# A Preliminary Study into the Conceptual Design of Aircraft using Simulation-Based Inference

**Aurelien Ghiglino**[1,2]*, **Daniel Elenius**[2], **Anirban Roy**[2], **Ramneet Kaur**[2], **Manoj Acharya**[2],

**Colin Samplawski**[2], **Brian Matejek**[2], **Susmit Jha**[2], **Juan J. Alonso**[1], **Adam D. Cobb**[2]

[1]Aerospace Design Laboratory, Stanford University
[2]Computer Science Laboratory, SRI International

## Abstract

In this paper, we explore the potential of likelihood-free inference for generating conceptual engineering designs of electric vertical take-off and landing (eVTOL) aircraft. In this preliminary study, our goal is to highlight two key findings: (1) simulation-based inference (SBI) can be effectively scaled to a large and complex multi-physics simulator composed of multiple interacting components, where each is governed by different physical principles; and (2) we introduce a new hierarchical diffusion-based SBI method that successfully captures the governing physical laws embodied in conceptual aircraft design. Using conditioning, we show physical laws are captured at both the component level and the system (full aircraft) level, underscoring promise for future research.

## 1 Introduction

Conceptual aircraft design is a complex engineering design problem that involves the interactions of many components representing a wide range of physics [18]. During this exploratory phase, domain experts face a high degree of design freedom and must apply their knowledge, supported by design tools, to configure parameters that align with performance goals and adhere to constraints. Therefore the challenge at the conceptual design phase is not in running expensive simulation models, such as computational fluid dynamics (CFD), but is in the narrowing down of the degrees of freedom while meeting the objectives and requirements for the aircraft. In this work, we show that simulation-based inference (SBI) is especially suited to accelerating this design process as it enables sampling from conditional distributions. We use SUAVE [15], a modular conceptual analysis and design tool for both conventional and unconventional aircraft configurations, to simulate the aircraft designs. This preliminary study focuses on electric vertical take-off and landing (eVTOL) aircraft simulated in SUAVE.

SBI performs Bayesian inference over simulators in scenarios where there is no analytical formulation of the likelihood, $p(\mathbf{x}|\boldsymbol{\theta})$ [5], where $\boldsymbol{\theta}$ corresponds to the input parameters to the simulator and $\mathbf{x}$ is the corresponding observation. Scaling SBI to high-dimensional simulators is a consistent challenge, where typical benchmark problems have focused on $\boldsymbol{\theta} \in \mathbb{R}^D$ for $D \approx 10$ [14], with a few recent notable exceptions [6, 7]. The most recent success in SBI has come from neural-based approaches, where each approach looks to compensate for the missing analytical likelihood, by either directly estimating the likelihood [17], directly modeling the posterior [16], or modeling the ratio of two likelihood functions [4, 20, 11, 3]. A more recent approach, that has achieved superior performance is the Simformer architecture [9]. This diffusion-based SBI approach learns the full joint distribution,

*Work completed while at SRI.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

with the advantage that one can condition on any subset of parameters when sampling. In engineering design, the requirements that drive the design are expressed in terms of both design parameters and design performance metrics. Therefore an architecture where one can condition on a subset of both observations and parameters is especially suitable for this kind of design.

Previous works in machine learning for engineering design have often focused on building benchmark datasets [2, 8, 12], or modeling specific components of a larger cyber-physical system, such as the CFD analysis of a wing [1, 19] or propeller design [21]. In this work we focus on the full conceptual design of a complete cyber-physical system that incorporates the non-linear interactions between multiple simulation modules, such as electrical and physical components. We develop a new physics-inspired architecture that maps its components to the structure of SUAVE, by combining multiple Simformer models. We then design an aggregation approach to predict the score used in the score-based diffusion sampling process. The result is an architecture that allows designers to condition on a mixture of input and output parameters of an eVTOL, and then use our SBI approach to identify a distribution of promising parameters that can be used for the next stage of the design process.

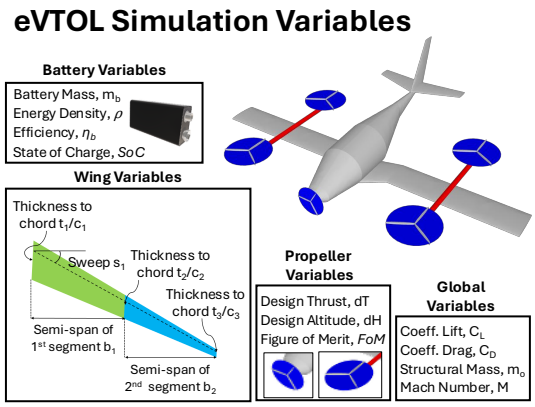



Figure 1: 18 eVTOL design variables.

