LAMP: DATA-EFFICIENT LINEAR AFFINE WEIGHT-SPACE MODELS FOR PARAMETER-CONTROLLED 3D SHAPE GENERATION AND EXTRAPOLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating high-fidelity 3D geometries that satisfy specific parameter constraints has broad applications in design and engineering. However, current methods typically rely on large training datasets and struggle with controllability and generalization beyond the training distributions. To overcome these limitations, we introduce LAMP (Linear Affine Mixing of Parametric shapes), a data-efficient framework for controllable and interpretable 3D generation. LAMP first aligns signed distance function (SDF) decoders by overfitting each exemplar from a shared initialization, then synthesizes new geometries by solving a parameter-constrained mixing problem in the aligned weight space. To ensure robustness, we further propose a safety metric that detects geometry validity via linearity mismatch. We evaluate LAMP on two 3D parametric benchmarks: DrivAerNet++ and BlendedNet. We found that LAMP enables (i) controlled interpolation within bounds with as few as 100 samples, (ii) safe extrapolation by up to 100% parameter difference beyond training ranges, (iii) physics performance-guided optimization under fixed parameters. LAMP significantly outperforms conditional autoencoder and Deep Network Interpolation (DNI) baselines in both extrapolation and data efficiency. Our results demonstrate that LAMP advances controllable, data-efficient, and safe 3D generation for design exploration, dataset generation, and performance-driven optimization.

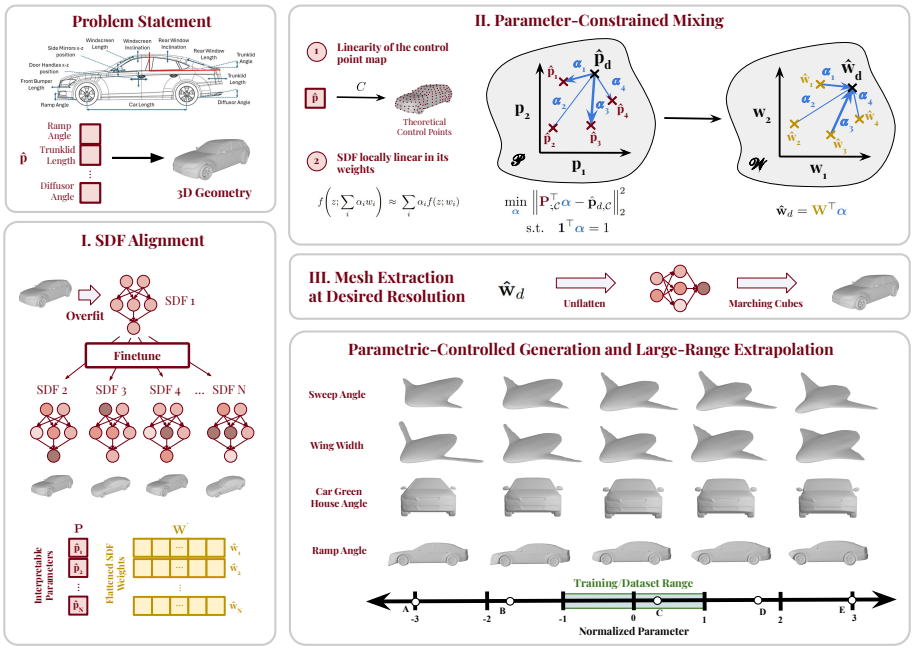


Figure 1: Overview of LAMP: (I) aligned SDF weight space construction, (II) parameter-constrained mixing, and (III) mesh extraction, enabling parametric control and large-range extrapolation.

1 INTRODUCTION

Engineering design applications often require generating 3D geometries that satisfy explicit, human-interpretable parameters (e.g., aerodynamic drag, roof height, and ramp angle). However, these applications are often data-scarce and require exploration beyond the limited span of available exemplars and parameter values unseen in the training distribution (e.g., generating a 3D car shape with a much larger ramp angle compared to known cars in the dataset). This opens up a fundamental challenge of how we can enable controllability and interpretability beyond training distributions in generative models across many domains, such as images, 3D shapes, text, and audio.

Many existing methods achieve control by learning a shared latent space and then traversing or disentangling latent dimensions to align with semantic or parametric factors (Vahdat et al., 2022; Zhao et al., 2023; Morita et al., 2024; Xiang et al., 2025). While powerful, latent approaches require a single generator to capture all variability, achieving clean, interpretable directions often proves difficult. The problem is particularly challenging in 3D shape generation, as most 3D generative models—whether voxel, point cloud, mesh, or implicit—rely on large datasets and provide few guarantees about parameter control or generalization beyond the training distribution.

We introduce **LAMP** (Linear Affine Mixing of Parametric shapes), a data-efficient framework for parameter-controlled 3D mesh generation. LAMP aligns signed distance function (SDF) decoders by overfitting each exemplar from a shared initialization, producing a consistent weight-space basis tied to design parameters. Given a target specification, we solve a constrained mixing problem to obtain coefficients, linearly combine exemplar weights, and decode a mesh. In other words, we cast controllable generation as parameter-space affine mixing across exemplar-specific networks.

This formulation leverages two simple but powerful observations: (i) interpretable attributes in many domains often combine approximately linearly in a suitable basis (e.g., control-point deformations in geometry), and (ii) neural decoders behave approximately linearly in their weights in a local regime around a common initialization. Together, these properties allow us to replace complex disentanglement or conditioning strategies with a lightweight linear solve and direct parameter mixing. We show that this method allows extrapolation beyond the training data and requires few samples.

To demonstrate reliability in meeting parameter constraints, we also introduce a method for direct *linearity-mismatch* safety check: for sampled 3D points, we compare the mixed decoder's output with the linear combination of individual SDFs, accepting generations only if the mean error is below 0.01. Constraint compliance is evaluated using a mesh-based surrogate that predicts parameters from PointNet embeddings with $R^2 > 0.9$ Qi et al. (2017), complemented by direct geometric measurements on decoded meshes. This combination provides both flexibility and accountability in design generation.

We validate LAMP on two parametric benchmarks, DrivAerNet++ and BlendedNet, across four case studies: (i) controlled interpolation within dataset bounds, (ii) safe extrapolation up to 100% beyond training ranges, and (iii) optimization for aerodynamic properties under fixed parameters. Together, these results demonstrate that LAMP enables controllable, data-efficient, and safe mesh generation for design exploration and performance-driven optimization.

LAMP thus bridges the gap between data-efficient generative modeling and practical engineering design. By combining aligned SDF weight spaces, provably safe mixing, and mesh-based surrogate validation, our framework provides a principled foundation for parameter-controlled 3D mesh generation. This enables not only faithful reproduction of in-distribution designs but also safe exploration of new configurations, including extrapolated and performance-optimized geometries. Ultimately, LAMP offers a scalable approach to transform a few annotated exemplars into a versatile design engine for constrained yet creative geometry synthesis.

More specifically, our contributions can be summarized as follows:

- **LAMP Method.** We propose LAMP, a data-efficient framework for parameter-controlled 3D mesh generation that aligns exemplar-specific SDF decoders to create a shared weight space and synthesizes new shapes through affine mixing from it.
- **Safety Metric.** We introduce a linearity-mismatch metric that certifies whether interpolated or extrapolated weight-space combinations by LAMP remain geometrically valid.

- **Engineering Applications.** We demonstrate LAMP on two benchmarks of aerodynamic 3D design (DrivAerNet cars and BlendedNet aircraft), showing controlled interpolation, large-range extrapolation (up to 100% beyond dataset bounds), and performance-driven optimization.

2 Related Work

Generative Models for 3D Shapes. A wide range of 3D generative models have been proposed, spanning voxel grids Wu & et al. (2016), point clouds Achlioptas & et al. (2018), meshes Groueix & et al. (2018), and neural implicit representations such as signed distance functions (SDFs) Park et al. (2019); Chibane et al. (2020). Neural SDFs capture high-resolution geometry and have been applied to generation and reconstruction Chen & et al. (2022) and reconstruction Atzmon & Lipman (2020). Recent diffusion-based 3D generators operate on implicit or latent representations, including LION, GET3D, Diffusion-SDF, SDFusion, and SALAD Zeng et al. (2022); Gao et al. (2022); Chou et al. (2023); Cheng et al. (2023); Koo et al. (2023). However, these methods typically assume abundant training data and lack explicit mechanisms for parameter-constrained generation or safe extrapolation. More recently, HyperDiffusion Erkoç et al. (2023) modeled the distribution of overfit implicit networks directly in weight space, sampling new fields via diffusion. These works treat networks themselves as data points in parameter space, but focus on unconditional sampling or learned meta-combination rather than explicit affine mixing under interpretable constraints.

Controllable and Conditional Generation. Efforts to introduce control often rely on conditioning on labels or attributes Gao et al. (2019); Niemeyer et al. (2020), or on parametric templates derived from CAD data Yumer & Mitra (2016); Wang et al. (2022). Diffusion models have recently been adapted for class-conditional and partially conditional shape generation Chen & et al. (2023); Liu & et al. (2023). Multimodal conditioning has also been explored for controllable 3D generation, e.g., CLIP-Forge and Michelangelo Sanghi et al. (2022); Zhao et al. (2023). However, these methods rarely support precise parameter specification (e.g., generating a car body with fixed ramp angle and width) and are not designed for data-efficient regimes. Our approach complements this line of work by enabling direct control through interpretable parameters.

Weight-Space Interpolation and Model Merging A few works have shown that linear operations in weight space can produce coherent outputs. Deep Network Interpolation (DNI) Wang et al. (2019) demonstrated smooth visual transitions by averaging parameters of two correlated image translation networks. In classification, model soups Wortsman et al. (2022) average fine-tuned networks to improve robustness. In generative modeling, researchers have merged GANs trained on different categories to yield hybrid semantics Avrahami et al. (2022), and also commonly combine diffusion models by interpolating or adding weight deltas to merge styles Biggs et al. (2024). These approaches confirm that weight-space mixing can yield meaningful interpolations, but typically operate on pairs of models and lack mechanisms for interpretable, constraint-driven control.

Parametric Design and Engineering Constraints. Engineering design relies heavily on parametric modeling, where small sets of interpretable variables govern global shape Seff & et al. (2020); Wang et al. (2022); Du & et al. (2023). While recent learning-based CAD systems leverage symbolic histories or constraint graphs, they often require large, structured datasets. By contrast, we target the data-efficient setting where only meshes and parameter annotations are available. LAMP directly links parameters to mesh geometry via aligned SDF weight spaces, and enforces validity through a linearity-mismatch safety metric and mesh-based surrogate checks.

