

DATA-DRIVEN FAST NOISE PREDICTION OF AN UNCONVENTIONAL AIRCRAFT USING HIGH-FIDELITY SIMULATION DATA

Dominik Eisenhut[1] & Andreas Strohmayer[1]

[1]University of Stuttgart, Institute of Aircraft Design, Pfaffenwaldring 31, 70569 Stuttgart, Germany

Abstract

Climate change is pressuring the aviation sector to reduce its environmental impact. Novel aircraft concepts seek to improve efficiency by close integration of components. To capture benefits and to mitigate potential downsides early in the aircraft design process, fast aircraft-level noise and performance models are necessary because conventional handbook methods based on empirical correlations are often insufficient for this task. This work presents a machine learning surrogate trained on a computational fluid dynamics database varying three geometric and five operational parameters for the e-Genius aircraft. The simulations reproduce main spectral trends and show plausible aerodynamic behavior, although overall noise levels appear to be underpredicted and validation uncertainties remain. For noise prediction, simple neural networks are unable to capture the trends in the data, whereas graph-based models that account for spatial relations between monitoring points show promising results. By contrast, the current thrust and power surrogates are not yet sufficiently robust. To improve the models efficiently, an active-learning strategy is applied, for which the first six additional simulation points are selected in this work, using a greedy variance-distance algorithm. Overall, the approach shows promise for enabling future aircraft-level noise/performance trade studies of an unconventional aircraft similar to the e-Genius configuration.

Keywords: aircraft design, machine learning, noise prediction, electric-aircraft, aircraft level trade-off

1. Introduction

Aviation, like many other industries, is directly affected by global warming. As part of the transport sector, its contribution to greenhouse gas emissions is not negligible [1]. Therefore, many initiatives are in place to reduce the aviation sector's environmental impact, for example, ICAO's CORSIA program [2, 3] and the European Unions visions Flightpath2050 [4] as well as Fly the Green Deal [5]. Tackling these challenges, the aviation industry and research institutes investigate many novel, disruptive concepts. This includes battery-electric [6, 7] and hybrid-electric aircraft [8, 9, 10, 6] with respect to the powertrain, as well as boundary-layer ingestion [11] and distributed propulsion [12, 13] for a synergistic integration of propulsors in the airframe, or configurational changes like blended-wing bodies [14, 15] and box wings [16]. Apart from performance benefits such closely integrated concepts can lead to increased noise emissions [17, 18, 19, 20] which are also in focus of the aforementioned initiatives.

In all these cases standard handbook methods including empirical estimations may not work seamlessly within the performance modeling in overall aircraft design. This is particularly true when the required level of detail in component interactions cannot be determined analytically. At the same time, the direct integration of high-fidelity simulations into the design loop is computationally too expensive. In this case, machine learning (ML) algorithms seem to be a promising solution, as underlined by countless review papers [21, 22, 23, 24, 25, 26].

Other authors have already applied ML successfully to aircraft (subsystem) design tasks. Poggi et al. [27] investigate propeller blade optimization together with noise emissions trading

propulsive efficiency with directional overall sound pressure level (OASPL); similarly, for helicopter blades there is work by Thurman and Zawodny [28] also focusing on aerodynamics and aeroacoustics. A second study by Poggi et al. [29] focuses on noise of distributed propellers, again more on optimization at subsystem rather than aircraft level. Manghnani et al. [30, 31, 32] investigate a wing with one installed propeller and derive a ML surrogate to estimate tonal noise on the aircraft level. However, the approach is built on medium-fidelity methods, which require tweaking to fit experimental data. In addition, the underlying geometry is simplified to a rectangular wing without any propeller nacelle or remaining aircraft. With regards to performance, Tong et al. [33, 34] use ML successfully to predict thrust-specific fuel consumption for isolated conventional turbofan engines. Overall, these papers leave a gap, whereby highly integrated aircraft concepts cannot be tackled in high detail in performance and noise estimation for overall aircraft design. If possible, reusing existing models or simplifications should be the priority. In other cases, where the actual geometry or a more detailed interaction has to be investigated, high-fidelity simulations like computational fluid dynamics (CFD) and approaches that reduce the number of required data points seem practical.



Figure 1 – The battery-electric touring motor glider e-Genius built by the Institute of Aircraft Design from the University of Stuttgart. Photographed by Tobias Barth.

To bridge this gap, this work will aim to demonstrate the feasibility of a ML approach for a known aircraft where reference data based on experimental overflights is available. The aircraft selected is the e-Genius shown in Figure 1, a two-seated touring motor glider built by the Institute of Aircraft Design from the University of Stuttgart. Predicting the emitted aircraft noise is the focus, as this is difficult in the overall aircraft design process, while posing development risks due to certification, annoyance, and potential harm to people. As the reduction of noise often comes at a cost, e.g. decreased propulsive performance, these trade-offs should be enabled on the aircraft-design level. As unsteady CFD simulations are computationally expensive, an active learning (AL) approach will be pursued to iteratively extend an initial database and reduce the required number of samples [35]. For example, uncertainty can be used as a basis for this extension [35]. Karcher and Franz have shown for CFD simulations that such an approach can reduce the number of required samples by $35\,\%$ to $44\,\%$ compared to a space-filling approach (Halton sampling) [36]. Overall, this paper addresses the following two research questions looking first, on the models based on the initial database, and second, the first extension of said database:

1. What predictive accuracy and consistency can surrogate models for aircraft noise and propeller performance achieve based on an initial simulation database with geometric variations?

2. How can new data points be derived by different active-learning-oriented strategies?

The first question tackles the suitability of ML for aircraft design performance estimation, where the highest accuracy is not always necessary if trends are well preserved. The second question builds on the model created in the first step, deriving new data points based on the model's uncertainty to increase predictive accuracy. This work ultimately builds on the initial experimental-based surrogates described in Eisenhut et al. [37], using neural networks as the ML algorithm, which have proven to be the best option out of the initial models.

The remaining paper is structured as follows, first the materials and methods are introduced covering the full process from data sampling over preprocessing and model creation to the database extension with AL. Next, the different models are shown in the section results together with a verification and validation of the simulative database. This section is followed by the discussion, addressing potential reasons for the performance of the models and identifying the next points to be simulated in the AL process. Finally, a conclusion is drawn including an overview on future steps.

2. Materials and Methods

To reduce the required number of data points, while also aiming at improving the model's performance, the approach in this work differs slightly from a standard ML approach. The adapted methodology is shown in Figure 2. The process is divided into three steps: first, building an initial working model; second, utilizing AL to extend this original surrogate and enhance performance; and third, using a final test set to check the final model's performance. The AL process allows weaknesses or high uncertainties in the training or model generation process to be factored in by suggesting additional data points in these regions [35]. Apart from a purely space-filling approach, this allows knowledge to be utilized and expensive simulations to be performed in regions that are difficult for the model to predict [35]. To work properly, this approach requires a functioning model as a starting point, which is created with a space-filling approach. In this work latin hypercube sampling (LHS) [38, 39] is used. Based on this initial set of points (LHS A), different ML models are trained. For evaluation purposes, a test dataset is also required (LHS B), which the model never sees in training.

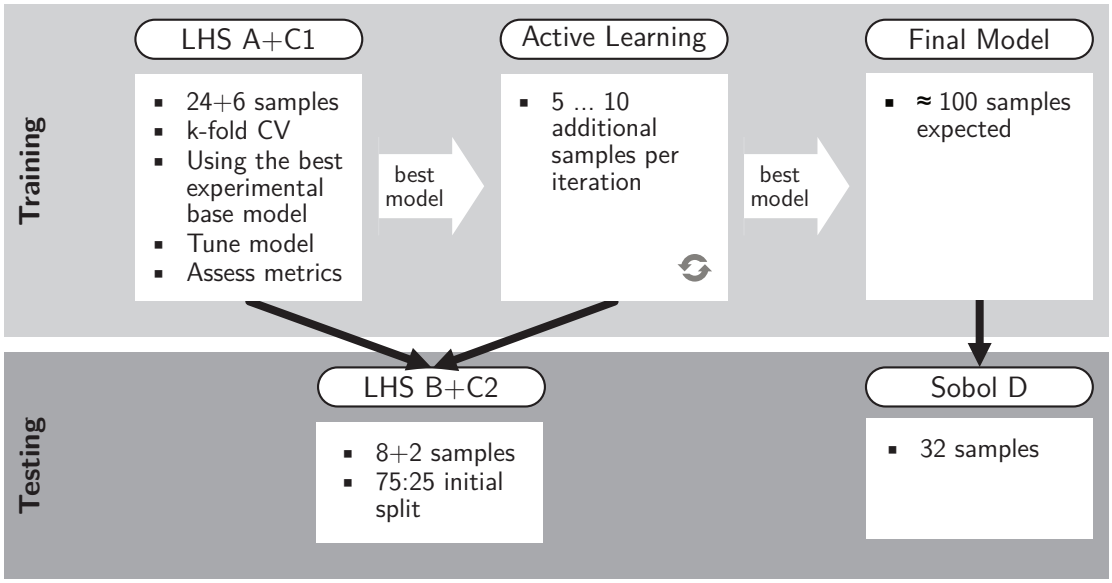


Figure 2 – Methodology used in this work for the ML approach.

Due to an error, the original design space has to be widened for the pitch angle by a third LHS (LHS C), which is randomly split into training and test samples. To allow for a trade-off between aircraft performance and noise emission, two models are trained based on the best-performing algorithm from the experimental surrogates developed in a previous step [37]. Based on the first models, the dataset will be expanded in an iterative AL approach, fine-tuning the model. For the final model, a second test dataset (Sobol D) is created to assess the model's performance, as otherwise information is leaked into the model. The size of this final test set depends on the number of points used for training the model, which is a direct result of the AL process. Depending on the effort to obtain additional data points relative to the potential performance gain (e.g. reduction of variance), the AL process will be stopped. Throughout this process, the initial test data will be used to monitor performance in comparison with training and validation metrics. In addition, it is used for early stopping in the AL step. Here, the best model from the architecture selection process with cross-validation (CV) is retrained n times to estimate model uncertainty.

