A Hierarchical Robust Linear Model for Cryo-EM Map Analysis

I-Ping Tu[a], Shu-Cheng Zheng[a], Yu-Hsiang Lien[a], Szu-Han Lin[a], Po-Chun Lin[a] and Wei-Hau Chang[b]
[a]Institute of Statistical Science, [b]Institute of Chemistry, Academia Sinica

July 4, 2025

Abstract

Cryo-electron microscopy (cryo-EM) has become a pivotal tool for determining the atomic structures of biological macromolecules. In this paper, we propose a hierarchical robust linear (HRL) model to estimate key atom-specific parameters—specifically the amplitude and width of Gaussian functions—that are utilized to assess the consistency between paired cryo-EM maps and their corresponding atomic models, or to improve their refinement. Our HRL modeling leverages minimum density power divergence estimates (MDPDE) to construct a heteroscedastic framework that could potentially mitigate the influence of outliers. We demonstrate the robustness of our method through both simulation studies and real data applications, showing its effectiveness in reducing the impact of outliers and ensuring reliable parameter estimates. Applying our HRL model to analyze real cryo-EM data from human apoferritin (PDB ID: 6Z6U, EMDB ID: 11103) reveals that the Gaussian parameters remain stable across most amino acids, with nitrogen atoms consistently exhibiting lower amplitude and width estimates than expected based on the commonly used Gaussian modeling. These findings highlight the need for a more systematic analysis of paired cryo-EM map and atomic model from the EMDB and PDB to further understand the atom-specific properties in cryo-EM data.

Key Words: cryo-EM map; atomic model; robust estimate; hierarchical linear model; heteroscedasticity; minimum density power divergence estimate; PDB; EMDB.

1 Introduction

Over the past decade, single-particle cryo-electron microscopy (cryo-EM) has emerged as a crucial tool for determining the atomic structures of biological macromolecules. Cryo-EM technology reconstructs three-dimensional (3D) maps from noisy two-dimensional (2D) projection images, and enables the creation of atomic models by fitting peptide chains into these 3D maps. The cryo-EM reconstruction map defines the electrostatic potential values at each 3D grid point, while the atomic model records the coordinate of each atom in the protein. These atomic coordinates collectively form a structural formula that provides essential chemical information for studying the protein functionality and aids in drug and vaccine development. The atomic models, along with

their corresponding cryo-EM maps, are archived in the Protein Data Bank (PDB) and the Electron Microscopy Data Bank (EMDB), respectively.

Q-Score [10], introduced by Wah Chiu's group, has become a standard metric for assessing the consistency quality of cryo-EM maps and their corresponding atomic models. When uploading atomic models to the Protein Data Bank (PDB), one is required to provide Q-scores. This score is calculated by correlating two vectors: one consisting of cryo-EM map values at grid points centered on the atom's position (provided by the atomic model), and the other comprising the 3D Gaussian values corresponding to the same grid points. The 3D Gaussian function is characterized by two parameters: amplitude and width. Since the Q-Score is a correlation value, the amplitude does not influence the score, but the width does. In Q-Score calculations, a single standardized width parameter is applied to all atoms of different types. Similarly, when using Gaussian models to refine cryo-EM maps or atomic models, the amplitude is typically set proportional to the atomic number of the atom, while the width remains standardized [1, 9].

Prior to answering whether these model parameters are properly assumed, a solid statistical method to estimate the parameters from data is required. To this end, we propose a hierarchical linear model to estimate these parameters from cryo-EM maps and atomic models. Specifically, for atoms of the same type, such as the α-carbon (C_α) in alanine (ALA), we assign a common parameter, $\boldsymbol{\mu}$. The parameter $\boldsymbol{\beta}_i$, specific to the ith atom within this group, is modeled as independently distributed according to $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}_i)$ for $1 \leq i \leq I$, where I is the number of atoms in this group.

In this framework, $\boldsymbol{\mu}$ captures the shared characteristics of the group members, reflecting their belonging to the same amino acid type (ALA), while the variability in $\boldsymbol{\beta}_i$ accounts for heterogeneity, possibly due to variations in map quality across different regions of the structure. In our model, the map values around the center of the ith group element, after log transformation, denoted as $\boldsymbol{y}_i$, follow a linear regression with the parameter $\boldsymbol{\beta}_i$. Our primary goal is on estimating the hyper-parameter $\boldsymbol{\mu}$ as the group parameter. This modeling approach can also be viewed as a hierarchical Bayesian linear regression model, where the hyper-parameter $\boldsymbol{\mu}$ is estimated by marginal likelihood, and the posterior distribution for $\boldsymbol{\beta}_i$ can be derived.

To account for variability in data quality across the protein structure, we incorporate covariance matrices $\boldsymbol{\Lambda}_i$ for $\boldsymbol{\beta}_i$ and $\boldsymbol{\Sigma}_{\boldsymbol{y}_i}$ for $\boldsymbol{y}_i$ in our statistical model, specifically for the ith atom. These covariance matrices are constructed using weights derived from minimum density power divergence estimates (MDPDE), as proposed by [2], and [7]. MDPDE is particularly effective in mitigating the impact of outliers by assigning a Gaussian weight, which helps manage outliers arising from noisy 2D projection images or inaccurate atomic fitting.

We first validated the effectiveness of our method using simulated data, demonstrating its robustness in reducing the impact of outliers and ensuring reliable parameter estimates. For real data, we estimated Gaussian parameters for backbone atoms, including α-carbon (C_α), carbonyl carbon (C) and nitrogen (N), across 20 amino acids using a high-resolution cryo-EM map of human apoferritin (PDB ID: 6Z6U, EMDB ID: 11103, resolution: 1.25 Å). The results showed that the parameters remained generally stable across the 20 amino acids, with only a few exceptions. This suggests that our statistical model effectively captures the underlying data structure.

Our observations reveal that nitrogen atoms (N) consistently exhibit the lowest amplitude and width estimates across most of the 20 amino acids. This finding contrasts with expectations based on atomic numbers, where nitrogen is typically anticipated to have the highest amplitude among the three atoms. Furthermore, nitrogen's consistently low ranking in width estimates suggests that certain aspects of the standardized width modeling may require closer examination to accurately capture the unique properties of each atom type. These results highlight the need for a systematic

analysis of paired data from the PDB and EMDB, which could further advance our understanding of atom-specific properties in cryo-EM data using this robust statistical approach.

2 Method

2.1 Statistical Model and the Estimate

A common model for numerically representing a cryo-EM map assigns the value $A_i\phi(r_{ij};\tau_i^2)$ at the jth sampling grid point of the ith atom. Here A_i represents the amplitude parameter and τ_i denotes the width parameter for the ith atom. $\phi(r_{ij};\tau_i^2) = (2\pi\tau_i^2)^{(-3/2)}exp(-\frac{1}{2}r_{ij}^2/\tau_i^2)$ is the 3D Normal density function value at the jth sampling point for the ith atom, where r_{ij} is is the distance from the sampling point to the atom's center. After taking log transformation, we build a linear regression model as

$$y_{ij} = \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_i + \epsilon_{ij}, \quad 1 \le j \le n_i,$$

where y_{ij} is the log transform of the jth map value, n_i is the sample size of the map values, the index i refers to the ith atom and $1 \le i \le I$. We let $\boldsymbol{\beta}_i = \left(\log A_i - \frac{3}{2}\log(2\pi\tau_i^2),\ \frac{1}{\tau_i^2}\right)^\top$, and $\boldsymbol{x}_{ij} = \left(1,\ -\frac{1}{2}r_{ij}^2\right)^\top$ and further let $\boldsymbol{X}_i = (\boldsymbol{x}_{i1},\ldots,\boldsymbol{x}_{in_i})^\top$ to have the vector form for this linear model as

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i,$$

that the length of $\boldsymbol{y}_i$ is n_i and $\boldsymbol{\epsilon}_i$ is the random error vector distributed as $\mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{y}_i})$.

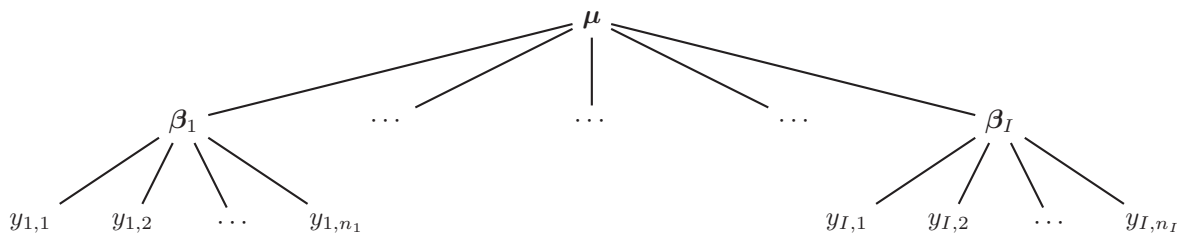
Using the alanine (ALA) residues from PDB ID 6Z6U of human apoferritin as an example, we observe that the protein contains a total of 288 ALA repeats. Due to the 24-fold symmetry in apoferritin, these 288 repeats yield 12 independent observations, resulting in $I = 12$ for the corresponding C_α of ALA. For simplicity and consistency, we assume an equal number of sampling points for each atom, meaning that $n_i = n$ for all $1 \le i \le I$. It is important to note that each amino acid in a protein contributes one alpha carbon (C_α), one carbonyl carbon (C), and one nitrogen (N) to the backbone structure, which serves as the most reliable feature for fitting the atomic model to the cryo-EM map. In this paper, we target to estimate the Gaussian parameters for these backbone atoms categorized by the amino acid types of the protein PDB ID 6Z6U.

Here, we employ a hierarchical model by introducing an additional layer to model the parameters $\boldsymbol{\beta}_i$. Specifically, we assume that $\boldsymbol{\beta}_i \overset{ind.}{\sim} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}_i)$ for each $1 \le i \le I$. This hierarchical structure captures the inherent similarity among the $\boldsymbol{\beta}_i$'s as they represent parameters for the same atom type (such as C_α) within the same amino acid (such as ALA). However, the model also accommodates deviations in these parameters, which may arise from variations in the quality of the reconstruction map across the protein.

This model can be represented as a tree structure, as shown in Table 1, where $\boldsymbol{\mu}$ serves as the group parameter for atoms of the same type (such as C_α) within the same amino acid (like ALA). In this C_α example, the statistical model for the map values $\boldsymbol{y}_i$ is of C_α in the ith ALA residue and follows:

$$\begin{aligned} \boldsymbol{y}_i &\overset{ind.}{\sim} \mathcal{N}(\boldsymbol{X}_i\boldsymbol{\beta}_i, \boldsymbol{\Sigma}_{\boldsymbol{y}_i}); \\ \boldsymbol{\beta}_i &\overset{ind.}{\sim} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}_i), \end{aligned} \tag{1}$$

Table 1: Model Structure: Estimate for C_α of ALA, $I = 12$



where $1 \leq i \leq I$, and I denotes the number of members in this group (e.g., amino acid). In this formulation, $\boldsymbol{y}_i$ represents the map vector (response vector) for the ith atom, while $\boldsymbol{\beta}_i$ is the corresponding regression coefficient vector, modeled with a conjugate prior Normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Lambda}_i$.

