Bootstrap-Based Robustness Analysis of Parameter Optimization in Climate Models Using QuadTune

by

Luis Hasenauer

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Mathematics

at

The University of Wisconsin–Milwaukee

May 2025

ABSTRACT

Bootstrap-Based Robustness Analysis of Parameter Optimization in Climate Models Using QuadTune

by

Luis Hasenauer

The University of Wisconsin–Milwaukee, 2025
Under the Supervision of Professor Vince Larson

Tuning the parameters of climate models is essential for improving their performance, but this process is often complicated by structural limitations, overfitting, and trade-offs between different regions or variables. This thesis combines the QuadTune optimization framework with nonparametric bootstrap resampling to analyze parameter uncertainty and identify tuning conflicts.

Bootstrap replicates of input metrics are used to generate empirical distributions of parameter estimates and construct nonparametric confidence intervals. Residuals from bootstrap-tuned parameters are compared with default and full-dataset residuals to assess spatial bias and model robustness.

A residual-based diagnostic method is introduced to detect tuning trade-offs by evaluating the minimum ℓ_2 norm and correlation of residual pairs, resulting in a binary conflict map that highlights jointly untunable metrics.

The analysis focuses on the `SWCF` (shortwave cloud forcing) variable using 10,000 bootstrap samples over a $20^\circ \times 20^\circ$ spatial grid. Results show that while QuadTune improves global performance, regional tuning conflicts and structural biases remain, highlighting limitations that cannot be addressed by parameter tuning alone.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BC_a Bias-Corrected and Accelerated. 5, 13, 14, 16, 32

MSR Mean Squared Residuals. 21, 22

1 Introduction

Climate models are essential tools for understanding the Earth's climate system and predicting future climate behavior. These models simulate complex physical processes and interactions within the atmosphere, oceans, land surface, and cryosphere. However, due to the complexity of these systems, climate models inevitably contain both *parametric errors*, inaccuracies in the numerical values of tunable parameters, and *structural errors*, which arise from limitations in the functional form or underlying assumptions of the model itself.

Tuning model parameters is a standard approach used to mitigate parametric error and improve model performance. Nevertheless, parameter tuning is computationally expensive, as it typically requires a large number of global simulations to explore parameter sensitivities and identify optimal parameter values. Moreover, parameter tuning alone cannot correct for structural errors, which often manifest as persistent regional biases or systematic deviations in model output.

1.1 Motivation

The introduction of new parameterizations or structural improvements in climate models frequently leads to degradation in model performance, necessitating re-tuning of parameters. Manual tuning is labor-intensive, and automated methods such as perturbed parameter ensembles or sequential optimization approaches can be prohibitively expensive in terms of computational resources [1].

QuadTune addresses this challenge by offering a fast and efficient "regional tuner" for global

atmospheric models. It approximates the parameter dependence of model metrics using a quadratic emulator and requires only a small number of global simulations. By dividing the globe into coarse regions and tuning parameters in a least-squares sense, QuadTune not only reduces computational cost but also provides diagnostic plots that help reveal persistent biases and tuning trade-offs.

Because QuadTune drastically reduces the number of simulations required per tuning run, it becomes feasible to apply computationally intensive uncertainty quantification methods such as bootstrapping, which would otherwise be impractical in conventional tuning workflows. This enables us to evaluate the stability of parameter estimates under resampling and gain deeper insight into model robustness and potential structural limitations.

1.2 Objectives

The goal of this thesis is to assess the robustness and uncertainty of parameter tuning in global climate models using the QuadTune framework in combination with a nonparametric bootstrap approach. The focus is not only on estimating parameter variability, but also on analyzing residual patterns, tuning trade-offs, and limitations imposed by model structure. More specifically, this thesis aims to:

- Evaluate the variability of optimized parameters by applying a bootstrap resampling approach to the input metrics, enabling nonparametric confidence interval construction.

- Compare the bootstrap-based parameter distributions with both default parameter settings and the estimates obtained from the full dataset to assess tuning stability.

- Analyze the distribution and variance of residuals to assess model performance across

different spatial regions and identify areas where tuning cannot sufficiently reduce biases.

- Classify and interpret different types of tuning interactions, and relate them to model structure and regional sensitivity.

- Develop a matrix-based method to detect tuning trade-offs between metrics by identifying metric pairs whose residuals cannot be minimized simultaneously.

Through this investigation, the thesis provides a detailed framework for quantifying parameter uncertainty and understanding structural tuning limitations in climate models. It also demonstrates how computational advances in tuning (via QuadTune) make such high-resolution robustness analysis practically feasible.

1.3 The Bootstrapping Approach

In order to assess the robustness of the parameter estimates produced by QuadTune, we employ a bootstrapping approach. Bootstrapping is a resampling technique that allows for the estimation of variability in model parameters by generating multiple datasets derived from the original input metrics. This method helps evaluate the sensitivity of the optimized parameters to variations in the input data and provides insights into potential overfitting and structural errors within the model.

Bootstrapping is particularly useful in cases where the sample size is limited or when quantifying uncertainty in model outputs. Since QuadTune optimizes parameters based on a given set of input metrics, applying bootstrapping allows us to explore how different subsets of the data influence the final parameter values. By comparing the parameter distributions obtained from bootstrapped samples with those optimized on the full dataset, we can assess

the stability of the tuning process.

The bootstrapping process is conducted as follows:

1. **Resampling Input Metrics:** Given the original dataset of input metrics, we generate B bootstrap samples by randomly sampling with replacement. Each bootstrap sample has the same size, N, as the original dataset.

2. **Parameter Estimation with QuadTune:** For each bootstrap sample, we apply QuadTune to optimize the parameters $\theta \in \mathbb{R}^P$ of the climate model. The model output metrics are approximated by a diagonal-quadratic emulator derived from a second-order Taylor expansion around the default parameter vector $\theta_{\text{def}} \in \mathbb{R}^P$. The i-th metric is modeled as:

$$m_i(\theta) = m_i(\theta_{\text{def}}) + \sum_{j=1}^{P} \frac{\partial m_i}{\partial \theta_j} \delta\theta_j + \frac{1}{2} \sum_{j=1}^{P} \frac{\partial^2 m_i}{\partial \theta_j^2} (\delta\theta_j)^2 + \epsilon_{e,i},$$

   where $\delta\theta_j = \theta_j - \theta_{j,\text{def}}$ and $\epsilon_{e,i}$ represents approximation error due to higher-order terms.

   The tuning process minimizes the following weighted least-squares loss:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \sigma_i^2 \left[-\delta b_i - \sum_{j=1}^{P} \left(\frac{\partial m_i}{\partial \theta_j} \delta\theta_j + \frac{1}{2} \frac{\partial^2 m_i}{\partial \theta_j^2} (\delta\theta_j)^2 \right) \right]^2,$$

   where $\delta b_i = m_i(\theta_{\text{def}}) - f_i^{\text{obs}}$ is the initial model bias and σ_i are weights reflecting regional area.

3. **Comparison with Full Dataset Results:** The resulting bootstrap parameter estimates $\hat{\theta}^{*(b)}$ are compared with:

   - The default parameter vector θ_{def},

   - The parameter vector $\hat{\theta}$ optimized using the full dataset.

4. **Uncertainty Quantification:** From the distribution $\{\hat{\theta}^{*(b)}\}_{b=1}^{B}$, we compute parameter-wise variability and construct confidence intervals using both the percentile method and the Bias-Corrected and Accelerated ($\mathrm{BC_a}$) bootstrap technique.

5. **Residual Analysis:** For each bootstrap-tuned parameter vector $\hat{\theta}^{*(b)}$, we compute the residuals for each metric, $r_i^{(b)}$, using the model estimate:

$$\hat{m}_i\left(\hat{\theta}^{*(b)}\right) := m_i\left(\hat{\theta}^{*(b)}\right) - \epsilon_{e,i}, \quad \text{and} \quad r_i^{(b)} = f_i^{\text{obs}} - \hat{m}_i^{(b)}.$$

   These residuals form an empirical distribution for each metric across bootstrap samples. We analyze the spread and shape of these distributions to identify regional biases, stubborn residuals, and potential trade-offs between metrics.

This procedure allows us to probe the sensitivity of the optimized parameters to changes in the data and assess tuning stability. Additionally, it helps reveal structural limitations in the model, as well as potential overfitting that may arise when tuning to specific realizations of the input metrics.

Throughout this thesis, we focus on the variable `SWCF` (shortwave cloud forcing) and conduct all bootstrap-based analysis using 10,000 bootstrap samples, each of size 162. This sample size corresponds to the total number of spatial regions in the $20^\circ \times 20^\circ$ grid that was used to partition the globe. Each metric in the `SWCF` dataset is indexed using the naming convention `SWCF_r_c`, where r denotes the row (latitude band) and c the column (longitude band) in a global $20^\circ \times 20^\circ$ spatial grid. The grid is structured in row-major order, meaning rows increment from north to south, and columns increment from west to east within each row. For example, `SWCF_1_1` corresponds to the top-left grid cell (northwest corner of the globe), while `SWCF_9_18` refers to the bottom-right cell (southeast corner).

2 Parameter Distributions

2.1 Analysis of Parameter Distributions

One of the main objectives of this study is to analyze how the parameters optimized by QuadTune vary across different bootstrap samples. By examining the distribution of these parameters, we can assess the stability of the optimization process and compare the results against both the default parameter values and those obtained from the full dataset.

Comparison with Default and Full Dataset Parameters

To evaluate the impact of bootstrapping on parameter estimation, we compare the distributions of the optimized parameters from bootstrap samples against:

- The **default parameter values** before any optimization, providing a baseline for comparison.
- The **parameters optimized using the full dataset**, which represent the standard tuning result without bootstrapping.

These comparisons allow us to determine whether the parameters derived from bootstrapped samples align closely with the full dataset estimates or exhibit significant variability. Large deviations between bootstrap-derived parameters and full dataset parameters may indicate sensitivity to input metrics, potential structural errors, or overfitting in the tuning process.

Statistical Analysis of Parameter Distributions

To further explore the variability in parameter estimates, we analyze the shape and spread of the distributions using statistical metrics and visualization techniques:

- **Kernel Density Estimation (KDE)** to visualize the probability density of the parameters.
- **Histograms** to observe the range and dispersion of the estimated parameters.
- **Mean and Standard Deviation** to quantify central tendency and variability.

