Estimating protein flexibility *via* uncertainty quantification of structure prediction models

Charlotte Sweeney*
Department of Engineering
University of Oxford
csweeney@robots.ox.ac.uk

Nele P. Quast*
Department of Statistics
University of Oxford
quast@stats.ox.ac.uk

Fabian C. Spoendlin
Department of Statistics
University of Oxford
spoendlin@stats.ox.ac.uk

Yee Whye Teh
Department of Statistics
University of Oxford
yee.teh@stats.ox.ac.uk

Abstract

Deep learning architectures such as AlphaFold2, have effectively solved the protein structure prediction problem, however, they do not rigorously account for conformational variance in structures despite many proteins exhibiting flexible regions in which a single amino acid sequence may occupy a variety of conformations. In particular, using confidence metrics such as the pLDDT score, it is not readily possible to distinguish between regions of the protein structure where the prediction model is uncertain because the region is out-of-distribution or because the region is intrinsically flexible. Here, we use a novel approach to estimate protein flexibility *via* uncertainty quantification. Specifically, we reformulate the protein structure prediction problem as sampling a backbone function from a Gaussian process which enables us to cast flexibility estimation as aleatoric uncertainty quantification. We adapt the AlphaFold2 Structure Module architecture to produce such estimates of aleatoric uncertainty and compare these to existing proxies for conformational variance. We demonstrate the utility of our formalisation for approximating protein flexibility in a prediction framework, and our experiments demonstrate the promise of our method whilst emphasising the relationship between epistemic and aleatoric uncertainty in protein structure prediction.

1 Introduction

The majority of protein structure prediction models consider protein structures as single conformations resulting from a bijective mapping from an amino acid sequence. This results in measures of protein flexibility[1] that are either calculated from prediction errors [1], or in the case of AlphaFold2 [2], are approximated by the pLDDT scores. While these proxies for flexibility can be empirically useful, they are unable to differentiate between deviations in structure predictions that arise due to inaccurate predictions and deviations that arise due to some protein regions being intrinsically flexible and therefore resulting in ambiguous predictions.

*Equal contribution and corresponding authors

[1]For brevity we use the terms 'flexibility' and 'exhibiting multiple conformations' interchangeably here. Due to the type of protein data we have used we are in fact concerned with multiple conformations, but our methods should extend to flexibility if an appropriate dataset, perhaps deriving from molecular dynamics simulations, were used.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

In this work, we approach the protein structure prediction problem from the perspective of the established field of uncertainty quantification. We formulate the approximation of protein flexibility as the estimation of the aleatoric uncertainty of the output of a structure prediction model. We adapt the architecture of the AlphaFold2 Structure Module [2] as implemented by Abanades et al. [1] to produce variance estimates alongside the structure predictions. Using deep ensembles [3], we then estimate the aleatoric uncertainty of the model's predictions and evaluate the quality of these predictions by comparing them to the empirical variance observed in our dataset, the crystal structure B-factors, and AlphaFold2's pLDDT score. Our results demonstrate the utility of our formalisation for approximating flexibility in a structure prediction framework, and our experiments indicate the promise of our method whilst emphasising the relationship between epistemic and aleatoric uncertainty in protein structure prediction.

2 Background

2.1 Uncertainty Quantification

The field of uncertainty quantification in machine learning aims to estimate how certain a model is about its predictions. Commonly, these uncertainty estimates are decomposed into two categories - *epistemic* and *aleatoric* [4]. Epistemic uncertainty is often considered 'reducible' and refers to the uncertainty specific to the deep learning model's parameters, typically due to a lack of training data. Aleatoric uncertainty is the uncertainty intrinsic to the data itself. Aleatoric uncertainty can be further subdivided into homoscedastic uncertainty, which is constant across all observations, and heteroscedastic uncertainty, which is dependent on the data [5]. Here, we are interested in the latter as proteins exhibit varying degrees of flexibility.