This two-level model is a simplified version of the three-level hierarchical Bayesian linear regression model formulated in [6] pp. 389, Chapter 15. It provides the flexibility of unequal variances, allowing for the assignment of weights to individual data points to mitigate the influence of potential outliers. Given our focus on estimating $\boldsymbol{\mu}$, we can utilize a straightforward marginal likelihood estimate to achieve this goal.

$$
\begin{aligned}
& f\left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_I | \boldsymbol{\mu}\right) \\
= & \int f\left(\boldsymbol{y}_1 | \boldsymbol{\beta}_1\right) \cdots f\left(\boldsymbol{y}_I | \boldsymbol{\beta}_I\right) f\left(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_I | \boldsymbol{\mu}\right) d\boldsymbol{\beta}_1 \cdots \ d\boldsymbol{\beta}_I \\
\propto & \ \Pi_{i=1}^{I} \left(\exp\left\{ -\frac{1}{2} \boldsymbol{\mu}^\top \left(\boldsymbol{\Lambda}_i^{-1} - \boldsymbol{\Lambda}_i^{-1} \tilde{\boldsymbol{\Sigma}}_i \boldsymbol{\Lambda}_i^{-1} \right) \boldsymbol{\mu} + \boldsymbol{\mu}^\top \left[\boldsymbol{\Lambda}_i^{-1} \tilde{\boldsymbol{\Sigma}}_i \left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1} \boldsymbol{y}_i \right) \right] \right\} \right),
\end{aligned} \tag{2}
$$

where $\tilde{\boldsymbol{\Sigma}}_i = \left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1} \boldsymbol{X}_i + \boldsymbol{\Lambda}_i^{-1} \right)^{-1}$.

Let $Q(\boldsymbol{\mu})$ be the exponent of (2), we have

$$
\frac{dQ(\boldsymbol{\mu})}{d\boldsymbol{\mu}} = \sum_{i=1}^{I} \boldsymbol{\Lambda}_i^{-1} \tilde{\boldsymbol{\Sigma}}_i \left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1} \right) \left(\boldsymbol{y}_i - \boldsymbol{X}_i \boldsymbol{\mu} \right). \tag{3}
$$

Then, the MLE by solving $\frac{dQ(\boldsymbol{\mu})}{d\boldsymbol{\mu}} = 0$ is as follows.

$$
\hat{\boldsymbol{\mu}} = \left[\sum_{i=1}^{I} \boldsymbol{\Lambda}_i^{-1} \tilde{\boldsymbol{\Sigma}}_i \left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1} \boldsymbol{X}_i \right) \right]^{-1} \left[\sum_{i=1}^{I} \boldsymbol{\Lambda}_i^{-1} \tilde{\boldsymbol{\Sigma}}_i \left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1} \boldsymbol{y}_i \right) \right]. \tag{4}
$$

In deriving $\hat{\boldsymbol{\mu}}$, we assume that the covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{y}_i}$ and $\boldsymbol{\Lambda}_i$, as specified in model (1), are known. In Section 2.2, we use the minimum density power divergence estimate [2, 7] to construct the covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{y}_i}$ and $\boldsymbol{\Lambda}_i$.

2.2 Robust Estimates Formulated as Weighted Estimates

[2] introduced the minimum density power divergence as a method for deriving a robust estimate, expressed as a weighted average of the observed data. In this approach, each data point contributes to the estimate based on its own weight, which is proportional to the power parameter α of the density function with the estimated parameter(s). This α plays a crucial role in balancing robustness and efficiency and is predetermined. Using weights on data has been demonstrated as an effective strategy for controlling outliers. For instance, [12] proposed robust model fitting for point estimation, [11] introduced the density-power-weight method to down-weight outliers in structured data applications, [4] suggested mean shift for clustering analysis, and [5] proposed the γ estimate, and [3] proposed the γ-SUP for clustering analysis. All of these methods appear to utilize weighting schemes to enhance the robustness of their respective estimates.

Assuming independent and identically distributed (i.i.d.) models with equal variance, the minimum density power divergence estimate (MDPDE) for both the mean estimate in a Normal distribution [2] and linear regression [7] produces weighted formulas that are equivalent to the maximum likelihood estimates for heteroscedastic models (with unequal variances). Our approach is to apply the weights derived from the MDPDE in the i.i.d. models to construct the covariance matrices $\mathbf{\Lambda}_i$ and $\mathbf{\Sigma}_{\boldsymbol{y}_i}$ in the heteroscedastic models, as represented in model (1). This strategy allows us to preserve the likelihood framework while incorporating the robustness offered by the minimum density power divergence method.

The density power divergence between two density functions g and f is given by

$$d_\alpha(g, f) = \int \left\{ f^{1+\alpha}(z) - \left(1 + \frac{1}{\alpha}\right) g(z) f^{\alpha}(z) + \frac{1}{\alpha} g^{1+\alpha}(z) \right\} dz \tag{5}$$

where the index $\alpha > 0$ controls the trade-off between robustness and efficiency [[2]]. Given the empirical density function g_n and $f_{\boldsymbol{t}}$, where $\{f_{\boldsymbol{t}}, \boldsymbol{t} \in B\}$ is the parametric family of the model density functions indexed by $\boldsymbol{t}$, the minimum density power divergence estimate (MDPDE) is to find

$$\operatorname*{argmin}_{\boldsymbol{t} \in B} d_\alpha(g_n, f_{\boldsymbol{t}}).$$

Notice that

$$\lim_{\alpha \to 0} d_\alpha(g_n, f_{\boldsymbol{t}}) = \int g_n \log\left(\frac{g_n}{f_{\boldsymbol{t}}}\right) dz,$$

which implies that the power divergence converges to the Kullback-Liebler divergence, when $\alpha \to 0$. Furthermore, the minimum Kullback-Liebler divergence estimate leads to the maximum likelihood estimate (MLE), since

$$\operatorname*{argmin}_{\boldsymbol{t} \in B} \int g_n \log\left(\frac{g_n}{f_{\boldsymbol{t}}}\right) dz = \operatorname*{argmax}_{\boldsymbol{t} \in B} \int \log\left(f_{\boldsymbol{t}}\right) dG_n(z) = \operatorname*{argmax}_{\boldsymbol{t} \in B} \frac{1}{n} \sum_{i=1}^{n} \log(f_{\boldsymbol{t}}(x_i)),$$

where $G_n(z) = \frac{1}{n} I_{[z \le x_i]}$ is the empirical distribution function of the observation variable $x_1, ..., x_n$. As a result, when $\alpha \to 0$, the MDPDE reduces to the maximum likelihood estimate (MLE) which provides the most efficient estimate. However, as α increases, the MDPDE gains robustness but may sacrifice some efficiency. It is recommended in [2] to use a small value of α, as this balances the trade-off by introducing robustness without significantly compromising efficiency.

The estimating equation is constructed by assigning 0 to the derivative of Equation (5) as follows.

$$U_n(\boldsymbol{t}) := \frac{1}{n}\sum_{i=1}^{n} u_{\boldsymbol{t}}(\boldsymbol{z}_i) f_{\boldsymbol{t}}^{\alpha}(\boldsymbol{z}_i) - \int u_{\boldsymbol{t}}(\boldsymbol{z}) f_{\boldsymbol{t}}^{1+\alpha}(\boldsymbol{z}) d\boldsymbol{z} = 0, \tag{6}$$

where $u_{\boldsymbol{t}}(\boldsymbol{z}) = \frac{\partial \log f_{\boldsymbol{t}}(\boldsymbol{z})}{\partial \boldsymbol{t}}$ is the score function.

2.2.1 Construct $\boldsymbol{\Sigma}_{\mathbf{y}}$ for the Linear Regression Model

We let $z_j = y_j - \boldsymbol{x}_j^\top \boldsymbol{\beta}$, and assume that z_j's are i.i.d. distributed as $\mathcal{N}(0, \sigma^2)$ for the linear regression model. Then, we have

$$f_{\boldsymbol{\beta}}(z_j|x_j) = \left(2\pi\sigma^2\right)^{-1/2} \exp\left\{-\frac{1}{2\sigma^2} z_j^2\right\}, \quad 1 \le j \le n, \text{ and}$$
$$u_{\boldsymbol{\beta}}(z_j) = z_j \boldsymbol{x}_j / \sigma^2.$$

In this case, the integral term in Equation (6) vanishes, and we obtain the estimating equation using power parameter α_r, where the lower index r refers to regression, as follows.

$$\frac{1}{n}\sum_{j=1}^{n} \left(y_j - \boldsymbol{x}_j^\top \boldsymbol{\beta}\right) \boldsymbol{x}_j \exp\left[-\frac{\alpha_r}{2\sigma^2}\left(y_j - \boldsymbol{x}_j^\top \boldsymbol{\beta}\right)^2\right] = \mathbf{0}.$$

Thus, the MDPDE for $\boldsymbol{\beta}$ is given by:

$$\hat{\boldsymbol{\beta}}_{\text{MDPDE}} = \left(\sum_{j=1}^{n} w_j \boldsymbol{x}_j \boldsymbol{x}_j^\top\right)^{-1} \left(\sum_{j=1}^{n} w_j \boldsymbol{x}_j y_j\right) = \left(\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X}\right)^{-1} \left(\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{y}\right), \tag{7}$$

where $w_j = \exp\left\{-\frac{\alpha_r}{2\sigma^2}(y_j - \boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}}_{\text{MDPDE}})^2\right\}$, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$, and $\boldsymbol{W}$ is a diagonal matrix with w_j as its jth element. Since $\hat{\boldsymbol{\beta}}_{\text{MDPDE}}$ also appears in w_j, we employed an iterative algorithm to find the solution for $\hat{\boldsymbol{\beta}}_{\text{MDPDE}}$. We use the solution from ordinary linear regression as the initial value, and in our experience, it typically takes fewer than 10 iterations to achieve convergence.

It is clear that the MDPDE estimate, $\hat{\boldsymbol{\beta}}_{\text{MDPDE}}$, for the homoscedastic linear regression model is equivalent to the least squares estimate or the maximum likelihood estimate (MLE) of a weighted linear regression model, where each data pair (x_j, y_j) is assigned a weight w_j. Thus, we conclude that the MDPDE estimate, $\hat{\boldsymbol{\beta}}_{\text{MDPDE}}$, for the linear regression model $\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n)$ is equivalent to the MLE of the following heteroscedastic linear regression model:

$$\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}_{\boldsymbol{y}}), \quad \text{where} \quad \boldsymbol{\Sigma}_{\boldsymbol{y}} = n\sigma^2 \boldsymbol{W}^{-1} / \text{tr}(\boldsymbol{W}^{-\mathbf{1}}). \tag{8}$$

In this definition, the trace of $\boldsymbol{\Sigma}_{\boldsymbol{y}}$ equals $n\sigma^2$, matching the trace of the covariance matrix $\sigma^2 \boldsymbol{I}_n$ in the i.i.d. model.

To derive $\hat{\sigma}^2_{\text{MDPDE}}$ (from the i.i.d. model), we have $u_\sigma(z_j) = \frac{\partial}{\partial \sigma} \log(f_{\sigma(z_j)}) = z_j^2/\sigma^3 - 1/\sigma$. The integral term of (6) is

$$\int u(z) f^{1+\alpha_r}(z) dz = -\left(2\pi\sigma^2\right)^{\frac{-1-\alpha_r}{2}} \frac{\sqrt{2\pi}\alpha_r}{(1+\alpha_r)^{3/2}}.$$

The estimating equation for σ

$$\frac{1}{n}\sum_{j=1}^{n}\left(\frac{z_j^2}{\sigma^2}-1\right)\exp\left\{-\frac{\alpha_r z_j^2}{2\sigma^2}\right\}+\frac{\alpha_r}{(1+\alpha_r)^{3/2}}=0, \tag{9}$$

and the MDPDE of σ^2 is as follows.

$$\hat{\sigma}^2_{\text{MDPDE}}=\left[\text{tr}\left(\boldsymbol{W}\right)-\frac{n\alpha_r}{(1+\alpha_r)^{3/2}}\right]^{-1}\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^{\top}\boldsymbol{W}\left(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right) \tag{10}$$

In real application, we substitute $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}$ for $\boldsymbol{\Sigma}_{\boldsymbol{y}}$ in Model (8), where

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}=n\hat{\sigma}^2_{\text{MDPDE}}\boldsymbol{W}^{-1}/\text{tr}(\boldsymbol{W^{-1}}). \tag{11}$$

Notice that, for the ith atom in the group, we need to calculate individual $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}_i}$ following Equation (11), and we substitute $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}_i}$ for $\boldsymbol{\Sigma}_{\boldsymbol{y}_i}$ in Equation (4) for the group estimate $\hat{\boldsymbol{\mu}}$.

