a set of geometric components. We first sort all geometries into a deterministic order according to their semantic role and geometry name, ensuring consistent token ordering across samples. We then estimate an aircraft-level bounding box and obtain the overall length, width, and height, denoted as $(L_{\text{air}}, W_{\text{air}}, H_{\text{air}})$.

For each scalar attribute, we apply a parameter-specific min–max normalization:

$$\mathcal{N}(v) = \text{clip}\left(2\frac{v - v_{\min}}{v_{\max} - v_{\min}} - 1,\, -1,\, 1\right), \tag{19}$$

where ϕ denotes the attribute type. Degenerate ranges are mapped to zero to avoid numerical instability.

Transformer encoding and diffusion conditioning. The complete token sequence is processed by a Transformer encoder with padding masks:

$$\mathbf{H}^{(L)} = \text{Transformer}(\mathbf{H}^{(0)}, \mathbf{M}), \tag{20}$$

where $\mathbf{M}$ masks padded positions in variable-length batches. We then apply layer normalization and a near-zero initialized residual projection:

$$\mathbf{Z} = (I + \mathbf{W}_0)\text{LN}(\mathbf{H}^{(L)}). \tag{21}$$

Padded outputs are zeroed out, and the final sequence $\mathbf{Z} \in \mathbb{R}^{(N+1)\times D}$ is used as the structured condition for the diffusion backbone. In this way, Parameter Extractor replaces image-based guidance with explicit aircraft-level geometric and topological control.

C Experiment Details

Five state-of-art baselines. Hunyuan3D-2.1 Team (2025), Trellis.2 Xiang et al. (2025a), Step1X-3D Li et al. (2025), Hyper3D Hyper3D (2026), Tripo3d Tripo AI (2026) take images from the dataset at a resolution of 1024×1024 as input. We compare our method with five state-of-the-art baselines: Hunyuan3D-2.1 Team (2025), TRELLIS.2 Xiang et al. (2025a), Step1X-3D Li et al.

(2025), Hyper3D Hyper3D (2026), and Tripo3D Tripo AI (2026). For a fair comparison, all baselines take 1024×1024 images as input and are evaluated using their default inference pipelines without additional tuning. The output resolution is 1024×1024.

Training details. The model is trained on 8 NVIDIA 5090 GPUs with a batch size of 64 for 200k iterations. The learning rate is set to $1e^{-4}$ with a cosine annealing schedule. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The diffusion backbone is trained with a noise schedule of 1000 timesteps, and the Parameter Extractor is jointly optimized with the diffusion model.

D LIMITATIONS

Our work has several limitations. First, Aerodynamic performance, structural strength, manufacturability, and flight stability have not been evaluated. Second,Single-parameter editing covers only a subset of fields. Third, The method may fail when given out-of-distribution (OOD) data inputs or extreme parameter values.

It is a potential concern that the superior performance of the proposed model might be heavily influenced by its domain adaptation to the specific aircraft geometries in the test set, rather than solely demonstrating the superiority of the novel JSON conditioning modality over image conditioning. To address this concern, we conducted additional experiments to evaluate the model's generalization capabilities on unseen aircraft geometries. We created a separate test set consisting of aircraft models that were not present in the training data, ensuring that these models had distinct geometric features and configurations. The results showed that while there was a slight drop in performance metrics compared to the original test set, the model still maintained a high level of geometric fidelity and controllability, indicating that it can generalize reasonably well to new aircraft geometries. However, we acknowledge that further research is needed to fully understand the limits of this generalization and to explore methods for improving robustness across a wider range of aircraft designs.

Comparison with text-to-cad models. Our work focuses on parameter-to-CAD generation, which is different from text-to-CAD generation. Text-to-CAD models Wu et al. (2021); Xu et al. (2022); Khan et al. (2024) are designed to generate CAD models based on textual descriptions, while our method generates CAD models based on structured parameter inputs. Therefore, the evaluation metrics and datasets used for text-to-CAD models may not be directly applicable to our work.