The total uncertainty of a machine learning model can be approximated as the variance of the posterior predictive distribution and decomposed into its epistemic and aleatoric components [6, 7],

$$\mathrm{Var}[y|x,\mathcal{D}] = \underbrace{\mathrm{Var}_{\theta\sim\pi(\cdot|\mathcal{D})}[\mathbb{E}[y|x,\theta]]}_{\text{Epistemic}} + \underbrace{\mathbb{E}_{\theta\sim\pi(\cdot|\mathcal{D})}[\mathrm{Var}[y|x,\theta]]}_{\text{Aleatoric}}. \tag{1}$$

Methods such as deep ensembles provide a simple approach to estimating the terms of this decomposition via the sample estimators of an ensemble's mean and variance predictions [3]. The output of each network is modelled as a Gaussian with the mean and variance predicted by the network,

$$y|x,\theta \;\sim\; \mathcal{N}(\mu(x;\theta),\sigma^2(x;\theta)). \tag{2}$$

Estimates of the terms in Equation 1 are given by the sample estimators of the outputs of the ensemble. For a more extensive overview of uncertainty quantification, we refer the reader to [4, 8, 9].

2.2 Protein Conformations and Flexibility

Rather than existing as single, static structures, many proteins have been experimentally shown to exist in multiple conformations or to exhibit flexible regions[10]. X-ray crystallography uses the B-factor to express uncertainty in atomic coordinates, or, in cases where distinct conformations are detected, these are typically annotated in the PDB entry, or deposited as separate structures [11–14]. Cryo-EM and nuclear magnetic resonance (NMR) imaging provide alternatives to x-ray crystallography for structure and flexibility determination. However, these approaches are all experimental and therefore cannot be used to estimate the flexibility or conformations of a protein structure that are predicted *in-silico*.

When predicting the structure of proteins from their sequence, the predicted local distance difference test (pLDDT) score, introduced as the confidence metric for AlphaFold2's structure predictions [2], has been used to quantify conformational variance [15, 16]. Previous studies suggest that low pLDDT values are linked to flexibility[17, 18], although some have reported that pLDDT does not align with with molecular dynamics (MD) derived flexibility estimates for intrinsically disordered proteins [18], nor do they correlate strongly with experimental B-factors [19]. Recently, the prediction of multiple protein structure conformations has also been achieved by sub-sampling the multiple-sequence alignments (MSAs) passed through the Evoformer component of the AlphaFold2 architecture [20]. This approach is related to ours, since sub-sampling the MSA of a protein sequence leads to sampling a distribution of predicted conformations, much like the deep ensemble method we propose here.

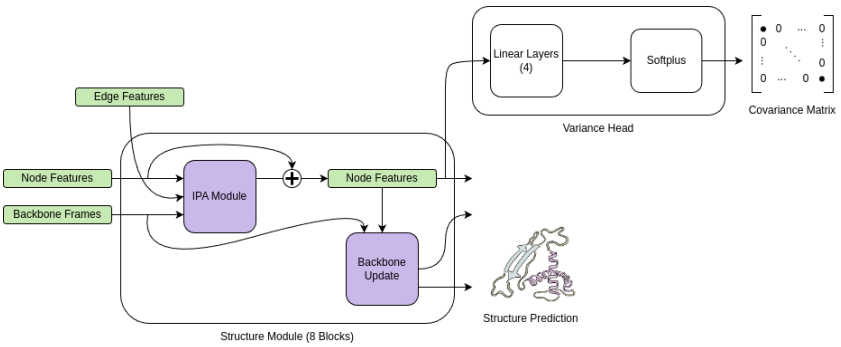

Figure 1: Augmented AlphaFold2 Structure Module architecture with covariance prediction head.

Finally, the root-mean-square fluctuation (RMSF) of MD simulations has also been used as a measure for protein flexibility, but MD studies are often prohibitively expensive, and, similarly to pLDDT scores, RMSF does not always correlate with experimental B-factors [18].

3 Methods

3.1 Formulating Protein Structure Prediction for Uncertainty Quantification

We model protein structure coordinates as originating from a Gaussian process to express the origins of uncertainty in the structure prediction problem. We model the backbone of a protein structure as a continuous curve f_α evaluated at points t_i, $i = 1 \cdots N_\alpha$ where N_α is the number of residues in the protein. Let α denote the amino acid sequence of the protein and let $f_\alpha : \mathbb{R} \to \mathbb{R}^3, t \mapsto (x, y, z)$ be the function that encodes the protein's backbone structure. In this paradigm, the objective of a structure prediction model is to learn the parametric curve as a function of the input amino acid sequence. We proceed to model f_α as a tuple of independent components, f^x, f^y, f^z, each of which is drawn from a Gaussian process with mean functions $m^x_\alpha, m^y_\alpha, m^z_\alpha : \mathbb{R} \to \mathbb{R}^3$ and covariance function $K_\alpha : (\mathbb{R} \times \mathbb{R}) \to \mathbb{R}$,

$$f_\alpha = (f^x, f^y, f^z) \mid \alpha \;\sim\; \mathcal{GP}_x(m^x_\alpha, K_\alpha) \times \mathcal{GP}_y(m^y_\alpha, K_\alpha) \times \mathcal{GP}_z(m^z_\alpha, K_\alpha). \tag{3}$$