By comparing these distributions, we can assess whether the optimization process produces stable parameter estimates or if there is high variability due to sensitivity to different input samples.

Interpretation and Insights

From these analyses, we aim to answer the following key questions:

- How much do the optimized parameters fluctuate across bootstrap samples?
- Do the bootstrapped parameter distributions overlap with those obtained from the full dataset?
- Are there any systematic biases or trends in the parameter distributions that suggest overfitting or structural error?

The results from this section will provide crucial insights into the robustness of the tuning process and the reliability of the optimized parameters in climate model calibration.

Figure 2.1 shows the histogram and Gaussian kernel density estimates [2] of the parameters fitted to the bootstrapped input metrics. The red dashed line indicates the default parameter values used in the model prior to tuning and the green dashed line shows the optimized parameter values obtained by tuning on the dataset.

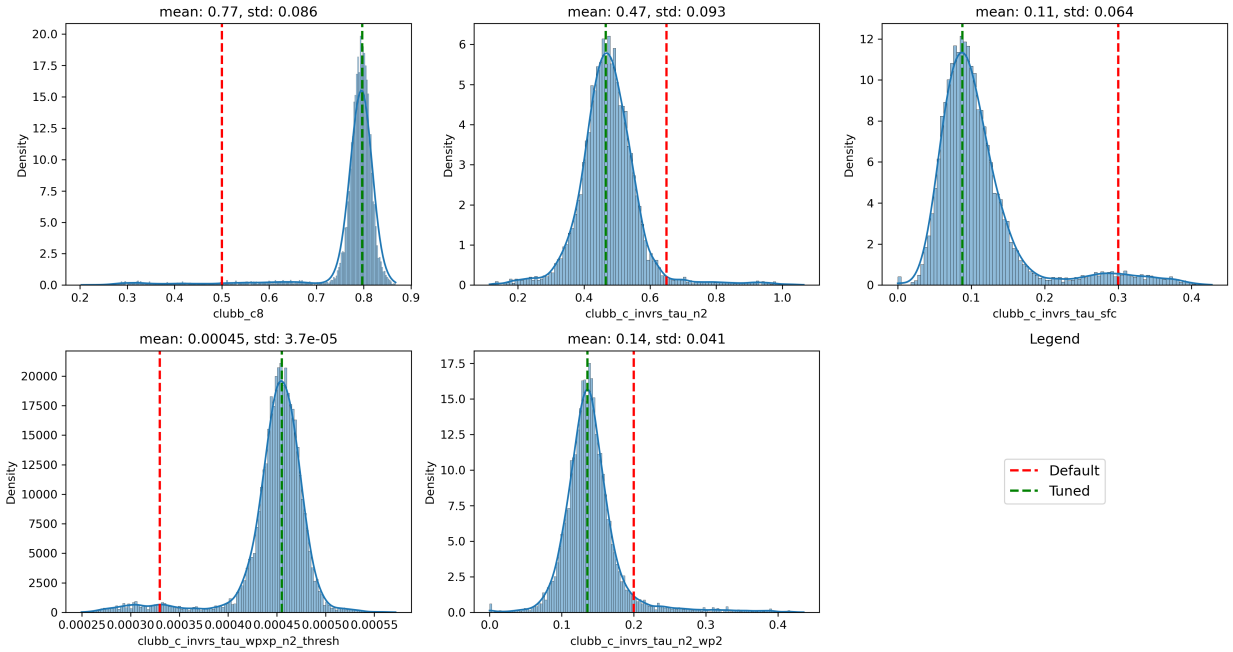

Figure 2.1: Histograms and kernel density estimations of parameters fitted to the bootstrapped metrics.

Most parameters show unimodal distributions centered around the full dataset estimate. This suggests that the tuning process is stable and the parameter estimates are consistent across different samples.

However, some parameters show wider spread, heavy tails, and skewed distributions. This indicates that the optimization process is sensitive to the input metrics. The optimization process may strongly depend on specific regions and leaving out these regions results in substantial changes in the estimated parameter values.

When comparing the parameter distributions to the default values, several parameters show substantial differences from the default values. This suggests that the default values are

biased and the tuned parameters improve the model fit.

To analyze the long tails of the parameter distributions, we can look at the conditional probabilities of a metric being excluded from a bootstrap sample, given that the parameter value lies in the tail. If we denote the set of bootstrapped metrics by M^* and the heavy tail of the distribution of $\hat{\theta}_j^*$ by Θ_j^{tail}, for each metric i we compute

$$\mathbb{P}(i \notin M^* \mid \hat{\theta}_j^* \in \Theta_j^{tail}) \approx \frac{1}{\sum_{b=1}^{B} 1_{\Theta_j^{tail}}(\hat{\theta}_j^{*(b)})} \sum_{b=1}^{B} (1 - 1_{M^{*(b)}}(i)) \cdot 1_{\Theta_j^{tail}}(\hat{\theta}_j^{*(b)}).$$

Here, $1_A(a)$ denotes an indicator function of a being contained in a set A.

If there is no dependence between the parameter value and tuning on a specific metric, this probability should be $(161/162)^{162} \approx 36.7\%$. This is because each metric is selected with probability $1/162$ and therefore the probability of not selecting a metric is $1 - 1/162 = 161/162$. Since the selection is repeated 162 times with replacement, the probability of a metric not being selected in a bootstrap sample of size 162 is $(161/162)^{162}$.

Table 2.1 shows the metrics with the highest probabilities of not being contained in the bootstrap sample if the tuned parameter value lies in the distributions' heavy tail. It can be observed that the parameters share similar metrics that seem to heavily influence the tuned parameter values. Especially `SWCF_6_18` and `SWCF_5_5` are contained in each of the top 5 metrics. Moreover, `SWCF_5_4` is also very prominent among all of the parameters. This induces that these metrics have the most influence on the tuned parameter values since excluding these regions results in strong deviations from the full data estimate and the majority of bootstrap estimates.

Parameter	Θ_j^{tail}	Metric	$\mathbb{P}(i \notin M^* \mid \hat{\theta}_j^* \in \Theta_j^{tail})$
θ_1 (c8)	$(-\infty, 0.7)$	SWCF_6_18	0.620
		SWCF_5_5	0.561
		SWCF_5_4	0.509
		SWCF_5_18	0.451
		SWCF_5_9	0.444
θ_2 (c_invrs_tau_n2)	$(0.7, \infty)$	SWCF_6_18	0.562
		SWCF_5_5	0.556
		SWCF_5_4	0.533
		SWCF_5_18	0.503
		SWCF_5_15	0.497
θ_3 (c_invrs_tau_sfc)	$(0.2, \infty)$	SWCF_6_18	0.641
		SWCF_5_5	0.534
		SWCF_6_15	0.497
		SWCF_4_6	0.486
		SWCF_5_15	0.482
θ_4 (c_invrs_tau_wpxp_n2_thresh)	$(-\infty, 0.0004)$	SWCF_6_18	0.701
		SWCF_5_5	0.558
		SWCF_6_14	0.500
		SWCF_5_4	0.485
		SWCF_5_9	0.456
θ_5 (c_invrs_tau_n2_wp2)	$(0.25, \infty)$	SWCF_5_5	0.605
		SWCF_5_4	0.585
		SWCF_6_18	0.574
		SWCF_6_8	0.496
		SWCF_4_6	0.461

Table 2.1: Top 5 metrics with highest probability of not being contained for each parameter.

2.2 Construction of Error Bars

When performing parameter optimization using QuadTune, it is important not only to obtain point estimates for the parameters but also to quantify the uncertainty associated with these estimates. Since analytical expressions for confidence intervals are often unavailable in complex models, we rely on bootstrap methods. The constructed confidence intervals can be interpreted as error bars for the parameter values as they account for uncertainty.

2.2.1 Quantile-Based Confidence Intervals for Parameters

Specifically, we repeatedly resample the input metrics, apply the same optimization procedure (QuadTune) to each resample, and collect the resulting parameter estimates $\hat{\theta}^{*(b)}$. These bootstrap replicates form an empirical distribution that approximates the sampling distribution of the estimator. From this empirical distribution, we can construct confidence intervals by taking the appropriate quantiles.

This approach is known as the *percentile bootstrap method*, where confidence intervals are constructed directly from the empirical quantiles of the bootstrap replicates. If we denote the full dataset estimate by $\hat{\theta}$ and the bootstrap replicates by $\hat{\theta}^{*(b)}$, then a nonparametric $(1-\alpha)100\%$ confidence interval is given by the $\alpha/2$ and $1-\alpha/2$ quantiles of the empirical distribution of $\hat{\theta}^{*}$. This method is particularly useful when the sampling distribution is unknown [3].

Figure 2.2 shows the constructed error bars of the tuned parameters based on 95% confidence intervals using the percentile method. It can be observed that the intervals are heavily asymmetric around the full dataset parameter value, which is highlighted as a dot and indicated by a green dashed line. This is due to the heavy tails of the parameter distributions, which the percentile method does not account for.

While the percentile method is simple and widely used, it may produce misleading or overly wide confidence intervals when the bootstrap distribution is skewed or biased. In such cases, the percentile method does not account for the asymmetry or bias in the estimator and can therefore exaggerate uncertainty or misrepresent the range of plausible values.

