# Latent Space Correlation for Interpretable Airfoil Parameterization Using Variational Autoencoders

Swannet, K.; Varriale, Carmine; Doan, Nguyen Anh Khoa

**Publication date**
2024

**Document Version**
Final published version

**Published in**
34th Congress of the International Council of the Aeronautical Sciences (ICAS)

**Citation (APA)**
Swannet, K., Varriale, C., & Doan, N. A. K. (2024). Latent Space Correlation for Interpretable Airfoil Parameterization Using Variational Autoencoders. In *34th Congress of the International Council of the Aeronautical Sciences (ICAS)*
https://www.icas.org/ICAS_ARCHIVE/ICAS2024/data/papers/ICAS2024_0484_paper.pdf


# LATENT SPACE CORRELATION FOR INTERPRETABLE AIRFOIL PARAMETERIZATION USING VARIATIONAL AUTOENCODERS

Kilian Swannet[1,2], Carmine Varriale[1,2] & Nguyen Anh Khoa Doan[1,3]

[1]Delft University of Technology, Delft, 2629 HS, The Netherlands
[2]Department of Flow Physics and Technology, Flight Performance and Propulsion section
[3]Department of Flow Physics and Technology, Aerodynamics section

## Abstract

This paper introduces an approach for parameterizing airfoil geometries using a Variational Autoencoder (VAE) with a focus on achieving a low-dimensional and interpretable model. The primary focus is to facilitate efficient use in design optimization environments by capturing essential airfoil features in a minimal number of latent dimensions. To address the black-box nature of VAEs and enhance interpretability, a correlation analysis is performed to uncover the relationships between the airfoil properties and these inferred latent dimensions. Key to this research is the incorporation of both geometric and aerodynamic properties in this analysis, enabling the generation of airfoils with desired aerodynamic characteristics through manual tuning of the latent vector by a designer. The method is evaluated using the extensive UIUC airfoil database, which includes a diverse range of airfoil categories. The VAE is trained on airfoil surface coordinate points, and the generated output geometries are refined using a composite Bezier curve to smooth out local imperfections. Results demonstrate that the VAE can successfully extract and parameterize key airfoil features using a limited number of interpretable latent parameters. These parameters show clear correlations with geometric and aerodynamic airfoil properties, providing a practical and understandable parameterization model that facilitates the intuitive generation of new airfoil designs through smooth interpolation of the training data.

**Keywords:** Machine learning, Variational Autoencoders, Interpretability, Explainable AI, Parameterization

## Nomenclature

| | | |
|---|---|---|
| $\vec{\cdot}$ | = | Vector variable |
| $x$ | = | Input |
| $x'$ | = | Generator output |
| $z_i$ | = | Latent parameter |
| $\mu$ | = | Mean |
| $\sigma$ | = | Standard deviation |
| $\varepsilon$ | = | Random value |
| $\mathcal{N}(\mu,\sigma)$ | = | Gaussian distribution |
| $\mathcal{U}(a,b)$ | = | Uniform distribution between $a$ and $b$ |
| $L_{\text{train}}$ | = | Training loss |
| $L_{\text{recon}}$ | = | Reconstruction loss |
| $D_{\text{KL}}$ | = | Kullback–Leibler divergence |
| $\beta$ | = | Weight factor for KL divergence |
| $t_{\text{max}}$ | = | Maximum thickness |
| $x_{t_{\text{max}}}$ | = | Chordwise max thickness location |
| $c_{\text{max}}$ | = | Maximum camber |
| $x_{c_{\text{max}}}$ | = | Chordwise max camber location |
| $z_{\text{u}}$ | = | Upper crest value |
| $x_{z_{\text{u}}}$ | = | Upper crest location |
| $z_{\text{l}}$ | = | Lower crest value |
| $x_{z_{\text{l}}}$ | = | Lower crest location |
| $\kappa_{z_{\text{u}}}$ | = | Upper crest curvature |
| $\kappa_{z_{\text{l}}}$ | = | Lower crest curvature |
| $R_{\text{LE}}$ | = | Leading edge radius |
| $\theta_{\text{LE}}$ | = | Leading edge angle |
| $\theta_{\text{TE}}$ | = | Trailing edge angle |
| $\gamma_{\text{TE}}$ | = | Trailing edge wedge angle |
| $\alpha$ | = | Angle of attack |
| $C_{L_\alpha}$ | = | Lift curve slope |
| $C_{M_\alpha}$ | = | Moment curve slope |
| $C_{L_0}$ | = | Lift coefficient for $\alpha = 0$ |
| $C_{M_0}$ | = | Moment coefficient for $\alpha = 0$ |
| $\alpha_{0L}$ | = | Angle of attack for $C_L = 0$ |

## 1. Introduction

Deep learning techniques have paved the way for innovative parameterization and design generation methodologies in various fields, including engineering and aerospace design [1–3]. Among these techniques are generative algorithms like variational autoencoders (VAEs) and generative adversarial networks (GAN's), which have proven capable of flow field estimation [4, 5] and aerodynamic shape generation [6–9].

This study uses VAEs to infer the key geometric and aerodynamic properties of airfoils [10]. VAEs were chosen over GANs for their more stable training process [11] and ability to explicitly infer the latent distributions of features in a dataset [12]. However, a big challenge when using any deep learning model is its "black-box" nature. Interpretability is crucial as it fosters trust in the model, enabling engineers to understand and intuitively interact with it to make informed design choices. The concept of explainable AI (XAI) [13] is therefore central to this study, as it seeks to demystify the decision-making processes of such complex models. To this end, VAEs have the additional benefit of latent space disentanglement which helps reduce the dimensionality of the latent representation and further facilitates interpretability [14, 15]. This concept is further explained in section 2.

While it is possible to enforce interpretability by manually imposing user-defined properties on the latent vector [8, 9], the present study instead aims to leverage the inherent feature extraction capabilities of the VAE, thereby minimizing user-imposed bias on the inferred latent parameters. By achieving a clearer understanding of how a VAE's latent dimensions correlate with both geometric and aerodynamic airfoil characteristics, the goal is to provide a low-dimensional yet robust and intuitive parametrization model. This enhances the usability of VAEs in design environments and aligns with the broader objectives of XAI, fostering greater trust and adoption of AI-driven solutions in engineering design optimization.

Following the introduction of the variational autoencoder in section 2, section 3 explains the training data setup and choice of some hyperparameters. section 4 describes how the output of the generator network is processed, and section 5 covers the validation of the trained model. Finally, section 6. dives into the latent space evaluation with a visual inspection of the generated geometries and a correlation analysis to quantitatively determine the relationships between airfoil properties and latent dimensions.

## 2. The Variational Autoencoder

Generative modeling is the process of creating a probabilistic, low-dimensional representation of data, called a latent representation, which allows for the generation of new data points through sampling [16]. An Autoencoder (AE) is a type of neural network designed to replicate its input data by extracting and regenerating features [17]. It consists of an encoder and a decoder. The encoder network compresses the input data vector $\bar{x}$ into a latent vector $\bar{z}$, capturing the essential characteristics of the data. The decoder network reconstructs the original input from these latent features, producing an output vector $\bar{x}'$ that approximates the input. This reconstruction process minimizes the loss of important information during compression. AEs can be viewed as a nonlinear extension of Principal Component Analysis (PCA). While PCA reduces dimensionality by finding orthogonal axes (principal components) that maximize variance, AEs use neural networks to learn complex, nonlinear mappings. This allows AEs to capture more intricate patterns and relationships in the data, outperforming PCA in terms of feature representation [17–19].

Variational Autoencoders (VAEs), introduced by Kingma et al. [14], enhance AEs by adding a probabilistic layer to the encoder, making the framework probabilistic instead of deterministic. This architecture, depicted in Figure 1, accounts for the inherent uncertainty and variability in real-world data.

The VAE encoder captures the posterior probability distribution of the latent variables given the input data, $p(\bar{z}|\bar{x})$, represented by the mean $\mu$ and standard deviation $\sigma$ (see Equation 1). This allows the network to infer the likely values of the latent variables, rather than deterministic values. The decoder does the opposite and models the likelihood distribution of the data given the latent variables, $p(\bar{x}|\bar{z})$. To sample the latent parameter values from the learned inferred distributions, the reparameterization trick is used as shown in Equation 2, where $\odot$ signifies element-wise multiplication. It mimics

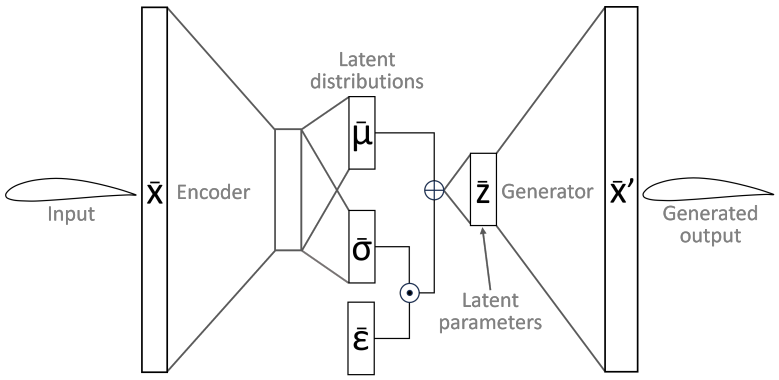



Figure 1 – Variational autoencoder network architecture.

sampling from the distribution defined by $\mu$ and $\sigma$ by sampling a random value $\varepsilon$ from a standard Gaussian and multiplying it with the standard deviation to obtain a value at a random distance from the mean.

$$\bar{\sigma}, \bar{\mu} = f_{\text{enc}}(\bar{x}) \tag{1}$$

$$\bar{z} = \bar{\mu} + \bar{\varepsilon} \odot \bar{\sigma}, \quad \text{where } \bar{\varepsilon} \sim \mathcal{N}(0,1) \tag{2}$$

The decoder uses the sampled latent vector to reconstruct the input, preserving the information contained in the original input as closely as possible, as defined in Equation 3. The probabilistic representation of the data allows the VAE to generalize and smoothly interpolate the data it was trained on. This allows the decoder to also produce a variety of new data points from a provided latent vector, unseen during training. Due to this capability, the decoder part of a VAE is also referred to as a generator.

$$\bar{x}' = f_{\text{dec}}(\bar{z}) \tag{3}$$

Training a VAE involves balancing reconstruction accuracy with shaping the latent space to follow a specific probabilistic distribution. The training loss function is defined by Equation 4. The first element is the reconstruction loss, which measures the difference between the original and reconstructed data. The reconstruction loss is computed using the relative mean squared error, as specified in Equation 5. This method measures the difference between the original input data and the reconstructed data relative to the original values. To ensure stability and avoid division by zero, the $y$ ordinates are offset upwards by one. The relative mean squared error was chosen to emphasize lower values, helping to shape the leading and trailing edges where points are concentrated, and small irregularities significantly affect airfoil surface smoothness.

$$\mathrm{L}_{\text{train}} = \mathrm{L}_{\text{recon}} + \beta \cdot \mathrm{D}_{\text{KL}} \tag{4}$$

$$\mathrm{L}_{\text{recon}} = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{\bar{x} - \bar{x}'}{\bar{x}} \right)^2 \tag{5}$$

$$\mathrm{D}_{\text{KL}} = \frac{1}{2} \sum \left( \bar{\mu}^2 + \bar{\sigma}^2 - \log(\bar{\sigma}^2) - 1 \right) \tag{6}$$

The Kullback-Leibler (KL) divergence term, shown in Equation 6, is included in the loss function to align the learned latent distribution with a standard Gaussian distribution [14]. This regularization encourages the model to produce a smooth and continuous latent space, promoting better generalization and reducing overfitting.

The weight factor $\beta$ determines the balance between reconstruction loss and KL divergence. A lower $\beta$ prioritizes reconstruction accuracy, causing the VAE to behave similarly to a standard AE, focusing on replicating the input data as closely as possible. This can lead to overfitting of the latent parameters and poor generalization, which in turn will result in failure of the decoder to generate new feasible samples unseen during training.