This allows us to explicitly define the covariance function K_α, which encodes the variability of the atomic coordinates of a given amino-acid sequence, and yields the following equation for the backbone coordinates:

$$(\mathbf{x}, \mathbf{y}, \mathbf{z})_i = \epsilon_1 + f_\alpha(t_i) = \epsilon_1 + \epsilon_2 + m_\alpha(t_i), \tag{4}$$

where $m_\alpha = (m^x_\alpha, m^y_\alpha, m^z_\alpha)$, $\epsilon_1 = (\epsilon_{1,1}, \epsilon_{1,2}, \epsilon_{1,3}) \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$ and $\epsilon_2 = (\epsilon_{2,1}, \epsilon_{2,2}, \epsilon_{2,3}) \overset{iid}{\sim} \mathcal{N}(0, K_\alpha(t_i, t_i))$. The σ^2 term allows for noise in the evaluation of f_α at points t_i that does not originate from conformational variance. We can now use deep ensembles, as is standard in uncertainty quantification, thereby attaining predictions of the mean structure m_α and its covariance $K_\alpha + \sigma^2\mathbb{I}$, rather than regressing directly to the experimental coordinates $f_\alpha(t_i)$. Hence, the aleatoric uncertainty estimates should capture the flexibility of the protein that is encoded in K_α.

3.2 Adapting Alphafold2's Structure Module

In order to quantify the prediction uncertainty *via* sampling estimates over a deep ensemble, the structure prediction model must provide a variance estimate alongside the mean structure prediction. To achieve this, we adapt the AlphaFold2 Structure Module [2], as implemented by Abanades et al. [1], and add a variance head which takes as input the original model's internal node features (Fig. 1). These features pass through four fully-connected layers and a soft-plus activation function, which ensures positivity, to yield a final variance estimate. This architecture is similar to the pLDDT module AlphaFold2 uses, however, where the pLDDT head is trained as a regression to predict experimental LDDT values, we train our variance head using maximum likelihood.

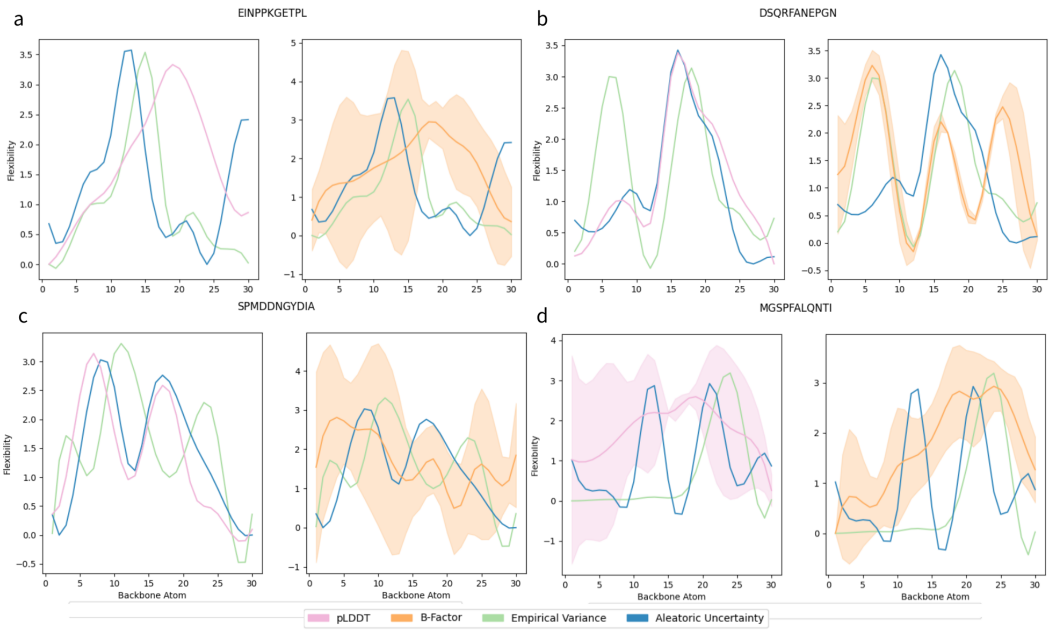

Figure 2: Results of flexibility estimation from our aleatoric uncertainty estimates compared to pLDDT values, B-factors, and the empirical variance of the structure's conformations in the loops dataset.

