Methods to Generate Consistent Distributed Aerodynamic Loads Under Uncertainty

By

AARON BURKHEAD
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Mechanical and Aerospace Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

____

Mohamed Hafez, Chair

____

Paul Erickson

____

Derek Dalle

Committee in Charge

2024

Abstract

Methods to Generate Consistent Distributed Aerodynamic Loads Under Uncertainty

by

Aaron Burkhead

Doctor of Philosophy in Mechanical and Aerospace Engineering

University of California, Davis

Professor Mohamed Hafez, Chair

The proper construction of aerodynamic databases to support an ascent vehicle during the development phases, and flight, is a difficult task. It encompasses many different flight regimes and several types of aerodynamic coefficient data. The focus of this work addresses integrated force and moment, line load, and surface pressure aerodynamic databases covering the ascent phase of the vehicles flight. In particular, the work deals with a missing component in the uncertainty quantification of these databases, in which existing processes are able to estimate integrated aerodynamic coefficient uncertainty, however estimating an equivalent uncertainty for corresponding distributed loads is not done. The phrase "distributed loads" is used to refer to both line load and surface pressure data, as opposed to the scalar-valued integrated aerodynamic coefficients. The primary motivation of this work was to create a method that could generate uncertain distributed loads that are consistent with the quantified uncertainty in the integrated aerodynamic coefficients, while also being consistent with the statistical distributions of known data. A brief motivation and outline is included, before Chapter 2 introduces required background concepts. The methods developed for this work are described in Chapter 3 along with a brief survey of similar published works. The methods are validated and tested in a simulated uncertainty scenario, using both line load and surface pressure data, which are presented in Chapter 4. These results showed that the method using a multivariate normal copula was the most successful of the presented methods. Discussions of the performance of the methods in the various results is also included in Chapter 4, with summarizing conclusions and future work included in Chapter 5.

Dedicated to my friends and family of the past, present, and future.

Contents

List of Figures

List of Tables

List of Acronyms

BSP	Bayesian Sequential Partitioning
cdf	Cumulative Distribution Function
CFD	Computational Fluid Dynamics
GKDE	Gaussian Kernel Density Estimation
gPC	Generalized Polynomial Chaos
IUQ	Inverse Uncertainty Quantification
LOO	Leave-One-Out
MAVERIC	Marshall's Aerospace Vehicle Representation in C
MCMC	Markov Chain Monte Carlo
MMD	Maximum Mean Discrepancy
RANS	Reynolds-Averaged Navier-Stokes
SLS	Space Launch System
SVD	Singular Value Decomposition

List of Symbols

C_A	Integrated axial force coefficient
C_Y	Integrated side force coefficient
C_N	Integrated normal force coefficient
C_ℓ, C_{LL}	Integrated rolling moment coefficient
C_n, C_{LN}	Integrated yawing moment coefficient
C_m, C_{LM}	Integrated pitching moment coefficient
$c_A(x)$, $\frac{dC_A}{dx}$	Axial force coefficient line load
$c_Y(x)$, $\frac{dC_Y}{dx}$	Side force coefficient line load
$c_N(x)$, $\frac{dC_N}{dx}$	Normal force coefficient line load
$c_{LL}(x)$, $\frac{dC_\ell}{dx}$	Rolling moment coefficient line load
$c_{LN}(x)$, $\frac{dC_n}{dx}$	Yawing moment coefficient line load
$c_{LM}(x)$, $\frac{dC_m}{dx}$	Pitching moment coefficient line load
x	Vehicle axial distance coordinate
L_{ref}	Vehicle reference length
$\hat{x}$	Normalized vehicle axial distance coordinate
$\hat{x}_i$	$i-$th component of $\hat{x}$
MRP	Vehicle moment reference point subscript

$c_A(\hat{x})$, $\frac{dC_A}{d\hat{x}}$	Normalized Axial force coefficient line load
$c_Y(\hat{x})$, $\frac{dC_Y}{d\hat{x}}$	Normalized Side force coefficient line load
$c_N(\hat{x})$, $\frac{dC_N}{d\hat{x}}$	Normalized Normal force coefficient line load
$c_{LL}(\hat{x})$, $\frac{dC_A}{d\hat{x}}$	Normalized Axial force coefficient line load
$c_{LN}(\hat{x})$, $\frac{dC_n}{d\hat{x}}$	Normalized Side force coefficient line load
$c_{LM}(\hat{x})$, $\frac{dC_m}{d\hat{x}}$	Normalized Normal force coefficient line load
c_p	Surface pressure
M	Mach number
α	Vehicle angle of attack
β	Vehicle angle of sideslip
α_p	Vehicle total angle of attack
ϕ_p	Vehicle roll angle
$\mathbf{c}_{N,i}$	$i-$th case's normal force coefficient line load
$\mathbf{C}_N$	Snapshot matrix of normal force coefficient line loads
$\mathbf{C}'_N$	Snapshot matrix of delta normal force coefficient line loads
n	Number of axial slices in line load
n_c	Number of line loads in matrix $\mathbf{C}_N$
$\mathbf{U}_N$	SVD left singular value vectors
$\mathbf{\Sigma}_N$	SVD singular values
$\mathbf{V}_N^T$	SVD right singular value vectors
κ	Number of SVD bases used in dispersion methods
ω_i	Dispersion method's linear equation coefficients
$\tilde{\mathbf{U}}_N$	Matrix of κ SVD left singular value vectors

$\mathbf{\Omega}$	$(n \times n_c)$ matrix of coefficients ω_i
$\boldsymbol{\Upsilon}$	Covariance matrix of $\mathbf{\Omega}$ in Resampling method
$p_{\min}$	Minimum probability density value for Resampling method
F_i	Cumulative distribution function for y_i
P	Probability
C	Copula of joint cumulative distribution function
c	Copula of joint probability density function
y_i	$i-$th component of generic random variable y
p_i	Probability density function of y_i
F_i^{-1}	Inverse cumulative density function
σ	Covariance matrix of $\mathbf{\Omega}$ in Copula-based method
ω^u	Output of marginal cumulative distribution function applied to ω
$\tilde{p}_i$	Estimated probability density function for y_i
$\mathcal{U}$	Uniform probability distribution
$\mathcal{N}$	Normal probability distribution

Acknowledgments

I appreciate the unwavering support of my family and friends while I took on the challenges that came with pursuing this work. In particular, I thank my parents for the sacrifices they've made in order to gift me not only this opportunity but countless others. I'd also like to acknowledge the guidance of Professor Mohamed Hafez. His encouraging words and unreserved advice played a large part in completing this work. Finally, I'm grateful for the feedback and help from Derek Dalle and the rest of the NASA Ames SLS Team while I undertook this research.

Chapter 1

Motivation and Outline

The motivation for this dissertation stems from the challenges encountered during the engineering work of the Space Launch System (SLS), in particular the challenges of constructing aerodynamic databases that cover the ascent phase of the vehicle. The types of databases discussed in this work are integrated force and moment, line load, and surface pressure databases. The details of line loads are described in Section 2.1. Line loads and surface pressures are both representations of the distributed loading on the vehicle and are thus sometimes collectively referred to as distributed loads in this work. The aerodynamic databases are constructed and given to users in other domains that require aerodynamic data to perform their analysis on the vehicle. For example, the databases can be used with Marshall's Aerospace Vehicle Representation in C (MAVERIC), that allows for a six degree of freedom simulation of vehicle ascent with a Monte Carlo capability to allow for uncertain properties such as atmosphere, winds, propulsion, navigation, aerodynamics, and mass properties.

In general these aerodynamic databases can cover many different vehicle flight regimes, but this work focuses on those covering the ascent phase. The ascent phase begins shortly after the vehicle clears the launch tower and extends to the first staging event [47]. The range of Mach numbers bounding this phase of flight is from sub-transonic to high supersonic. In addition to varying the Mach number, a variety of values for angle of attack and angle of sideslip must be considered in order to envelope the likely flight trajectories the vehicle may experience. Attempting to provide aerodynamic data across this wide range of speeds and flight orientations requires data from a large number of Computational Fluid Dynamics (CFD) simulations. For example, the SLS ascent database products are primarily built from data taken from thousands of Reynolds-Averaged Navier-Stokes (RANS) CFD simulations. These databases are queried at a given point of the independent variables, such as Mach, angle of attack, and angle of sideslip, and output an interpolated response based on the data used to construct it. In the case of the line load and surface pressure databases, the data to construct the database is taken only from CFD simulations. However, the integrated force and moment database also includes data from wind tunnel tests. The wind tunnel data is considered high-fidelity data and is critical to include when it is available.

Each aerodynamic database should include an uncertainty model [7] that quantifies the uncertainty that goes into its calculated output. In the case of the integrated force and moment database, the task of blending the experimental and CFD data is quantified in the uncertainty model [19]. The idea being that instead of a single deterministic output from the databases, the output accounting for the uncertainty will instead be a statistical distribution of many possible outputs. The key challenge arises due to the other databases of interest, the line load and surface pressure databases, providing separate but still dependent outputs. The outputs are separate in the sense that the aerodynamic databases are distinct products based on their respective data types, so the integrated force and moment database is a separate product than the line load database. The dependency of the outputs can be seen as requiring that the output of the line load database at a given point needs to integrate to match the output from the force and moment database at the same point. The issue of this dependency also needs to hold true in the case of an uncertain output from those databases. On top of maintaining this integral consistency, the additional challenge is to generate uncertain line loads, and/or surface pressures, that corresponds to the quantified uncertainty in the integrated force and moment database. Investigations in developing the proper manner in which to generate these uncertain distributed loads are the focus of the included work.

The motivation for this work can be summarized by examining the following hypothetical scenario. An engineer looking at the loading on the vehicle is given a force and moment database, including uncertainty, from an aerodynamics team. The engineer generates an integrated normal force coefficient on the vehicle at the 3-sigma bound of the uncertainty, assume the uncertainty model provided follows a normal distribution. Then, the loads engineer wishes to know the surface pressure distribution on the vehicle that generated this normal force. Being able to provide this surface pressure, from a well founded procedure, is the core motivation underlying the included work.

1.1 Outline

Chapter 2 introduces the required background ideas that this work builds off of. Section 2.1 covers the concept of a line load and its definitions. Then, Section 2.2 discusses ascent vehicle aerodynamic databases with specific focus on the development process for NASA's Space Launch System vehicle aerodynamic databases. A brief introduction to uncertainty quantification is included in Section 2.3 in order to display how the ideas presented in this work are a novel progression of the field. Section 2.4 mentions applying uncertainty methods specifically to aerodynamic databases. Finally, Section 2.5 describes the uncertainty in distributed loads that is addressed by the methods in Chapter 3.

In Chapter 3 there is a brief discussion of existing and related work included in Section 3.1. The different methods developed for addressing distributed load uncertainty within aerodynamic databases are described in Section 3.2 and Section 3.3. Section 3.2 includes the Re-sampling method which is only feasible to be applied to line load data. Section 3.3 discusses the copula-based methods that can be applied to both line load and surface pressure data.

Chapter 4 includes the results, and corresponding discussions, from the various tests performed on the methods introduced in Chapter 3. Select figures do not add significantly to the discussions of the results and are instead included in A for the sake of completeness. The results are from performing the validation test, a leave-one-out error estimation, and a simulated uncertainty test. The validation test is an attempt to ensure each method can generate distributed loads that are governed by the distribution of the known data. The leave-one-out error estimation approximates the error of each method when generating distributed loads at known data points. The simulated uncertainty test uses each method in a modeled uncertainty scenario as actual Space Launch System data cannot be used without sanitization. Section 4.1 mentions the procedure for generating the data set that is used in the results contained within this chapter. Section 4.2 and 4.3 contain the various results using line load and surface pressure data respectively. Finally, Chapter 5 includes a distillation of the conclusions generated by this work and mentions potential avenues for future research work.

Chapter 2

Introduction

The development process of an ascent vehicle typically requires a complex organization of teams that can focus on specific aspects of the vehicle. One important aspect of this analysis is investigating the aerodynamics of the vehicle. Historically, this area would heavily focus on wind tunnel testing but the advancement of Computational Fluid Dynamics (CFD) has allowed for computer simulations to provide results that are not feasible to do experimentally. An important set of aerodynamic results, that involve both physical and computational results, are contained in aerodynamic force and moment databases. It is important to include results from both experiment and CFD as each data source has advantages and disadvantages so the combination will hopefully provide a more complete answer. These databases provide an estimate of the integrated force and moment coefficients on the vehicle over a space of potential flight conditions parameters such as Mach, angle of attack, angle of sideslip. There is a related database that provides the surface pressures on a large number of discretized points on the vehicle, over the same flight condition space. A more compact version of the surface pressure database is the line load database. Line loads can be viewed as a projection of these surface pressures onto a 1-dimensional array. These line loads can be estimated by dividing the vehicle into axial slices and calculating the load on each slice [13]. All these aerodynamic databases are constructed from data at discrete points in the parameter space of interest and can provide output at any requested parameter point by an interpolation method.

As mentioned prior, the force and moment database can contain both wind tunnel experiment and CFD simulation results. It is prohibitively expensive to completely measure line loads or surface pressures in a physical experiment and thus physical wind tunnel results are not included in these two databases. The differences between the databases becomes important when quantifying the uncertainty in their respective outputs. The force and moment database will have to include uncertainty stemming from experimental sources in addition to computational sources. As the forces and moments are scalar quantities, it is much more simple to utilize a relatively complete uncertainty treatment. Thus, with the force and moment databases having the uncertainty quantified, and including high value experimental results, the task is then to somehow incorporate this uncertainty into the line load and surface pressure databases. The challenge is then not only how to incorporate the quantified force and moment uncertainty, but also that the relationship between integrated value and distributed loads need to be maintained. The following sections in this chapter will build up the required background knowledge in order to understand the proposed solutions to these challenges contained in this work. First, the relationship between integrated and distributed load is defined in Section 2.1.

2.1 Line Loads

The discussion below utilizes the notation presented in [13]. The force and moment aerodynamic coefficients discussed follow the SLS coordinate system conventions and are given in the body axis coordinate system. Figure 2.1 show a diagram comparing the body axis and missile axis coordinate systems. The figure shows the six integrated aerodynamic force and moment coefficients that are used throughout this work. The axial force C_A, normal force C_N, side force C_Y, rolling moment C_{LL}, pitching moment C_{LM}, and yawing moment C_{LN}. The "L" in the moment notations stands for little and are sometimes referred to using the alternative notations of C_l, C_m, and C_n, representing the rolling, pitching, and yawing moments respectively. Finally, the figure also includes a visual comparison between the angle of attack α and angle of sideslip β_p against the total angle of attack α_p and roll angle ϕ_p.

The following discussion presents the line load relationships using normal force $c_N(x)$ and pitching moment coefficients $c_m(x)$. However, near identical relationships exist between

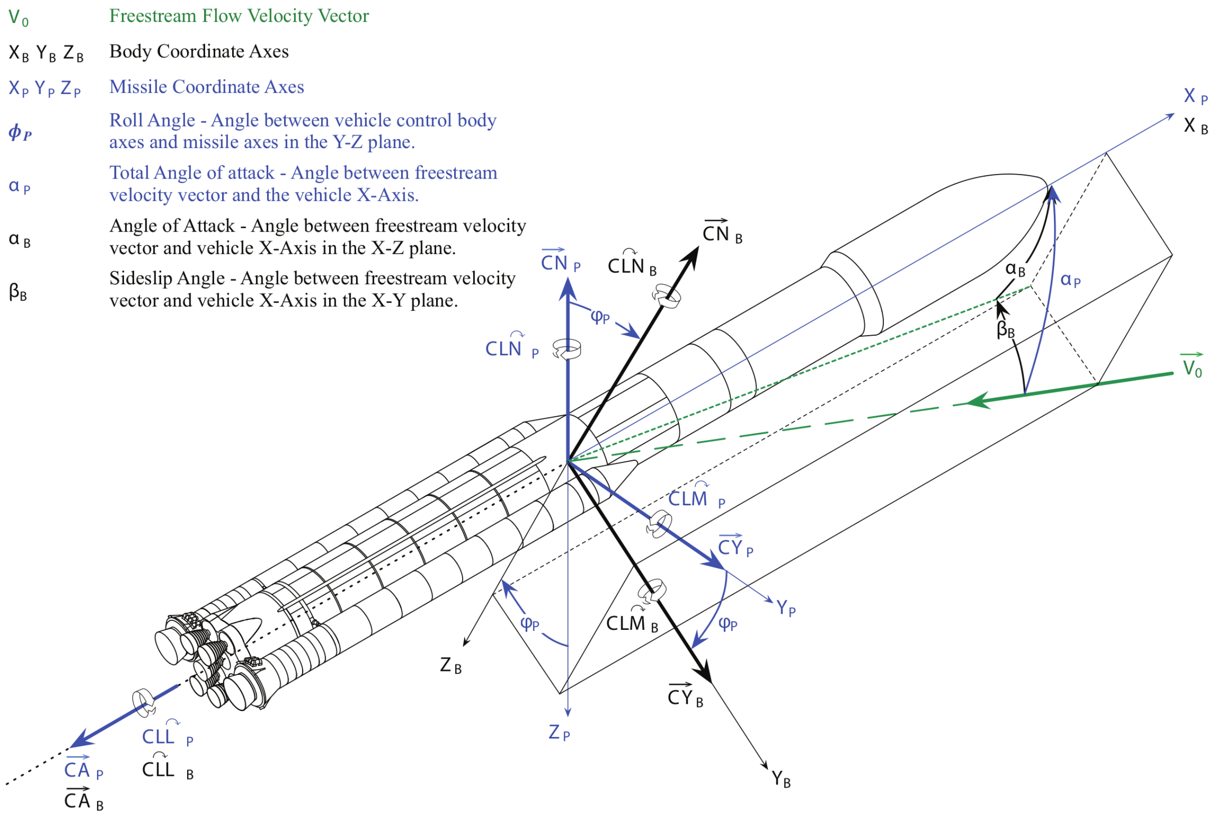

V0 Freestream Flow Velocity Vector
XB YB ZB Body Coordinate Axes
XP YP ZP Missile Coordinate Axes
ϕP Roll Angle - Angle between vehicle control body axes and missile axes in the Y-Z plane.
αP Total Angle of attack - Angle between freestream velocity vector and the vehicle X-Axis.
αB Angle of Attack - Angle between freestream velocity vector and vehicle X-Axis in the X-Z plane.
βB Sideslip Angle - Angle between freestream velocity vector and vehicle X-Axis in the X-Y plane.
XP
XB
CN B
CN P
CLN B
CLN P
φP
αB
αP
βB
V0
CLM P
CY P
YP
φP
CLM B
φP
ZB
ZP
CY B
YB
CLL P
CLL B
CA P
CA B

Figure 2.1: Definition of SLS coordinate systems [55]

side force $c_Y(x)$ and yawing moment $c_n(x)$. The line load notations for the two remaining aerodynamic coefficients are axial force $c_A(x)$ and rolling moment $c_l(x)$, these two coefficients do not have an equivalent relationship and are independent. Each CFD simulation results in an integrated normal force coefficient C_N and the corresponding line load profile $c_N(x)$, which gives the distribution of the normal force along the length of the vehicle. Likewise, the pitching moment C_m and its corresponding line load profile $c_m(x)$ are similarly defined. For convenience a single flight condition, or set of values for α, β, and M, is denoted by $\mathbf{r}$.

$$C_N = C_N(\mathbf{r}) \qquad\qquad c_N = c_N(x, \mathbf{r}) \tag{2.1}$$

$$C_m = C_m(\mathbf{r}) \qquad\qquad c_m = c_m(x, \mathbf{r}) \tag{2.2}$$

The line load profiles must integrate to equal the integrated loads, taking x_1 and x_2 to be the coordinate of the nose and tail of the vehicle respectively, the integration constraints are written as follows.

$$C_N = \int_{x_1}^{x_2} c_N(x)dx \qquad\qquad C_m = \int_{x_1}^{x_2} c_m(x)dx \tag{2.3}$$

Due to this integral relation, an alternate notation is sometimes used in this work to refer to c_N and c_m. These alternate forms are $\frac{dC_N}{dx}$ and $\frac{dC_m}{dx}$, respectively. There is also a relationship between the line load normal force and pitching moment distributions. Taking the moments to be calculated about a moment reference point x_{MRP}, the relationship is as follows.

$$c_m(x) = \frac{x_{MRP} - x}{L_{ref}} c_N(x) \tag{2.4}$$

Typically, these equations are written replacing the axial coordinate x with a non-dimensionalized version $\hat{x}{=}x/L_{ref}$, such that

$$c_m(\hat{x}) = (\hat{x}_{MRP} - \hat{x})c_N(\hat{x}) \tag{2.5}$$

$$C_N = \int_{\hat{x}_1}^{\hat{x}_2} c_N(\hat{x})d\hat{x} \tag{2.6}$$

$$C_m = \int_{\hat{x}_1}^{\hat{x}_2} (\hat{x}_{MRP} - \hat{x})c_N(\hat{x})d\hat{x}. \tag{2.7}$$

A similar relationship exists between the side force C_Y and the yawing moment C_n. The axial force C_A and rolling moment C_ℓ are independent.

The line load data set used is built from the results of 29 unique CFD simulations. Each line load is discretized into points at axial coordinates $\hat{x}_1, \hat{x}_2, ..., \hat{x}_n$ along the length of the vehicle. It is convenient to represent each line load as an array $\mathbf{c}_{N,i}$, shown below, the i being the case index corresponding to the CFD simulation results ran with the array of flight conditions $\mathbf{r}_i$.

$$\mathbf{c}_{N,i} = \begin{bmatrix} c_{N,i,1} \\ c_{N,i,2} \\ \vdots \\ c_{N,i,n} \end{bmatrix} = \begin{bmatrix} c_N(\hat{x}_1, \mathbf{r}_i) \\ c_N(\hat{x}_2, \mathbf{r}_i) \\ \vdots \\ c_N(\hat{x}_n, \mathbf{r}_i) \end{bmatrix} \tag{2.8}$$

2.2 Aerodynamic Databases

The development of the aerodynamic databases referenced to in this work is well documented and there are many published works from members of the SLS aerodynamics team. The databases described in these published works will be followed as the primary source for a modern and practical approach to constructing aerodynamic databases for ascent vehicles. After the successful first launch of the SLS Block 1 configuration during the Artemis I mission, a great deal of validation work has been done to justify the approaches taken to develop the SLS aerodynamic databases. These validations against actual flight data justifies following the SLS aerodynamic team's overall approach to aerodynamic databases. One such validation is that of the aerodynamic force and moment database by Shea et al. [62]. The result of which showed similar trends in ascent from Mach 0.2 up to Mach 3.5, with the largest discrepancy being axial force coefficient. This discrepancy is explained as being due to the difficulty reconstructing the axial force from the flight data where the relatively small aerodynamic force must be extracted from the overall vehicle thrust. Shea et al. recommended that no updates are required to future SLS force and moment databases based on their flight data comparisons. Another recent validation study is of the SLS ascent surface pressure databases against flight instrumentation data, shown by Meeroff et al. [47]. There were several points on the vehicle where the CFD and flight data did not match, primarily due to the thermal effects and geometry near those points being modeled with low accuracy. However, the overall comparisons strongly validate the choice of CFD methods used in the SLS Block 1 ascent surface pressure database.

As an ascent vehicle experiences many different flight phases during its operation, different aerodynamic databases are constructed to optimize for representing each different phase. Thus, the regimes of the vehicle's flight are typically split up into separate groups of aerodynamic databases. Each group containing individual databases that provide different output data such as force and moments, line loads, or surface pressures. Examples of the regimes that separate aerodynamic database groups focus on are the liftoff and transition period of the flight [19], the booster separation event [46], and the ascent phase [63]. The booster separation databases are particularly challenging and computationally expensive databases to generate. The challenges stem from the complex aerodynamic environment that occurs during booster separation as well as the high dimensional independent variable parameter space required [8]. The independent variables need to account for the orientation of the center-body, the relative position and orientations of the boosters, and the thrust levels of various engines. Recent work by Lee et al. [37] shows that there may be opportunities to reduce the dimensionality of the independent variable space by removing those which have little effect on vehicle or booster forces and moments. However, even using these recommended dimension reductions, a booster separation database would still need to include eleven independent variable parameters.

The databases covering the ascent phase are chosen to be the focus of this research. There are essentially three different types of ascent aerodynamic databases, providing the integrated forces and moments, the line loads, and the surface pressures on the vehicle during ascent. There is a potential use case for a database that addresses the aerodynamic loading on the vehicle's protuberances, however this is not addressed in this work. The surface pressure and viscous components of the discrete force and moment data is taken from the CFD simulation results [14] and is used as the base data for all three types. As mentioned previously, the integrated force and moments also includes data from wind tunnel tests [61]. There is also promising work in developing pressure sensitive paint methodologies for extracting the surface pressure data directly from wind tunnel experiments [57, 65, 64]. The comparison of steady surface pressures from RANS CFD to those from pressure sensitive paint in [45] indicated predicted pressure differences in areas of high geometric curvature and a failure to record experimental data in the locations between the boosters and the center body of the vehicle. These pressure sensitive paint methods also do not capture skin friction which is a contributing factor to line loads [56].

The details of developing the force and moment ascent database for SLS Block 1B Crew configuration [63] is used as a reference to explore the details of these types of databases. The methods used were based off previous methods developed for the Ares I ascent database [28] and SLS Block 1 Crew configuration [55]. The database is developed using a combination of wind tunnel and CFD results. The wind tunnel force and moment data was recorded using a 1.3% scale model of the SLS Block 1B Crew with several modelled protuberances. Two interconnected wind tunnels located at NASA Ames, the 11-Foot Transonic Wind Tunnel and 9- by 7-Foot Supersonic Wind Tunnel, were used to cover Mach ranges of 0.3 to 1.3 and 1.6 to 2.5 respectively. The NASA Langley Unitary Plan Wind Tunnel was used to cover the higher Mach numbers ranging from 2.5 to 4.5. The CFD data was taken from simulations using NASA's solver FUN3D at conditions that matched both the wind tunnel tests and the full-scale flight conditions. The output from the database is essentially a query to the response surfaces built from the CFD data at full-scale flight conditions with an included quantified uncertainty bound for each output coefficient. The specifics of the uncertainty quantification developed for this, and similar, aerodynamic databases are considered in Section 2.4. First, a brief background discussion about uncertainty quantification is provided in Section 2.3.

2.3 Uncertainty Quantification

Uncertainty Quantification Background

The task of uncertainty quantification is an aspect of finding a solution to an engineering problem that is becoming increasingly researched [10] [24]. The motivation being to attempt to provide the most complete prediction from a model, one that identifies and includes uncertainties, so that any engineering decisions based on that prediction can be as well-justified as possible. The maturation of the field and the increase in computational capabilities has allowed for the development of more comprehensive uncertainty quantification approaches. The best uncertainty quantification approach will depend on the characteristics of the problem and motivation behind its solution. A complete discussion of uncertainty quantification is not possible as the field is far too large, however it is possible to get a high level view of the motivating problems and a basic sampling of a few common methods. Before discussing different uncertainty quantification methodologies, it is useful to discuss the sources of uncertainty.

Uncertainty Sources

Sources of uncertainty are often broken into two categories, aleatoric and epistemic. The distinction between the two is defined by Jakeman et al. [29] as follows. Aleatoric uncertainty is uncertainty driven by the inherent variation of the system under consideration, it is irreducible. Epistemic uncertainty represents a lack of knowledge in any phase or activity of the modeling process. Acquiring additional knowledge could be gained in principle and would allow this uncertainty to be reduced. While differentiating these aid in discussing uncertainty, in practice categorizing an uncertainty can be difficult. Der Kiureghian and Ditlevsen [34] include a discussion that attempts to establish an uncertainty framework where these two uncertainty categories can be defined consistently. While the examples in the paper are from risk and reliability analysis, the conclusions presented are general enough to apply to most engineering settings. The conclusion presented by the paper is that all uncertainties are the same lack of knowledge and only make sense when considering from the perspective of a given engineering model. The separating between aleatoric and epistemic is a choice made by the model designer and indicates which uncertainties could be reduced, epistemic, and which are less likely to be reduced in the near term, aleatoric. Thus, the proper way to address these sources of uncertainty for a given problem typically depends on the methodology being used.

Methods applicable to epistemic uncertainty are less commonly researched. Jakeman et al. [29] argue that probabilistic representations of epistemic uncertainty with well-defined probability distributions is inappropriate as it imposes unjustified structure on the influence of these inputs. The authors suggest limiting this imposed structure by using alternative uncertainty quantification approaches based on evidence theory [51], possibility theory [16], or interval analysis [67]. Evidence theory is closely related to probability theory and estimates cumulative belief and plausibility functions that represent the uncertainty in a model output. Its representation of uncertainty approaches the probabilistic representation as the amount of information about the model input data increases [51]. Possibility theory is closely related to fuzzy set theory and utilizes two descriptions of likelihood, necessity and possibility, to characterize the uncertainty in function evaluations [27]. Interval theory uses only upper and lower bounds on the uncertain inputs to a model to generate similar interval bounds on the model output [48]. The proper choice of method is problem dependent and is based on the amount of information available to characterize the uncertainty.

Methods addressing aleatoric uncertainty tend to be the focus of more research effort, with generalized polynomial chaos (gPC) being a popular sampling-based approach. It is built on the 1938 work on chaos by Weiner [71], with Ghanem and Spanos [21] generally credited with introducing the basic approach to engineering applications, followed by Xiu and Karniadakis [75] generalizing the approach. Solving for the coefficients in a polynomial chaos model is typically broken into intrusive and non-intrusive approaches. The intrusive approach requires manipulations of the governing equations, with the best-known method from this approach being the stochastic Galerkin method discussed in [22]. The non-intrusive approaches don't require a potentially complex manipulation of a problem's governing equations. Examples of this approach are sparse quadrature [32] and stochastic collocation [40]. The core idea is that one can represent a random variable as a function of a random vector by using a polynomial basis that is orthogonal to the distribution of this random vector. An in depth mathematical description of this can be seen in [15]. This representation of a random variable is the response gPC and its coefficients are solved for by using either an invasive or non-invasive approach mentioned prior. Once a sufficient gPC approximation is calculated it is an analytical representation of the output of interest in terms of the random uncertain inputs. Thus, practically all statistical information for the output can be estimated. This style of uncertainty problem is known as forward propagation. This is one of two common uncertainty quantification archetypes, the other common archetype being backward propagation.

Uncertainty Propagation

Forward propagation refers to the process of quantifying the uncertainty in the outputs of a model or system due to uncertain inputs. Lee and Chen [38] performed a study comparing several forward propagation approaches. This includes the aforementioned Polynomial Chaos method as well as the full factorial numerical integration method and the univariate dimension reduction method. The former attempting to estimate the statistical moments of the output of interest using an appropriate quadrature formula for direct numerical integration. The latter approximating a multivariate function with multiple univariate functions which can be used to calculate multivariate statistical moments. Another family of methods to handle forward propagation are Monte Carlo based methods [76], [31]. These methods have the advantages of being non-intrusive, flexible, and simple to implement. However, they are slow to converge, $\mathcal{O}(N^{-\frac{1}{2}})$ with N being the number of function evaluations. Thus the overall cost of the method greatly depends on how expensive each function evaluation is to run. There are approaches to try and reduce the cost. Quasi-Monte Carlo [6] employs low-discrepancy sequences, such as a Sobol sequence, in order to improve the convergence rate to $\mathcal{O}(N^{-1})$. Multilevel Monte Carlo methods [23] reduce the computational cost by performing the majority of function evaluations at a low accuracy and incorporating a small amount of high accuracy evaluations. A closely related method is Multi-fidelity Monte Carlo [20]. This approach is a generalization of the multilevel idea where a majority of function evaluations are taken from low-fidelity models, typically models with known simplifications or neglected physical effects to drive down cost. An example of this for a CFD application would be using cheaper 2-dimensional simulations or inviscid flow solvers instead of a fully 3-dimensional direct numerical simulation of Navier-Stokes.

The problem framework for backward propagation, sometimes referred to as inverse uncertainty quantification or an inverse problem, is where output results from a trusted source are compared with outputs from a model. The trusted source can be any sort of data considered higher fidelity in comparison to the model output, such as data from physical experiments. These comparisons are used to quantify, and ideally reduce, the bias and parameter uncertainty in the model. A large portion of the recent research in this area is focused on applications for machine learning and neural network training processes. A key issue with an inverse problem is that they are generally ill-posed, in that small perturbations in the data may lead to large errors in the inversion estimates [4]. An extensive survey of existing backwards uncertainty propagation methods, applied to uncertainty in inputs to nuclear system thermal-hydraulics codes, can be found in work by Wu et al. [74]. The authors compare and evaluate 12 different inverse uncertainty methods to quantify the uncertainties of input parameters to their system thermal-hydraulics codes based on experimental data. Wu et al. identify three different groups of inverse uncertainty quantification (IUQ) methods:

frequentist, Bayesian, and empirical. "Frequentist IUQ tries to identify most likely parameter values, with which the TH model can reproduce the experimental data. Bayesian IUQ targets at finding parameter uncertainties that can explain the disagreement between model and data, typically with MCMC sampling. Empirical IUQ seeks parameter uncertainties with which the model predictions can envelop the measurement data to a desired level."

Regularization methods are one type of approach to solve these inverse problems at the heart of the frequentist perspective to backwards propagation [2]. For example, start with a canonical example of an ill-posed inverse problem $Ku = f$, with K having an unbounded inverse. The general approach is to find some parametric approximation to the linear operator K with better stability properties. The next solution approach is to utilize a Bayesian framework, which is a natural fit with the problem style. In these methods, such as in [41], the unknown model parameter is modelled as a random variable and additional information from output observations update its probabilistic description. The last solution approach is the empirical approach. An example of an approach in this group is the non-parametric clustering method developed in [68], which applies a multi-dimensional clustering technique based on comparing model error samples with the Kruskall-Wallis test.

2.4 Database Uncertainty

There are several published works on uncertainty quantification methods that are specifically applied to the aerodynamic databases developed to support the SLS. The uncertainty quantification of the SLS Block 1B Crew integrated force and moment database can be seen in the work by Shea et al. [63]. The total uncertainty in [63] is built up by combining quantified uncertainties from the two data sources, wind tunnel data and CFD data. The total uncertainty, U_{final}, is the root-sum-square of the quantified uncertainty sources multiplied by an additional factor of safety of 1.05. All the different quantified uncertainty sources are briefly discussed here in order to get an idea of the database uncertainties that are explicitly addressed. U_{DBM} accounts for smoothing, fitting, or averaging the source data in order to construct the final database. U_{tst} which estimated how well the database predicted test data that was left out of the construction of the database. U_{G2F} which was intended to cover uncertainties due to sting and wall interference in the wind tunnel, OML differences between the wind tunnel and flight OMLs, Reynolds number effects, and plume effects. U_{WTcorr} which accounted for corrections applied to raw balance data. U_{bal} which accounted for balance calibration fit error in the balance used for measuring the forces and moments in the wind tunnel tests. U_{rpt} which accounted for experimental error in the wind tunnel data due to repeatability. The total uncertainty is an estimate to the amount of uncertainty included in an integrated force and moment value output from the database. This quantified uncertainty step for the integrated force and moment data, or an similar equivalent quantification step, is assumed to be done before the methods discussed in Chapter 3 are then used.

The uncertainty quantification of an ascent booster separation database is shown in [8]. A similar breakdown of the contribution of each quantified uncertainty term is presented, with the database interpolation term contributing the largest uncertainty to the boosters forces and moments. This can be seen as a direct consequence of the high dimensionality of the independent variable space. The dominance of the interpolation uncertainty lead to the work by Lee et al. [36]. Lee aimed to reduce specifically the interpolation error by using a data-driven approach to selecting which points in the independent variable parameter space should be studied by CFD and experiments. In addition, the parameter spaces were skewed according to the covariance of the independent variables observed in a set of simulated trajectories.

2.5 Distributed Load Uncertainty

The methods discussed in Chapter 3 become relevant once all the ascent aerodynamic databases have been constructed and an estimated uncertainty in the integrated force and moment database has been quantified. The goal of these methods does not fall under the typical scope of work addressed in existing uncertainty quantification literature. Uncertainty quantification is typically concerned with forward propagation of a set of uncertain independent variables in order to quantify their impacts on dependent quantities of interest. Forward propagation in an aerodynamic database scenario is not practical as it would be prohibitively expensive and the contribution of each source of uncertainty is typically not known. In the scenario addressed in this work, the step of quantifying the uncertainty has already been performed on the integrated force and moment database and the goal is to attempt to generate uncertain distributed loads that correspond with this uncertainty. This goal aligns closer to that of the inverse problems solved with backwards propagation of uncertainty, but it does not completely fit within these approaches either.

The core motivation behind this goal, comes from the SLS aerodynamic databases development procedure, in which the integrated force and moment database contains high-fidelity wind tunnel data and the distributed load databases do not. The objective is essentially attempting to incorporate the uncertainty that is quantified on the high-fidelity integrated forces and moments onto an estimated uncertainty for the distributed loads. There are published attempts to develop methods that address this task in [13, 73]. In depth analysis of these existing published works are discussed further in Chapter 3.