Increasing $\beta$ enhances the impact of the KL divergence term, which encourages the latent variables to follow a standard Gaussian distribution. This regularization helps the model generalize better by smoothing the learned features and reducing overfitting. Consequently, the latent space becomes more structured, with less redundant information, leading to a disentangled representation where each latent variable captures distinct aspects of the data. This also helps reduce the dimensionality of the latent space as redundant parameters become inactive. These inactive parameters will simply produce gaussian noise which gets filtered out by the generator network. However, setting $\beta$ too high can overly constrain the model, prioritizing the alignment with the Gaussian distribution at the expense of reconstruction accuracy. This causes the generated outputs to converge towards an average, losing the variety and specific details of the input data. The challenge is to find an optimal $\beta$ value that balances reconstruction fidelity and latent space regularization.
The goal is to create a network that captures key data features in as few dimensions as possible, balancing detailed data reproduction with the ability to generate coherent and smooth shapes.

## 3. Network Development and Training

This section details the development and training of the VAE network using PyTorch [20]. It includes a discussion on the dataset preparation, the definition of reconstruction loss, and the selection of the $\beta$ parameter for the KL divergence term in the loss function.

### 3.1 Data Preparation

The VAE network is trained on a dataset comprising 1619 airfoil geometries sourced from the UIUC airfoil database [21], which offers extensive geometric data for various airfoil shapes. To standardize the data, all airfoil coordinate sets are fit with a B-spline, which is evaluated to provide surface coordinates at consistent cosine-spaced chordwise locations. Each airfoil is represented by 199 points, with 100 points each for the upper and lower surfaces, excluding the duplicate leading edge point. Each of the coordinate sets are adjusted by scaling, rotating and translating such that the chord line goes from $(0,0)$ to $(1,0)$.
Given the fixed chordwise locations, the network utilizes only the $y$ ordinates of each point as input, simplifying the data structure. The dataset is divided into three subsets: 75% for training, 15% for validation, and 10% for testing.
As the name suggests, the training set is used to train the model and iteratively find its optimal weights and biases. The validation set is used to monitor the network's performance during training, tracking the average loss per epoch to ensure the model is learning effectively and not overfitting to the training data. Finally, the test set is used to evaluate and compare the performance of fully trained networks when given previously unseen data. This ensures the model generalizes well beyond the training data, and is especially useful for hyperparameter tuning; evaluating different configurations to optimize the final network's performance.

### 3.2 Hyperparameter Selection

Determining optimal hyperparameters for a VAE involves heuristic methods that combine empirical evidence, theoretical understanding, and experimentation.
For training hyperparameters, a balanced approach was chosen. The batch size, set to 64, strikes a balance between convergence speed and generalization [17, 22]. The learning rate was set to $3 \times 10^{-4}$ based on best practices [23]. Training is designed to halt with early stopping if the training loss does not improve over 250 epochs, preventing overfitting.
A PCA on the dataset indicated that most data variance is captured by the first three components, with additional details by two more. To ensure robustness, the network was designed with eight latent parameters, providing flexibility to capture essential features while allowing the KL divergence term to render redundant parameters inactive.
The network is composed of linear layers with Leaky ReLU activation functions (Equation 7), preferred over standard ReLU for allowing negative values to retain significance. The final decoder layer uses a hyperbolic tangent activation to ensure that the outputs fall within the range $y_i \in [-1,1]$.

$$f(x) = \max(0.1x,;x) \tag{7}$$

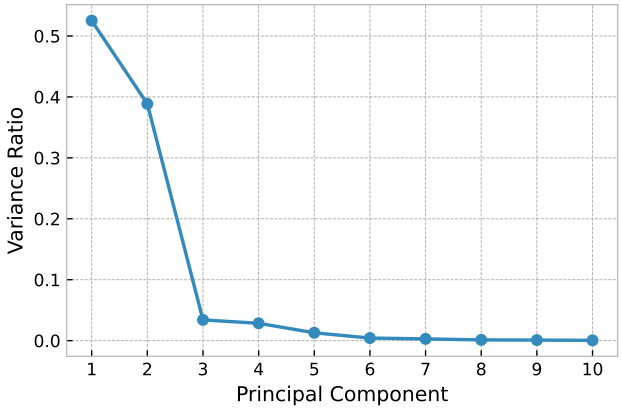



Figure 2 – Scree plot showing the variance ratio of the first ten principal components of the training data.

Hyperparameter optimization was conducted using Optuna [24], maintaining an hourglass shape with a maximum of five hidden layers. The optimal architecture identified by Optuna consisted of four hidden layers with neuron counts of 196, 172, 172, and 143, respectively, mirrored in both the encoder and decoder.
Finally, the value for the weight in the loss function was set to $\beta = 5 \times 10^{-7}$, which approximately equalizes the contributions of the reconstruction loss and the KL divergence, balancing data reconstruction and generalization.

## 4. Output Processing

Post-training, the encoder and decoder networks of the VAE serve distinct yet complementary roles. The encoder captures the essential features of each airfoil and encodes them into a compact latent space. Meanwhile, the decoder functions as a generator, capable of producing airfoil shapes from specified sets of latent parameters. As expected, the inferred latent distributions from the UIUC database, shown in Figure 3, take on shapes resembling normal distributions. Since these distributions are indefinite, in theory there is no limit on the latent values which can be sampled or provided as input to the generator.

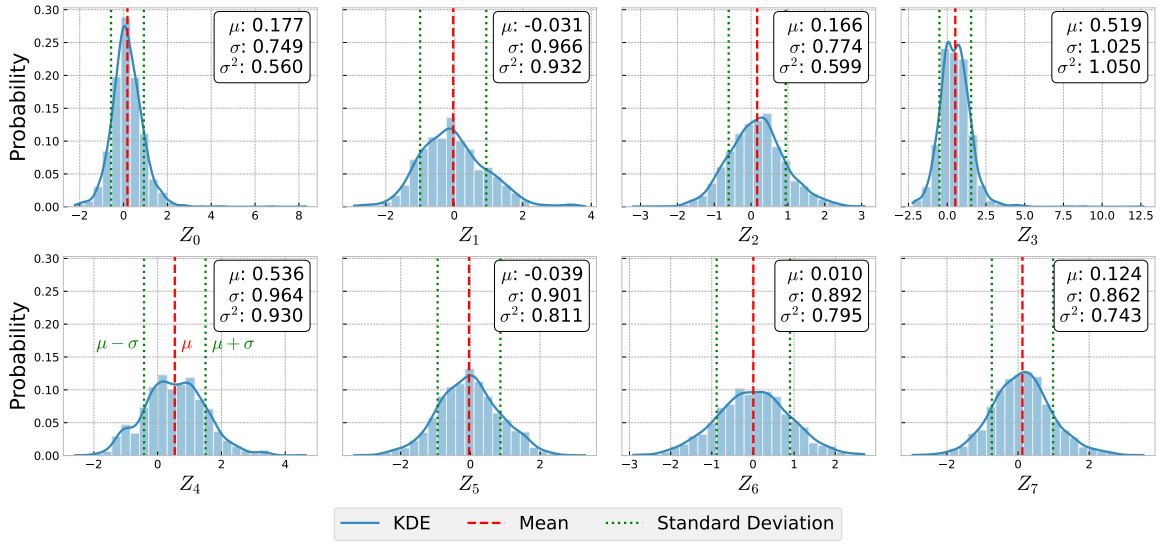



Figure 3 – Inferred latent distributions from the UIUC dataset

However, the generator produces the smoothest geometries when latent parameter values are close to the mean of the inferred distribution, as these values correspond to the most common features found in the dataset. When latent values are chosen in the tails of the distribution, the generator

struggles to accurately reproduce the respective shapes due to their rarity in the dataset. The generator was not trained to produce such shapes, and while there is limited extrapolation capability, this often results in generated airfoils with jagged surfaces. To address this issue, the generated coordinates are fit with a Bezier curve to obtain smooth airfoils across a wide range of latent parameter values.

Fitting a Bezier curve through the generated points follows a modified version of the approach by Agarwal and Sahu [25]. Initially, Bezier curves of degree $k$ are fit through the upper and lower surface points separately to determine the locations of the $n$ control points, where $k = n + 1$. These Bezier curves are then converted to B-splines by combining the Bezier control points with a knot vector of length $n + k + 1 = 2n$. For both the upper and lower surface B-spline, two distinct knot values, 0 and 1, are used. Each knot value has a multiplicity $n$, clamping the B-spline endpoints and mimicking the behavior of the Bezier curve.
These curves are then combined into a single B-spline, functioning as a composite Bezier curve, by concatenating the control points in order, following the Selig format (excluding the duplicate leading edge control point). The knot values 0 and 1 are maintained to clamp the curve at the trailing. An interior knot for the leading edge is added with multiplicity $k$ to ensure $C^2$ continuity in the leading edge.
Figure 4 illustrates the Bezier control points and the resulting surface curve for the MH70 airfoil. The figure demonstrates the surface produced by fitting the coordinates with a Bezier curve composed of 21 control points: 10 for the upper surface, 10 for the lower surface, and one leading edge point. Fixing the leading edge coordinate point at $(0,0)$ and keeping the first control points at $x = 0$ ensures gradient continuity at the leading edge. This process effectively removes surface roughness caused by minor errors in the generator output. By allowing the VAE to train directly on the geometry of the airfoils, it captures the essential features of the coordinates, thereby improving the quality of the generated airfoils without the need to retrain the network on a different data format.

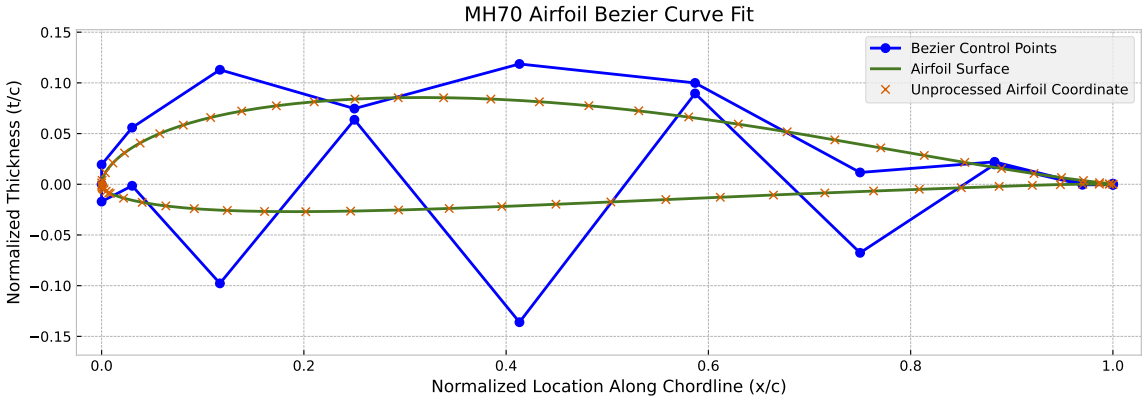
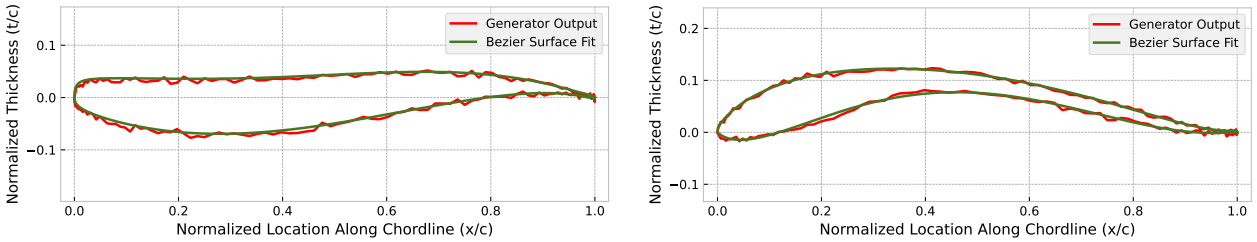



Figure 4 – Example of Bezier curve fitting to produce a smooth airfoil from the generator output.