Prior work has demonstrated that variance estimates can be trained by minimising the negative log likelihood of predictions while using a stop gradient to prevent the variance estimates from affecting the accuracy of the mean predictions [21, 22]. Here we have augmented the FAPE loss term [2] with a negative log-likelihood term, attaining a loss function for uncertainty quantification ($\mathcal{L}_{UQ}$).

$$\mathcal{L}_{UQ} = \mathcal{L}_{FAPE} + \frac{1}{2}\log|\hat{\Sigma}(\lfloor s \rfloor)| + \frac{c}{2}(\hat{x}-x)^T\hat{\Sigma}^{-1}(\lfloor s \rfloor)(\hat{x}-x), \tag{5}$$

where $\hat{\Sigma}$ is a diagonal covariance matrix that is a function of the node features, s, with stop-gradient $\lfloor \cdot \rfloor$. We explore the effect of the independence assumption made by using a diagonal covariance matrix on the aleatoric uncertainty estimates in App.A.1. The target and predicted backbone coordinates are given by $x, \hat{x} \in \mathbb{R}^{N_\alpha \times 3}$, and c is a scaling factor from the FAPE loss that regularises the variance loss term. In this framework the covariance estimate attenuates the error in the predicted coordinates, enabling the model to learn to predict high variance for highly flexible datapoints. For more details on the loss function see App.A.2.

We train an ensemble of five models with different seeds to use deep ensembles to produce our estimates of aleatoric uncertainty. We plot the aleatoric uncertainty estimates as a curve of predicted flexibility across the loop. Details of the training procedure are provided in App.A.3.

3.3 Dataset

The loops dataset consists of 68,390 amino acid sub-sequences that refer to loop regions that bridge beta sheets in protein structures [23]. This dataset contains many examples of identical amino-acid sequences mapping to multiple distinct loop conformations and we therefore use it to examine and evaluate our method for quantifying aleatoric uncertainty in protein structures. Due to the computational expense of training large protein structure prediction ensembles we have restricted our experiments here to loops of sequence length 11. We have characterised the dataset in App.A.4.

4 Results

4.1 Aleatoric Uncertainty as an Estimator of Protein Flexibility

After training the ensemble, we take the mean of the variance predictions to obtain estimates of aleatoric uncertainty. Table 1 reports the RMSD of the loop predictions, and shows that the prediction

accuracy varies substantially across the different models in the ensemble. Figure 2 shows examples of the aleatoric uncertainty calculated from our model compared to the empirical variance from the dataset, AlphaFold2's pLDDT score, and the crystal structures' B-factors. Broadly, the aleatoric uncertainty follows the shape of the empirical variance (EINPPKGETPL) or pLDDT (DSQRFANEPGN & SPMDDNGYDIA) in some cases, while not approximating either in others (MGSPFALQNTI, App.B.1). This suggests that our aleatoric uncertainty estimate can occasionally capture elements of the variance arising from the data, but that this desired behaviour is inconsistent, implying that the variance predictions of the deep ensemble could be improved through further optimisation. Our results also indicate a potential link between the aleatoric uncertainty and AlphaFold2's pLDDT score that warrants exploration. In general, the aleatoric uncertainty does not correlate well with the B-factors, in line with prior work that has identified inconsistencies between conformational variance metrics (such as pLDDT and RMSF) and B-factors [24].

Since we are initially seeking to validate the use of the maximum likelihood objective and the adapted architecture, two of the examples (EINPPKGETPL & DSQRFANEPGN) presented in Fig.2 originate from the training set. This enables us to explore the aleatoric uncertainty estimate attained with our method while minimising the interference of epistemic uncertainty. It is also worth noting that while some examples of aleatoric uncertainty correlate well with other metrics for flexibility (Fig. 2a, b, & c), we also observed examples where the aleatoric uncertainty does not map onto the empirical variance or the pLDDT scores (Fig. 2d, App.B.1). We hypothesise that this is due to high epistemic uncertainty masking the aleatoric uncertainty; a phenomenon which has previously been reported by [25] and which we further explore below.