## 2 Dataset: Conceptual aircraft design in SUAVE

We use SUAVE to generate a dataset of $10{,}000$ eVTOL aircraft that adhere to the design topology in Figure 1. For this study, we vary the wing geometry, the battery, and the propeller designs, while keeping the rest of the design fixed. This includes fixing the fuselage shape, the horizontal and vertical tailplanes, as well as the motors. We select the priors for each of the input design parameters using domain-knowledge of typical ranges for aircraft of this type, ensuring a large support to thoroughly explore the design space [18]. All the aircraft are evaluated for a fixed mission composed of a cruise segment of 20 nautical miles of length at a 1,000 ft altitude. For each aircraft, the structural mass and the battery mass are allowed to vary, altering the total mass. The observed performance variables are the cruise lift coefficient (non-dimensional $C_L$), drag coefficient (non-dimensional $C_D$), figure of merit (FoM, hovering efficiency of a rotor), battery efficiency ($\eta_b$), and State of Charge (SoC, the charge remaining in the battery after the mission). Overall, we have $\boldsymbol{\theta} \in \mathbb{R}^{13}$ and $\mathbf{x} \in \mathbb{R}^5$. In order to simulate measurement uncertainties, we add Gaussian noise for the Mach number $M \sim N(0.2, 0.01)$ and flight altitude $H \sim N(3000, 10)$ feet. Battery age is uniformly chosen to be between one and ten years, and the motor efficiency is chosen uniformly to be between $[0.95, 1.00]$.

## 3 Method: Hierarchical Simformer for multi-physics SBI

We now introduce our Hierarchical component-wise SBI approach for a multi-physics simulation model. We leverage the Simformer architecture as described in Gloeckler et al. [9], which directly models the joint distribution $p(\boldsymbol{\theta}, \mathbf{x})$ via a score-based diffusion model, where a transformer estimates the score [22]. Unlike in the original Simformer, we introduce a multi-physics, multi-component architecture, where the individual scores of each component are learned via their own Simformer and are then aggregated via a physically-inspired second stage. We explicitly use separate Simformer models for different components of the design. Our motivation for this architecture comes from the flexibility of modeling individual components both separately and jointly depending on the current stage of the design process. A further advantage of modeling in this component-wise manner is that it makes the task of incorporating the adjacency matrix into each component's architecture easier, whereas the original Simformer required the full adjacency matrix. For example, the interactions between the different variables for a single wing is simpler to derive compared to the full eVTOL.

To implement our new Hierarchichal Simformer architecture, we instantiate $C$ transformer models corresponding the $C$ design components. In our scenario, we set $C = 3$ corresponding to the wings,

the propellers, and the battery. We then need to aggregate all the scores from the individual Simformer models, $\boldsymbol{\alpha}_c$, to output the overall diffusion score, $\boldsymbol{\alpha}$, which must be the same dimension as the number of design parameters and observations. We cannot directly concatenate all the component-wise scores due to two subtleties. Firstly, some design variables may be shared across components. For example, the Mach number and aircraft mass may be important to the wing design, the propeller design and the battery design. Therefore, the scores corresponding to these common design variables must be recombined. We define a sparse matrix to perform this weighted sum described by matrix $\boldsymbol{\Phi}$. Secondly, some design attributes may be global properties or global performance characteristics not associated with a specific component, such as the charge remaining after the mission, which strongly depends on all components. In our architecture, these global variables are inputted separately and treated as their own component related to the whole design, $\boldsymbol{\alpha}_g$, and are then combined with other components using our aggregation approach.