However, the primary aspect that these early attempts are missing is that they generate their uncertain distributed loads without considering the spatial relationships contained in the CFD data. These spatial relationships can be approximately represented by calculating the covariance matrix of the distributed load data. However, the covariance only captures linear relationships between the spatial locations implied by the distributed loads. In order to more completely capture any possible non-linear relationships, the full probability distribution function of the known distributed data should be considered. The probability distribution function of the distributed load data is taken to be a statistical representation of the physics driven spatial relationships captured within the known CFD data. Generating a line load that does not follow the probability distribution function is then at risk of violating the underlying physics. This risk is the primary motivation for attempting to preserve these relationships, along with the idea that a generated uncertain distributed load that follows the spatial relationships is more valid than an uncertain distributed load that does not. In other words, following these spatial relationships restricts the possible distributed load shapes to those seen in the CFD data and are thus more physically realistic than distributed loads generated through a procedure that does not consider these relationships. Since there are infinitely many possible distributed loads that can be generated for a given integrated value, or values, it is important to be able to justify that the method used to generated uncertain loads is not arbitrarily chosen. The absolute justification would be checking if the flow fields that would produce the distributed loads obey the governing equations used in the CFD. An approach that could do this is an avenue for future research but restricting the generated distributed load distributions to the known CFD distributions is currently taken as a small step towards that eventual goal.

A test needs to be used to determine if the distributed loads that are generated for the uncertainty estimation follows the probability distribution of the known CFD data. Two-sample tests are statistical tests that can be used for this purpose. These tests determine if there is a statistically significant difference between two different distributions by comparing samples drawn from each. There are several of these tests to choose from when comparing univariate samples, such as the Kolmogorov-Smirnov test [66]. However, the use of statistical two-sample tests to compare high dimensional data is an active area of research, and the line loads and surface pressures are certainly considered high dimensional data.

The two-sample test chosen in this work is the test developed by Gretton et al., the Maximum Mean Discrepancy (MMD) two-sample test [25], [26]. This test is commonly used in machine learning work and there is an extensive list of published research works that have implemented it successfully [39, 70, 54]. It compares samples drawn from two different distributions by finding a smooth function that has large values on the points drawn from one distribution and small values on points drawn from the other. The differences between the mean function values on the two samples are used as the test statistic. Gretton et al. describe three potential non-parametric two-sample tests in [26] but the one used in this work is the third test that is based on the asymptotic distribution of the maximum mean discrepancy test statistic. The recommendation of Gretton et al. to use a kernel bandwidth equal to the median distance between points in the aggregate of the data to be compared and the bootstrapping approach to determine the critical score are both followed when the MMD test is used in this work.

As mentioned prior this type of high-dimensional testing is an active area of research and there are alternative different approaches to the one selected. An example would be the work in [18], which used kernel Fisher discriminant analysis to provide them a consistent nonparametric test statistic for two-sample testing. There is also continued development on improving the maximum mean discrepancy two-sample test itself, such as the work in [42] that uses kernels parameterized by deep neural networks in order to maximize testing power.

Validation Test

As described in Section 2.5, the goal for the methods developed in this work, described in Chapter 3, are to generate distributed loads that follow the probability distribution of the distributed loads from the CFD data. The test used to compare is the previously discussed Maximum Mean Discrepancy two-sample test. However, if the methods are applied to an uncertain integrated force and moment at a single point the resulting distributed loads are not expected to match the overall distribution of the CFD data. It should be restricted to a limited space of the distributed load probability distribution corresponding to the uncertain integrated value bounds. Instead of attempting to test this limited space, the validation test used in this work is as follows. First, N_d values are drawn from the integrated forces and moments data, with equal probability and no replacements. N_d was set to 100 for both the line load and surface pressure validation test. Then, the method will generate a distributed load that integrates to each of these drawn values. The resulting collection of distributed loads are then tested, with the Maximum Mean Discrepancy two-sample test, against the known distributed loads to determine if the difference between the generated distribution and the known distribution is statistically significant. A significance level of 0.05 is chosen for this validation test. This level is the chosen upper bound on the probability of falsely rejecting that the two samples come from the same distribution [26]. If the method can generate distributed loads that pass this validation test then it is considered validated as a method that can generate distributed loads that follow the known distributed load distributions from CFD. The validated method can then be used with confidence to generate distributed loads for an uncertain integrated force and moment at a single point, such as is done in the simulated uncertainty examples in Chapter 4.

Chapter 3

Methods

3.1 Existing Related Works

The previous works by Dalle et al. [13] initially investigated the approach of generating uncertain line loads by using their developed adjustment procedure. Their core idea is to find a method to adjust known line load data so that the resulting line load integrates to a chosen target force and moment. To do this, the authors first used the singular value decomposition of a matrix including all the line loads at a given Mach number, referred to as a snapshot matrix. This is done in order to extract the left singular vectors of the snapshot matrix and use these as orthonormal basis vectors to construct their adjustments. The adjustments are then calculated by setting up an optimization problem and solving it via the method of Lagrangian multipliers.

The optimization problem seeks to find an adjustment using a linear combination of the left singular vectors such that the adjustment integrates to a target integrated force, and moment, in some cases. The "cost" of using each of the left singular vectors in the optimization problem is chosen to be a function inversely proportional to their respective singular values. This cost is included in a squared sum of the linear combination coefficients in order to bound the potential magnitude of these coefficients. The authors suggests implementing the idea of an adjustment profile at each Mach number in order to save computation time if the process was to be implemented in a practical setting. This essentially means that one adjustment mode shape is calculated for a target force or moment value set to one. These adjustment profiles can then be scaled from unity to any target force or moment and applied easily. This approach was then later applied to surface pressures directly in a later work [12]. The adjustment procedure remains essentially the same as in the line load work, with some slight linear algebra optimizations in order to minimize the additional cost of working with the much larger surface pressure data.

An example of the application of this adjustment method in an uncertainty setting is given as well. The basic idea is, given statistical representations of uncertain integrated forces and moments, target integrated values can be drawn and the method can then generate line loads, and surface pressures, that integrate to the target values. The collection of generated distributed loads are then considered to be the uncertain line loads, or surface pressures, that correspond to the uncertain integrated forces and moments. This overall idea of the using a method to generate uncertain distributed loads according to draws from uncertain integrated values is carried on into the methods presented in Chapter 3. However, one issue with the approach of Dalle et al. is that the adjustment profile is only dependent on Mach number. Once it is calculated, it applies the same adjustment mode over the whole angle of attack and angle of sideslip parameter space. It is constraining the possible spatial uncertainty in the distributed loads to the same locations whether the vehicle is flying at high angles or at very low angles. Another issue is that the adjustment profile is strictly a result of a numerical optimization procedure and is not required to follow the spatial relationships that can be observed in the CFD data. In other words the adjustments satisfy the optimization procedure but there's not guarantee that they obey the physics underlying the results seen in the CFD data. The methods presented later in this chapter will address both of these issues.

There are related works from other members of the SLS team such as Wignall et al. [72]. The underlying adjustment approach presented in this work is quite similar to that in [13]. However, the process is described from the perspective of calculating the eigenvalue and eigenvectors of covariance matrices from the CFD data. It uses a different cost function in the optimization process but the overall adjustment procedure is the same. The approach thus suffers the similar underlying issues as in [13] of determining the line load from an optimization procedure. However, one change of note is that Wignall allows their line load generation model to vary with angle of attack and angle of sideslip.

The author goes on in future work to integrate their method in a more complete treatment of database uncertainty [73]. The total uncertainty in the line loads is presented as a root-sum-square of standard deviations from model form and data fusion uncertainties. The model form uncertainty stems from the interpolation error in using the model, estimated by a Leave-One-Out approach. The data fusion uncertainty arises from using the model to propagate the uncertainty in the integrated force and moment database. The additional development to incorporate the method's uncertainty into the overall quantified database uncertainty is a necessary step in order to provide a complete uncertainty model with the aerodynamic databases. This step of accounting for the model error by leave-one-out error estimation is included in the results shown in Chapter 4.

3.2 Resampling Method

An initial attempt at generating uncertain line loads that follow the statistical relationships is shown in [5]. The method begins by utilizing the Singular Value Decomposition (SVD) in order to generate the basis vectors which are then used to construct new potential line loads. The known CFD line load arrays are stacked into a snapshot matrix. For example, the snapshot matrix for the normal force profiles $\mathbf{c}_{N,i}$ would be $\mathbf{C}_N = \left[\mathbf{c}_{N,1}, \mathbf{c}_{N,2}, \ldots, \mathbf{c}_{N,n}\right]$. This matrix has the dimensions $n \times n_c$, with n being the number of axial slices in each line load and n_c being the number of line load cases. Before performing the SVD, the average value over the n_c cases for each axial line load slice is subtracted from $\mathbf{C}_N$. The resulting matrix, $\mathbf{C}'_N$, represents a line load perturbation matrix and is the matrix on which the SVD is performed as

$$\text{SVD}(\mathbf{C}'_N) = \mathbf{U}_N \mathbf{\Sigma}_N \mathbf{V}_N^T. \tag{3.1}$$

The columns of the left singular vector $\mathbf{U}_N$ contain the orthonormal bases that will be used to construct the line load dispersion. The number of bases resulting from this depends on n_c; it will provide up to n_c orthonormal bases. The number of orthonormal bases kept for the next step will be labeled κ, however all n_c bases were kept for the results shown in Chapter 4. First, the κ selected basis vectors from $\mathbf{U}_N$ are used to build a linear estimation to each of the n_c line loads used in the snapshot perturbation matrix $\mathbf{C}'_N$. This is done by solving the linear system below for the coefficients ω_i, where the matrix $\tilde{\mathbf{U}}_N$ contains the κ bases vector columns kept from $\mathbf{U}_N$.

$$\tilde{\mathbf{U}}_N \omega_i = \mathbf{c}_{N,i} \tag{3.2}$$

The result is n_c arrays of κ coefficients, that can be stacked into a $\kappa \times n_c$ matrix, $\boldsymbol{\Omega}$. This procedure can be seen as equivalent to $\boldsymbol{\Omega}{=}\boldsymbol{\Sigma}_N \mathbf{V}_N^T$ when $\kappa{=}n_c$. The next step is to generate line load dispersions using coefficients that follow the joint distribution of the coefficients in $\boldsymbol{\Omega}$. With the goal being to use these coefficients as a lower dimensional proxy for the full joint distribution of the known line load data.

In order to achieve this, the first coefficient is separated from the rest of the $\kappa{-}1$ coefficients. It will be used later in order to match the target integrated coefficient for the generated line load. Next, a novel draw for the $\kappa{-}1$ coefficients from their joint distribution can be made by using an approach that effectively samples from a Gaussian Kernel Density Estimate (GKDE) [59] of the distribution. To start, a uniform weighting is assigned to each of the n_c columns in the known coefficient matrix $\boldsymbol{\Omega}$ and a column of known coefficients is randomly chosen. Then a small perturbation is created for these $\kappa{-}1$ coefficients. The perturbation is generated by sampling a centered multivariate normal distribution with a scaled covariance matrix. This covariance matrix $\boldsymbol{\Upsilon}$ is calculated by multiplying a scalar, effectively the equivalent to the GKDE bandwidth, by the covariance matrix of the known coefficient values. The newly generated coefficient can then be estimated by,

$$\omega_j^* = \omega_j + \delta\omega_j \tag{3.3}$$

$$\delta\omega_j \sim \mathcal{N}(0, \boldsymbol{\Upsilon}). \tag{3.4}$$

Where ω_j is the randomly selected coefficient vector from the known data, and ω_j^* is the resulting new draw that approximately follows the distribution of the known coefficients.

Next, the first coefficient that was left out of this process is used so that the resulting line load when multiplying the SVD bases by these coefficients will equal the target integrated value. In order to do this, the resulting integrated value of ω_j^* using all but the unused first coefficient is estimated. Then the difference between this value and the target is used to algebraically solve for the first coefficient. However, the relationships between the first coefficient and the rest must be enforced somehow as they are neglected during the algebraic solving procedure. In order to approximate these relationships, the two-dimensional joint distributions between the first coefficient and the remaining $\kappa-1$ coefficients are estimated by GKDE. Then the probability density of the $\kappa-1$ coefficient pairs, each pair containing the first coefficient and one of the remaining $\kappa-1$ coefficients, is estimated. If any of these probability densities are below a set filter level $p_{\min}$, then the results are filtered out and the process is repeated from the start with a new initial randomly drawn sample of coefficients ω_j^* until the minimum probability density level is achieved.

It is relatively straightforward to extend the approach to simultaneously match a target integrated force and moment. In this case, two coefficients are solved algebraically and remaining $\kappa-2$ coefficients are drawn from the GKDE sampling procedure described prior. The force and moment from using the $\kappa-2$ coefficients is estimated and the first and second coefficients are solved for by using the remaining differences in integrated force and moment. Then the unenforced relationships between the first, second, and the rest of the coefficients must be addressed. The distributions of the coefficient triplets, each triplet being the first, second, and one of the remaining coefficients, are estimated by GKDE. Much the same as before, the probability density for these triples are estimated and tested against a chosen filter level $p_{\min}$. If there is a probability density below the filter level, the entire coefficient generation process is repeated until all probability densities are above the filter level. The first and second coefficients, corresponding to SVD bases with large singular values, are chosen to be algebraically solved as they have the greatest potential to change the integrated value while still being contained by the coefficient's joint distribution.

3.3 Copula-based Methods

The method is described as it would be applied to surface pressures, as line loads can be viewed as a simplified version of the surface pressure problem. This is because the line loads are smaller arrays and require fewer constraints on their integrated values when compared to surface pressures. Similarly to the work in the previous section and in [13] and [5], the dispersion method begins by essentially performing Principal Component Analysis (PCA) on the surface pressure data. The PCA procedure is repeated here for completeness, however it is the same description provided at the start of Section 3.2. The description of the new aspects of the copula-based method's procedure starts after the coefficient matrix $\mathbf{\Omega}$ is calculated.

First, the Singular Value Decomposition is utilized in order to generate the basis vectors which are then used to construct new potential surface pressures. The 29 surface pressure arrays are stacked into a snapshot matrix. For example, the snapshot matrix for the normal force profiles $\mathbf{c}_{N,i}$ would be $\mathbf{C}_N = \left[\mathbf{c}_{N,1},\ \mathbf{c}_{N,2}, ..., \mathbf{c}_{N,n}\right]$. This matrix has the dimensions $n \times n_c$, with n being the number of axial slices in each line load and n_c being the number of line load cases.

Before performing the SVD, the average over the n_c cases for each surface pressure point is subtracted from $\mathbf{C}_N$. The resulting matrix, $\mathbf{C}'_N$, represents a surface pressure perturbation matrix and is the matrix on which the SVD is performed, as shown below

$$\mathrm{SVD}(\mathbf{C}'_N) = \mathbf{U}_N \mathbf{\Sigma}_N \mathbf{V}_N^T. \tag{3.5}$$

The columns of $\mathbf{U}_N$ contain the orthonormal bases that will be used to construct the surface pressure dispersion. The number of bases resulting from this depends on n_c; it will provide up to n_c orthonormal bases. The number of orthonormal bases kept for the next step will be labeled κ, however all n_c bases were kept for the results shown in Chapter 4.

First, the κ selected basis vectors from $\mathbf{U}_N$ are used to build a linear estimation to each of the n_c surface pressures used in the snapshot perturbation matrix $\mathbf{C}'_N$. This is done by solving the linear system below for the coefficients ω_i, where the matrix $\tilde{\mathbf{U}}_N$ contains the κ bases vector columns kept from $\mathbf{U}_N$.

$$\tilde{\mathbf{U}}_N \omega_i = \mathbf{c}_{N,i} \tag{3.6}$$

The result is n_c arrays of κ coefficients, that can be stacked into a $\kappa \times n_c$ matrix, $\mathbf{\Omega}$. If $\kappa = n_c$, the coefficients $\mathbf{\Omega}$ can be found directly from the product of $\mathbf{\Sigma}_N \mathbf{V}_N^T$. The next step is to generate surface pressure dispersions using coefficients that follow the joint distribution of the coefficients in $\mathbf{\Omega}$. With the goal being to use these coefficients as a lower dimensional proxy for the full joint distribution of the known surface pressure data. A test of the credibility of this assumption is investigated in the various validation test results shown in Chapter 4.

After the coefficients $\mathbf{\Omega}$ are found, the next step is to use them in order to generate new surface pressures that could come from the distribution implied by the known data while also integrating to given target forces and moments. The statistical concept of a copula [50] motivates a method to achieve this. A copula is effectively a decomposition of a multivariate statistical distribution into a multivariate distribution containing the dependence structure between the variables and univariate marginal distributions for each variable. Start with a random vector $(y_1, y_2, ..., y_d)$, with a continuous marginal cumulative distribution function (cdf) in each dimension $F_i(y) = P(y_i < y)$. Applying the marginal cdfs to the random vector results in

$$(u_1, u_2, ..., u_d) = (F_1(y_1), F_2(y_2), ..., F_d(y_d)), \tag{3.7}$$

with each u_i uniformly distributed on $[0, 1]$. The copula of the random vector $(y_1, y_2, ..., y_d)$ is then the joint cumulative distribution function $C(u_1, u_2, ..., u_d)$. Similarly, the probability density function of the random vector can be written as

$$p(y_1, y_2, \ldots, y_d) = c(F_1(y_1), \ldots, F_d(y_2))p_1(y_1), \ldots, p_d(y_d). \tag{3.8}$$

Where $c(F_1(y_1), \ldots, F_d(y_2))$ is the probability density function of the copula C and p_i is the probability density of the *i-th* variable y_i [30]. In this use case, the copula is used in order to pseudo-randomly draw variables from the distribution underlying the random vector. The process to draw a new random vector sample x^* simply takes a random sample from the copula $u^* = (u_1^*, u_2^*, ..., u_d^*)$ and applies the inverse marginal cdfs F_i^{-1} as

$$x^* = (F_1^{-1}(u_1^*), F_2^{-1}(u_2^*), ..., F_d^{-1}(u_d^*)). \tag{3.9}$$

Applying this sampling process to the coefficients $\mathbf{\Omega}$ to draw a new coefficient sample ω^* is done by first approximating the copula of $\mathbf{\Omega}$. This is done with a multivariate Gaussian distribution $\mathcal{N}(0, \sigma)$, with σ being the estimated covariance matrix of $\mathbf{\Omega}$. A sample is drawn from $\omega' \sim \mathcal{N}(0, \sigma)$, then the marginal cdfs of $\mathcal{N}(0, \sigma)$ are applied to ω' resulting in ω^u. This new sample is bound by the interval $[0, 1]$ but maintains the dependence structure of $\mathcal{N}(0, \sigma)$, and thus approximately the dependence structure of $\mathbf{\Omega}$.

Finally, the inverse marginal cdfs of $\mathbf{\Omega}$ are applied in each dimension to approximately recover a novel coefficient sample $\omega^* = (F_1^{-1}(\omega_1^u, F_2^{-1}(\omega_2^u, ..., F_d^{-1}(\omega_d^u))$. The integral of the surface pressure generated from the new coefficient sample can be any value. Since the goal is to generate a surface pressure that integrates to target integrated values, then a small modification can be made the copula. The idea of a conditional copula, as shown in [53], can be used for this purpose. Assuming the conditioned variable is W, then a copula conditioned for a given realization of $W{=}w$ is

$$p(y_1, y_2, \ldots, y_d|w) = c(F_1(y_1), \ldots, F_d(y_2)|w)p_1(y_1|w), \ldots, p_d(y_d|w). \tag{3.10}$$

In the sampling use case discussed here, the conditioned variables are the target integrated values, so the conditioned inverse marginal cdfs are now applied as $F_i^{-1}(\omega_i^u|C_A, C_N, ...)$. The conditioning of the copula, which is being modelled with a multivariate normal, results in a new multivariate normal with a mean and covariance given by [17]

$$\bar{\mu} = \mu_1 + \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}(\mathbf{a} - \mu_2) \tag{3.11}$$

$$\bar{\mathbf{\Sigma}} = \mathbf{\Sigma}_{11} + \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}. \tag{3.12}$$

As discussed prior this is imposing a structure on the data by estimating the copula with a normal distribution. There is another approach that can be taken, however it will be discussed later in this section.

Drawing samples from this conditioned copula will now result in surface pressures that are normally distributed about the target integrated values. The inverse conditioned marginal cdfs used are based off using Gaussian kernel density estimation to approximate the joint probability distribution $p(\omega_i, C_A, C_N, ..., C_{LN})$. The n_c known cases are used in the construction of these kernel density estimates, as well as 100 cases from the 5000 radial basis function sampled values. The n_c known cases are given a weight of 5, so they have a impact on the output probability density estimate than the 100 sampled data. The sampled data is included to better fill out the data space. The bandwidths used in these inverse conditioned marginal cdfs are a free parameter. The choice of kernel function can also be considered free parameter, however the bandwidth generally affects the output to a much larger degree.

There exists rule of thumb calculations to set the bandwidths [60] or methods based on criteria such as cross-validation [58]. However, the optimal choice for choosing a bandwidth is an ongoing field of research.

The approach used for selecting the bandwidths in this work was motivated by the validation test discussed in Section 2.5. The idea being that while generating a line load for a known value, as the bandwidth converges to zero, the most probable generated line load should converge to the known line load. In other words, the method should output a distributed load very similar to the known distributed load if the target integrated values match a known case. This assumes that the copula is a decent approximation to the multivariate relationship it is representing. However, a kernel density estimate with a vanishingly small bandwidth has no capability to estimate the probability density of values beyond those that are used in its construction. The bandwidth cannot be too large either as the bandwidth increases, the overlapping nature of the kernel density estimates means that, for a known target integrated value, the most probable line load will likely be different than the corresponding known line load. The objective in creating a procedure to select the bandwidths for the copula methods was thus to balance the issues that arise when it is too small or too large.

The chosen bandwidth selection procedure is described as follows. Each bandwidth, corresponding to each of the κ coefficients, is initiated at 0.6. This is slightly larger than the values suggested by the rule of thumb estimates mentioned prior. Then for each coefficient ω_i, the initialized bandwidth is used to estimate a kernel density such as $\tilde{p}_i(\omega_i, C_i, C_j)$, with C_i and C_j being an integrated force and its corresponding moment respectively. The $\tilde{p}_i$ is then conditioned on each of the n_c known values for C_i and C_j. On each of the conditioned $\tilde{p}_i$, the most probable value of ω_i is estimated and if it is not within 0.1 percent of the known ω_i the bandwidth is decreased by 0.01 and the process is repeated. The lower limit set on the possible value of each bandwidth is 0.05. Once the bandwidths are found, the result is a $\kappa \times n_c$ matrix of bandwidths based on the known integrated values. The high-level view of this procedure is to gradually reduce the amount of overlapping data due to a large bandwidth until the known coefficient has the highest probability density when $\tilde{p}_i$ is conditioned on the corresponding known integrated values. The idea is then to linearly interpolate these bandwidths to estimate a set of bandwidths for any target integrated values. A floor is put on the bandwidths output from the interpolation as too small of a bandwidth value caused issues with the inverse conditioned marginal distribution estimation. The floor depends on whether line load or surface pressure data is being used, with the floors used being 0.05 and 0.125 respectively. The difference in values is due to the increase in dimensions that are conditioned when using surface pressure data. This method has obvious potential issues with extrapolation, however the line load generation method in general is not intended to be used as an extrapolation tool for integrated values beyond those seen in the known data.

Once the bandwidths have been estimated, the conditioned probability distribution, $p(\omega_i|C_A, C_N, ..., C_{LN})$, can be numerically estimated by sampling p over the domain of values of ω_i with fixed values for the integrated coefficients C_A, C_N, etc. Using Gaussian Kernel Density estimation on this 7 dimensional data distribution is pushing the limits of the technique. Previous work suggests 6 dimensional data as the upper limit for an effective application of the technique [59], however recent work suggests it can perform poorly in as few as 4 dimensions [69]. It is possible to get around this dimensionality problem by conditioning the inverse marginal cdfs on a smaller subset of the integrated aerodynamic coefficients. This leaves the unconditioned variables to vary somewhat freely, however this may be acceptable since there are correlations between say C_N and C_{LM} and conditioning on one will have an affect on the values the other can take. The rotational moment C_{LL} is also quite small and not a large factor in the data set used for this work so leaving it out of the conditioning step could be justified. However, the work presented here simply uses the 7-dimensional data as mentioned prior. In the case of line loads, there are only a maximum of 2 constraints, a force and moment pair, and thus there is no similar issue with high dimensionality.

Since many samples can be drawn for a single target integrated force and moment, there is an open decision to make about how to select a sample to output. There are two criteria that seem reasonable, the first is to choose the sample with the smallest error to the target integrated coefficients. A tolerance on this error could also be set and if the best sample has an error above the tolerance then a small correction can be applied using the adjustment mode ideas in [13]. If the error is small enough the correction to the coefficients should also be small enough to maintain statistical consistency. The other reasonable criteria could be to select the sample that has the highest probability of being drawn, which can be estimated by Equation 3.8. However, for the case of using a multivariate normal for the copula, the $c(F_1(y_1), \ldots, F_d(y_2))$ term in Equation 3.10 is neglected. This assumption essentially assigns each draw from the copula an equal probability density and thus relies on the $p_i(y_i)$ marginals to differentiate the probability densities. The estimation of the copula joint probability density is a high dimensional problem and is thus not a reasonable use case of Gaussian Kernel Density estimation. It falls into high-dimensional density estimation, which is an active area of research, and motivated the possible alternative method to handling the copula that will be discussed soon. Since this is a probabilistic model, using the most probable sample as the output seems to be the most appropriate output criteria. However, in a well constructed model, the error to integrated values and probability density of the samples would be correlated to some degree. As a sanity check for the model, if it is sampled at an integrated force and moment value that was included in the data it was constructed with, then, given a large number of samples, the distributed load corresponding to that integrated force and moment value should be output. Following this logic, a natural model form error estimation method would be a leave-one-out style approach of generating the most probable model output at left out integrated values and measuring the distributed load error to the actual known distributed load.

Outside of stretching the dimensional limitations of gaussian kernel density estimation in the marginal conditioned distributions, the other aspect of this approach that may be an issue is using a multivariate normal copula. This is essentially imposing a linear correlation relationship structure to the copula as an approximation to the true relationships. Bayesian Sequential Partitioning (BSP) [43] is an approach that can be taken as an alternative to imposing structure in the copula. The basic motivation of BSP is to estimate the probability density of high dimensional data by essentially adaptively generating a non-equispaced histogram of the data. The histogram is algorithmically generated by a binary partitioning scheme, using sequential importance sampling [35], where the best partition is chosen by the partition score

$$\text{score}(m) = \log(\pi(m)) = -\beta j + \log \frac{B(n_1 + \alpha, \ldots, n_j + \alpha)}{B(\alpha, \ldots, \alpha)} - \sum_{p_s=1}^{j} n_{p_s} \log(|\nu_{p_s}|). \tag{3.13}$$

Where α and β are parameter constants, ν_{p_s} and n_{p_s} are the volume and number of data points in the partition subregion p_s. The sequential importance sampling attempts a certain number of trial cuts before choosing the best scoring cut, in the results shown in Chapter 4 the number of trial cuts was set to 100 in all cases. The score is based off minimizing the Kullback-Leibler divergence (KLD) between the actual and estimated probability density from the partition. It can be shown [44] that the log of the posterior distribution $\pi(m)$ of a sample partition is a linear function of the KLD between the actual and estimated densities, with a negative slope. So in order to minimize the KLD, the partition with the largest $\log(\pi(m))$ is chosen. The result from the bsp algorithm allows for a probability density estimate of some data sample x to be calculated as

$$p(x) \approx \frac{1}{N} \times \frac{n_j}{h_j}. \tag{3.14}$$

Where N is the total number of data samples, and n_j is the data count in bin volume h_j. Thus, the BSP is able to give an estimate of the sample which has the largest probability density. An equally important aspect, for the described method of generating distributed loads, is the ability to sample from the copula. It is a simple task to sample from the partition generated by BSP. Take a simple 1-dimensional histogram as a very simple case equivalent to the BSP partition. In order to sample from the histogram, one would randomly choose a histogram bin with the probability of drawing proportional to the fraction of known data within it. Then draw from a uniform distribution with lower and upper bounds matching that of the bounds of the drawn histogram bin. Effectively giving equal probability to draw any value within the bin bounds. This same process can be done to the high-dimensional BSP partition, however, the added dimensionality of the bins can cause accuracy issues. For example, take a highly correlated 2-dimensional data set. BSP can only take vertical or horizontal cuts on the 2-dimensional space and as such will likely have highly correlated data within each partition subregion. In perfectly correlated data this would amount to a diagonal line going through the partition subregion. Sampling from this subregion by assigning equal probability to any location within it effectively misses the strong correlation. This could be avoided by generating a huge number of subregions in the partition but for a practical case the computation will be stopped far before then.

In order to try and account for this issue, instead of assigning equal probability across the subregion, a sampling procedure similar to the approach taken in the Resampling method, described by Equation 3.3 in Section 3.2, can be used. The procedure will give high probability to possible values close to known values within the subregion and decreasing probability to possible values as distance from known values increases. The bandwidth used for this procedure was set to a fixed value of 0.1. In addition, a random draw is limited to the bounding values in each dimension of the partition it is contained in. For example, if the drawing procedure generates a two-dimensional array where the first dimension is larger than the first dimension bound in its associated partition, then the value in the first dimension of the array is lowered to the max bound of the partition. This addresses the issue that the draws are generated using a perturbation that has no knowledge of the domain it is supposed to be contained within and can easily generate draws that are outside this domain.

3.4 Methods Discussion

The results presented in Chapter 4 are separated into two types of distributed load data, line load or surface pressure data. The methods presented thus far can theoretically be applied to either of these two types of data. However, the Resampling method from Section 3.2 exhibited issues when applied to line load data that motivated its exclusion from the surface pressure results. In particular, the conditional distribution check step is not expected to be feasible with the higher dimensional restraints required by surface pressure data. In order to generate surface pressures, there would now be six integrated force and moment coefficients that must be considered simultaneously. The Resampling method essentially checks each of its generated distributed loads against several conditional distributions. The number of variables these distributions are conditioned on is the number of integrated force and moment coefficients that are to be matched. Thus these distributions are conditioned on six variables for surface pressures data. The issue is the vanishing probability that the generated surface pressures will pass these conditioned distribution checks. The Resampling method performs well enough using line load data as there is only one or two variables that need to be conditioned simultaneously. Even in the case of line load data, the number of failed conditional distribution checks increased from 10 to 1000 when increasing from one to two conditioned variables. This quickly growing number of failures when just increasing the constraints from one to two motivated the development of the copula-based methods presented in Section 3.3. As mentioned prior, it also disqualified the Resampling method from being applied to surface pressure data.

In addition to resolving these surface pressure issues, the copula-based methods offer another distinct advantage over the Resampling method. The copula methods have the built-in capability of easily approximating the probability density of the distributed loads they generate. This capability allows one to be able to approximate which of the generated distributed loads is most likely for the given integrated coefficient targets. When generating distributed loads at a known integrated value, it is a reasonable conclusion that the most probable should be the same as the corresponding known distributed load. In other words, the methods should generate the known distributed load for known integrated coefficients. In order to enforce this an external approach would be required in order to approximate the probability densities of the Resampling method's output distributed loads. Considering the two discussed factors, the copula-based methods appear to be the stronger candidate methods, however the Resampling method is still included as an exploration of alternate approaches.

The main contribution that this work provides is a method for generating distributed loads that correspond to integrated loads under uncertainty. There exists methods that attempt to do this, as discussed in Section 3.1, however these do not accommodate the spatial relationships that exist in distributed load data. The methods provided in this chapter utilize the statistical distributions of known distributed loads in order to ground the generated distributed loads, with the goal being to preserve the spatial relationships that exist in the known data. The Maximum Mean Discrepancy two-sample test is used in order to quantify the distance between the statistical distributions of the generated and known distributed loads.

Chapter 4

Results

4.1 Data Set

Even though this work is motivated by and follows the development procedures of the SLS ascent aerodynamic databases, there is no SLS data included in this work. This is because most of the data produced by the SLS program falls under International Traffic and Arms Regulations which greatly restricts its publication. Instead, the data used in this work was taken from CFD simulation results on a generic ascent vehicle geometry. The launch vehicle geometry selected for study was a hammerhead rocket design [49] with several simple protuberances added. Figure 4.1 contains an image of the geometry used in the CFD simulations that generated the data set used in this work.

This data set used the vehicle angle of attack (α) and angle of sideslip (β) as independent variables, which were varied over the 29 CFD simulations that were ran. The values (α, β) were set by fourth level nested Clenshaw-Curtis quadrature points [1] shown in Figure 4.2. The Mach numbers for all the CFD simulations used in this data set were fixed at M=1.8.

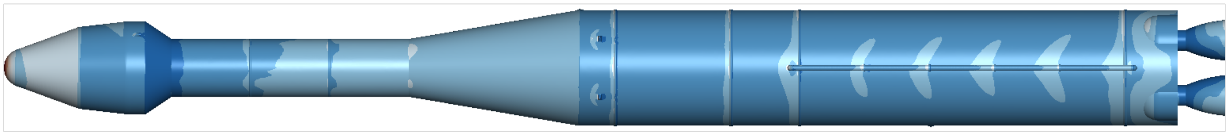


Figure 4.1: Hammerhead geometry used to generate the data set.

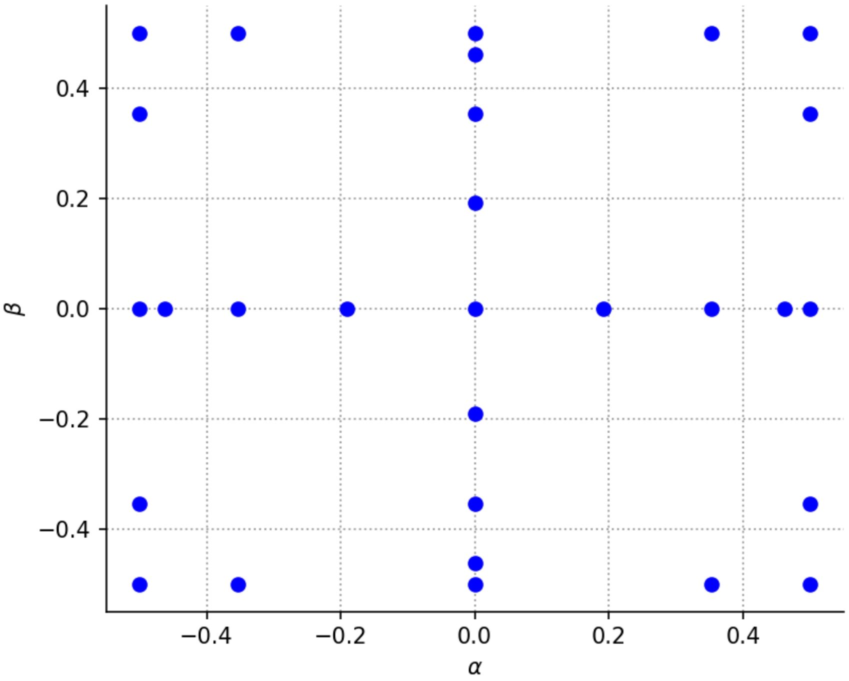


Figure 4.2: α & β values used in the data set.

The FUN3D 13.7 code [3] from NASA Langley Research Center was used to compute the CFD solutions. The simulations were run steady state with compressible flow. They used the Spalart-Allmaras turbulence model and Harten-Lax-van Leer-Contact flux construction with the stencil-based van Albada flux limiter augmented with a heuristic pressure limiter. A triangular unstructured mesh was used with volume cells generated via Simsys' unstructured grid generator software AFLR3. Once the CFD was run, the integrated forces and moments as well as the line loads were extracted from each case using the "triload" tool [52] from Chimera Grid Tools [9]. Version 1.0 of the NASA CAPE [11] software was used in both executing the required CFD simulations and in post-processing the resulting data. Response surfaces were built using the distributed load data from the 29 CFD solutions. Each line load is a 101×1 column vector, corresponding to the line load taken from a single CFD solution, and, similarly, each CFD solution surface pressure is a 402909×1 column vector. The line load response surface was built directly from the line load data and likewise for the surface pressures. These response surfaces were built using Gaussian Radial Basis functions with bandwidths equal to the average distance between the α and β locations shown in Figure 4.2. The response surfaces were sampled at N_s uniformly distributed points within the CFD bounds for α and β, i.e. points drawn from $\mathcal{U}(-0.5, 0.5)$ for both variables. The N_s=5000 sampled line loads were taken as the full line load data set and the N_s=1000 sampled surface pressures were taken as the full surface pressure data set for the results presented in this chapter.

4.2 Line Loads

Validation

The Resampling method from Section 3.2 and the copula-based methods from Section 3.3 are first applied to line load data. These initial results are to validate these methods using the test outlined in Section 2.5. The validation test was performed with a simultaneous draw of 300 normal force and pitching moment coefficients from known data cases. First the results for the Resampling method are shown. A value of $p_{\min}$=0.01 was used for these results. This controls the minimum value of probability density the distribution tests described in Section 3.2 need to output in order to consider a generated line load as acceptable. These results used the all 5000 samples line loads from the data set as the known data.

Figure 4.3 shows the generated line loads in comparison to the known data for both normal force and pitching moment coefficient. Similar figures containing the other aerodynamic coefficients are included in Figures A.1 and A.2 in Appendix A. The blue "Data" is the all the line loads from the known data and the red "Disps." is the generated line loads from the Resampling method. Not much can be determined from this figure except for it appears that the generated line loads tend to fill the space of possible values covered by the known data. Another qualitative observation is that there doesn't appear to be any sign of the dispersions exceeding the boundaries the known data.