For the CFD simulations, the commercial solver Simerics is used. The solver setup is compressible using the ideal gas law and the k-ε turbulence model—the simulation models including all geometrical and operational variations are provided within the research project WhisperProp by the partner CFD Consultants GmbH. In the steady simulations, the propeller is modeled with a moving reference frame, whereas a sliding mesh is used in the unsteady simulations. In the unsteady simulations, FW-H is used to extract acoustic information.
The remainder of this section is mainly grouped into the different steps of the process, starting with dataset sampling, followed by data preprocessing, moving to model selection and parameter tuning, and concluding with final assessment and database extension. Finally, ending with an overview on validation and verification.

2.1 Dataset Sampling

As this work also focuses on the underlying sampling to reduce the required number of simulations, it offers the privilege of setting up a database tailored to this problem. Unlike the standard engineering approach, where oftentimes a clustering happens at parameters or geometries 'interesting' to the engineer, we can deviate from 'human-informed' sampling and previous knowledge. In total, the number of samples per feature could be as low as 2 for linear regression [40]; however, for statistical purposes it should be higher [40], somewhere in the range of 10 to 20 samples per variable [40, 41]. Using ML could potentially reduce the number of samples per variable required. To optimally make use of this assumption, the dataset creation is twofold: first, creating a small, purely space-filling initial database, and second, expanding this with the help of an iterative, data-informed approach focusing not only on exploration (space-filling) but also on exploitation of model knowledge and uncertainty.
Without a readily available database, not only the sampling is open, but also the features and their respective ranges are to be defined. Here, the aim is to create a database with a mix of operational and configurational parameters, focusing on practical aircraft-level trade-offs. However, too drastic changes to the e-Genius geometry not only complicate the validation of results but also increase the complexity of preparing CAD models for simulation. Thus, more drastic changes had to be avoided by selecting parameters that allow easy modeling and potential validation and verification by flight testing and additional simulations.

Table 1 – Geometric and operational parameters varied for the simulation database and their defined design ranges. [1]defined relative to propeller diameter, [2]defined relative to vertical tail and the pivot point is at the motor mount attachment, [3]defined relative to propeller axis.

Parameters	lower	upper
x position propeller hub[1]	-0.3	0.0
Number of blades		2, 3, 4, 5
Propeller axis tilt angle[2]	$0°$	$9°$
Rotational speed	$1200\,\mathrm{rpm}$	$2300\,\mathrm{rpm}$
Propeller pitch[3]		
LHS A & B	$8.5°$	$32°$
LHS C	$0°$	$8.5°$
Sobol D	$0°$	$32°$
Indicated Airspeed	$100\,\mathrm{km\,h^{-1}}$	$200\,\mathrm{km\,h^{-1}}$
Altitude	$0\,\mathrm{m}$	$3000\,\mathrm{m}$
Angle of attack	$-3°$	$9°$

Table 1 lists the design ranges of all eight selected, three geometrical and five operational, parameters. Geometrical or configurational variations are the x-position of the propeller hub, the number of blades, and the tilt angle of the propeller axis. Operational variations are the rotational speed, propeller pitch, flight speed, altitude, and angle of attack (AoA). To relate these ranges to the original aircraft, the propeller axis tilt angle is $4.35°$ downward given relative to the vertical tail. The pitch range is $4.15°$ to $27.65°$ measured at the outermost position of the three-bladed propeller currently installed on the e-Genius. Furthermore, the pitch angle is always stated relative to the propeller axis,

which itself is influenced by the tilt angle. This results in different angles of attack at the propeller blades for the same pitch value when the tilt angle is different. All features are continuous values except for the number of propeller blades. The three ranges given for propeller pitch are due to the already mentioned error in the original definition (LHS A&B) and the subsequent extension (LHS C), the final test set (Sobol D) is then created over the full range. It has to be noted, that the sampling will result in non stationary flight points, and they are also not enforced during the simulations.
Since the required dataset size for a cost-effective model is not known a priori and the total number of samples depends on the intended accuracy of the resulting surrogate model, a sequential augmentation strategy is desirable to minimize computational effort. LHS requires a predefined sample size and does not inherently support incremental extension while preserving its design properties. Sequences like Halton [42, 43] or Sobol [44, 45] allow extensibility but do not work with previous knowledge. Therefore, an AL approach is employed to iteratively enrich the dataset, addressing space-filling and exploitation of knowledge on uncertain model regions. Based on the defined parameter space, an initial dataset consisting of 24 training samples (LHS A) and 8 test samples (LHS B) is generated using LHS. The initial training and test sets are extended by 6 and, respectively, 2 samples generated from a third LHS (LHS C) where two samples are randomly taken and added to the test set. This was done to take into account the difference in propeller blade pitch between the aircraft and simulations and to account for an early mix-up in the pitch definition. The decision to use two LHS designs for initial training and testing ensures a good representation of the available design space for both subsets. Especially with smaller sample numbers this can be important to allow the model to generalize. To extend the dataset with the help of AL, the model variance is estimated and new points are generated. The exact approach is covered in greater detail in a subsequent subsection.
Lastly, a final test set is created (Sobol D). The final test set is necessary because the initial test set is too small in relation to the later extended training set. Furthermore, its usage throughout the development within the AL approach leads to an unwanted bias in the model due to the evaluation of test metrics and early stopping decisions. In contrast to AL, where regions with high variance are targeted, a Sobol sequence, similarly to LHS, ensures good coverage of the overall design space [38, 39, 44, 45]. The switch from LHS to the Sobol sequence is made for two reasons. First, the final test dataset is methodologically independent from the whole training and initial test set, so no potential artifacts from one approach is repeated in both datasets. Second, the sequence allows data points to be extended if necessary. This again has two benefits. First, if the final dataset is bigger, the sequence allows additional test samples to be created. Second, simulations can already be performed in parallel to the generation of the ML model. This allows computational resources to be used for the test set when the current AL iteration is processed and additional points are selected, therefore no simulations for the database are running. However, it has to be checked in the end whether some test points ended up ‘too’ close to training points. In this case, the point has to be omitted, or this part considered with caution for the final metrics.
Overall, two transient simulations from LHS C did diverge, one of which is part of the training set and the other part of the test set. Therefore, the total number of training samples is 29 and the number of test samples is 9 for this work before the extension with AL.

2.2 Data Preprocessing

The simulation database feeds two surrogate models, one for installed propeller performance estimation and one for noise estimation. The raw data contains pressure and shear forces used for aerodynamic related assessment and the pressure signal of 1001 virtual microphones placed on a partial sphere used for noise assessment.
The performance data is evaluated by the project partner CFD Consultants GmbH and is provided by means of an Excel spreadsheet. It contains pressure, shear, and total forces in x-, y-, z-direction for the propeller blades alone, as well as for the overall aircraft without the propeller blades. For the first, power and torque are also given. Based on this data, lift, drag, thrust, and propulsive efficiency can be calculated. This is the basis for the data exploration and model development.
For the second aspect, noise, the toolchain to evaluate the simulations is built on the work on the experimental noise campaigns in 2024 [46] and 2025 [37]. On the one hand, for the database simulations a fixed number of propeller rotations is analyzed with a Fast Fourier Transform (FFT).

For the validation and comparison to the overflight measurements, on the other hand, a window length of $0.2\,\mathrm{s}$ is used in agreement with the experimental assessment [37].
Noise directionality is captured using a partial sphere already introduced in Eisenhut et al. [37] for the multi-output models. The selected approach results in a noise hemisphere with discrete points at a radius of $R = 20\,\mathrm{m}$. As a stationary climb or descent ($\gamma \neq 0$, with force equilibrium) is similar to stationary horizontal flight, the hemisphere is extended by 20° upwards in all directions [37]. Any sound propagation toward the ground is unlikely for higher angles, unless the flight path is modeled close to mountainous terrain where the aircraft is actually flying beneath its surroundings. However, this is not considered in this work. To ensure good sampling on this extended hemisphere, a Fibonacci sphere is used [47]. This approach results in improved spacing in contrast to selecting an equiangular grid [47]. Creating such a sphere with 1493 points and cutting off the top 70° of the polar angle, leaves 1001 points when explicitly adding the south pole [37].
To further simplify the approach and to calculate mainly the parts that cannot be performed analytically, the combined noise on this sphere is what the ML model should predict. This reduces the complexity for the ML model, as information on flight path, height, and more is not necessary. For example noise propagation can be calculated afterwards within the design environment. For this, RCAIDE [48, 49] is used—it is an open-source Python-based aircraft design tool also used at the Institute of Aircraft Design, providing already basic functionality for noise estimation. In addition, this approach allows the aircraft (performance) to be directly remodeled in the environment, without the need to modify the underlying noise surrogate model, as long as the configuration and parameters are covered by the model. Furthermore, overall trade-offs and optimization can be performed within the tool as well, utilizing both surrogate models.
Preprocessing of the noise simulation data is much simpler in comparison to the experimental overflight measurements [37]. There is no need to align timestamps of different measurement boxes and the aircraft data, to remove Doppler shift, or to calculate the actual emission angles when accounting for retarded time. In addition, no geometrical spreading or atmospheric damping needs to be removed, as all monitoring points lie on the same radius. For the database points a window of five propeller rotations is evaluated, using a Hann weighting together with a $50\,\%$ overlap between windows. In total, at least ten rotations are necessary to get three windows. The pressure signals in the simulations need some time to stabilize: therefore, first ten rotations are simulated, then the pressure signal is assessed manually, and additional rotations are performed for each case individually. Again, as for the experimentally based surrogates [37], OASPL is used for training. To obtain this value, the frequency range is limited to a range of one-third-octave bands. After data screening, the next step is scaling both the input features and targets. In this case, StandardScaler from scikit-learn [50] is used. Finally, the training data is split into the actual training and validation data using k-fold [50]. In general, fewer than five folds are computationally inexpensive at the cost of high bias, while more than ten folds become computationally expensive [51]. The splits ensure that the model is capable of learning the underlying trends and reduces the risk of overfitting [52, 53]. Good metrics across all folds are an indication of good generalization.