2.2.2 Construct $\boldsymbol{\Lambda_i}$ for the Normal Distribution

Assuming the observation data $\boldsymbol{\beta}_i\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Lambda})$, $1\le i\le I$, we have $u_{\boldsymbol{\mu}}(\boldsymbol{\beta}_i)=\boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}_i-\boldsymbol{\mu})$ and

$$f_{\boldsymbol{\mu}}^{\alpha_g}(\boldsymbol{\beta}_i)=(2\pi)^{-\alpha_g p/2}|\boldsymbol{\Lambda}|^{-\alpha_g/2}\exp\left\{-\frac{\alpha_g}{2}(\boldsymbol{\beta}_i-\boldsymbol{\mu})^{\top}\boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}_i-\boldsymbol{\mu})\right\},$$

where the lower index of the power parameter α_g refers to the group parameter. In this case, the integral term of Equation (6) would vanish and the estimating equation is simplified as

$$U_I(\boldsymbol{\mu}):=\frac{1}{I}\sum_{i=1}^{I}u_{\boldsymbol{\mu}}(\boldsymbol{\beta}_i)f_{\boldsymbol{\mu}}^{\alpha_g}(\boldsymbol{\beta}_i)=0, \tag{12}$$

The MDPDE is

$$\hat{\boldsymbol{\mu}}_{\text{MDPDE}}=\Omega^{-1}\sum_{i=1}^{I}\omega_i\boldsymbol{\beta}_i, \tag{13}$$

where $\Omega=\sum_{i=1}^{I}\omega_i$ and ω_i is the weight for $\boldsymbol{\beta}_i$

$$\omega_i=\exp\left[-\frac{\alpha_g}{2}\left(\boldsymbol{\beta}_i-\hat{\boldsymbol{\mu}}\right)^{\top}\boldsymbol{\Lambda}^{-1}\left(\boldsymbol{\beta}_i-\hat{\boldsymbol{\mu}}\right)\right].$$

Since $\hat{\boldsymbol{\mu}}_{\text{MDPDE}}$ also appears in ω_i, we used an iterative algorithm on (13) to find the solution for $\hat{\boldsymbol{\mu}}$. We use the sample mean as the initial value, and it typically takes fewer than 10 iterations to achieve convergence.

Parallel to the analogy of interpreting $\hat{\boldsymbol{\beta}}_{\text{MDPDE}}$ as the MLE of a heteroscedastic linear regression model, we can similarly treat $\hat{\boldsymbol{\mu}}_{\text{MDPDE}}$ as the MLE of the following statistical model:

$$\boldsymbol{\beta}_i\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Lambda_i}),\quad \boldsymbol{\Lambda}_i=(I\Omega_h\omega_i^{-1})\boldsymbol{\Lambda}, \tag{14}$$

where $\Omega_h=1/(\sum_{i=1}^{I}\omega_i^{-1})$ and $1\le i\le I$. In this definition, $\sum_{i=1}^{I}\boldsymbol{\Lambda_i}=I\boldsymbol{\Lambda}$, aligning with the summation of the covariance matrix in the i.i.d. model for $\boldsymbol{\beta_i}$, where $1\le i\le I$. Here, ω_i represents

the weight assigned to the ith atom in contributing to the group mean estimate, capturing the variability across group members in this hierarchical structure. This formulation accommodates heterogeneity among member-specific parameters while leveraging the robustness of the MDPDE approach.

[2] developed the asymptotic properties for the minimum density power divergence estimate (MDPDE) under a more general framework. Here, with the specific normal distribution model, we can directly establish the consistency property by showing that $\lim_{I\to\infty} \hat{\boldsymbol{\mu}}_{\text{MDPDE}} = \boldsymbol{\mu}$ if $\boldsymbol{\beta}_i$'s are i.i.d. with density function $f_{\boldsymbol{\mu}}(\boldsymbol{\beta})$. We provide the proof in Appendix A.3.2.

The MDPDE for $\boldsymbol{\Lambda}$ is given by

$$\hat{\boldsymbol{\Lambda}}_{\text{MDPDE}} = \frac{\sum_{i=1}^{I} \omega_i (\boldsymbol{\beta}_i - \boldsymbol{\mu})(\boldsymbol{\beta}_i - \boldsymbol{\mu})^\top}{\Omega - I\alpha_g (1+\alpha_g)^{-1-\frac{p}{2}}}, \tag{15}$$

where the technical derivation is in Appendix (A.3.3). For the ith atom, we need to calculate individual $\boldsymbol{\Lambda}_i$ in Model (14). We substitute $\hat{\boldsymbol{\Lambda}}_i$ for $\boldsymbol{\Lambda}_i$ in Equation (4) for the group estimate $\hat{\boldsymbol{\mu}}$, where

$$\hat{\boldsymbol{\Lambda}}_i = (I\Omega_h \omega_i^{-1}) \hat{\boldsymbol{\Lambda}}_{\text{MDPDE}}. \tag{16}$$

2.3 Algorithms

We present three algorithms here. In Algorithm 1, we solve the MDPDE for $\boldsymbol{\beta}_i$'s, and the output includes $\hat{\boldsymbol{\beta}}_i^{(\text{MDPDE})}$'s to be used in Algorithms 2, and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}_i}$'s to be used in Algorithm 3. In Algorithm 2, we solve the MDPDE for $\boldsymbol{\mu}$, and its output includes the covariance matrix $\boldsymbol{\Lambda}_i$'s to be used in Algorithm 3. In Algorithm 3, we calculate the marginal likelihood estimate $\hat{\boldsymbol{\mu}}$.

Algorithm 1 MDPDE for $\boldsymbol{\beta}$

Input: α_r, $\boldsymbol{y}_i$, $\boldsymbol{X}_i$, $1 \le i \le I$.
Output: $\hat{\boldsymbol{\beta}}_i^{(\text{MDPDE})}$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}_i}$, $1 \le i \le I$.

```
for i ← 1 to I do
    β̂_i^(0) ← (X_i^⊤ X_i)^{-1} X_i^⊤ y_i
    σ̂_i^{2(0)} ← (n − 1)^{-1} (y_i − X_i β̂_i^(0))^⊤ (y_i − X_i β̂_i^(0))
    while t ≤ N and (S^(t) < S^(t−1) with tolerance 10^{-5}) do
        w_ij^(t) ← exp{ −α_r (2σ̂_i^{2(t−1)})^{-1} (y_ij − x_ij^⊤ β̂_i^(t−1))^2 }        ▷ ∀j = 1 ··· n to build W_i^(t)
        β̂_i^(t) ← (X_i^⊤ W_i^(t) X_i)^{-1} (X_i^⊤ W_i^(t) y_i)
        σ̂_i^{2(t)} ← [tr(W_i^(t)) − nα_r(1 + α_r)^{-3/2}]^{-1} (y_i − X_i β̂_i^(t))^⊤ W_i^(t) (y_i − X_i β̂_i^(t))
        S^(t) ← (β̂_i^(t) − β̂_i^(t−1))^⊤ (β̂_i^(t) − β̂_i^(t−1))
    end while
    β̂_i^(MDPDE) ← β̂_i^(t)
    Σ̂_{y_i} ← nσ̂_i^{2(t)} (W_i^(t))^{-1} / tr((W_i^(t))^{-1})
end for
```

Algorithm 2 MDPDE for $\boldsymbol{\mu}$

Input: α_g, $\hat{\boldsymbol{\beta}}_i^{(\mathrm{MDPDE})}$, $1 \le i \le I$.
Output: $\hat{\boldsymbol{\Lambda}}_i$, $1 \le i \le I$.

$\hat{\boldsymbol{\mu}}^{(0)} \leftarrow \mathrm{median}\left\{\hat{\boldsymbol{\beta}}_1^{(\mathrm{MDPDE})} \cdots \hat{\boldsymbol{\beta}}_I^{(\mathrm{MDPDE})}\right\}$
$\hat{\boldsymbol{\Lambda}}^{(0)} \leftarrow \boldsymbol{I}_p$
while $t \le N$ **and** ($S^{(t)} < S^{(t-1)}$ **with tolerance** 10^{-5}) **do**
  $\omega_i^{(t)} \leftarrow \exp\left[-\frac{\alpha_g}{2}\left(\hat{\boldsymbol{\beta}}_i^{(\mathrm{MDPDE})} - \hat{\boldsymbol{\mu}}^{(t-1)}\right)^\top \left(\hat{\boldsymbol{\Lambda}}^{(t-1)}\right)^{-1}\left(\hat{\boldsymbol{\beta}}_i^{(\mathrm{MDPDE})} - \hat{\boldsymbol{\mu}}^{(t-1)}\right)\right]$ ▷ $\forall i = 1 \cdots I$
  $\Omega^{(t)} \leftarrow \sum_{i=1}^I \omega_i^{(t)}$
  $\hat{\boldsymbol{\mu}}^{(t)} \leftarrow \sum_{i=1}^I \omega_i^{(t)} {\Omega^{(t)}}^{-1} \hat{\boldsymbol{\beta}}_i^{(\mathrm{MDPDE})}$
  $\hat{\boldsymbol{\Lambda}}^{(t)} \leftarrow \left[\Omega^{(t)} - I\alpha_g(1+\alpha_g)^{-1-p/2}\right]^{-1} \sum_{i=1}^I \omega_i^{(t)} \left(\hat{\boldsymbol{\beta}}_i^{(\mathrm{MDPDE})} - \hat{\boldsymbol{\mu}}^{(t)}\right)\left(\hat{\boldsymbol{\beta}}_i^{(\mathrm{MDPDE})} - \hat{\boldsymbol{\mu}}^{(t)}\right)^\top$
  $S^{(t)} \leftarrow \left(\hat{\boldsymbol{\mu}}^{(t)} - \hat{\boldsymbol{\mu}}^{(t-1)}\right)^\top \left(\hat{\boldsymbol{\mu}}^{(t)} - \hat{\boldsymbol{\mu}}^{(t-1)}\right)$
end while
  $\Omega_h^{(t)} \leftarrow 1/(\sum_{i=1}^I (\omega_i^{(t)})^{-1})$
$\hat{\boldsymbol{\Lambda}}_i \leftarrow I\Omega_h^{(t)}\left(\omega_i^{(t)}\right)^{-1} \hat{\boldsymbol{\Lambda}}^{(t)}$, $1 \le i \le I$.

Algorithm 3 To Calculate the Maximum Marginal Likelihood Estimate $\hat{\boldsymbol{\mu}}$

Input: $\boldsymbol{y}_i$, $\boldsymbol{X}_i$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}_i}$, $\hat{\boldsymbol{\Lambda}}_i$, $1 \le i \le I$.
Output: $\hat{\boldsymbol{\mu}}$.

$\boldsymbol{\Sigma}_{\boldsymbol{y}_i} \leftarrow \hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}_i}$
$\boldsymbol{\Lambda}_i \leftarrow \hat{\boldsymbol{\Lambda}}_i$
for $i \leftarrow$ **1 to** I **do**
  $\tilde{\boldsymbol{\Sigma}}_i \leftarrow \left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1} \boldsymbol{X}_i + \boldsymbol{\Lambda}_i^{-1}\right)^{-1}$
end for
$\hat{\boldsymbol{\mu}} \leftarrow \left[\sum_{i=1}^I \boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i\right)\right]^{-1}\left[\sum_{i=1}^I \boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{y}_i\right)\right]$

3 Numerical Studies

3.1 Simulation Study

In this simulation study, our goal is to compare the performance of robust and non-robust estimates within the hierarchical modeling framework (1), particularly under conditions involving outliers. Notably, the estimates can be distinguished by the power parameter α, where a positive value refers to a robust estimate and a zero value for the the non-robust. Here, the parameter α is of the power in Minimum Density Power Divergence Estimate (MDPDE).