Figure 4.4 shows $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ histograms for the 16th, 34th, and 73rd line load slice. Similar figures containing the other aerodynamic coefficients are included in Figures A.3 and A.4 in Appendix A. These slices were chosen because previous unsuccessful methods typically failed to generate line loads that agreed with the known data at these locations. Once again,

Coefficient	Critical Value	Score
C_A	0.005451	0.005593
C_ℓ	0.005978	0.007325
C_N	0.007042	0.006229
C_m	0.006937	0.005394
C_Y	0.007196	0.004720
C_n	0.006392	0.005123

Table 4.1: MMD 2-sample test results on Resampling method generated line loads.

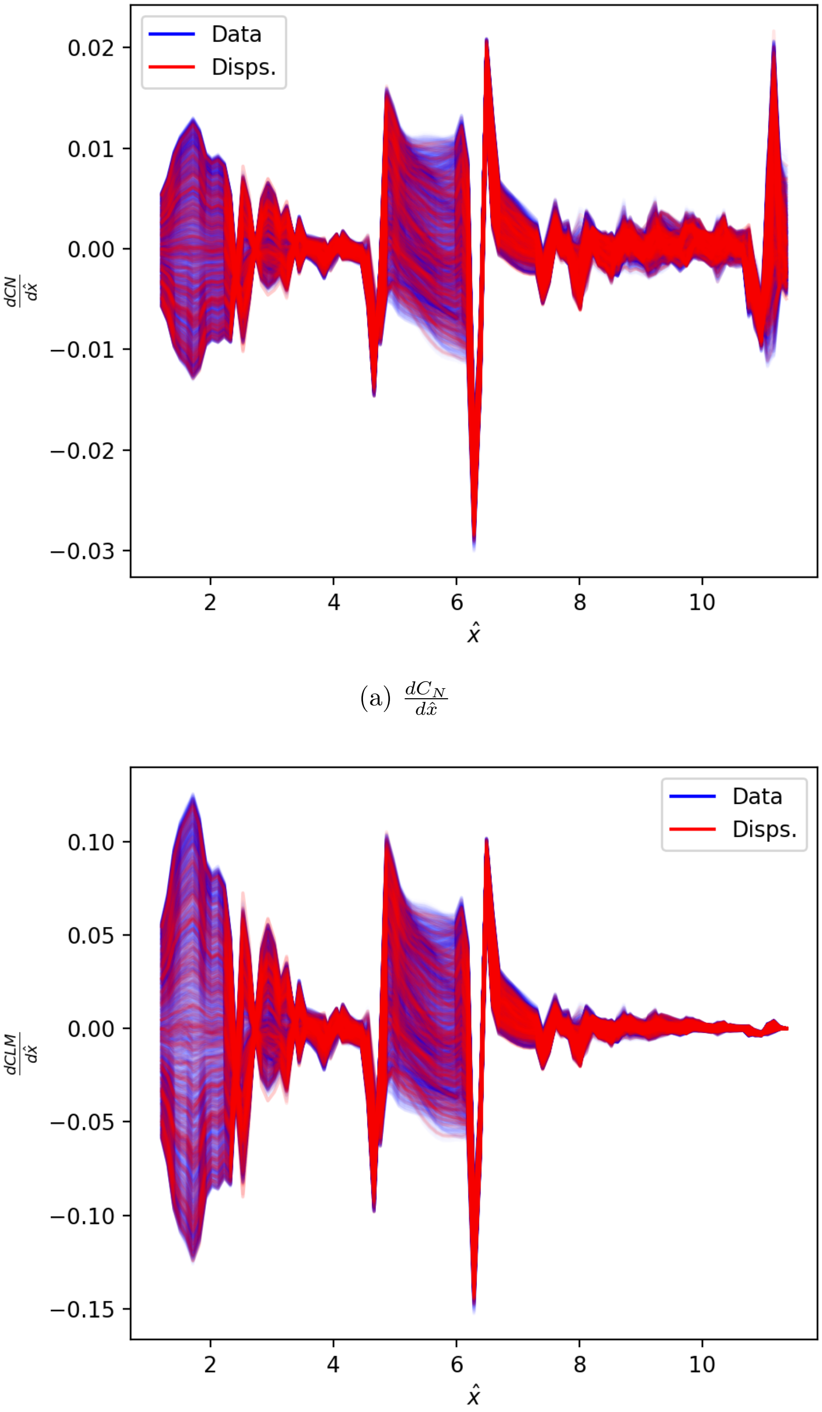
Coefficient	Critical Value	Score
C_A	0.0055192	0.005593
C_ℓ	0.006110	0.007325
C_N	0.006362	0.006229
C_Y	0.006545	0.004720

Table 4.2: MMD 2-sample test results on Resampling method generated line load's $\mathbf{\Omega}$ coefficients.

the blue "Data" is the histogram of the known line load data at the given slice and the red "Disps." are the line loads generated by the Resampling method. From a qualitative analysis the generated line loads do a decent job of covering the known data histogram, and univariate two-sample tests could provide some quantitative measure of this. However, this is more of a qualitative look at the performance of the methods and the multivariate MMD two-sample test is used to assess performance quantitatively.

Table 4.1 shows the MMD two-sample test scores for both the force and moment line loads. Table 4.2 shows the MMD two-sample test scores for the coefficients $\mathbf{\Omega}$. As most are below the critical value, there is a statistically insignificant difference between the generated line loads and known line loads distributions for these aerodynamic coefficient line loads. However, the validation test failed for the axial force and the rolling moment coefficients.

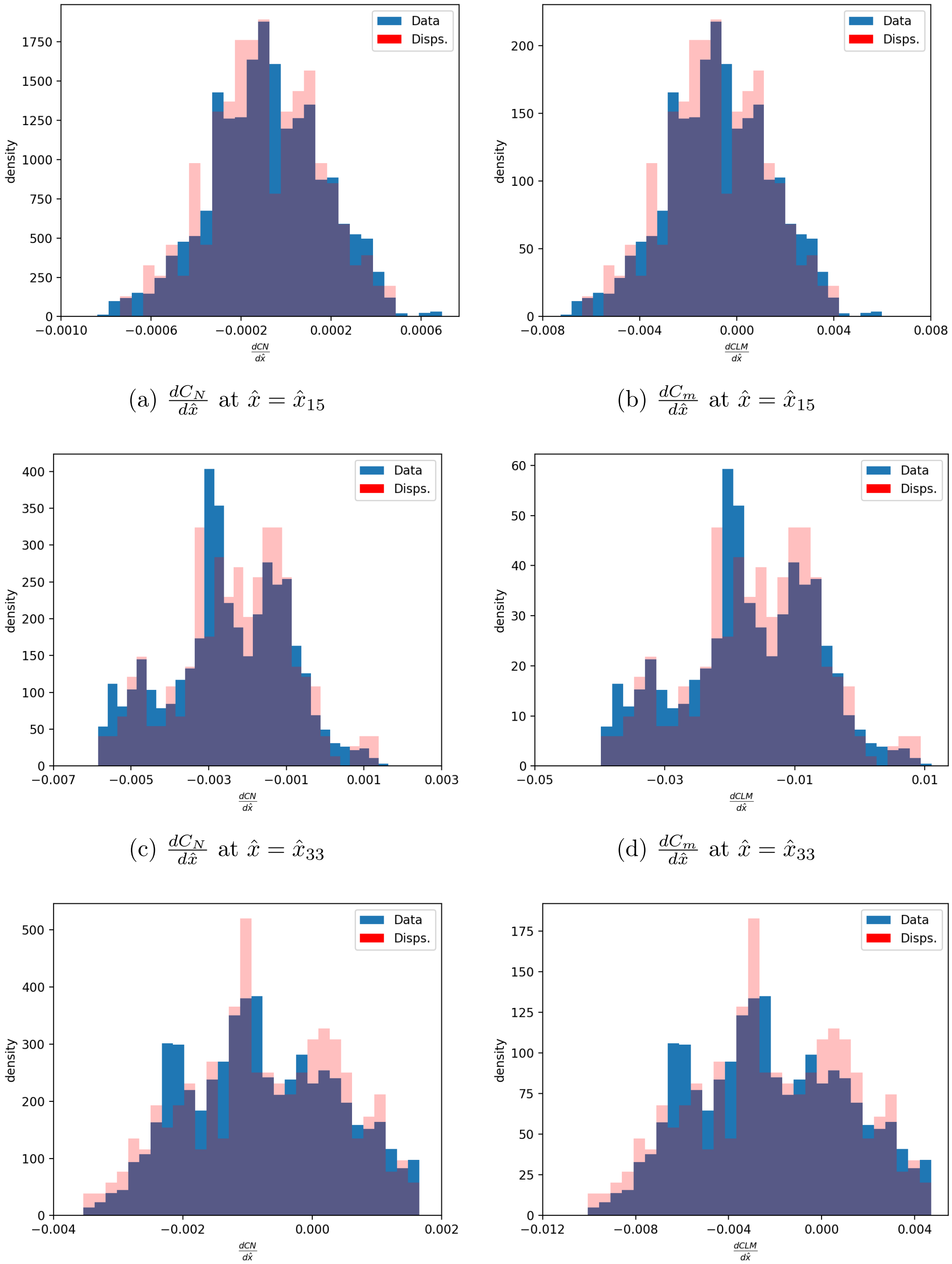
Next, the same validation test was performed for the copula-based methods from Section 3.3. The initial results are for the MVN-Copula method, the method that models the copula with a multivariate normal distribution. These results compare the 100 generated line loads from this method against the 29 known CFD data in addition to 100 interpolated data drawn



(a) $\frac{dC_N}{d\hat{x}}$

(b) $\frac{dC_m}{d\hat{x}}$

Figure 4.3: Re-sample generated $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ for 300 C_N & C_m taken from dataset.



(a) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(c) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.4: Re-sample $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ histograms for 300 C_N & C_m taken from dataset.

Coefficient	Critical Value	Score
C_A	0.027928	0.0190488
C_ℓ	0.0300154	0.011976
C_N	0.032318	0.0119036
C_m	0.0337099	0.0125241
C_Y	0.0337521	0.0146955
C_n	0.0335387	0.0145765

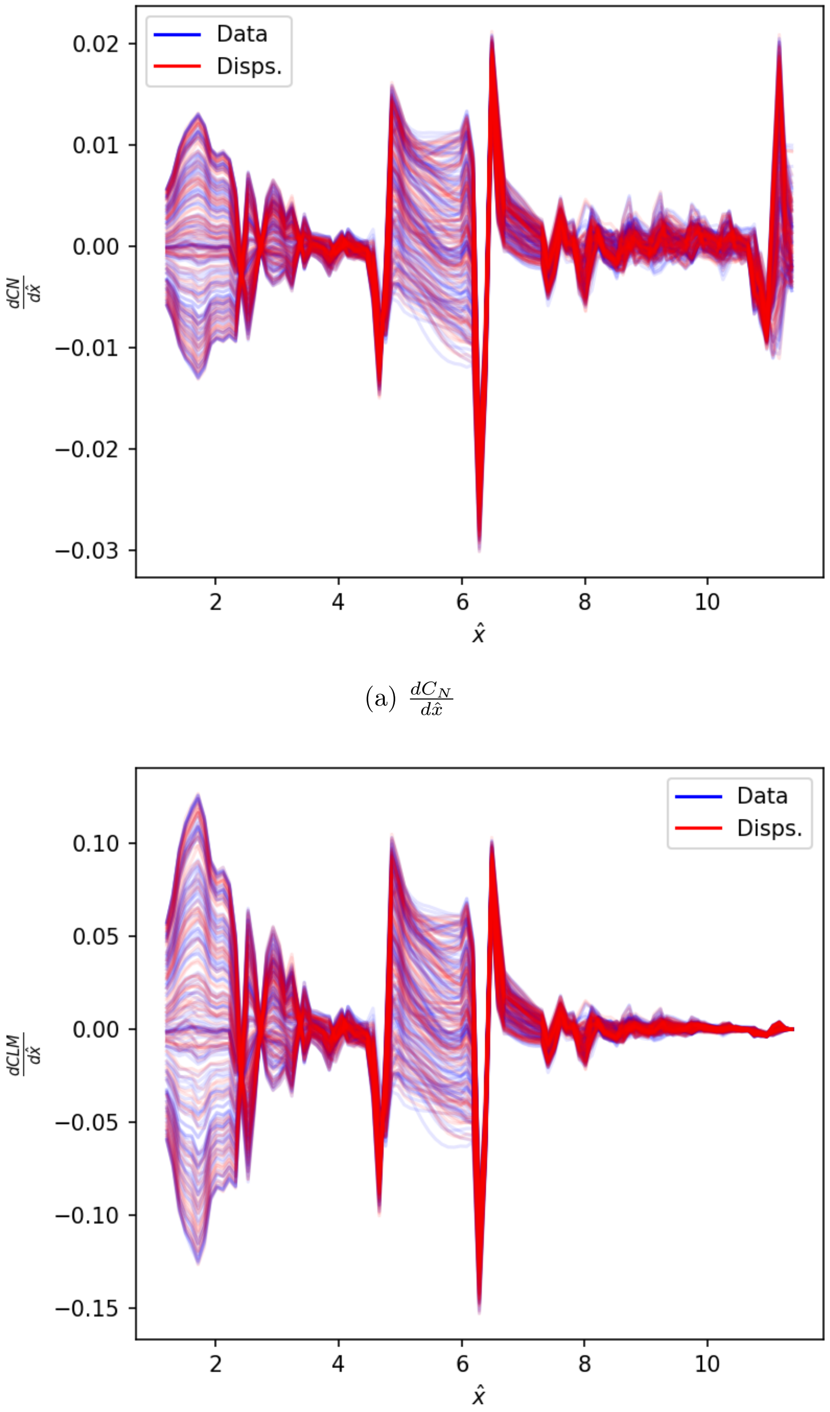
Table 4.3: MMD 2-sample test results on MVN-Copula generated line loads.

randomly from the 5000 line loads in the data set. This was done so to only compare the generated line loads with the data actually used in the process of constructing the method. In addition, in order to avoid extrapolation, the drawn integrated coefficients to be used as target values were ensured to be within the bounds of the data actually used to construct the method.

The generated line loads for normal force and pitching moment coefficients are shown in Figure 4.5. Similar figures containing the other aerodynamic coefficients are included in Figures A.7 and A.8 in Appendix A. The data used in the construction of the method is shown in blue and the dispersions are red. Ideally, the red and blue lines should be spread over a similar area of values and qualitatively that appears to be the case.

Histograms of slices from the generated normal force and pitching moment line loads are shown in Figure 4.6. Similar figures containing the other aerodynamic coefficients are included in Figures A.9 and A.10 in Appendix A. The histograms are again another qualitative comparison, but an important check that the generated data and the known data are aligned when looking at a single line load slice at a time. For example, the histogram of the slice at $\hat{x} = \hat{x}_{15}$ shows a decent matching of the minimum and maximum bounds as well as the location of the modes. An important observation from these histograms is that the slices are not normally distributed and any hypothetical method that assumes them to be would be ignoring the true shape of these distributions from the known data. The results in Table 4.3 and 4.4 show that the method passes the validation test in all six of the aerodynamic coefficients.

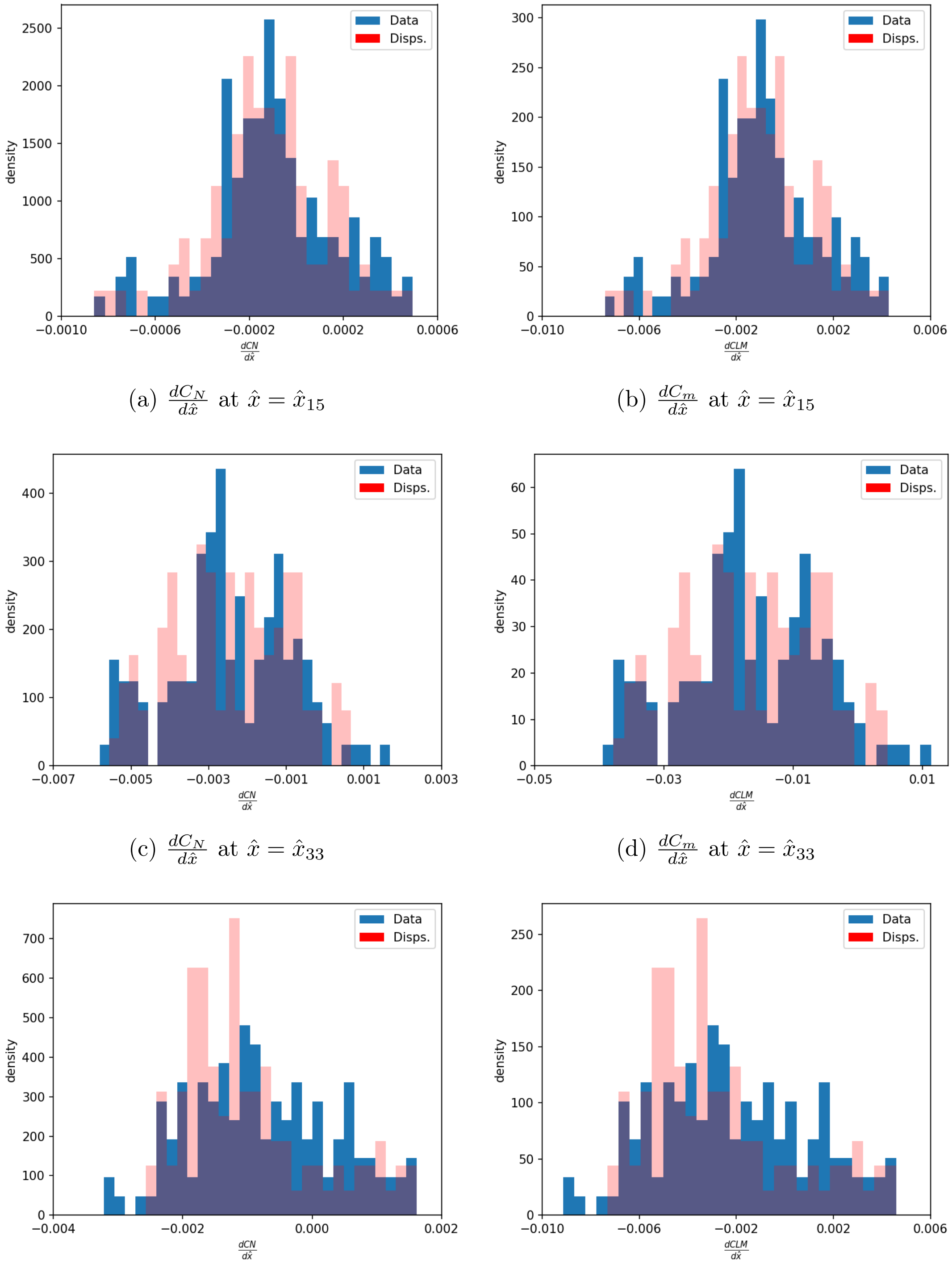
Finally, the next results are using the BSP-Copula method from Section 3.3. This method



(a) $\frac{dC_N}{d\hat{x}}$

(b) $\frac{dC_m}{d\hat{x}}$

Figure 4.5: MVN-Copula generated $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ for 100 C_N & C_m taken from dataset.



(a) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(c) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.6: MVN-Copula $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ histograms for 100 C_N & C_m taken from dataset.

Coefficient	Critical Value	Score
C_A	0.0303446	0.0190488
C_ℓ	0.0299224	0.0119766
C_N	0.0322981	0.0119036
C_Y	0.033669	0.0146955

Table 4.4: MMD 2-sample test results on MVN-Copula generated line load's $\boldsymbol{\Omega}$ coefficients.

Coefficient	Critical Value	Score
C_A	0.029208	0.029135
C_ℓ	0.028897	0.026631
C_N	0.032769	0.016083
C_m	0.034305	0.0162716
C_Y	0.033396	0.0130797
C_n	0.0323513	0.0125448

Table 4.5: MMD 2-sample test results on BSP-Copula generated line loads.

models the copula using a Bayesian Sequential Partitioning approximation to the copula distribution. Similar to the MVN-Copula method results given prior, these results compare the 100 generated line loads with this method against the 29 known CFD data in addition to 100 interpolated data drawn randomly from the 5000 calculated by radial basis functions. The results included here used partitions with 2000 cuts for each coefficient. Similar to the previous methods' results, the generated dispersions are compared against the data used to construct the method in Figures A.13 - A.15. Histograms of select slices from the generated line loads are shown in Figures A.16 - A.18. Both sets of figures are included in Appendix A for completeness, however the conclusions drawn from the qualitative analysis of them do not differ significantly from the corresponding figures shown for the previous two methods.

The results in Table 4.5 and 4.6 follow a similar trend as in the MVN-Copula method. The BSP-Copula method in particular was close to not generating a passable axial force line load score, but it still does pass. Further tweaking of the sampling bandwidth may allow for optimizing the validation test results and is left as an avenue for future research work.

Coefficient	Critical Value	Score
C_A	0.0294168	0.029135
C_ℓ	0.029727	0.026631
C_N	0.034902	0.016083
C_Y	0.0311131	0.013079

Table 4.6: MMD 2-sample test results on BSP-Copula generated line load's $\mathbf{\Omega}$ coefficients.

Discussion

The Resampling method did not pass the validation test for the axial force and rolling moment coefficients, but did pass for the other four coefficients. The MVN-Copula methods passed validation for all aerodynamic coefficients, while the BSP-Copula method passed all coefficients except it nearly failed validation for the axial force coefficient line load. Therefore, based just on the results of the validation test, the MVN-Copula method is the strongest performing candidate. Although the BSP-Copula method is close behind with potential to improve the axial coefficient validation test with some small tweaks. A key take-away from the validation test is that testing the coefficient or testing the full dimensional line loads always gave the same result. This gives credence to using the distribution of the singular value coefficients, $\mathbf{\Omega}$, as a low-dimensional proxy to the full line load distribution. It also justifies the results presented in the next section, containing the surface pressure data, as only the coefficients could be tested since testing the full surface pressure distribution is not feasible due to the data having 402909 dimensions.

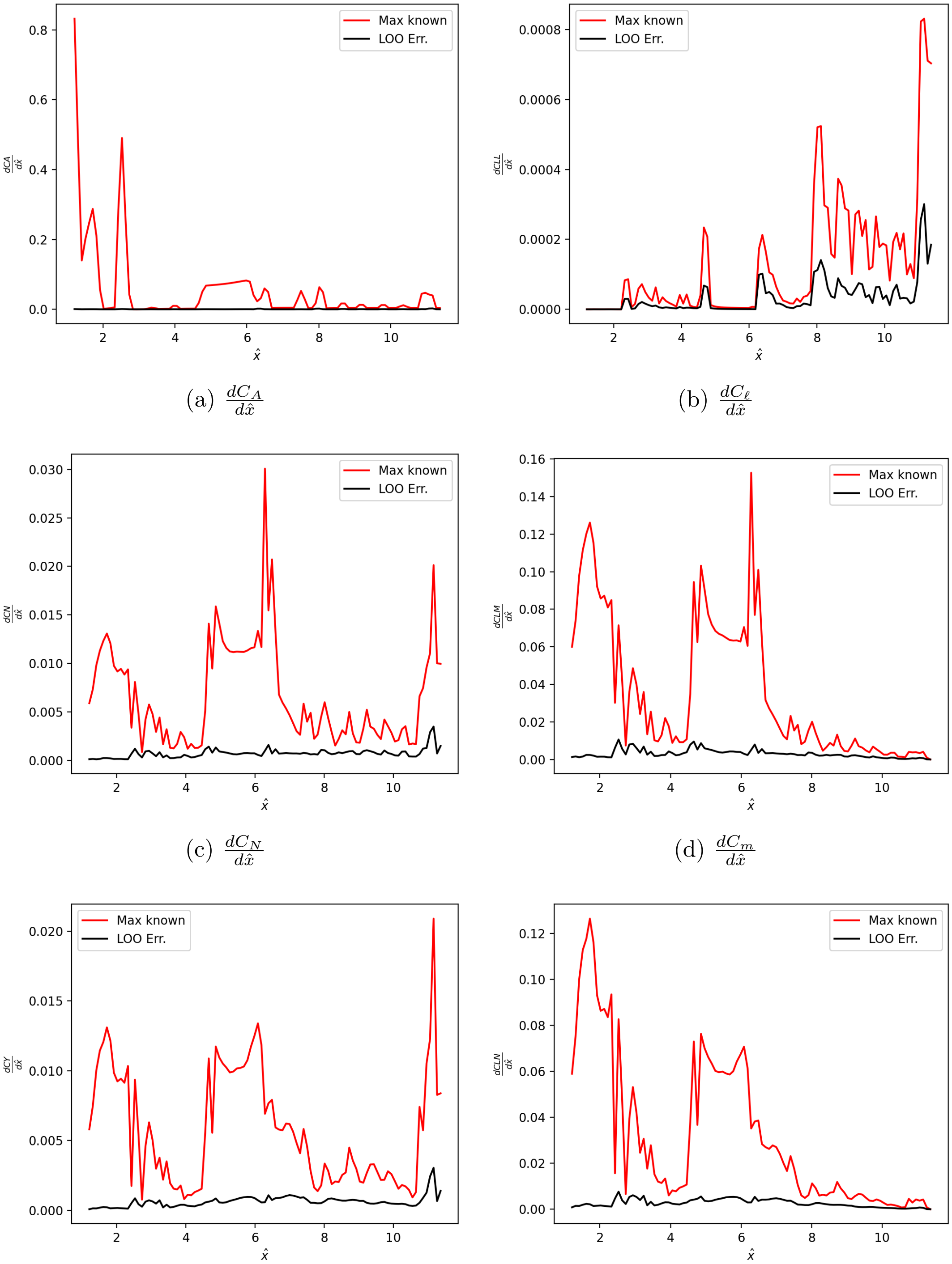
Leave-One-Out Error Estimation

The Resampling method had difficulty generating a valid line load for some of the left out cases. For each left out case, 100000 attempts were made to generate a valid line load. If a valid line load could not be made in that many attempts, the case was skipped and not included in the error estimates. One case failed to generated in the C_N LOO data, with $\alpha = -0.5$ and β=0.5. While one case also case failed in the C_Y LOO data, with α=0.5 and β=0.5. These are both located at the corners of the data set parameter space, which may be factor in them failing. Since these methods are based off the known data and are not intended to extrapolate, it is not unexpected that predicting the line loads at the edges of the data space would be challenging. The Resampling method in particular is relatively strictly contained to the known data bounds due to the conditional distribution checks to determine the validity of a generated line load.

Figure 4.7 contains the LOO error results for each aerodynamic coefficient. In order to get a relative sense of scale, the LOO error is plotted along with the maximum absolute value of the coefficient at each axial slice. The $\frac{dC_\ell}{d\hat{x}}$ appears to have the largest LOO error across the axial locations, while $\frac{dC_A}{d\hat{x}}$ is consistently low. No issues stand out immediately from looking at this figure alone and the results are similar to the MVN-Copula results presented next.

For the MVN-Copula and BSP-Copula methods, the previously calculated bandwidths using the full 29 dimensional coefficients was used in the GKDEs even though there is one fewer coefficient due to the LOO process. A quick examination of optimizing the 28 bandwidths in one leave-one-out case showed only small changes in the bandwidths associated with the small magnitude singular value bases. Since the high singular value coefficients showed very little change, it was deemed an acceptable simplification to use the 29 bandwidths optimized with all the data and just use them with the final value truncated. The LOO errors for the MVN-Copula method, shown in Figure 4.8, seem to be small relative to the max seen values at each slices. The exception being rolling moment, shown in Figure 4.8b, which was the same result seen in the Resampling method data. This can perhaps be explained as the known data having very little rolling moment in any solution and the data then being primarily driven by noise in the CFD solution.

The cost of performing a full leave-one-out error estimation using the BSP-Copula method is quite computationally expensive. This is due to a new partition needing to be calculated for each $\kappa-1$ dimensional coefficient data. In order to reduce the cost, partitions were generated using a decreasing number of partition cuts. These partitions were then used to generate 100 $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ line loads using the BSP-Copula method. The MMD test was run on the output line loads for each, the results of which are included in Table 4.7 and 4.8. These results seem to suggest that the MMD test can be passed with a relatively few number of cuts. This could be because the method used to sample from the partition is relatively robust and that



(a) $\frac{dC_A}{d\hat{x}}$

(b) $\frac{dC_\ell}{d\hat{x}}$

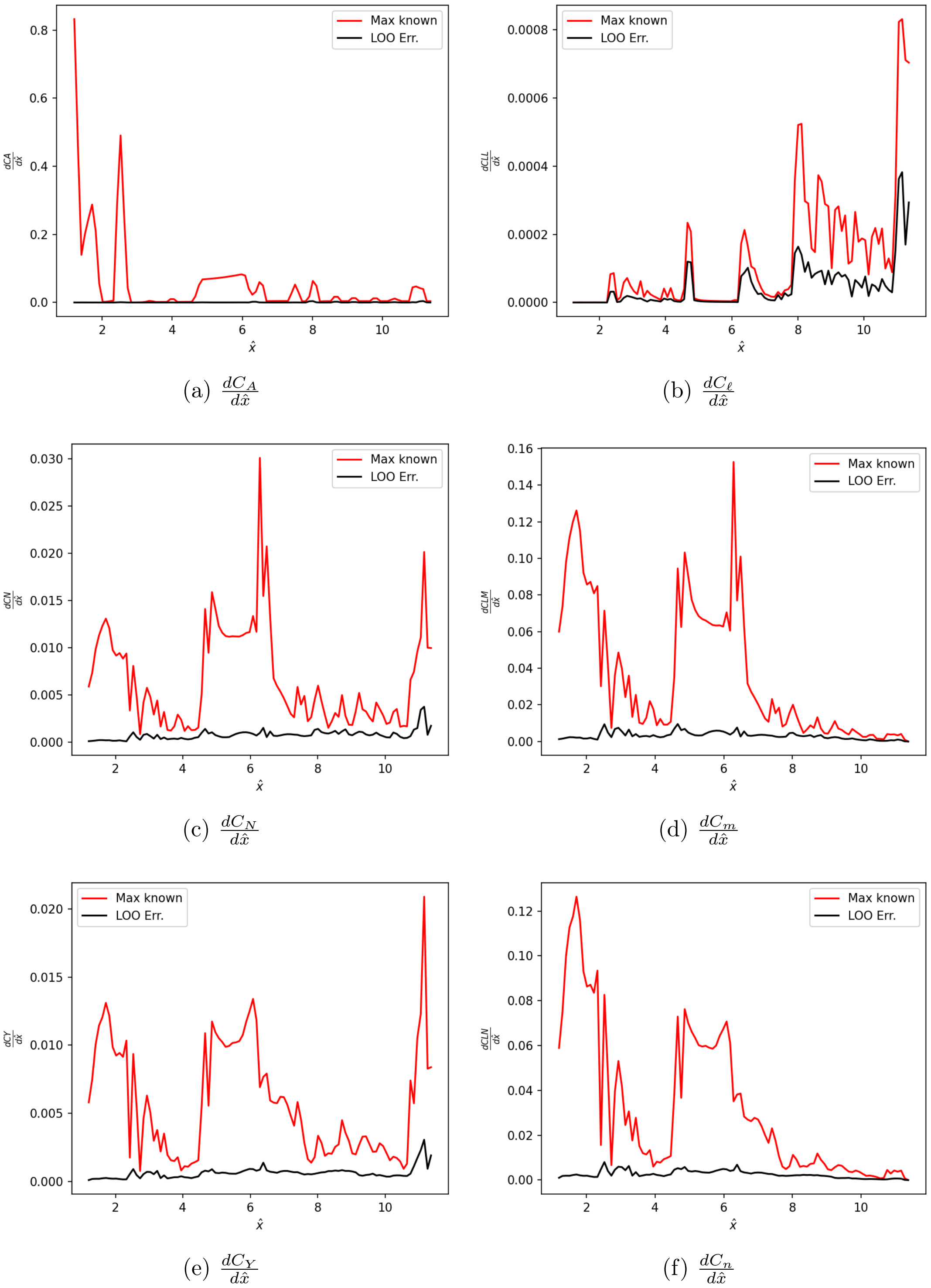
(c) $\frac{dC_N}{d\hat{x}}$

(d) $\frac{dC_m}{d\hat{x}}$

(e) $\frac{dC_Y}{d\hat{x}}$

(f) $\frac{dC_n}{d\hat{x}}$

Figure 4.7: Resampling method leave-one-out errors for 29 known data points, 100k samples



(a) $\frac{dC_A}{d\hat{x}}$

(b) $\frac{dC_\ell}{d\hat{x}}$

(c) $\frac{dC_N}{d\hat{x}}$

(d) $\frac{dC_m}{d\hat{x}}$

(e) $\frac{dC_Y}{d\hat{x}}$

(f) $\frac{dC_n}{d\hat{x}}$

Figure 4.8: MVN-Copula leave-one-out errors for 29 known data points, 100k copula draws

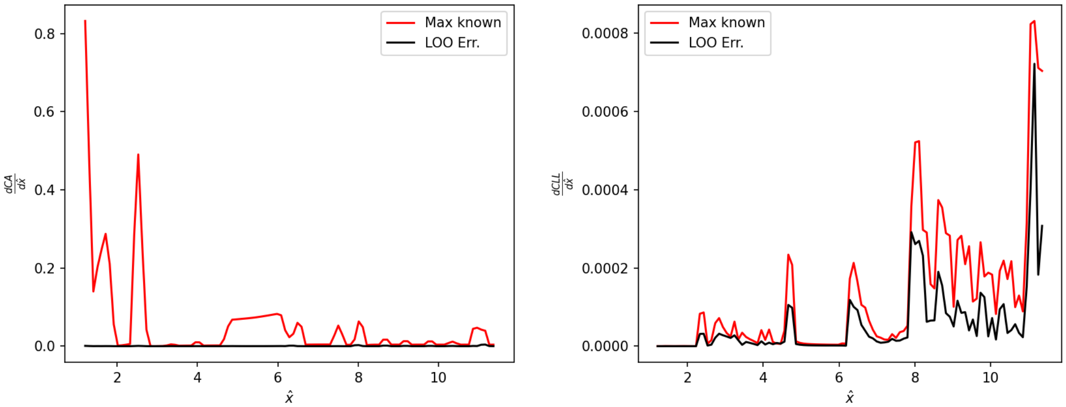
n_{cuts}	500	1000	2000
Critical Value	0.034102	0.033241	0.029808
Score	0.018199	0.021574	0.016095

Table 4.7: Partition cut number sensitivity MMD 2-sample test results on BSP-Copula generated $\mathbf{\Omega}$ coefficients

n_{cuts}	500	1000	2000
C_N Critical Value	0.033083	0.033866	0.0323507
C_N Score	0.018199	0.021574	0.016095
C_m Critical Value	0.035027	0.036331	0.034111
C_m Score	0.016647	0.017793	0.016189

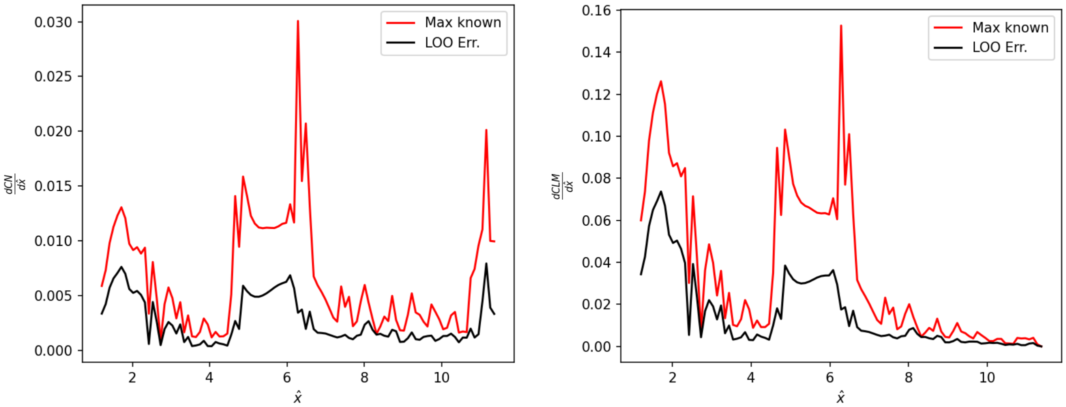
Table 4.8: Partition cut number sensitivity MMD 2-sample test results on BSP-Copula generated force and moment line loads.

the largest contribution the partition makes to the overall process, the estimation of the conditioned joint probability density, is largely outweighed in the total probability density estimation, Equation 3.10, by the conditioned marginal probability densities. The leave-one-out error estimates used partitions with 500 cuts, a quarter of the number in the results shown in the validation section, and the results are shown in Figure 4.9 - 4.11. The error estimates are quite a bit larger than those from the other methods, indicating poor predictive performance. One interesting observation is that the errors also appear to be well correlated with the maximum values for every coefficient, whereas the Resampling and MVN-Copula methods only exhibited that behavior for the rolling moment coefficient results. There is potential for this to be improved however, with an initial idea being to increase the number of cuts in the partition. Even though it can pass the validation with a lower number of partition cuts doesn't mean that it's ability to predict a line load in a leave-one-out fashion didn't decrease. Another potential idea is to improve the selection process of the bandwidth used in the sampling procedure, this is discussed further in Chapter 5. The BSP-Copula results used significantly fewer draws in each attempt to generate a potential line load, 10 times fewer than the MVN-Copula method, this is due to the MVN-Copula code being much better optimized for operating with large amounts of data.