2.3 Model Selection and Parameter Tuning

After the first two steps, namely sampling and preprocessing, the actual models for performance and noise are trained. For the initial dataset, NNs are used, as they were the best-performing approach from the experimental surrogate [37], and a single model has to be selected to guide the AL process. The tuning of hyperparameters can be performed either manually or assisted by scripts using sampling to cover the parameter space. Variations are the number of layers and neurons, but also dropout layers and regularization. For model selection, a k-fold CV is used to ensure that the model performance is not due to a favorable or unfavorable selection of samples. Therefore, the training data is split into k folds, where each fold is used once as validation samples. Metrics for model assessment are the mean squared error (MSE), mean absolute error (MAE), and R^2 score.
Newly defined configurational parameters and the widened operational space have increased the dimensionality of the design space in comparison to the experimental models [37]. To account for this in the noise models, on top of the standard NN new approaches are considered. These should apart from the global parameters also consider interrelations between microphones on the partial sphere.

For example, closely located points oftentimes have similar noise levels. Therefore, graph-based NNs are tested, as they allow providing the global design and operational parameters, while additionally working with information on connections between different virtual microphones (nodes) on the sphere, and finally predicting noise on the whole sphere at once. The node edges (connections) are created based on a triangular mesh of the microphones, leading to a sparse representation. This graph-based NN can be coupled with convolutional layers to further aggregate information from neighboring nodes (graph convolutional network: GCN) [54], additionally considering the importance of each connection in a graph attention network (GAT) [55], or in a graph U-Net (g-U-Net) [56] adding pooling and unpooling layers. Pooling reduces the graph by selecting important nodes and unpooling restores the original graph [56]. For all three graph networks, the PyTorch Geometric library is used in version 2.7.0 [57], which builds on the PyTorch library used in version 2.11.0 [58]. Either the number of convolutional layers or the depth of the U-Net is varied alongside the number of hidden channels to find an optimal architecture. For the GCN and GAT, multilayer perceptron (MLP) encoder and decoder stages are also tested with varying numbers of hidden layers. In addition to MSE as a purely global loss over all microphones and samples, an additional loss is tested. This additional loss aims, apart from global minimization over all graphs, to penalize local deviations from the sound power level (SWL) within a single graph. To ensure comparable magnitudes of the loss terms, the MSE of the OASPL derived from SWL under a uniform spherical assumption is used, rather than the SWL MSE directly.

2.4 Final Assessment and Database Extension

Finally, the best initial models for noise and performance are selected based on performance metrics. Each model is then trained multiple times on the full training set with different random seeds, using the test data for early stopping. Then, candidates for future points are generated by a large LHS covering the whole design space. This ensures a good distribution along the design space and, due to the randomness, a non-deterministic result for each iteration of AL. Using a linear grid lattice with the same setup will result in the same points within each AL step and poor spacing in high dimensions. By predicting the candidates for each model, the model variance can be estimated, showing areas with high uncertainties. This information is used to define additional points for future simulations. This aims directly at the basic premise of uncertainty sampling as part of AL [35]. To ensure that these points do not result from a single cluster and thus prevent major clustering, different considerations can be made. using n points with maximum variance only is the opposite extreme to a purely space-filling approach. Thus, both aspects have to be considered, aiming for both exploitation of current knowledge and exploration of the design space. Together, this can help create a generalizing model with reduced computational cost. Monitoring how metrics evolve over iterations can help in deciding whether additional simulations are worth the effort. The present paper focuses only on the first AL step, building the initial model and deriving the first set of additional points. In summary, the general approach for active learning consists of the following steps:

1. Train n models, with the same architecture and setup, initialized with different random seeds.
2. Generate a sufficiently large candidate pool by sampling the design space and make predictions for all models.
3. Compute the prediction variance.
4. Optional: perform pre-selection by filtering out low-variance points.
5. Select next simulation points, either directly or with an intermediate step by identifying interesting clusters/regions first.

For the performance model variance can be used directly; on the other hand for the noise models, this is not the case, as the variance is known for each microphone and not at the simulation-level. Therefore, the results from all microphones need to be transformed into a metric on simulation-level. This is done by considering the mean and median variance per candidate. If the number of candidates

is higher, where the mean variance is above the median variance, the 95-percentile of the sample is used as a metric; whereas if the number is lower, the mean of the sample is used. This allows to focus on 'outliers' (mean $>$ median) or on improving the general prediction error (mean $<$ median), depending on which of the two occurs more frequently.
To identify new points and interesting clusters/regions beforehand, this work will discuss three approaches: namely a greedy algorithm, k-means clustering [50, 59], and a density-based approach with Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [60]. As the cost of unsteady simulation is high, the idea of building two separate databases tailored to the variance of each surrogate (noise and performance) is not pursued. Thus, the same database of unsteady simulations will be used and extended for both models together.

2.4.1 Greedy variance-distance algorithm

The simplest approach is selecting n points with the absolute highest variance within the candidate pool. However, this approach has a risk of choosing points within the same regions, resulting in unwanted clustering effects. To mitigate this, also the distance between points should be considered. The opposite, selecting points based on the distance alone, risks selecting points with low variance where additional simulations can have limited effect on improving the underlying model. Thus, the idea is a combined algorithm taking into account both the model variance and distance. To account for the different orders of magnitude, the actual features from which the distance is calculated are normalized with MinMaxScaler from scikit-learn, as well as the resulting distance and variance itself. A greedy algorithm uses a iterative approach that selects a local optimal solution within each step and does not account for any future consequences of this decision [61]. In this work Equation 1 allows a trade-off between variance and distance.

$$x_{\mathrm{n}} = \arg\max \left[\alpha \cdot V'(X) + (1-\alpha) \cdot D'(X,S) \right] \tag{1}$$

where $\alpha \in [0,1]$,
$V'(X) =$ normalized variance,
$D'(X,S) =$ normalized distance to the closest already selected training point.

It is important to note that $D'(X,S)$ explicitly includes not only selected points from the current iteration but also all training points already within the database. This ensures that not only newly created points are well spread out, but also that distance to already simulated points is considered. However, it can also be expected that points within the database are among the points with lower variance.
On the other hand, creating new training points equal to test or validation data will compromise them. To take this into account, an additional constraint is added to the algorithm in Equation 2, enforcing a minimal distance to be kept from the test sets. Based on the maximum distance in an 8-dimensional unit cube, $\sqrt{8} \approx 2.828$, the threshold is set to roughly $1\,\%$ of this distance as $\tau = 0.03$.

$$x_{\mathrm{n}} = \arg \max_{X : D'(X,P) \geq \tau} \left[\alpha \cdot V'(X) + (1-\alpha) \cdot D'(X,S) \right] \tag{2}$$

where $D'(X,P) =$ normalized distance to the closest test point,
$\tau =$ minimum required normalized distance to the closest test point,
$D'(X,S) =$ normalized distance to the closest already selected training point.

Normalization of variance and distance together with α ensures, that the maximum of the sum is also 1. The parameter α can be used to tailor the selection of new points either towards exploitation of current knowledge ($\alpha > 0.5$, favoring high-variance points) or towards further exploration of the design space ($\alpha < 0.5$, favoring high-distance points).
Risks with this approach can include selecting outliers in the data [35], that are distant from already selected points, but are not part of a region with high variance. In contrast, a smaller focus on exploitation allows to select points closer together by a higher α value, which risks a clustering of selected points if dominant high-variance regions are present in the design space. However, this

simple approach allows to select points even if no clear structure (e.g. clusters) is present within the candidates. To mitigate the clustering risk the following two approaches k-means clustering and HDBSCAN are investigated.

2.4.2 K-means clustering

The second approach is k-means clustering, where the goal is to find k clusters with their mean (centroid) [50, 59]. The number of clusters is predefined and can be selected, for example, based on the planned number of additional simulations per iteration. Also, different numbers can be tested to achieve a better overall clustering score. Each cluster is constructed from points that have the shortest distance to one centroid [50, 59]. This definition can lead to a bias in cluster size, resulting in spherical clusters of similar size [60, 62]. Partially, this can be offset by weights, for example, using the actual value of variance. However, if cluster size or density varies fundamentally, it could separate 'coupled' points or actual clusters [60].

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(w_i \cdot \left\| x_i - \mu_j \right\|^2 \right) \tag{3}$$

where $w_i =$ weight associated with data point i,
$x_i =$ data point i,
$\mu_j =$ mean of the samples in cluster (centroid) C,
$\|x_i - \mu_j\|^2 =$ inertia, or within-cluster sum-of-square.

Furthermore, this algorithm is influenced by random initialization (initial centroid locations), potentially making multiple runs beneficial, finally selecting the clusters with the lowest inertia [50, 59]. When varying the number of clusters to be defined, inertia is not the sole measure to be considered, as it will naturally decrease with an increase in number of clusters, as overall distance reduces. In this case, the silhouette score can be a better metric [63].
To define new points, for example, the point with the highest variance could be selected as an additional training point from each cluster. Another approach which can increase the exploration of the design space is using the above greedy function once within each cluster, starting with the cluster with the highest variance.
As k-means is known to struggle with unequally sized and shaped clusters [60, 62], which could be present in the variance data, a third approach HDBSCAN is considered.