We introduce two score aggregation approaches, Covariance Aggregation and Transformer Aggregation, which we will refer to as generally as $a(\cdot)$. (1) **Covariance Aggregation (CA):** We model the interrelationships between design variables across components using a lower triangular weight matrix. In this case, at training time, we learn a matrix $\mathbf{L}$ such that the output of the aggregator is $\mathbf{L}\mathbf{L}^\top\boldsymbol{\alpha}$. The hypothesis is that this approach will directly learn interpretable correlations between variables, which we see later in the experimental results. (2) **Transformer Aggregation (TA):** We pass all the concatenated scores into a final transformer. We summarize our aggregation approach here,

$$\boldsymbol{\alpha} = a\left(\boldsymbol{\Phi}\left[\boldsymbol{\alpha}_1(\mathbf{x}_1, \boldsymbol{\theta}_1), ..., \boldsymbol{\alpha}_c(\mathbf{x}_c, \boldsymbol{\theta}_c), ..., \boldsymbol{\alpha}_C(\mathbf{x}_C, \boldsymbol{\theta}_C), \boldsymbol{\alpha}_g\right]^T\right). \tag{1}$$

Finally, we also explore whether each component-wise architectures can be kept frozen when training the full architecture, with each aggregation method, or if jointly training the aggregator and the component Simformers might be superior.

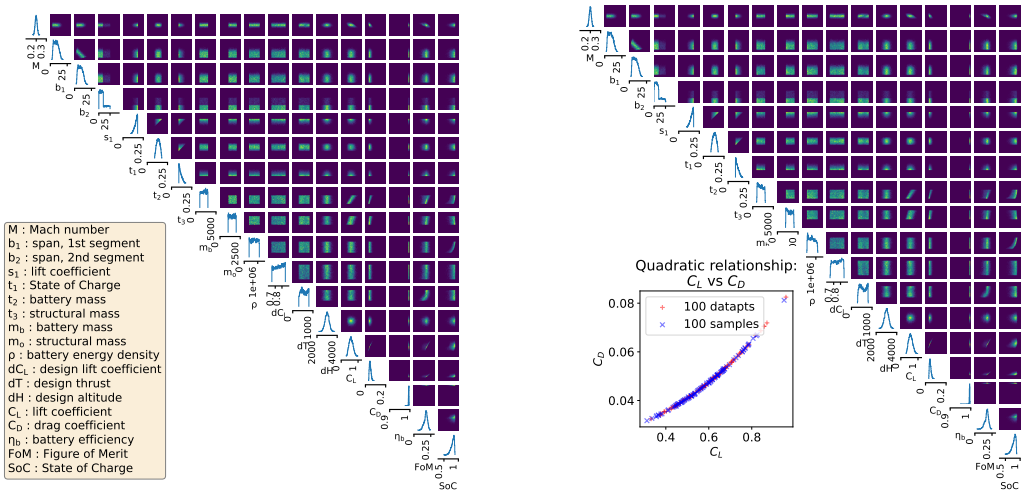
## 4 Results and discussion

**Experimental Details** Each individual component transformer follows the same architecture as in Gloeckler et al. [9], except that we set the number of layers to the number of variables for each component transformer, consistent with the graphical model. The largest component is given by the wing architecture with 9 layers. We fix the learning rate to $1 \times 10^{-3}$, and the train/validation split is 80% training, 20% validation, where we use the best validation to implement early stopping.

**Metrics** We compare our approaches using two metrics. The first is the maximum mean discrepancy (MMD), which describes the distance between the mean embeddings between the training data and samples of the learned joint distribution in a reproducing kernel Hilbert space [10]. The second metric is the classifier two-sample test (C2ST) [13], which trains a binary classifier to distinguish between the training data and a learned distribution. The closer the classifier behaves to a random generator (C2ST=0.5), the closer the learned distribution is to the training data. We report both the C2ST on the full joint, as well as the marginal C2ST's, where we take the mean and median C2ST across the individual marginals.

**Methods** We compare the original Simformer approach, with our covariance aggregated Simformer, which we label Covariance Simformer, and our transformer aggregated Simformer, which we label as Hierarchical Simformer. We also compare performance of when we freeze the weights of the component-wise transformers to when we jointly train the full architecture (the aggregation weights and the components).