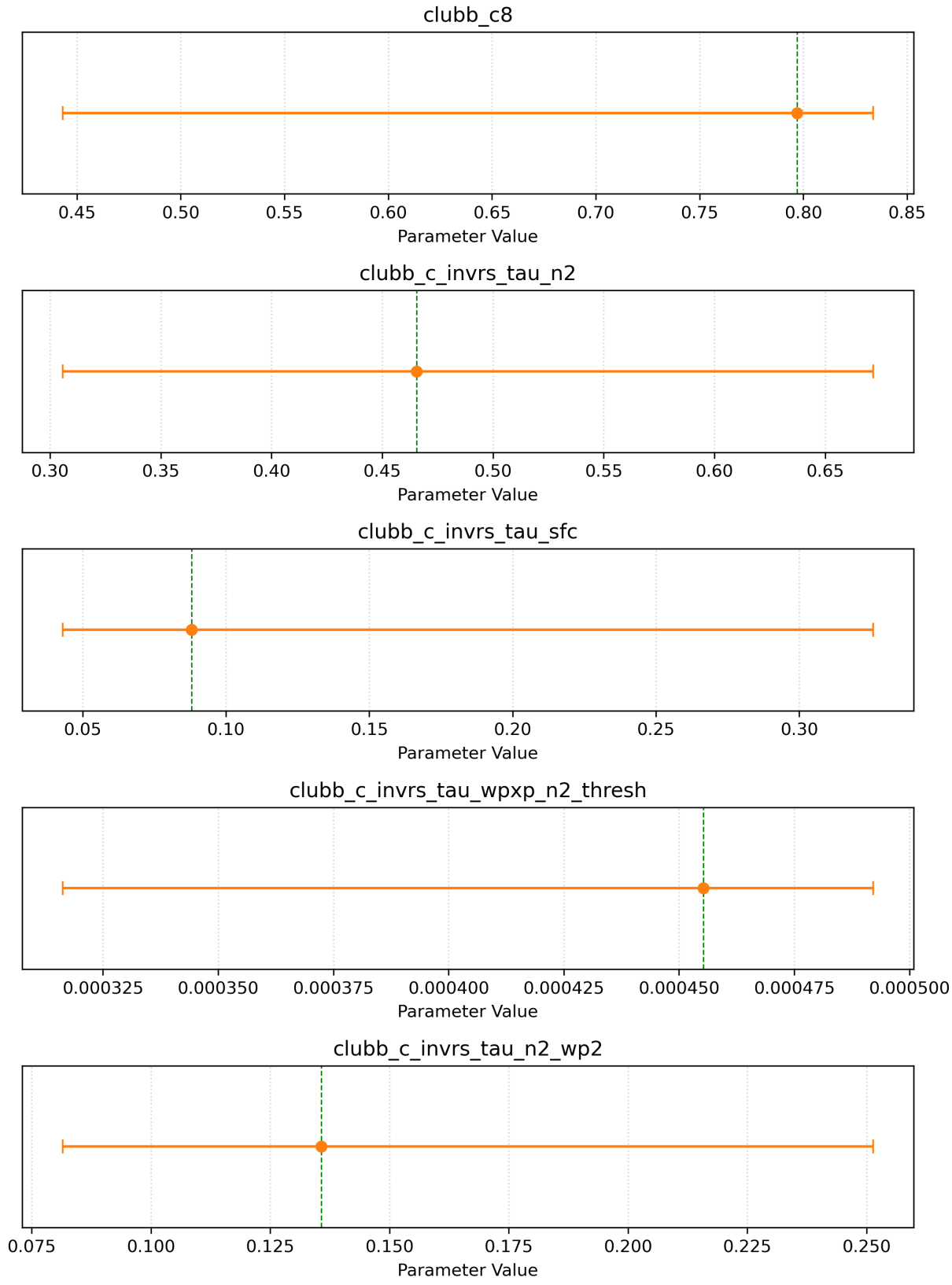

Figure 2.2: 95% Confidence intervals of the tuned parameter values using the percentile method.

2.2.2 Bias-Corrected and Accelerated Bootstrap Intervals

To address these issues, we also compute $\mathrm{BC_a}$ bootstrap confidence intervals [4]. The $\mathrm{BC_a}$ method improves on the percentile method by adjusting the confidence limits in two ways:

- **Bias correction:** A factor z_0 is computed to shift the confidence bounds based on the proportion of bootstrap estimates less than the full-data estimate:

$$z_0 = \Phi^{-1}\left(\frac{\#\{\hat{\theta}^{*(b)} < \hat{\theta}\}}{B}\right),$$

  where Φ^{-1} is the inverse standard normal cumulative distribution function. This corrects for systematic bias in the bootstrap distribution.

- **Acceleration:** A factor a ("acceleration") is calculated to account for the skewness in the estimator, using the jackknife method. The acceleration term is defined as:

$$a = \frac{\sum_{i=1}^{n}(\bar{\theta}_{(\cdot)} - \hat{\theta}_{(-i)})^3}{6\left[\sum_{i=1}^{n}(\bar{\theta}_{(\cdot)} - \hat{\theta}_{(-i)})^2\right]^{3/2}},$$

  where $\hat{\theta}_{(-i)}$ is the parameter estimate with the i-th observation removed, and $\bar{\theta}_{(\cdot)}$ is the mean of all jackknife estimates. This corrects for curvature in the sampling distribution.

The adjusted quantiles are then computed as:

$$\alpha_1 = \Phi\left(z_0 + \frac{z_0 + z_{\alpha/2}}{1 - a(z_0 + z_{\alpha/2})}\right), \quad \alpha_2 = \Phi\left(z_0 + \frac{z_0 + z_{1-\alpha/2}}{1 - a(z_0 + z_{1-\alpha/2})}\right),$$

and used to extract the lower and upper bounds of the $\mathrm{BC_a}$ confidence interval from the sorted bootstrap replicates.

The jackknife [5] is a resampling technique used to estimate the sensitivity of a parameter estimator to individual data points. Given a dataset of size n, we construct n leave-one-out subsamples, where each subsample removes one observation. For each subsample, we

recompute the parameter estimate:

$$\hat{\theta}_{(-i)} = \text{estimate based on data without observation } i.$$

The set $\{\hat{\theta}_{(-1)}, \hat{\theta}_{(-2)}, \ldots, \hat{\theta}_{(-n)}\}$ provides information on how the parameter responds to local perturbations in the dataset. This sensitivity is summarized through the acceleration factor a, which appears in the $\mathrm{BC_a}$ correction formula.

In the BCa method, two important quantities are computed: the bias-correction parameter z_0 and the acceleration parameter a.

The bias-correction parameter z_0 measures the median bias of the estimator. If $z_0 = 0$, approximately 50% of the bootstrap estimates are less than or equal to the observed estimate $\hat{\theta}$, indicating no noticeable bias. If $z_0 > 0$, most bootstrap estimates are smaller than $\hat{\theta}$, suggesting positive bias. Conversely, if $z_0 < 0$, most bootstrap estimates are larger than $\hat{\theta}$, suggesting negative bias. In the extreme case where $z_0 \to \infty$, almost all bootstrap estimates are smaller than $\hat{\theta}$, indicating very strong positive bias.

The acceleration parameter a measures the skewness of the estimator's sampling distribution and how the standard error changes as a function of the parameter. A value of $a = 0$ indicates that the standard error is roughly constant and the distribution is symmetric. If $a > 0$, the distribution is positively skewed, meaning the right tail is longer and larger estimates are more variable. If $a < 0$, the distribution is negatively skewed, meaning the left tail is longer and smaller estimates are more variable.

If both $z_0 = 0$ and $a = 0$, the $\mathrm{BC_a}$ method reduces to the simple percentile method, using the unadjusted quantiles of the bootstrap distribution to form the confidence interval.

Figure 2.3 shows a comparison of the error bars constructed using the $\mathrm{BC_a}$ and percentile method. Especially `clubb_c8` shows a significant difference between the two confidence intervals. The percentile interval is much more asymmetric to the left due to the heavy

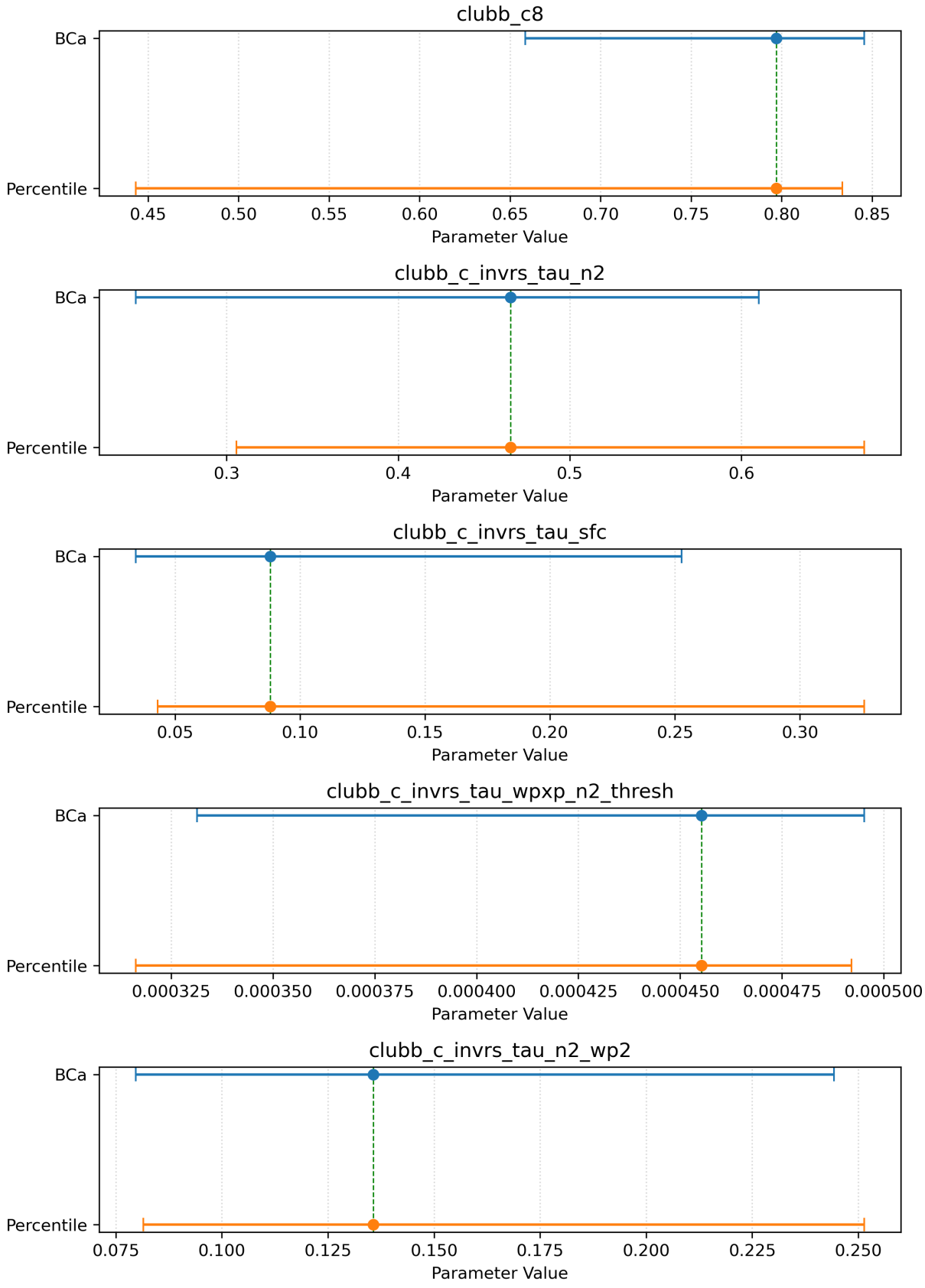

Figure 2.3: Comparison of confidence intervals using the percentile and BC_a method.

lower tail of the bootstrap distribution, while the BC_a method provides a clearly shorter and more symmetric interval even though both are constructed at the 95% confidence level.