2.4.3 Density approach: HDBSCAN

The third approach uses density information to extract clusters and addresses the risk of disjoint clusters of the k-means approach. This method is solely based on the geometrical information without any information on the actual variance, and thus also without weighting. Therefore, in a first step interesting points have to be preselected, for example, by removing all points with 'low' variance. Otherwise, there is no real clustering, as the sampling leads to well spread points and thus mostly uniform density. The variance filtering has to be broad enough to keep density information of high variance clusters, but strict enough to exclude regions of low variance and thus low interest.
The DBSCAN algorithm starts by defining 'relevant' points as 'core points'—meaning there are at least k points in the vicinity (distance ε) of X_i [60]. Next, a cluster is formed by all core points that are ε-reachable, points that are either connected directly or via points in between [60]. This approach directly tackles the previously described potential issues with the greedy algorithm or k-means, namely, defining the number of clusters and potential outliers that could be selected [60]. However, the selection of hyperparameters is difficult for DBSCAN, and additionally, densities still need to match over multiple clusters [60]. Therefore, HDBSCAN extends DBSCAN and addresses the aspect of different cluster densities by allowing flexible ε values [60]. By varying ε—thus calculating many clustering options with the DBSCAN approach—only stable clusters are ultimately selected [60]. Here, stability can be interpreted as how consistently a cluster is identified over varying distances.
Finally, the algorithm is mostly deterministic, apart from potential influences by data ordering [64], so only one run is required per dataset. As the number of clusters is not predefined and thus not

connected to the desired number of new points, n clusters of highest density or size could be selected to define a new point from. Now, in addition to the point with highest variance the one with highest density can be taken from each cluster or once again applying the greedy algorithm within each cluster additionally.

2.5 Validation and Verification

Before defining the reference cases used to validate the simulations an overview of the differences between the experimental results and the simulative setup, as well as potential error sources leading to uncertainties for the comparison, is given. These uncertainties can be grouped into three main areas: geometrical aspects, operational aspects, and differences in setup.

Starting with the geometrical aspects, especially the uncertainty of the pitch angle: First, the device used to measure the pitch angle on the real aircraft has an accuracy of $2°$; its scale allows to roughly estimate the angle between $0.5°$ to $1°$. Next, the selected outermost position on the blade to be measured can vary slightly between measurements, influencing the result. The CAD model also entails some uncertainties. It is derived from a 3D scan, where the outermost chord has to be estimated to align with the measurement. This is difficult, as the blade tip region is quite slender and the 3D scan cannot capture this area with the same accuracy as other portions of the blade. Additionally, the pitch angle is measured relative to the top of the vertical tail; in this way, the ground incline below the aircraft can be removed to make measurements comparable. However, this additional measurement introduces further uncertainties.

Moving to the second aspect, the operational values. Out of each overflight, a single rotational speed (propeller rpm) is selected, with which the simulations are performed. However, slight variations in airspeed and power during the overflight can change the rotational speed. Furthermore, the assumption of stationary flight could be false, also influencing airspeed and thus rotational speed at fixed pitch angle. In addition, a non-stationary flight also impacts assumptions on thrust and torque required to satisfy the flight condition. This also leads to uncertainties as thrust and shaft power can currently only be estimated due to the lack of measurement in flight. In the experiments, shaft power is estimated with higher confidence than thrust, as battery power is known and the electrical efficiencies of the powertrain can be characterized reasonably well. Thrust has to be derived based on the shaft power by applying the propeller efficiency, which introduces additional uncertainties. However, in the simulations the confidence level for thrust is higher, as it is mainly a pressure force that is easier to derive than the required shaft power, which in contrast is more dependent on viscous forces as well.

Finally, moving to the differences in setup for the simulations. Unlike the overflights, they are not performed with the ground in the vicinity of the aircraft. In the experiments, the aircraft was flying more or less in the influence of ground effect, which impacts both aerodynamics and acoustics. This can result in less drag and thus lower thrust and power required but can also impact the sound propagation. For simplicity, the simulations, however, are set up in the same way as the database variants, thus without any ground, leading to additional uncertainties.

Two cases from the experimental noise measurement campaign in 2025 are defined as reference cases to verify and validate the simulations (see Table 2). Within the simulations, the pitch angle is reduced by $1.35°$ as a effort to calibrate the results to the estimated required power/resulting thrust. This is done to account for uncertainties in the measurement of the pitch angle as well as the uncertainty in restoring the blade tip from the 3D scan. However, the offsets can actually differ between the two cases, as each measurement can have an individual error.

Table 2 – Selected reference cases from the 2025 e-Genius noise overflight campaign which are used to validate and verify the CFD simulations. In the experiment the first number describes the recording and the second the overflight, whereas for the simulations a single case numbering is used.

Experiment	Simulation	v_{TAS} / $\mathrm{km\,h^{-1}}$	P_{bat} / kW	n / rpm	β_{exp} / °	β_{sim} / °
1/11	994	177	42	1977	9.65	8.30
5/2	993	176	25	1200	22.15	20.80

3. Results

This chapter focuses on the results of this work. First, a validation and verification of the simulations and the resulting database is made. This is followed by the results of the different tested models for both surrogates, propeller performance and noise.

3.1 Validation and Verification

The main validation and verification for aerodynamics and thus installed propeller performance is performed by a project partner. However, a couple of quick checks are performed on the database cases by examining the correlation matrix in Figure 3.

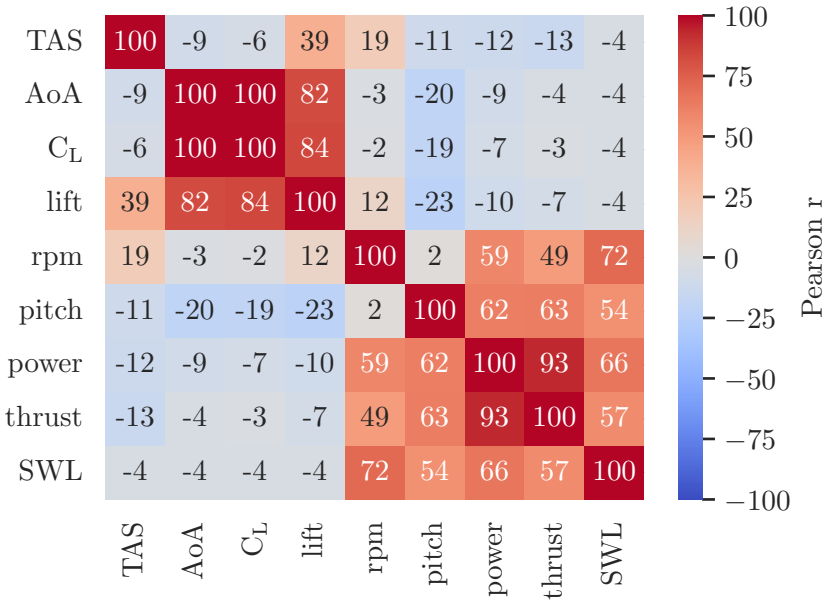


Figure 3 – Matrix of the linear correlations of the initial database cases. The values are the Pearson correlation given as a percentage.

The main point is the perfect positive correlation between AoA and the lift coefficient. This highlights the common trend that lift is often reliably estimated in CFD. In this case, the lift coefficient C_L can be used instead of AoA as an input for the surrogate models, which has the benefit that it is easier to obtain in early design performance estimation. Furthermore, it is more comparable, as weight is generally more reliable than AoA. Based on the initial simulations, the constants in Equation 4 can be determined.

$$C_L(\alpha) = \frac{\mathrm{d}C_L}{\mathrm{d}\alpha} \cdot (\alpha - \alpha_0) \tag{4}$$

$$\text{where} \quad \frac{\mathrm{d}C_L}{\mathrm{d}\alpha} = 0.104 \ \mathrm{deg}^{-1},$$
$$\alpha_0 = -2.277°.$$

Furthermore, there is no real correlation between the lift coefficient and TAS, which is expected because the influence of airspeed is removed. For the lift force, the expected increase in lift with airspeed is visible. Drag and its coefficient follow a quadratic law and is therefore not shown in the linear correlation matrix. Looking at both rotational speed and pitch angle, a positive correlation with the required shaft power as well as the generated thrust can be observed. Together, all four parameters have a positive correlation with the SWL, which is also expected. Finally, the four parameters that are inputs to the simulation (TAS, AoA, rpm, and pitch) have no or only marginal correlation with each other, which is also expected because of the well spread out sampling with LHS. The main correlations are among the results from the simulation (C_L, lift, power, thrust, and SWL).

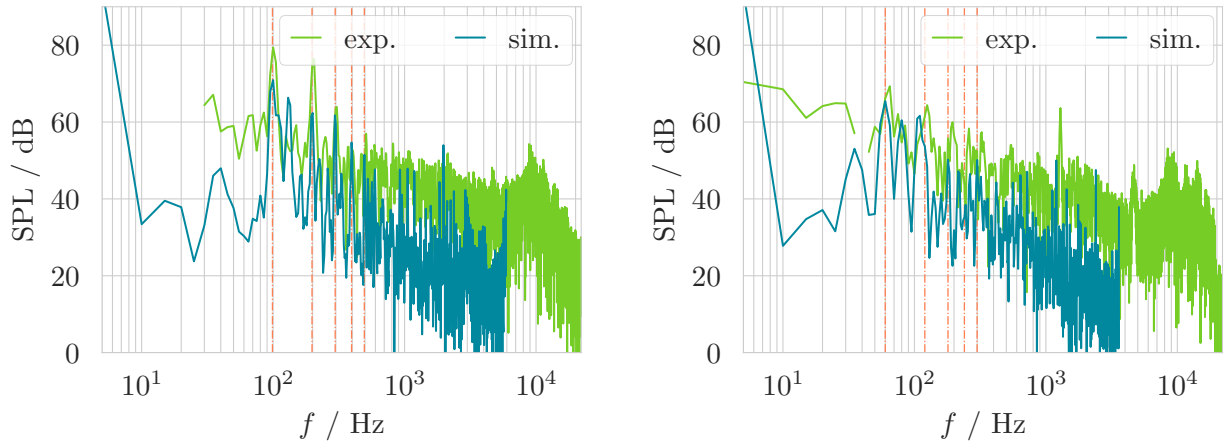
For the noise validation and verification, as a first step the SWL is compared to the experimental results. Both are set up similarly, using a Hann correction and a window length of $0.2\,\mathrm{s}$ [37]. As the experimental coverage of the noise sphere is smaller in comparison to the simulation partial Fibonacci sphere, the simulation is limited to the obtained directions from the overflight. In addition, points at shallow emission angles have proven unreliable in the experimental evaluation [37], therefore, the polar angle is limited to 40° below the horizon. The spherical partial surface of the experiments is always higher, because only microphones within the experimental area are considered for the simulations. To rule out a high influence of this difference in sphere area, first a normalized SWL is calculated assuming uniform coverage of a full sphere. Second, the simulation results are interpolated to the experimental locations and then the integration is made for the same sphere.