We use the edit distance of test-set examples to their closest training example as a proxy for the expected epistemic uncertainty of the model for that test point. We evaluated the quality of the aleatoric uncertainty estimates against this proxy for the epistemic uncertainty but were not able to find a significant correlation (Table 3). To better characterise the relationship between epistemic and aleatoric uncertainty, we reverted to simulations with synthetic data to which epistemic and aleatoric uncertainty were injected in controlled amounts.

4.2 Simulations Show Dependency on Epistemic Uncertainty

We simulated the effect of epistemic uncertainty on estimates of the aleatoric uncertainty by training an ensemble of neural networks on a simple regression task. We generated a toy dataset with heteroscedastic noise and simulated the epistemic uncertainty of the model predictions by varying the number of training samples. We then assessed the quality of the aleatoric uncertainty estimates by calculating the error between the estimate and the true variance of the data-generating distribution. For more details and results of the simulations, please see App.B.2. The Spearman's rank correlation coefficient for a two-tailed hypothesis test examining the existence of a monotonic relationship between epistemic uncertainty and the error of aleatoric uncertainty predictions is 0.5631 with a p-value of $0.0007 < 0.05$ for $n = 29$. It is therefore highly likely that, in our framework, high epistemic uncertainty has a detrimental impact on the quality of aleatoric uncertainty estimates.

This finding is supported by our results that, while aleatoric uncertainty estimates appear in some cases to be well correlated with existing metrics for protein flexibility, the high epistemic uncertainty, implied by the relatively high RMSD and estimated epistemic uncertainty of predictions of training set structures (Tables 1, 2), potentially prohibits more consistent flexibility estimates.

5 Discussion

We have proposed a novel method for exploring protein flexibility with structure prediction models by recasting the prediction of conformational variance as the estimation of the aleatoric uncertainty. After adapting and training an implementation of the AlphaFold2 Structure Module on our loops dataset, we found that a select number of our flexibility estimates align well with existing measures of conformational variance. However, we also identified instances of disagreement, which we hypothesise is due to high epistemic uncertainty and could potentially be remediated by hyperparameter tuning. Promisingly, some models in the ensemble exhibit more accurate mean predictions, suggesting that the architecture remains sufficiently expressive to accurately predict protein structures after adding the variance head. Alongside the impact of high epistemic uncertainty on the quality of aleatoric uncertainty predictions, we have explored the impact of the diagonal covariance matrix

assumption (App.A.1), showing that the aleatoric uncertainty estimates suffer under the assumption of independence between neighbouring residues in the covariance matrix; this insight offers another potential avenue of improvement for our method. In conclusion, we consider the results presented here as preliminary indicators that approaching protein structure prediction with established methods from uncertainty quantification holds promise for differentiating prediction errors due to the intrinsic conformational variance of proteins from errors due to model training and data biases.

References

[1] Brennan Abanades et al. ImmuneBuilder: Deep-Learning models for predicting the structures of immune proteins. *Communications Biology*, May 2023.

[2] John Jumper et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, August 2021.

[3] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, November 2017.

[4] Eyke Hüllermeier and Willem Waegeman. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Machine Learning*, March 2021.

[5] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, September 2016.

[6] Matias Valdenegro-Toro and Daniel Saromo. A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement, April 2022. arXiv:2204.09308.

[7] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A General Framework for Uncertainty Estimation in Deep Learning. *IEEE Robotics and Automation Letters*, April 2020.

[8] Jakob Gawlikowski et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, October 2023.

[9] Moloud Abdar et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, December 2021.

[10] Katherine Henzler-Wildman and Dorothee Kern. Dynamic personalities of proteins. *Nature*, December 2007.

[11] P. A. Karplus and G. E. Schulz. Prediction of chain flexibility in proteins. *Naturwissenschaften*, April 1985.

[12] Mauno Vihinen, Esa Torkkila, and Pentti Riikonen. Accuracy of protein flexibility predictions. 1994.

[13] Rune Linding et al. Protein Disorder Prediction: Implications for Structural Proteomics. *Structure*, November 2003.