Shape Interpolation and Extrapolation. Latent-space interpolation has been widely explored in autoencoders Achlioptas & et al. (2018) and implicit representations Park et al. (2019), but these spaces are often not semantically aligned, leading to unrealistic interpolations or invalid extrapolations. Our method leverages affine mixing of aligned SDF weights, which—combined with the linearity-mismatch criterion—ensures that interpolated or extrapolated meshes remain geometrically consistent and satisfy parametric constraints.

Position of This Work. Our approach bridges these threads. Like HyperDiffusion, we treat overfit exemplar networks as aligned points in parameter space, but instead of learning a generative model

over them, we provide direct, interpretable control by solving for mixing coefficients in parameter space and applying them in weight space. Unlike latent traversal or disentanglement, we do not rely on a single model to encode all variation. And unlike prior weight interpolation methods, we generalize beyond two-model blends to a bank of exemplars, enabling constraint-driven, multi-way affine mixing. To our knowledge, this is the first work to formulate controllable generation as parameter-space affine mixing of aligned exemplar networks.

3 METHOD

We present **LAMP**, a data-efficient framework for controllable 3D mesh generation that can safely interpolate and extrapolate in parameter space. LAMP (i) constructs an *aligned* weight-space basis by overfitting signed distance function (SDF) networks to a small set of 3D shapes, (ii) solves a parameter-constrained *mixing* problem to synthesize new SDF weights and decode meshes, and (iii) evaluates reliability using a linearity-based safety metric and a surrogate that predicts parameters directly from generated meshes.

Problem Setup and SDF Weight Space We are given N exemplars, each with mesh $\mathcal{M}_i$ and parameter vector $\mathbf{p}_i \in \mathbb{R}^d$ (e.g., length, width, roof height, ramp angle). For every design, we overfit an SDF network f_{θ_i}, starting from a shared initialization at the mean design. This yields weights $\mathbf{w}_i = \theta_i \in \mathbb{R}^D$ that live in an approximately *aligned* weight space. Stacking rows gives

$$\mathbf{P} \in \mathbb{R}^{N\times d}, \qquad \mathbf{W} \in \mathbb{R}^{N\times D}.$$

An arbitrary weight vector $\mathbf{w}$ is decoded into a mesh $\mathcal{M} = \mathrm{Decode}(\mathbf{w})$ by evaluating the zero-level set of the SDF distribution on a dense voxel grid at the desired resolution, and extracting the isosurface using the marching cubes algorithm Lorensen & Cline (1998).

Parameter-Constrained Mixing Given a target parameter vector $\mathbf{y}_d$ with a constrained index set $\mathcal{C}$, we solve the following optimization problem for mixing coefficients α:

$$\min_{\alpha} \left\|\mathbf{P}_{:\mathcal{C}}^{\top}\alpha - \mathbf{p}_{d,\mathcal{C}}\right\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^{\top}\alpha = 1. \tag{1}$$

The synthesized weights and decoded mesh are

$$\mathbf{w}_d = \mathbf{W}^{\top}\alpha, \qquad \mathcal{M}_d = \mathrm{Decode}(\mathbf{w}_d). \tag{2}$$

Negative α is allowed, enabling extrapolation beyond dataset bounds.

Theoretical Justification of Mixing Our framework builds on two key assumptions:

(A1) *Linearity of the control-point map.* Each mesh $\mathcal{M}_p$ can be described by a set of geometric control points $C(p) \in \mathbb{R}^{n\times 3}$, such as spline knots or characteristic vertices that define the surface. We assume the map from parameters to control points is linear:

$$C\left(\sum_i \alpha_i p_i\right) = \sum_i \alpha_i C(p_i).$$

This reflects how most engineering deformations are modeled: affine transformations (translation, scaling, stretching), spline coefficient adjustments, or superpositions of independent deformations. Even when nonlinear parameterizations are used (e.g., quadratic variations in thickness or curvature), they can often be re-expressed in a linear basis of control-point coefficients (see Appendix A).

(A2) *Local linearity in SDF weights.* For fixed input z, the decoder $f(z; w)$ is approximately linear in w in a neighborhood of a reference w_0:

$$f\left(z; \sum_i \alpha_i w_i\right) \approx \sum_i \alpha_i f(z; w_i),$$

with error $O(\max_i \|w_i - w_0\|^2)$. (Proof in Appendix C)

Under (A1)–(A2), interpolating weights $\hat{w}_\alpha = \mathbf{W}^{\top}\alpha$ produces an SDF close to $\mathrm{SDF}(C(\hat{p}_\alpha))$ with $\hat{p}_\alpha = \mathbf{P}^{\top}\alpha$, ensuring that mixing in weight space corresponds to faithful geometric interpolation and extrapolation. (Proof in Appendix B)

Safety Metric: Linearity Mismatch We additionally quantify whether affine mixing remains in a valid linear regime (A2). For N_z sampled 3D coordinates $\{z\}$, we compute

$$\frac{1}{N_z}\sum_z \Big| f\big(z; \sum_i \alpha_i w_i\big) - \sum_i \alpha_i f(z; w_i) \Big|.$$

A mesh is accepted if this mismatch is below ϵ. This provides a quantitative safety threshold: low mismatch implies faithful linear mixing, while high mismatch indicates collapse (Fig. 5).

4 Experiments, Results, and Discussion

We evaluate LAMP against two representative baselines for parameter-controlled 3D generation:

DNI (Deep Network Interpolation): a learned model mapping design parameters directly to SDF decoder weights Wang et al. (2019). **AE-LPA (Autoencoder with Latent-Parameter Alignment)**: an autoencoder trained to reconstruct SDF weights with its latent subspace linearly aligned to design parameters Jain et al. (2021).

We benchmark these methods on two recent parametric datasets (Appendix I): **DrivAerNet++**: a large-scale multimodal car dataset with $\sim 8{,}000$ distinct geometries, each annotated with 26 interpretable design parameters, high-resolution meshes, and CFD-based aerodynamic coefficients. Elrefaie et al. (2024) **BlendedNet**: a blended wing-body (BWB) aircraft dataset with 999 geometries, each simulated under 9 flight conditions, and annotated with planform parameters such as chord-length ratios, spanwise widths, and sweep angles. Sung et al. (2025) For brevity, we focus on DrivAerNet++ in the main paper and report full BlendedNet results in Appendix D.

Evaluation Metrics For in-dataset generation (when the target mesh is available), we evaluate: **Chamfer Distance (CD)** and **Intersection-over-Union (IoU)** between generated and reference meshes (see Appendix F). **Parameter Error:** mean absolute error (MAE) between target parameters and surrogate-predicted parameters for the generated mesh. For out-of-distribution extrapolation (no ground-truth mesh), we evaluate: **Parameter Fidelity:** surrogate-predicted MAE and R^2 between target parameters p_d and inferred $\hat{p}$, **Distributional Similarity:** Minimum Matching Distance (MMD) between generated shapes and a reference set of geometries. (see Appendix F).

Constraint Compliance via Mesh-Based Surrogates To assess whether generated shapes respect design-parameter constraints, we employ a mesh-based surrogate model trained to predict interpretable parameters (e.g., geometric or performance attributes). We compute fixed, randomly initialized PointNet embeddings for each decoded mesh Amid et al. (2022), and fit a LASSO regressor Tibshirani (1996) to map embeddings to physical parameters. Despite using an untrained encoder, the surrogate consistently achieves $R^2 > 0.9$ on held-out test sets, providing a robust parameter validator. When possible, we further cross-check compliance through *direct geometric measurements* of the meshes (details in Appendix G).

Interpolation within Dataset Range We evaluate interpolation by reconstructing meshes from randomly sampled dataset examples that fall within the parameter range but are excluded from training. As shown in Table 5, LAMP achieves the best performance across Chamfer Distance, IoU, and parameter error. Notably, with only 100 samples, it surpasses AE-LPA trained on 1000 samples, demonstrating strong sample efficiency in low-data regimes.

Table 1: Interpolation results on DrivAerNet++. Comparison of Chamfer Distance (CD), IoU, and parameter error (MAE) across methods.

Method	# Samples	CD ↓	IoU ↑ (%)	MAE ↓
DNI	100	0.0118	97.12	0.0015
AE-LPA	100	0.0181	88.21	0.0025
AE-LPA	1000	0.0144	92.63	0.0020
LAMP (Ours)	100	**0.0117**	**97.24**	**0.0014**

Extrapolation within Dataset Range We next evaluate extrapolation slightly beyond the training set. Training samples are drawn from a centered 50% interval of the parameter range, and evaluation is performed on cars with parameter values outside that interval. Figure 8 illustrates the *single-parameter* case, showing extrapolation along front bumper curvature. While DNI—the strongest baseline within the dataset range—begins to drift outside the training span, LAMP maintains smooth, parameter-consistent geometry.

Table 6 reports quantitative results. For single-parameter extrapolation, LAMP improves Chamfer Distance, IoU, and surrogate error relative to DNI. The advantage grows in the *multi-parameter* setting, where three random parameters are simultaneously extrapolated outside a 60% centered range: LAMP reduces surrogate MAE by 20–30%, showing greater robustness under multiple constraints.

Table 2: Extrapolation within dataset range (DrivAerNet++). LAMP outperforms DNI in both single- and multi-parameter settings, with especially large gains when multiple parameters are extrapolated simultaneously.

Method	**Single Parameter**			**Multi-Parameter (3)**		
	CD ↓	IoU ↑	MAE ↓	CD ↓	IoU ↑	MAE ↓
DNI	0.0129	95.32	0.098	0.0139	94.28	0.186
LAMP (Ours)	**0.0126**	**95.75**	**0.077**	**0.0130**	**95.29**	**0.144**

Large-Range Extrapolation Beyond the Dataset Bounds We next evaluate extrapolation far outside the training span, extending parameters up to $\pm 100\%$ beyond the dataset limits (i.e., three times the original parameter range). This task is challenging, as models must generate plausible geometries without support from nearby training examples. We consider two extrapolation settings in our experimental setup:

1. **Single-Parameter Extrapolation.** For each parameter, we sweep across the extrapolated range ($3\times$ the dataset span) using 10 uniformly sampled target values, while allowing all other parameters to vary freely. This setup evaluates whether models sustain smooth and consistent shape evolution along a single direction of variation (Figs. 3, 2).
2. **Multi-Parameter Extrapolation (4D).** We repeat 100 trials where four parameters are randomly selected and set outside the dataset range (up to 50% extrapolation, i.e., $2\times$ the span). Each method generates meshes under these conditions, which are then converted to point clouds, embedded with a fixed PointNet encoder, and visualized in 2D via multidimensional scaling (MDS). This reveals both fidelity and diversity of extrapolated generations (Fig. 4).