To demonstrate the roughness of the generator output and the resulting smooth Bezier surface, two arbitrary airfoils are generated by choosing latent parameter vectors with values far from the mean. Figure 5 shows two airfoils generated from latent vectors that deviate from the mean, along with the resulting Bezier curves that smooth the surfaces. The respective latent input vectors are provided in the captions.

## 5. Validation

An important post-training step is to assess the VAE's performance in reproducing the airfoils it was trained on. This involves passing all dataset airfoils through the network and comparing the reconstructed airfoils with their originals. Figure 6 shows the reconstruction validation for the 12 airfoils with the highest reconstruction loss, with the respective loss indicated in each plot. Note that the reconstructed airfoils plotted are not the direct outputs of the generator, but the fitted Bezier splines discussed in section 4.




(a) $\bar{w} = [1.5,\ 1.9,\ -2.1,\ -1.0,\ -1.3,\ -1.7,\ -1.5,\ -1.3]$ (b) $\bar{w} = [-1.2,\ -1.5,\ -2.9,\ -2.1,\ -2.5,\ 1.7,\ 0.9,\ 2.1]$

Figure 5 – Demonstration of the generator output smoothing with Bezier curve fitting for two airfoils generated from arbitrary latent vectors with values far from their respective means. Latent vectors are defined by $\bar{z} = \bar{\mu} + \bar{w} \odot \bar{\sigma}$, and the arbitrary weight vector $\bar{w}$ is provided in the respective captions.

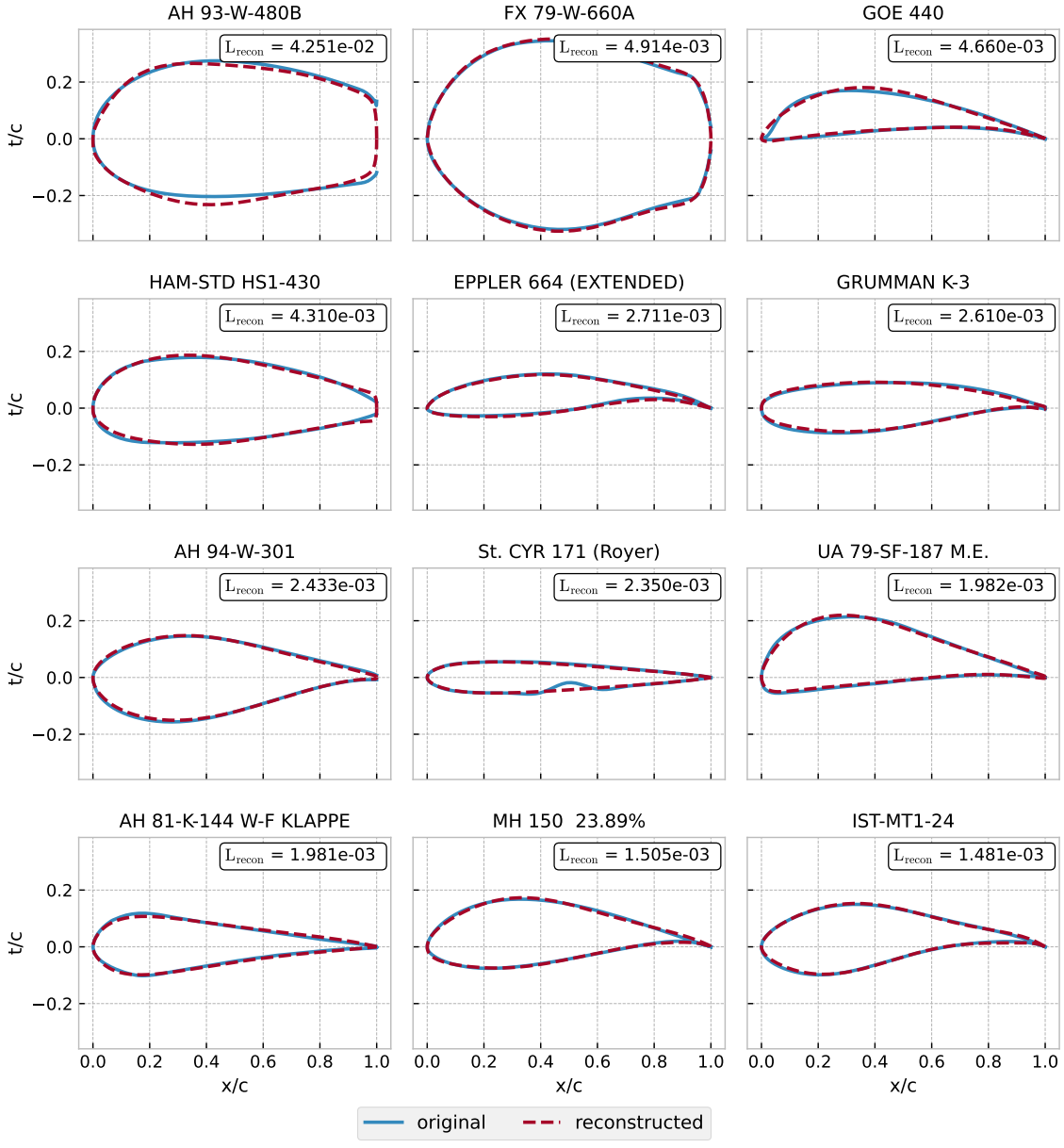



Figure 6 – Validation plots showing the reconstruction of the 12 airfoils with the highest reconstruction loss. Reconstruction loss values are marked in the corner of each plot.

The highest reconstruction losses are observed for uncommon airfoil shapes, geometries underrepresented in the dataset. Of the four airfoils with the highest loss, three are turbine or propeller blade cross-sections with significantly larger trailing edge gaps and higher thickness-to-chord ratios. The

fourth one, the GOE440, has an oddly shaped, sharp leading edge which the network does not know how to reproduce. Instead, the leading edge is rounded to a shape more akin to a typical airfoil. Similarly, a misplaced coordinate point in the coordinate file of the StCYR171 airfoil is corrected during the reconstruction process. While the generator does not perfectly reproduce these airfoils, it does a commendable job of capturing their features, and the summed relative error is less than $L_{recon} < 0.3\%$, which quickly drops to $L_{recon} < 0.15\%$ for the last airfoil plotted in Figure 6. Considering this is the worst performance out of more than 1600 airfoils, it can be concluded that the VAE has indeed captured the geometric features of the training data.

## 6. Captured Features in the Latent Dimensions

Understanding the relationships between the latent variables of the VAE and the various geometric and aerodynamic properties of airfoils is crucial for enhancing the interpretability of the model. By identifying how changes in latent variables influence specific airfoil features, we can provide designers with valuable insights into the geometric and aerodynamic implications of modifying the latent parameters when generating airfoils.

### 6.1 Qualitative Inspection

The effects of each latent parameter on the produced airfoil is visualized in Figure 7. Airfoils are generated by taking the mean latent vector, and individually adjusting the value of each latent parameter to a value two standard deviations from its mean, $z_i = \mu_i \pm 2 \cdot \sigma_i$, keeping the rest at their mean values $z_{j \neq i} = \mu_j$. This approach allows for a visual identification of the features captured by each latent dimension. For the first latent parameter, $z_0$, the primary features are the leading edge radius and the upper surface curvature. The second latent parameter, $z_1$ modifies the camber line, transitioning from a more aerodynamically aft-loaded airfoil to a more front-loaded configuration. The effects of $z_2$ are more subtle but mainly involve changes to the camber line angle (or gradient) at the leading edge and the trailing edge wedge angle. The remaining dimensions clearly capture the maximum thickness, camber, and chordwise position of maximum thickness in $z_3$, $z_4$, and $z_7$ respectively, while $z_5$ and $z_6$ do not have any effect at all on the generated shape.

### 6.2 Q-Q Plots and Inactive Latent Dimensions

Redundant latent parameters should become inactive as their distributions become Gaussian under the influence of the KL-divergence term, and their values are filtered out by the generator. This appears to be the case for $z_5$ and $z_6$ in Figure 7. From Figure 3 one might deduce that some latent parameters may be active from the shape of the distributions, for others it is less clear. Quantile-Quantile (Q-Q) plots are a visual method for comparing two distributions by plotting their quantiles against each other. They can be used to measure how closely the distributions of the latent parameters align with the normal distribution, helping to identify which latent parameters are active or inactive.

Figure 8 shows the Q-Q plot of the learned latent distributions compared to a Gaussian distribution, along with the corresponding squared error values. The errors in the Q-Q plot indicate that the distributions of the inactive latent dimensions indeed have the lowest error values, confirming their similarity to a Gaussian distribution. This demonstrates that Q-Q plots may be used for evaluating latent parameter activity. However, there is no definitive threshold that determines when a latent dimension becomes active. Additionally, the magnitude of the error value does not necessarily correlate with the impact of the parameter on the output. For instance, the Q-Q plot for $z_4$ exhibits an error value ten times lower than that for $z_0$, yet Figure 7 reveals that variations in $z_4$ result in significantly greater changes in the generated airfoil compared to variations in $z_0$. It is important to note that if a latent dimension captures a feature which happens to be normally distributed in the dataset, the Q-Q plot and corresponding error value will show a close resemblance between the feature distribution and the Gaussian. This could lead to the incorrect conclusion that this latent dimension is inactive. Therefore, this method should be used as an indication rather than definitive proof of latent activity.

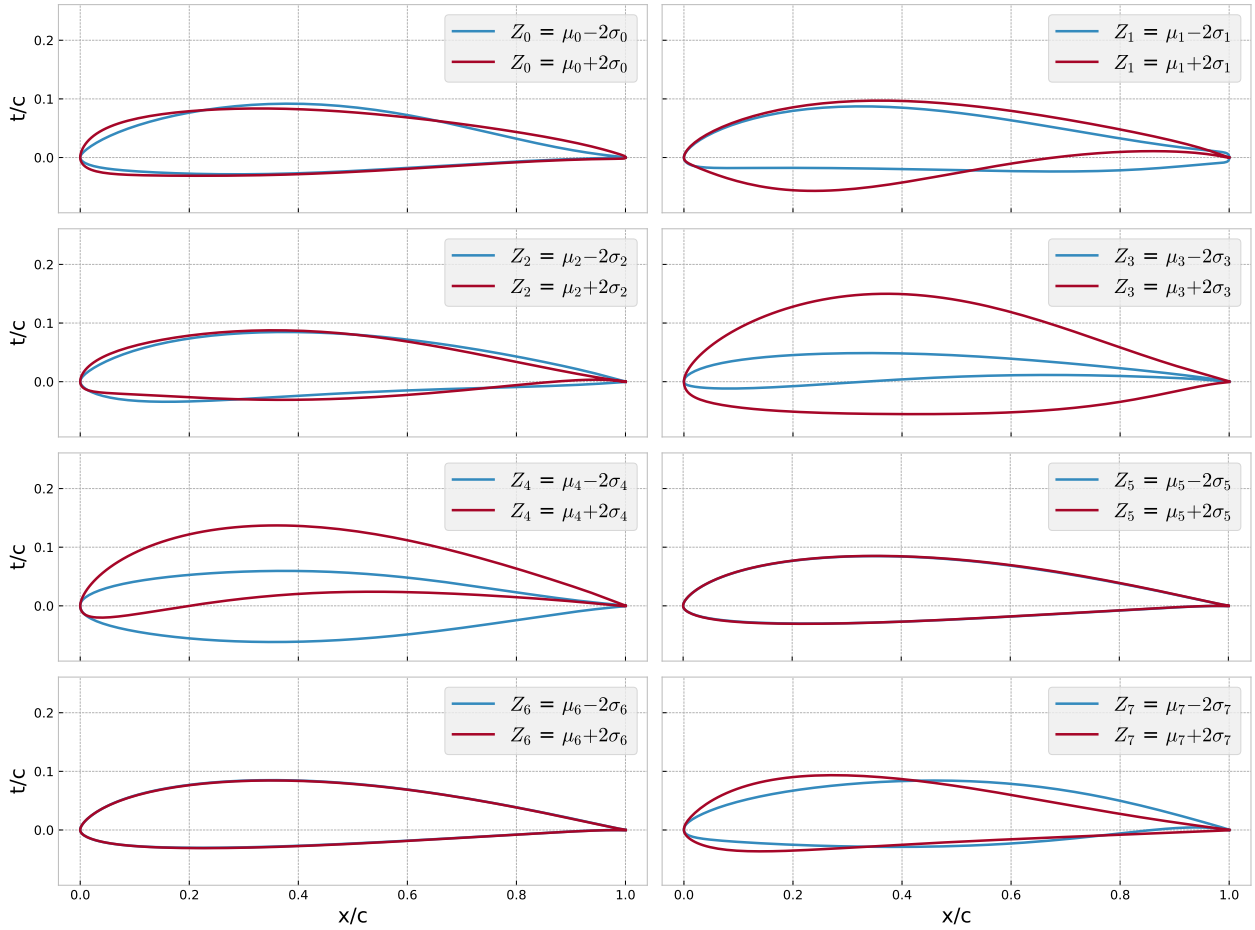



Figure 7 – Effect of the latent parameters on the generated shape. Each parameter is set to $z_i = \mu_i \pm 2\sigma_i$ while the rest are kept at their mean values $z_{j \neq i} = \mu_j$. [10]