While the percentile method provides an easy-to-compute baseline interval, the BC_a method offers a more accurate and adaptive alternative in the presence of bias and skewness. Using both methods allows us to compare the symmetry and robustness of the parameter estimates. In practice, we find that the BC_a intervals are often narrower and more centered around the full-data estimate, offering a better characterization of uncertainty, especially for parameters with skewed bootstrap distributions [6, Ch. 5].

3 Residual Analysis

3.1 Residual Distribution Analysis

Beyond analyzing the distribution of optimized parameters, we evaluate the residuals to assess model performance. To evaluate the impact of parameter tuning, we compare the residual distributions obtained from different optimization approaches:

- **Default residuals**: Computed using the model's default parameter values before optimization, serving as a baseline ($f_i^{\text{obs}} - \hat{m}_i\left(\theta_{def}\right)$).
- **Optimized residuals (full dataset)**: Residuals obtained after tuning QuadTune on the full dataset ($f_i^{\text{obs}} - \hat{m}_i(\hat{\theta})$).
- **Bootstrapped residuals**: Residual distributions computed from bootstrap-tuned parameter sets ($f_i^{\text{obs}} - \hat{m}_i(\hat{\theta}^{*(b)})$).

Comparing these residuals allows us to assess whether tuning leads to systematic improvements and how much variability is introduced by bootstrapping.

To further assess the properties of residuals, we examine:

- **Symmetry and Skewness**: A symmetric residual distribution suggests unbiased predictions, whereas skewness may indicate systematic overestimation or underestimation.
- **Heavy-Tailed Distributions**: If the residuals exhibit heavy tails, the model occasionally produces large errors, indicating sensitivity to specific conditions.

- **Heteroscedasticity**: We check whether residual variance remains constant across different metrics. Increasing variance suggests that model performance is inconsistent across different regions.

Figure 3.3 shows the distributions of the bootstrapped residuals for every region approximated using histograms and kernel density estimation. The red dashed line shows the residuals corresponding to the default model parameters while the green dashed line shows the residuals obtained by tuning on the full dataset. Lastly, the gray vertical line represents the line $x = 0$ and therefore we expect the distributions to center around the gray lines if the model is able to effectively remove the bias of the approximated metrics.

One observation that can be made is that the default residuals differ from the bootstrapped residuals by a lot in most cases. This highlights the fact that the default parameters are not optimal and the tuning process is effective in improving the overall model performance by adjusting the parameter values. However, this does not mean that the tuning process improves every metric individually.

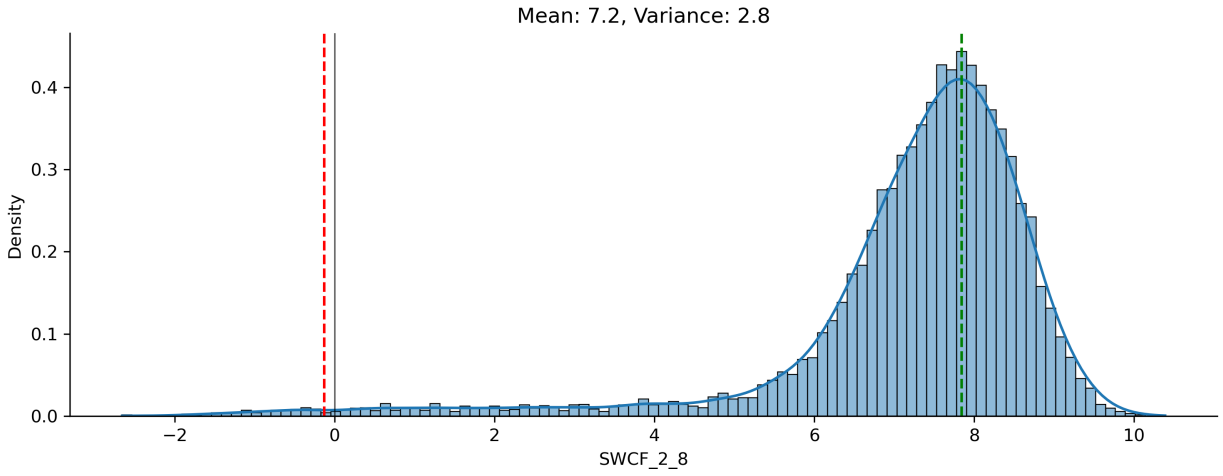

Figure 3.1: Bootstrapped residuals of `SWCF_2_8`.

For example, `SWCF_2_8` (see Figure 3.1) was approximated perfectly using the default parameters and the model performance for this metric is significantly worse for most bootstrap

samples, as well as the full dataset. This is due to the fact that the loss function minimizes the average error and not the individual error of this metric. It can therefore be assumed that this region might be in conflict with some other metrics and therefore some tuning trade-offs need to be made to improve the overall model performance.

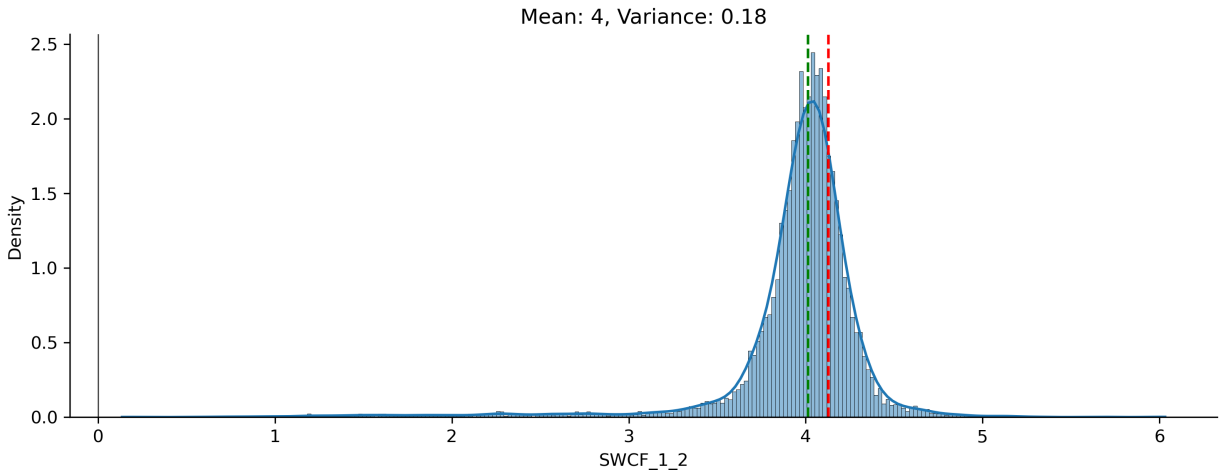

Figure 3.2: Bootstrapped residuals of `SWCF_1_2`.

In contrast, the majority of bootstrapped residuals of `SWCF_1_2` (see Figure 3.2) are centered around the full data residual which is almost equal to the default value. This suggests that for this metric, the default value was already a good fit for the data. However, all of these value are not close to zero and even long lower tail does not reach the origin. Therefore, this region is just not tunable for any set of parameters which suggests model structural error.

Since the histograms are on different scales, it is difficult to identify heteroscedasticity. Therefore, Figure 3.4 shows a plot of the variances of each distribution of bootstrapped residuals. Visualizing these variances allows us to evaluate the stability of residuals and identify cases where model predictions have high uncertainty. It is clear to see that the variance is not constant across all of the regions. Especially two of these regions' variances are vastly different from the other ones with these are `SWCF_6_14` and `SWCF_6_18`. This indicates that these two metrics are the most sensitive to changes in the input metrics of the tuning process.