In our hierarchical framework, we incorporate two levels of statistical models: the bottom level represents the regression model, and the top level integrates group-specific information. We introduce α_r to denote the power parameter at the regression level and α_g to signify the power parameter at the group integration level, where $\alpha_r = 0$ refers to the ordinary least squares estimate

and $\alpha_g = 0$ refers to the mean estimate. The bias of the estimate serves as the primary metric for comparing performance, to measure the impact from the outliers.

We begin by testing a linear regression model, comparing OLS with the MDPDE at $\alpha_r = 0.1$. The data are generated from a homoscedastic linear regression model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

with specified proportions of outliers. Here, the parameter vector $\boldsymbol{\beta}$ is set to be

$$\left(\log A - \frac{3}{2}\log(2\pi\tau^2),\ \frac{1}{\tau^2}\right)^\top,$$

where the designed matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ is constructed with each data point $\boldsymbol{x}_j$ given by $\left(1,\ -\frac{1}{2}r_j^2\right)^\top$, and parameters $(A, \tau) = (1.0, 0.5)$. The error terms are sampled from the Normal distribution $\epsilon_j \sim \mathcal{N}(0, \sigma_\epsilon^2)$. To investigate the effect of varying error levels, we test three values of σ_ϵ: $0.1D_{\max}$, $0.2D_{\max}$ and $0.3D_{\max}$, where $D_{\max} = 0.508$ is the peak of the 3D Gaussian density function, given by $A\phi(0; \tau^2)$. Outliers are generated under the same model specification, but with a modified parameter $\boldsymbol{\beta}_{\text{outlier}} = \left(\log A - \frac{3}{2}\log(2\pi\tau^2),\ 0\right)^\top$, which eliminates the relationship between the predictors and response. Data examples for the three error intensities with 10% of outliers are illustrated in Figure 1.

In each experiment, a sample of size 1000 is generated, and the experiment is repeated 100 times to calculate the bias and standard deviation of the estimates under varying outlier ratios, from 0 to 20%, of outliers. We compute $\hat{\boldsymbol{\mu}}$ from Equation (4), as obtained from Algorithm 3, and then derive the amplitude parameter $\hat{A}$ and the width parameter $\hat{\tau}$ as final estimates using the variable transformation $\hat{\boldsymbol{\mu}} = \left(\log \hat{A} - \frac{3}{2}\log(2\pi\hat{\tau}^2), \frac{1}{\hat{\tau}^2}\right)^\top$. The resulting performance, in terms of both bias and variability, is summarized in Figure 2.

Figure 2 shows that, MDPDE outperforms the OLS in the presence of outliers, and performs comparably to OLS when no outliers are present. For the 10% ratio of outliers, MDPDE with a small value of $\alpha_r = 0.1$ can well resist the outliers. As the ratio of outliers increase, in our simulation setting, MDPDE needs a larger value of α_r to resist the influences from the outliers, which is to be shown in Figure 7 in Appendix B.1.

Secondly, we compare the estimates for $\boldsymbol{\mu}$ with $\alpha_g = 0$ and $\alpha_g = 0.2$ for the top level of the hierarchical regression model in (1). Here, to facilitate comparison with real data applications, we generate the data according to the homoscedastic hierarchical regression model, but apply the random distribution on the amplitude parameter A_i and width τ_i directly rather than on $\boldsymbol{\beta_i}$. We have $A_i \sim \mathcal{N}_1(1, 0.05^2)$ and $\tau_i \sim \mathcal{N}_1(0.5, 0.025^2)$.

Here, outliers are introduced at the group level, meaning that for $i \in I_{\text{outliers}}$, a different distribution is applied to (A_i, τ_i), where I_{outliers} is a randomly selected subset from $\{1, \ldots, I\}$ with a specified proportion. We design two types of outliers: one by modifying the distribution of A_i and the other by modifying τ_i. Specifically, for outliers in the amplitude, we set $A_i \sim \mathcal{N}_1(1.5, 0.05^2)$ and keep the same distribution for τ_i as $\mathcal{N}_1(0.5, 0.025^2)$. For outliers in the width, we keep the same distribution for A_i as $\mathcal{N}_1(1, 0.05^2)$ and set $\tau_i \sim \mathcal{N}_1(1.00, 0.025^2)$.

In each experiment, we generate 100 groups ($I = 100$), with each group containing 1,000 data points ($n = 1000$). The experiment is repeated 100 times to calculate the bias and standard deviation of the estimates across varying outlier ratios, ranging from 0 to 20%. The resulting performance, in terms of both bias and variability, is summarized in Figure 3.

Figure 3 shows that, the robust estimator with $\alpha_g = 0.2$, outperforms the non-robust estimator with $\alpha_g = 0$ in the presence of outliers, and performs comparably well when no outliers are present. For the 10% ratio of outliers, the robust estimate can well resist the outliers. As the ratio of outliers increase, in our simulation setting, the estimator needs a larger value of α_g to resist the influences from the outliers, which is to be shown in Figure 8 of Appendix B.1.

An interesting observation from Figure 3 is that when outliers affect only the amplitude, the bias appears solely in the amplitude estimate. This is depicted by the solid circles, red for the robust estimator and blue for the non-robust estimator. The blue and red circles as for the bias of the width estimate in the right panel remain at zero, indicating that amplitude outliers do not impact width estimation for either robust or non-robust estimates. Conversely, when outliers influence the width parameter, bias emerges in the estimates for both the width and amplitude parameters, as shown by the empty squares of both colors across both panels.

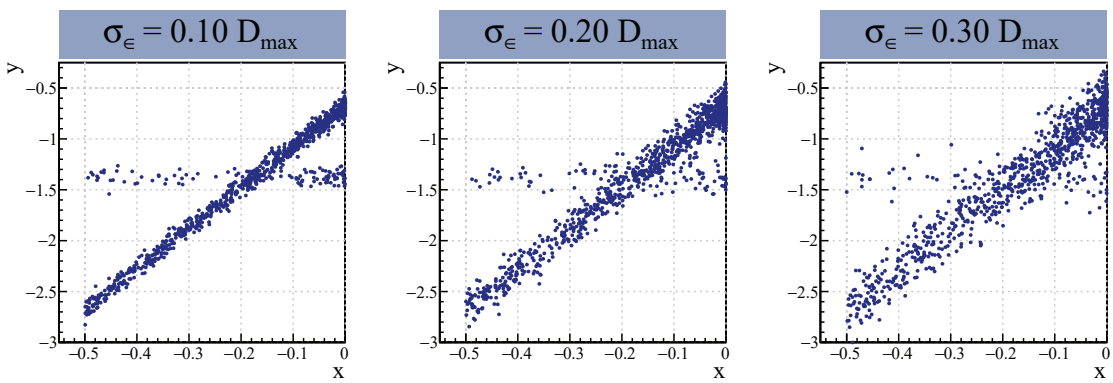
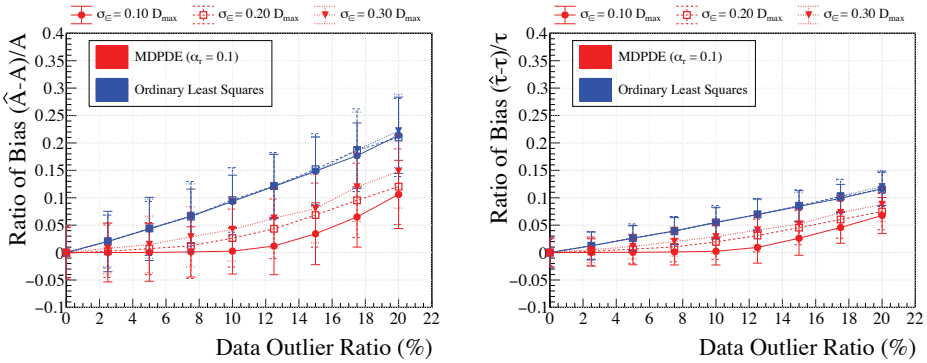


Figure 1: **Illustration of the Simulation Data:** We generate 900 data points following the model $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta} = \left(\log A - \frac{3}{2}\log(2\pi\tau^2),\ \frac{1}{\tau^2}\right)^\top$, and $(A, \tau) = (1.0, 0.5)$. $\boldsymbol{\epsilon}$ is sampled from the Normal distribution $\epsilon_j \sim \mathcal{N}(0, \sigma_\epsilon^2)$ with 3 intensities for σ_ϵ: $0.1D_{\max}$, $0.2D_{\max}$ and $0.3D_{\max}$, where $D_{\max} = 0.508 = A\phi(0; \tau^2)$. Additionally, 100 data points (10% of the dataset) are generated as outliers using the same model structure but with a modified parameter: $\boldsymbol{\beta}_{\text{outliers}} = \left(\log A - \frac{3}{2}\log(2\pi\tau^2),\ 0\right)^\top$, which removes the relationship between predictors and the response.

3.2 Real Data Analysis

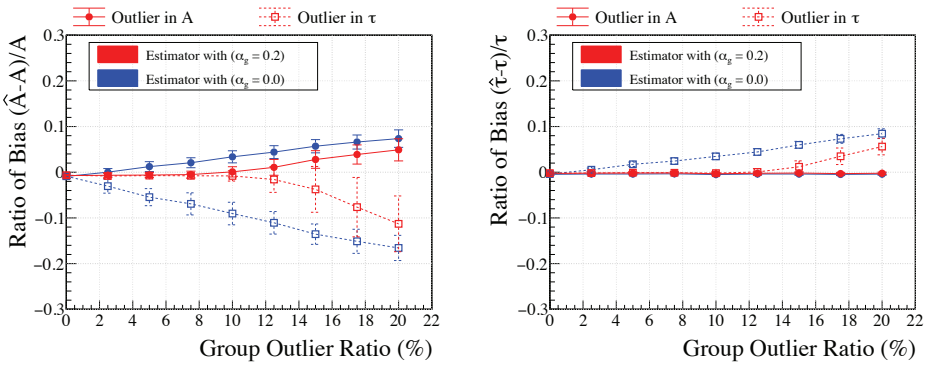
We apply our method to real data using paired information from EMD-11103 (EMDB) and PDB-6Z6U (PDB) of the cryo-EM benchmark protein apoferritin, which achieves a reported FSC resolution of 1.25 Å. PDB-6Z6U comprises 4,128 amino acids organized with 24-fold symmetry, yielding 172 independent atomic observations. Given that backbone atoms—including alpha carbon (C_α), carbon (C), and nitrogen (N)—are typically the first and most stable set to fit the atomic model, we focus on estimating these atoms to demonstrate the method's effectiveness. The frequency of repeats for each of the 20 amino acids in PDB-6Z6U is shown in Table 2.

For each of the three backbone atom types across the 20 amino acids, we calculate $\hat{\boldsymbol{\mu}}$ from Equation (4) as outlined in the three Algorithms in Subsection 2.3, with $\alpha_r = 0.1$ and $\alpha_g =$



(a) Amplitude (b) Width

Figure 2: **Bias comparisons of OLS and MDPDE in the presence of outliers.** Using the simulation data illustrated in Figure 1, we assessed the estimation performance of OLS and MDPDE. The bias ratios for amplitude (left panel) and width (right panel) are plotted across varying proportions of outliers on the x-axis. Both panels demonstrate that MDPDE consistently outperforms OLS across all cases. In this figure, MDPDE is represented in red and OLS in blue, with three different error intensities depicted by three shapes: solid circles, empty squares, and solid triangles.