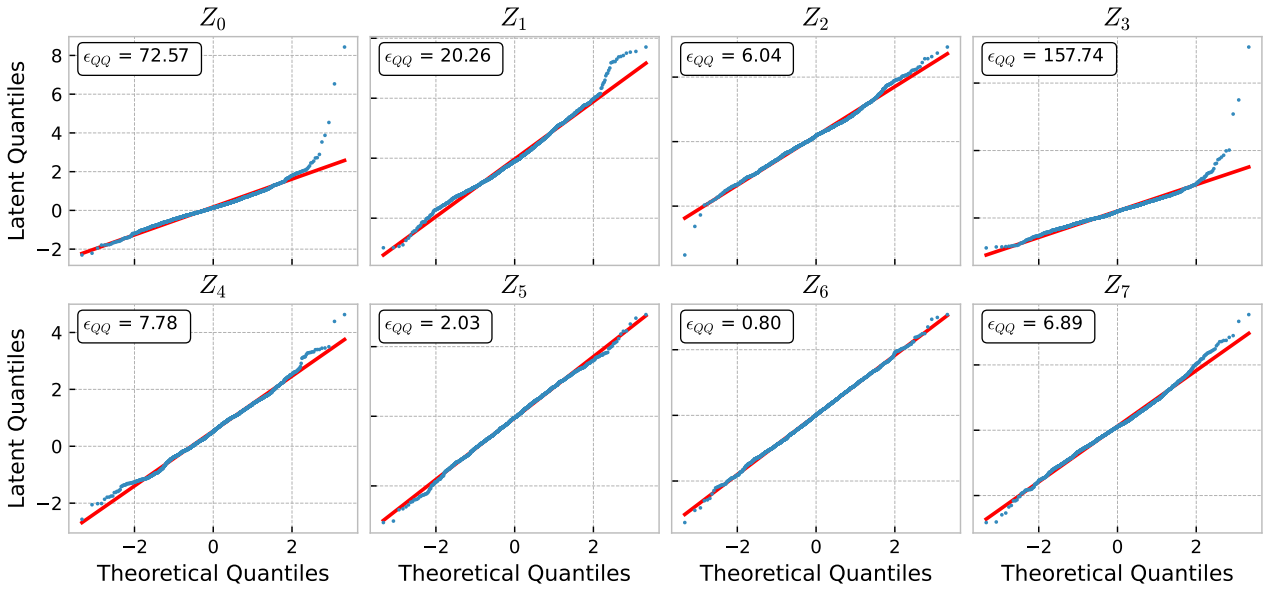



Figure 8 – Q-Q plot of the latent distributions compared to the Gaussian distribution pushed by the KL-divergence loss term during training.

## 6.3 Correlation Analysis

Correlation analyses provide a quantitative method to understand the relationships between latent parameters and specific airfoil features, moving beyond visual comparisons. Four different methods

were used: Pearson, Spearman, and Kendall-Tau correlation, and Mutual Information.

The Pearson correlation coefficient measures the linear relationship between parameters. In this case these are the latent variables of the VAE and the various geometric and aerodynamic properties of the airfoils. The Pearson correlation coefficient, $\rho_{X,Y}$, is defined in Equation 8, where $Cov(X,Y)$ is the covariance of variables $X$ and $Y$, and $\sigma_X$ and $\sigma_Y$ are the standard deviations of $X$ and $Y$, respectively. [26–28]

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_y} \tag{8}$$

The Spearman correlation coefficient is utilized to measure the monotonic relationship between the latent variables of the VAE and the various geometric and aerodynamic properties of the airfoils. Unlike the Pearson correlation, which assesses linear relationships, the Spearman correlation evaluates how well the relationship between two variables can be described using a montonic function (non-increasing, or non-decreasing, but not necessarily linear). The Spearman correlation coefficient, $\rho_s$, is defined by Equation 9, where $R(X)$ and $R(Y)$ represent the ranks of the values of $X$ and $Y$, respectively. $cov(R(X),R(Y))$ is the covariance of the ranks, and $\sigma_{R(X)}$ and $\sigma_{R(Y)}$ are the standard deviations of the ranks. [28, 29]

$$\rho_s = \frac{cov(R(X),R(Y))}{\sigma_{R(X)} \sigma_{R(Y)}} \tag{9}$$

The Kendall-Tau correlation coefficient is another non-parametric measure used to assess the strength and direction of association between two ranked variables. It evaluates the correspondence between the rankings of data pairs, providing a measure of the ordinal relationship between the variables. The Kendall Tau correlation coefficient, $\tau$, is obtained using Equation 10. where $C$ is the number of concordant pairs, $D$ is the number of discordant pairs, and $n$ is the number of observations. Concordant pairs are those where the order of the ranks for both variables is the same, while discordant pairs are those where the order of the ranks is different. [28, 30, 31]

$$\tau = \frac{(C-D)}{\frac{1}{2}n(n-1)} \tag{10}$$

The Pearson, Spearman, and Kendall correlation methods all return coefficient ranging from -1 to 1, with values close to 1 indicating a strong positive relationship, values close to -1 indicating a strong negative relationship, and values around 0 indicating no relationship.
Finally, the mutual information $I(X;Y)$ between two variables $X$ and $Y$, given by Equation 11. It is a measure of the mutual dependence between two variables, essentially Kullback-Leibler divergence between the joint distribution $p(x,y)$ and the product of the marginal distributions $p(x)p(y)$. Unlike Pearson, Spearman, and Kendall's Tau correlations, which measure linear or monotonic relationships, mutual information captures both linear and non-linear dependencies. It quantifies the amount of information obtained about one variable through another variable, making it a versatile tool for understanding complex relationships in data. [32, 33]

$$I(X;Y) = D_{KL}(p(x,y)|p(x)p(y)) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{11}$$

These correlation methods were applied to two types of generated data. The first comes from latent vectors sampled randomly from a uniform distribution $\bar{z} \sim \mathscr{U}(\bar{\mu}-2\bar{\sigma}, \bar{\mu}+2\bar{\sigma})$. The second follows a similar approach as depicted in Figure 7, where each latent parameter is varied systematically between $z_i = \mu_i \pm 2\sigma_i$, while the rest is kept at the mean value, $z_{j\neq i} = \mu_j$. Using randomly sampled data, Pearson, Spearman, and Kendall all yielded similar results for the correlations. When using the data obtained from traversing individual latent parameters, Pearson's correlation returns similar results, but does a slightly better job now at highlighting stronger correlations than when using random samples. This makes sense since traversing individual latent dimensions highlights their impact, while

it takes significantly more random sampled latent vectors to cover a similar value range for each latent parameter. Simply using latent traversal data yields near identical results while needing only a fraction of the samples. Spearman and Kendal, however, overestimate correlations between all parameters when using latent traversal, making these results less useful. Therefore, only the Pearson Correlation matrix from the latent traversal data is shown here. Additional plots are provided in Appendix A for reference.

The airfoil features used for the correlation analysis consist of geometric, and inviscid aerodynamic properties. For the geometry, the values considered are the maximum thickness ($t_{max}$), maximum camber ($c_{max}$), upper and lower crest value ($z_u$ and $z_l$), the curvatures at the upper and lower crest ($\kappa_{z_u}$ and $\kappa_{z_l}$), the chordwise locations of each of these properties ($x_{t_{max}}$, $x_{c_{max}}$, $x_{z_u}$, $x_{z_l}$), the leading edge radius ($R_{LE}$), the angles of the camber line at the leading and trailing edge ($\theta_{LE}$ and $\theta_{TE}$) and the trailing edge wedge angle ($\gamma_{TE}$). The inviscid aerodynamic properties, found using XFOIL [34], are the lift and drag coefficients at zero angle of attack ($C_{L_0}$ and $C_{M_0}$), the lift and moment curve gradients ($C_{L_\alpha}$ and $C_{M_\alpha}$), and the zero-lift angle of attack ($\alpha_{0L}$). Figure 9 shows the absolute values of the Pearson correlation coefficients between these properties. This highlights the inherent correlations between these parameters, and high correlations in this matrix will likely result in those features changing together when the latent dimensions are changed.

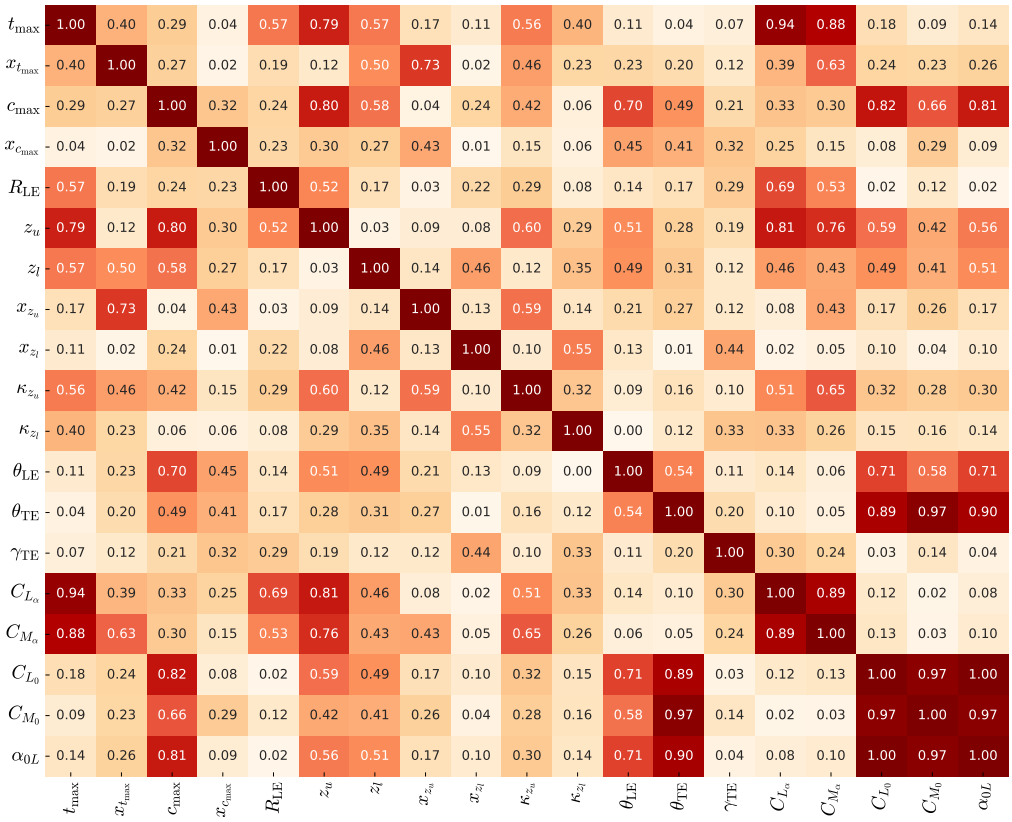



Figure 9 – Pearson correlation matrix of the airfoil properties only

The Pearson correlation matrix showing the links between the airfoil properties and the latent dimensions is provided in Figure 10. The results from the Mutual Information method are displayed in Figure 11, along with the absolute value Pearson Correlation matrix as a comparison. It can be seen that the results of the Pearson correlation and the mutual information are consistent, meaning all 4 methods yield similar results. It should be noted that this was done using latent traversal data. When using random samples, initially fewer correlations stand out. as the number of samples is increased, results converge to those presented in the figures below. These correlation matrices provide a quantitative way of evaluating which features are captured in the latent dimensions. The correlation coefficients confirm the findings from visually inspecting Figure 7.