Table 3 – Comparison of the SWL for experiment and simulation. $\mathrm{SWL}_{\mathrm{norm}}$ is corrected for the difference in areas, stating SWL for a full sphere instead of the partial sphere. In all cases the radius is $20\,\mathrm{m}$ and frequency is limited to the $20\,\mathrm{Hz}$ up to $800\,\mathrm{Hz}$ one-third octave bands to allow comparison. The *i* cases are the simulation results interpolated to the experimental points on the sphere and then evaluated with the experimental based area weights.

Description	Case	Area / m^2	SWL / dB	$\mathrm{SWL}_{\mathrm{norm}}$ / dB
Exp.	1/11	1030	114.7	121.6
Sim.	994	702	104.9	113.4
Sim.	994i	1030	106.8	113.6
Exp.	5/2	960	105.3	112.4
Sim.	993	649	99.9	108.8
Sim.	993i	960	102.1	109.3

Table 3 shows the SWL for the reference cases in the simulation and the experiment. Overall, the simulations underestimate the total noise emission with a larger gap in the first reference case than in the second. However, the uncertainties in setup and for thrust and power remain. Furthermore, after calibration the simulation results are still slightly below the estimated values. Therefore, it can be expected that with a slight increase in power also the SWL will increase, thus reducing the gap. The different gaps between both reference cases can be due to different pitch offsets. This can be a result of the two measurements on ground with differing errors. A second option is the pitch angle for the 5/2 case, which had to be selected in flight timing the movement from the high-pitch stop to the desired pitch angle. This had to be made because the climb take-off performance were not satisfying enough to perform the whole flight in this setting. Overall, without in-flight thrust and torque measurements, there is no way to determine the true causes. Even with such measurement, uncertainties will remain due to newly introduced ones.

Apart from SWL, a second validation is performed by a spectral analysis on the virtual microphone directly under the aircraft. This corresponds to the experiment when the aircraft is directly above the microphone. In this case the distance between aircraft and microphone is the lowest in the experiments, and thus also atmospheric and geometric dampening. Figure 4 shows a comparison between experiment and simulation for the two reference cases. Due to the difference in sampling rate, the maximum frequency (Nyquist frequency) in the simulations is lower than in the experiment and differs between the two simulations. The locations of the BPF mostly align between simulation and experiment. For the second reference case in Figure 4b, it appears that the rotational speed in the experiment was already slightly higher than the value set for the simulation. Other than that, the peaks match the experimental results well in terms of position, but their magnitude is lower. This is in line with the findings for the SWL. Similarly, in Figure 4a the difference is higher, as noted previously. It must be noted that the experiment is only corrected by geometric spreading, but not atmospheric damping. The influence is small, but the distance of the aircraft during the overflight was lower than the $20\,\mathrm{m}$ radius. All aforementioned uncertainties still apply. Furthermore, attention is drawn once again to the necessary corrections for the experimental analysis. In each selected FFT window, the aircraft is moving a few meters, however, one fix point for the aircraft's location is defined and thus the direction of sound emission. This also adds to the uncertainties on experimental side.

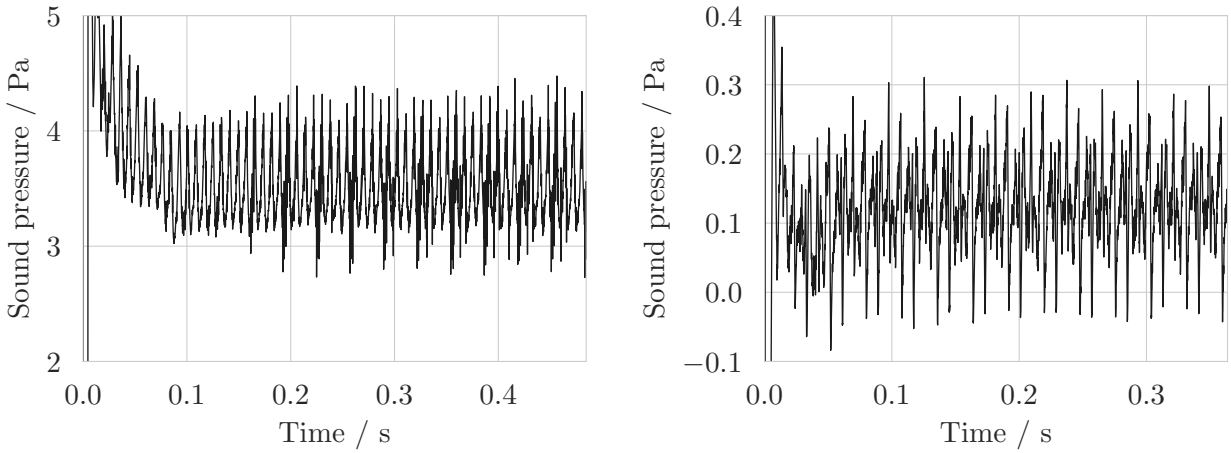


(a) Experiment 1/11, Simulation 994

(b) Experiment 5/2, Simulation 993

Figure 4 – Comparison of spectral analysis of experimental (green) and simulation (blue) cases. The red dot-dashed lines are the multiples 1 up to 5 of the BPF aligned with the value selected for the simulations. (a) Experiment 1/11, Simulation 994 (b) Experiment 5/2, Simulation 993.

In the process of fitting the reference case simulations, some simulations showed a rising trend in SPL with increasing frequency, starting somewhere between $800\,\mathrm{Hz}$ and $1100\,\mathrm{Hz}$. Simulations 993 and 994 do not show this tendency, whereas simulation 001 in Figure 4 does. The experimental results, however, show a mostly decreasing trend. For this reason, the OASPL for the simulation database will be calculated using the one-third octave bands from $20\,\mathrm{Hz}$ to $800\,\mathrm{Hz}$. The influence on the OASPL is marginal because it is mainly determined by the first multiples of the BPF.
In the following, the actual database cases are considered. Here, different window lengths can be considered, as the approach not necessarily has to align with the experimental assessment.

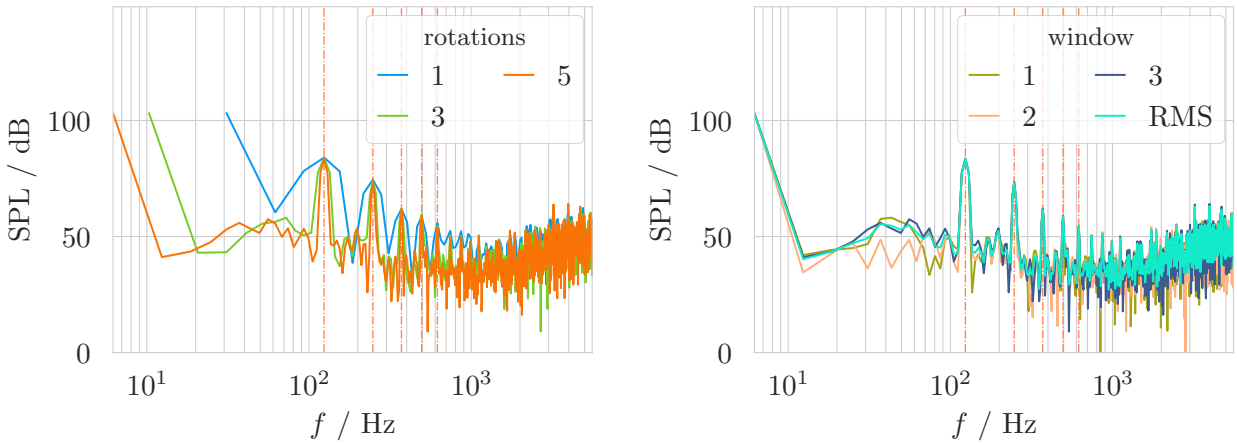


(a) Simulation 001

(b) Simulation 032

Figure 5 – Sound pressure signals for two simulations. (a) is a simulation where the start from steady results was not set up properly and (b) the setup was properly using the initial results. Here, the signal becomes periodic sooner.

Figure 5a shows the pressure signal for simulation case 001. Here, the setup was faulty, and the steady initialization was not used, leading to large spikes at the beginning and a longer time for the signal to stabilize. In comparison, Figure 5b shows a shorter initial period and smaller spikes. Apart from a longer simulation time, the results are not affected, and the initial portion is removed to obtain a ‘well established’ periodic signal that is then assessed via FFTs.



(a) Variation on number of rotations

(b) Variation of windows

Figure 6 – Spectral plot for simulation 001 (a) different FFT window lengths with n rotations per window and (b) three windows along the signal with $50\,\%$ overlap aligned with the end of the signal and RMS of all three windows. Again, the vertical dash-dot lines are the multiples of the BPF.

Figure 6a shows the expected trends when the window length is varied. If the number of investigated revolutions is reduced, the BPF peaks become broader due to the lower resolution associated with the shorter window length. This is especially visible in the broadband regime below the first BPF. However, the absolute value of each multiple of the BPFs remains constant, which is in line with other studies [18]. Overall, the OASPL is mostly unaffected as Table 4 shows.

Table 4 – OASPL for different number of rotations for a single microphone.

Rotations	1	3	5
OASPL / dB	103.30	103.25	103.25

Figure 6b shows different windows analyzed over the signal. Here, for all windows, five revolutions of the propeller are considered, and a $50\,\%$ overlap is used. The numbering is in chronological order, and the third window is aligned with the end of the simulation. In addition, the root mean square (RMS) of all three windows is plotted. Overall, good agreement across all windows and their RMS is visible in the plot which is also supported by the OASPL values in Table 5.