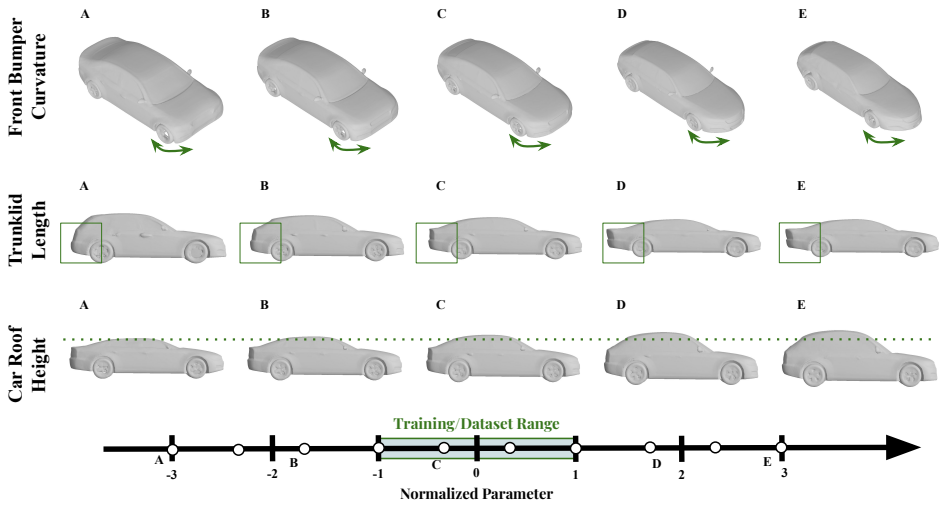


Figure 2: Single-parameter extrapolation showing LAMP's smooth, plausible geometries.

Table 3: Large-range extrapolation (DrivAerNet++). LAMP sustains high fidelity ($R^2 > 0.86$), while DNI collapses ($R^2 < 0$).

Method	**Single Parameter**			**Multi-Parameter (4)**		
	MMD ↓	MAE ↓	R^2 ↑	MMD ↓	MAE ↓	R^2 ↑
DNI	0.043	0.705	0.143	0.060	1.313	-5.768
AE-LPA	0.031	0.405	0.750	0.030	0.420	0.685
LAMP (Ours)	**0.030**	**0.247**	**0.902**	**0.030**	**0.324**	**0.867**

Results. Quantitative results are reported in Table 3. For single-parameter extrapolation, LAMP reduces parameter error by more than 40% compared to DNI and achieves $R^2 = 0.90$ versus $R^2 = 0.14$ for DNI. In the four-parameter case, DNI collapses completely ($R^2 < 0$), AE-LPA remains confined to the convex hull of the dataset, while LAMP sustains high fidelity ($R^2 = 0.87$) with low MMD and surrogate error.

Figures 2 and 3 illustrate the single-parameter sweeps: DNI collapses outside the training range and AE-LPA undershoots, whereas LAMP produces smooth, parameter-consistent variations that remain geometrically valid across the entire sweep. In the more challenging four-parameter extrapolation (Fig. 4), DNI collapses to invalid meshes and scatters randomly in embedding space, AE-LPA stays trapped in the dataset's convex hull with low diversity, while LAMP extrapolates beyond the convex hull and generates high-fidelity meshes in previously unobserved regions. This shows that LAMP can be used for dataset augmentation and controlled generation from a few samples.

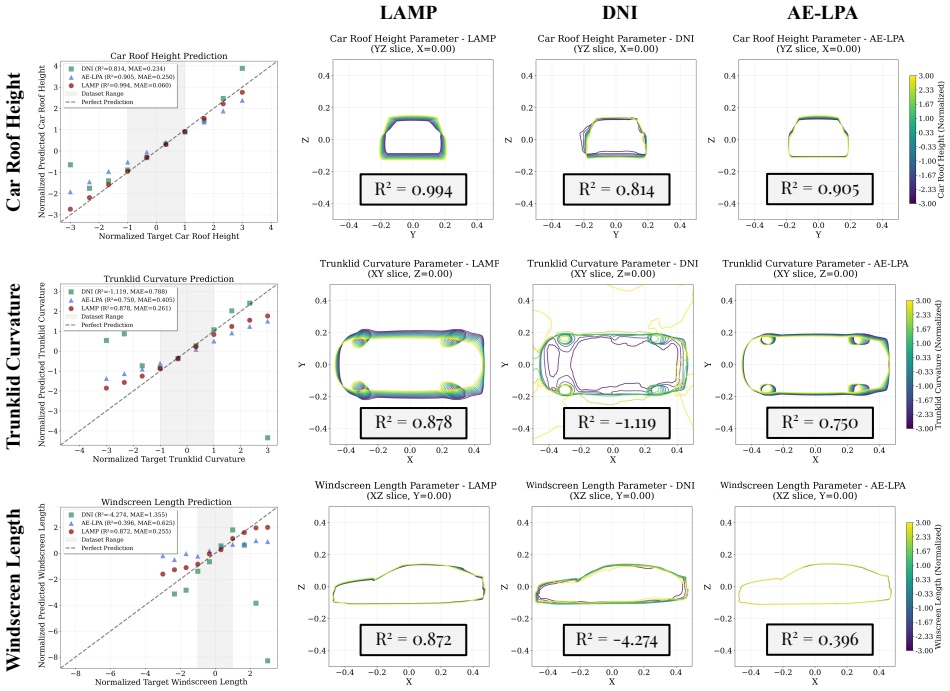


Figure 3: Single-parameter extrapolation beyond the dataset range, with all other parameters allowed to vary. Left: surrogate-predicted vs. target parameters. Right: decoded cross-sections. LAMP extrapolates smoothly, while DNI collapses and AE-LPA fails to reach the expected parameter range.

Challenges and Safety in Extrapolation with Limited Data A major challenge of large-range extrapolation is validating the plausibility of generated geometries when training data are scarce. With only 100 samples, mesh-based surrogates cannot be trained reliably to evaluate out-of-distribution designs. In such low-data regimes, models may occasionally produce collapsed or implausible shapes, especially when extrapolating far beyond the dataset span.

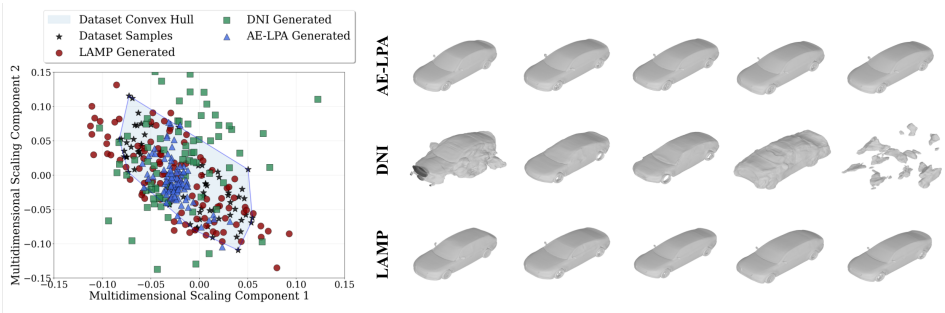


Figure 4: Four-parameter extrapolation. Left: distribution of generated meshes in a 2D point cloud embedding. Right: decoded examples. LAMP remains within plausible regions, DNI collapses to invalid meshes, and AE-LPA remains stuck in the dataset convex hull, lacking diversity.

To address this, we introduce a *linearity-mismatch safety metric* (Sec. 3), which quantifies whether affine weight mixing remains locally valid in SDF space. Unlike surrogate-based validation, this metric is lightweight and data-independent, enabling it to flag unsafe generations even when labeled training data are unavailable. As shown in Fig. 5, failure cases (f) arise precisely when the mismatch score exceeds a threshold.

We validate this metric against a human-annotated dataset of valid and invalid meshes (Appendix H). The results show excellent discriminative power (ROC AUC = 0.989, PR AUC = 0.990), and $\epsilon = 0.01$ emerges as a reliable threshold for separating valid from invalid generations. Additional examples of failure cases and validation analysis are provided in Figs. 17–18.

Another limitation is that we were only able to hard-code the measurement of car length, as explained in Appendix G, achieving $R^2 = 0.999$ under $\pm 100\%$ extrapolation sweeps. In contrast, other parameters were far more difficult to hard-code due to the complexity of their deformations across different car geometries and the absence of reliable methods for accurate measurement.

Finally, we study how reliability scales with data availability (Appendix E). As shown in Table 7, increasing the training set size improves both predictive accuracy (R^2, MAE) and the mean safe extrapolation range, which grows from $\sim 145\%$ with 10 samples to over 400% with 1000 samples before saturating. This ablation highlights both the limitations of extrapolation in extremely low-data regimes and the safety metric's role in flagging unreliable extrapolations.

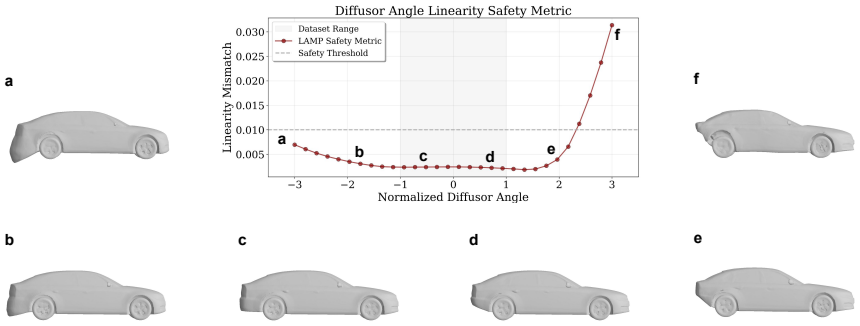


Figure 5: Linearity-mismatch safety metric for diffuser angle extrapolation. Failures (e.g., sample f) occur when the metric exceeds the threshold. See Appendix H for more.

Performance-Driven Optimization Beyond geometric parameters, we also test whether LAMP can enable *performance-based control*, where the goal is to optimize aerodynamic properties while constraining selected physical parameters. Specifically, we sample 100 random test examples from DrivAerNet++ outside the training set. For each example, we decay the drag coefficient (C_d) by 10% and select a random subset of physical parameters to be constrained to their original values, while treating the decayed C_d as an additional desired parameter. We then solve for mixing coefficients α that jointly satisfy these constraints.