**Results** Figure 2 shows the pair-wise plots of both the data distribution (left) and our Hierarchical Simformer (right). We first note that the Hierarchical Simformer distributions are qualitatively indistinguishable from the data distribution, showing that the architecture has successfully learned the joint distribution of the SUAVE simulator. The model has also captured some key correlations, such as the quadratic relationship between $C_L$ and $C_D$, with sampling being approximately 50x faster than the SUAVE simulation. Table 1 shows the results, where we see comparable performance between the baseline Simformer and Hierarchical approaches. Interestingly, we also see that the frozen approaches are able to achieve similar or better performance than the jointly trained models. This appears to be a promising step for future larger data distributions, where we would like to




(a) Data distribution

(b) Sampled Hierarchical Simformer distribution

Figure 2: Comparison of data distribution (a) and sampled joint from Hierarchical Simformer (b).

replace some of the expensive multiphysics simulations by cheaper component-only simulations. While neither approach significantly outperforms the other, we can conclude that our Hierarchical Simformer can achieve comparable performance to a single Simformer model. We can therefore assess that the architectural benefits of the componentwise approach - modularity, interpretability and the ability to re-use individual components - will not induce a significant degradation in performance.

**Learned Covariances** We evaluate the learned relationships between variables by looking at the $\mathbf{L}\mathbf{L}^\top$ matrix. We found that the learned matrix reveals the known interactions between components, as can be seen from the square blocks in Figure 3, and vehicle-level interactions, particularly between the Mach number and some of the other design variables, such as the masses, $C_L$ and $C_D$.

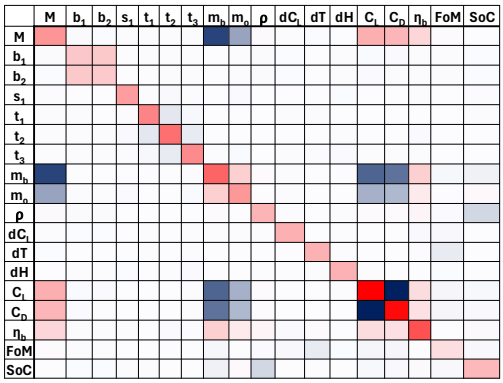
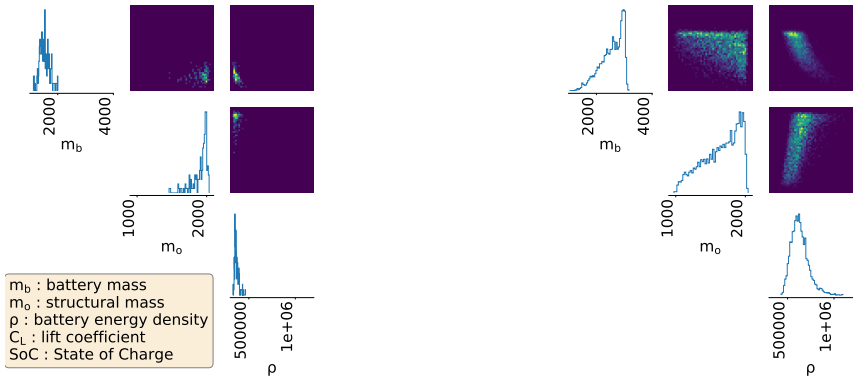



Figure 3: Learned covariances of Covariance Simformer. Red is positive correlation, and blue is negative correlation, with color intensity indicating correlation strength. Notice the negative correlation between weight and aerodynamic performance.

**Conditioning** We show the learned interdependencies in Figures 4a and 4b, by conditioning on the SoC and $C_L$. When we decrease the SoC of the aircraft for a fixed lift coefficient, we would expect the distributions to bias towards heavier aircraft. This trend is seen in the structural mass $m_o$ in Figure 4a. We also see that the lower SoC case is typically a result of lower battery energy densities, which means for a given battery weight, the battery can store less charge. Notice that battery weight $m_b$ is narrower and biased lower - increasing the battery weight increases both the weight of the aircraft and the battery capacity, and this study would suggest that for an eVTOL of the same design flying the same mission, the battery capacity dominates the weight. This preliminary analysis suggests that such techniques could be used to uncover broader design strategies and paradigms.




(a) $SoC = 0.7$ (b) $SoC = 0.8$

Figure 4: Conditional distributions for two different states of charge, while $C_L = 0.8$.