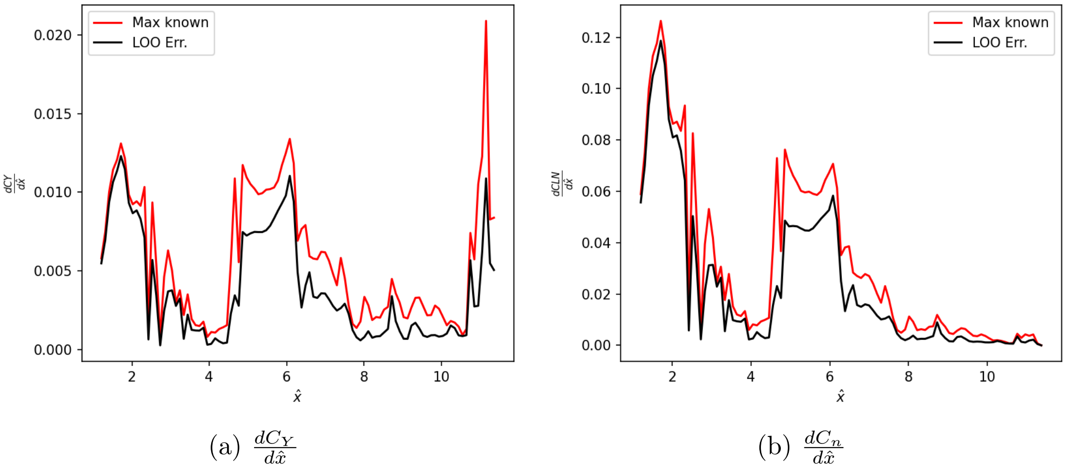
(a) $\frac{dC_A}{d\hat{x}}$ (b) $\frac{dC_\ell}{d\hat{x}}$

Figure 4.9: BSP-Copula leave-one-out errors for 29 known data points, 10k copula draws



(a) $\frac{dC_N}{d\hat{x}}$ (b) $\frac{dC_m}{d\hat{x}}$

Figure 4.10: BSP-Copula leave-one-out errors for 29 known data points, 10k copula draws



(a) $\frac{dC_Y}{d\hat{x}}$

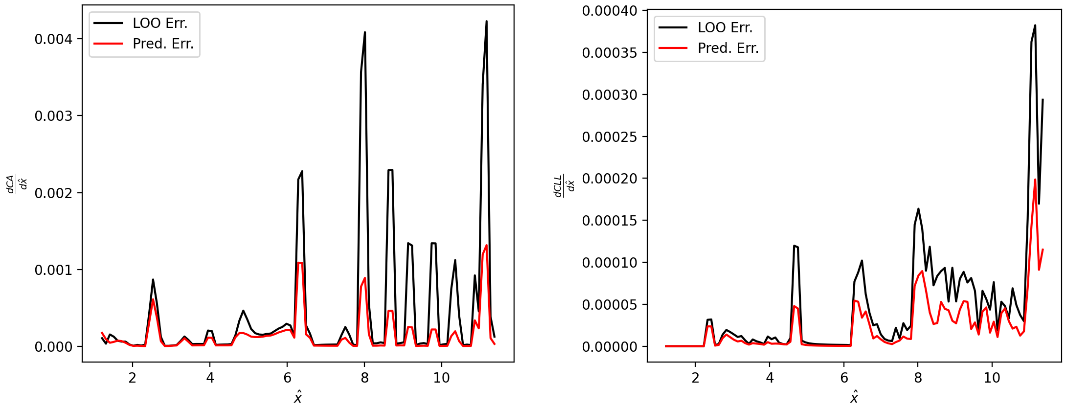
(b) $\frac{dC_n}{d\hat{x}}$

Figure 4.11: BSP-Copula leave-one-out errors for 29 known data points, 10k copula draws

An additional test of note related to the LOO error is the method of bandwidth selection for the copula based methods' conditioned marginal distributions. The underlying motivation in the bandwidth selection procedure is driven by balancing the predictive performance of novel integrated values and returning the known distributed load at known integrated values. The leave-one-out test addresses the first aspect, but the performance of the second aspect was investigated using the MVN-Copula method. The test calculated the average error that the line load generated from the MVN-Copula method had relative to the known line load for the 29 known CFD integrated coefficients. The predictive error results for this test are shown in Figures 4.12 - 4.14, with the leave-one-out error also shown for comparison. The predictive errors follow generally similar trends with the leave-one-out error across the vehicle, with the predictive error almost always being smaller than the leave-one-out error. This indicates that the bandwidth selection process is working as intended and the bandwidths are not over-fitted to the known data.

Discussion

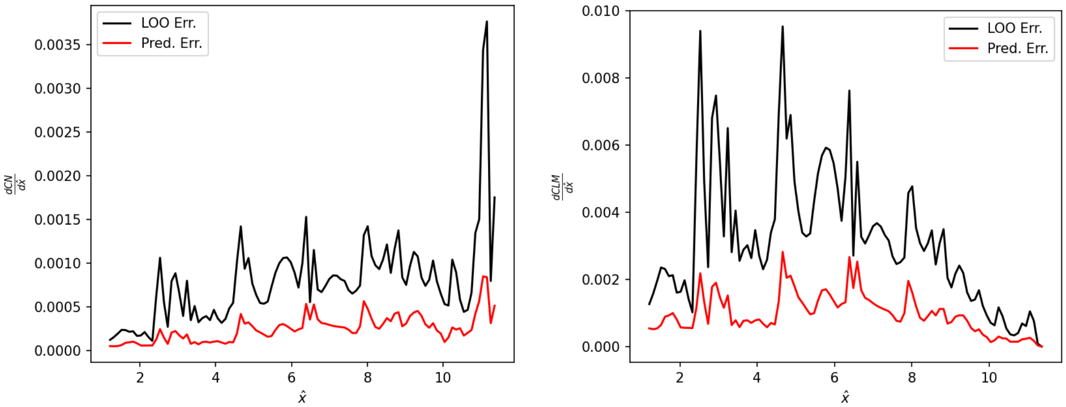
The LOO error for the Resampling and MVN-Copula method are quite similar, and much lower than the BSP-Copula method error. The rolling moment error sticks out as having the largest relative error in all methods but particularly in the Resampling and MVN-Copula



(a) $\frac{dC_A}{d\hat{x}}$

(b) $\frac{dC_\ell}{d\hat{x}}$

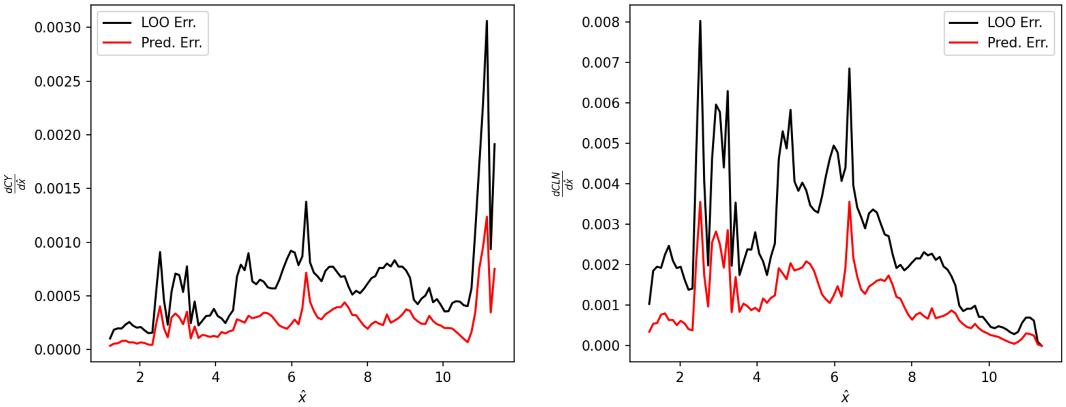
Figure 4.12: MVN-Copula predictive errors for 29 known data points, 100k copula draws



(a) $\frac{dC_N}{d\hat{x}}$

(b) $\frac{dC_m}{d\hat{x}}$

Figure 4.13: MVN-Copula predictive errors for 29 known data points, 100k copula draws



(a) $\frac{dC_Y}{d\hat{x}}$ (b) $\frac{dC_n}{d\hat{x}}$

Figure 4.14: MVN-Copula predictive errors for 29 known data points, 100k copula draws

results when compared with the other coefficients. This is potentially due to the fact that the rolling moments, over all the simulated cases, had small magnitudes and correspondingly small variance, at least partially due to the geometry used being quite symmetric. Therefore, the differences in rolling moment results between cases are suspected to be primarily driven by various sources of noise in the CFD solutions. Thus, the methods could be having trouble generating line loads that are driven by stochastic noise and not related to any effect driven by flow physics. The axial force coefficient appears to have low relative error even though it doesn't pass the validation test for the Resampling and BSP-Copula methods. That means that these methods can predict with a low overall error but that the generated line loads don't follow the distribution of the known data. A potential avenue for future work could be to investigate this failure of the Resampling method, and near failure of the BSP-Copula method, to follow the axial force coefficient distribution and potentially adjust the method in a manner that depends on the coefficient that is being generated.

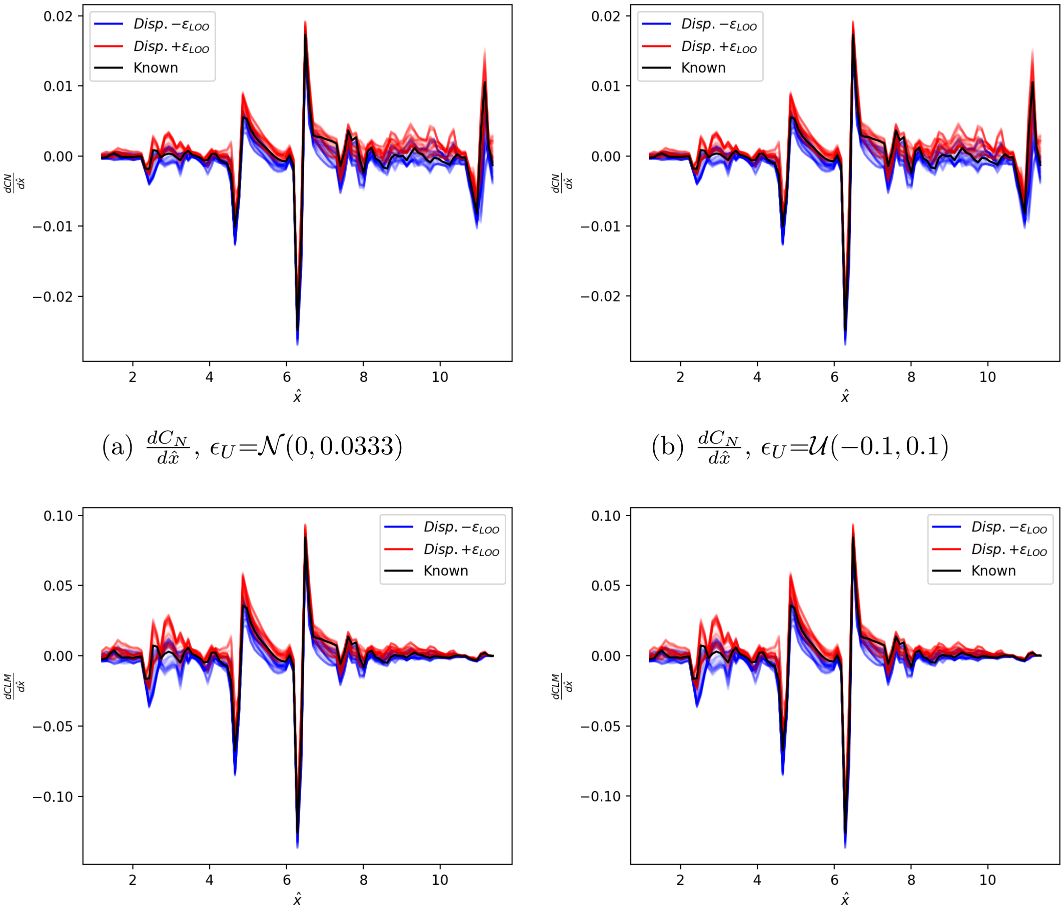
Another result of note presented in the results of this section is that the BSP-Copula validation test results varied only slightly with a decreasing number of partition cuts. This, along with the success of the MVN-Copula method, suggests that the distribution of the generated line loads are not particularly sensitive to the quality of the estimated joint distribution modelled in the copula methods. However, restricting the number of partition cuts would seemingly decrease the accuracy of the probability density estimation, which in turn increases the estimated leave-one-out error. This may explain the comparatively large leave-one-out error that the BSP-Copula method exhibited, however arbitrarily increasing the number of partition cuts used in the leave-one-out test becomes incredibly computationally expensive. This is primarily due to the cost of calculating a brand new BSP partition for each of the left out cases.

The bandwidth selection procedure for the copula based methods' conditioned marginal distributions is validated by looking at the predictive performance of the MVN-Copula method. These results did not indicate the bandwidths were over-fitted to the known data.

Simulated Uncertainty

The simulated uncertainty test is performed using either a normal or uniform distribution in order to model the aleatoric uncertainty in the integrated aerodynamic coefficients. Each method generates uncertain line loads according to the provided aleatoric uncertainty in the integrated coefficients. The aleatoric uncertainties were simulated according to $\tilde{C}^i_{z,j} = C^i_z(1 + \epsilon_{U,j})$, where C^i_z is the integrated coefficient z from the $i-$th known data case and $\epsilon_{U,j}$ is a value drawn from the aleatoric uncertainty model. The same $\epsilon_{U,j}$ was used for related forces and moments, for example one $\epsilon_{U,j}$ was drawn for normal force and then used to generate $\tilde{C}^i_{N,j}$ and $\tilde{C}^i_{LM,j}$. The $\epsilon_{U,j}$ are drawn from two different distributions to model the uncertainty, ϵ_U=$\mathcal{N}(0, 0.0333)$ and ϵ_U=$\mathcal{U}(-0.1, 0.1)$. This is done to compare the difference in using two commonly assumed aleatoric uncertainty models. The LOO error estimates from the previous sections are added to these generated line loads and a probability box is created for each line load slice.

The simulated uncertainty results for the MVN-Copula method is shown in Figure 4.19, with the figures containing the remaining coefficients included in Figures 4.20 and 4.22 in Appendix A. These results attempt to show an upper and lower bound on the line load uncertainty. Figures 4.20 - 4.22 show the probability boxes at several slices in order better compare the differences between the aleatoric uncertainty model. The comparison between the two varies pretty significantly depending on the coefficient of interest and the location of the line load slice. For example there is a relatively large difference in the probability box



(a) $\frac{dC_N}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(b) $\frac{dC_N}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

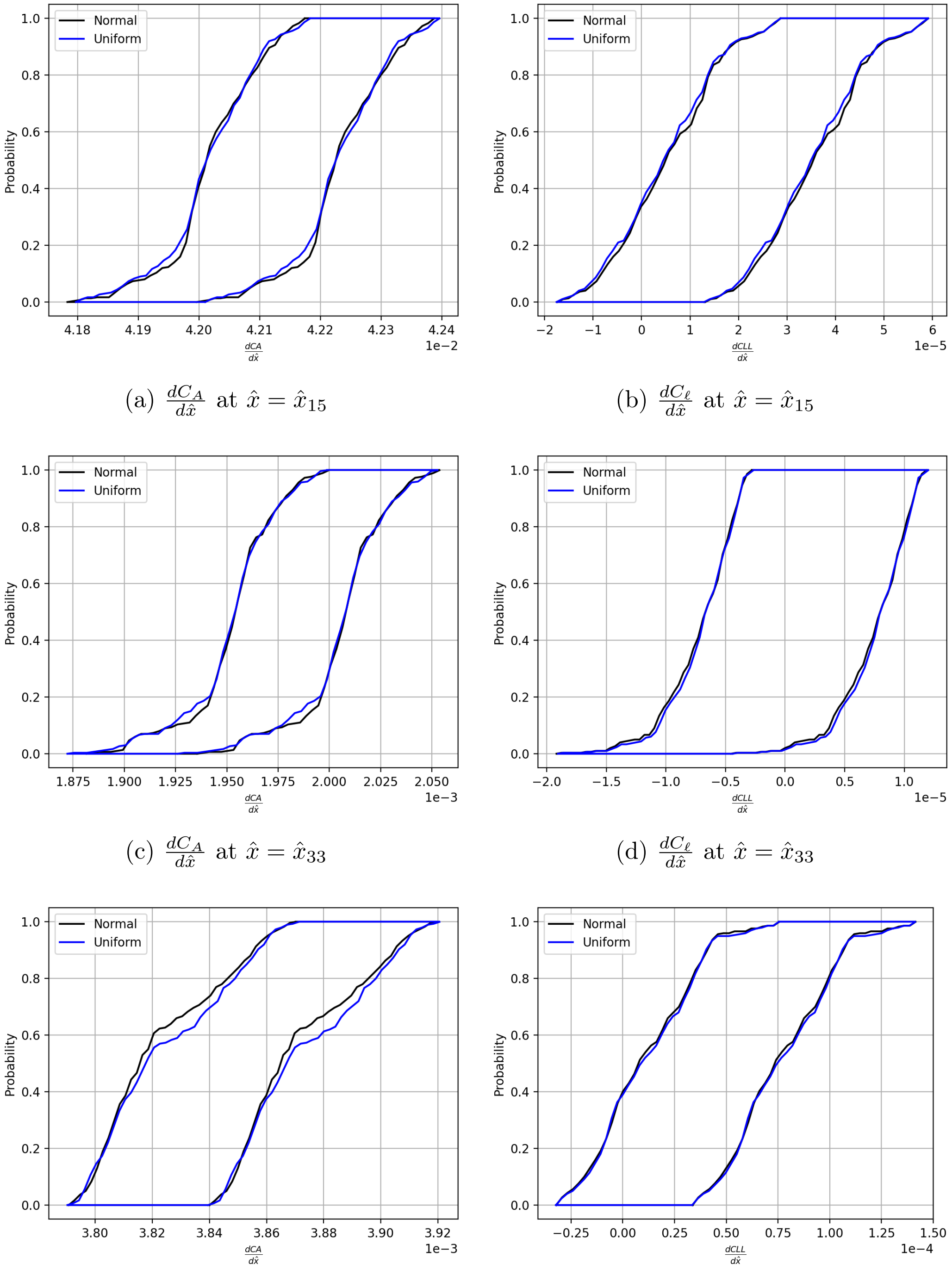
(c) $\frac{dC_m}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(d) $\frac{dC_m}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

Figure 4.15: Resampling $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ simulated uncertainty dispersions for 300 C_N & C_m

shape for the rolling moment coefficient line load result shown at $\hat{x} = \hat{x}_{33}$ in Figure 4.20d but at $\hat{x} = \hat{x}_{15}$ in Figure 4.20b there is very little difference. The probability box shapes for related forces and moments are similar, regardless of location, shown for $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ in Figure 4.21 and for $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ in Figure 4.22.

The simulated uncertainty results for the BSP-Copula method is shown in Figures 4.23, with the figures containing the remaining aerodynamic coefficients in Figures A.19 and A.20 in Appendix A. The consequence of the larger leave-one-out errors can be seen particularly in Figures 4.23 and A.20. For example the wide gap in potential values around $\hat{x}$=15 in these figures is due to this large error dominating over the simulated aleatoric uncertainty.



(a) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

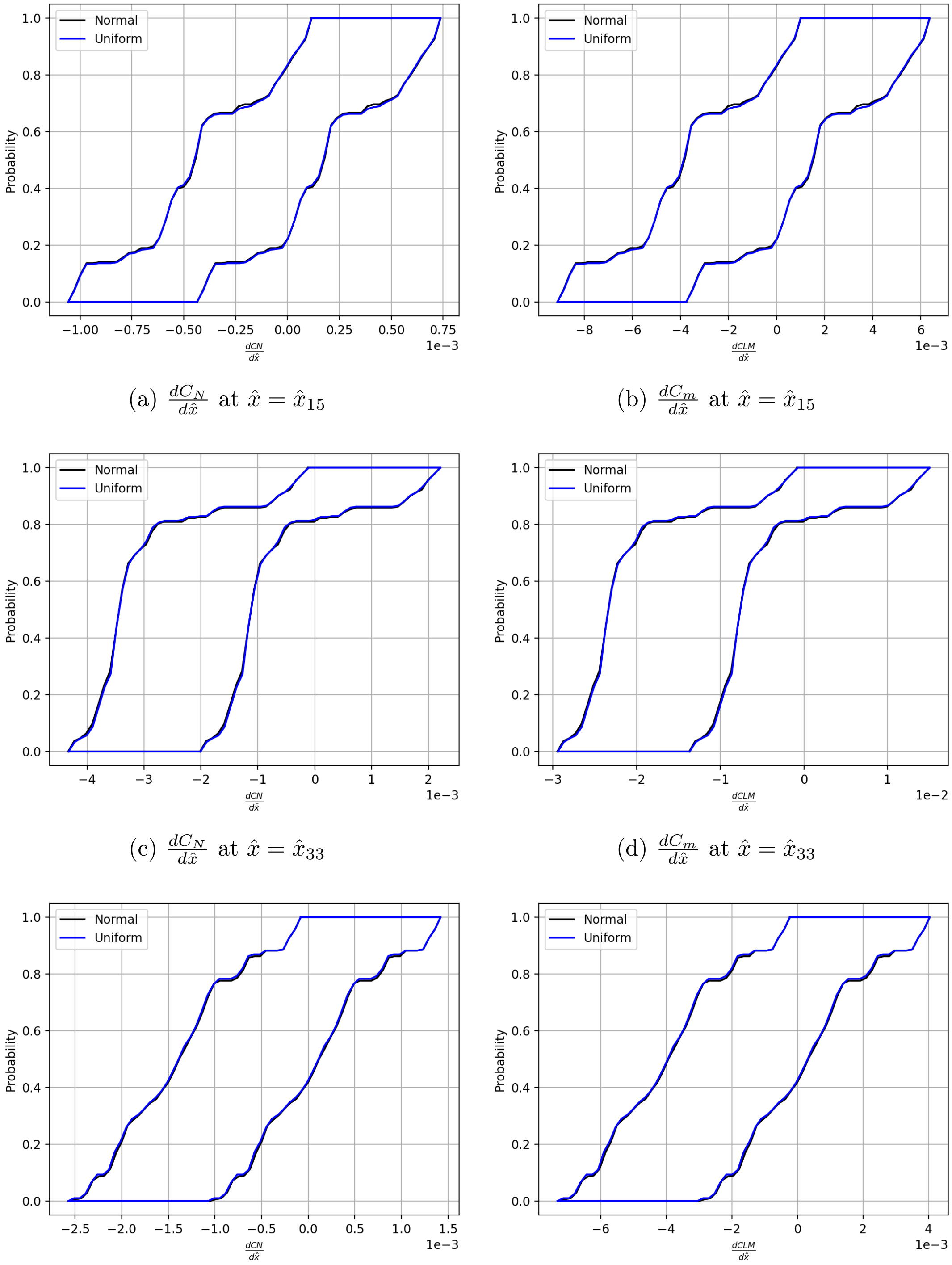
(c) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.16: Resampling $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ probability boxes for simulated uncertainty



(a) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

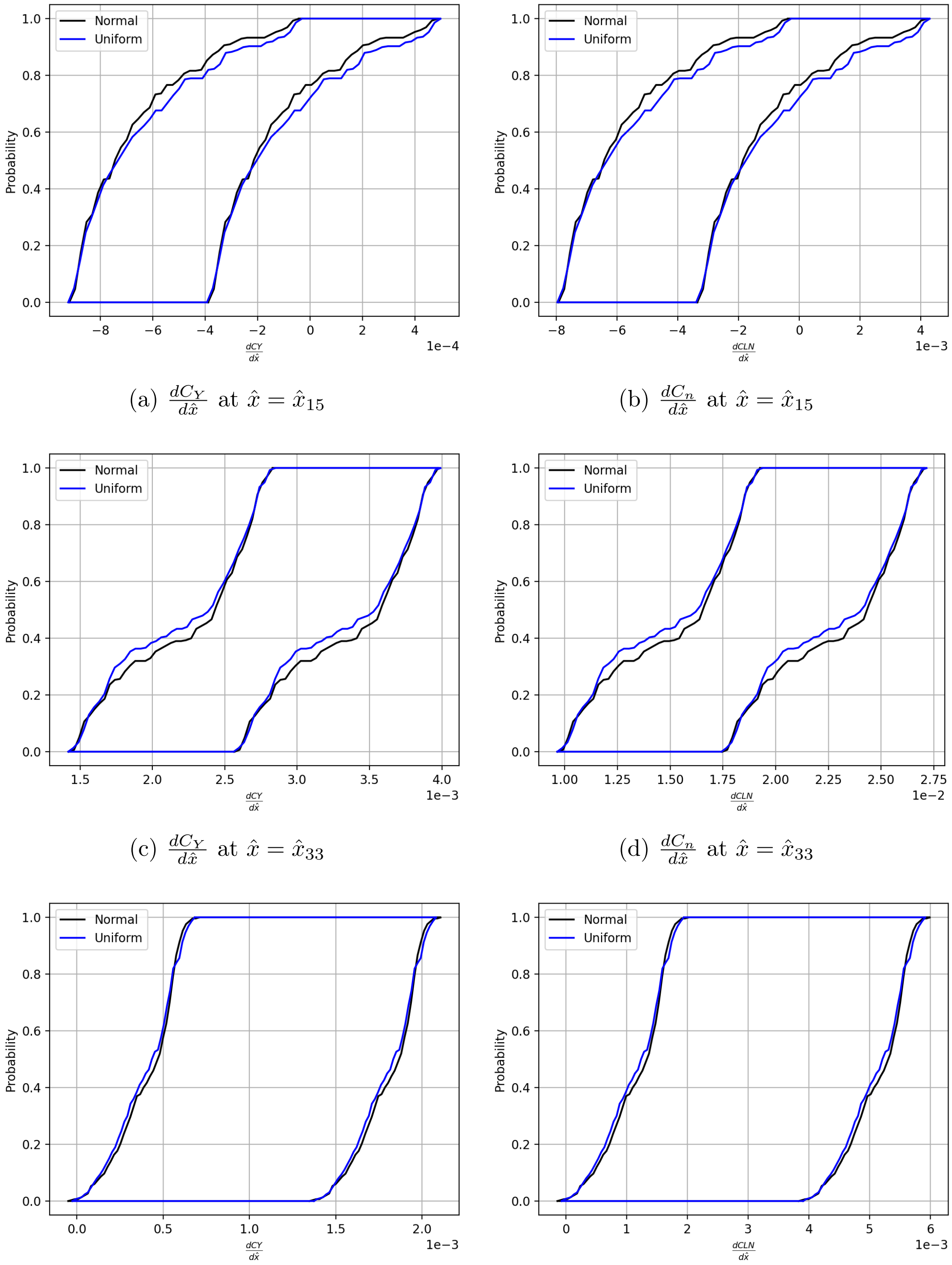
(c) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.17: Resampling $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ probability boxes for simulated uncertainty



(a) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

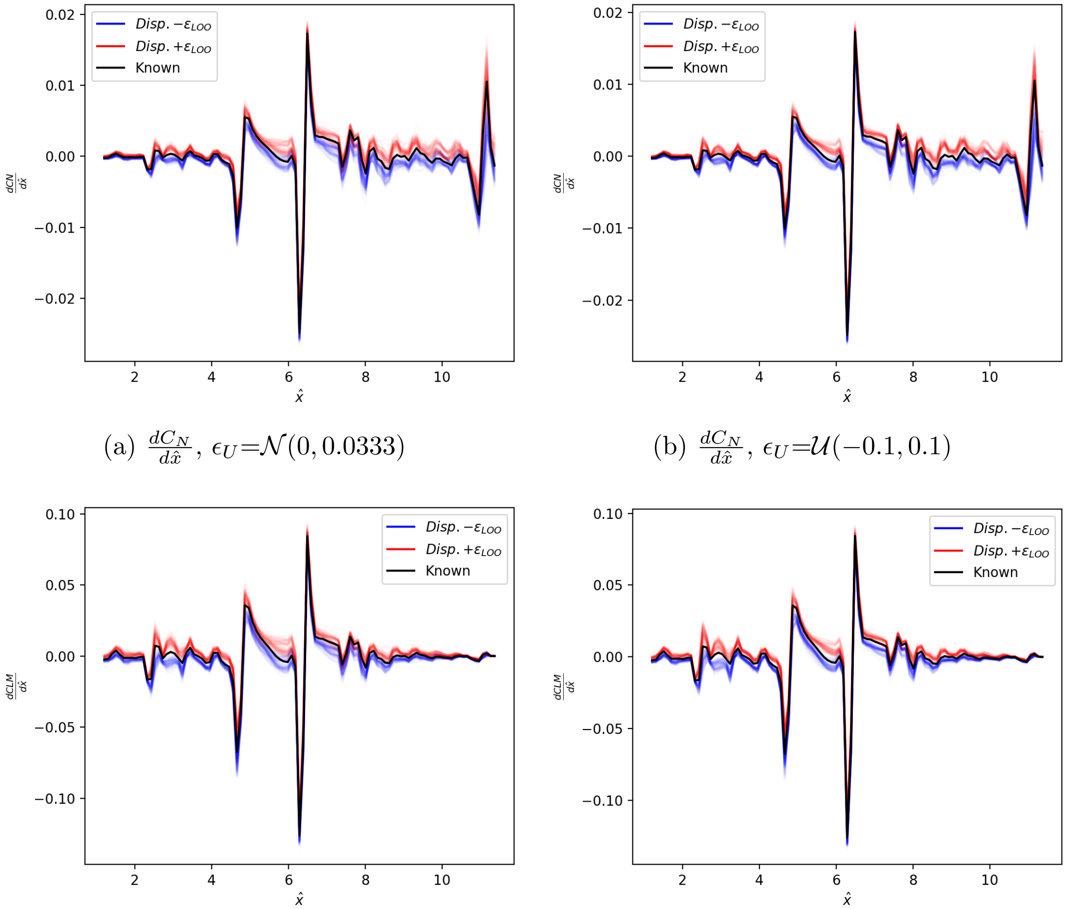
(c) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.18: Resampling $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ probability boxes for simulated uncertainty



(a) $\frac{dC_N}{d\hat{x}}$, $\epsilon_U = \mathcal{N}(0, 0.0333)$

(b) $\frac{dC_N}{d\hat{x}}$, $\epsilon_U = \mathcal{U}(-0.1, 0.1)$

(c) $\frac{dC_m}{d\hat{x}}$, $\epsilon_U = \mathcal{N}(0, 0.0333)$

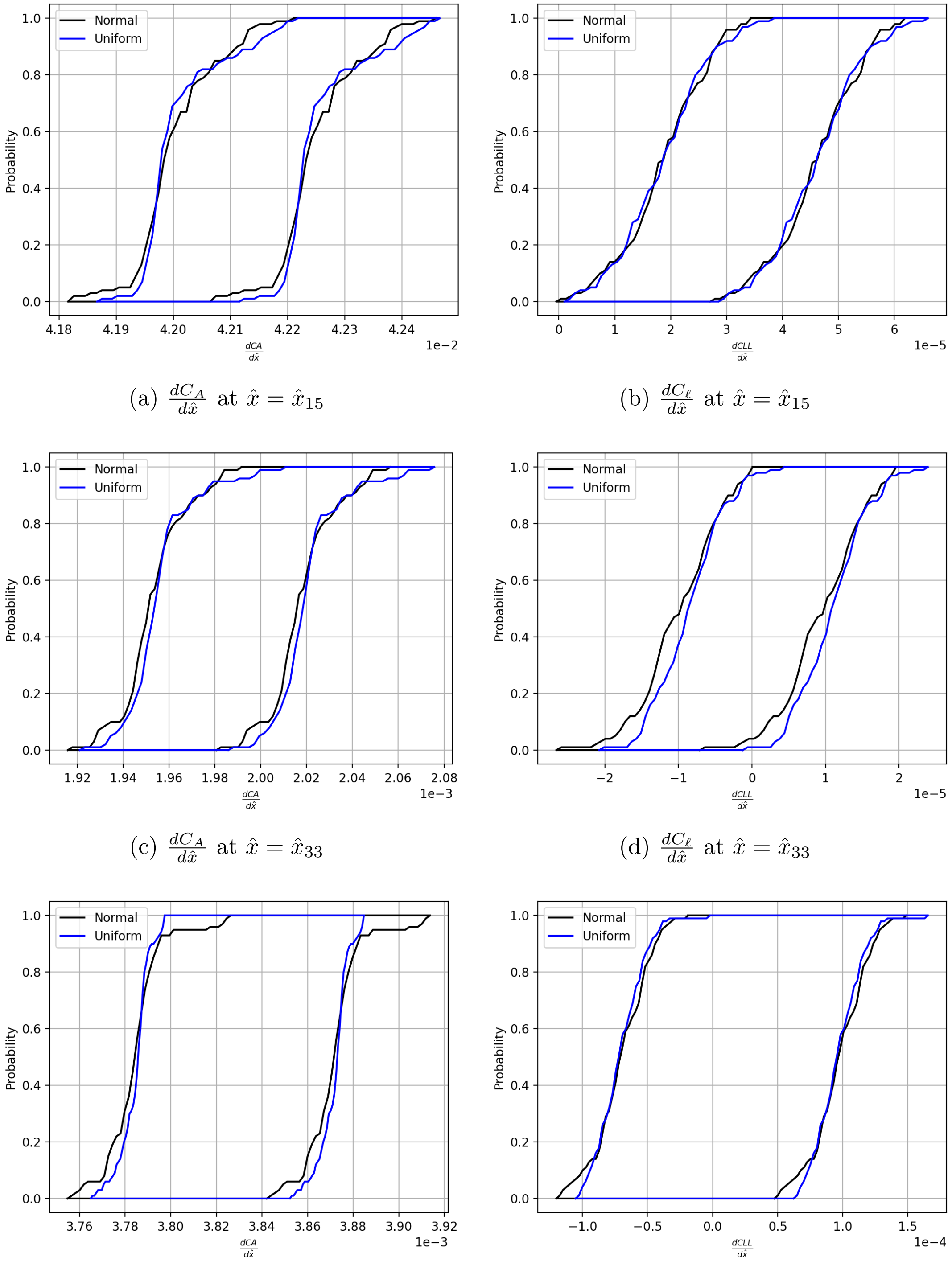
(d) $\frac{dC_m}{d\hat{x}}$, $\epsilon_U = \mathcal{U}(-0.1, 0.1)$

Figure 4.19: MVN-Copula $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ simulated uncertainty dispersions for 100 C_N & C_m

The probability boxes in Figure 4.24 - 4.26 show some differences in how the aleatoric uncertainties were modelled. This is not consistent with the previous method's results and is another warning sign to consider that this method was not performing as desired.