Table 5 – OASPL for the different windows for a single microphone.

Window	1	3	5	RMS
OASPL / dB	103.22	103.24	103.25	103.24

3.2 Performance Surrogate

As for performance estimation of the propeller in aircraft design both thrust and power are necessary, two separate NNs are trained, one for the estimation of each parameter. An extensive search across many hyperparameters is performed, variations in the numbers of layers, the number of neurons per layer, the use of dropout, and also L1/L2 regularization. The mean scaled metrics from a 5-fold CV are given in Table 6 for the two models with the lowest validation loss.

Table 6 – Mean of the 5-fold CV metrics of the performance NN's in the scaled frame.

Model	$\text{MSE}_{\text{train}}$	MSE_{val}
Power	0.455	0.359
Thrust	0.251	0.424

At first glance, the results look promising; however, the power model achieves a better loss on validation data than on the training data, which warrants further investigation. Therefore, the metrics for all five individual folds are also examined in Table 7. Some folds achieve very low MSE with a high R^2 score, whereas others MSE is higher and their R^2 scores are worse than those obtained by predicting the mean.

Table 7 – 5-fold CV metrics of each individual fold for the performance NN's in the scaled frame.

Model	Metric	Split	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Power	MSE	train	1.099	0.080	1.080	0.036	0.001
		val	0.640	0.067	0.310	0.628	0.109
	R^2	train	−0.012	0.923	0.095	0.953	0.999
		val	−0.197	0.921	−0.332	0.667	0.902
Thrust	MSE	train	0.025	0.409	0.792	0.037	0.003
		val	0.249	0.434	0.088	0.718	0.671
	R^2	train	0.975	0.632	0.329	0.960	0.996
		val	0.668	0.192	0.706	0.429	0.555

Switching to 10-fold CV improves the metrics for many folds at the cost that now only two or three samples are available for validation, while training is performed on the remaining 28 or 29 samples. This can be seen already as quite aggressive, as only $6\,\%$ to $9\,\%$ are withheld for validation and the model sees quite a big chunk of the overall database. However, some folds still exist in which the model did not really learn the underlying patterns. This can be seen both in excellent training scores paired with bad validation scores, but also in some folds where again validation metrics are far better than the training metrics.

3.3 Noise Surrogate

As expected, a simple NN for noise is not capable of handling the increased complexity of the additional design and operational space while also capturing directivity effects. Even the simplest models overfit heavily, memorizing the training data without any generalization. Thus, only the performance of the GCN, GAT, and g-U-Net is discussed further.

For all three options, two sets of input features are tested. First, all parameters of the design space are used, and second, the thrust and power obtained from the simulations are additionally provided. However, while in theory the information on thrust and power, which also indicates efficiency, should help the models, this is not observed in practice. The thrust and power from the simulations seem to have too high variance for the model to make use of this information given the low number of samples. This is supported by looking at the calculated value of efficiency, which can lie outside the physically meaningful range (e.g. $\eta_{\text{prop}} > 1$). Thus, efficiency is also not tested as an input feature.

In some cases, the modified loss function, which focuses on local energy-wise correctness, helped with lowering training and validation loss. However, some of the best models achieved better metrics with the simple MSE loss. Table 8 shows the best model of each of the three tested approaches according to the lowest validation loss. Whereas the GCN and GAT networks achieve their highest scores with the MSE loss, the g-U-NET benefits from the combined loss.

Table 8 – Mean of the 5-fold CV metrics of the graph-based noise NN's. Loss and MSE are given in the scaled frame while MAE is in the absolute frame in dB. g-U-NET is best with the combined loss function, whereas GCN and GAT performed better with the standard MSE loss function.

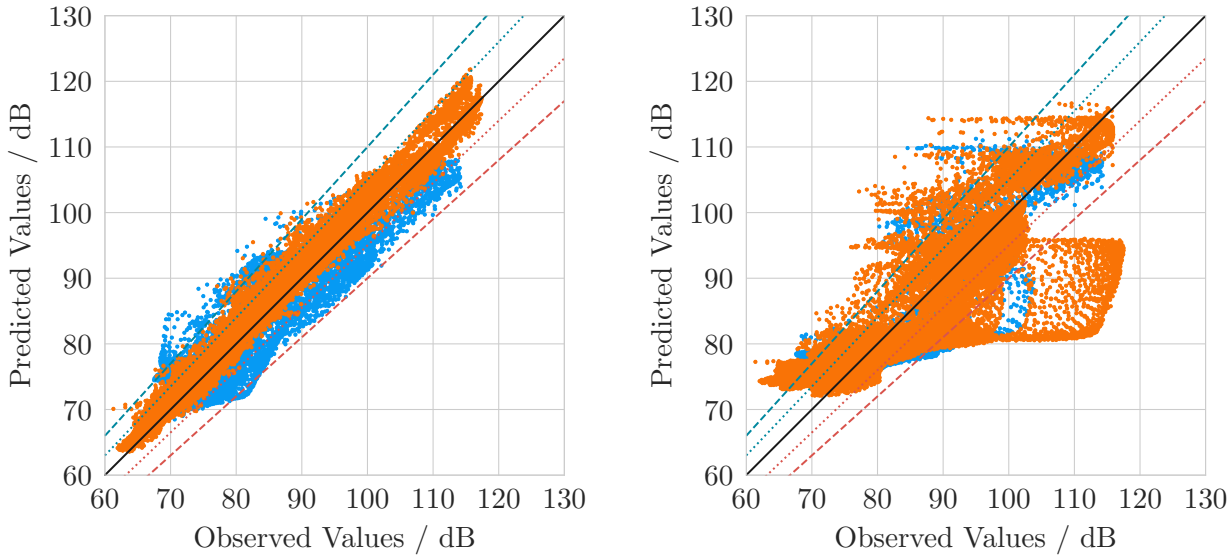
Model	$\text{Loss}_{\text{train}}$	Loss_{val}	$\text{MSE}_{\text{train}}$	MSE_{val}	$\text{MAE}_{\text{train}}$	MAE_{val}	$R^2{}_{\text{train}}$	$R^2{}_{\text{val}}$
GCN	0.077	0.176	0.077	0.176	2.448	3.950	0.923	0.824
GAT	0.053	0.198	0.053	0.198	2.010	4.215	0.948	0.802
g-U-NET	0.100	0.222	0.093	0.222	2.487	4.279	0.907	0.778

Looking at all the tested hyperparameters, no clear trend regarding complexity (depth and width of the models) can be found. Similarly, no clear trend is identified for the number of layers for the encoder and decoder MLPs. Overall, the GCN achieves the lowest validation loss, and its validation R^2 is also highest. In contrast, the GAT shows a greater tendency to overfit, as it achieves the lowest training loss and the highest training R^2 out of all models. Overall, the GCN model achieves an MAE for training of $2.45\,\mathrm{dB}$ and $3.95\,\mathrm{dB}$ for the validation data.
Table 9 shows the metrics for all five folds for the best-performing GCN model. It shows that some folds overfit more (Fold 1, 3, and 4) than others (Fold 2 and 5). However, this is partially expected, as each fold leaves out larger portions of the design space. Therefore, this architecture is retrained ten times in the next step on the full training set using different random initialization. Based on this, the model variance can be estimated.

Table 9 – 5-fold CV metrics of the individual folds for the best performing noise architecture GCN. The loss function is MSE which is given in the scaled frame, whereas MAE is in the absolute frame in dB.

Metric	Split	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
MSE	train	0.016	0.086	0.046	0.036	0.140
	val	0.347	0.132	0.100	0.145	0.153
MAE	train	1.130	2.774	2.004	1.713	3.435
	val	5.569	3.489	3.045	3.750	3.891
R^2	train	0.982	0.911	0.958	0.964	0.867
	val	0.745	0.884	0.840	0.852	0.796

When looking at the prediction vs. true plot for the retrained models in Figure 7, two main patterns can be identified. Figure 7a shows a good fit for both training and test data, with a slight tendency to overfit, as indicated by the wider spread of the test data (blue). On the other hand, Figure 7b shows wider regions where outliers are visible, both for training (orange) and test data (blue). This is attributed to the low number of samples, which leads to a higher variance across random seeds. However, the number of seeds leading to a good fit is much higher than the number leading to a poor fit with a ratio of roughly 8 to 2.



(a) Good fit with low variance

(b) High variance regions present

Figure 7 – Prediction vs. true plots for different randomly initialized models of the best performing GCN. (a) shows a good fit with low variance and (b) shows high variance regions are present with substantial outliers. Orange is the training data and blue is the test data. The dashed lines show $\pm 10\,\%$ deviation, the dotted lines show $\pm 5\,\%$ deviation, and the solid black line is the ideal prediction.

4. Discussion and Future Model Extension

Building on the results from the previous chapter, this section discusses both surrogates and the main factors influencing their performance. As before, the discussion distinguishes between the propeller performance model and the noise estimation model. Finally, the different AL approaches are assessed, leading to the selection of six additional points to be simulated in the first iteration.

4.1 Performance Model

The reasons for the unsatisfying performance of the thrust and power models in CV can be attributed to the k-fold split, where chunks of the training data, and thus parts of the whole design space, are withheld from the model. Due to the low number of samples and the high dimensionality of input features, this means that a rather large portion of the design space is missing. This is supported both by the 10-fold CV and by retraining the model on the full training set while using the test set for early stopping. In this case, the overall metrics are improved, at the cost of either a small test set or foregoing prior validation of the model architecture. Grouping the training samples into clusters and using stratified folds is also not helpful, as LHS produces well-spread samples without a cluster-like structure.

With respect to the first research question, the results should be used with caution, as the available database size is not yet sufficient, which restricts the generalizability of the models. Consequently, the performance model is not considered mature enough to be used in the AL process addressed in the second research question. If the metrics improve with additional samples generated based on the noise AL extension, the performance models can be considered in the future.