- The correlation of $z_1$ with the trailing edge wedge angle, leading-edge radius and upper crest

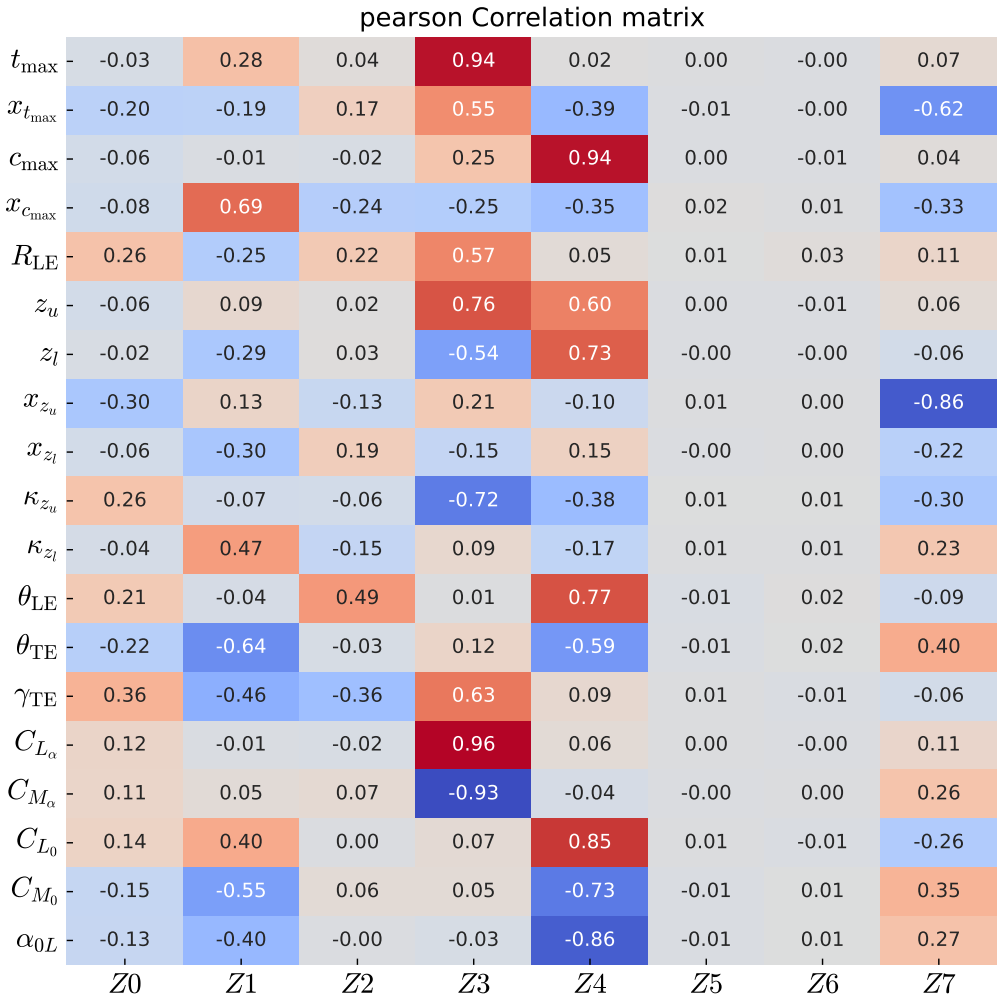
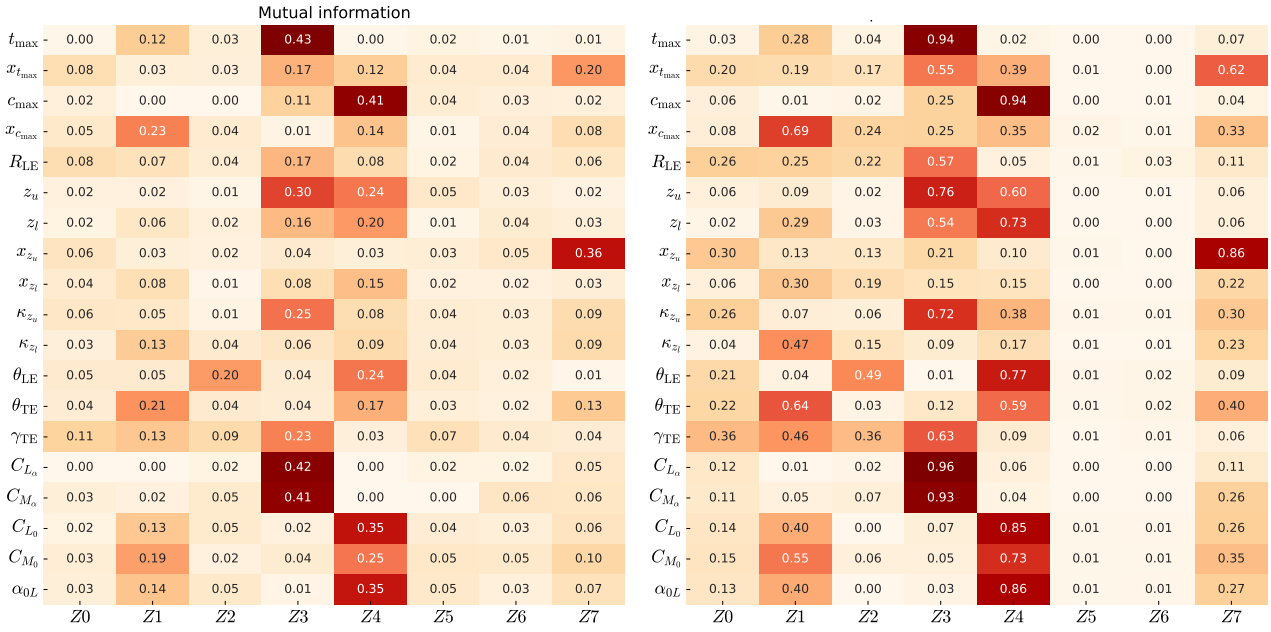



Figure 10 – Pearson correlation matrix constructed with data obtained by individually varying each latent dimension of the mean latent vector




(a) Mutual Information matrix (b) Absolute value Pearson correlation matrix

Figure 11 – Mutual information matrix compared to the absolute value Pearson correlation matrix

curvature indicates that this parameter changes the thickness distribution between a thicker middle section with sharper leading and trailing edges, to a more flattened airfoil with more constant thickness distribution. The increasing gradients of the camber line at the leading and trailing edge, indicated by changes in $\theta_{LE}$ and $\theta_{TE}$ indicates it is mostly the upper surface which changes shape, and an increase in leading edge radius induces a forward move in the location of maximum thickness. Aerodynamically, this parameter slightly increases the lift and moment curve slopes and lift and zero angle of attack, while reducing the moment at zero angle of attack and the angle of attack for zero lift.

- $z_1$ mostly affects the maximum thickness, camber location, and trailing edge angles. The relatively large correlation with the lower crest value, location, and curvature indicates that mostly the lower surface is affected. The impact in lift coefficient and moment indicate that the aerodynamic loading both changes in magnitude and location.

- Just like the first latent parameter, $z_2$ correlations have lower values indicating more subtle changes to the geometry. Mainly the leading edge angle, and trailing edge wedge angle are altered in this latent dimension. This parameter appears to have little to no impact on the lift coefficient, and only minor changes to the moment coefficient are observed. Testing this out by varying the third latent parameter in the mean latent vector between $z_2 = \mu_2 \pm 2\sigma_2$, the $C_{L_0}$ only changes between $0.366$ and $0.371$, and the zero-lift angle of attack varies between $\alpha_{0L} = -3.05°$ and $\alpha_{0L} = -3.1°$. Values for the moment coefficient are observed between $C_{M_0} = -0.084$ and $C_{M_0} = -0.07$. So indeed, the aerodynamic properties are not changed much by $z_2$.

- Features captured by $z_3$ and $z_4$ are the easiest to identify. $z_3$ clearly captures the thickness of the airfoil, which in turn is correlated with the leading edge radius, surface crest values, and the trailing edge wedge angle. The upper surface is changed more than the lower surface, as indicated by the different values for crest and crest curvature correlations. $z_4$ controls the maximum camber, which again is correlated with the angles of the leading and trailing edge, and the surface crest values. Surprisingly, though, the values of the curvature at the crests is not as subject to these changes as one might expect. Aerodynamically, the maximum thickness mainly changes the lift and moment curve gradients, while the camber shifts the curve while maintaining a constant slope.

- $z_5$ and $z_6$ have no link with any properties, and are inactive. This was already visible in Figure 7, and speculated in subsection 6.2 Fixing these latent parameters to their mean value has no effect on the validation discussed in section 5.

- Finally, the last latent dimension mainly shifts the location of the maximum thickness. The high correlation with the upper surface crest again shows that changing thickness mainly affects the upper surface, as previously observed with the property correlations shown in Figure 9.

## 7. Conclusion and Future Work

This study has presented an approach to improve the interpretability of Variational Autoencoders (VAEs) in airfoil design. By analyzing the effects of each latent parameter, we show that VAEs can capture and represent complex geometric and aerodynamic features of airfoils.

Key findings include the reveal of latent dimensions that influence well-known airfoil characteristics, such as thickness and camber, but also some which vary the airfoil geometry in a less common way, for example by changing the chordwise thickness distribution or surface curvatures. Correlation analyses, both visual and quantitative, reveal these relationships between latent parameters and airfoil features, providing valuable insights for designers. The application of Bezier curve fitting to the generated airfoil shapes addresses the issue of surface roughness, ensuring the generation of smooth airfoils even when choosing values far away from the mean of the latent distributions inferred by the network.

The validation process confirms the VAE's capability to reconstruct airfoils from the training dataset, with relatively low reconstruction errors even for the most challenging shapes. Additionally, identifying

inactive latent dimensions through Q-Q plots and correlation matrices enhances the understanding of the VAE's latent space structure.

This study underscores the importance of interpretability in machine learning models for engineering applications. By making the latent space more comprehensible, integrating VAEs into the airfoil design process becomes more intuitive and practical. These methods and insights can prove useful for further exploring network architectures and tuning other hyperparameters to improve the capability of learning a disentangled latent representation while maintaining accurate data reconstruction. This would further improve the model's applicability in various design scenarios.

The methodologies and findings in this study aim to contribute to the goal of explainable AI, fostering greater trust and adoption of AI-driven solutions in engineering design optimization. Through detailed exploration of latent dimensions and their impacts, this work sets the stage for more interpretable machine learning applications in the field of aerodynamic design.

## References


[1] L. Regenwetter, A. H. Nobari, and F. Ahmed. *Deep Generative Models in Engineering Design: A Review*. Mar. 2022.

[2] J. Li, X. Du, and J. R. Martins. "Machine Learning in Aerodynamic Shape Optimization". In: *Progress in Aerospace Sciences* 134 (Oct. 2022), p. 100849. ISSN: 03760421. DOI: 10.1016/j.paerosci.2022.100849.