To validate the results, we use the mesh-based surrogate (Appendix G) to predict both physical parameters and drag coefficients from the generated meshes. The surrogate predictions are compared to ground-truth values, with predicted vs. target plots provided in the Appendix. We evaluate two objectives: (i) *parameter fidelity*, i.e. how closely the generated meshes respect the selected physical parameter constraints, and (ii) *drag fidelity*, i.e. how accurately the achieved reduction matches the 10% target. Here, *decay MAE* denotes the mean absolute error between the desired 10% decay and the observed (predicted) decay, averaged across all samples.

Table 4: Performance-driven optimization on DrivAerNet++ for a 10% drag reduction target. LAMP achieves the best balance between parameter fidelity and aerodynamic performance.

Method	**Physical Parameters**		**Drag Coefficient**		
	MAE ↓	R^2 ↑	Decay MAE ↓ (%)	MAE ↓	R^2 ↑
DNI	0.810	-0.184	10.2	0.333	-8.917
AE-LPA	0.161	0.797	5.2	0.146	0.297
LAMP (Ours)	**0.087**	**0.938**	**2.7**	**0.121**	**0.792**

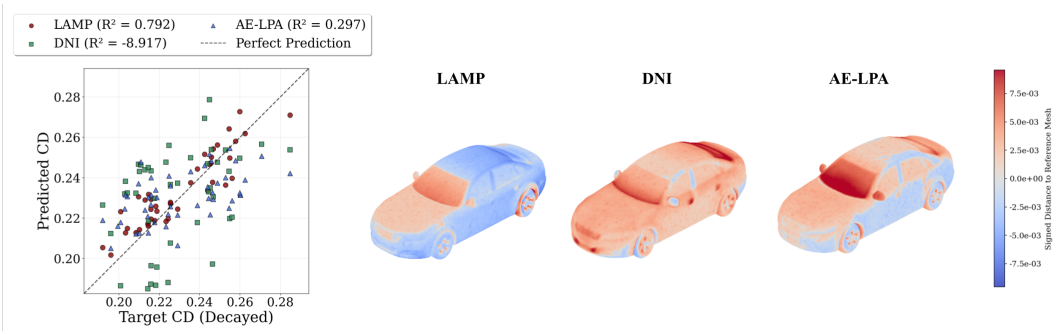


Figure 6: Performance-driven drag optimization on DrivAerNet++. Left: target vs. predicted drag reduction for LAMP, DNI, and AE-LPA. Right: error heatmaps relative to the reference mesh. LAMP achieves accurate prediction and a physically interpretable modification (flattened windscreen), while DNI and AE-LPA fail to produce aerodynamically meaningful changes.

Results. Figure 6 and Table 4 summarize the outcomes. The scatter plot confirms that LAMP aligns strongly with the target drag decay ($R^2 = 0.792$), while DNI diverges completely ($R^2 < 0$) and AE-LPA shows weaker correlation. Error heatmaps highlight the geometric changes driving drag reduction: LAMP produces a visibly flatter windscreen angle, reducing flow separation and lowering drag, whereas DNI and AE-LPA introduce noisy or less interpretable deformations.

Quantitatively, DNI fails to satisfy both aerodynamic and parametric constraints, with large parameter drift (MAE $= 0.810$) and unstable drag predictions ($R^2 = -8.917$). AE-LPA maintains moderate parameter fidelity ($R^2 = 0.797$) but overshoots the decay target (decay MAE $= 5.2\%$). In contrast, LAMP achieves the best trade-off: parameter fidelity improves to MAE $= 0.087$ with $R^2 = 0.938$, and drag reduction error is reduced to just 2.7%, while maintaining the strongest correlation for drag. Together, these results show that LAMP not only respects parameter constraints but also identifies physically meaningful pathways for aerodynamic optimization.

5 Conclusion

We presented LAMP, a data-efficient framework for parameter-controlled 3D mesh generation that leverages affine mixing in aligned SDF weight spaces and a linearity-based safety metric. Experiments on DrivAerNet++ and BlendedNet show that LAMP outperforms conditional autoencoders and DNI across interpolation, large-range extrapolation, and performance-guided optimization, achieving reliable control with as few as 100 exemplars. The safety score provides a principled safeguard in low-data regimes, addressing a key challenge for robust generalization. LAMP advances the goal of controllable, efficient, and verifiable 3D generation, common in engineering applications. Future directions include extending the framework to partial parameter observability, multi-modal conditioning, and integrating physics-aware constraints.

REFERENCES

Panos Achlioptas and et al. Learning representations and generative models for 3d point clouds. In *ICML*, 2018.

Ehsan Amid, Rohan Anil, Wojciech Kotłowski, and Manfred K Warmuth. Learning from randomly initialized neural network features. *arXiv preprint arXiv:2202.06438*, 2022.

Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. *CVPR*, 2020.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Gan cocktail: Mixing GANs without dataset access. In *Computer Vision – ECCV 2022*, volume 13683 of *Lecture Notes in Computer Science*, pp. 205–221. Springer, 2022.

Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Diffusion soup: Model merging for text-to-image diffusion models. In *European Conference on Computer Vision (ECCV)*, 2024.

Hang Chen and et al. Sdf-diff: Differentiable rendering of signed distance fields for 3d shape editing. In *ECCV*, 2022.

Yilun Chen and et al. Maskedgen: Conditional 3d shape generation with masked diffusion. 2023.

Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G. Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4456–4465, June 2023.

Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion. *arXiv.2003.01456*, 2020. doi: 10.48550/arXiv.2003.01456.

Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2262–2272, October 2023.

Ruida Du and et al. Cad2shape: Learning geometry representations from cad. In *ICCV*, 2023.

Mohamed Elrefaie, Florin Morar, Angela Dai, and Faez Ahmed. Drivaernet++: A large-scale multimodal car dataset with computational fluid dynamics simulations and deep learning benchmarks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 499–536. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/013cf29a9e68e4411d0593040a8a1eb3-Paper-Datasets_and_Benchmarks_Track.pdf`.

Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14300–14310, October 2023.

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. GET3D: A generative model of high quality 3d textured shapes learned from images. In *Advances in Neural Information Processing Systems*, 2022.

Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38 (6):1–15, 2019.

Thibault Groueix and et al. Atlasnet: A papier-mâché approach to learning 3d surface generation. In *CVPR*, 2018.

Saachi Jain, Adityanarayanan Radhakrishnan, and Caroline Uhler. A mechanism for producing aligned latent spaces with autoencoders. *arXiv preprint arXiv:2106.15456*, 2021.

Juil Koo, Seungwoo Yoo, Minh Hieu Nguyen, and Minhyuk Sung. Salad: Part-level latent diffusion for 3d shape generation and manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14441–14451, October 2023.

Hao Liu and et al. Gen-ldm: A generative latent diffusion model for 3d shape generation. *arXiv preprint arXiv:2301.XXXX*, 2023.

Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *Advances in neural information processing systems*, 36:44860–44879, 2023.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

Hayata Morita, Kohei Shintani, Chenyang Yuan, and Frank Permenter. Vehiclesdf: A 3d generative model for constrained engineering design via surrogate modeling. In *NeurIPS 2024 Workshop on Data-driven and Differentiable Simulations, Surrogates, and Solvers*, 2024.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3504–3515, 2020.

Jeong Joon Park, Peter Florence, and et al. Deepsdf: Learning continuous signed distance functions for shape representation. *CVPR*, 2019.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Aditya Sanghi, Hang Chu, Joseph G. Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18603–18613, June 2022.

Ari Seff and et al. Sketchgraphs: A large-scale dataset for modeling relational geometry in cad. In *ECCV*, 2020.

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. *arXiv preprint arXiv:2509.07209*, 2025.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022.

Chi Wang, Yewen Tian, and et al. Deepcad: A deep generative network for computer-aided design models. In *CVPR*, 2022.

Xintao Wang, Ke Yu, Chao Dong, Xiaoou Tang, and Chen Change Loy. Deep network interpolation for continuous imagery effect transition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1701, 2019.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Jiajun Wu and et al. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NeurIPS*, 2016.

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 Computer Vision and Pattern Recognition Conference*, pp. 21469–21480, 2025.

Mehmet Ersin Yumer and Niloy Mitra. Learning semantic deformation flows with 3d convolutional networks. In *ECCV*, 2016.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems*, 2022.

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. In *Advances in Neural Information Processing Systems*, 2023.

TABLE OF CONTENTS FOR APPENDICES

A On the Linearity of the Control-Point Map

Our theoretical justification in §3 requires assumption (A1), i.e., that the control points defining a mesh are a linear function of design parameters. Here we explain why this assumption is natural and broadly applicable.

Affine deformations. If a mesh is transformed by translation, scaling, or uniform stretching along a coordinate axis, then each control point is exactly a linear function of the corresponding parameter. For example, increasing the wing span of an aircraft by Δs simply adds Δs to the x-coordinates of the wingtip control points.

Parameterized curves and surfaces. For many design families, parameters control polynomial or spline coefficients. Since a spline curve is itself a linear combination of control points, perturbing these coefficients changes the embedding linearly in parameter space. Even nonlinear geometric trends (e.g., quadratic camber variation) can be re-expressed in a linear basis of coefficients.

Superposition of deformations. When multiple independent deformations (length, width, rotation about an axis) are applied, the resulting control-point positions are affine functions of all parameters. Thus, any convex combination of parameter vectors yields a convex combination of control-point sets, consistent with (A1).

Coverage of practical deformations. Most engineering shape variation in engineering practice can be decomposed into linear control-point operations: extrusion height, lofting length, angle of attack, or wheelbase translation are all captured. More exotic nonlinear changes (e.g., tree-like topological branching) violate (A1) but are outside the scope of our controlled parametric families.

Takeaway. Assumption (A1) is not an artificial simplification but instead reflects how engineering models are actually parameterized: the majority of mesh variations of interest in engineering design are affine in a suitable control-point basis. This ensures that our interpolation scheme faithfully reproduces the geometry implied by parameter mixing in nearly all practical scenarios.

B Theoretical Justification of SDF Weight Interpolation

Theorem. Let $p_1, \ldots, p_N \in \mathbb{R}^d$ be parameter vectors defining meshes $\mathcal{M}_{p_i}$ via control points $C(p_i) \in \mathbb{R}^{n\times 3}$, where the map $x \mapsto C(x)$ is linear. Let $w_i \in \mathbb{R}^m$ denote the weights of an MLP SDF decoder $f(z; w)$ overfit to the mesh $\mathcal{M}_{p_i}$, and trained from a shared initialization. Suppose that:

(1) Control point interpolation is linear:

$$C\left(\sum_{i=1}^{N} \alpha_i p_i\right) = \sum_{i=1}^{N} \alpha_i C(p_i), \quad \sum_{i=1}^{N} \alpha_i = 1.$$

(2) Each MLP decoder satisfies:

$$f(z; w_i) \approx \text{SDF}(z; C(p_i)) := d_i(z),$$

where $d_i(z)$ denotes the signed distance from a queried location z to mesh $\mathcal{M}_{p_i}$.