(a) Amplitude (b) Width

Figure 3: **Bias comparisons of group parameter estimates in the presence of outliers.** Bias ratios for amplitude (left panel) and width (right panel) are plotted across varying proportions of outliers on the x-axis. Both panels demonstrate that the robust estimate ($\alpha_g = 0.2$) consistently outperforms non-robustness estimate ($\alpha_g = 0$) across all cases, except where the blue and red circles coincide in the right panel. We consider two types of outliers: those affecting amplitude (solid circles) by increasing its mean by 50%, and those affecting width (empty squares) by doubling its mean. The blue and red circles in the right panel remain at zero, indicating that amplitude outliers do not impact width estimation for either robust or non-robust estimates.

0.2. We then derive the amplitude estimator $\hat{A}$ and the width estimator $\hat{\tau}$ using the parameter transformation $\hat{\boldsymbol{\mu}} = \left(\log \hat{A} - \frac{3}{2}\log(2\pi\hat{\tau}^2), \frac{1}{\hat{\tau}^2}\right)^\top$.

Figure 4 displays amplitude estimates in the top panel and width estimates in the bottom panel for the three backbone atoms across 20 amino acids. The estimated parameters for these atoms appear stable across most amino acids. Notably, the ranking of both amplitude and width estimates are generally consistent across the amino acids. For the amplitude, the ranking follows the order of C as the highest, followed closely by C_α and then N, with two exceptions: N ranks highest in PRO and C_α surpasses C in TRP. For the width estimation, C_α and C appear to be even more closely aligned and N stays at bottom generally, with two exceptions similar to those observed in the amplitude estimates. Among the exceptions, the amplitude of C_α in GLY decreases by around 30%, aligning more closely with N rather than with C. This shift may relate to the unique characteristic of GLY, as its side chain contains only hydrogen atoms.

Figure 5 presents further details from our analysis of the C_α atom for three amino acids, ALA, GLU and SER. For each atom, the three-dimensional voxel data is reduced to one dimensional curve by taking the median of map values across grid points equidistant from the atom center, displayed as solid blue and red curves, with red used to highlight the identified outliers. Here, one prominent outlier curve for GLU and two for SER are highlighted. The red dashed curves represent the estimations generated by our method with parameters $\alpha_r = 0.1$ for the regression analysis at the bottom level of the hierarchical model, and $\alpha_g = 0.2$ for integrating $\hat{\boldsymbol{\beta}}_{\boldsymbol{i}}$'s to estimate $\hat{\boldsymbol{\mu}}$ at the top level. Meanwhile, the blue dotted lines correspond to estimations with $\alpha_r = \alpha_g = 0$.

When no outliers are present, the estimates from both methods align closely, as shown in the ALA example in Figure 5. However, in the presence of a single extreme outlier, as seen in the SER example in Figure Figure 5, the blue dotted curve deviates noticeably towards the outlier. In contrast, our method effectively resists this deviation, demonstrating robustness against outlier effects. The GLU example illustrates a moderate outlier scenario. Further analysis details for the C_α atom across all 20 amino acids are provided in Figure 9 in Appendix B.2.

In this analysis, the outliers are prominent and can be readily identified through visual inspection. Notably, the three outliers identified in Figure 5, all meet the criterion that the weight w_i/Ω contributing to the estimated red dashed curve (with $\alpha_g = 0.2$), is less than $0.05/I$, where $1/I$ represents the weight used in a direct average shown as the blue dot curve (with $\alpha_g = 0$). This consistency suggests a potential empirical criterion for identifying outliers when visualizing this data.

Given the observed stability of the estimated parameters across the 20 amino acids, we take a step further to estimate the parameters for the three atoms without differentiating by amino acid type. The results of estimating parameters for the three backbone atoms, using all 172 repeats from the amino acid structure ($I = 172$), are presented in Figure 6. We use $0.05/I$ as the threshold for weight contribution in the outlier-labeling strategy. For C_α, 7 curves are labeled as outliers, with 2 originating from GLY. GLY is distinctive in that its side chain contains only hydrogen atoms, suggesting that these outliers may not be due to data anomalies but rather reflect unique chemical properties.

The amplitude estimate for C_α is 102.75, for C it is 115.67, and for N, it is 80.12. The width estimate for C_α is 0.72, for C it is 0.73, and for N, it is 0.65, as shown in Figure 6. In addition to N having the lowest value for collectively estimating both the amplitude and width parameters, N consistently ranks lowest in both across most of the 20 amino acids shown in Figure 4. This contrasts with the expectations based on atomic numbers, where N would be anticipated to have

the highest amplitude. Moreover, this deviates from the standardized width model, where all three values would be expected to be the same. This unexpected pattern suggests that certain aspects of the commonly used model may require closer scrutiny to ensure it accurately captures the unique properties of each atom type.

Amino Acid	ALA	ARG	ASN	ASP	CYS	GLN	GLU	GLY	HIS	ILE
No. of repeats	12	7	11	13	2	11	15	7	10	6
Amino Acid	LEU	LYS	MET	PHE	PRO	SER	THR	TRP	TYR	VAL
No. of repeats	22	12	4	6	3	10	5	1	9	6

Table 2: Number of Amino Acids (I) in 20 Amino Acid Types in PDB-6Z6U

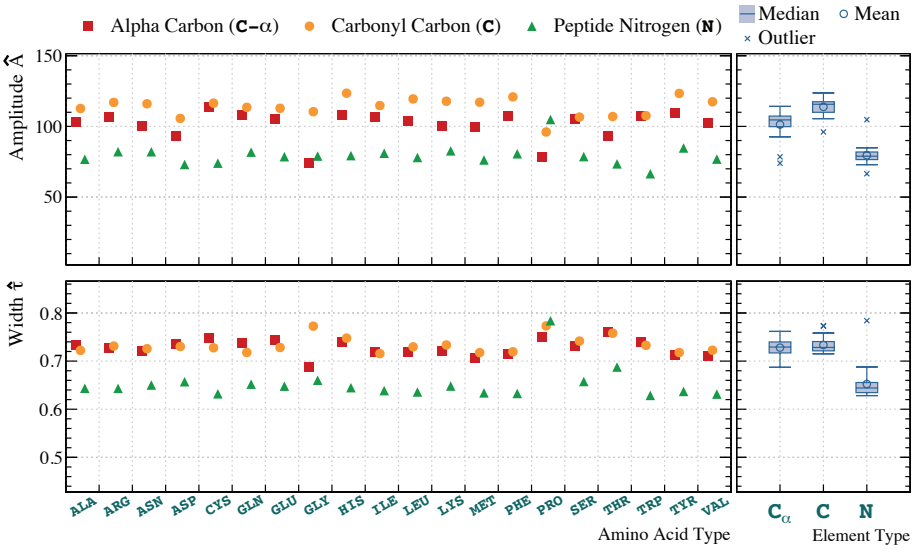


Figure 4: **Amplitude and width estimates for the paired data EMD-11103 and PDB-6Z6U.** The top panel displays the amplitude estimates, and the bottom panel shows the width estimates for the three backbone atoms across 20 amino acids, organized by column. In this figure, the estimates for C_α are shown as red squares, C as orange circles and N as green triangles. The estimated parameters appear stable across most amino acids, and box plots in the right panels summarize these estimates. Generally, amplitude estimates follow the order: C as the highest, followed closely by C_α and then N. For the width estimation, C_α and C appear to be even more closely aligned, with N consistently lower. Notable exceptions to these trends are observed in PRO and TRP.

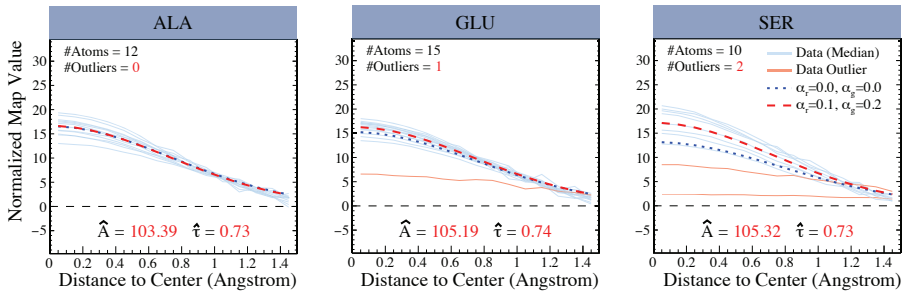


Figure 5: **Estimated Curves for C_α of three amino acids: ALA, GLU and SER from the paired data EMD-11103 and PDB-6Z6U.** For each atom, the three-dimensional voxel data is simplified to a one-dimensional curve by taking the median of map values across grid points equidistant from the atom center, shown as solid curves (blue and red) with red highlighting identified outliers. Red dashed curves represent the robust estimates with parameters $\alpha_g = 0.2$ and $\alpha_r = 0.1$, while blue dotted curves show the non-robust estimates with $\alpha_g = \alpha_r = 0$. Notably, one prominent outlier is highlighted for GLU and two for SER. An outlier is identified when the weight for the ith curve contributing to the robust estimate is below $0.05/I$. When no outliers are present, as in the ALA example, both methods produce closely aligned estimates. However, with two outliers, including one extreme case (SER), the non-robust curve noticeably deviates toward the outlier. In contrast, the robust curve effectively resists this influence, illustrating its robustness against outliers. The GLU example presents a moderate outlier scenario.

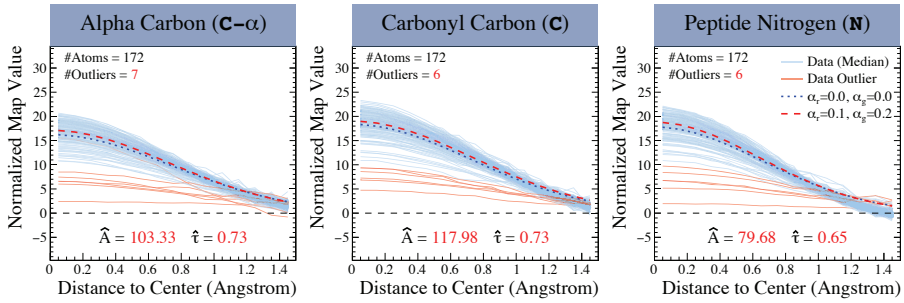


Figure 6: **Estimated Curves for C_α, C and N from the paired data EMD-11103 and PDB-6Z6U.** The results of estimating parameters for the three backbone atoms, using all 172 repeats from the amino acid structure ($I = 172$), are presented as curves. These figures are generated in the same manner as Figure 5, except that here all 172 repeats are used. We use $0.05/I$ as the threshold for weight contribution in the outlier-labeling strategy. The amplitude estimate is 102.75 for C_α, 115.67 for C and 80.12 for N. The width estimate is 0.72 for C_α, 0.73 for C and 0.65 for N.

4 Discussion

In this paper, we propose the hierarchical robust linear (HRL) model for atom-specific analysis of paired data from cryo-EM reconstruction maps and their corresponding atomic models. In this framework, $\boldsymbol{\mu}$ captures the common features shared by atoms within the same category, while the variability in $\boldsymbol{\beta}_i$ accounts for heterogeneity arising from differences in local environments. The real data application demonstrates the feasibility of this model, highlighting its potential for detailed analysis of atomic structures.

To enhance the robustness of our approach, we integrate minimum density power divergence estimates (MDPDE) into the likelihood framework. This is achieved by constructing the covariance matrix of the heteroscedastic model using the inverse of a diagonal matrix, where the diagonal elements represent the weights derived from the MDPDE. Both simulation and real data applications show that this approach effectively mitigates the impact of outliers—an important advantage of using MDPDE. Given outliers are nearly inevitable due to the extreme noise of the data and challenges associated with atomic model fitting, this method proves particularly valuable in ensuring robust and reliable parameter estimates.

Applying our method to a high-resolution cryo-EM map (PDB ID: 6Z6U, EMDB ID:11103), we find that the estimates for backbone atoms—typically considered the most reliable—do not align with the assumptions of the commonly used Gaussian model. In this Gaussian model, the amplitude is typically expected to be proportional to the atomic number, and the width parameter is standardized across different atom types. However, our results reveal that nitrogen atoms consistently exhibit lower amplitude estimates than anticipated, diverging from expectation based on

the atomic numbers. Additionally, the persistent discrepancy in width estimates suggests that the standardized approach may fail to fully account for the unique properties of different atom types. These findings underscore the need for a more systematic analysis of paired cryo-EM and atomic data from the PDB and EMDB.