[3] S. Le Clainche et al. "Improving Aircraft Performance Using Machine Learning: A Review". In: *Aerospace Science and Technology* 138 (July 2023), p. 108354. ISSN: 12709638. DOI: 10.1016/j.ast.2023.108354.

[4] J. Wang et al. "Flow Field Prediction of Supercritical Airfoils via Variational Autoencoder Based Deep Learning Framework". In: *Physics of Fluids* 33.8 (Aug. 2021), p. 086108. ISSN: 1070-6631, 1089-7666. DOI: 10.1063/5.0053979.

[5] E. Saetta, R. Tognaccini, and G. Iaccarino. "AbbottAE: An Autoencoder for Airfoil Aerodynamics". In: *AIAA AVIATION 2023 Forum*. San Diego, CA and Online: American Institute of Aeronautics and Astronautics, June 2023. ISBN: 978-1-62410-704-7. DOI: 10.2514/6.2023-4364.

[6] X. Du, P. He, and J. R. R. A. Martins. "A B-Spline-based Generative Adversarial Network Model for Fast Interactive Airfoil Aerodynamic Optimization". In: *AIAA Scitech 2020 Forum*. Orlando, FL: American Institute of Aeronautics and Astronautics, Jan. 2020. ISBN: 978-1-62410-595-1. DOI: 10.2514/6.2020-2128.

[7] X. Du, P. He, and J. R. Martins. "Rapid Airfoil Design Optimization via Neural Networks-Based Parameterization and Surrogate Modeling". In: *Aerospace Science and Technology* 113 (June 2021), p. 106701. ISSN: 12709638. DOI: 10.1016/j.ast.2021.106701.

[8] K. Yonekura and K. Suzuki. "Data-Driven Design Exploration Method Using Conditional Variational Autoencoder for Airfoil Design". In: *Structural and Multidisciplinary Optimization* 64.2 (Aug. 2021), pp. 613–624. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-021-02851-0.

[9] Y.-E. Kang, D. Lee, and K. Yee. "Physically interpretable airfoil parameterization using variational autoencoder-based generative modeling". In: *AIAA SCITECH 2024 Forum*. Orlando, Florida. DOI: 10.2514/6.2024-0685.

[10] K. Swannet, C. Varriale, and A. K. Doan. "Towards Universal Parameterization: Using Variational Autoencoders to Parameterize Airfoils". In: *AIAA SciTech 2024 Forum*. Orlando, Florida. DOI: 10.2514/6.2024-0686.

[11] A. Creswell et al. "Generative Adversarial Networks: An Overview". In: *IEEE Signal Processing Magazine* 35.1 (Jan. 2018), pp. 53–65. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2765202.

[12] J. Gui et al. “A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022), pp. 1–1. ISSN: 1041-4347, 1558-2191, 2326-3865. DOI: 10.1109/TKDE.2021.3130191.

[13] A. B. Arrieta et al. *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI*. Dec. 2019.

[14] D. P. Kingma and M. Welling. “Auto-Encoding Variational Bayes”. In: (2013). DOI: 10.48550/ARXIV.1312.6114.

[15] I. Higgins et al. “Early Visual Concept Learning with Unsupervised Deep Learning”. In: (2016). DOI: 10.48550/ARXIV.1606.05579.

[16] D. Foster. *Generative Deep Learning*. Sebastopol, California: O’Reilly Media, Inc., 2019. ISBN: 978-1-492-04194-8.

[17] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning. Cambridge, Massachusetts: The MIT Press, 2016. ISBN: 978-0-262-03561-3.

[18] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. “Machine Learning for Fluid Mechanics”. In: *Annual Review of Fluid Mechanics* 52.1 (Jan. 2020), pp. 477–508. ISSN: 0066-4189, 1545-4479. DOI: 10.1146/annurev-fluid-010719-060214.

[19] G. E. Hinton and R. R. Salakhutdinov. “Reducing the Dimensionality of Data with Neural Networks”. In: *Science* 313.5786 (July 2006), pp. 504–507. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1127647.

[20] A. Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019.

[21] M. Selig. *UIUC Airfoil Data Site*. https://m-selig.ae.illinois.edu/ads/coord_database.html. Urbana, Illinois, 1996.

[22] D. Wilson and T. R. Martinez. “The general inefficiency of batch training for gradient descent learning”. In: *Neural Networks* 16.10 (2003), pp. 1429–1451. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(03)00138-2.

[23] A. Karpathy. *A recipe for training neural networks*. 2019.

[24] T. Akiba et al. “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: (2019). DOI: 10.48550/ARXIV.1907.10902.

[25] D. Agarwal and P. Sahu. “A Unified Approach for Airfoil Parameterization Using Bezier Curves”. In: *Computer-Aided Design and Applications* 19.6 (Mar. 2022), pp. 1130–1142. ISSN: 16864360. DOI: 10.14733/cadaps.2022.1130-1142.

[26] M. Dekking, ed. *A Modern Introduction to Probability and Statistics: Understandig Why and How*. Springer Texts in Statistics. London [Heidelberg]: Springer, 2005. ISBN: 978-1-85233-896-1.

[27] “Pearson’s Correlation Coefficient”. In: *Encyclopedia of Public Health*. Ed. by W. Kirch. Dordrecht: Springer Netherlands, 2008, pp. 1090–1091. ISBN: 978-1-4020-5613-0 978-1-4020-5614-7. DOI: 10.1007/978-1-4020-5614-7_2569.

[28] M. Hollander, D. A. Wolfe, and E. Chicken. *Nonparametric Statistical Methods*. 1st ed. Wiley Series in Probability and Statistics. Wiley, July 2015. ISBN: 978-0-470-38737-5 978-1-119-19603-7. DOI: 10.1002/9781119196037.

[29] “Spearman Rank Correlation Coefficient”. In: *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, 2008, pp. 502–505. ISBN: 978-0-387-31742-7 978-0-387-32833-1. DOI: 10.1007/978-0-387-32833-1_379.

[30] M. G. KENDALL. “A NEW MEASURE OF RANK CORRELATION”. In: *Biometrika* 30.1-2 (June 1938), pp. 81–93. ISSN: 0006-3444. DOI: 10.1093/biomet/30.1-2.81.

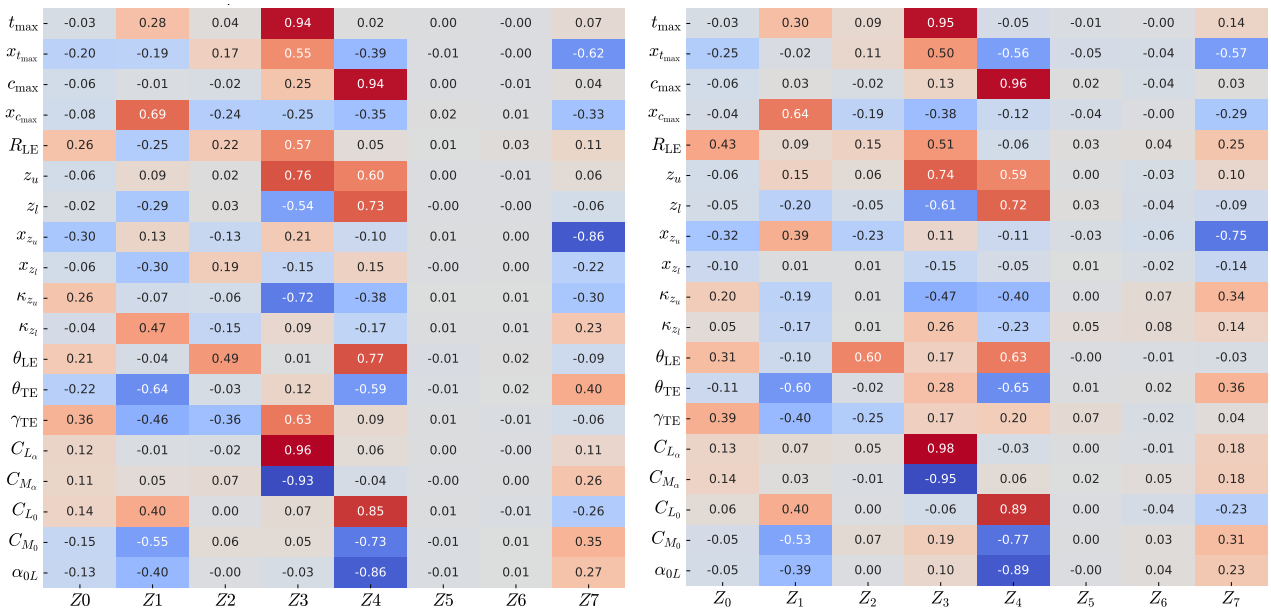
[31] L. Puka. “Kendall’s Tau”. In: *International Encyclopedia of Statistical Science*. Ed. by M. Lovric. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 713–715. ISBN: 978-3-642-04897-5 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_324.

[32] "Entropy, Relative Entropy, and Mutual Information". In: *Elements of Information Theory*. John Wiley Sons, Ltd, 2005. Chap. 2, pp. 13–55. ISBN: 9780471748823. DOI: https://doi.org/10.1002/047174882X.ch2.

[33] M. Beraha et al. *Feature Selection via Mutual Information: New Theoretical Insights*. 2019. DOI: 10.48550/ARXIV.1907.07384.

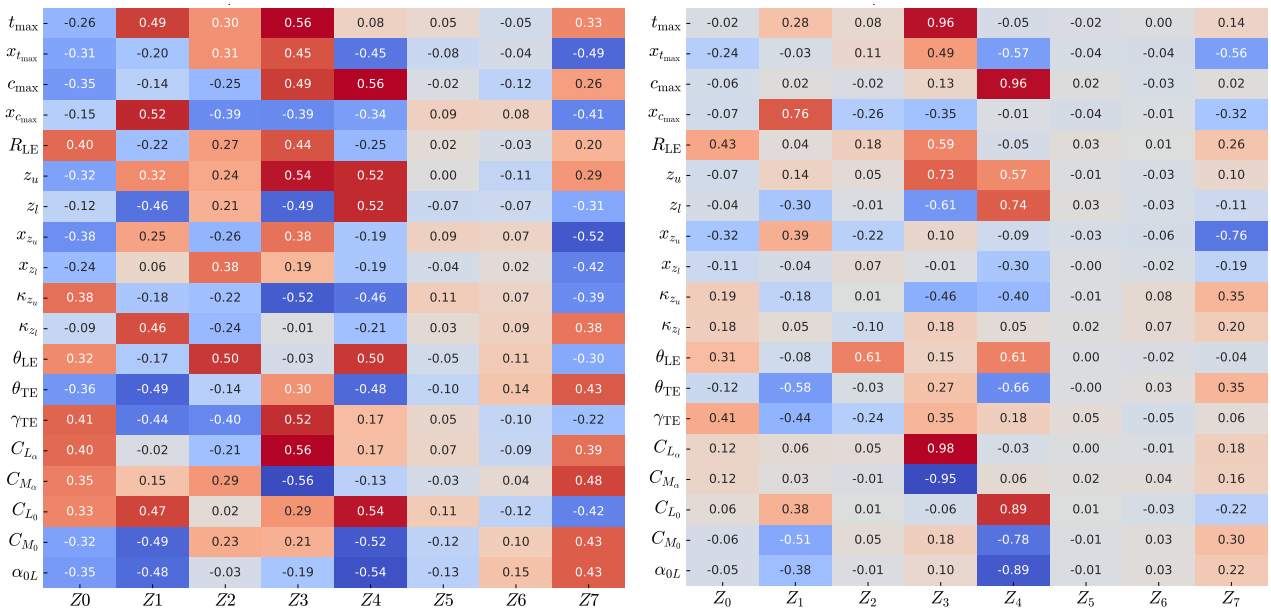
[34] M. Drela. "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils". In: *Low Reynolds Number Aerodynamics*. Ed. by T. J. Mueller. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 1–12. ISBN: 978-3-642-84010-4. DOI: 10.1007/978-3-642-84010-4_1.