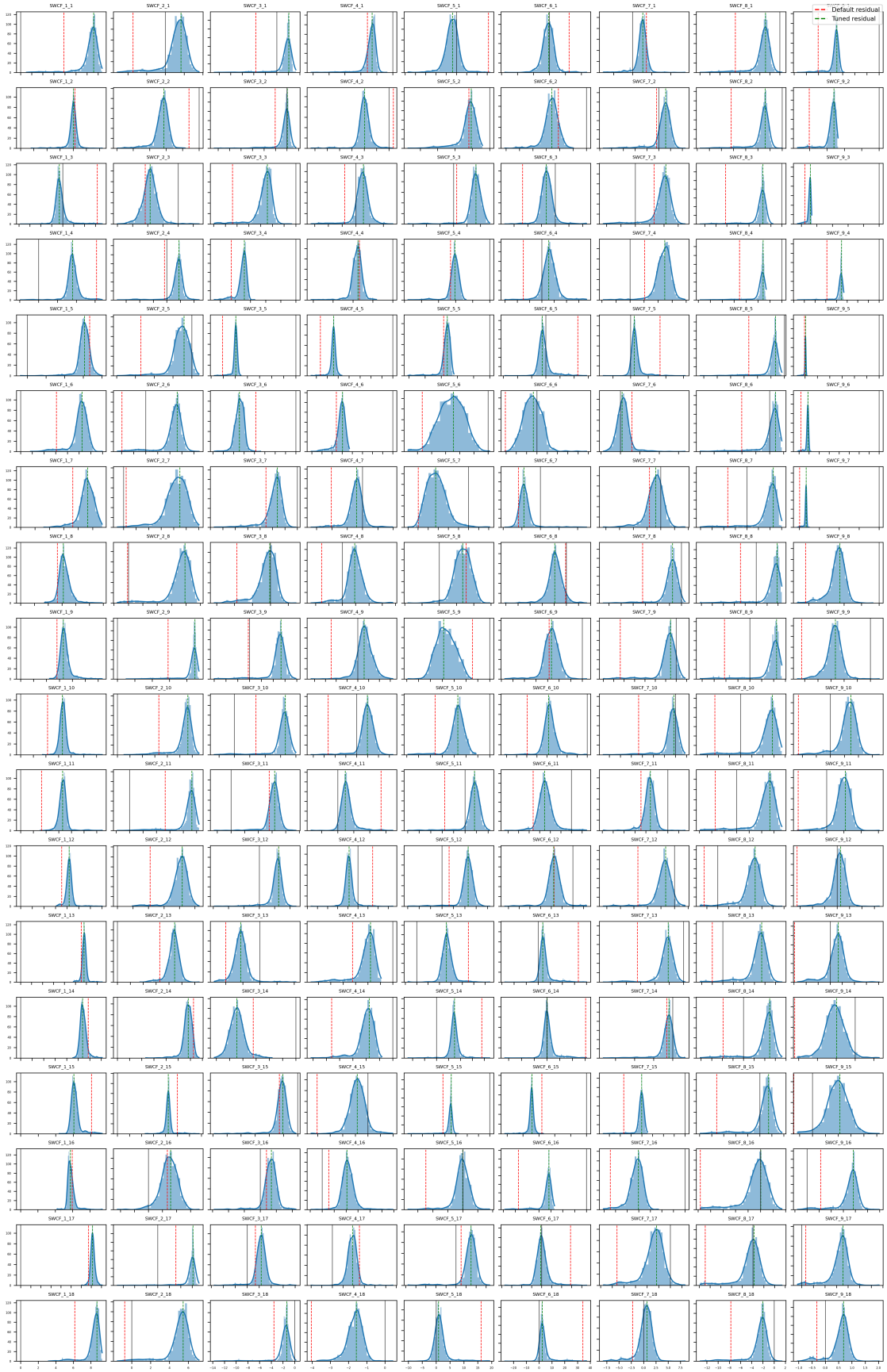

Figure 3.3: Distribution of bootstrapped residuals for each metric on a $20° \times 20°$ grid.

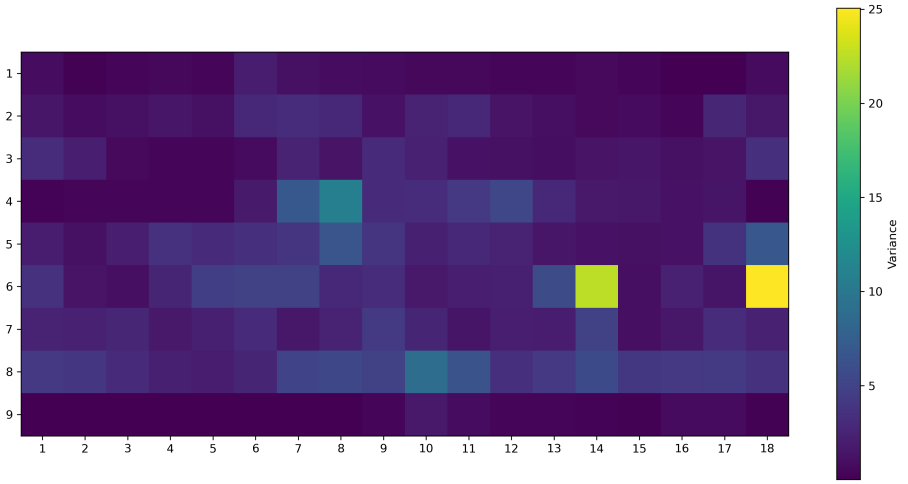

Figure 3.4: Variances of bootstrapped residuals for each metric on a $20° \times 20°$ grid.

To gain a better understanding of model performance, we compute the Mean Squared Residuals (MSR) for each individual grid point, i.e. $\frac{1}{B}\sum_{b=1}^{B}\left(r_i^{(b)}\right)^2$. This helps identify whether specific regions exhibit larger residuals, indicating spatial variability in model accuracy. By analyzing MSR values, we can determine whether the tuning process improves overall model performance consistently across different metrics.

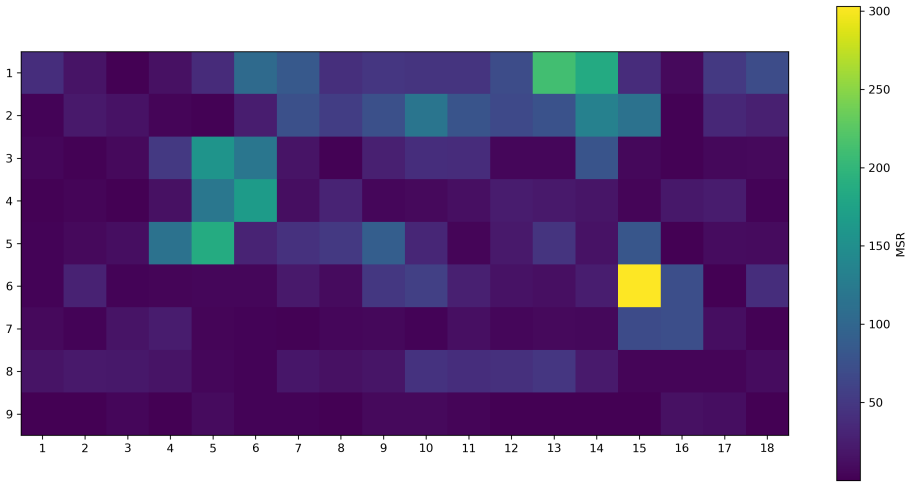

Figure 3.5: Mean Squared Residuals of each metric.

Figure 3.5 shows that the model performance is not consistent across all of the metrics. Especially `SWCF_6_15` performs significantly worse than all of the other regions. However,

in Figure 3.4 it can be observed that the variance of this metrics is comparably low which indicates that the model performs bad on this regions consistently and therefore the model is not feasible of tuning this region due to a lack of sensitivity or model structural error.

It is also interesting to observe that the metrics with the highest variances (`SWCF_6_14` and `SWCF_6_18`) have low MSR. This suggests that even though they are sensitive to the input metrics in the tuning process, these regions are still approximated well on average.

3.2 Identifying Tuning Trade-Offs Between Metrics

In order to investigate potential trade-offs during the parameter tuning process, we analyze the relationship between residuals across different model output metrics. A trade-off occurs when the reduction of bias in one metric systematically coincides with an increase in bias in another. Such behavior may indicate limitations in the model structure or emulator that prevent simultaneous bias reduction across all regions.

A direct approach to identifying trade-offs involves examining the residuals jointly for each pair of metrics across all bootstrap samples. Ideally, if the model could be tuned perfectly for all metrics simultaneously, the residuals would approach zero. For any pair of metrics, this corresponds to bootstrap samples near the origin in a 2D residual scatter plot.

To illustrate this idea, one could imagine the relationship of two metrics i and j and their corresponding residuals $r_i^{(b)}$ and $r_j^{(b)}$ for a bootstrap sample b. Then if the pairwise residuals have a strong relationship that runs through the origin, that means that there is a set of parameters that removes the bias in both for both of these metrics and also tuning metric i improves the approximation of metric j and vice versa. Figure 3.6 shows an illustration of this when there is a linear relationship between the two metrics. This would be a case where

we see no tuning trade-off between the pair of metrics.

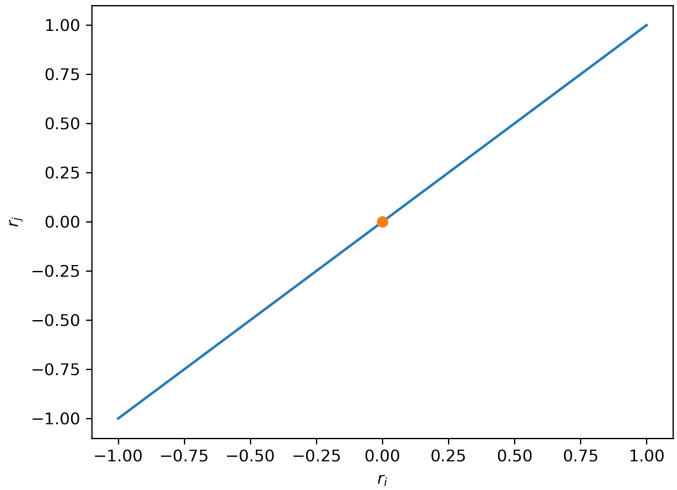

Figure 3.6: Example of two metrics that are able to be tuned jointly.

A tuning trade-off could be quantified if there is a relationship between the pair of residuals of two metrics, i.e. the residuals lie on some type of line, and the line crosses the x-axis as well as the y-axis but does not get close to the origin. This signals that even though the metrics can be tuned well individually, they can not be simultaneously approximated well. An example for this is given in figure 3.7 where the residuals lay on a line but do not cross the origin.

Lastly, if one the metrics is not able to be tuned individually, then the pair of metrics will also not be able to be tuned jointly. Therefore, this is not due to a tuning trade-off, but a problem in the model structure that does not allow the tuner to improve the given metric.

However, since the number of metric pairs is large, generating all pairwise scatter plots becomes impractical. To prioritize potentially conflicting pairs, we construct a triangular matrix (Figure 3.8) in which each entry represents the minimum ℓ_2 norm of the bootstrap residual vector for a given pair of metrics. Specifically, for two metrics i and j, and each bootstrap sample b, we compute the residuals $r_i^{(b)}$ and $r_j^{(b)}$, and define the minimum pairwise

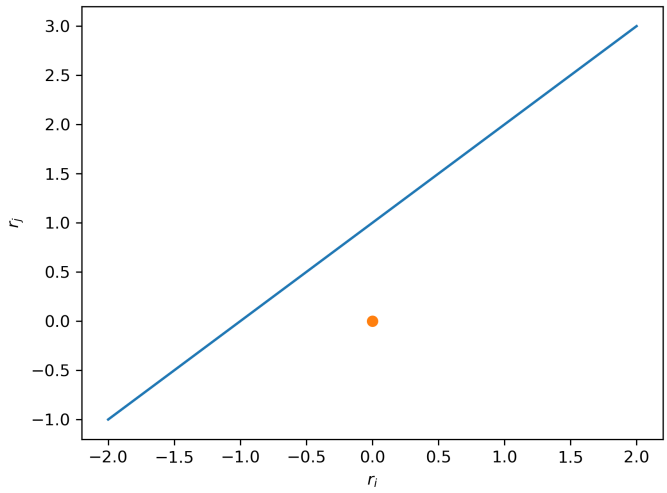

Figure 3.7: Example of two metrics that are not able to be tuned jointly, i.e. a tuning trade-off.

residual norm as:

$$\min_{b} \left\| \begin{pmatrix} r_i^{(b)} \\ r_j^{(b)} \end{pmatrix} \right\|_2 .$$

This quantity is small if at least one bootstrap sample brings both residuals simultaneously close to zero, and large otherwise.