Discussion

The simulated uncertainty results, in particular the probability boxes, shows some amount of variability depending on the method used. Using the different models in aleatoric uncertainty shouldn't greatly affect the probability boxes, as integrated coefficient values sampled from



(a) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

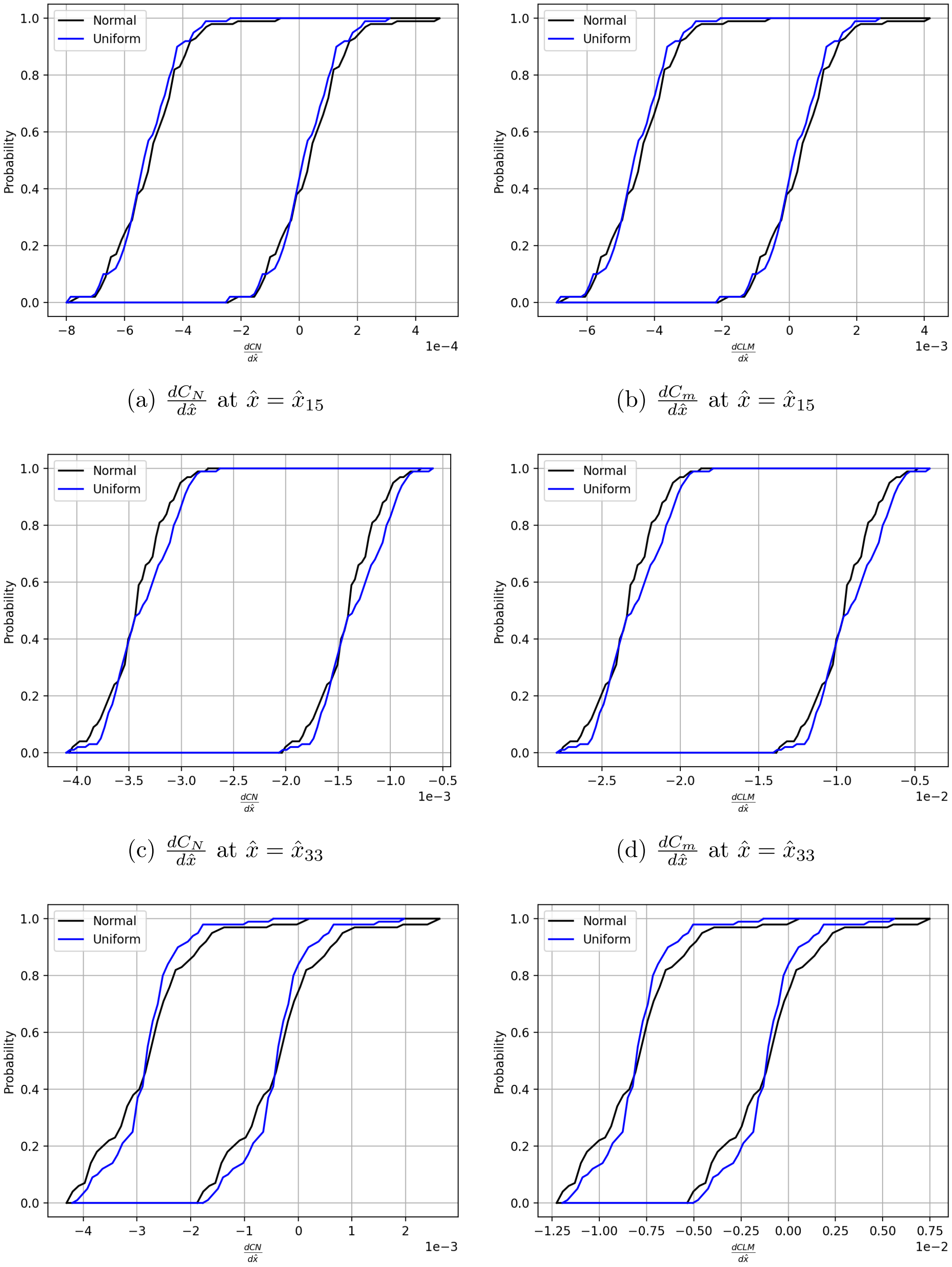
(c) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.20: MVN-Copula $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ probability boxes for simulated uncertainty



(a) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

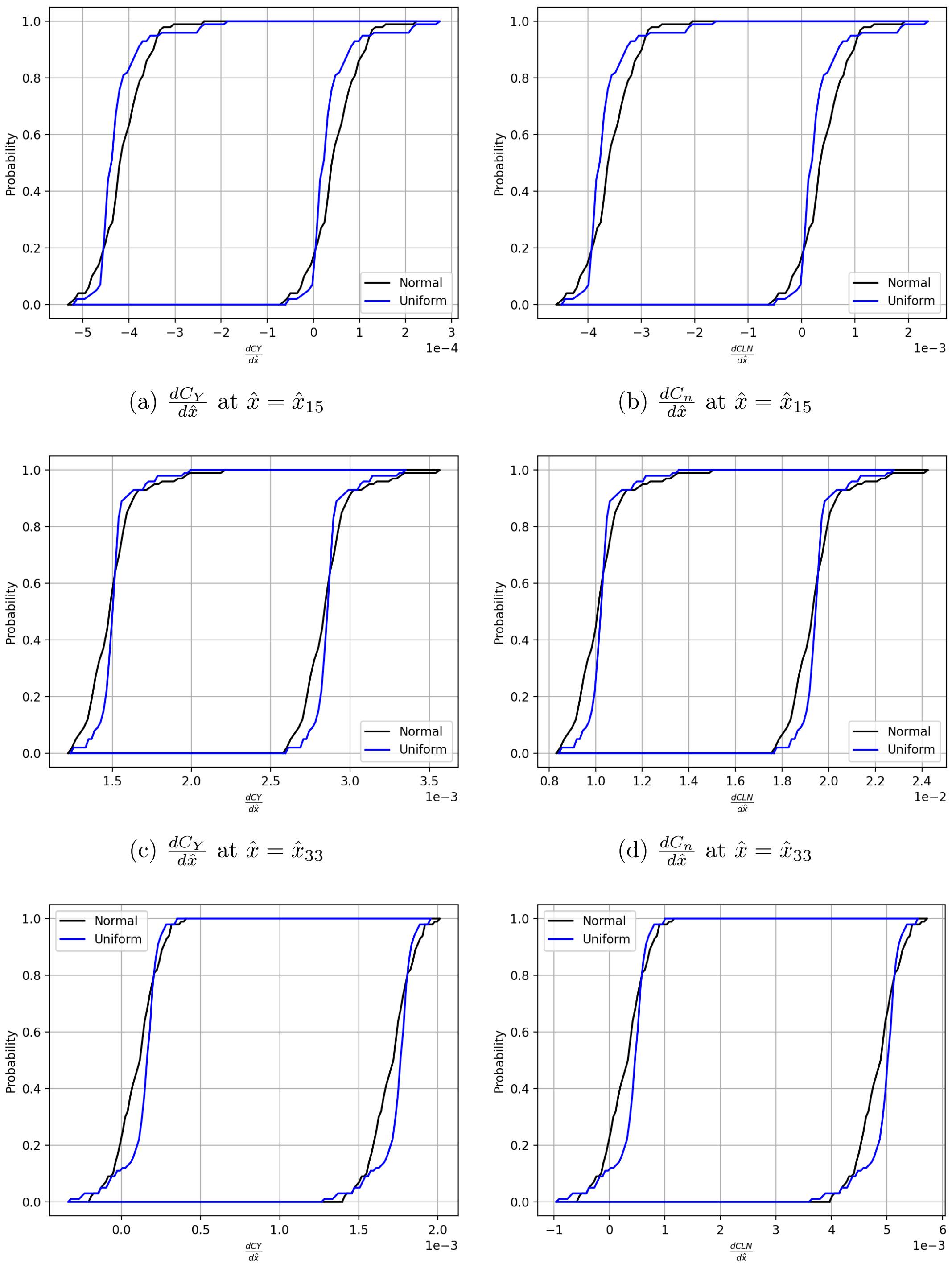
(c) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.21: MVN-Copula $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ probability boxes for simulated uncertainty



(a) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

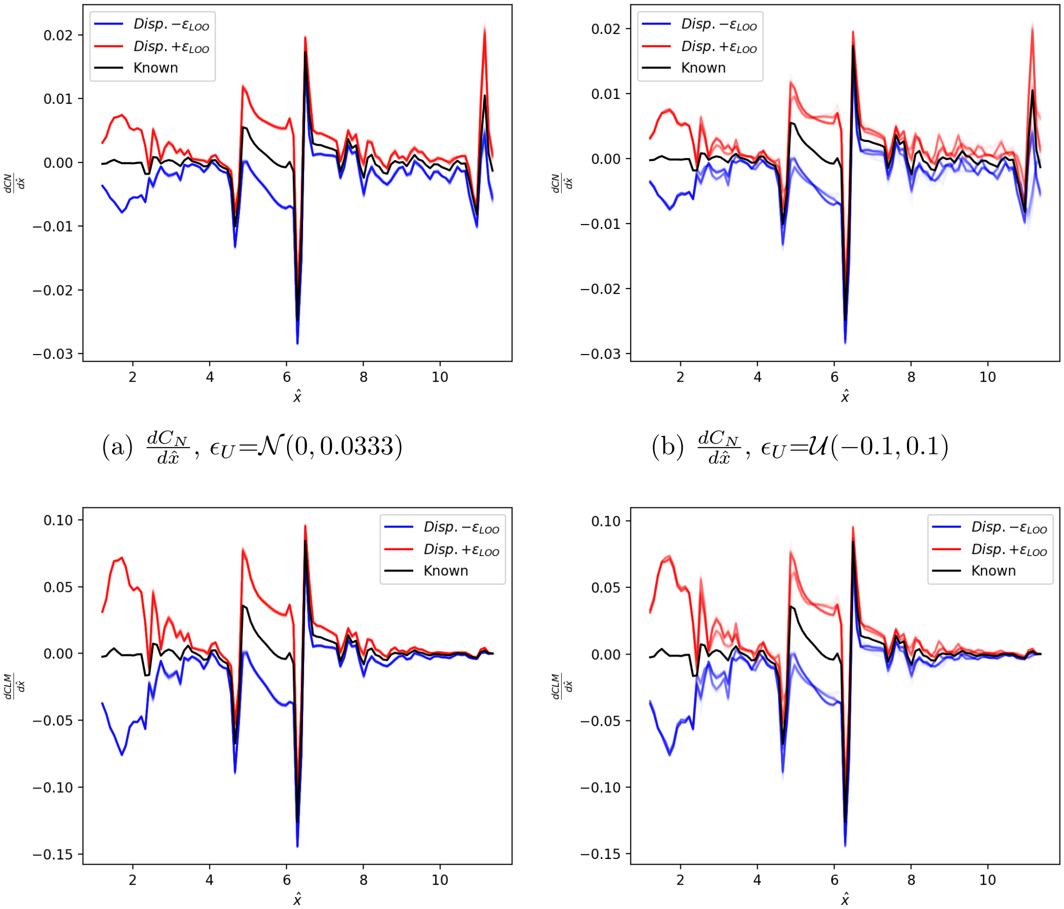
(c) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.22: MVN-Copula $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ probability boxes for simulated uncertainty



(a) $\frac{dC_N}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

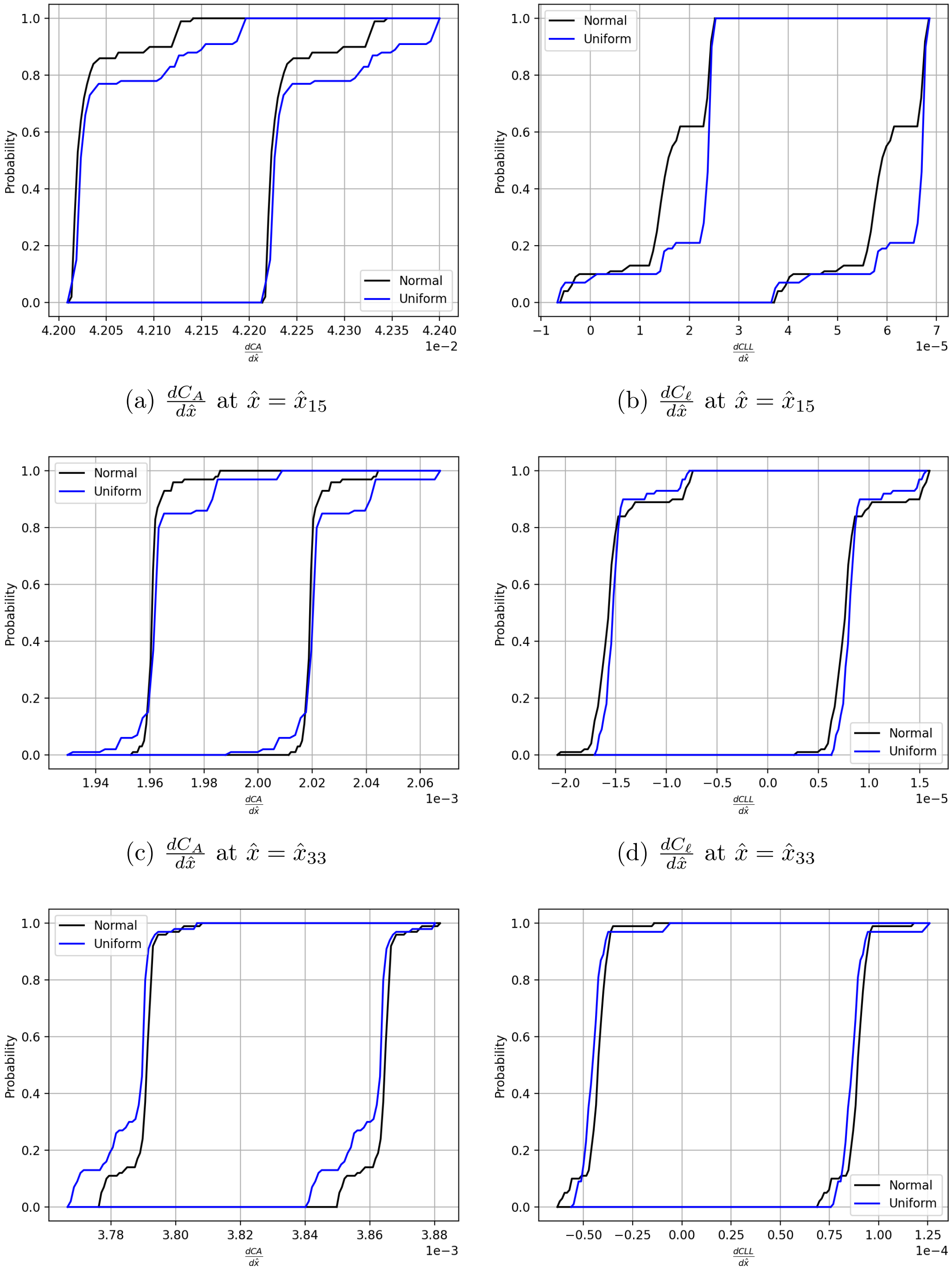
(b) $\frac{dC_N}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

(c) $\frac{dC_m}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(d) $\frac{dC_m}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

Figure 4.23: BSP-Copula $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ simulated uncertainty dispersions for 100 C_N & C_m

the normal and uniform distributions should be quite similar, just distributed differently. It is suspected that this difference between aleatoric models is due to the low number of samples taken from each uncertainty model. This is shown to be the case in the surface pressure probability box results, and that conclusion should be valid for these line load results as well. There are some large differences in shapes, for the Resampling and BSP-Copula methods, but that is attributed more to underlying deficiencies present in those methods. In general, the Resampling method probability boxes shapes are quite different than the other methods, although the BSP-Copula method has some variance in probability box shapes as well. The MVN-Copula seems to have a relatively consistent probability box



(a) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

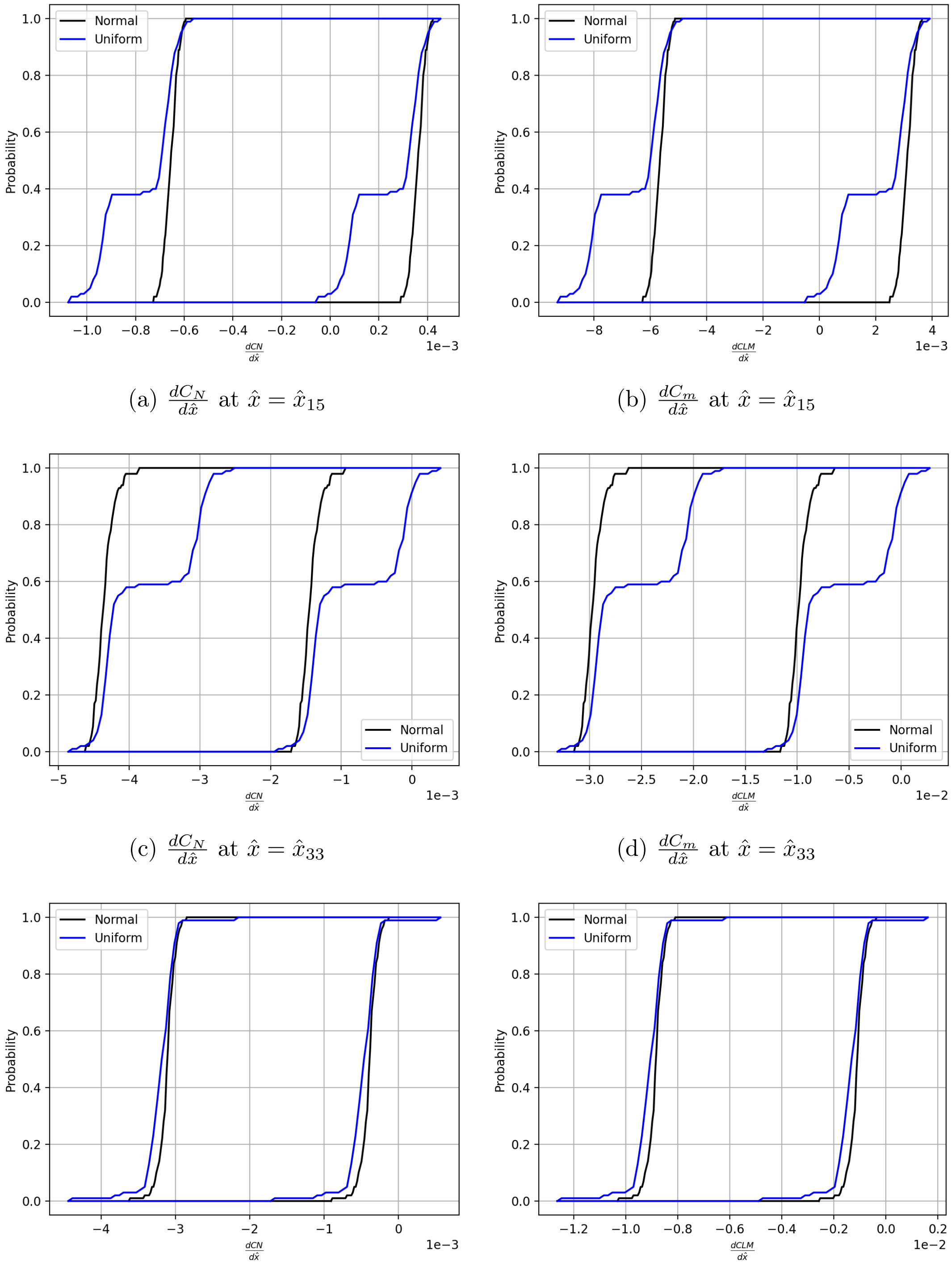
(c) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.24: BSP-Copula $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ probability boxes for simulated uncertainty



(a) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

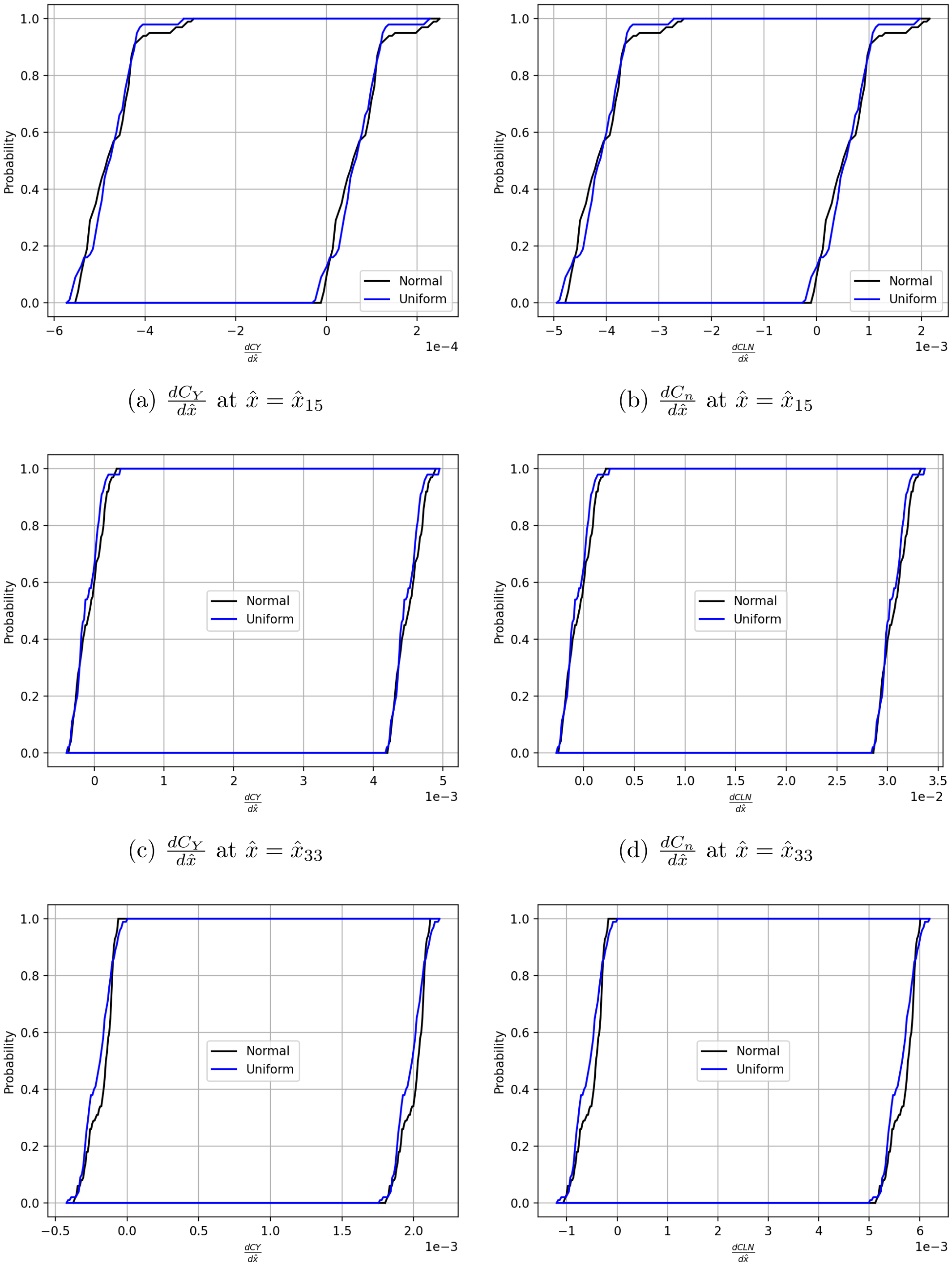
(c) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.25: BSP-Copula $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ probability boxes for simulated uncertainty



(a) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(c) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure 4.26: BSP-Copula $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ probability boxes for simulated uncertainty

shape over the different coefficients.

One of the core motivations for this work can be visualized in the non-normality of many of these probability boxes, due to the methods maintaining some representation of the known data distributions. If the bandwidths used in the copula methods were too large than the probability boxes would be overly smoothed and impose a more normal shape. While, the Resampling and MVN-Copula probability boxes appear to not be sensitive to the form of aleatoric uncertainty, the BSP-Copula probability boxes in general show a much greater sensitivity to the aleatoric uncertainty model, with some probability boxes displaying very different shapes depending on the aleatoric model. The dispersions shown for the BSP-Copula gives a clue as to why this could be happening, it appears the line load dispersions at many points along the vehicle are clustered quite close together. Compared to the other methods' dispersions where the dispersions tend to be more spread out in general. This means the BSP-Copula dispersions do not include many different possible values and the clustering is manifested in the probability boxes as sharp horizontal movements. It is possible that the bandwidth used in sampling from the BSP-Copula was too small and didn't allow for large enough variance in drawn values. Overall, it is unclear what caused the large differences between results using the different aleatoric distributions in some of the BSP-method probability boxes, but it is likely caused by the sampling procedure.

The difference in probability boxes between the Resampling method and the MVN-Copula method could potentially be explained by looking closely at the differences between the approaches. The Resampling method essentially just constructs a viable line load while the MVN-Copula method tries to find the line load with the highest probability density. This can, and likely is, resulting in the Resampling method providing line loads that have very small probability density values or in other words providing line loads that are not likely to occur, according to the statistical model used in the MVN-Copula method. The Resampling method also used 300 samples to construct the probability boxes, whereas the copula methods used just 100. This is potentially another factor driving the differences between aleatoric uncertainty distributions that is shown for the copula methods.

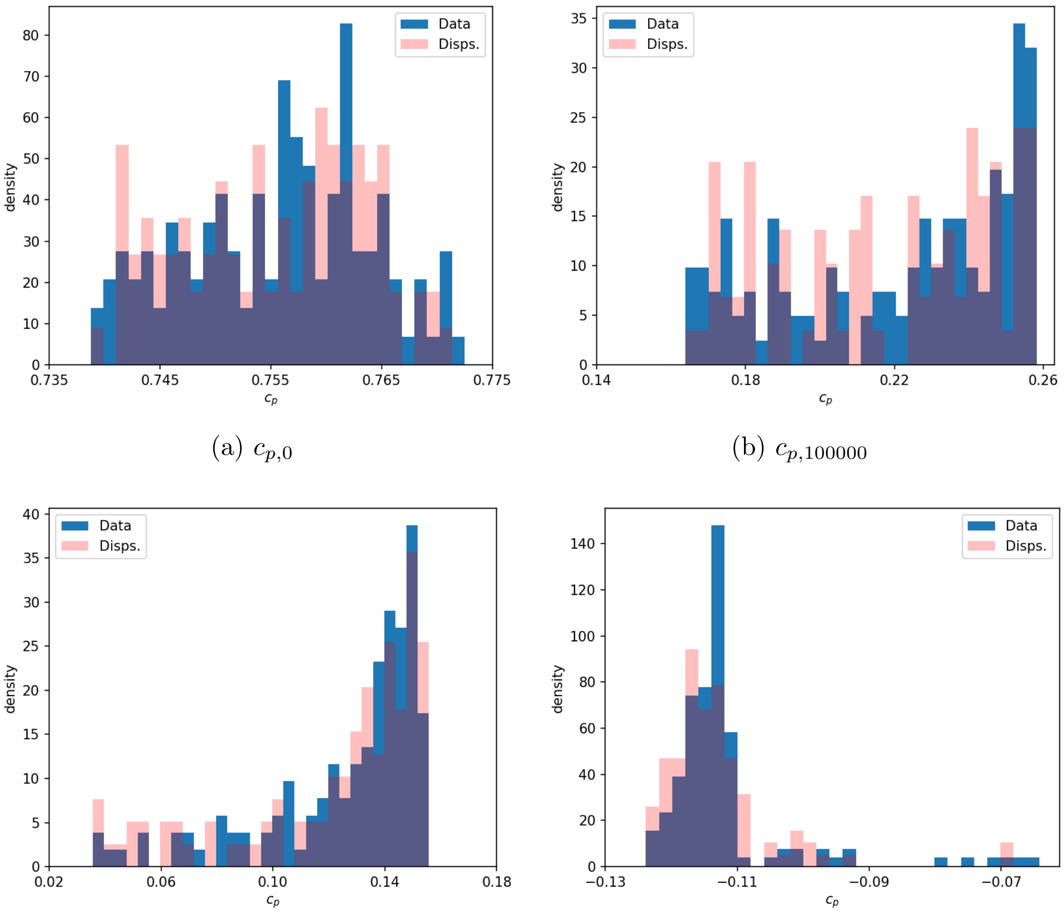
Overall it appears the MVN-Copula seemed to perform the best out of all the methods and is therefore also tested on surface pressure data. The BSP-Copula method results show clear issues that could potentially be resolved in order to improve the results. The Resampling method results are comparable to the MVN-Copula in many respects. However, the failure to pass validation for two coefficients and the lack of ability to estimate the probability density estimates of the generated line loads leave it as a less appealing method.

4.3 Surface Pressure

The MVN-Copula-based method from Section 3.3 is next applied to surface pressure data. As discussed in Section 3.4 the Resampling method from Section 3.2 is not applied to surface pressure data as the conditional distribution check step is not feasible with this data. The BSP-Copula method is not applied as the leave-one-out error in the line load data results was determined to be too large Due to this, and other performance issues, it was deemed not worth investigating the BSP-Copula method, at least in the form described in Chapter 3, on the surface pressure data.

Validation

The validation test was performed with a simultaneous draw of 100 force and moment coefficients from known data cases. Figure 4.27 shows histogram comparisons of generated surface pressures (red) with the known data used in the conditional marginal distribution estimates (blue). The distributions of the generated data and the known data in these histograms look to be similar in a qualitative inspection. The histograms at $c_{p,200000}$ and $c_{p,300000}$ give a clear example of the qualitative success of the method to match well with the known data. These histograms are skewed with easily apparent modes that the generated results are able to replicate. The generated surface pressures passed the MMD test as shown in Table 4.9. The surface pressure data only compared the coefficient distributions as the dimensionality of the data makes the test computationally too expensive. This is justified as the MMD test results in the line load section show that comparing the coefficient distributions always had the same concluding results as comparing the line load distributions directly. Although there is no guarantee the trend will hold for surface pressure data it is not feasible to test the full dimensional data.



(a) $c_{p,0}$

(b) $c_{p,100000}$

(c) $c_{p,200000}$

(d) $c_{p,300000}$

Figure 4.27: MVN-Copula c_p histograms for 100 integrated aerodynamic coefficient sets taken from known dataset.

Coefficient	Critical Value	Score
c_p	0.0260077	0.0192575

Table 4.9: MMD 2-sample test results on MVN-Copula generated surface pressures' $\mathbf{\Omega}$ coefficients.

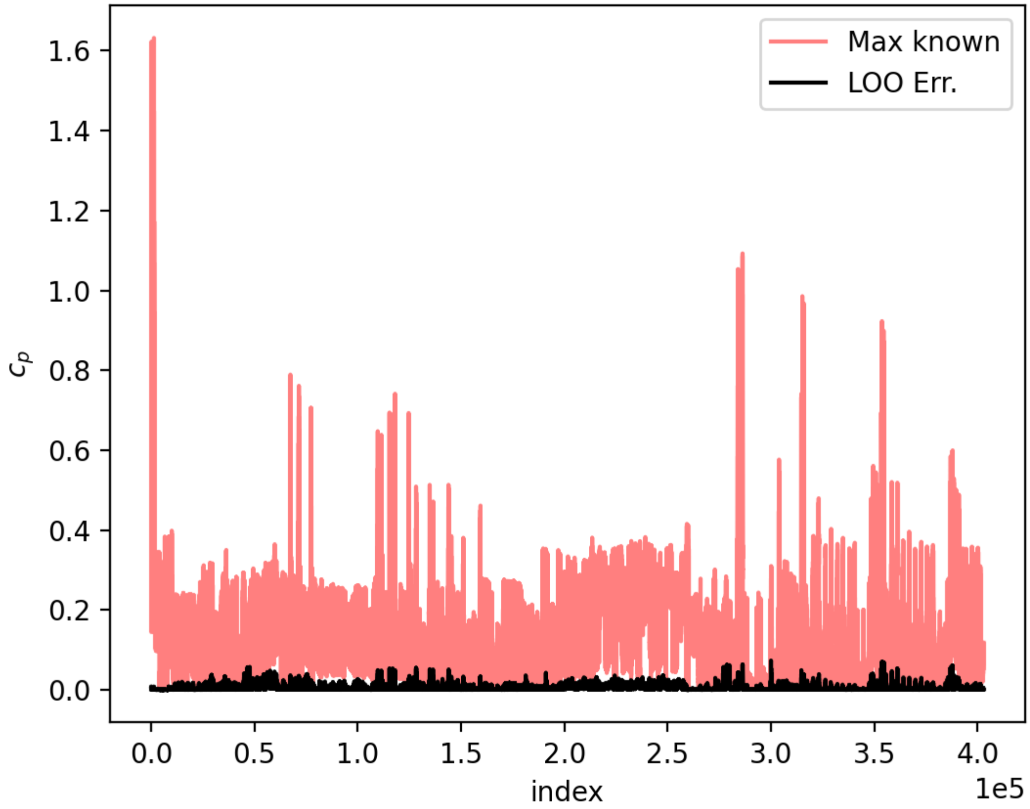


Figure 4.28: Leave-one-out errors for 29 known data points, 20k copula draws

Leave-One-Out Error Estimation

Figure 4.28 shows the leave-one-out error estimation for each of the 402909 surface pressure points. The red data is the maximum absolute value seen amongst the n_c known cases, the black data is the error estimate. The dimensionality of the data makes a total visualization of this data difficult. Figure 4.29 shows the ratio of the leave-one-out error to the max absolute value of the c_p at each index. The scale of the error estimate seems reasonable in general, however there is always a chance that LOO error will dominate at surface pressure locations with small absolute magnitudes. However, in the simulated uncertainty results presented next, the leave-one-out error will be applied to surface pressures that have accounted for an aleatoric uncertainty. Thus the relative magnitudes of the realized aleatoric uncertainty to the leave-one-out error is perhaps more indicative of the method's overall performance.

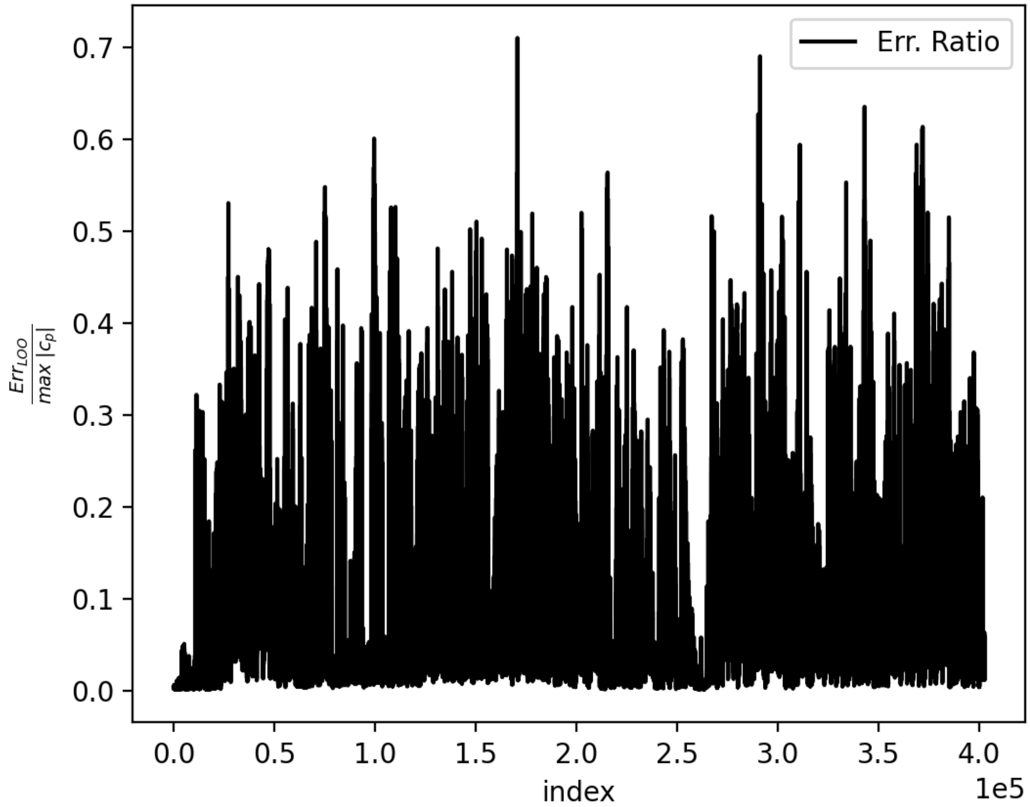


Figure 4.29: Ratio of leave-one-out errors to max absolute c_p for 29 known data points, 20k copula draws

Simulated Uncertainty

The simulated uncertainty process is the same as done for the line load case, however the description is repeated here for completeness. A simulated aleatoric uncertainty is applied to the integrated coefficients and surface pressures are generated using the copula-based methods. The aleatoric uncertainties were simulated according to $\tilde{C}^i_{z,j} = C^i_z(1 + \epsilon_{U,j})$, where C^i_z is the integrated coefficient z from the $i-$th known data case and $\epsilon_{U,j}$ is a value drawn from the aleatoric uncertainty model. The same $\epsilon_{U,j}$ was used for related forces and moments, for example one $\epsilon_{U,j}$ was drawn for normal force and then used to generate $\tilde{C}^i_{N,j}$ and $\tilde{C}^i_{LM,j}$. The $\epsilon_{U,j}$ are drawn from two different distributions to model the uncertainty, ϵ_U=$\mathcal{N}(0, 0.0333)$ and ϵ_U=$\mathcal{U}(-0.1, 0.1)$. This is done to compare the difference in using two commonly assumed aleatoric uncertainty models. The LOO error estimates from the previous sections are added to these generated surface pressures and a probability box is created for each surface pressure point.

It is not possible to show even a small fraction of the over 400000 probability boxes generated from this data, but several are included in Figures 4.30 - 4.32. Figure 4.30 shows the probability boxes for four different surface pressure points along the vehicle. The black line used the normal distribution uncertainty model, and the blue line used the uniform distribution. The figures show how the MVN-Copula method allows for spatial dependence in the uncertain surface pressure realizations of the aleatoric uncertainty in the integrated coefficients. For some surface pressure points, like those as the one shown in Figure 4.30b, the LOO error is approximately the same magnitude as the aleatoric uncertainty and both have similar contributions to the width of the probability box. However, in the case of Figure 4.30a, the aleatoric uncertainty is larger and contributes more to the overall uncertainty of that surface pressure point. The different aleatoric uncertainty models do show some differences in the resulting probability boxes depending on the surface pressure locations. For the included figures, there is not too much difference between the shapes of the probability boxes, depending just on the aleatoric uncertainty model. This is to be expected since the draws from the different aleatoric uncertainties are realized in the method as two different sets of integrated coefficients that have similar minimum and maximum bounds. The normal distribution could draw values far outside the uniform distribution bounds, however, by design, the 3-sigma bounds are similar.