Table 1: Metrics describing the different hierarchical architectures when compared to using one large Simformer in terms of the maximum mean discrepency (MMD), joint C2ST (J. C2ST), mean marginal C2ST (Mean M. C2ST) and median marginal C2ST (Med. M. C2ST). (Fr. = Frozen.)

| **Algorithm** | **MMD** $\times 10^{-5}$ (↓) | **J. C2ST** (↓) | **Mean M. C2ST** (↓) | **Med. M. C2ST** (↓) |
|---|---|---|---|---|
| Simformer | 4.15 | 0.518 | 0.517 | 0.516 |
| Covariance Sim. (Fr.) | 8.88 | **0.497** | 0.632 | 0.515 |
| Covariance Sim. | 13.1 | 0.778 | 0.517 | 0.517 |
| Hierarchical Sim. (Fr.) | **2.40** | 0.686 | **0.507** | **0.506** |
| Hierarchical Sim. | 5.22 | 0.540 | 0.525 | 0.524 |

## 5 Conclusion

Our preliminary study into using SBI for eVTOL design has shown significant promise, by highlighting the ability of our hierarchical Simformer model to capture known physical interdependencies between design variables. Future work will build on this study by scaling to large dimensional distributions, exploring the method's sample efficiency and comparing against traditional aerospace surrogate models.

## 6 Acknowledgments

This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-23-C-0519 and HR0011-24-9-0424, and the U.S. Army Research Laboratory under Cooperative Research Agreement W911NF-17-2-0196. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force, DARPA, the U.S. Army Research Laboratory, or the United States Government.

## References

[1] 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.

[2] 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.

[3] Adam D Cobb, Brian Matejek, Daniel Elenius, Anirban Roy, and Susmit Jha. Direct amortized likelihood ratio estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20362–20369, 2024.

[4] Kyle Cranmer, Juan Pavez, and Gilles Louppe. Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv preprint arXiv:1506.02169*, 2015.

[5] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

[6] Maximilian Dax, Stephen R Green, Jonathan Gair, Nihar Gupte, Michael Pürrer, Vivien Raymond, Jonas Wildberger, Jakob H Macke, Alessandra Buonanno, and Bernhard Schölkopf. Real-time inference for binary neutron star mergers using machine learning. *Nature*, 639(8053): 49–53, 2025.

[7] Michael Deistler, Jan Boelts, Peter Steinbach, Guy Moss, Thomas Moreau, Manuel Gloeckler, Pedro LC Rodrigues, Julia Linhart, Janne K Lappalainen, Benjamin Kurt Miller, et al. Simulation-based inference: A practical guide. *arXiv preprint arXiv:2508.12939*, 2025.

[8] 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. *Advances in Neural Information Processing Systems*, 37:499–536, 2024.

[9] Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H Macke. All-in-one simulation-based inference. *arXiv preprint arXiv:2404.09636*, 2024.

[10] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

[11] Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free MCMC with amortized approximate ratio estimators. In *International conference on machine learning*, pages 4239–4248. PMLR, 2020.

[12] Seongjun Hong, Yongmin Kwon, Dongju Shin, Jangseop Park, and Namwoo Kang. Deepjeb: 3d deep learning-based synthetic jet engine bracket dataset. *Journal of Mechanical Design*, 147 (4):041703, 2025.

[13] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017.

[14] Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking simulation-based inference. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 343–351. PMLR, 13–15 Apr 2021. URL `https://proceedings.mlr.press/v130/lueckmann21a.html`.

[15] Trent W Lukaczyk, Andrew D Wendorff, Michael Colonno, Thomas D Economon, Juan J Alonso, Tarik H Orra, and Carlos Ilario. Suave: an open-source environment for multi-fidelity conceptual vehicle design. In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 3087, 2015.

[16] George Papamakarios and Iain Murray. Fast $\varepsilon$-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

[17] George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. PMLR, 2019.

[18] Daniel Raymer. *Aircraft design: a conceptual approach*. American Institute of Aeronautics and Astronautics, Inc., 2012.

[19] Yiren Shen and Juan J Alonso. Performance evaluation of a graph neural network-augmented multi-fidelity workflow for predicting aerodynamic coefficients on delta wings at low speed. In *AIAA SciTech 2025 Forum*, page 2360, 2025.

[20] Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 17(1):1–31, 2022.

[21] Harsh Vardhan, Peter Volgyesi, and Janos Sztipanovits. Fusion of ml with numerical simulation for optimized propeller design. *arXiv preprint arXiv:2302.14740*, 2023.

[22] Christian Dietrich Weilbach, William Harvey, and Frank Wood. Graphically structured diffusion models. In *International Conference on Machine Learning*, pages 36887–36909. PMLR, 2023.