Although the primary motivation for developing our method is to analyze cryo-EM map data paired with its atomic model, it is perceived that our method is applicable to other problems characterized by hierarchical structures. These structures often involve a central entity (the "head") connected to multiple parallel groups (nodes), such as the integration of medical data from multiple hospitals. The HRL model provides a flexible and robust framework for addressing such problems, extending its applicability beyond cryo-EM analysis.

References

[1] P. V. Afonine, B. K. Poon, R. J. Read, O. V. Sobolev, T. C. Terwilliger, A. Urzhumtsev, and P. D. Adams. Real-space refinement in phenix for cryo-em and crystallography. *Acta Crystallographica Section D*, 74(6):531–544, 2018.

[2] A. Basu, I. R. Harris, N. L. Hjort, and M. Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

[3] T.-L. Chen, D.-N. Hsieh, H. Hung, I.-P. Tu, P.-S. Wu, Y.-M. Wu, W.-H. Chang, and S.-Y. Huang. γ-sup: A clustering algorithm for cryo-electron microscopy images of asymmetric particles. *The Annals of Applied Statistics*, 8(1):259–285, 2014.

[4] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, August 1995.

[5] H. Fujisawa and S. Eguchi. Robust parameter estimation with a small bias against heavy contamination. *J. Multivariate Anal.*, 99:2053–2081, 2008.

[6] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2013.

[7] A. Ghosh and A. Basu. Robust estimation for independent non-homogeneous observations using density power divergence with applications to linear regression. *Electronic Journal of Statistics*, 7:2420–2456, 2013.

[8] D. A. Harville. *Matrix Differentiation*, pages 289–335. Springer New York, New York, NY, 1997.

[9] J. He, T. Li, and S.-Y. Huang. Improvement of cryo-em maps by simultaneous local and non-local deep learning. *Nature Communications*, 14:3217, 2023.

[10] G. Pintilie, K. Zhang, Z. Su, S. Li, M. F. Schmid, and W. Chiu. Measurement of atom resolvability in cryo-em maps with q-scores. *Nature methods*, 17(3):328—334, March 2020.

[11] M. Wang, L. Jiang, R. Jian, J. Y. Chan, Q. Liu, M. P. Snyder, and H. Tang. RobNorm: model-based robust normalization method for labeled quantitative mass spectrometry proteomics data. *Bioinformatics*, 37(6):815—821, March 2021.

[12] M. P. Windham. Robustifying Model Fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(3):599–609, 1995.

A For Method Section

A.1 Statistical Model

The physics model for the cryo-EM map value on the jth sampling grid point of the ith atom is $A_i\phi(r_j;\tau_i^2)$ where A_i is the amplitude parameter for the ith atom, $\phi(r_{ij};\tau_i^2)$ is the 3D Normal density function for the ith atom with mean 0 and width parameter τ_i and r_{ij} is the distance from the jth sampling grid point to the ith atom center. After taking log transformation, we build a linear regression model as

$$\begin{aligned} y_{ij} =& \log[\phi_(r_{ij};\tau_i^2)A_i] + \epsilon_{ij} \\ =& \log[(2\pi\tau_i^2)^{\frac{-3}{2}} \exp\left(-\frac{1}{2\tau_i^2}r_{ij}^2\right) A_i] + \epsilon_{ij} \\ =& \left[-\frac{3}{2}\log(2\pi\tau_i^2) + \log A_i\right] + \frac{1}{\tau_i^2}\left[-\frac{1}{2}r_{ij}^2\right] + \epsilon_{ij} \\ =& \boldsymbol{x}_{ij}^\top\boldsymbol{\beta}_i + \epsilon_{ij} \end{aligned} \tag{17}$$

where $1 \le j \le n_i$ with n_i being the sampling size for the ith atom, $1 \le i \le I$, $\boldsymbol{\beta}_i = \left(\log A_i - \frac{3}{2}\log(2\pi\tau_i^2),\ \frac{1}{\tau_i^2}\right)^\top$, $\boldsymbol{x}_{ij} = \left(1,\ -\frac{1}{2}r_{ij}^2\right)^\top$ with r_{ij} being the distance of the jth sampling grid point to the atom center. We further let $\boldsymbol{X}_i = (\boldsymbol{x}_{i1},\ldots,\boldsymbol{x}_{in_i})^\top$ to have the vector form for this linear model as

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i,$$

that the length of $\boldsymbol{y}_i$ is n_i.

A.2 Marginal Likelihood Estimates

The marginal likelihood for $\boldsymbol{\mu}$ given $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_I$ based on Model (1) is as follows.

$$\begin{aligned} & Lik\left(\boldsymbol{\mu}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_I\right) = f\left(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_I|\boldsymbol{\mu}\right) \\ =& \int f\left(\boldsymbol{y}_1|\boldsymbol{\beta}_1\right)\cdots f\left(\boldsymbol{y}_I|\boldsymbol{\beta}_I\right) f\left(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_I|\boldsymbol{\mu}\right) d\boldsymbol{\beta}_1\cdots\ d\boldsymbol{\beta}_I \\ \propto& \Pi_{i=1}^I\left(\int e^{-\frac{1}{2}(\boldsymbol{y}_i-\boldsymbol{X}_i\boldsymbol{\beta}_i)^\top\boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}(\boldsymbol{y}_i-\boldsymbol{X}_i\boldsymbol{\beta}_i)}\cdot e^{-\frac{1}{2}(\boldsymbol{\beta}_i-\boldsymbol{\mu})^\top\boldsymbol{\Lambda}_i^{-1}(\boldsymbol{\beta}_i-\boldsymbol{\mu})}d\boldsymbol{\beta}_i\right) \\ \propto& \Pi_{i=1}^I\left(\int e^{-\frac{1}{2}(\boldsymbol{\beta}_i-\tilde{\boldsymbol{\beta}}_i)^\top\tilde{\boldsymbol{\Sigma}}_i^{-1}(\boldsymbol{\beta}_i-\tilde{\boldsymbol{\beta}}_i)}d\boldsymbol{\beta}_i\cdot\exp\left\{-\frac{1}{2}\boldsymbol{y}_i^\top\boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{y}_i - \frac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Lambda}_i^{-1}\boldsymbol{\mu} + \frac{1}{2}\tilde{\boldsymbol{\beta}}_i^\top\tilde{\boldsymbol{\Sigma}}_i^{-1}\tilde{\boldsymbol{\beta}}_i\right\}\right) \\ \propto& \Pi_{i=1}^I\left(\exp\left\{-\frac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Lambda}_i^{-1}\boldsymbol{\mu} + \frac{1}{2}\tilde{\boldsymbol{\beta}}_i^\top\tilde{\boldsymbol{\Sigma}}_i^{-1}\tilde{\boldsymbol{\beta}}_i\right\}\right) \\ \propto& \Pi_{i=1}^I\left(\exp\left\{-\frac{1}{2}\boldsymbol{\mu}^\top\left(\boldsymbol{\Lambda}_i^{-1} - \boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\boldsymbol{\Lambda}_i^{-1}\right)\boldsymbol{\mu} + \boldsymbol{\mu}^\top\left[\boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top\boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{y}_i\right)\right]\right\}\right), \end{aligned} \tag{18}$$

where $\tilde{\boldsymbol{\beta}}_i = \tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top\boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{y}_i + \boldsymbol{\Lambda}_i^{-1}\boldsymbol{\mu}\right)$ and $\tilde{\boldsymbol{\Sigma}}_i = \left(\boldsymbol{X}_i^\top\boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i + \boldsymbol{\Lambda}_i^{-1}\right)^{-1}$.

Let $Q(\boldsymbol{\mu}) = \sum_{i=1}^{I} -\frac{1}{2}\boldsymbol{\mu}^\top \left(\boldsymbol{\Lambda}_i^{-1} - \boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\boldsymbol{\Lambda}_i^{-1}\right)\boldsymbol{\mu} + \boldsymbol{\mu}^\top \left[\boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{y}_i\right)\right]$, then taking the derivative of $Q(\boldsymbol{\mu})$, we have

$$\frac{dQ(\boldsymbol{\mu})}{d\boldsymbol{\mu}} = \sum_{i=1}^{I} -\tilde{\boldsymbol{\Lambda}}_i^{-1}\boldsymbol{\mu} + \boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{y}_i\right),$$

where

$$\begin{aligned}
\tilde{\boldsymbol{\Lambda}}_i^{-1} &= \boldsymbol{\Lambda}_i^{-1} - \boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\boldsymbol{\Lambda}_i^{-1} \\
&= \boldsymbol{\Lambda}_i^{-1}(\boldsymbol{\Lambda}_i - \left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i + \boldsymbol{\Lambda}_i^{-1}\right)^{-1})\boldsymbol{\Lambda}_i^{-1} \\
&= \boldsymbol{\Lambda}_i^{-1}\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i + \boldsymbol{\Lambda}_i^{-1}\right)^{-1}\left((\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i + \boldsymbol{\Lambda}_i^{-1})\boldsymbol{\Lambda}_i - I_2\right)\boldsymbol{\Lambda}_i^{-1} \\
&= \boldsymbol{\Lambda}_i^{-1}\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i + \boldsymbol{\Lambda}_i^{-1}\right)^{-1}\left((\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i\boldsymbol{\Lambda}_i + I_2) - I_2\right)\boldsymbol{\Lambda}_i^{-1} \\
&= \boldsymbol{\Lambda}_i^{-1}\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i + \boldsymbol{\Lambda}_i^{-1}\right)^{-1}\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i\right) \\
&= \boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i).
\end{aligned} \tag{19}$$

Then, the MLE by solving $\frac{dQ(\boldsymbol{\mu})}{d\boldsymbol{\mu}} = 0$ is as follows.

$$\begin{aligned}
\hat{\boldsymbol{\mu}} &= \left[\sum_{i=1}^{I}\tilde{\boldsymbol{\Lambda}}_i^{-1}\right]^{-1}\left[\sum_{i=1}^{I}\boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{y}_i\right)\right] \\
&= \left[\sum_{i=1}^{I}\boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{X}_i\right)\right]^{-1}\left[\sum_{i=1}^{I}\boldsymbol{\Lambda}_i^{-1}\tilde{\boldsymbol{\Sigma}}_i\left(\boldsymbol{X}_i^\top \boldsymbol{\Sigma}_{\boldsymbol{y}_i}^{-1}\boldsymbol{y}_i\right)\right]
\end{aligned} \tag{20}$$

A.3 Weights

A.3.1 Weights for Liner Regression

To estimate σ^2, we have $u_\sigma(z_j) = \frac{\partial}{\partial\sigma}\log(f_{\sigma(z_j)}) = \frac{\partial}{\partial\sigma}(-z_j^2/(2\sigma^2) - \log(\sigma)) = z_j^2/\sigma^3 - 1/\sigma$.

$$\int u(z)f^{1+\alpha}(z)dz = \left(2\pi\sigma^2\right)^{\frac{-1-\alpha}{2}}\int\left(\frac{z^2}{\sigma^3} - \frac{1}{\sigma}\right)\exp\left[-\frac{(1+\alpha)z^2}{2\sigma^2}\right]dz = -\left(2\pi\sigma^2\right)^{\frac{-1-\alpha}{2}}\frac{\sqrt{2\pi}\alpha}{(1+\alpha)^{3/2}}$$

The estimating equation (6) becomes

$$\frac{1}{n}\sum_{j=1}^{n}\left(\frac{z_j^2}{\sigma^2} - 1\right)\exp\left\{-\frac{\alpha z_j^2}{2\sigma^2}\right\} + \frac{\alpha}{(1+\alpha)^{3/2}} = 0, \tag{21}$$

and the MDPDE of σ^2 is as follows.