Furthermore, additional steady simulations can be used to improve the model. These are roughly 100 times faster than the unsteady ones. Using a model or function to transform steady thrust and power into unsteady results, or simply using the steady results alone, can be sufficient to bridge the gap and allow the necessary trade-offs at the aircraft level.

Both approaches can be tested in the future; however, the correlations currently do not indicate strong connections with either the blade count or the inflow angle. Both can cause errors in the steady simulation with a moving reference frame. Thus, it remains to be seen whether a suitable model can be found to transform steady metrics into unsteady results, and a trade-off must be made as to whether this additional uncertainty is more acceptable than the error in the steady simulations.

4.2 Noise Model

The results show that the current model is indeed learning the underlying relationships. Looking at the first research question, all metrics, MSE in the scaled frame and the MAE in absolute values, show a good general result and R^2 shows that mostly trends are well reserved. However, the number of samples is still small, leading to higher variance across models. For the AL process, this is necessary, as it allows a reasonable extension of the database. However, this general higher variance also means that the points with highest variance are well spread out and distributed across the design and operational space. Therefore, the two approaches focusing on clustering are not particularly meaningful, resulting either in essentially one cluster or a high number of clusters with only a few candidates each. For this reason, only the results from the greedy function are shown. Still, in later iterations, the clustering approaches could become meaningful.

For the greedy approach in Equation 2, two metrics are important: first, the distance between candidate points and selected points, and second, the variance of the model at each candidate point. Distance is equally distributed in the beginning, and with each additionally selected point, the region around that candidate is avoided as the remaining candidates are now closer to an already selected point. Looking at the variance, points are well distributed, and even within areas of high variance (percentiles) low-variance points can still exist. However, there are certain parameters that have regions in their design space where higher variance is concentrated. For example, higher variance at the lower end of the design space can be found for the x-position of the propeller, IAS, and altitude. By contrast, for the blade count, tilt angle, AoA, rotational speed, and pitch angle, higher variance is found at the upper end of the design space.

Table 10 shows the results of the greedy function. $\alpha = 0.55$ is selected such that variance (exploitation) has relevance for all selected points, while strong clustering (limited exploration) is prevented. Both aspects can be seen, first, high variance points are selected ($V'(X) > 0$) while distance is also of interest ($D'(X,S) > 0$). For the points five and six the distance actually is more important than variance. In general, it should be noted that the normalized distance follows a standard distribution, whereas the normalized variance is strongly concentrated on lower values. Thus, a stronger focus on exploitation also partially balances out these differences in the distribution. Due to the iterative approach, the normalized distance changes in every iteration because the newly selected point is removed from the candidate set and added to the selected points. The normalized variance remains constant, as the models do not change over the greedy iterations. Additional simulations are performed collectively after selecting n points.

Table 10 – Results of the greedy function $f(V', D', \alpha)$ for each selected point and contributions of normalized variance $V'(X)$ and distance $D'(X,S)$. The selected α value is 0.55.

	$f(V', D', \alpha)$	$V'(X)$	$D'(X,S)$
1	0.719 16	1.000 00	0.375 90
2	0.661 29	0.806 53	0.483 78
3	0.541 94	0.722 67	0.321 04
4	0.537 59	0.506 75	0.575 27
5	0.489 42	0.243 36	0.790 17
6	0.485 03	0.368 75	0.627 14

Table 11 shows the points selected in the first AL step. It can be seen that the previously described regions with higher variance, for example higher pitch and rpm, received more focus (exploitation). Still, in most cases, the overall design space is covered globally. Other than that propeller tilt is a bit constrained, as well as altitude and the x-position of the propeller; however, the latter two had a smaller impact on the overall noise. In subsequent iterations, these areas of the design space may receive more focus, both because of potentially higher variance and because of their greater distance from already selected points.

Table 11 – Selected Points in the first active learning step using the greedy variance-distance approach.

	x-pos	n_{blades}	$prop_{tilt}$ / °	rpm	pitch / °	IAS / $\mathrm{km\,h^{-1}}$	Altitude / m	AoA / °
1	−0.283	5	6.9	2292	17.2	119.16	541	4.8
2	−0.292	4	7.2	1858	28.7	114.84	161	1.4
3	−0.299	5	8.3	2128	20.5	111.60	933	0.0
4	−0.299	2	8.1	1884	15.9	100.08	1295	7.1
5	−0.250	3	8.8	1604	31.8	192.96	310	6.0
6	−0.135	5	7.1	2110	25.5	140.76	41	−1.0

Overall, with a look at the second research question, it can be said, that the greedy approach currently is the best option out of the three investigated ones. It allows the dataset to be extended both based on exploration and exploitation of the design space. The other two approaches can be helpful in the future, if the model variance results in actual clusters in the design space.
Finally, some confidence in the AL approach was gained in the process of extending the database from LHS A&B with LHS C. Here, an improvement in model quality was visible, both for the k-fold metrics and the prediction vs. true plot. Furthermore, a speed-up seems possible, as the simulations seem to stabilize early within the first five rotations and the difference of multiple windows in negligible. Therefore, in the future ten rotations could be enough when evaluating a single FFT window. This saves additional simulation time and allows for more iterations in the AL process and thus further improvement.

5. Conclusions and Outlook

For unconventional, tightly integrated aircraft configurations, a gap exists in aircraft design for estimating performance-related metrics, particularly aircraft noise, for which standard low-fidelity or empirical models are often inadequate. High-fidelity approaches such as CFD can provide accurate results, but are computationally too expensive to be used within the design loop, making surrogate models that are capable of fast prediction necessary.

The main contribution of this work is a data-driven fast workflow to predict noise and performance figures of the e-Genius aircraft. A simulation database based on high-fidelity CFD is created by varying three geometric and five operational parameters, enabling aircraft-level trade-offs. The simulations reproduce the main spectral trends observed in experimental overflight measurements; however, overall noise levels appear to be underpredicted. At the same time, uncertainties in geometry, operating conditions, and the validation setup remain, and without direct thrust and torque measurements these discrepancies cannot yet be resolved completely.

Initial surrogate models are created for both noise directivity and propeller performance. With respect to the first research question, graph-based models outperform simple neural networks for noise prediction, achieving promising predictive accuracy while preserving relevant trends. The mean absolute error based on the initial model is $2.45\,\mathrm{dB}$ for the training data and $3.95\,\mathrm{dB}$ for the validation data. In contrast, the current thrust and power surrogates are not yet sufficiently reliable. In both cases an improvement during the successive active learning steps is expected. Both for accuracy (MSE, MAE) and underlying trends (R^2).

Based on these findings, only the graph convolutional network noise model is retrained to guide the active learning process and extend the database while limiting the number of expensive simulations. Addressing the second research question, the greedy variance-distance strategy proves most practical for the first active learning iteration, whereas clustering-based approaches are not yet meaningful because high-variance regions are still broadly distributed and do not form distinct clusters.

The next steps are to run the simulations for the newly selected points and continue with multiple iterations in the active learning process. The observed improvement over successive iterations will indicate the trade-off between further model refinement and simulation cost. In parallel, the performance surrogates can be improved either by benefiting from the extended unsteady database or by building a dedicated database from significantly cheaper steady simulations.

6. Author Contact Details

Dominik Eisenhut
+49 711 685 61996
eisenhut@ifb.uni-stuttgart.de

Author contributions: Concept, D.E.; methodology, D.E; analysis, D.E.; data acquisition and processing, D.E.; writing, D.E.; review and editing, D.E. and A.S.; supervision, A.S.; project administration, D.E.; funding acquisition, D.E.

Acknowledgment: This Project is supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag.

Parts of this work was performed on the computational resource bwUniCluster funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC.

Finally, the authors would like to acknowledge the fruitful discussions with all partners involved in the project WhisperProp, especially with Max Kassera from yasAI UG in terms of the machine learning aspects.

7. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] European Environment Agency. Greenhouse gas emissions from transport in Europe. https://www.eea.europa.eu/en/analysis/indicators/greenhouse-gas-emissions-from-transport, November 2025.

[2] International Civil Aviation Organization. Resolution A41-22: Consolidated statement of continuing ICAO policies and practices related to environmental protection — Carbon Offsetting and Reduction Scheme for International Aviation (CORSIA).

[3] International Civil Aviation Organization. CORSIA Implementation Plan.

[4] European Commission. Directorate General for Research and Innovation. and European Commission. Directorate General for Mobility and Transport. *Flightpath 2050 Europe's Vision for Aviation*. Publications Office, LU, 2011.

[5] European Commission. Directorate General for Research and Innovation. *Fly the Green Deal: Europe's Vision for Sustainable Aviation.* Publications Office, LU, 2022.

[6] University of Stuttgart, Institute of Aircraft Design. E-Genius. https://www.ifb.uni-stuttgart.de/en/research/aircraftdesign/mannedaircraft/e-genius/.

[7] Reynard de Vries, Rob E Wolleswinkel, Daniel Rosen Jacobson, Maarten Bonnema, and Sebastian Thiede. BATTERY PERFORMANCE METRICS FOR LARGE ELECTRIC PASSENGER AIRCRAFT. In *34th Congress of the International Council of the Aeronautical Sciences*, Florence, Italy, 9–2024.

[8] Dominik Eisenhut, Andreas Bender, Niclas Grüning, Jonas Mangold, and Andreas Strohmayer. Comparison of Actual Hybrid-Electric Flights with a Digital Twin in a Preliminary Aircraft Design Environment. *Aerospace*, 12(5):401, May 2025.

[9] J Mangold, D Eisenhut, F Brenner, N Moebs, and A Strohmayer. Preliminary hybrid-electric aircraft design with advancements on the open-source tool SUAVE. *Journal of Physics: Conference Series*, 2526(1):012022, June 2023.

[10] D Eisenhut, J Mangold, N Moebs, F Brenner, and A Strohmayer. Case study on hybrid-electric aircraft designs enabled by an enhanced SUAVE version. *Journal of Physics: Conference Series*, 2526(1):012019, June 2023.