(3) The decoder $f(z; w)$ is locally linear in weights w for fixed input z:

$$f\left(z; \sum_{i=1}^{N} \alpha_i w_i\right) \approx \sum_{i=1}^{N} \alpha_i f(z; w_i),$$

with error bounded by $O(\max_i \|w_i - w_0\|^2)$.

Then the interpolated SDF $f(z; \hat{w}_\alpha)$ approximates the signed distance function of the mesh defined by the control points $C(\hat{p}_\alpha)$, where $\hat{p}_\alpha = \sum_i \alpha_i p_i$. That is,

$$f(z; \hat{w}_\alpha) \approx \text{SDF}(z; C(\hat{p}_\alpha)).$$

Proof. By assumption (2), for each i,

$$f(z; w_i) \approx d_i(z) = \text{SDF}(z; C(p_i)).$$

Then by local linearity of f in weights (3),

$$f(z; \hat{w}_\alpha) \approx \sum_i \alpha_i f(z; w_i) \approx \sum_i \alpha_i d_i(z).$$

Now, because control points interpolate linearly by assumption (1), we define:

$$C_\alpha := \sum_i \alpha_i C(p_i) = C\left(\sum_i \alpha_i p_i\right) = C(\hat{p}_\alpha).$$

If the SDFs $d_i(z)$ correspond to shapes with shared topology and smooth variation in geometry, then the signed distances satisfy:

$$\sum_i \alpha_i d_i(z) \approx \text{SDF}(z; C_\alpha).$$

Therefore,

$$f(z; \hat{w}_\alpha) \approx \text{SDF}(z; C(\hat{p}_\alpha)).$$

Thus, the zero-level set of $f(z; \hat{w}_\alpha)$ corresponds to the mesh defined by $\hat{p}_\alpha$, completing the proof.

C Approximate Linearity of the SDF Decoder in Weights

Let $z \in \mathbb{R}^3$ be a fixed 3D input point, and let $\gamma(z) \in \mathbb{R}^D$ denote its Fourier positional encoding, defined as:

$$\gamma(z) = \left[z, \sin(2^0\pi z), \cos(2^0\pi z), \ldots, \sin(2^L\pi z), \cos(2^L\pi z)\right].$$

Let $f(z; w)$ be a feedforward multilayer perceptron (MLP) with parameters w and input $\gamma(z)$. The network is composed of K layers with weights and biases $\{W_k, b_k\}_{k=1}^K$, where:

$$\begin{aligned} h_0 &= \gamma(z), \\ h_k &= \phi(W_k h_{k-1} + b_k), \quad \text{for } k = 1, \ldots, K-1, \\ f(z; w) &= W_K h_{K-1} + b_K, \end{aligned}$$

with $\phi(\cdot)$ a fixed elementwise nonlinearity (e.g., ReLU). The parameter vector w collects all $\{W_k, b_k\}$.

Claim. For fixed z, if all weights $\{w_i\}_{i=1}^N$ lie in a small neighborhood of a reference w_0, then $f(z; w)$ is approximately linear in w. In particular, for convex coefficients $\{\alpha_i\}_{i=1}^N$ with $\sum_{i=1}^N \alpha_i = 1$, we have

$$f\left(z; \sum_{i=1}^N \alpha_i w_i\right) \approx \sum_{i=1}^N \alpha_i f(z; w_i),$$

with an error term of order $O(\|w_i - w_0\|^2)$.

Proof. Fix the input z, so $\gamma(z)$ is constant. Consider the Taylor expansion of $f(z; w)$ about w_0:

$$f(z; w) = f(z; w_0) + \nabla_w f(z; w_0)^\top (w - w_0) + R(w),$$

where $R(w)$ is the second-order remainder term.

Applying this to each w_i gives

$$f(z; w_i) = f(z; w_0) + \nabla_w f(z; w_0)^\top (w_i - w_0) + R(w_i).$$

Now evaluate at the convex combination $\hat{w}_\alpha = \sum_i \alpha_i w_i$:

$$f(z; \hat{w}_\alpha) = f(z; w_0) + \nabla_w f(z; w_0)^\top \Big(\sum_i \alpha_i (w_i - w_0)\Big) + R(\hat{w}_\alpha).$$

On the other hand, the convex combination of outputs is

$$\sum_i \alpha_i f(z; w_i) = f(z; w_0) + \nabla_w f(z; w_0)^\top \Big(\sum_i \alpha_i (w_i - w_0) \Big) + \sum_i \alpha_i R(w_i).$$

Subtracting the two expressions gives

$$f(z; \hat{w}_\alpha) - \sum_i \alpha_i f(z; w_i) = R(\hat{w}_\alpha) - \sum_i \alpha_i R(w_i).$$

Since the remainder terms $R(\cdot)$ are second-order in the deviations $(w_i - w_0)$, this difference is $O(\max_i \|w_i - w_0\|^2)$. Thus, when all weights are close, the error is small and the decoder behaves approximately linearly in w.

D ADDITIONAL QUANTITATIVE AND QUALITATIVE RESULTS

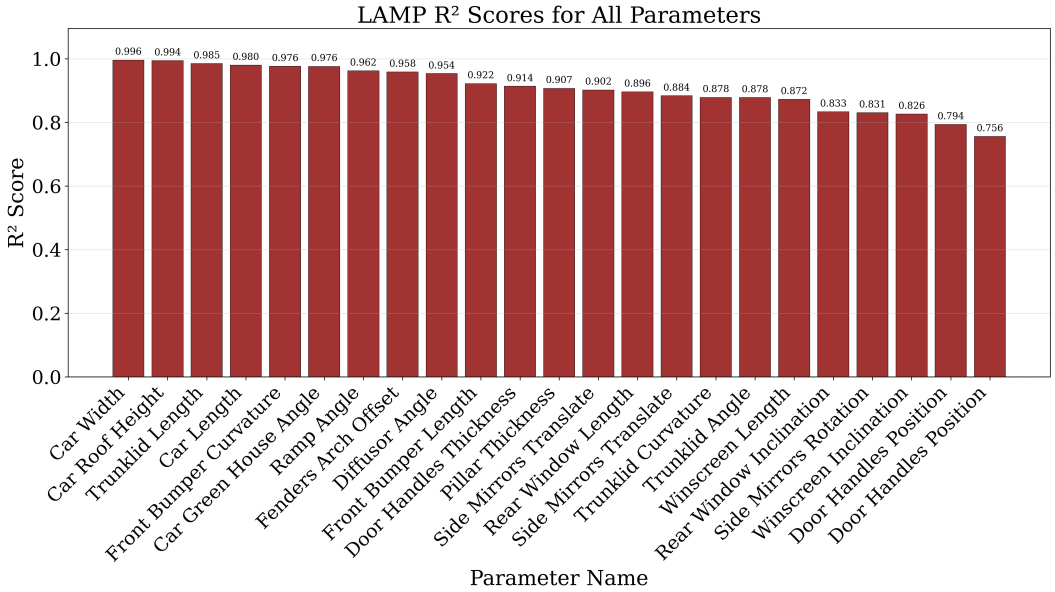


Figure 7: LAMP's R^2 scores for single-parameter sweeps on the DrivAerNet++ dataset, extrapolated $\pm 100\%$ beyond the dataset range.

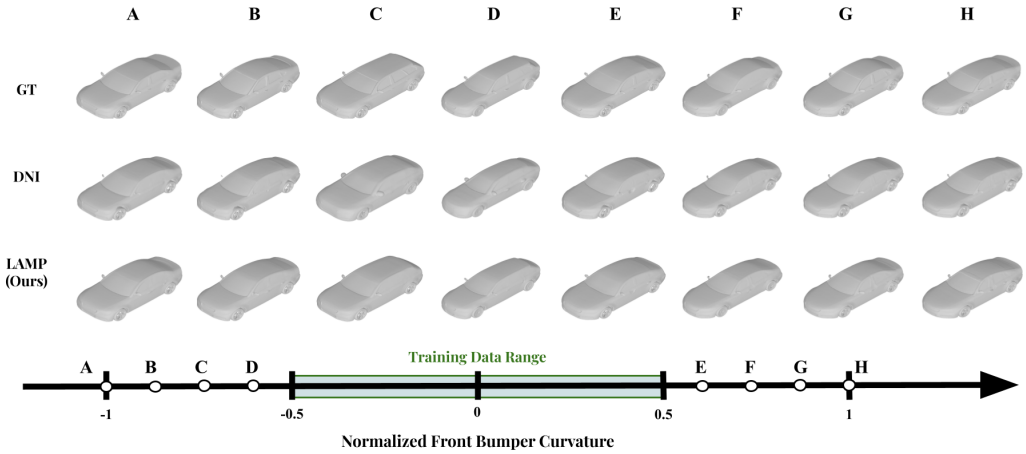


Figure 8: Single-parameter extrapolation within dataset range on DrivAerNet++. Training samples are restricted to a centered 50% interval of the parameter, while evaluation is performed outside this interval. LAMP maintains smooth, plausible extrapolation, while DNI drifts away from target shapes.

Table 5: Interpolation performance. We compare LAMP against DNI and AE-LPA baselines on BlendedNet using Chamfer Distance (CD ↓), Intersection-over-Union (IoU ↑, in %), and surrogate-based parameter error (↓). Training uses 100 samples from the dataset, and testing uses 200 held-out samples.

Dataset	**Method**	**# Samples**	CD ↓	IoU ↑ (%)	MAE ↓
BlendedNet	DNI	100	0.0346	94.21	0.0038
	AE-LPA	100	0.0393	88.26	0.0078
	LAMP (Ours)	100	**0.0172**	**95.35**	**0.0031**

Table 6: Large-range extrapolation (BlendedNet) up to $\pm 50\%$. LAMP sustains high fidelity ($R^2 > 0.78$) for both single- and multi-parameter extrapolation. Training uses 100 samples from the dataset. For single-parameter extrapolation, we sample 10 values uniformly per parameter within the extrapolated range, constraining that parameter while allowing the others to vary. For multi-parameter extrapolation, we repeat 100 trials where four parameters are randomly selected and set outside the dataset range (up to 50% extrapolation, i.e., twice the original span).