[14] Predrag Radivojac et al. Protein flexibility and intrinsic disorder. *Protein Science*, 2004.

[15] Paloma Tejera-Nevado et al. Analysis of the Confidence in the Prediction of the Protein Folding by Artificial Intelligence. *Lecture Notes in Networks and Systems*, 2023.

[16] Wensi Zhu, Aditi Shenoy, Petras Kundrotas, and Arne Elofsson. Evaluation of AlphaFold-Multimer prediction on multi-chain protein complexes. *Bioinformatics*, July 2023.

[17] Puyi Ma, Da-Wei Li, and Rafael Brüschweiler. Predicting protein flexibility with AlphaFold. *Proteins: Structure, Function, and Bioinformatics*, 2023.

[18] Hao-Bo Guo et al. AlphaFold2 models indicate that protein sequence determines both structure and dynamics. *Scientific Reports*, June 2022.

[19] Oliviero Carugo. pLDDT Values in AlphaFold2 Protein Models Are Unrelated to Globular Protein Local Flexibility. *Proteins: Structure, Function and Bioinformatics*, November 2023.

[20] Gabriel Monteiro da Silva et al. Predicting Relative Populations of Protein Conformations without a Physics Engine Using AlphaFold 2. December 2023. arXiv:2307.14470 [physics.bio-ph].

[21] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems 2017*, December 2017.

[22] Andrew Stirn et al. Faithful Heteroscedastic Regression with Neural Networks. In *Proceedings of Machine Learning Research 2023*, April 2023.

[23] Fabian C. Spoendlin et al. Abflex: Predicting the conformational flexibility of antibody CDRs. In *ICML'24 Workshop ML for Life and Material Science: From Theory to Industry Applications*, 2024. URL `https://openreview.net/forum?id=or4tArwd5a`.

[24] Alexander Rashin, Abraham Rashin, and Robert Jernigan. Protein flexibility: Coordinate uncertainties and interpretation of structural differences. *Acta Crystallographica Section D: Biological Crystallography*, November 2009.

[25] Rebecca L. Russell and Christopher Reale. Multivariate Uncertainty in Deep Learning, June 2021. URL `http://arxiv.org/abs/1910.14215`. arXiv:1910.14215 [cs, stat].

[26] Michele Vendruscolo, Edo Kussell, and Eytan Domany. Recovery of protein structure from contact maps. *Folding and Design*, October 1997.

[27] Yan Li, Sheng-Long Hu, Jie Wang, and Zheng-Hai Huang. An Introduction to the Computational Complexity of Matrix Multiplication. *Journal of the Operations Research Society of China*, March 2020.

[28] David Kristjanson Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014.

[29] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks, April 2022. arXiv:2203.09168 [cs, stat].

[30] Esko Ukkonen. On approximate string matching. In *Foundations of Computation Theory*, 1983.

[31] Liyuan Liu et al. On the Variance of the Adaptive Learning Rate and Beyond, October 2021. arXiv:1908.03265 [cs, stat].

Appendix

Tables

	Each Conformation			Mean Conformation		
Type	Train	Val	Test	Train	Val	Test
Model 1	2.0985	2.7710	2.9211	2.0760	2.7131	2.9107
Model 2	1.0616	2.4403	2.6345	0.9926	2.3987	2.6060
Model 3	1.3830	2.4122	2.6320	1.3282	2.3791	2.6079
Model 4	1.9821	2.8083	2.9911	1.9426	2.7442	2.9775
Model 5	0.8508	2.1904	2.4119	0.7683	2.1904	2.3864
Ensemble	2.841	3.108	3.3419	2.715	3.103	3.3387

Table 1: Table of mean RMSDs over predictions made on loops in the training, validation, and test set. The 'Type' column refers which model(s) are making the predictions. The 'Ensemble' row is the RMSD of the mean prediction made by all 5 models. The 'Each Conformation' column uses the target structure in the dataset as the target when calculating the RMSD. The 'Mean Conformation' column uses the mean of all conformations corresponding to the loop as the target when calculating the RMSD. RMSD values in the table indicate a generally poor performance.