$$\hat{\sigma}^2 = \left[\mathrm{tr}\left(\boldsymbol{W}\right) - \frac{n\alpha}{(1+\alpha)^{3/2}}\right]^{-1}\left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^\top \boldsymbol{W}\left(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}\right) \tag{22}$$

A.3.2 Consistency for $\hat{\boldsymbol{\mu}}_{\text{MDPDE}}$

Let $X_I = \frac{1}{I}\sum_{i=1}^{I} \boldsymbol{\beta}_i f_{\boldsymbol{\mu}}^{\alpha}(\boldsymbol{\beta}_i)$ and $Y_I = \frac{1}{I}\sum_{j=1}^{I} f_{\boldsymbol{\mu}}^{\alpha}(\boldsymbol{\beta}_j)$, then we have

$$\lim_{I\to\infty} \hat{\boldsymbol{\mu}}_{\text{MDPDE}} = \lim_{I\to\infty} \frac{\sum_{i=1}^{I} \omega_i \boldsymbol{\beta}_i}{\sum_{i=1}^{I} \omega_i} = \lim_{I\to\infty} \frac{\frac{1}{I}\sum_{i=1}^{I} \boldsymbol{\beta}_i f_{\boldsymbol{\mu}}^{\alpha}(\boldsymbol{\beta}_i)}{\frac{1}{I}\sum_{j=1}^{I} f_{\boldsymbol{\mu}}^{\alpha}(\boldsymbol{\beta}_j)} = \lim_{I\to\infty} \frac{X_I}{Y_I}.$$

Given that

$$\lim_{I\to\infty} X_I = E_{f_\mu}[\boldsymbol{\beta} f_{\boldsymbol{\mu}}^{\alpha}(\boldsymbol{\beta})] = \int \boldsymbol{\beta} f_{\boldsymbol{\mu}}^{1+\alpha}(\boldsymbol{\beta}) d\boldsymbol{\beta}$$
$$\lim_{I\to\infty} Y_I = E_{f_\mu}[f_{\boldsymbol{\mu}}^{\alpha}(\boldsymbol{\beta})] = \int f_{\boldsymbol{\mu}}^{1+\alpha}(\boldsymbol{\beta}) d\boldsymbol{\beta},$$

we can then apply Slutsky's Theorem which is provided in the end of this section, to get

$$\lim_{I\to\infty} \hat{\boldsymbol{\mu}}_{\text{MDPDE}} = \lim_{I\to\infty} \frac{X_I}{Y_I} = \frac{\lim_{I\to\infty} X_I}{\lim_{I\to\infty} Y_I} = \frac{\int \boldsymbol{\beta} f_{\boldsymbol{\mu}}^{1+\alpha}(\boldsymbol{\beta}) d\boldsymbol{\beta}}{\int f_{\boldsymbol{\mu}}^{1+\alpha}(\boldsymbol{\beta}) d\boldsymbol{\beta}} = \boldsymbol{\mu}. \tag{23}$$

Slutsky's Theorem
Let X_n and Y_n be sequences of scalar or vector or matrix random elements. If X_n converges in distribution to a random element X and Y_n converges in probability to a constant c, then

$$X_n + Y_n \to X + c$$
$$X_n * Y_n \to X * c$$
$$X_n / Y_n \to X/c \text{ if c is invertible}$$

A.3.3 The MDPDE for estimating the covariance of Normal distribution Λ

For $\boldsymbol{\beta}_i \sim N(\boldsymbol{\mu}, \boldsymbol{\Lambda})$, $1 \le i \le I$, to estimate $\boldsymbol{\Lambda}$, we follow Equation (6) to estimate $\boldsymbol{\Lambda}$ by derive $u_{\Lambda_k}(\boldsymbol{\beta})$ first, where Λ_k is the kth element of $\boldsymbol{\Lambda}$. The maximum likelihood score function of Λ_k is

$$u_{\Lambda_k}(\boldsymbol{\beta}) = \frac{\partial \log f_{\boldsymbol{\Lambda}}(\boldsymbol{\beta})}{\partial \Lambda_k} = -\frac{1}{2}\frac{\partial}{\partial \Lambda_k}\left[\log|\boldsymbol{\Lambda}| + (\boldsymbol{\beta}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})\right] \tag{24}$$

Here we employ the matrix differentiation by [8]

$$\begin{cases} \frac{\partial \log|\boldsymbol{F}|}{\partial x_j} = \operatorname{tr}\left(\boldsymbol{F}^{-1}\frac{\partial \boldsymbol{F}}{\partial x_j}\right) \\ \frac{\partial \boldsymbol{F}^{-1}}{\partial x_j} = -\boldsymbol{F}^{-1}\frac{\partial \boldsymbol{F}}{\partial x_j}\boldsymbol{F}^{-1} \end{cases} \tag{25}$$

$$\begin{aligned} u_{\Lambda_k}(\boldsymbol{\beta}) &= -\frac{1}{2}\operatorname{tr}\left(\boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_k}\right) + \frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_k}\boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}) \\ &= \frac{1}{2}\operatorname{tr}\left\{\boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})(\boldsymbol{\beta}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_k} - \boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_k}\right\} \\ &= \frac{1}{2}\operatorname{tr}\left\{\left[\boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})(\boldsymbol{\beta}-\boldsymbol{\mu})^\top - \boldsymbol{I}_p\right]\boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_k}\right\} \end{aligned} \tag{26}$$

Now the estimation equation of Λ_k is given as follows.

$$U_I(\Lambda_k) := \frac{1}{I}\sum_{i=1}^{I} u_{\Lambda_k}(\boldsymbol{\beta}_i) f^{\alpha}_{\Lambda_k}(\boldsymbol{\beta}_i) - \int u_{\Lambda_k}(\boldsymbol{\beta}) f^{1+\alpha}_{\Lambda_k}(\boldsymbol{\beta}) \mathrm{d}\boldsymbol{\beta} = 0, \quad \forall k = 1 \cdots p \times p \tag{27}$$

Since $\boldsymbol{\Lambda}$ and Λ_k do not depend on $\boldsymbol{\beta}$, the integral term in (27) involves the following two integrals (28) and (29).

$$\begin{aligned}
&\int (\boldsymbol{\beta}-\boldsymbol{\mu})(\boldsymbol{\beta}-\boldsymbol{\mu})^\top f^{1+\alpha}(\boldsymbol{\beta})\mathrm{d}\boldsymbol{\beta} \\
&= (2\pi)^{\frac{-p(1+\alpha)}{2}} |\boldsymbol{\Lambda}|^{\frac{-1-\alpha}{2}} \int (\boldsymbol{\beta}-\boldsymbol{\mu})(\boldsymbol{\beta}-\boldsymbol{\mu})^\top e^{-\frac{1+\alpha}{2}(\boldsymbol{\beta}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})}\mathrm{d}\boldsymbol{\beta} \\
&= \left[(2\pi)^p |\boldsymbol{\Lambda}|\right]^{-\alpha/2} \frac{|\boldsymbol{\Lambda}_\alpha|^{1/2}}{|\boldsymbol{\Lambda}|^{1/2}} \int (2\pi)^{-p/2} |\boldsymbol{\Lambda}_\alpha|^{-1/2} (\boldsymbol{\beta}-\boldsymbol{\mu})(\boldsymbol{\beta}-\boldsymbol{\mu})^\top e^{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}_\alpha^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})}\mathrm{d}\boldsymbol{\beta} \\
&= \left[(2\pi)^p |\boldsymbol{\Lambda}|\right]^{-\alpha/2} (1+\alpha)^{-p/2} \boldsymbol{\Lambda}_\alpha \\
&= \left[(2\pi)^p |\boldsymbol{\Lambda}|\right]^{-\alpha/2} (1+\alpha)^{-1-\frac{p}{2}} \boldsymbol{\Lambda},
\end{aligned} \tag{28}$$

where $\boldsymbol{\Lambda}_\alpha = \boldsymbol{\Lambda}/(1+\alpha)$, and

$$\begin{aligned}
&\int \boldsymbol{I}_p f^{1+\alpha}(\boldsymbol{z})\mathrm{d}\boldsymbol{\beta} \\
&= (2\pi)^{\frac{-p(1+\alpha)}{2}} |\boldsymbol{\Lambda}|^{\frac{-1-\alpha}{2}} \int \boldsymbol{I}_p e^{-\frac{1+\alpha}{2}(\boldsymbol{\beta}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})}\mathrm{d}\boldsymbol{\beta} \\
&= \left[(2\pi)^p |\boldsymbol{\Lambda}|\right]^{-\alpha/2} \frac{|\boldsymbol{\Lambda}_\alpha|^{1/2}}{|\boldsymbol{\Lambda}|^{1/2}} \int \boldsymbol{I}_p (2\pi)^{-p/2} |\boldsymbol{\Lambda}_\alpha|^{-1/2} e^{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}_\alpha^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})}\mathrm{d}\boldsymbol{\beta} \\
&= \left[(2\pi)^p |\boldsymbol{\Lambda}|\right]^{-\alpha/2} (1+\alpha)^{-p/2} \boldsymbol{I}_p.
\end{aligned} \tag{29}$$

By (28) and (29), we have the integral term in (27) as follows.

$$\begin{aligned}
&\int u_{\Lambda_i}(\boldsymbol{\beta}) f^{1+\alpha}(\boldsymbol{\beta})\mathrm{d}\boldsymbol{\beta} \\
&= \int \frac{1}{2}\mathrm{tr}\left\{\left[\boldsymbol{\Lambda}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})(\boldsymbol{\beta}-\boldsymbol{\mu})^\top - \boldsymbol{I}_p\right]\boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_i}\right\} f^{1+\alpha}(\boldsymbol{\beta})\mathrm{d}\boldsymbol{\beta} \\
&= \frac{1}{2}\mathrm{tr}\left\{\left[\boldsymbol{\Lambda}^{-1}\int (\boldsymbol{\beta}-\boldsymbol{\mu})(\boldsymbol{\beta}-\boldsymbol{\mu})^\top f^{1+\alpha}(\boldsymbol{\beta})\mathrm{d}\boldsymbol{\beta} - \int \boldsymbol{I}_p f^{1+\alpha}(\boldsymbol{\beta})\mathrm{d}\boldsymbol{\beta}\right]\boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_i}\right\} \\
&= \frac{1}{2}\left[(2\pi)^p |\boldsymbol{\Lambda}|\right]^{-\alpha/2} \mathrm{tr}\left\{\left[(1+\alpha)^{-1-\frac{p}{2}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda} - (1+\alpha)^{-p/2}\boldsymbol{I}_p\right]\boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_i}\right\} \\
&= -\frac{\alpha}{2}\left[(2\pi)^p |\boldsymbol{\Lambda}|\right]^{-\alpha/2} (1+\alpha)^{-1-\frac{p}{2}} \mathrm{tr}\left(\boldsymbol{I}_p \boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_i}\right)
\end{aligned} \tag{30}$$