Dataset	**Method**	**Single Parameter**			**Multi-Parameter**		
		MMD ↓	MAE ↓	R^2 ↑	MMD ↓	MAE ↓	R^2 ↑
BlendedNet	DNI	0.038	0.392	0.784	0.040	0.435	0.521
	AE-LPA (100)	0.039	0.611	0.169	0.043	0.823	-0.069
	LAMP (Ours)	**0.035**	**0.305**	**0.868**	**0.037**	**0.353**	**0.782**

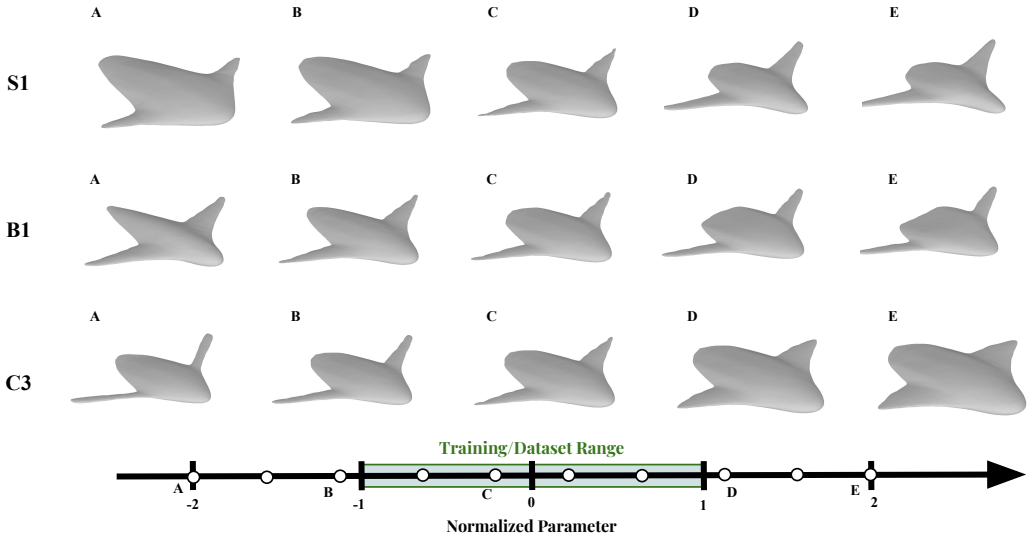


Figure 9: Single-parameter sweep on BlendedNet. LAMP sustains smooth, plausible geometry under large parameter shifts.

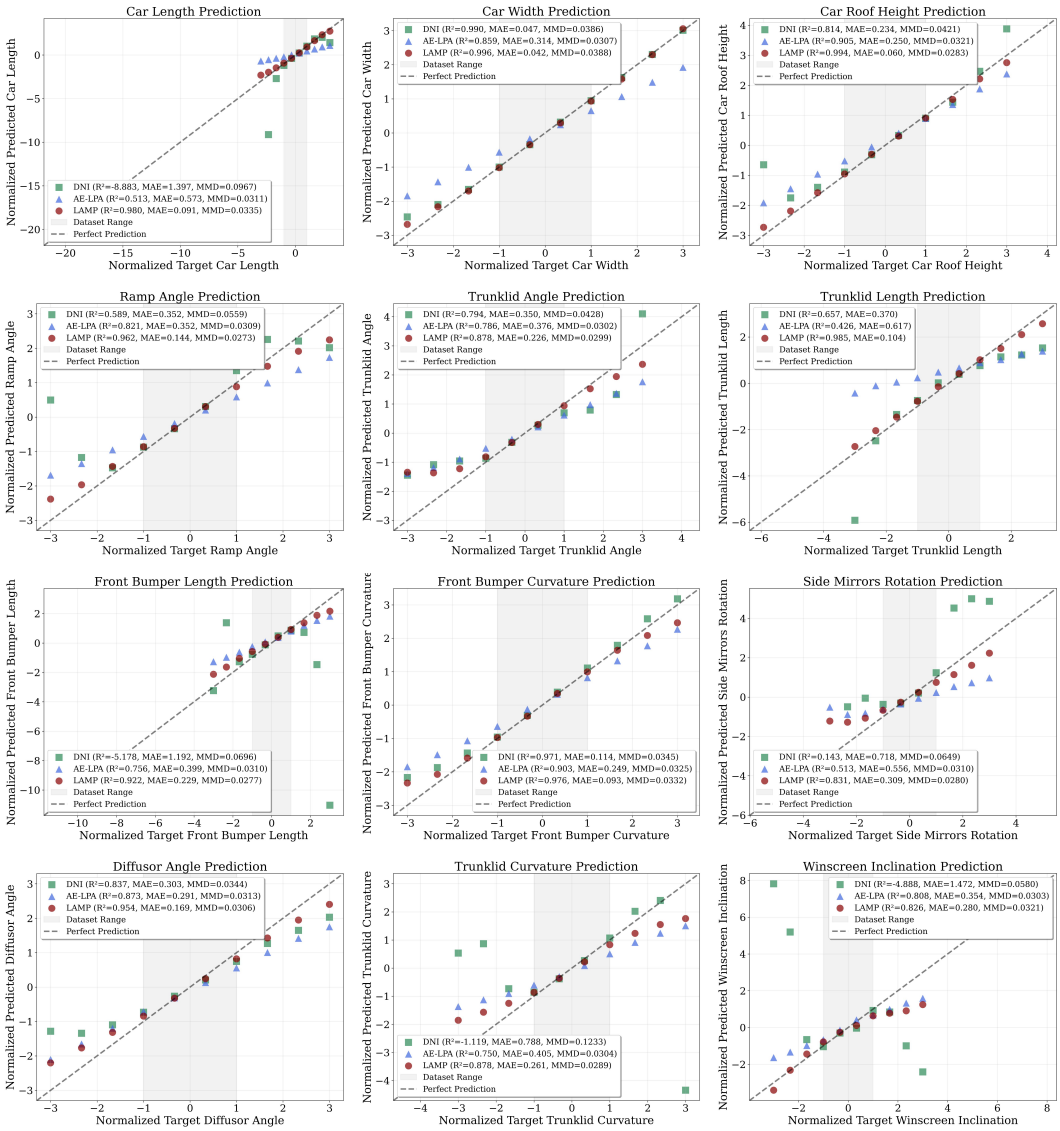


Figure 10: Single-parameter extrapolation beyond the dataset range. All other parameters are allowed to vary. The plots show surrogate-predicted versus target parameters, comparing LAMP against the baselines DNI and AE-LPA.

E Ablation Study: How Does Sample Size Affect Reliability and Extrapolation in LAMP?

We ablate the effect of sample size in Table 7. As the number of samples increases, MAE decreases while both R^2 and the mean safe extrapolation range (%) increase, before plateauing at larger sample counts. This trend indicates that performance improves with more samples but saturates beyond a certain scale.

Table 7: Ablation study on mixing quality and reliability across different numbers of samples using LAMP

Number of Samples	$\mathbf{R^2} \uparrow$	**MAE** $\downarrow$	**Mean Safe Extrapolation Range (%)** $\uparrow$
10	-7.289	2.650	145.8
50	-0.214	1.083	213.9
100	0.838	0.507	330.6
500	0.849	0.479	418.1
1000	0.862	0.486	427.8

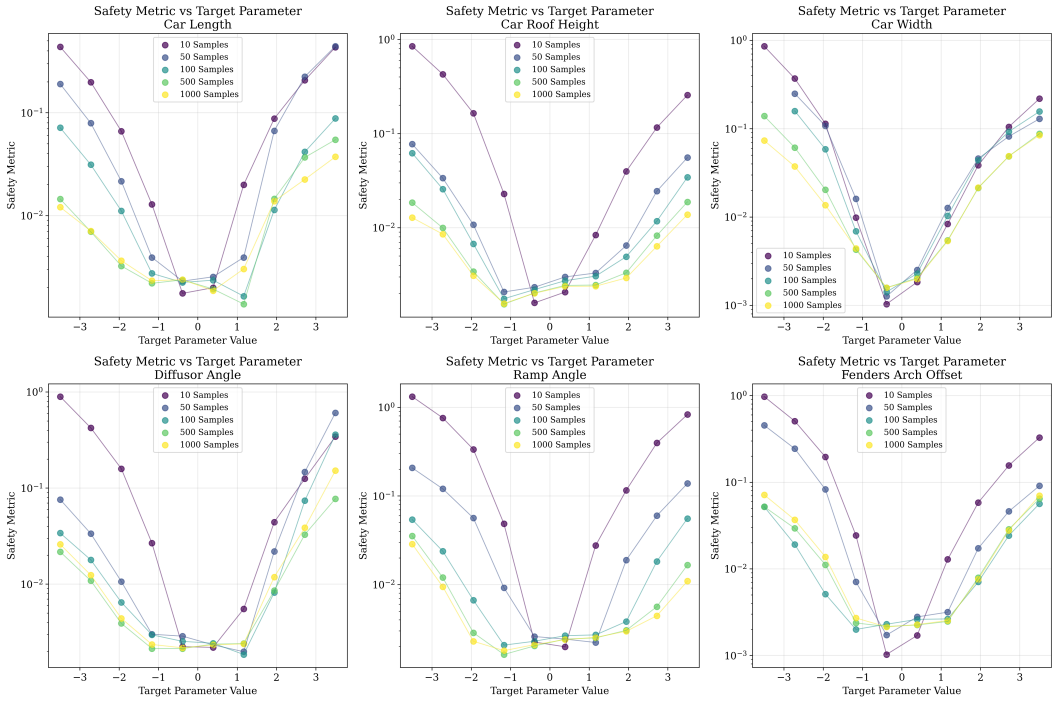


Figure 11: Safety metric values as a function of target parameter sweeps across six design parameters in DrivAerNet++. Curves correspond to different training set sizes (10–1000 samples). Larger datasets consistently reduce the safety metric, indicating more reliable extrapolation across parameter ranges.

F Evaluation Metrics

- **Chamfer Distance (CD):**

$$\mathrm{CD}(X,Y) = \frac{1}{|X|}\sum_{x\in X}\min_{y\in Y}\|x-y\|_2^2 \;+\; \frac{1}{|Y|}\sum_{y\in Y}\min_{x\in X}\|y-x\|_2^2,$$

  where X and Y are point clouds sampled from the predicted and reference meshes.
- **Intersection-over-Union (IoU):**

$$\mathrm{IoU}(A,B) = \frac{|A\cap B|}{|A\cup B|},$$

  where A and B are voxelizations of the predicted and reference meshes.
- **Minimum Matching Distance (MMD):**

$$\mathrm{MMD}(S_g,S_r) = \frac{1}{|S_g|}\sum_{x\in S_g}\min_{y\in S_r} d(x,y),$$

  where S_g and S_r are sets of generated and reference point clouds, and $d(\cdot,\cdot)$ is typically the Chamfer distance between individual shapes.