# A Additional Correlation Results



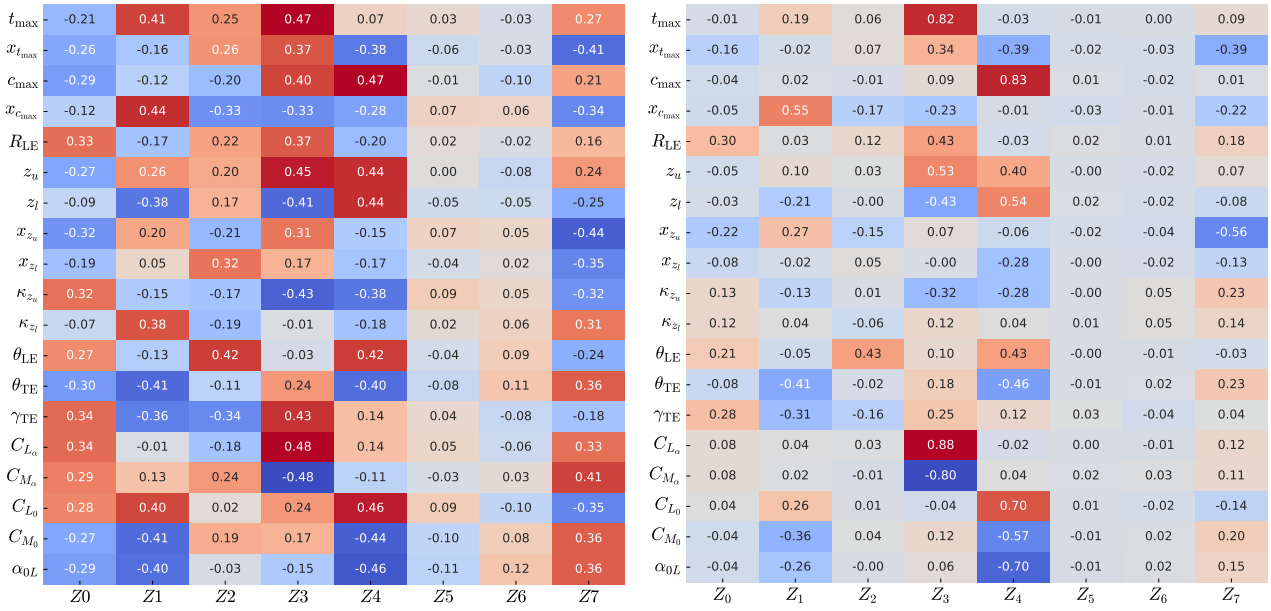

(a) Latent parameter traversal data (b) 500 random samples $\bar{z} \sim \mathscr{U}(\bar{\mu} - 2\bar{\sigma}, \bar{\mu} + 2\bar{\sigma})$

Figure 12 – Pearson correlation matrix for latent parameter traversal and random samples



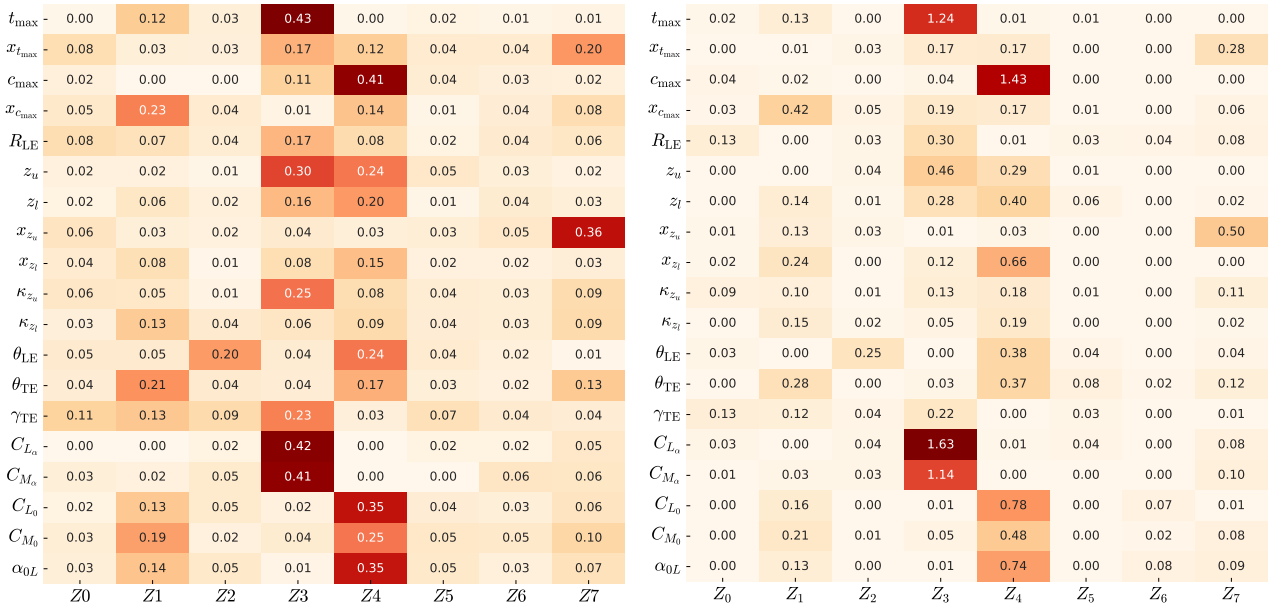

(a) Latent parameter traversal data (b) 500 random samples $\bar{z} \sim \mathscr{U}(\bar{\mu} - 2\bar{\sigma}, \bar{\mu} + 2\bar{\sigma})$

Figure 13 – Spearman correlation matrix for latent parameter traversal and random samples




(a) Latent parameter traversal data (b) 500 random samples $\bar{z} \sim \mathscr{U}\left(\bar{\mu} - 2\bar{\sigma}, \bar{\mu} + 2\bar{\sigma}\right)$

Figure 14 – Kendall-Tau correlation matrix for latent parameter traversal and random samples




(a) Latent parameter traversal data (b) 500 random samples $\bar{z} \sim \mathscr{U}\left(\bar{\mu} - 2\bar{\sigma}, \bar{\mu} + 2\bar{\sigma}\right)$

Figure 15 – Mutual Information matrix for latent parameter traversal and random samples

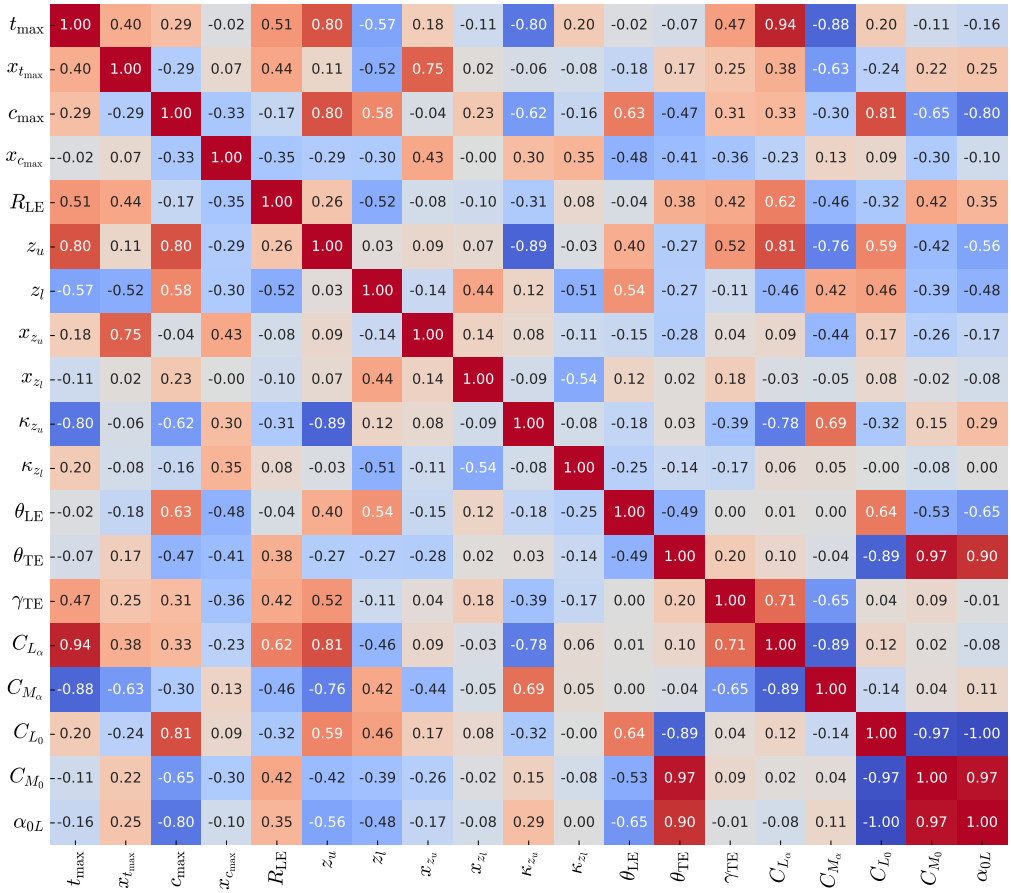



Figure 16 – Pearson property correlations for latent traversal data

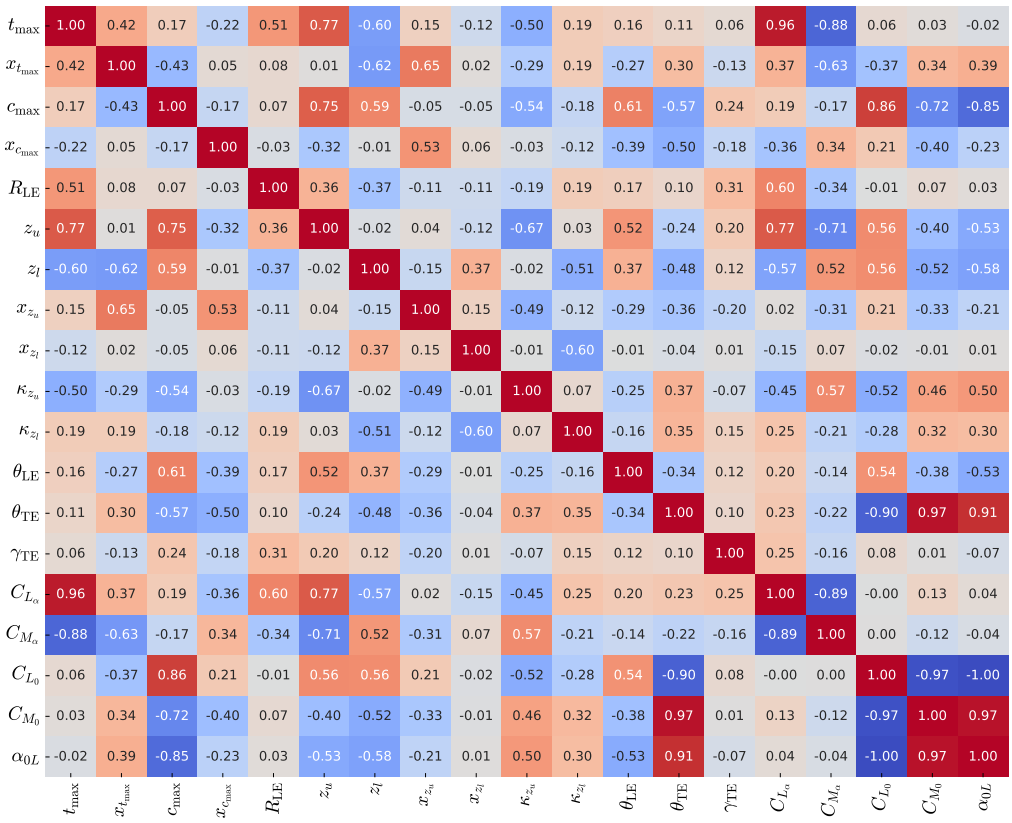



Figure 17 – Pearson property correlations for 500 random samples $\bar{z} \sim \mathscr{U}(\bar{\mu} - 2\bar{\sigma}, \bar{\mu} + 2\bar{\sigma})$