Now, the estimating equation (27) with the score function (26) and the integral term in (30), we

have

$$\frac{1}{I}\sum_{i=1}^{I} u_{\Lambda_i}(\boldsymbol{\beta}_i) f_{\Lambda_i}^{\alpha}(\boldsymbol{\beta}_i) - \int u_{\Lambda_i}(\boldsymbol{\beta}) f_{\Lambda_i}^{1+\alpha}(\boldsymbol{\beta}) \mathrm{d}\boldsymbol{\beta} = 0$$

$$\Rightarrow \operatorname{tr}\left\{\left\{\sum_{i=1}^{I}(\boldsymbol{\beta}_i - \boldsymbol{\mu})(\boldsymbol{\beta}_i - \boldsymbol{\mu})^\top \omega_i \boldsymbol{\Lambda}^{-1} - \left[\left(\sum_{i=1}^{I}\omega_i\right) - I\alpha(1+\alpha)^{-1-\frac{p}{2}}\right]\boldsymbol{I}_p\right\}\boldsymbol{\Lambda}^{-1}\frac{\partial \boldsymbol{\Lambda}}{\partial \Lambda_i}\right\} = 0$$

$$\Rightarrow \boldsymbol{\Lambda} = \frac{\sum_{i=1}^{I}(\boldsymbol{\beta}_i - \boldsymbol{\mu})(\boldsymbol{\beta}_i - \boldsymbol{\mu})^\top \omega_i}{\left(\sum_{i=1}^{I}\omega_i\right) - I\alpha(1+\alpha)^{-1-\frac{p}{2}}}, \tag{31}$$

where the weight for $\boldsymbol{\beta}_i$ is $\omega_i = \exp\left[-\frac{\alpha}{2}\left(\boldsymbol{\beta}_i - \hat{\boldsymbol{\mu}}\right)^\top \boldsymbol{\Lambda}^{-1}\left(\boldsymbol{\beta}_i - \hat{\boldsymbol{\mu}}\right)\right]$ and $\Omega = \sum_{i=1}^{I}\omega_i$. Thus,

$$\boldsymbol{\Lambda} = \frac{\sum_{i=1}^{I}\omega_i(\boldsymbol{\beta}_i - \boldsymbol{\mu})(\boldsymbol{\beta}_i - \boldsymbol{\mu})^\top}{\Omega - I\alpha(1+\alpha)^{-1-\frac{p}{2}}} \tag{32}$$

B For Numerical Study

B.1 Comparison on robust estimates with various power parameters

We compare the robust regression estimates with power parameters $\alpha_r = 0.1$ and $\alpha_r = 0.4$ in Figure 7, and compare the robust group mean estimates with power parameters $\alpha_g = 0.2$ and $\alpha_g = 0.5$, while fixing $\alpha_r = 0.1$, in Figure 8.

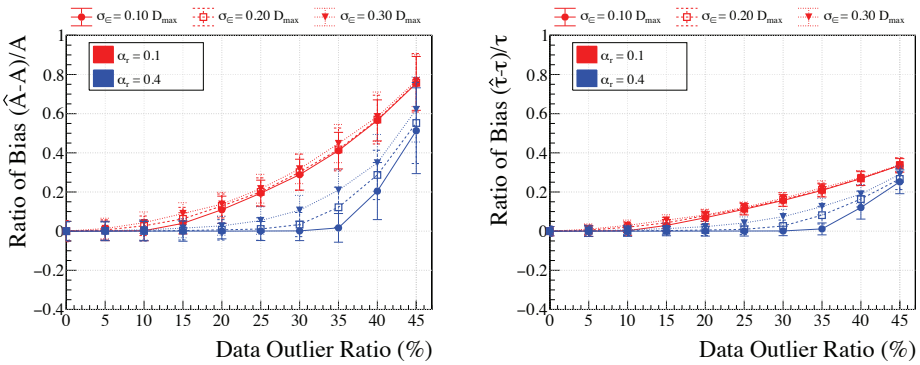
Figure 7 demonstrates that even at the highest error intensity, $\sigma_\epsilon = 0.3D_{\max}$: with $\alpha_r = 0.4$, the bias can be effectively controlled within 10% for outlier ratios up to 25% for the amplitude parameter and 30% for the width parameter. This figure also illustrates that, while larger α_r improves robustness against outliers, it also introduces greater variability in the estimates, which can reduce efficiency.

Figure 8 shows that, with $\alpha_g = 0.5$, the bias remains controlled within 10% for outlier ratios up to 30% for estimating the amplitude parameter and 35% for the width parameter. Similar to Figure 7, this figure illustrates that while increasing α_g enhances robustness against outliers, it also introduces more variability in the estimates.

B.2 Real Data Analysis on the paired data EMD-11103 (EMDB) and PDB-6Z6U

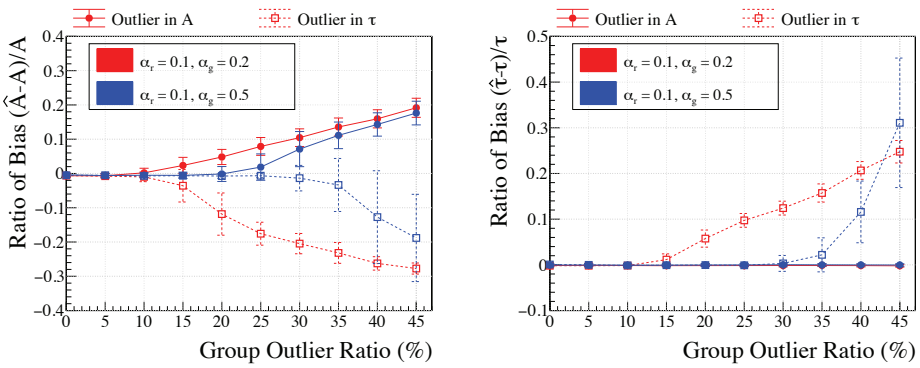
Figure 9 presents one-dimensional projections of the cryo-EM map values for C_α across 20 amino acid types, based on paired data from EMD-11103 (EMDB) and PDB-6Z6U. Each panel corresponds to an amino acid type, labeled at the top with three bold letters. The solid curves represent averaged data values at varying distances from the atom center, indexed on the X-axis. Outliers, identified using the criterion that the weight contribution is less than a threshold of $0.05/I$, are highlighted in red. The red dashed lines indicate robust estimates with $\alpha_r = 0.1$ and $\alpha_g = 0.2$, while the blue dotted lines represent non-robust estimates with $\alpha_r = \alpha_g = 0$.

The figure clearly illustrates that when no outliers are present, the robust (red dashed) and non-robust (blue dotted) estimates align closely. However, in the presence of outliers, as observed for SER, the robust estimates (red dashed line) effectively mitigate the outlier impact, demonstrating enhanced robustness compared to the non-robust estimates.



(a) Amplitude (b) Width

Figure 7: **Comparison on the robust regression estimates with power parameters $\alpha_r = 0.1$ and $\alpha_r = 0.4$.** Using the simulation data illustrated in Figure 1, we assessed the estimation performance of the robust regression estimates with power parameters $\alpha_r = 0.1$ and $\alpha_r = 0.4$. The bias ratios for amplitude (left panel) and width (right panel) are plotted across varying proportions of outliers on the x-axis. Both panels demonstrate that the estimates with $\alpha_r = 0.4$ consistently outperforms those of $\alpha_r = 0.1$ that it can better resist the impacts from outliers across all cases. Notice that, while larger α_r improves robustness against outliers, it also introduces greater variability in the estimates, which can reduce efficiency. In this figure, the estimates with $\alpha_r = 0.1$ is represented in red and $\alpha_r = 0.4$ in blue, with three different error intensities depicted by three shapes: solid circles, empty squares, and solid triangles.



(a) Amplitude (b) Width

Figure 8: **Bias comparisons of group parameter estimates with power parameters $\alpha_g = 0.2$ and $\alpha_r = 0.5$.** This figure compares the robust group mean estimates with power parameters $\alpha_g = 0.2$ and $\alpha_g = 0.5$, while fixing $\alpha_r = 0.1$. Bias ratios for amplitude (left panel) and width (right panel) are plotted across varying proportions of outliers on the x-axis. Both panels demonstrate that the robust estimate ($\alpha_g = 0.5$) consistently outperforms non-robustness estimate ($\alpha_g = 0.2$) across all cases, except where the blue and red circles coincide in the right panel. Similar to Figure 7, this figure illustrates that while larger α_g enhances robustness against outliers, it also introduces more variability in the estimates. We consider two types of outliers: those affecting amplitude (solid circles) by increasing its mean by 50%, and those affecting width (empty squares) by doubling its mean. The blue and red circles in the right panel remain at zero, indicating that amplitude outliers do not impact width estimation for either robust or non-robust estimates.

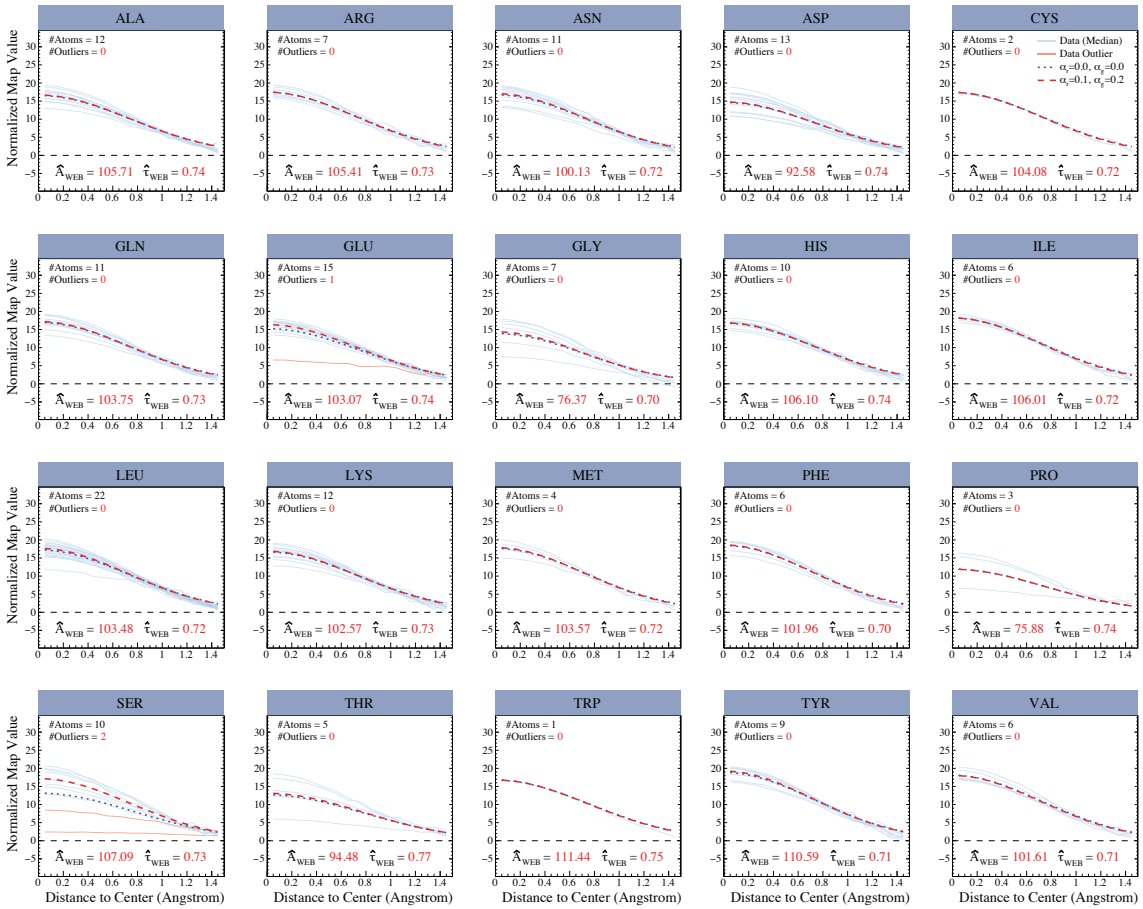


Figure 9: A summary figure for the real data analysis, presenting the one dimensional projection of the cryo-EM map values of C_α across the 20 amino acids, using paired information from EMD-11103 (EMDB) and PDB-6Z6U. Each panel presents the information attached one amino acid type labeled with 3 bold letters on top. Solid curves represent the data, averaging over those grid points the same distances from the center indexed on the X-axis. Red curve represents the outliers detected by our method. Red dash lines are for the curves generated by the robust estimators with $\alpha_r = 0.1$ and $\alpha_g = 0.2$ while the blue dot lines for the non-robust estimators with $\alpha_r = \alpha_g = 0$.