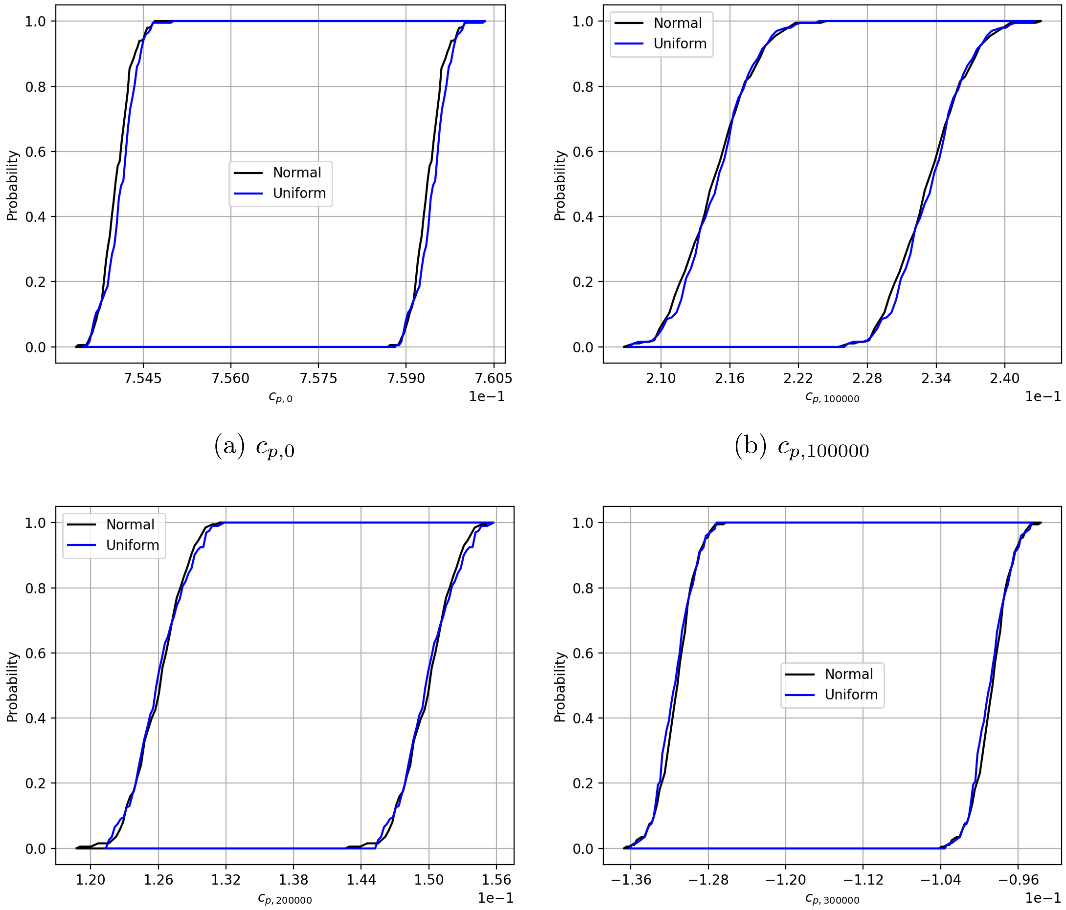
Figure 4.31 shows the probability boxes for the four surface pressure points that had the largest values for the ratio shown in Figure 4.29. The probability boxes for these points don't show significantly wider boxes as might be expected due to having the largest relative errors. Although the largest seen difference between normal and uniform uncertainty occurs in Figure 4.31b. Figures 4.32 shows the probability boxes for the four surface points that had the largest leave-one-out error ratio values shown in Figure 4.28. This figure shows that it doesn't appear that a large leave-one-out error ratio necessarily corresponds to a wider probability box. The overall proportions of these probability boxes are impacted by the balance between the leave-one-out error and the aleatoric uncertainty. This point is discussed further in the Discussion section included at the end of the included surface pressure results.

The differences between the aleatoric models is not expected, it is suspected to be due to natural probability variances of sampling with a low number of random samples. To investigate this, the number of drawn uncertain integrated samples was doubled to 400 and the probability boxes from those results are shown in Figure 4.33. Two probability boxes from Figures 4.31 and 4.32 were chosen for comparison in this figure, and it appears that the difference between the two aleatoric model's probability boxes becomes smaller with a larger number of random samples. However, there is a still a difference between the two aleatoric models that is driven by the uniform model being much more likely to draw values near the bounds of its distribution. This would change with a huge number of samples however, where now the normal distribution could likely sample a value beyond its three sigma bounds thus beyond the used uniform distribution's bounds. The key observation is that the aleatoric model doesn't have an impact on the shapes of the probability bounds as the number of randomly drawn integrated values increases. Therefore, with large enough sampling, the difference between the probability boxes using different aleatoric models is seen as a result of the possible minimum and maximum values.

Discussion

The only method used on the surface pressure data was the MVN-Copula method. The results were promising as it passed the validation test on the coefficients and, as discussed previously, there is confidence in validating with the coefficient MMD test. The small number of histograms presented, comparing point-wise distributions of the surface pressures, also showed a good qualitative agreement between the generated and known surface pressures. The leave-one-out errors also seem to have reasonably low values even though there are many more points to consider and the presented figures are quite crowded. The leave-one-out errors needs to be compared with the range of values generated by the aleatoric uncertainty in order to understand whether its magnitude is acceptable.

The probability boxes look reasonable, while there does appear to be small differences between the aleatoric uncertainty models used, this is determined to be driven by the low



(a) $c_{p,0}$

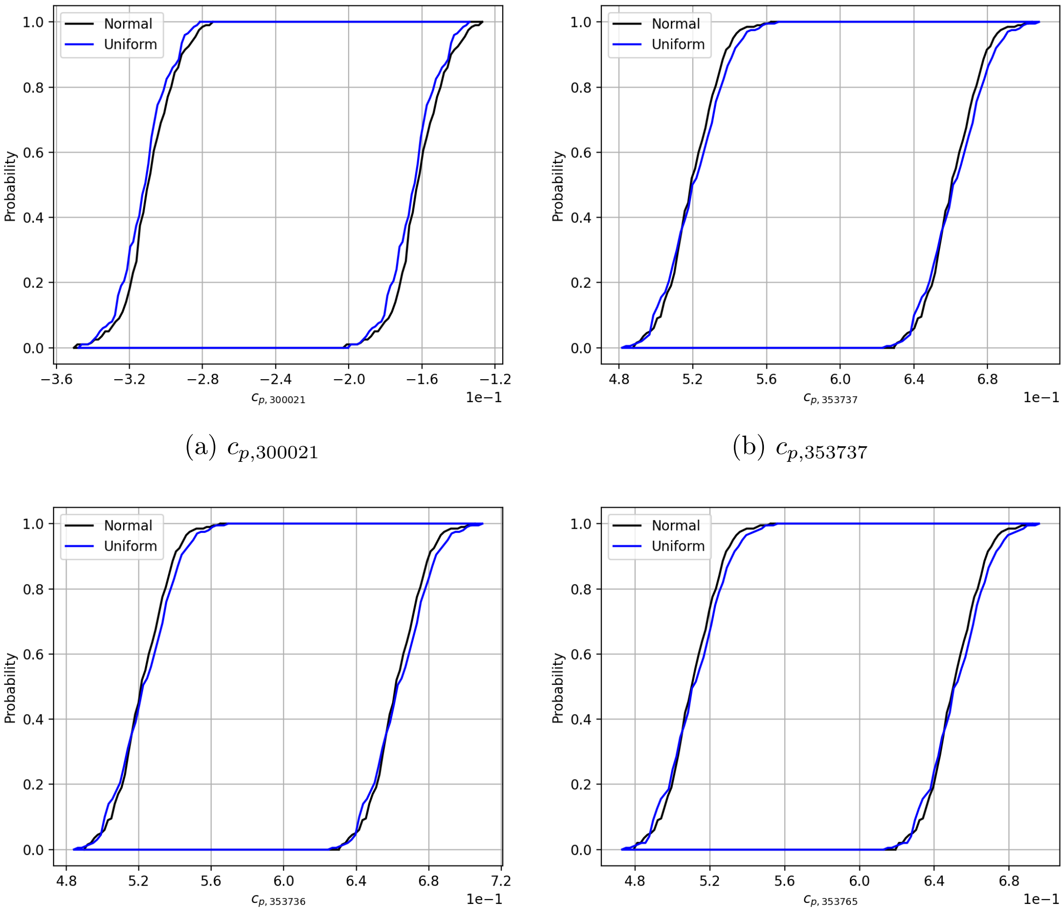
(b) $c_{p,100000}$

(c) $c_{p,200000}$

(d) $c_{p,300000}$

Figure 4.30: MVN-Copula c_p probability boxes for simulated uncertainty.

number of samples taken from the models. With a large enough number of samples, the main difference between the aleatoric models is mainly due to the different possible minimum and maximum values of each aleatoric model. However, this is more of a comparison of the difference in relative magnitude between the leave-one-out error estimate and the, arbitrarily, chosen aleatoric model. The probability boxes would look much differently if, for example, $\mathcal{U}(-0.01, 0.01)$ was used in place of $\mathcal{U}(-0.1, 0.1)$. In this new case, the leave-one-out error would be unchanged and would now dwarf the aleatoric uncertainty, thus the probability boxes would look much wider and square-like. This inspires the question, how realistic is the chosen aleatoric uncertainty models in a real world database scenario? Unfortunately the



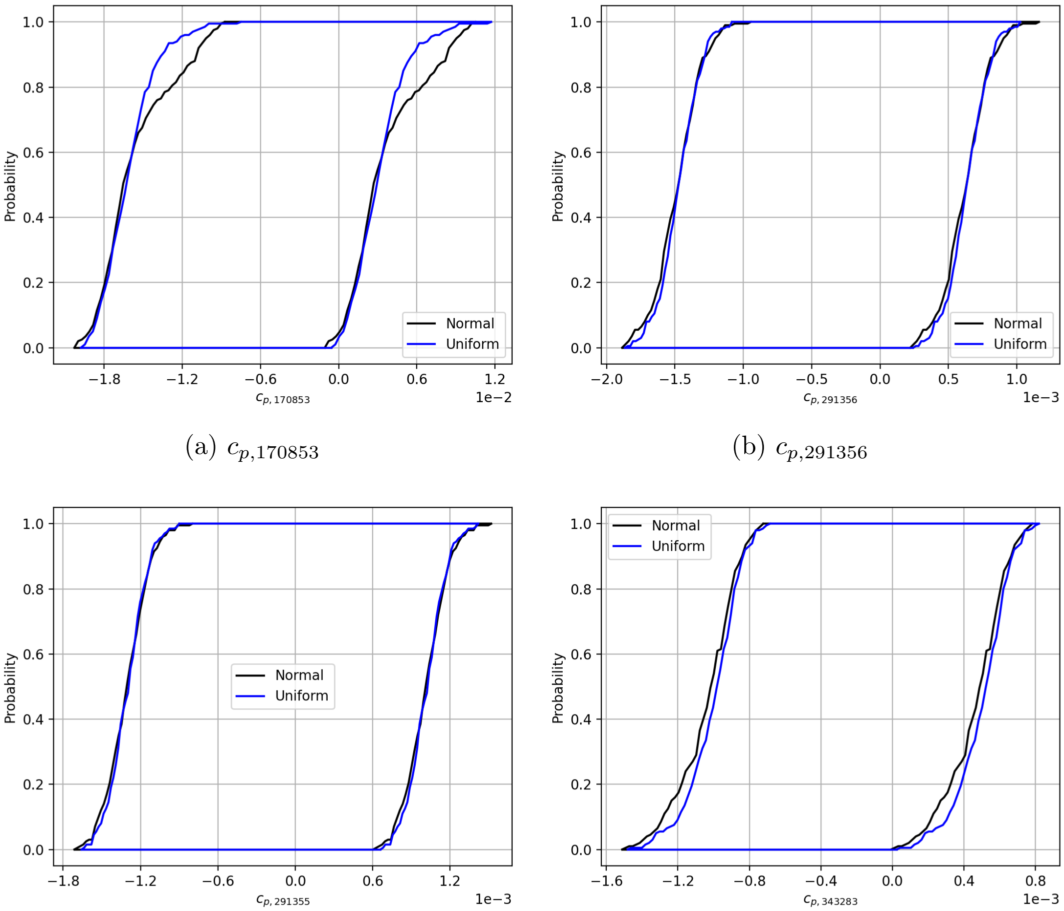
(a) $c_{p,300021}$

(b) $c_{p,353737}$

(c) $c_{p,353736}$

(d) $c_{p,353765}$

Figure 4.31: MVN-Copula c_p probability boxes for simulated uncertainty for locations of largest leave-one-out error.



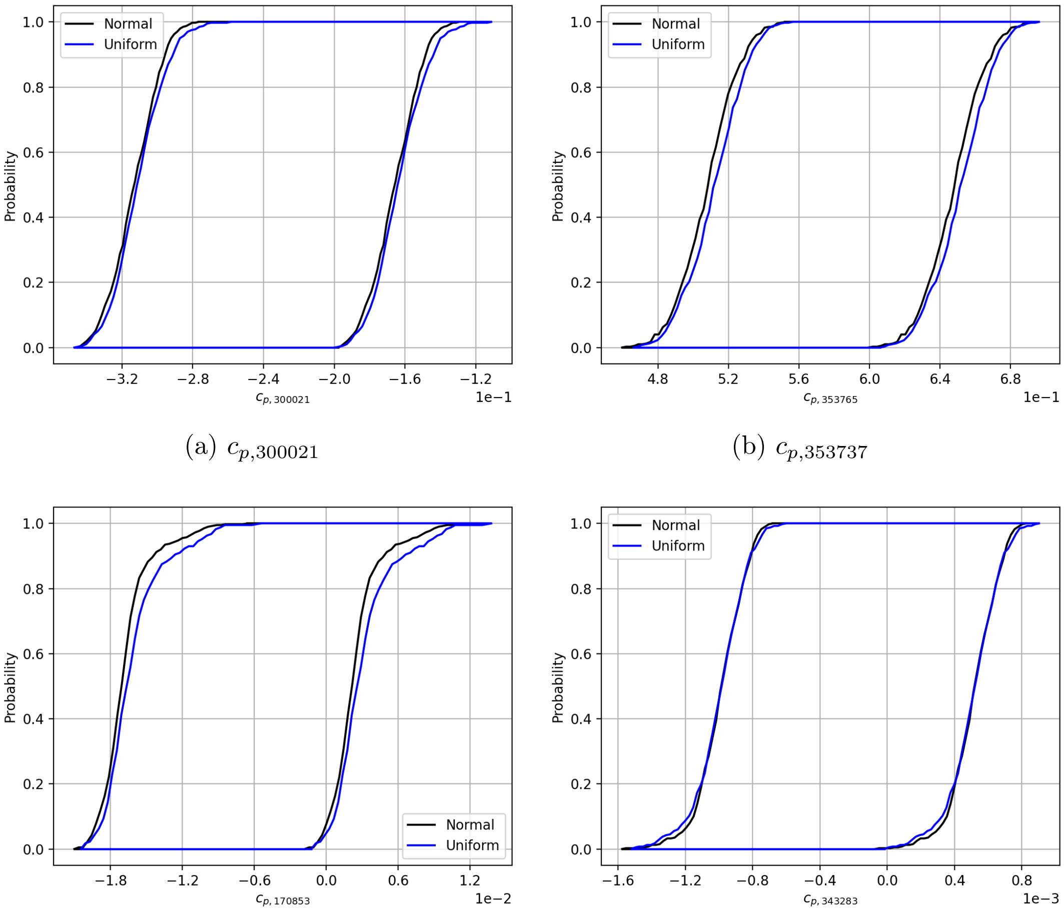
(a) $c_{p,170853}$

(b) $c_{p,291356}$

(c) $c_{p,291355}$

(d) $c_{p,343283}$

Figure 4.32: MVN-Copula c_p probability boxes for simulated uncertainty for locations of largest leave-one-out error ratios.



(a) $c_{p,300021}$

(b) $c_{p,353737}$

(c) $c_{p,170853}$

(d) $c_{p,343283}$

Figure 4.33: MVN-Copula c_p dense probability boxes for simulated uncertainty for select locations.

publicly available results for the Space Launch System aerodynamic databases are sanitized prior to publishing.

There is non-sanitized uncertainty data published in older work dealing with the Ares I vehicle in [28], but there is no scale to translate these values into percent uncertainties. The work in [33] provides a comparison of axial force coefficient results across different CFD solvers, reporting percent differences of around 10 to 20 percent across various mach numbers. The aleatoric uncertainty used in the simulated uncertainty results in this chapter represents about a 10 percent uncertainty in each integrated coefficient. Thus it seems like a relatively conservative test of uncertainty considering that this type of solver-to-solver comparison term is often included in a total uncertainty term used in aerodynamic databases. For example, [7] reports a CART3D-to-OVERFLOW comparison term in the authors' total uncertainty build up, where this term is never the largest contributor for any aerodynamic coefficient. This gives confidence that the simulated uncertainty is a conservative test of the method's performance in a practical scenario with a complete uncertainty quantification procedure.

Similarly to the line loads, the method is not built to extrapolate out beyond the data that is used to construct it. This could be an issue if the uncertainty is very large and valid uncertain draws are far beyond the known data. In this scenario, the best option is to use a method like that in [13] to generate data. Even though it is essentially a mathematical construction that could violate physical relationships between locations of the distributed load data, it will provide an answer that constructed from known data. It is another task to build a justification in using this generated answer, however this is not going to be discussed here.

Overall, the MVN-Copula method was relatively successful as an approach that can generate statistically consistent distributed loads according to uncertain integrated values. Another interesting capability of the copula-based methods is they are capable of providing any number of distributed loads at a given integrated coefficient value, including a corresponding estimate for probability density. This allows for the ability to analyze some chosen number of the generated loads with the largest probability density. For example, the spatial variance of these distributed loads could be examined to estimate the variability at a given spatial location in comparison to others.

Chapter 5

Conclusion and Future Work

The primary motivation of this work was to create a method that could generate uncertain distributed loads of a launch vehicle. Without uncertain distributed loads, engineers cannot estimate the probability that a launch vehicle will experience a load that exceeds its qualifications. In practice, predicting these loads has been done by relying on only nominal distributed loads and applying a safety factor. The methods investigated in this work attempt to improve on this practice by generating uncertain distributed loads that follow the statistical distributions implied by the known distributed load data.

In traditional engineering of launch vehicle aerodynamics the uncertainty quantification of integrated force and moment databases is able to estimate integrated aerodynamic coefficient uncertainty, however estimating an equivalent uncertainty for corresponding distributed loads is not done. These referred to distributed loads can take on at least two forms. Line loads and surface pressures, including skin friction, are the two forms directly addressed in this work. The surface pressure data allows for a more complete description of the aerodynamic loading on the vehicle at the cost of being a much larger, and thus computationally expensive, data set to work with. On the other hand, line loads allow for a more condensed description of the aerodynamic loading by integrating the surface pressure data over axial slices of the vehicle. However, this means the line loads don't allow for the examination of aerodynamic loading on a specific area of the vehicle, such as on a protuberance for example. The methods discussed in Chapter 3 attempt to provide the capability to generate distributed loads that are both consistent with the estimated uncertain integrated loads and consistent with the statistical distributions of the known data.

The results shown in Chapter 4 assess the success of the methods being applied to both line load and surface pressure data, with the MVN-Copula method being the only method also applied to surface pressure data. The investigated methods are the Resampling, MVN-Copula, and BSP-Copula methods. The copula methods utilize a statistical idea to model the dependence of between multi-dimensional random variables, in this case the random variables are coefficients used to linearly construct uncertain distributed loads. The MVN-Copula method uses a multi-variate normal distribution to model the copula, with the BSP-Copula method uses the Bayesian Sequential Partitioning method for this purpose. The computational time required for the Resampling method to be applied to the surface pressure data was the reasoning for it being omitted. The low performance of the BSP-Copula method when applied to the line load data was the justification for not applying it to the surface pressure data. However, the MVN-Copula method was successful when applied to both line load and surface pressure data. The MVN-Copula had similar leave-one-out error performance as the Resampling method; however, it also provides an estimate to the probability density of its generated line loads. Considering these observations, the MVN-Copula method is the recommended approach to be used for the task of generating uncertain distributed loads according to uncertainty quantified in integrated loads. Another key conclusion was that the surface pressure probability boxes were relatively independent of the choice of aleatoric uncertainty model, with only difference in the possible drawn extrema affecting the probability boxes.

The further development of the BSP-Copula method is a potential avenue for future work. That idea being that more work could be done on establishing a method for determining the BSP-Copula method's sampling procedure bandwidth parameter. It is possible that tuning this parameter could improve the samples generated from this method and allow for the axial force line load validation test to be passed. An initial idea for this is somehow relating the size of the partition that is being sampled from into the bandwidth. The motivation being to more naturally limit the drawn values to be within the drawn partition as the current method essentially applies floor or ceiling functions to the drawn values. Another available avenue for future work could be to investigate the near-failure of the BSP-Copula method to pass the validation test on the axial force coefficient line loads. It is possible that the methods could be altered or tuned for the different aerodynamic coefficients in order to improve performance, at least beyond the process for setting the bandwidths, which are already variable based on the coefficients.

An additional aspect that could be investigated in the future is due to launch vehicles with one or more boosters, including SLS, usually maintain separate line loads for the main center body and each booster. Discussing the rules for dispersing line loads in such a situation (for example considering how normal and lateral loads on the boosters create rolling moment loads on the combined stack) could be a fruitful area of future work. However, it should be noted that using the surface-pressure method can avoid this minefield of potential sign errors.

Moving forward, a potential extension of this work could be to extend even further from surface pressure data to using full flow solution fields. The motivation for doing that is so that the residuals of the generated flow fields could be checked directly against the governing equations used to generate the known data. This would show that the generated solutions are valid according to a desired residual tolerance level. It is a stretch however, as the size of the flow fields would require making data management and storage a consideration. Mathematical procedures could also become quite computationally costly and time prohibitive. Another motivation for producing uncertain volume solutions is that many applications that are not launch vehicle aerodynamics cannot be derived from surface quantities alone. Consider as just one example an estimate of the mass captured by an airplane engine inlet.

In this work, we have sought to create uncertain distributed loads that match the statistics of the known loads as completely as possible. To assess this goal for each of the methods, we have employed various multidimensional statistical tests and were able to meet most or all of the criteria. It should be noted that for most applications, the use of distributed loads is to calculate one or more scalars. For example, in the mass flow problem just mentioned as an application for uncertain volume solutions, the scalar is just mass flow, but estimating it requires knowledge of the density and velocity solution of a large part of the flow field. For launch vehicle aerodynamics, there is usually a collection of 20 or more critical loads, and each of them can be calculated from the distributed loads. A weaker set of success criteria would be to match the statistics for these integral loads, which might allow for simpler methods, and thus might be easier to implement in complex programs. However, aerodynamicists typically do not know these loads in advance, which depend on the internal structure of the vehicle. The goal in this work of matching the underlying statistics of the entire distributed load database, which may have seemed esoteric, is to ensure that users of the resulting uncertain distributed loads can be sure that the statistics of the scalar critical indicators will also be correct.

Bibliography

[1] Tim Barth. An overview of combined uncertainty and a-posteriori error bound estimates for cfd calculations. In *54th AIAA Aerospace Sciences Meeting*, 2016. AIAA Paper 2016-1062.

[2] Martin Benning and Martin Burger. Modern regularization methods for inverse problems. *Acta Numerica*, 27:1–111, 2018.

[3] R. T. Biedron, J. Calrson, J. M. Derlaga, P. A. Gnoffo, D. P. Hammond, K. E. Jacobson, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielson, M. A. Park, C. L. Rumsey, J. L. Thomas, K. B. Thompson, A. C. Walden, L. Wang, and W. A. Wood. FUN3D manual: 13.7. Technical Report TM-20205010139, NASA, 2020.

[4] Lorenz Biegler, George Biros, Omar Ghattas, Matthias Heinkenschloss, David Keyes, Bani Mallick, Youssef Marzouk, Luis Tenorio, Bart van Bloemen Waanders, and Karen Willcox. *Introduction.* John Wiley & Sons, Ltd, 2010.

[5] Aaron C. Burkhead and Derek J. Dalle. Statistically consistent dispersion of line loads to uncertain integrated forces and moments. In *AIAA SCITECH 2024 Forum*, 2024.

[6] Russel E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, January 1998.

[7] David T. Chan, Derek J. Dalle, Stuart E. Rogers, Jeremy T. Pinier, Floyd J. Wilcox, and Reynaldo J. Gomez. Space Launch System booster separation aerodynamic database development and uncertainty quantification. In *54th AIAA Aerospace Sciences Meeting*, 2016.

[8] David T. Chan, Derek J. Dalle, Stuart E. Rogers, Jeremy T. Pinier, Floyd J. Wilcox, and Reynaldo J. Gomez. Space Launch System booster separation aerodynamic database development and uncertainty quantification. In *54th AIAA Aerospace Sciences Meeting*, 2016.

[9] W. M. Chan. The overgrid interface for computational simulations on overset grids. In *32nd AIAA Fluid Dynamics Conference*, June 2002. AIAA Paper 2002-3188.

[10] Michael Christie, James Glimm, John Grove, David Higdon, David Sharp, and Merri Wood-Schultz. Error analysis and simulations of complex phenomena. *Los Alamos Natl. Lab. Rep. LA*, 29, 01 2005.

[11] Derek Dalle. CAPE: Computational Aerosciences Productivity & Execution, 2023. https://github.com/nasa/cape.

[12] Derek J. Dalle, Stuart E. Rogers, Aaron C. Burkhead, and Jamie Meeroff. Distribution of SLS integrated load uncertainty to surface pressures and sectional loads. In *ICCFD11 Conference 2022*, 2022.

[13] Derek J. Dalle, Stuart E. Rogers, Henry C. Lee, and Jamie Meeroff. Adjustments and uncertainty quantification for SLS aerodynamic sectional loads. In *2018 Applied Aerodynamics Conference*, 2018.

[14] Derek J. Dalle, Stuart E. Rogers, Jamie G. Meeroff, Aaron C. Burkhead, Daniel G. Schauerhamer, and Joshua F. Diaz. Launch vehicle ascent computational fluid dynamics for the Space Launch System. *Journal of Spacecraft and Rockets*, 0(0):1–14, 2023.

[15] Sonjoy Das, Roger Ghanem, and Steven Finette. Polynomial chaos representation of spatio-temporal random fields from experimental measurements. *Journal of Computational Physics*, 228:8726–8751, 12 2009.

[16] Didier Dubois and Henri Prade. An introduction to fuzzy systems. *Clinica Chimica Acta*, 270(1):3–29, 1998.

[17] Morris L. Eaton. *Multivariate statistics : a vector space approach.* Wiley, New York, 1983.

[18] Moulines Eric, Francis Bach, and Zaïd Harchaoui. Testing for homogeneity with kernel fisher discriminant analysis. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[19] Amber L. Favaregh, Heather Houlden, Jeremy T. Pinier, and Michael J. Hemsch. Space Launch System aerodynamic database uncertainty quantification methodologies. In *54th AIAA Aerospace Sciences Meeting*, 2016.

[20] Gianluca Geraci, Michael S. Eldred, and Gianluca Iaccarino. A multifidelity multilevel Monte Carlo method for uncertainty propagation in aerospace applications. In *19th AIAA Non-Deterministic Approaches Conference*, 2017.

[21] Roger Ghanem and P. D. Spanos. Polynomial Chaos in Stochastic Finite Elements. *Journal of Applied Mechanics*, 57(1):197–202, 03 1990.

[22] Roger G. Ghanem and Pol D. Spanos. *Stochastic Finite Elements: A Spectral Approach.* Springer New York, 1991.

[23] Michael B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

[24] J Glimm and D.H Sharp. Prediction and the quantification of uncertainty. *Physica D: Nonlinear Phenomena*, 133(1):152–170, 1999.

[25] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel approach to comparing distributions. In *AAAI-07: Twenty-Second Conference on Artificial Intelligence*, 2007.

[26] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(null):723–773, mar 2012.

[27] J.C. Helton, J.D. Johnson, and W.L. Oberkampf. An exploration of alternative approaches to the representation of uncertainty in model predictions. *Reliability Engineering & System Safety*, 85(1):39–71, 2004. Alternative Representations of Epistemic Uncertainty.

[28] Heather Houlden, Amber Favaregh, and Michael Hemsch. Quantification of the uncertainties for the Ares I A106 ascent aerodynamic database. In *27th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, 2010.

[29] John Jakeman, Michael Eldred, and Dongbin Xiu. Numerical approach for quantification of epistemic uncertainty. *Journal of Computational Physics*, 229(12):4648–4663, June 2010.

[30] P. Jaworski, F. Durante, W.K. Härdle, and T. Rychlik. *Copula Theory and Its Applications: Proceedings of the Workshop Held in Warsaw, 25-26 September 2009*. Lecture Notes in Statistics. Springer Berlin Heidelberg, 2010.

[31] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods*. Wiley, 2008.

[32] A. Keese and H. G. Matthies. Sparse quadrature as an alternative to Monte Carlo for stochastic finite element techniques. *PAMM*, 3(1):493–494, 2003.

[33] Cetin Kiris, Jeffrey Housman, Marshall Gusman, Daniel Schauerhamer, Karen Deere, Alaa Elmiligui, Khaled Abdol-Hamid, Edward Parlette, Mark Andrews, and John Blevins. Best practices for aero-database CFD simulations of Ares V ascent. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[34] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, 2009. Risk Acceptance and Risk Communication.

[35] Augustine Kong, Jun S. Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

[36] Michael W. Lee, Aaron Burkhead, Derek Dalle, and Heather Houlden. Informing the Space Launch System booster separation initial CFD run matrix with observed parametric sensitivity. In *AIAA SCITECH 2022 Forum*, 2022.

[37] Michael W. Lee, Derek J. Dalle, Michael Sanders, and Carole J. Addona. Development of aerodynamic loads databases for the Space Launch System booster separation event. In *AIAA SCITECH 2024 Forum*, 2024.

[38] S. H. Lee and W. Chen. A comparative study of uncertainty propagation methods for black-box-type problems. *Structural and Multidisciplinary Optimization*, 37(3):239–253, January 2009.

[39] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] Weixuan Li, Zhiming Lu, and Dongxiao Zhang. Stochastic analysis of unsaturated flow with probabilistic collocation method. *Water Resources Research*, 45(8), 2009.

[41] Alexander Litvinenko and Hermann G. Matthies. Inverse problems and uncertainty quantification. *arXiv preprint arxiv.org/abs/1312.5048*, 2014.

[42] Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J. Sutherland. Learning deep kernels for non-parametric two-sample tests. In *ICML'20: Proceedings of the 37th International Conference on Machine Learning*, 2020.

[43] Hui Jiang Luo Lu and Wing H. Wong. Multivariate density estimation by bayesian sequential partitioning. *Journal of the American Statistical Association*, 108(504):1402–1410, 2013.

[44] Aref Majdara and Saeid Nooshabadi. Nonparametric density estimation using copula transform, bayesian sequential partitioning, and diffusion-based kernel estimator. *IEEE Transactions on Knowledge and Data Engineering*, 32(4):821–826, 2020.

[45] Susan N. McMillin, Patrick R. Shea, Derek J. Dalle, Stuart E. Rogers, Nettie Roozeboom, Jamie Meeroff, and Henry C. Lee. Comparison of Space Launch System aerodynamic surface pressure measurements from experimental testing and CFD. In *AIAA Aviation 2019 Forum*, 2019.

[46] Jamie Meeroff, Derek J. Dalle, Stuart E. Rogers, Aaron C. Burkhead, Daniel G. Schauerhamer, and Joshua F. Diaz. Advances in Space Launch System booster separation cfd. In *AIAA SCITECH 2023 Forum*, 2023.

[47] Jamie Meeroff, Derek J. Dalle, Stuart E. Rogers, Aaron C. Burkhead, Daniel G. Schauerhamer, and Joshua F. Diaz. Validation of SLS ascent cfd surface pressure databases using Artemis I flight data. In *AIAA SCITECH 2024 Forum*, 2024.

[48] Ramon E. Moore. *Methods and Applications of Interval Analysis*. Society for Industrial and Applied Mathematics, 1979.

[49] Scott M. Murman and Laslo T. Diosady. Simulation of a hammerhead payload fairing in the transonic regime. In *54th AIAA Aerospace Sciences Meeting*, 2016. AIAA Paper 2016-1548.

[50] R.B. Nelsen. *An Introduction to Copulas*. Springer, New York, NY, USA, 1999.

[51] William Oberkampf and Jon Helton. Investigation of evidence theory for engineering applications. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2002.

[52] Shishir Pandya and William M. Chan. Computation of sectional loads from surface triangulation and flow data. In *20th AIAA Computational Fluid Dynamics Conference*, 2011. AIAA Paper 2011-3680.

[53] Andrew J. Patton. Modelling asymmetric exchange rate dependence. *International Economic Review*, 47(2):527–556, 2006.

[54] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[55] Jeremy T. Pinier, David W. Bennett, Gary E. Erickson, Noah M. Favaregh, Heather P. Houlden, William G. Tomek, and John A. Blevins. Space Launch System ascent static aerodynamics database development. In *52nd Aerospace Sciences Meeting*, 2014.

[56] Brent W. Pomeroy and Steven E. Krist. Validation of Kestrel IDDES Simulations for SLS Transition Analysis. In *AIAA SCITECH 2022 Forum*, 2022.

[57] Nettie Roozeboom and Jennifer K. Baerny. Customer guide to pressure-sensitive paint testing at NASA Ames Unitary Plan Wind Tunnels. In *55th AIAA Aerospace Sciences Meeting*, 2017.

[58] Mats Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9(2):65–78, 1982.

[59] David W. Scott and Stephan R. Sain. Multidimensional density estimation. In C.R. Rao, E.J. Wegman, and J.L. Solka, editors, *Data Mining and Data Visualization*, volume 24 of *Handbook of Statistics*, pages 229–261. Elsevier, 2005.

[60] D.W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, Hoboken, NJ, USA, 1992.

[61] Patrick R. Shea, David T. Chan, Morgan A. Walker, Sarah Langston, Jesse G. Collins, Lee Mears, Elizabeth Rieken, and Jeremy Pinier. Force and moment analysis for the high reynolds number wind tunnel test of the Space Launch System at ascent conditions. In *AIAA SCITECH 2023 Forum*, 2023.

[62] Patrick R. Shea, Jeremy T. Pinier, and Timothy E. Curry. Evaluation of Artemis I aerodynamic force and moment reconstructions and preflight predictions for ascent flight. In *AIAA SCITECH 2024 Forum*, 2024.

[63] Patrick R. Shea, Jeremy T. Pinier, Heather Houlden, Amber Favaregh, Michael J. Hemsch, Derek J. Dalle, Stuart E. Rogers, Jamie Meeroff, and Henry C. Lee. Ascent aerodynamic force and moment database development for the Space Launch System. In *AIAA Aviation 2019 Forum*, 2019.

[64] Francesco Soranna, Patrick S. Heaney, Martin K. Sekula, David J. Piatak, and James M. Ramey. Space Launch System unsteady forces developed from unsteady-pressure-sensitive-paint–based corcos model parameters. *Journal of Spacecraft and Rockets*, 0(0):1–17, 2023.

[65] Francesco Soranna, Martin K. Sekula, Patrick S. Heaney, James M. Ramey, and David J. Piatak. Coherence analysis of the Space Launch System using unsteady pressure sensitive paint. In *AIAA Scitech 2021 Forum*, 2021.

[66] M. A. Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, 1974.

[67] Laura Swiler, Randall Mayes, and Michael Eldred. Epistemic uncertainty in the calculation of margins. *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 05 2009.

[68] Paolo Vinai, Rafael Macian-Juan, and Rakesh Chawla. A statistical methodology for quantification of uncertainty in best estimate code physical models. *Annals of Nuclear Energy*, 34(8):628–640, 2007.

[69] Zhipeng Wang and David W. Scott. Nonparametric density estimation for high-dimensional data—algorithms and applications. *WIREs Computational Statistics*, 11(4):e1461, 2019.

[70] Long Wen, Liang Gao, and Xinyu Li. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1):136–144, 2019.

[71] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[72] Thomas J. Wignall. Liftoff and transition database generation for launch vehicles using data-fusion-based modeling. In *AIAA Aviation 2019 Forum*, 2019.

[73] Thomas J. Wignall and Heather Houlden. Uncertainty quantification for launch vehicle aerodynamic lineloads. In *AIAA Scitech 2020 Forum*, 2020.

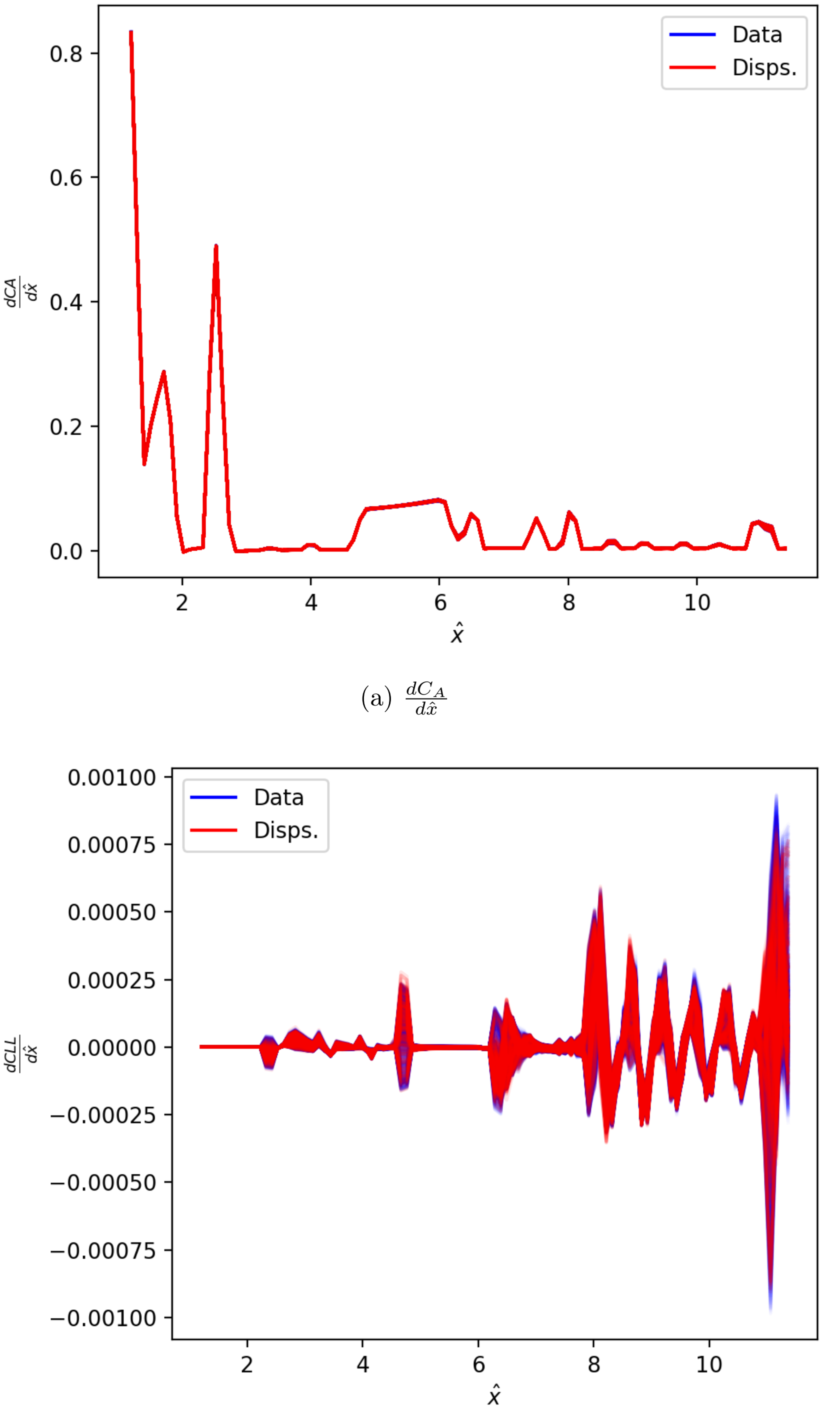
[74] Xu Wu, Ziyu Xie, Farah Alsafadi, and Tomasz Kozlowski. A comprehensive survey of inverse uncertainty quantification of physical model parameters in nuclear system thermal–hydraulics codes. *Nuclear Engineering and Design*, 384:111460, 2021.