By visualizing these minimum residual norms as a heatmap, we can identify pairs of metrics that are difficult to tune jointly. High values in the matrix suggest that the residuals for the corresponding metrics cannot be minimized at the same time, indicating a potential trade-off in the optimization process. In contrast, low values suggest compatibility in tuning.

The diagonal entries of the residual norm matrix represent the minimum residual norm achieved for each individual SWCF metric across all bootstrap samples. A large diagonal value for a given metric indicates that, regardless of the tuning configuration, this metric consistently exhibits high residuals. This suggests a region or variable that is fundamentally difficult to tune, either due to structural model limitations or incompatibility with the tuning approach.

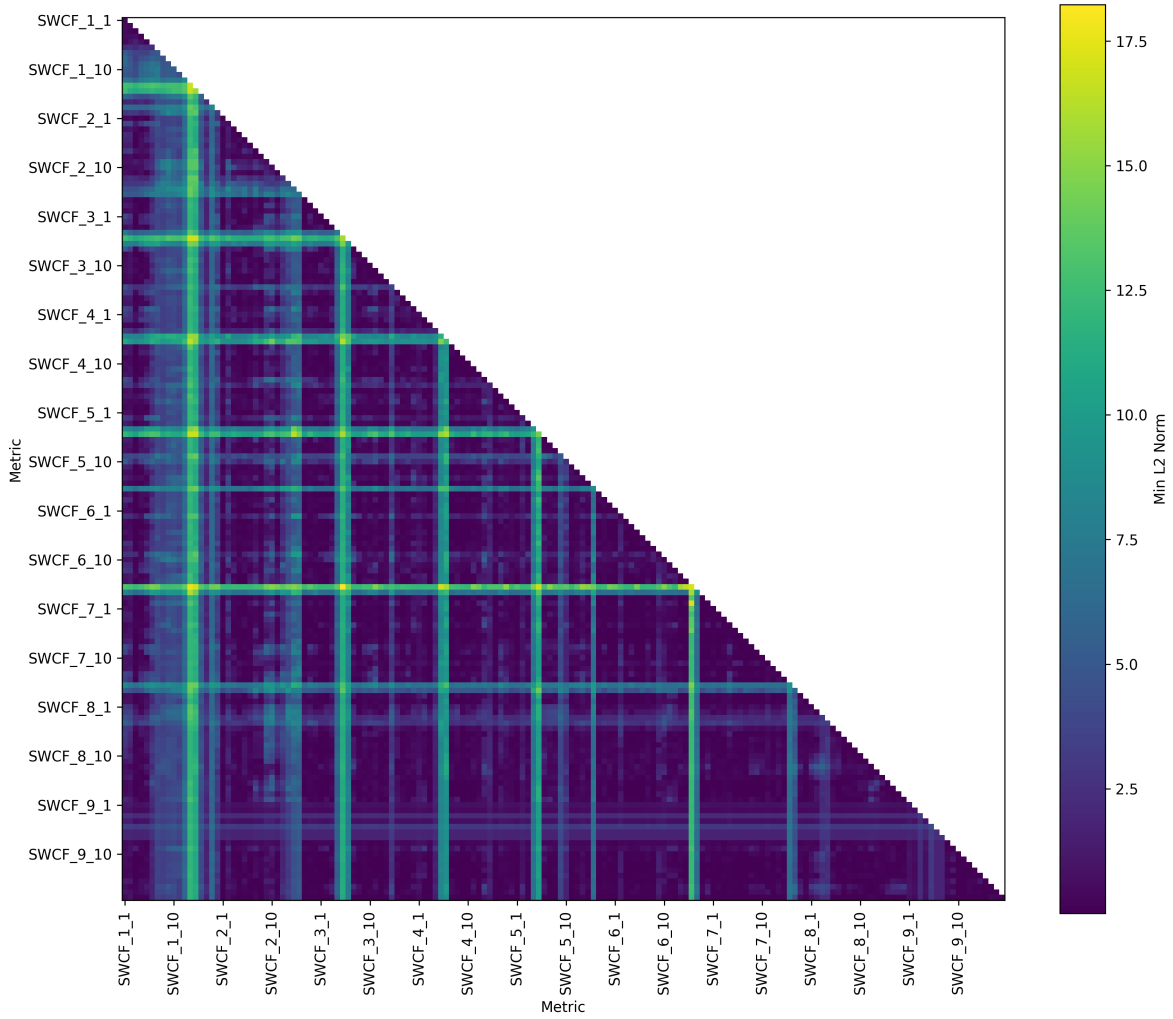

Figure 3.8: Heatmap of minimum ℓ_2 norm between residuals for each pair of metrics. Each entry represents the smallest distance to the origin across bootstrap samples, indicating how closely both residuals can be minimized simultaneously. Bright regions indicate potential tuning trade-offs.

Conversely, small diagonal values indicate that a metric can, at least in some bootstrap samples, be tuned well in isolation. More subtle insights emerge from analyzing the off-diagonal entries in the context of their corresponding diagonal values. Specifically, when two metrics both have low diagonal values (individually tunable) but a high off-diagonal residual norm, this suggests that while each metric can be well-tuned on its own, they cannot be simultaneously minimized. This reveals an underlying tuning trade-off: the parameter configurations that reduce the residual for one metric are incompatible with those needed for the other. Such pairwise conflicts are critical for understanding global limitations in the tuning process, as they highlight competing regional or structural requirements that cannot be simultaneously satisfied within the current model framework.

To illustrate the types of tuning interactions revealed by the residual norm matrix in Figure 3.8, we examine four representative metric pairs. Each is visualized using a 2D scatter plot of bootstrap residuals, where each point corresponds to the residuals for the two selected metrics in a single bootstrap sample.

1. **Individually untunable:** `SWCF_6_15` **vs.** `SWCF_5_5` Both of these metrics exhibit large diagonal entries in the residual norm matrix, indicating that even individually, their residuals cannot be significantly reduced in any bootstrap sample (Figure 3.9).

2. **Asymmetric Tuning:** `SWCF_6_15` **vs.** `SWCF_9_16` In this pair, `SWCF_6_15` continues to exhibit poor tunability, while `SWCF_9_16` has a small diagonal entry, indicating it can be tuned well individually (Figure 3.10).

3. **Tuning Trade-Off:** `SWCF_8_18` **vs.** `SWCF_9_16` Both metrics in this pair have small diagonal values, meaning they can each be tuned well individually. However, the pairwise residual norm is larger, indicating that there is no bootstrap sample in which both residuals are simultaneously small (Figure 3.11).

4. **Jointly tunable: `SWCF_3_1` vs. `SWCF_8_9`** Both metrics in this pair have small diagonal values, meaning they can each be tuned well individually. Additionally, the pairwise residual norm is also small, suggesting that these metrics are also able to be tuned jointly (Figure 3.12).

As shown in Figure 3.9, the residuals for both metrics remain far from zero in all bootstrap samples. The scatter points form a dense cluster in the bottom-left quadrant, indicating persistent negative bias in both regions. No sample comes close to the origin, and the correlation between the residuals is moderate ($\rho = 0.63$), suggesting that while both residuals tend to be biased together, the model lacks the flexibility to reduce either of them. This pair exemplifies a situation where the underlying model structure or parameterization fails to capture the observed behavior in these regions, resulting in residuals that cannot be effectively minimized through tuning.

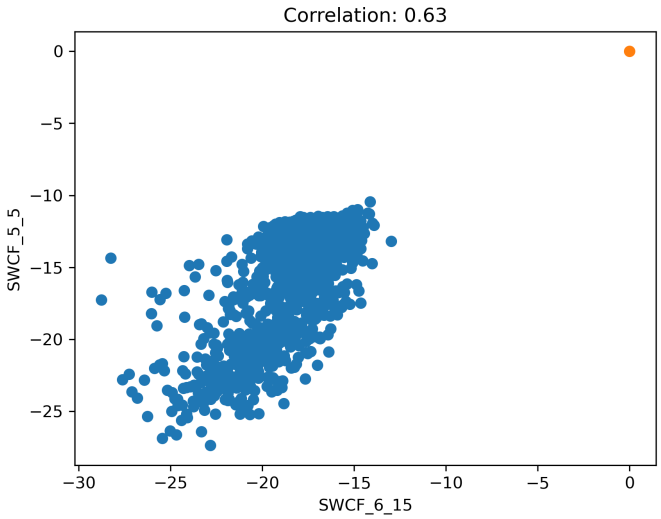

Figure 3.9: Bootstrap residuals for `SWCF_6_15` vs. `SWCF_5_5`.

Figure 3.10 reveals a strong asymmetry in tunability between the two metrics. While residuals for `SWCF_9_16` span a wide range and include values near zero, the residuals for `SWCF_6_15` are consistently large and negative, forming a vertical distribution. The correlation is weak ($\rho = 0.28$), reflecting the independence of tuning outcomes between the two metrics. This

case demonstrates that SWCF_9_16 can be tuned successfully, but SWCF_6_15 remains persistently biased across all samples. The model is only partially flexible in this region of parameter space, yielding asymmetrical tuning success.

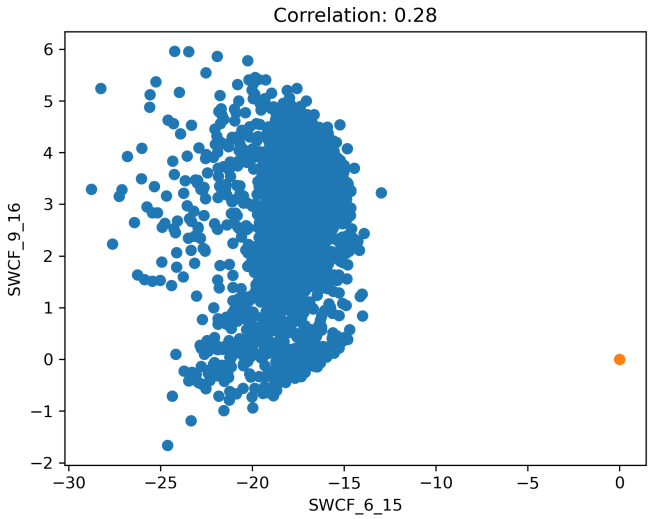

Figure 3.10: Bootstrap residuals for SWCF_6_15 vs. SWCF_9_16.