[11] Arne Seitz, Anaïs Luisa Habermann, Fabian Peter, Florian Troeltsch, Alejandro Castillo Pardo, Biagio Della Corte, Martijn Van Sluis, Zdobyslaw Goraj, Mariusz Kowalski, Xin Zhao, Tomas Grönstedt, Julian Bijewitz, and Guido Wortmann. Proof of Concept Study for Fuselage Boundary Layer Ingesting Propulsion. *Aerospace*, 8(1):16, January 2021.

[12] electra.aero. Electra.aero | Electra Completes World's First Hybrid-Electric eSTOL Aircraft Flight. https://www.electra.aero/news/worlds-first-hybrid-electric-estol-flight, November 2023.

[13] electra.aero. Electra.aero | Electra Reveals Design for EL9 Ultra Short Hybrid-Electric Aircraft. https://www.electra.aero/news/electra-reveals-design-for-el9-ultra-short-hybrid-electric-aircraft, November 2024.

[14] J. Schneider, S. Blessing, J. Zander, and A. Strohmayer. IDENTIFICATION AND CHARACTERIZATION OF THE WING REFERENCE AREA OF BLENDED WING BODY AIRCRAFT FOR ASPECT RATIO DEFINITION. In *Research & Education in Aircraft Design*, page 19, Warsaw, Poland, 6–2024.

[15] Philipp Hansmann and Eike Stumpf. Blended Wing Body Design in UNICADO, July 2025.

[16] University of Stuttgart, Institute of Aircraft Design. Project DOXWING. https://www.ifb.uni-stuttgart.de/forschung/flugzeugentwurf/AnalytischerFlugzeugentwurf/#id-c22a3d19-0, July 2024.

[17] Robin Wickersheim, Manuel Keßler, Julia Sembowski, Dustin Bongen, and Jorge Gomes. Numerical Investigation and Validation of Noise Sources of a Distributed Propulsion System. *AIAA Journal*, pages 1–13, November 2024.

[18] Robin Wickersheim. *Noise Mechanisms and Installation Effects of Distributed Propulsion Systems: A Simulative and Experimental Work*. PhD thesis, Universität Stuttgart, Stuttgart, 2025.

[19] Jack Barker, Alessandro Zarri, Julien Christophe, and Christophe Schram. *Numerical Investigation of Tonal Noise Emissions from Propeller-Wing Aerodynamic and Acoustic Interactions*. June 2023.

[20] Feroz Ahmed, Carlos Ramos-Romero, Antonio Martinez, and Mahdi Azarpeyvand. *Boundary Layer Ingestion Ducted Fan: Aeroacoustic and Psychoacoustic Insights*. May 2024.

[21] Steven L. Brunton, J. Nathan Kutz, Krithika Manohar, Aleksandr Y. Aravkin, Kristi Morgansen, Jennifer Klemisch, Nicholas Goebel, James Buttrick, Jeffrey Poskin, Adriana W. Blom-Schieber, Thomas Hogan, and Darren McDonald. Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning. *AIAA Journal*, pages 1–26, July 2021.

[22] Yiqun Dong, Jun Tao, Youmin Zhang, Wei Lin, and Jianliang Ai. Deep Learning in Aircraft Design,

Dynamics, and Control: Review and Prospects. *IEEE Transactions on Aerospace and Electronic Systems*, 57(4):2346–2368, August 2021.

[23] Soledad Le Clainche, Esteban Ferrer, Sam Gibson, Elisabeth Cross, Alessandro Parente, and Ricardo Vinuesa. Improving aircraft performance using machine learning: A review. *Aerospace Science and Technology*, 138:108354, July 2023.

[24] Jichao Li, Xiaosong Du, and Joaquim R. R. A. Martins. Machine learning in aerodynamic shape optimization. *Progress in Aerospace Sciences*, 134:100849, October 2022.

[25] Michael J. Bianco, Peter Gerstoft, James Traer, Emma Ozanich, Marie A. Roch, Sharon Gannot, and Charles-Alban Deledalle. Machine learning in acoustics: Theory and applications. *The Journal of the Acoustical Society of America*, 146(5):3590–3628, November 2019.

[26] Behzad Amirsalari and Joana Rocha. Recent Advances in Airfoil Self-Noise Passive Reduction. *Aerospace*, 10(9):791, September 2023.

[27] Caterina Poggi, Monica Rossetti, Giovanni Bernardini, Umberto Iemma, Cristiano Andolfi, Christian Milano, and Massimo Gennaretti. Surrogate models for predicting noise emission and aerodynamic performance of propellers. *Aerospace Science and Technology*, 125:107016, June 2022.

[28] Christopher Thurman and Nikolas Zawodny. Aeroacoustic Characterization of Optimum Hovering Rotors using Artificial Neural Networks. In *Proceedings of the Vertical Flight Society 77th Annual Forum*, pages 1–10, Virtual Conference, May 2021. The Vertical Flight Society.

[29] Caterina Poggi, Monica Rossetti, Jacopo Serafini, Giovanni Bernardini, Massimo Gennaretti, and Umberto Iemma. Neural network meta–modelling for an efficient prediction of propeller array acoustic signature. *Aerospace Science and Technology*, 130:107910, November 2022.

[30] Jatin Manghnani, Vincent Domogalla, Roland Ewert, Lothar Bertsch, and Jan W. Delfs. A First Principle Based Approach for Prediction of Tonal Noise From Isolated and Installed Propeller. In *30th AIAA/CEAS Aeroacoustics Conference (2024)*, Rome, Italy, June 2024. American Institute of Aeronautics and Astronautics.

[31] Jatin Manghnani, Roland Ewert, and Jan W Delfs. Predicting Propeller Tonal Noise with AI trained First-Principle Models: A Novel Methodology. In *DAS | DAGA 2025*, Copenhagen, 2025.

[32] Jatin Manghnani, Roland Ewert, and Jan Delfs. Towards Data-Driven Aeroacoustic Modeling to Predict Propeller Installation Noise: A First-Principle Based Approach. In *AIAA AVIATION FORUM AND ASCEND 2025*, Las Vegas, Nevada, July 2025. American Institute of Aeronautics and Astronautics.

[33] Michael T. Tong. Using Machine Learning to Predict Core Sizes of High-Efficiency Turbofan Engines. Technical Report E-19727, August 2019.

[34] Michael T. Tong. Machine Learning-Based Predictive Analytics for Aircraft Engine Conceptual Design. Technical Report E-19871, September 2020.

[35] Burr Settles. *Active Learning*. Number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer Nature Switzerland, Cham, reprint of original edition edition, 2022.

[36] Niklas Karcher and Thomas Franz. Adaptive sampling strategies for reduced-order modeling. *CEAS Aeronautical Journal*, 13(2):487–502, April 2022.

[37] Dominik Eisenhut and Andreas Strohmayer. Fast Data-Driven Noise Prediction for an Aircraft in Unconventional Configuration Using Flight Test Data. *Aerospace*, 13(3):292, March 2026.

[38] M. D. McKay, R. J. Beckman, and W. J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):239, May 1979.

[39] The SciPy Community. LatinHypercube — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy-1.17.0/reference/generated/scipy.stats.qmc.LatinHypercube.html#scipy.stats.qmc.LatinHypercube.

[40] Peter C. Austin and Ewout W. Steyerberg. The number of subjects per variable required in linear regression analyses. *Journal of Clinical Epidemiology*, 68(6):627–636, June 2015.

[41] Frank E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer Series in Statistics. Springer International Publishing, Cham, 2015.

[42] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90, December 1960.

[43] The SciPy Community. Halton — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy-1.17.0/reference/generated/scipy.stats.qmc.Halton.html.

[44] I.M Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, January 1967.

[45] The SciPy Community. Sobol — SciPy v1.17.0 Manual. https://docs.scipy.org/doc/scipy-1.17.0/reference/generated/scipy.stats.qmc.Sobol.html.

[46] Dominik Eisenhut, Andreas Bender, and Andreas Strohmayer. Overflight Noise Measurements of a crewed All-Electric Flying Demonstrator. In *11th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, Roma, 2025.

[47] Álvaro González. Measurement of Areas on a Sphere Using Fibonacci and Latitude–Longitude Lattices. *Mathematical Geosciences*, 42(1):49–64, January 2010.

[48] Matthew Clarke, Emilio Botero, Jordan Smart, and Rachael Erhard. RCAIDE_LEADS, January 2025.

[49] University of Illinois, Urbana-Champaign. RCAIDE L.E.A.D.S. https://www.rcaide.leadsresearchgroup.com/, February 2025.

[50] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[51] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, New York, NY, 2013.

[52] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, Orlando, FL, 2023.

[53] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc, Beijing [China] ; Sebastopol, CA, second edition edition, 2019.

[54] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017.

[55] Shaked Brody, Uri Alon, and Eran Yahav. How Attentive are Graph Attention Networks?, January 2022.

[56] Hongyang Gao and Shuiwang Ji. Graph U-Nets, May 2019.

[57] Matthias Fey and Jan E Lenssen. FAST GRAPH REPRESENTATION LEARNING WITH PYTORCH GEOMETRIC, 2019.

[58] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 929–947, La Jolla CA USA, April 2024. ACM.

[59] Scikit-learn User Guide: K-means. https://scikit-learn.org/stable/modules/clustering.html#k-means.

[60] Leland McInnes and John Healy. Accelerated Hierarchical Density Based Clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42, November 2017.

[61] Thomas H. Cormen, Charles Eric Leiserson, Ronald Linn Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts London, England, fourth edition edition, 2022.

[62] Yordan P. Raykov, Alexis Boukouvalas, Fahd Baig, and Max A. Little. What to Do When K-Means Clustering Fails: A Simple yet Principled Alternative Algorithm. *PLOS ONE*, 11(9):e0162259, September 2016.

[63] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, November 1987.

[64] Scikit-learn User Guide: DBSCAN. https://scikit-learn.org/stable/modules/clustering.html#dbscan.