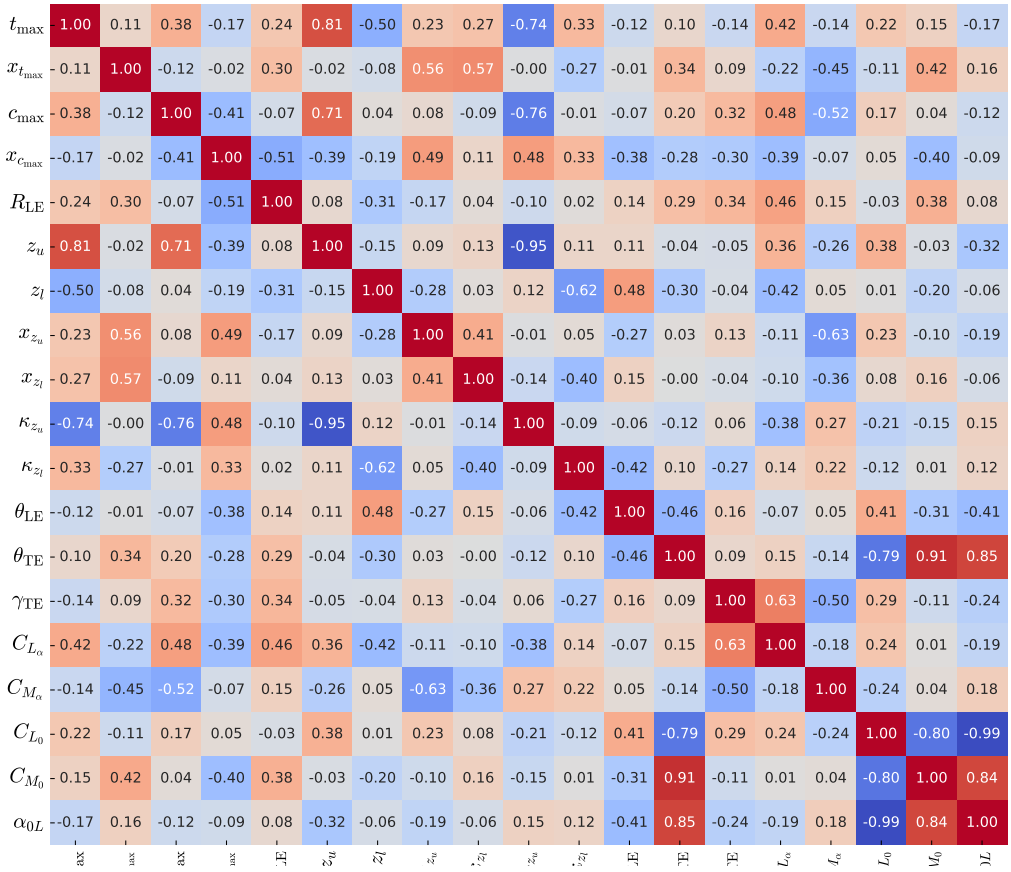



Figure 18 – Spearman property correlations

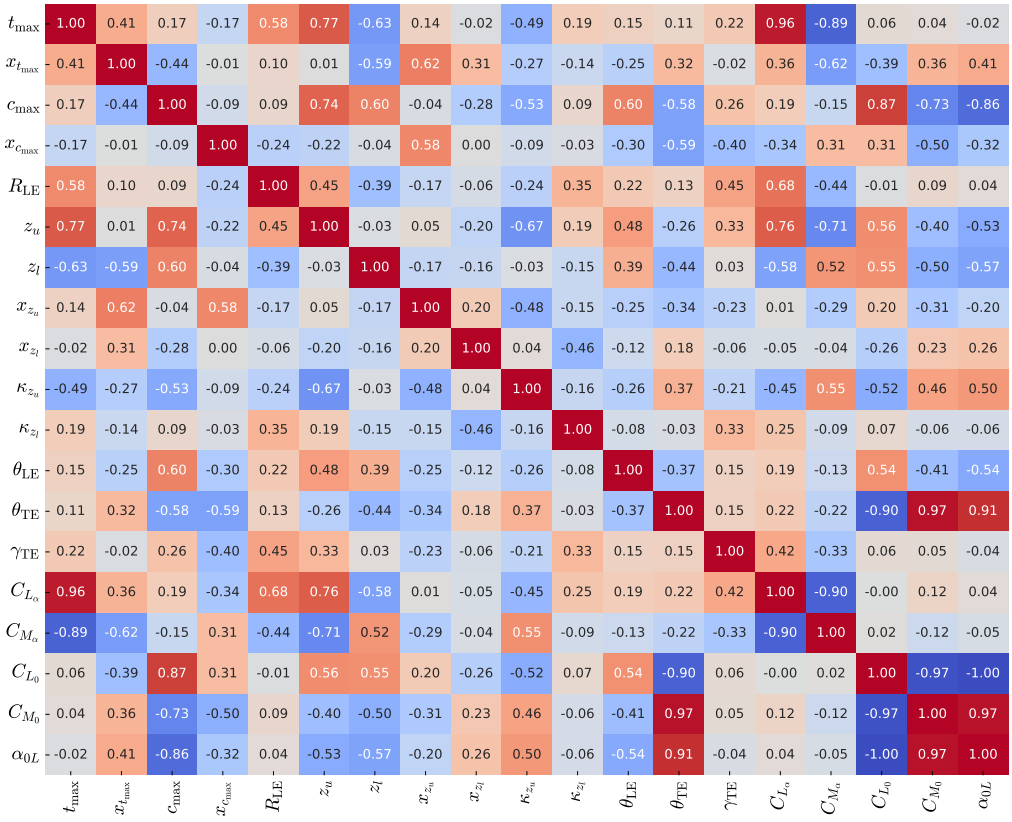



Figure 19 – Spearman property correlations for 500 random samples $\bar{z} \sim \mathscr{U}\left(\bar{\mu} - 2\bar{\sigma}, \bar{\mu} + 2\bar{\sigma}\right)$

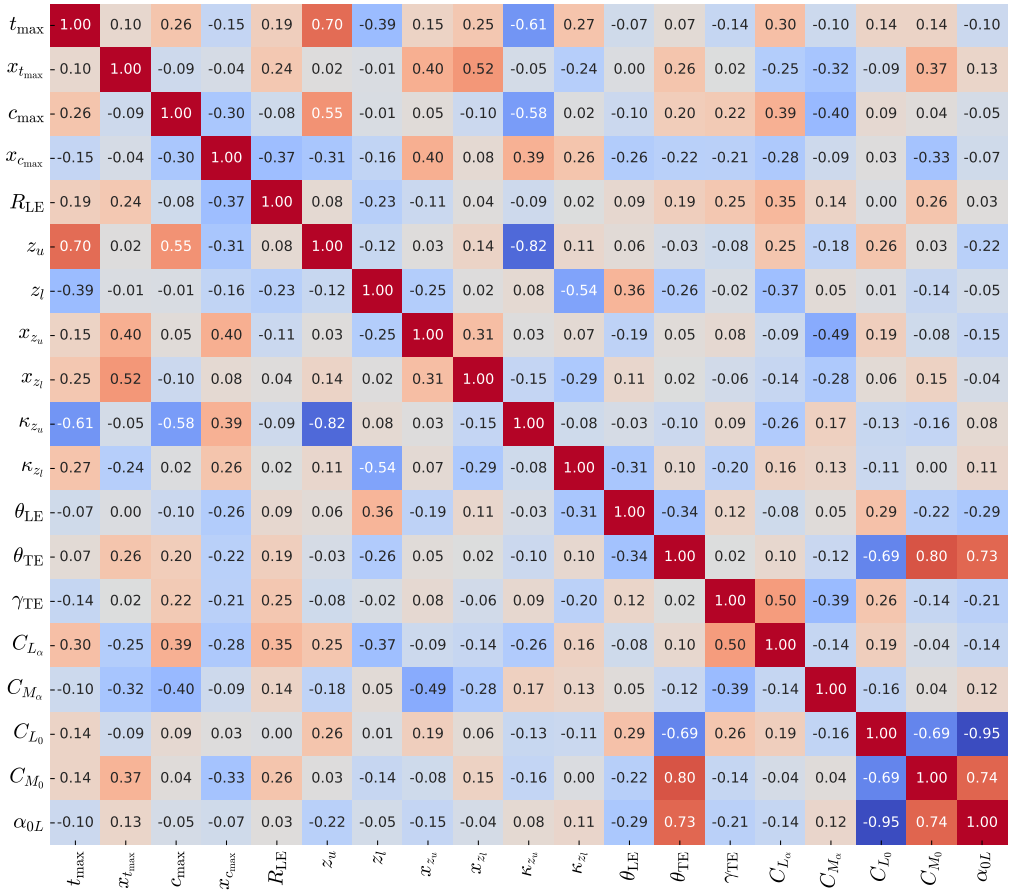



Figure 20 – Kendall-Tau property correlations

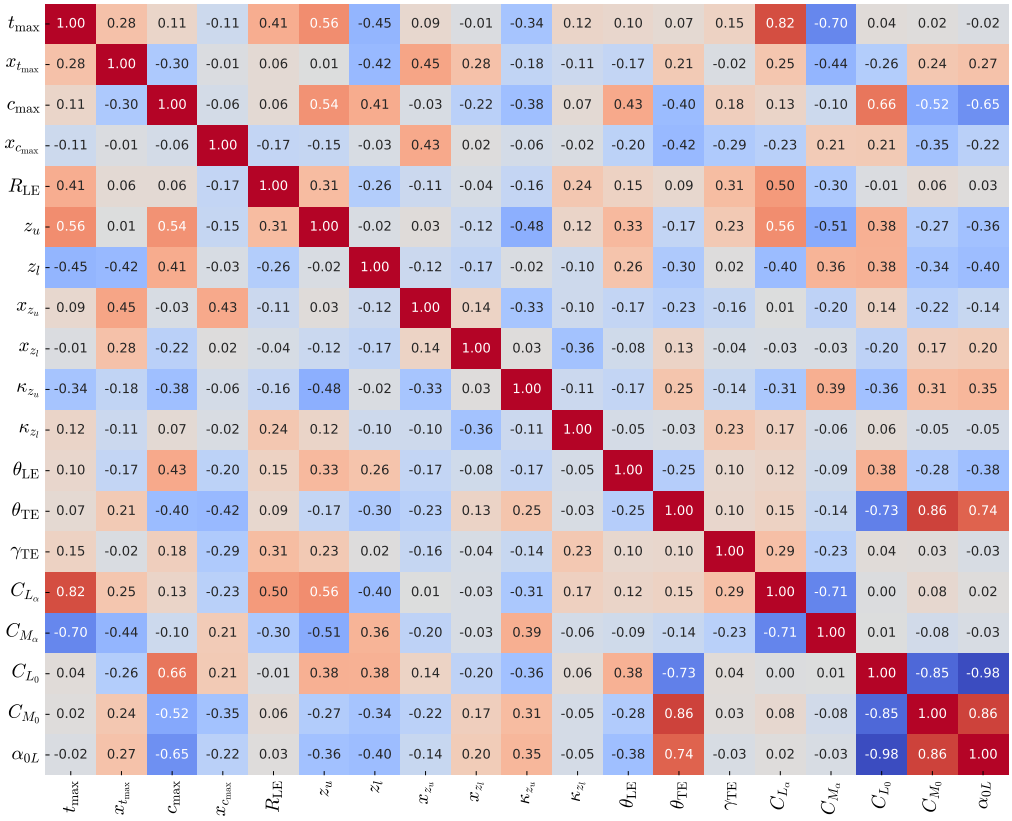



Figure 21 – Kendal-Tau property correlations for 500 random samples $\bar{z} \sim \mathscr{U}(\bar{\mu} - 2\bar{\sigma}, \bar{\mu} + 2\bar{\sigma})$