Figure 3.11 shows that both SWCF_8_18 and SWCF_9_16 exhibit broad ranges of residuals that include values near zero, suggesting that each metric can be tuned effectively on its own. However, the residuals are highly positively correlated ($\rho = 0.97$), forming a tight linear pattern across bootstrap samples. Despite individual tunability, there is no sample in which both residuals are simultaneously small. In particular, no points cluster near the origin (highlighted in orange for reference).

This behavior reflects a structural limitation in the model: parameter changes that reduce the bias in one metric necessarily move the other in the same direction. As a result, the model cannot independently control these two outputs, and simultaneous bias reduction is effectively blocked. This indicates a deeper coupling in the emulator or underlying process representation, which restricts the ability to achieve jointly optimal tuning for these two regions.

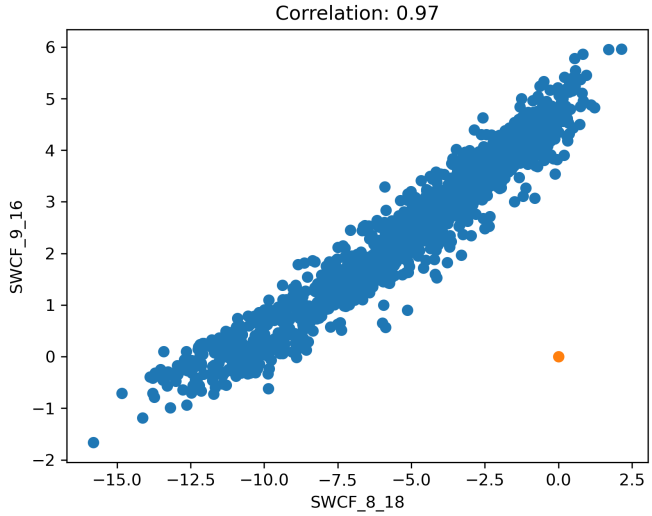

Figure 3.11: Bootstrap residuals for `SWCF_8_18` vs. `SWCF_9_16`.

In figure 3.12 it can be seen that the residuals have a strong linear relationship ($\rho = 0.91$) and the corresponding linear pattern crosses the origin. Therefore, these two regions are not conflicting as they are able to be tuned well jointly and improving one of the metrics also improves the other one due to the linear relationship.

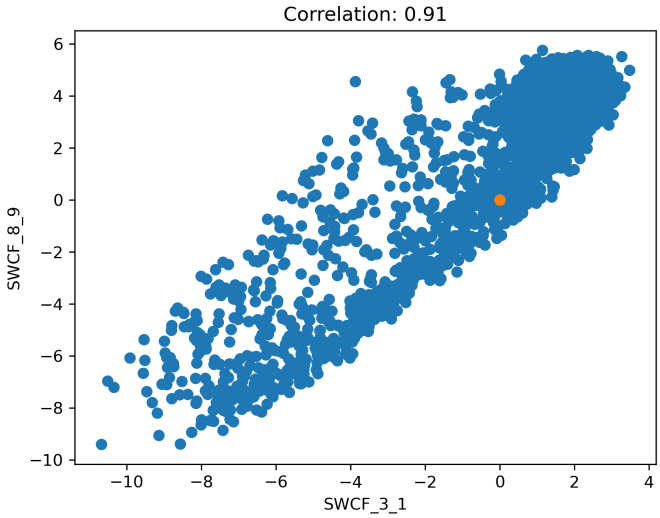

Figure 3.12: Bootstrap residuals for `SWCF_3_1` vs. `SWCF_8_9`.

One approach to systematically identifying tuning trade-offs is to automate their detection

using threshold criteria. Specifically, a residual can be considered non-negligible if its absolute value exceeds a chosen threshold, such as 0.5. Under this criterion, a pair of metrics is flagged as jointly biased if their combined residual norm exceeds $1/\sqrt{2}$, which corresponds to the Euclidean norm when both residuals are exactly at the individual threshold.

To further isolate structurally driven conflicts, this can be combined with a correlation threshold, for example, requiring the residuals to exhibit a correlation above 0.8. A high residual correlation implies that both metrics tend to respond similarly across parameter configurations, indicating structural coupling in the model that prevents their independent adjustment.

Applying both criteria results in a binary trade-off map (Figure 3.13), where black dots highlight metric pairs that are not only jointly biased, but also tightly linked through the model structure. This diagnostic tool provides a systematic method of identifying regions where the tuning flexibility is restricted. This information can inform future efforts to improve model parameterizations or develop region-specific tuning strategies.

In Section 3.1 it was discussed that `SWCF_2_8` can be approximated very well using the default parameters but the model performs substantially worse after the tuning process on the full dataset and the bootstrap samples. Now, Figure 3.13 provides an explanation of this behavior because the tuner has to perform tuning trade-offs between `SWCF_2_8` and multiple other regions.

Choosing an appropriate residual threshold is inherently challenging. Setting the threshold too low results in a large number of identified tradeoffs, including many false positives, which can obscure meaningful relationships. Conversely, setting the threshold too high increases the risk of false negatives, potentially overlooking important tradeoffs. Therefore, the thresholds were selected empirically to balance these considerations. A correlation heatmap and the resulting scatter plots of the valid tradeoffs are provided in Appendix A and Appendix B.

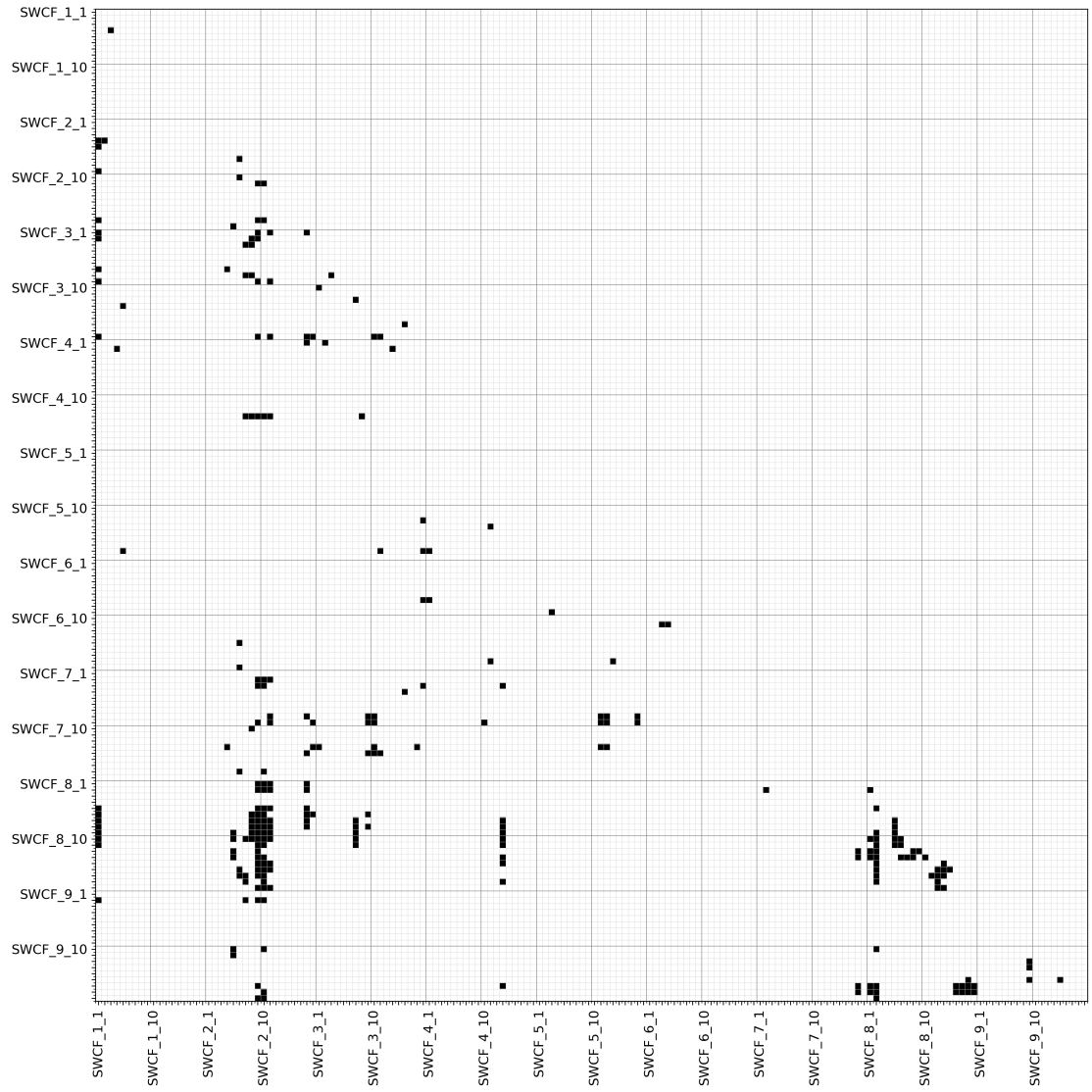

Figure 3.13: Binary map of identified tuning trade-offs based on residual norm and correlation thresholds. Each black dot represents a pair of SWCF metrics that cannot be simultaneously tuned well, defined by exceeding a residual norm threshold of $1/\sqrt{2}$ and a residual correlation above 0.8

4 Conclusion and Outlook

4.1 Conclusion

This thesis presented a comprehensive analysis of parameter uncertainty and tuning robustness in climate models using the QuadTune framework, with a particular focus on spatial patterns of residuals and the impact of bootstrap-based resampling. By applying nonparametric bootstrap techniques, we quantified the variability in parameter estimates and evaluated the consistency of model performance across different spatial regions.

Our findings show that most tuned parameters exhibit unimodal and relatively tight distributions around the full-dataset estimate, suggesting a stable optimization process. However, some parameters demonstrated significant skewness or sensitivity to small perturbations in the input metrics. The construction of empirical confidence intervals, particularly through the $\mathrm{BC_a}$ method, provided a deeper understanding of asymmetries and the limitations of the percentile method in skewed distributions.

In terms of model performance, residual analysis revealed that while tuning improves global error metrics, some regions remain poorly approximated. These residual patterns often reflect structural limitations in the climate model that cannot be addressed through parameter tuning alone. The use of residual distributions, variances, and mean squared residuals highlighted both stubborn biases and heteroscedastic behavior, revealing areas where the model lacks flexibility.