[75] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[76] Jiaxin Zhang. Modern Monte Carlo methods for efficient uncertainty quantification and propagation: A survey. *WIREs Computational Statistics*, 13(5), 2021.

Appendix A

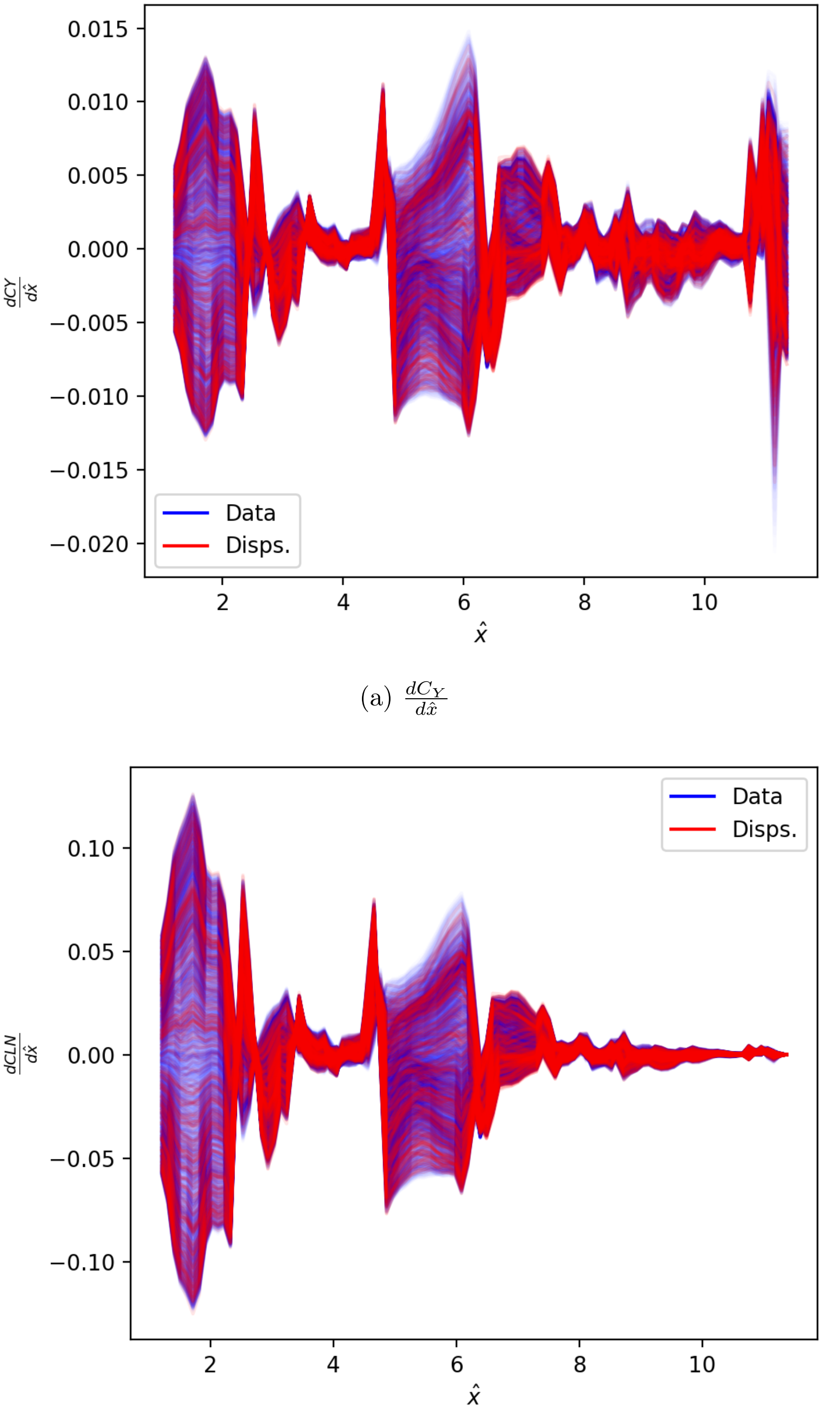
Appendix



(a) $\frac{dC_A}{d\hat{x}}$

(b) $\frac{dC_\ell}{d\hat{x}}$

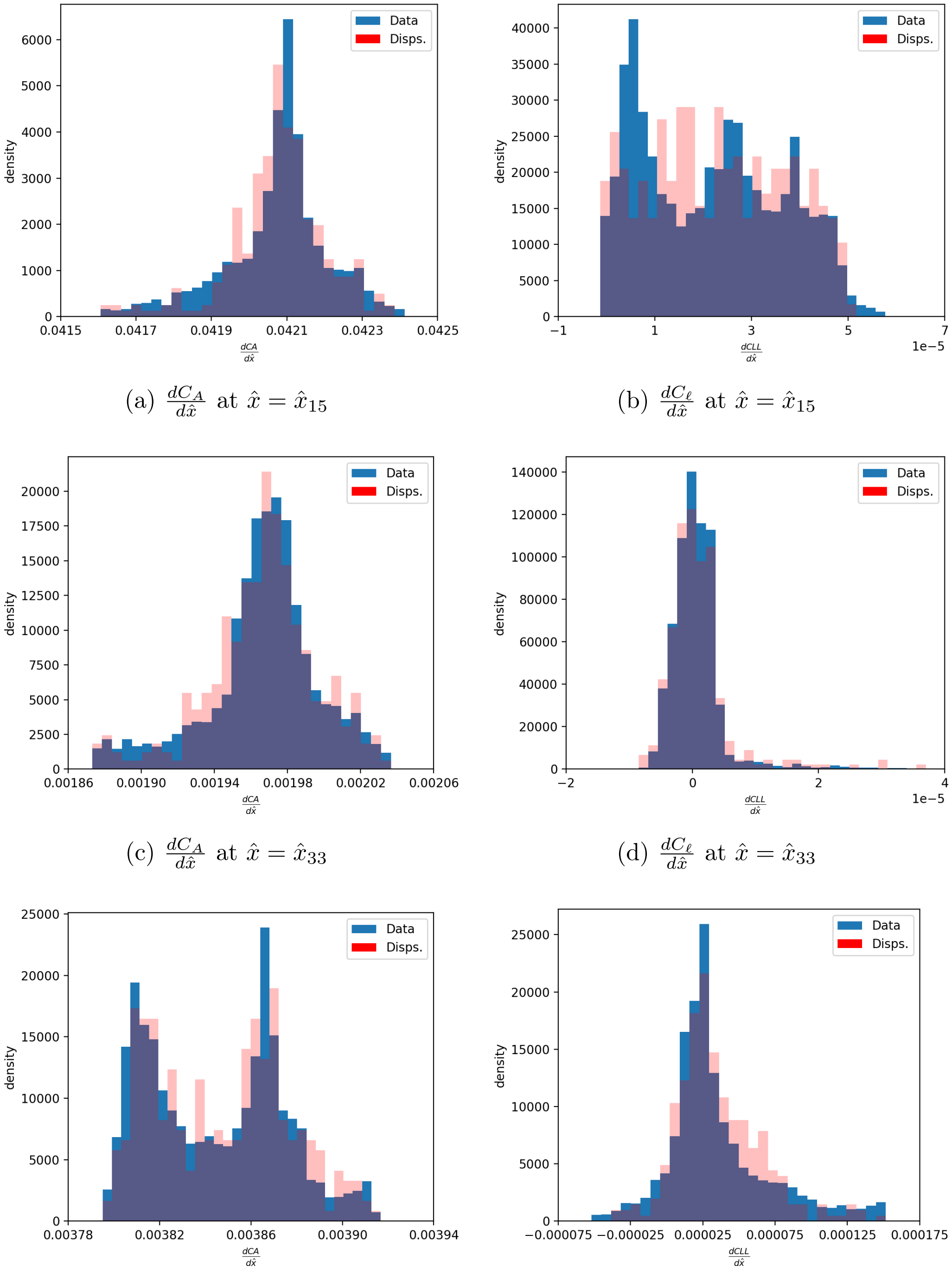
Figure A.1: Re-sample generated $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ for 300 C_A & C_ℓ taken from dataset.



(a) $\frac{dC_Y}{d\hat{x}}$

(b) $\frac{dC_n}{d\hat{x}}$

Figure A.2: Re-sample generated $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ for 300 C_Y & C_n taken from dataset.



(a) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

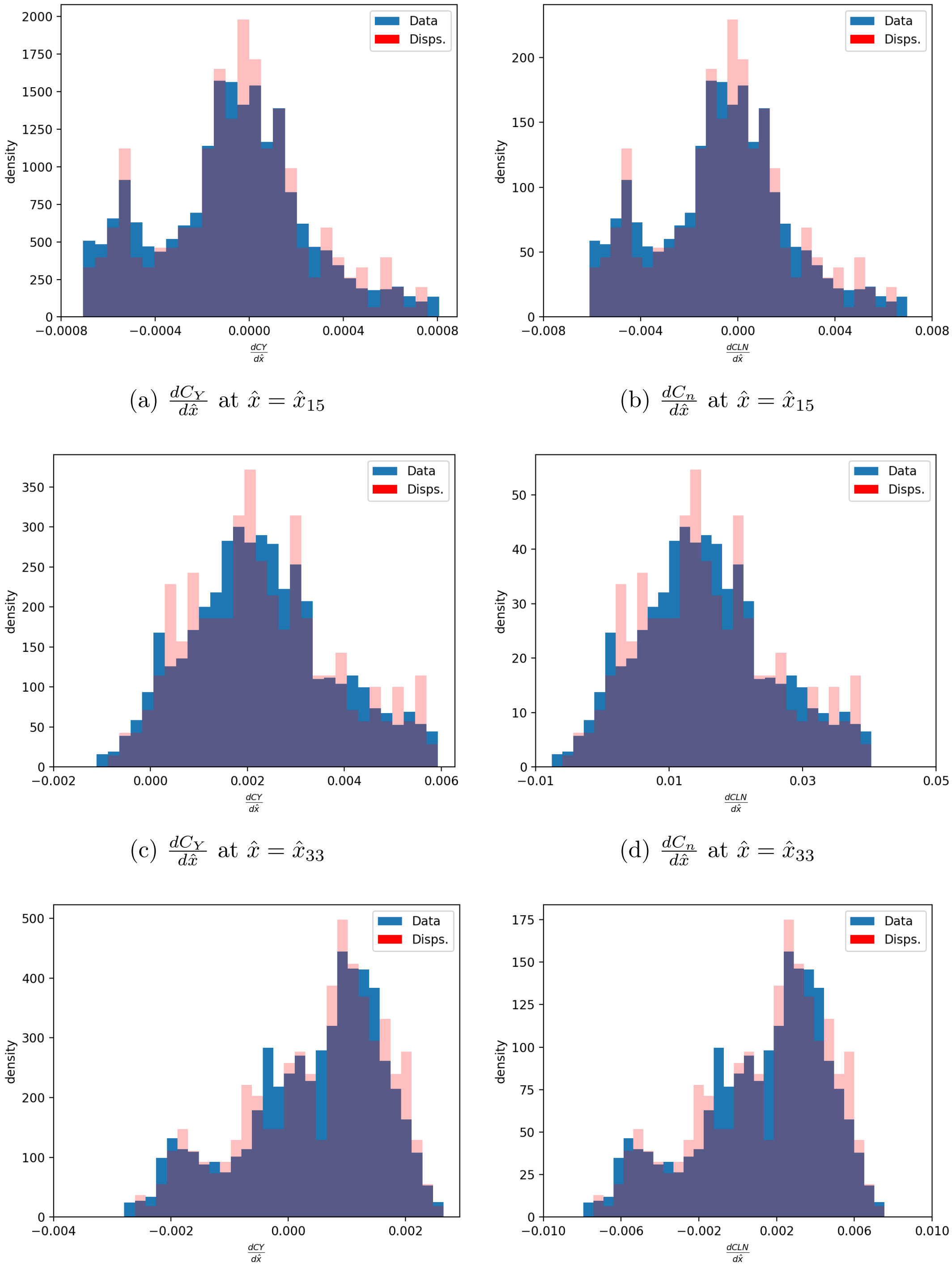
(c) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure A.3: Re-sample $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ histograms for 300 C_A & C_ℓ taken from dataset.



(a) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

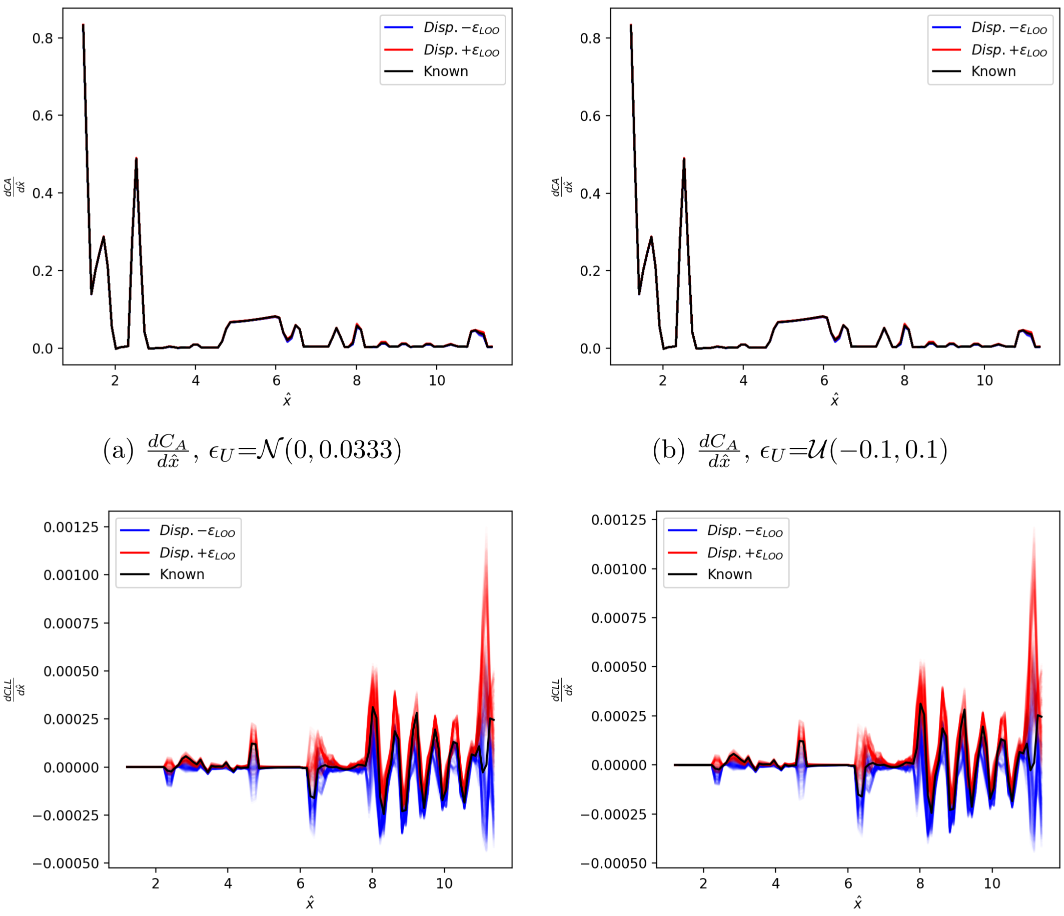
(c) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure A.4: Re-sample $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ histograms for 300 C_Y & C_n taken from dataset.



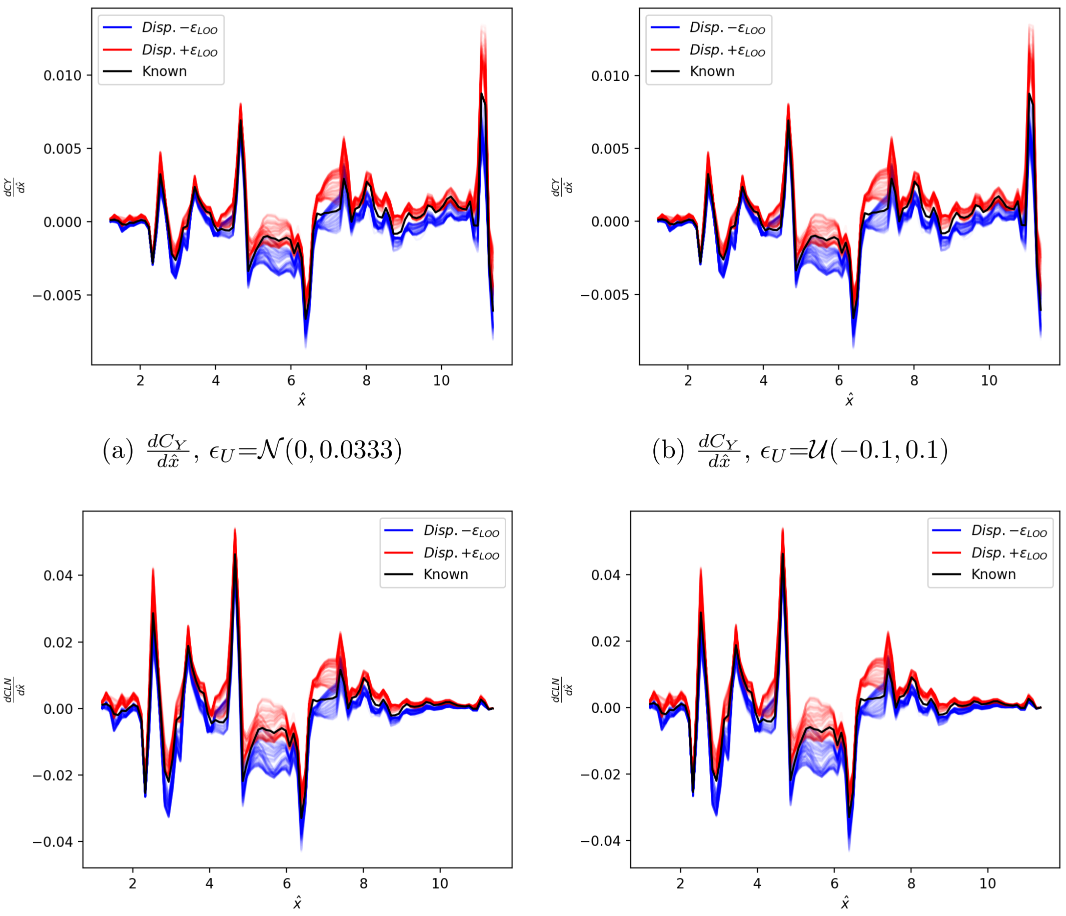
(a) $\frac{dC_A}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(b) $\frac{dC_A}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

(c) $\frac{dC_\ell}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(d) $\frac{dC_\ell}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

Figure A.5: Resampling $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ simulated uncertainty dispersions for 300 C_A & C_ℓ



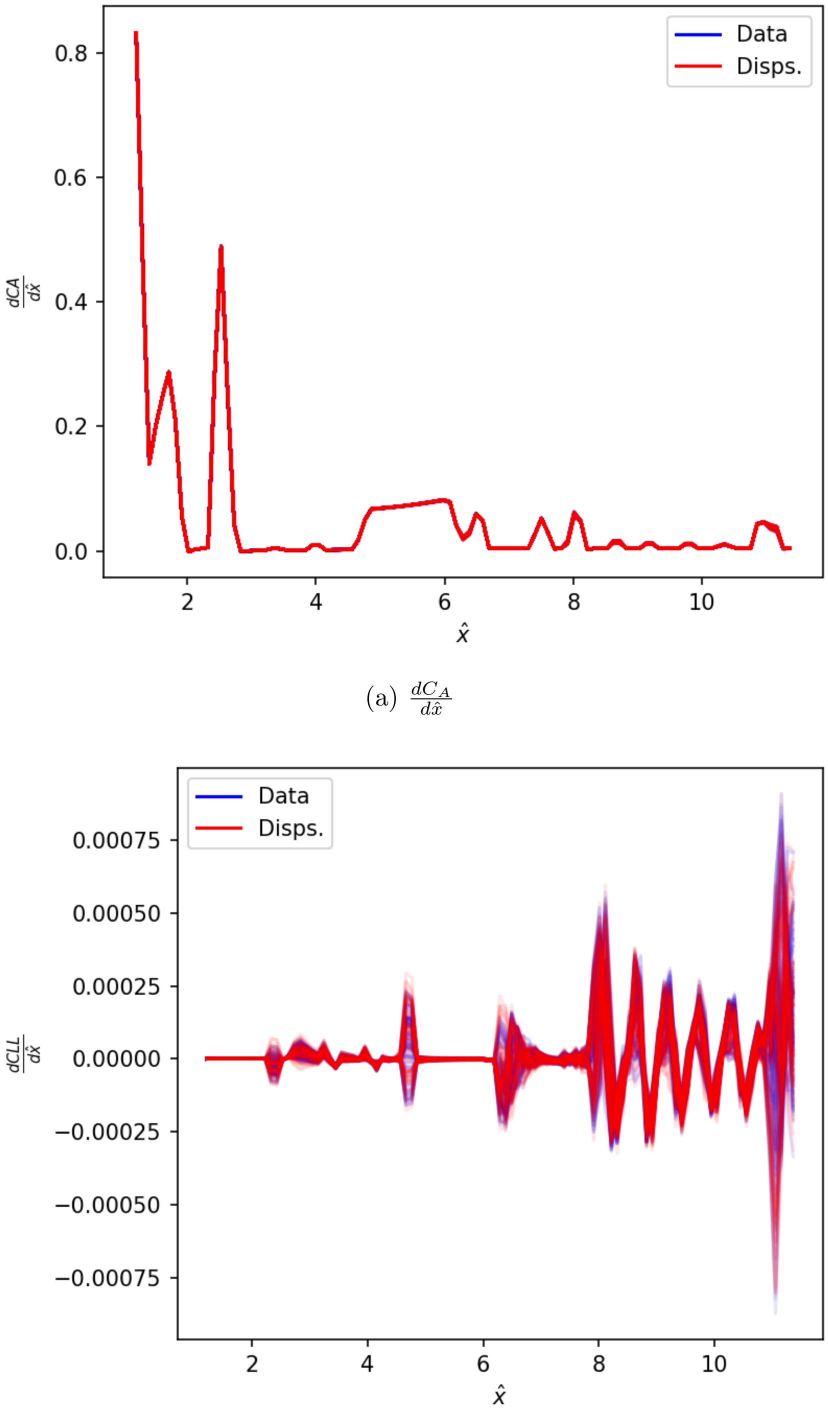
(a) $\frac{dC_Y}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(b) $\frac{dC_Y}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

(c) $\frac{dC_n}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(d) $\frac{dC_n}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

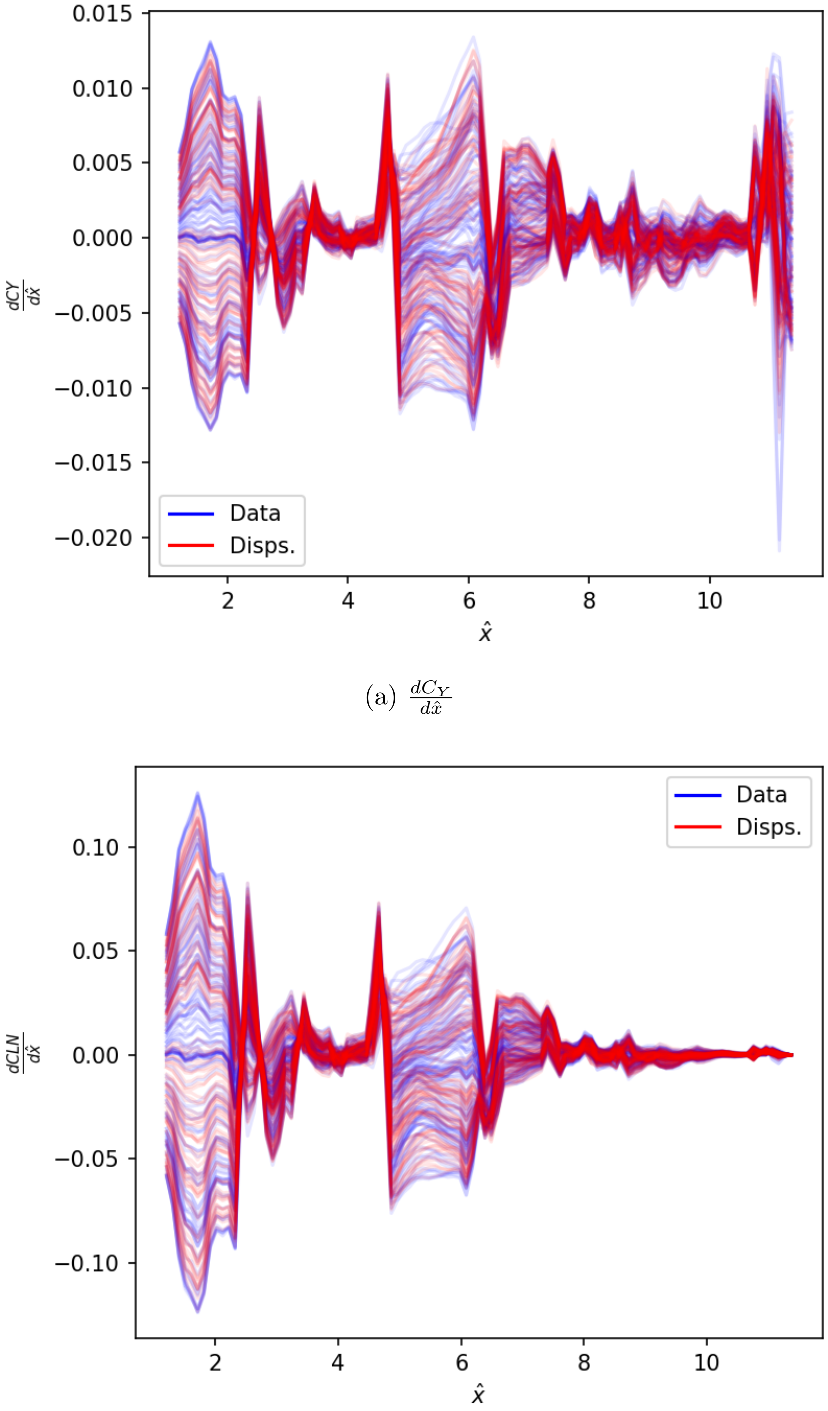
Figure A.6: Resampling $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ simulated uncertainty dispersions for 300 C_Y & C_n



(a) $\frac{dC_A}{d\hat{x}}$

(b) $\frac{dC_\ell}{d\hat{x}}$

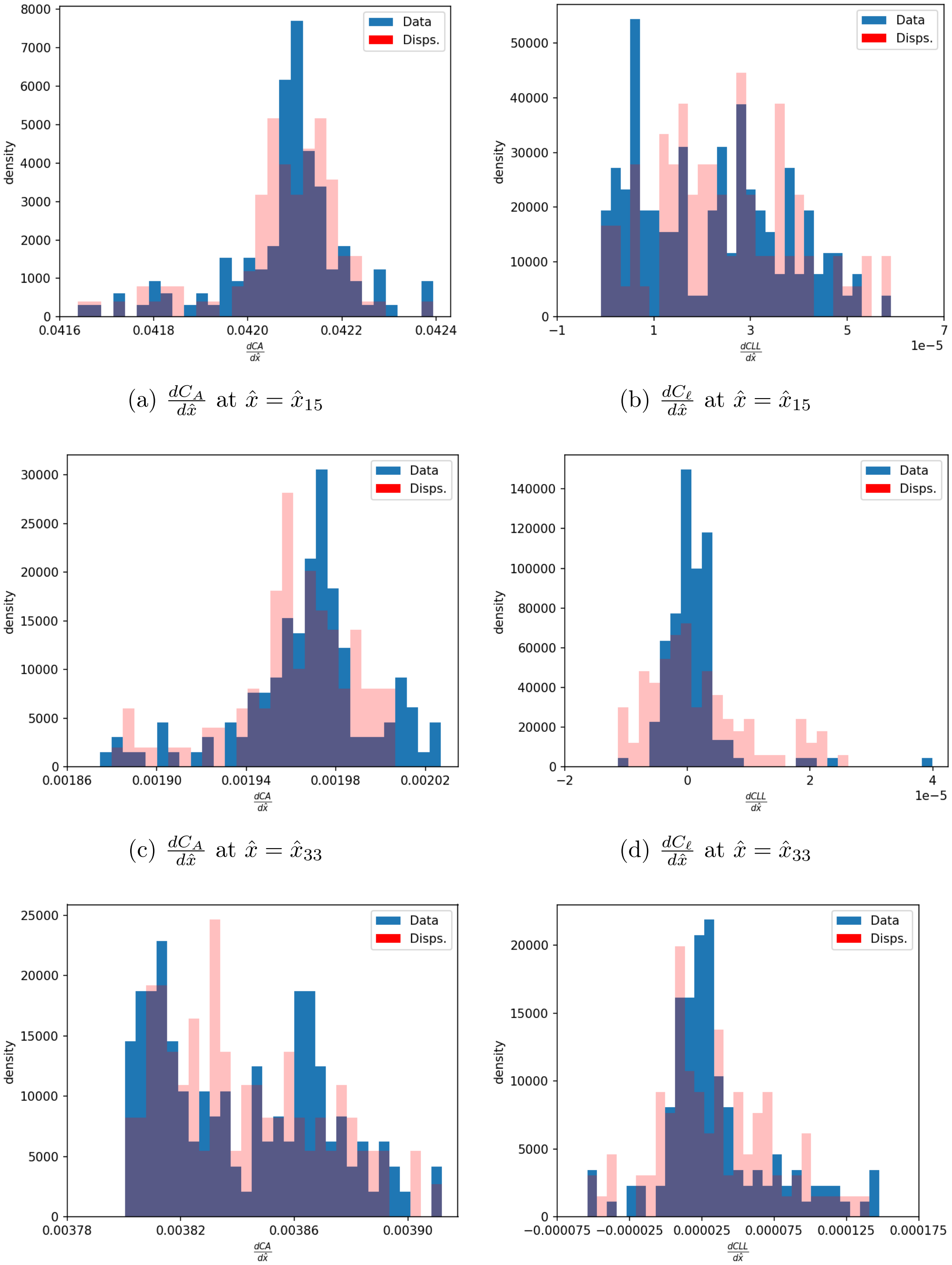
Figure A.7: MVN-Copula generated $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ for 100 C_A & C_ℓ taken from dataset.



(a) $\frac{dC_Y}{d\hat{x}}$

(b) $\frac{dC_n}{d\hat{x}}$

Figure A.8: MVN-Copula generated $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ for 100 C_Y & C_n taken from dataset.



(a) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

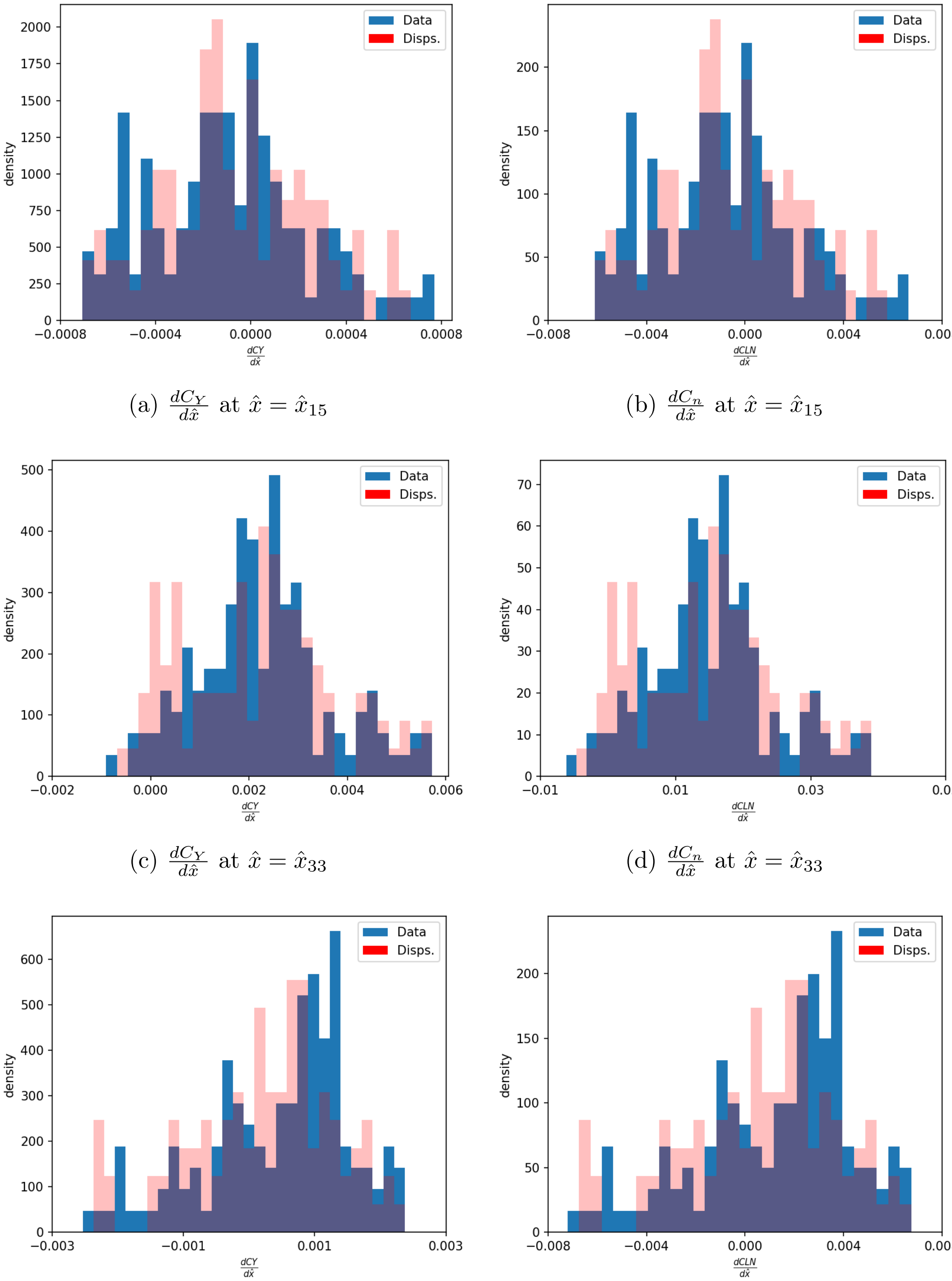
(c) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure A.9: MVN-Copula $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ histograms for 100 C_A & C_ℓ taken from dataset.