G Constraint Compliance Validation: Surrogates and Direct Measurements

Comparison of Mesh-Based Surrogates. In the main text, we validated design-parameter compliance using a mesh-based surrogate model: we fixed random PointNet embeddings of each decoded mesh (deterministic initialization) and fit a LASSO regressor to predict physical parameters. Interestingly, this simple surrogate achieves strong accuracy ($R^2 > 0.9$ on held-out test sets), despite the encoder being untrained.

To test whether stronger pretrained representations improve performance, we compared against the OpenShape point cloud embedding model Liu et al. (2023). Across all parameters, the OpenShape-based surrogate achieved consistently lower R^2 scores than the randomly initialized PointNet embeddings. This suggests that domain-specific geometric structure is better captured by lightweight randomized encoders than by pretrained embeddings trained on natural 3D categories.

Figure 12 illustrates the surrogate pipeline. Figures 14 and 13 show predicted versus ground-truth parameter values on the BlendedNet and DrivAerNet++ datasets, respectively, demonstrating high accuracy across test sets. We train the surrogates on 800 samples and evaluate on a held-out test set of 200 samples.

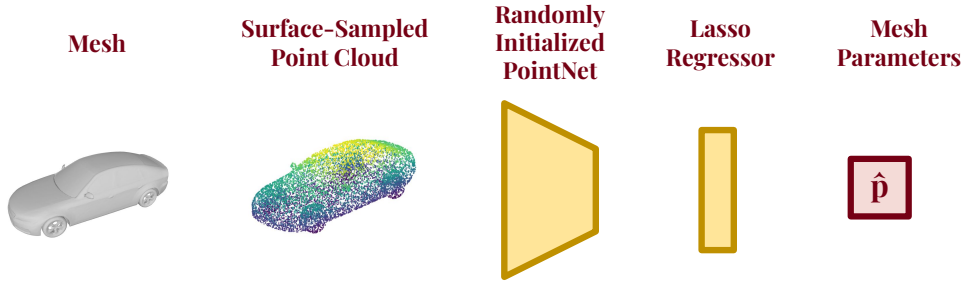


Figure 12: Diagram of the mesh-based surrogate pipeline. A decoded mesh is first converted into a surface-sampled point cloud. The point cloud is passed through a randomly initialized PointNet encoder to produce fixed embeddings, which are then mapped to interpretable mesh parameters via a LASSO regressor. This simple pipeline achieves strong predictive accuracy ($R^2 > 0.9$) despite the encoder being untrained.

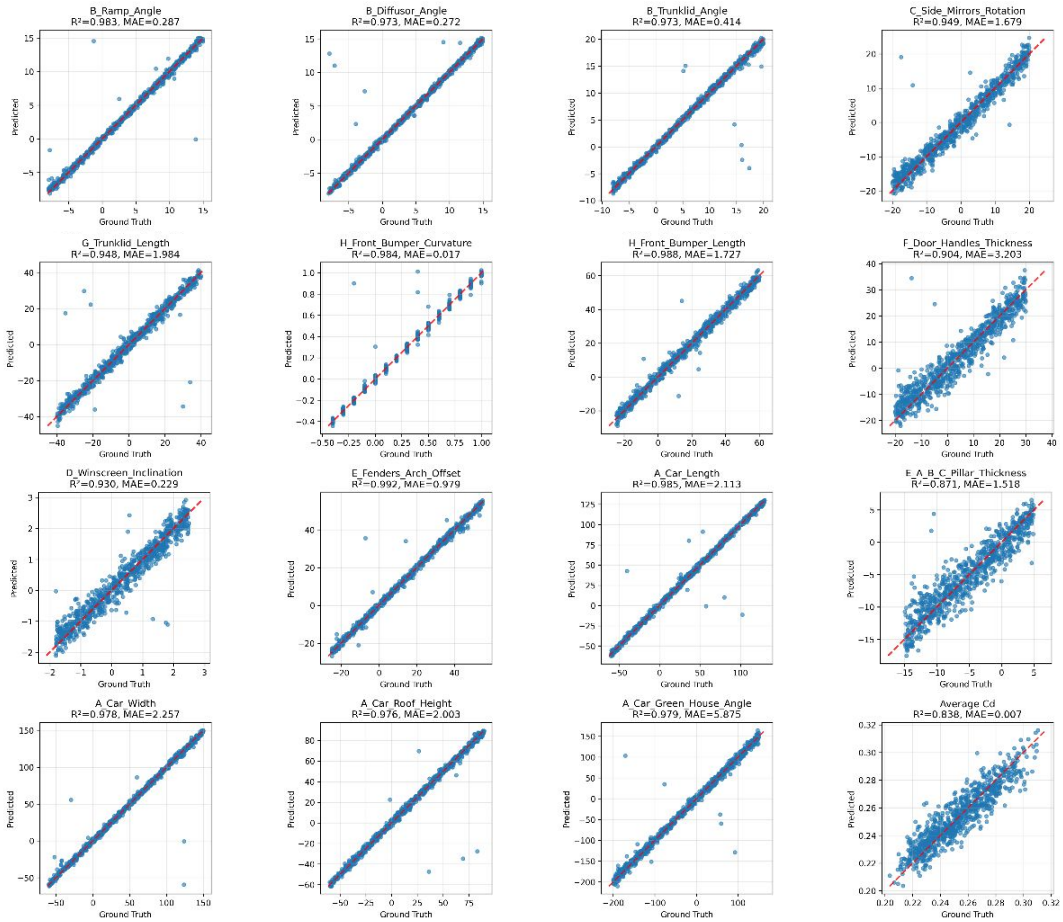


Figure 13: Predicted vs. ground truth parameters on the DrivAerNet++ test set, evaluating the mesh-based surrogate model for parameter prediction.

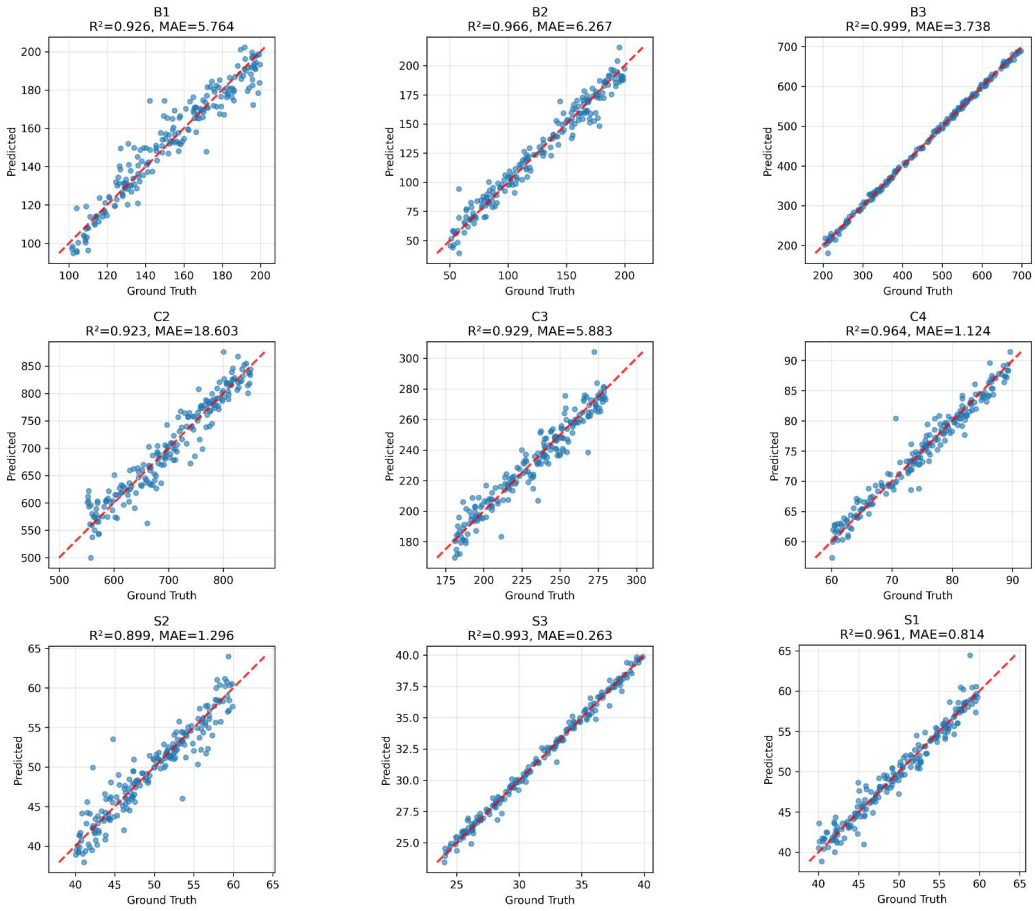


Figure 14: Predicted vs. ground truth parameters on the BlendedNet test set, evaluating the mesh-based surrogate model for parameter prediction.

Direct Geometric Measurements. Beyond surrogate-based validation, we also implemented direct geometric measurements for certain parameters. For example, to compute *Car Length* on DrivAerNet++ cars, we measure the distance between the centers of the front and rear wheels. Specifically:

1. We take slices of the decoded SDF along the wheel plane.
2. We detect circular cross-sections with radii in the expected range of wheel radii.
3. We identify the front and rear wheel centers and compute their distance.
4. We map this distance back to the labeled Car Length value by calibrating on ground-truth SDFs from the dataset.

This method provides a parameter-compliant, geometry-based validation of mesh outputs. Figure 15 shows an example of wheel detection and length estimation on generated meshes.