A key contribution of this work is the identification of tuning trade-offs where improvements

in one region or variable systematically degrade performance in another. By constructing a matrix of minimum residual norms and analyzing pairwise residual scatterplots, we developed a method to systematically detect conflicting regions. The final binary trade-off map provides a concise summary of structural constraints within the model that limit simultaneous bias reduction across metrics.

4.2 Outlook

The methodology and diagnostics developed in this thesis can be extended in several directions:

- **Model improvement:** Regions identified as structurally untunable should be prioritized for improvements in parameterizations or physical process representation. The binary conflict map can serve as a guide for targeted development.

- **Threshold sensitivity:** Future work could explore how varying residual norm and correlation thresholds affects trade-off detection and whether more adaptive, data-driven thresholds are beneficial.

- **Extending the binary trade-off map:** The binary conflict map could be expanded to detect not only conflicting metrics due to linear patterns but also potential nonlinear relationships between them. This may involve alternative dependence measures beyond linear correlation, such as mutual information.

- **Beyond SWCF:** The methods developed here can be applied to other output variables of interest in climate modeling, such as precipitation.

Overall, this work demonstrates the value of combining statistical resampling methods with

QuadTune. The use of bootstrapping, residual analysis, and tuning diagnostics provides a powerful toolkit for quantifying and understanding uncertainty in complex model calibration settings.

Bibliography

[1] Vincent E. Larson, Zhun Guo, Benjamin A. Stephens, Colin Zarzycki, Gerhard Dikta, Yun Qian, and Shaocheng Xie. Quadtune version 1: A regional tuner for global atmospheric models. 2025.

[2] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gaussian_kde.html#scipy.stats.gaussian_kde`, 2001–.

[3] Penn State Eberly College of Science. Stat 200: 4.4 - bootstrap confidence intervals. `https://online.stat.psu.edu/stat200/lesson/4/4.4`, 2024.

[4] Nathaniel E. Helwig. Bootstrap confidence intervals. `http://users.stat.umn.edu/~helwig/notes/bootci-Notes.pdf`, 2017.

[5] Herwig Friedl and Erwin Stampfer. Jackknife resampling. 06 2001.

[6] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, Cambridge, 1997.

A Correlation between Residuals

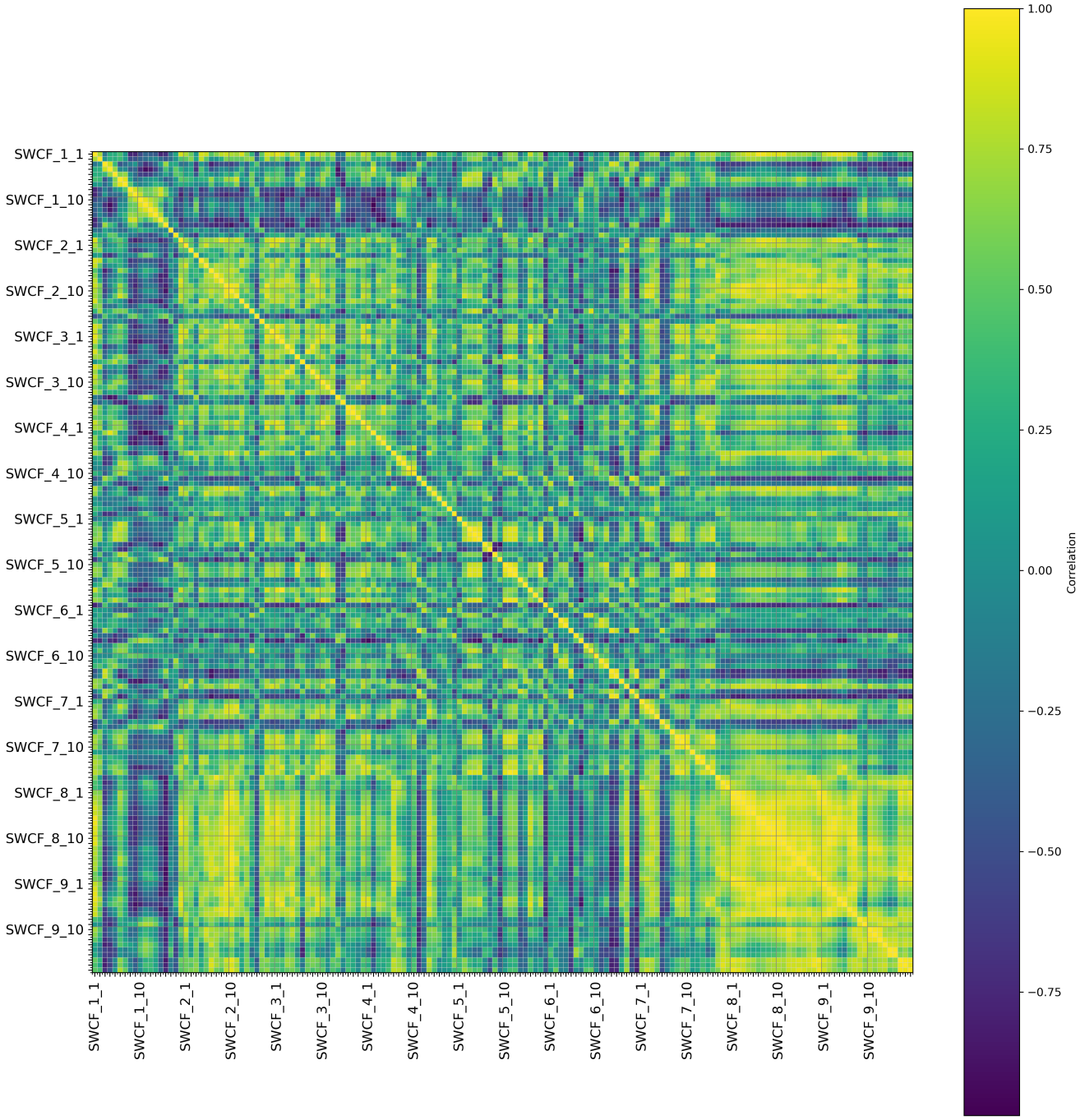

Figure A.1: Heatmap of the correlation of the residuals corresponding to every metric on the grid.

B Valid Tradeoffs between Metrics

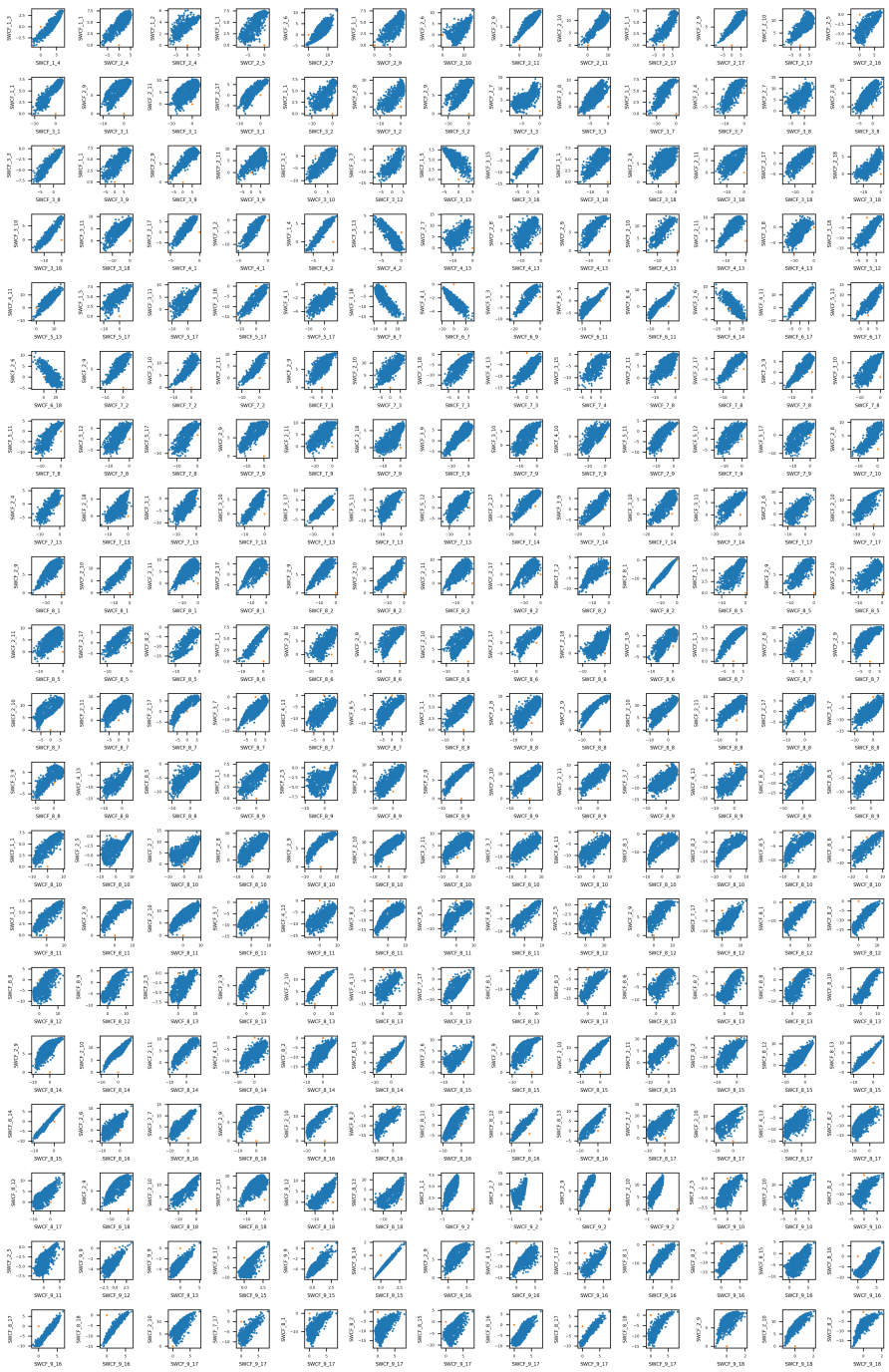

Figure B.1: Scatter plots of valid tradeoffs between metric residuals. Each plot shows the relationship between two metrics where a significant tradeoff was detected based on bootstrap analysis.