(a) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

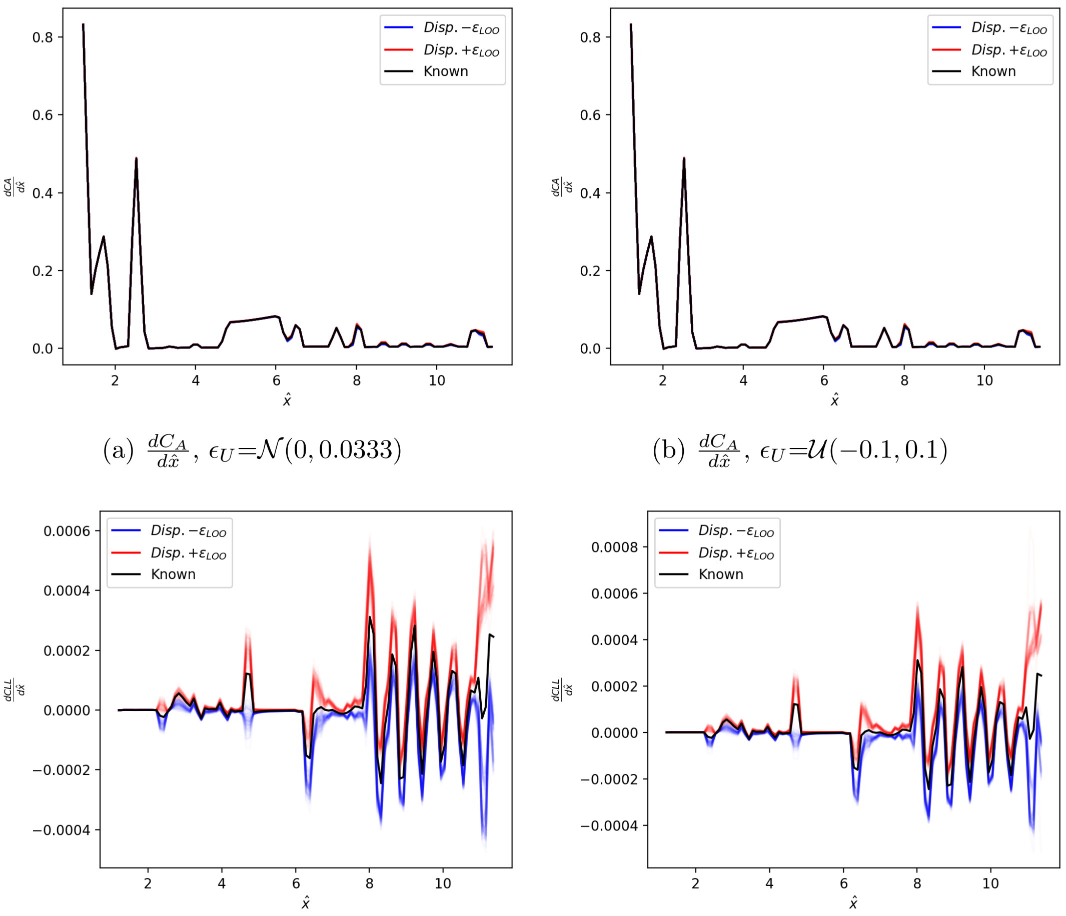
(c) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure A.10: MVN-Copula $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ histograms for 100 C_Y & C_n taken from dataset.



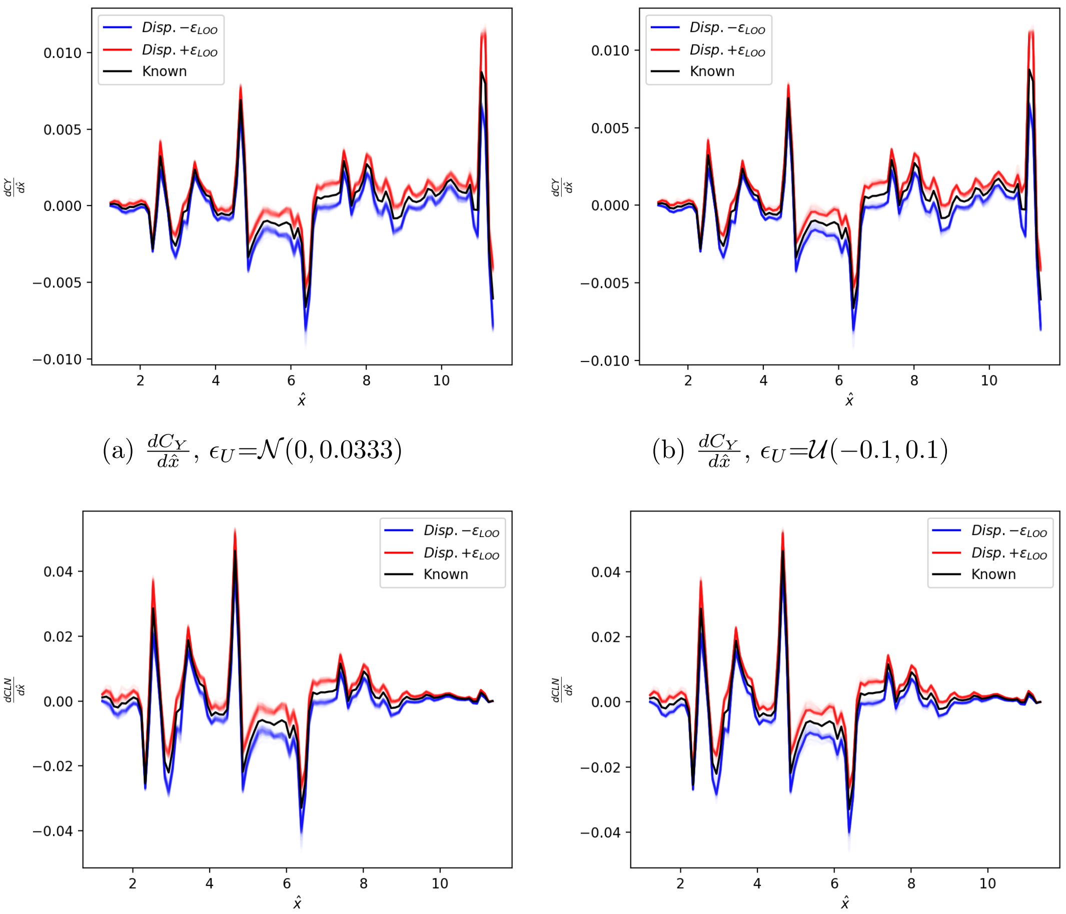
(a) $\frac{dC_A}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(b) $\frac{dC_A}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

(c) $\frac{dC_\ell}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(d) $\frac{dC_\ell}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

Figure A.11: MVN-Copula $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ simulated uncertainty dispersions for 100 C_A & C_ℓ



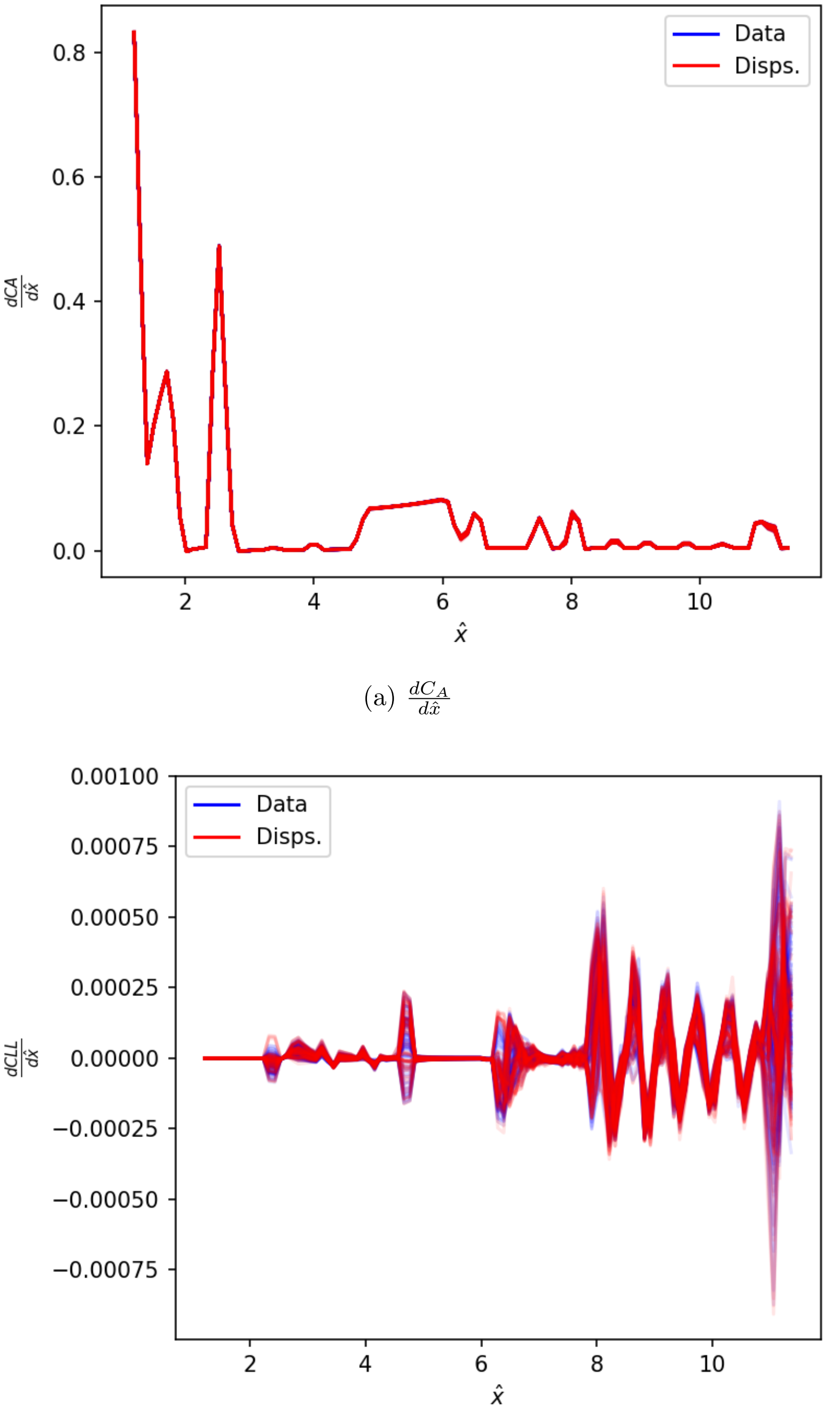
(a) $\frac{dC_Y}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(b) $\frac{dC_Y}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

(c) $\frac{dC_n}{d\hat{x}}$, ϵ_U=$\mathcal{N}(0, 0.0333)$

(d) $\frac{dC_n}{d\hat{x}}$, ϵ_U=$\mathcal{U}(-0.1, 0.1)$

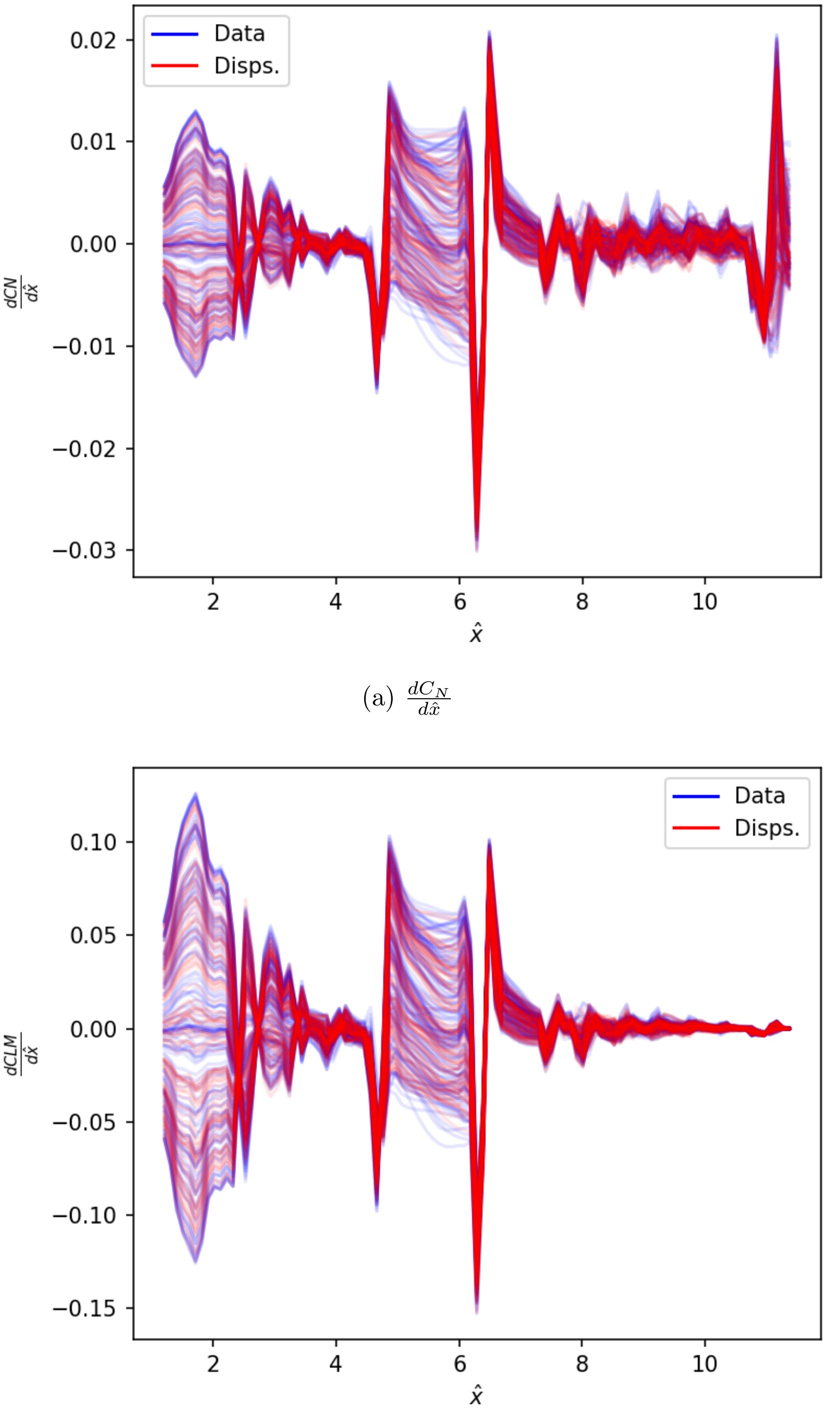
Figure A.12: MVN-Copula $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ simulated uncertainty dispersions for 100 C_Y & C_n



(a) $\frac{dC_A}{d\hat{x}}$

(b) $\frac{dC_\ell}{d\hat{x}}$

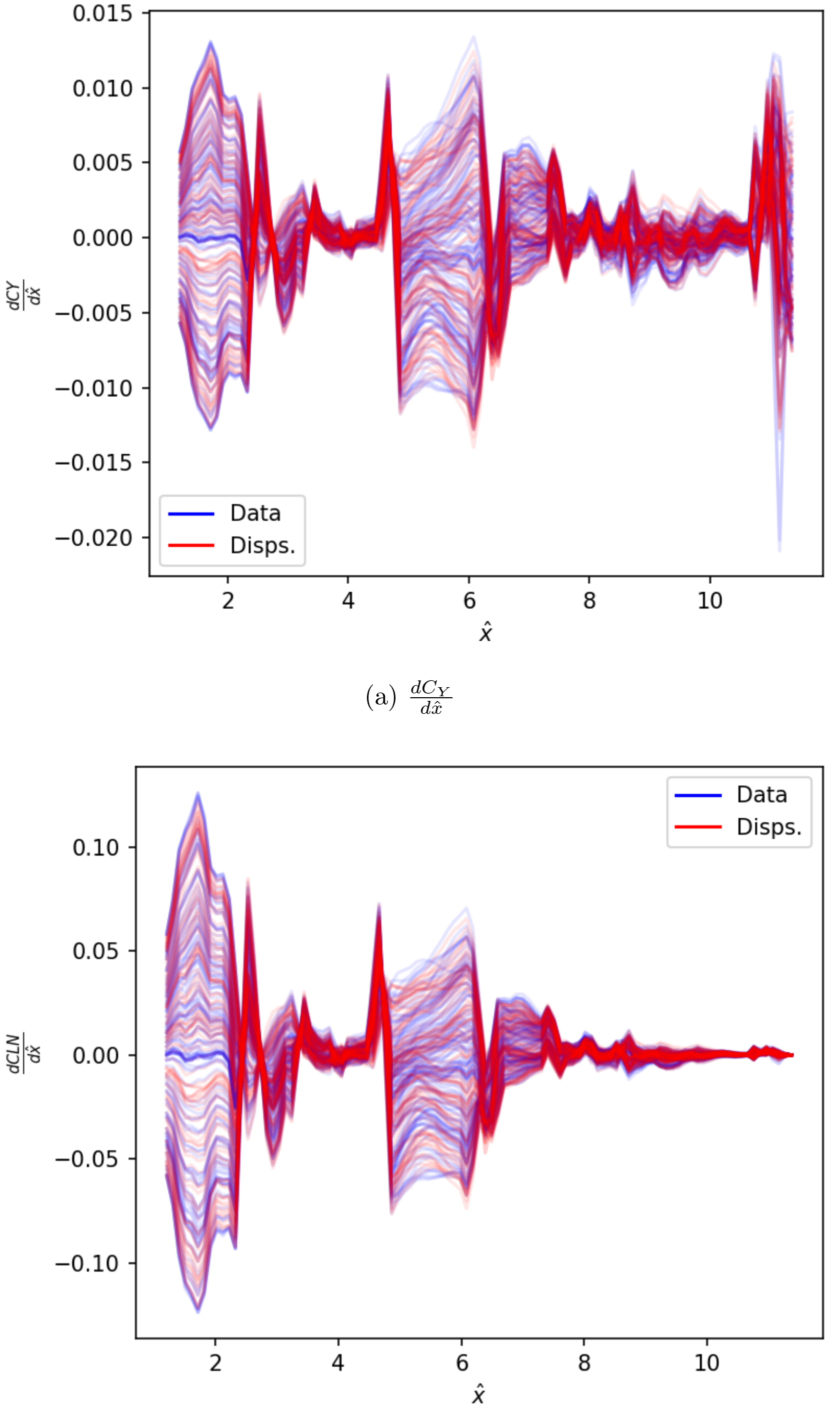
Figure A.13: BSP-Copula generated $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ for 100 C_A & C_ℓ taken from dataset.



(a) $\frac{dC_N}{d\hat{x}}$

(b) $\frac{dC_m}{d\hat{x}}$

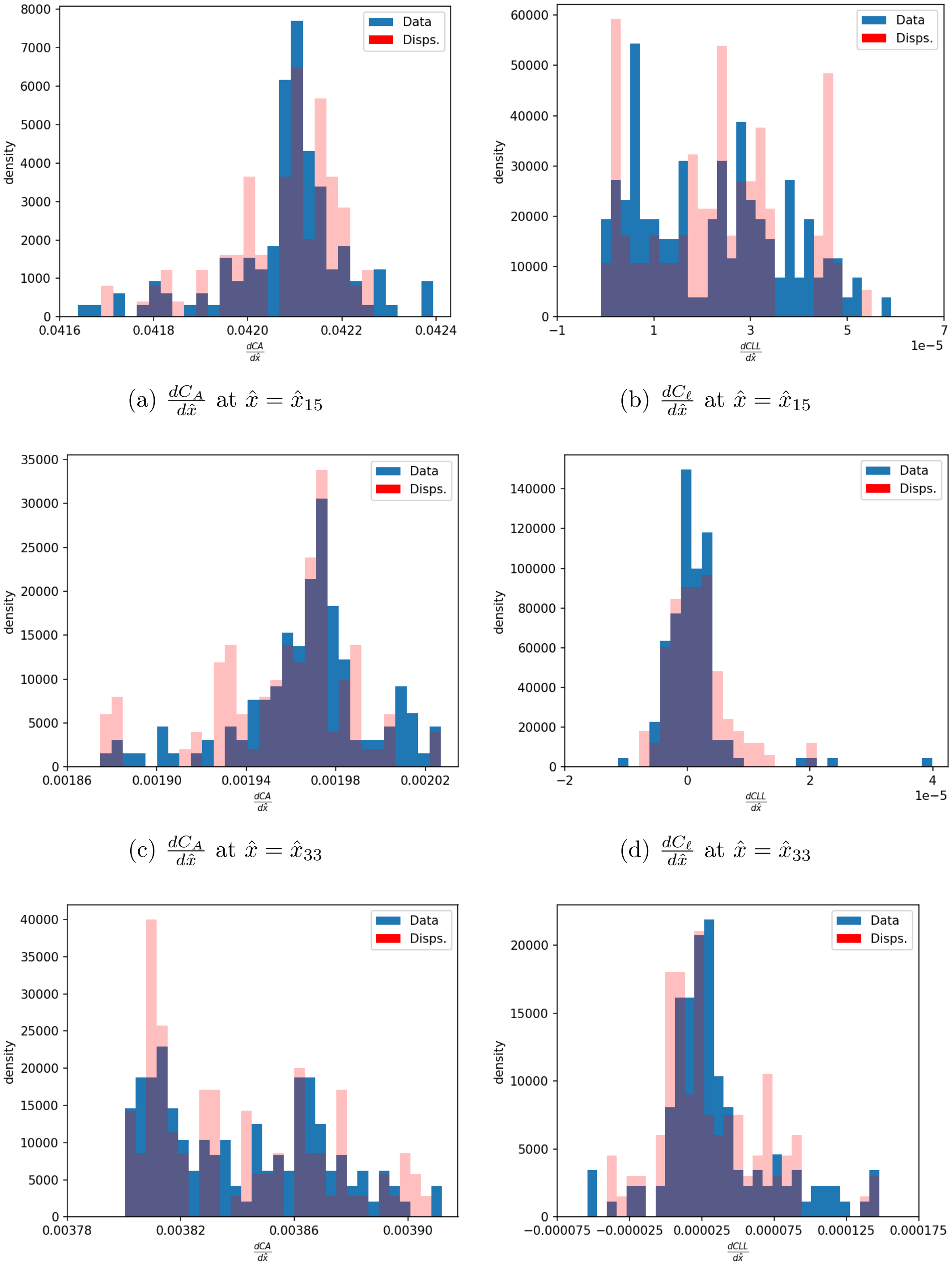
Figure A.14: BSP-Copula generated $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ for 100 C_N & C_m taken from dataset.



(a) $\frac{dC_Y}{d\hat{x}}$

(b) $\frac{dC_n}{d\hat{x}}$

Figure A.15: BSP-Copula generated $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ for 100 C_Y & C_n taken from dataset.



(a) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

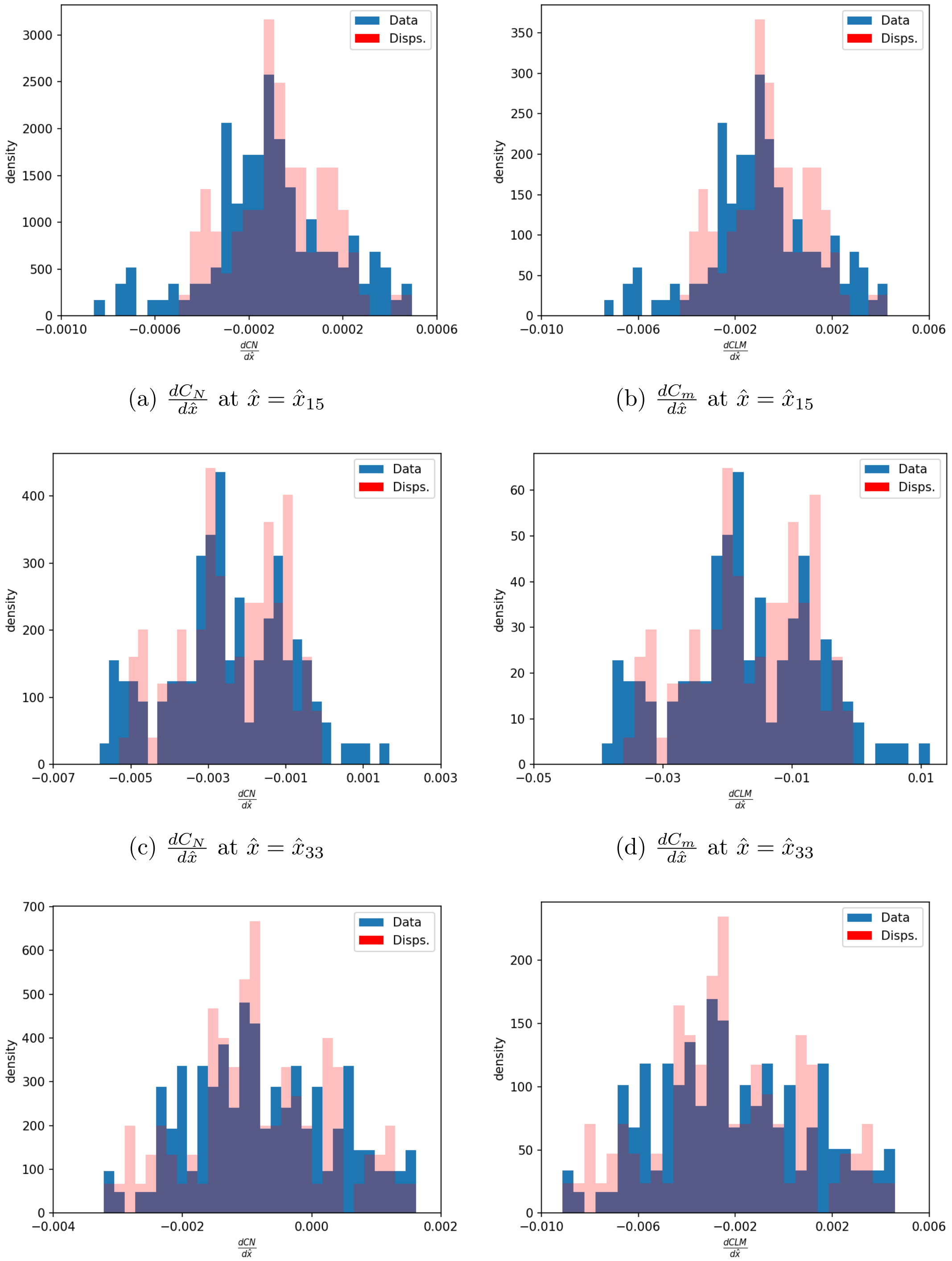
(c) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_A}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_\ell}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure A.16: BSP-Copula $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ histograms for 100 C_A & C_ℓ taken from dataset.



(a) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

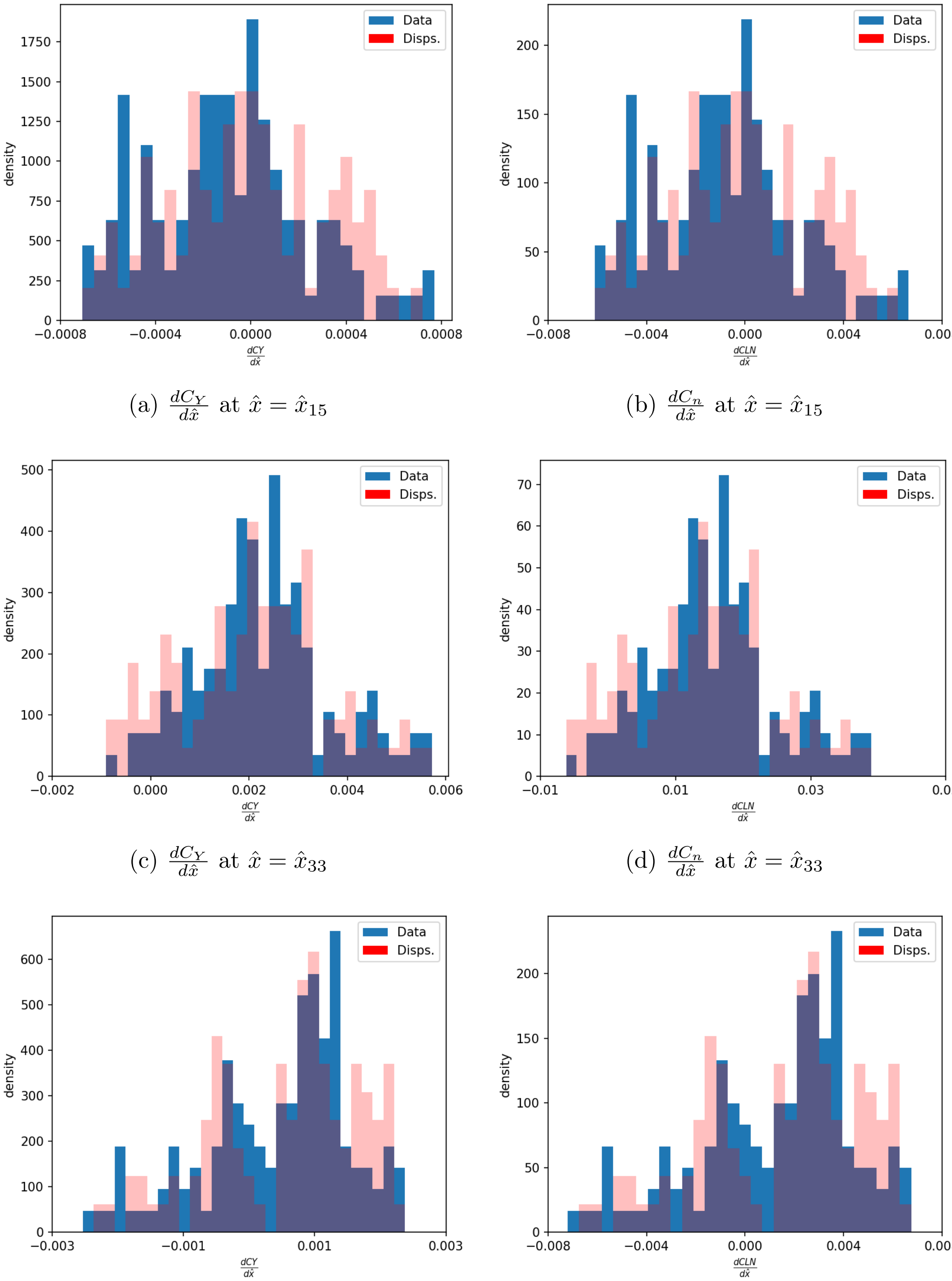
(c) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_N}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_m}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure A.17: BSP-Copula $\frac{dC_N}{d\hat{x}}$ and $\frac{dC_m}{d\hat{x}}$ histograms for 100 C_N & C_m taken from dataset.



(a) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

(b) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{15}$

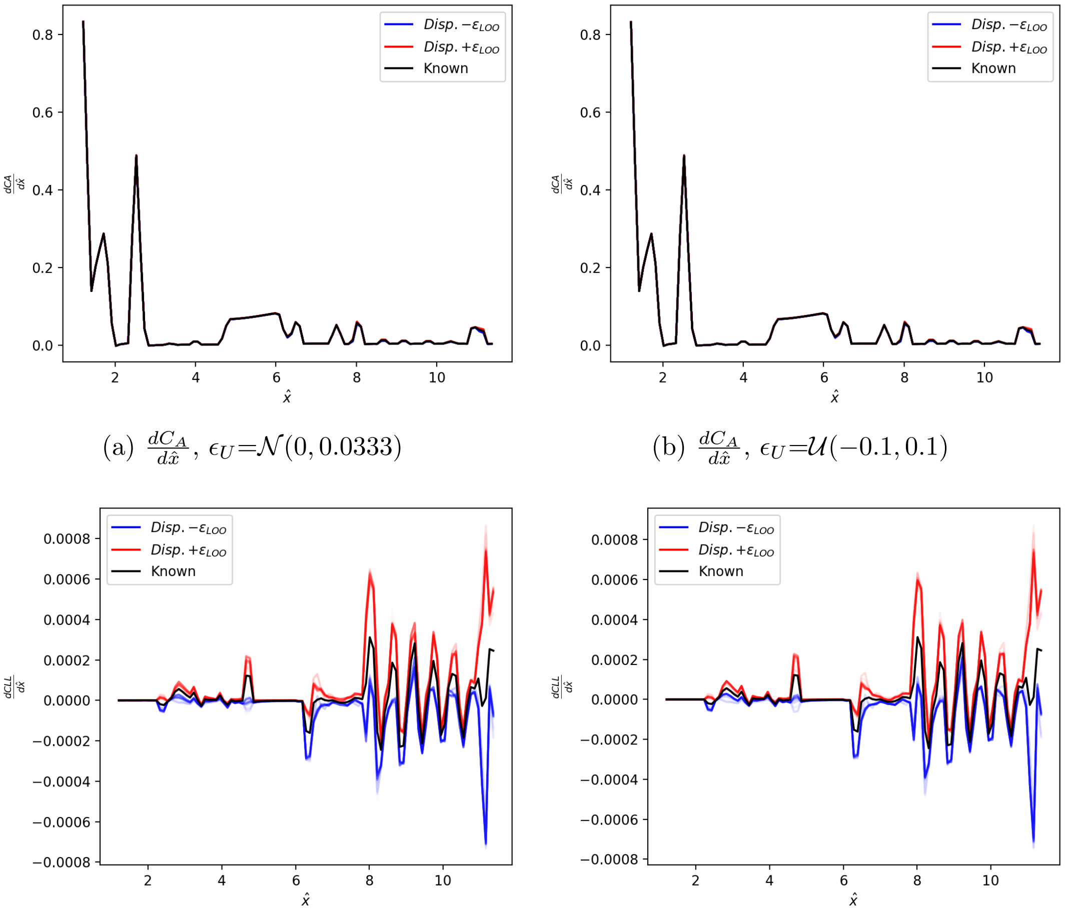
(c) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(d) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{33}$

(e) $\frac{dC_Y}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

(f) $\frac{dC_n}{d\hat{x}}$ at $\hat{x} = \hat{x}_{72}$

Figure A.18: BSP-Copula $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ histograms for 100 C_Y & C_n taken from dataset.



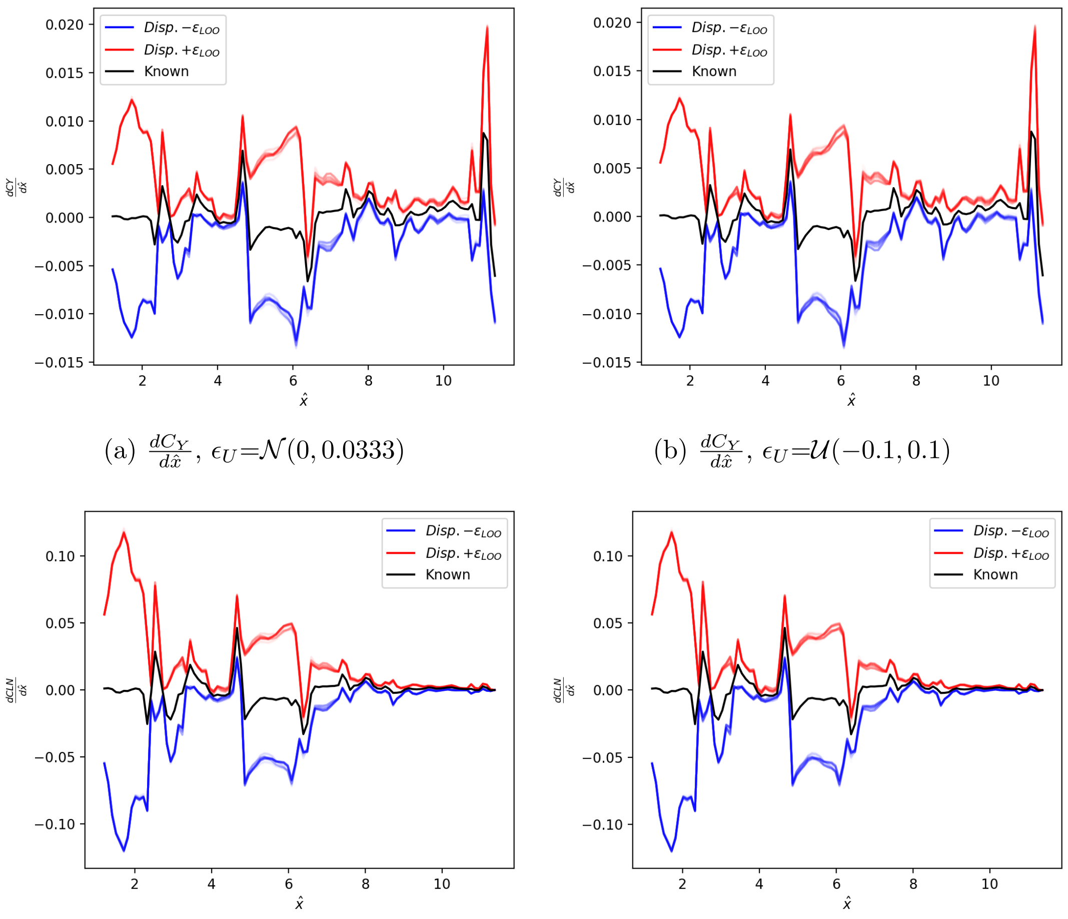
(a) $\frac{dC_A}{d\hat{x}}$, $\epsilon_U = \mathcal{N}(0, 0.0333)$

(b) $\frac{dC_A}{d\hat{x}}$, $\epsilon_U = \mathcal{U}(-0.1, 0.1)$

(c) $\frac{dC_\ell}{d\hat{x}}$, $\epsilon_U = \mathcal{N}(0, 0.0333)$

(d) $\frac{dC_\ell}{d\hat{x}}$, $\epsilon_U = \mathcal{U}(-0.1, 0.1)$

Figure A.19: BSP-Copula $\frac{dC_A}{d\hat{x}}$ and $\frac{dC_\ell}{d\hat{x}}$ simulated uncertainty dispersions for 100 C_A & C_ℓ



(a) $\frac{dC_Y}{d\hat{x}}$, $\epsilon_U = \mathcal{N}(0, 0.0333)$

(b) $\frac{dC_Y}{d\hat{x}}$, $\epsilon_U = \mathcal{U}(-0.1, 0.1)$

(c) $\frac{dC_n}{d\hat{x}}$, $\epsilon_U = \mathcal{N}(0, 0.0333)$

(d) $\frac{dC_n}{d\hat{x}}$, $\epsilon_U = \mathcal{U}(-0.1, 0.1)$

Figure A.20: BSP-Copula $\frac{dC_Y}{d\hat{x}}$ and $\frac{dC_n}{d\hat{x}}$ simulated uncertainty dispersions for 100 C_Y & C_n