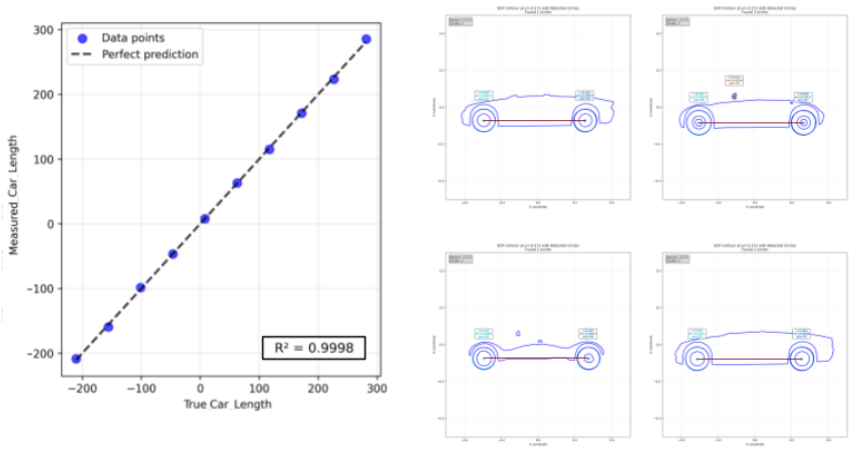


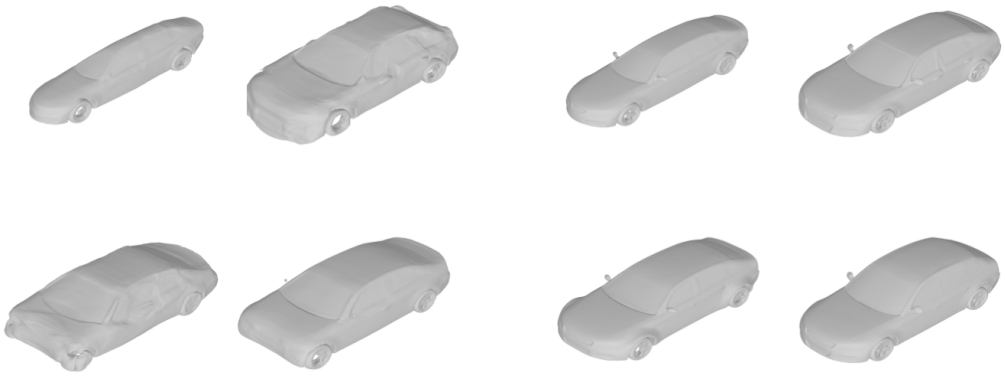
Figure 15: Direct geometric measurement of car length in DrivAerNet++. We slice the decoded SDF, detect wheel cross-sections by circle fitting, compute the distance between wheel centers, and map this measurement back to the dataset-defined Car Length parameter.

H Validation of the Linearity-Mismatch Metric Against Human Annotated Data

Assumption. Our safety metric relies on the assumption that the decoder $f(z; w)$ is locally linear in weights w for a fixed spatial coordinate z. Formally,

$$f\left(z; \sum_{i=1}^{N} \alpha_i w_i\right) \approx \sum_{i=1}^{N} \alpha_i f(z; w_i),$$

with approximation error bounded by $O\big(\max_i \|w_i - w_0\|^2\big)$. This implies that as long as interpolations in weight space remain sufficiently close to the training exemplars, affine mixing should yield faithful mesh reconstructions. The linearity mismatch defined in Sec. 3 measures deviations from this assumption.



Invalid Meshes **Valid Meshes**

Figure 16: Examples of meshes labeled during human annotation. Left: *Invalid meshes*, which exhibit collapsed, distorted, or implausible geometries. Right: *Valid meshes*, which maintain smooth, realistic car shapes with high geometric fidelity. These labels are used as ground truth to validate the safety metric.

Dataset Construction. To empirically validate this assumption, we constructed a diagnostic dataset by systematically varying one shape parameter at a time. Each parameter was interpolated and extrapolated up to a 700% (±300%) increase in range compared to its span in the main dataset. For every setting, we decoded a mesh $\mathcal{M}_d$ using mixed weights $\mathbf{w}_d = \sum_i \alpha_i w_i$ and computed the linearity-mismatch score. (Fig. 16

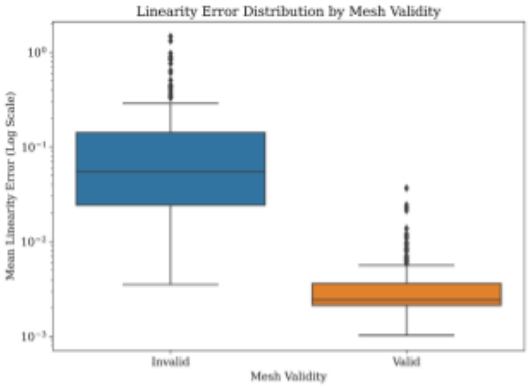


Figure 17: Box plot of mean linearity error (log scale) across meshes labeled as valid vs. invalid. Valid meshes concentrate at low mismatch values, while invalid meshes show significantly higher errors, confirming that the linearity-mismatch metric is a strong predictor of mesh validity.

Human Annotation Protocol. All meshes were visually inspected and annotated as either *valid* or *invalid*. A mesh was considered valid if it was smooth and resembled a high-fidelity car geometry without collapse or severe distortion. Invalid meshes were those with degenerate or implausible deformations. This produced a binary ground-truth dataset for evaluation.

Metric Validation. We used the mismatch score to predict mesh validity and compared it against human annotations:

- The ROC curve (Fig. 18, top left) shows excellent discriminative power with an area under the curve (AUC) of **0.989**.
- The precision–recall curve (Fig. 18, top right) yields an AUC of **0.990**, indicating reliable separation of valid from invalid meshes.
- Threshold analysis (Fig. 18, bottom) reveals that $\epsilon = 0.01$ provides a good tradeoff, achieving high recall while preserving precision.

Distributional Analysis. To further assess robustness, we examined the distribution of linearity errors across mesh validity labels. As shown in Fig. 17, valid meshes cluster at low mismatch values, while invalid meshes exhibit substantially higher errors, confirming that the safety metric is well aligned with perceptual mesh quality.

Discussion. These experiments demonstrate that the linearity-mismatch metric is a reliable quantitative proxy for mesh validity. Its strong agreement with human annotations justifies our use of $\epsilon = 0.01$ as the default safety threshold throughout this work.

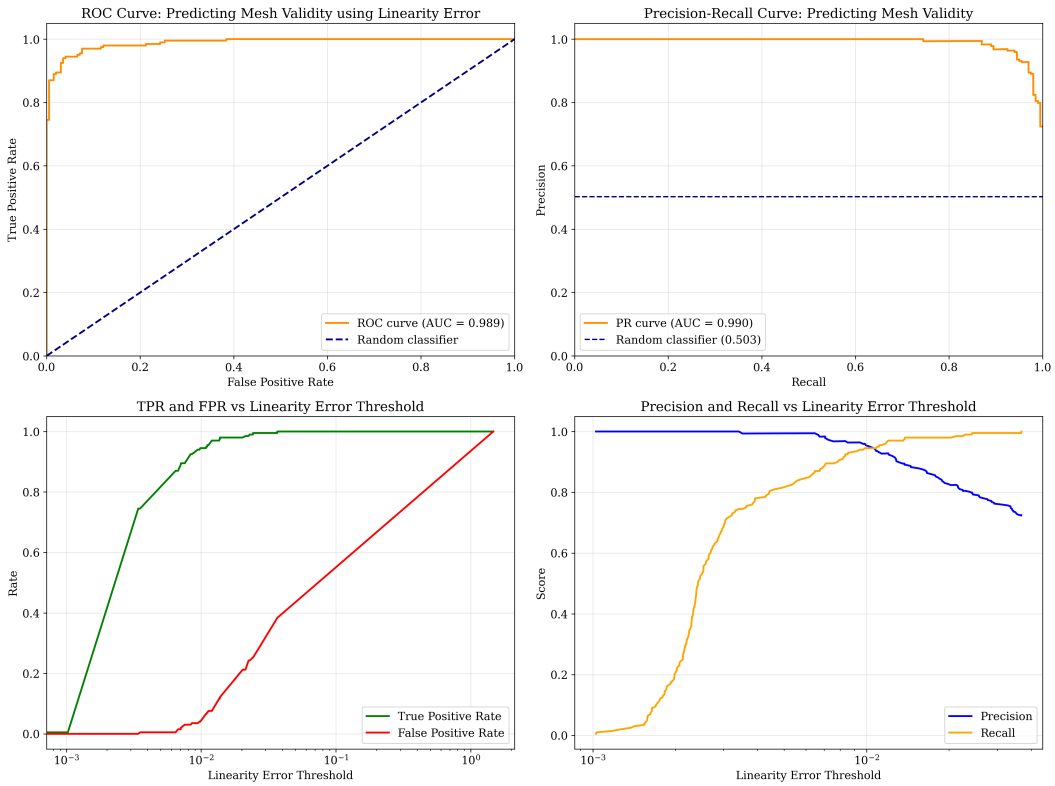


Figure 18: Validation of the linearity-mismatch safety metric against human-annotated mesh validity. Top left: ROC curve showing high discriminative power (AUC = 0.989). Top right: Precision–recall curve (AUC = 0.990). Bottom left: true positive rate (TPR) and false positive rate (FPR) as a function of linearity error threshold. Bottom right: precision and recall as a function of threshold. Together, these results confirm that the safety metric reliably predicts mesh validity, with $\epsilon = 0.01$ providing a good tradeoff between precision and recall.

I PARAMETRIZATION OF DRIVAERNET++ AND BLENDEDNET

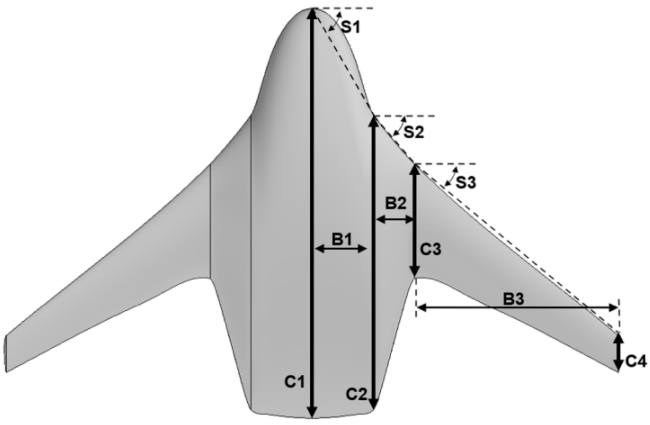


Figure 19: BlendedNet Parametrization Sung et al. (2025)

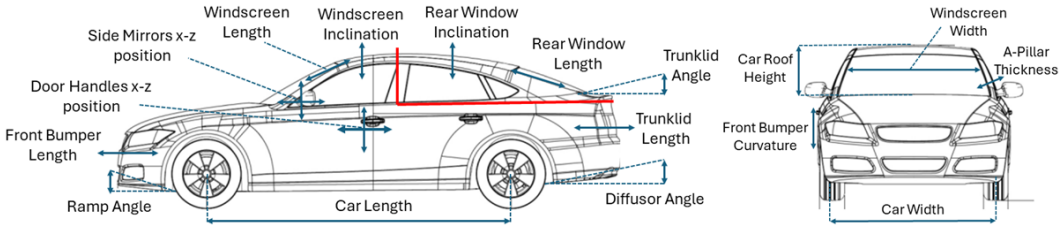


Figure 20: DrivAerNet++ Parametrization Elrefaie et al. (2024)

J LLM USAGE

Large Language Models (LLMs) were used to aid or polish writing. The authors reviewed and take full responsibility for the content.