Loop	RMSD	Epistemic	Aleatoric	Confs	Confs 1.25Å
SPLPGPSGNVE	2.513	10.147	0.3135	3	1
MGSPFALQNTI	2.691	10.147	0.3135	8	2
LDQSEAPVRQN	3.304	10.147	0.3135	7	3
AEHSELQGQKQ	2.527	7.084	0.2794	3	1
CTYPAHYAGGM	2.689	7.084	0.2794	2	1
EINPPKGETPL	3.163	10.147	0.3135	6	1
DSQRFANEPGN	2.625	10.147	0.3135	2	1

Table 2: Table showing the performance of the model in predicting the mean and variance for the selected points in the training set. It includes the RMSD of the ensemble when compared with the mean conformation of stored conformations, the mean epistemic and aleatoric uncertainties of the prediction, and information about the total number of conformations for this loop in the loops dataset and the number of distinct conformations remaining after combining loops within 1.25Å of one another. We do not observe the mean aleatoric uncertainty showing higher values for loops with more distinct conformations at 1.25Å.

Loop	Edit Dist	RMSD	Epistemic	Aleatoric	Confs	Confs 1.25Å
NSWGTTWGEEG	3	2.719	7.508	0.1398	4	1
SPQDDMGYDIA	4	2.437	7.226	0.2732	5	1
KNNNEVGIGAP	7	3.352	12.229	0.2407	6	1

Table 3: Table showing the predictive performance of our model on select points of the test set. The mean epistemic uncertainty roughly reflects the minimum edit distance of the loops to the loops of the training set but it is better reflected in the RMSDs of the predictions. As with the training set, mean aleatoric uncertainty shows no connection to the number of conformations columns.

A Methods

A.1 Covariance Matrix Structure

As the Structure Module makes joint predictions on the three dimensional coordinates of the protein structure, the network has the capacity to predict a full covariance matrix rather than solely the diagonal components. The residues within a protein are heavily dependent on both the residues that neighbour them in the amino acid sequence but also the residues that are close to them in 3D space [26]. This suggests that consideration of covariance terms could be an important factor in the

prediction of variance terms. However, consideration of the full covariance matrix is very costly due to the inversion of the matrix in the loss function of equation 5 [27]. Therefore, despite the capacity of the model to estimate the covariance terms, we only consider the estimation of the variance terms.

We make the assumption that the backbone function is a draw from the following Gaussian process,

$$(f^x, f^y, f^z) \mid \alpha \quad \sim \quad \mathcal{GP}(m^x_\alpha, D_\alpha) \times \mathcal{GP}(m^y_\alpha, D_\alpha) \times \mathcal{GP}(m^z_\alpha, D_\alpha),$$

where D_α is a covariance function such that $D_\alpha(t_i, t_j) = \delta_{ij} K_\alpha(t_i, t_j)$ and δ_{ij} is the Kronecker delta.

To investigate the impact of the assumption of a diagonal covariance matrix across protein structure predictions, we consider a regression problem on a parametric curve where the underlying function is drawn from three Gaussian processes for each coordinate basis and is evaluated at a number of sensor locations along the curve, emulating the problem formulation in Section 3.1. We implement a feedforward neural network to learn the mean and variance along the curve from a dataset of conformation samples. The model will make joint predictions over each conformation sample, predicting a mean conformation and a diagonal covariance matrix. We train an ensemble of 10 models in each case to produce estimates of the aleatoric uncertainty.

We compare the quality of the aleatoric estimates given by the ensemble across a range of lengthscale values for the Radial Basis Function kernel of the Gaussian processes [28]. This is so that we are able to see how well the aleatoric estimates correspond to the true variance values as the correlation strength of off-diagonal points in the ground truth covariance matrix varies. We also consider an 'altered' RBF kernel to consider the effect of covariance between distant points. It takes the form

$$k(t_1, t_2) = \text{RBF}(t_1, t_2) + 0.2|t_1 - t_2| \cdot \mathbb{1}\{|t_1 - t_2| > \pi\},$$

where $\mathbb{1}$ is the indicator function. In both cases, a diagonal matrix is added to the covariance matrix derived from the kernel to add heteroscadastic noise of the form $\sin^2(\frac{t}{3})$, with the following training details,

Data: A training example is a sample from the multivariate normal distributions, $\mathbf{x} \sim \mathcal{N}(\mu_x, \Sigma), \mathbf{y} \sim \mathcal{N}(\mu_y, \Sigma), \mathbf{z} \sim \mathcal{N}(\mu_z, \Sigma)$ where $\mu_{x,i} = \frac{5t_i}{4}\cos^2(t_i), \mu_{y,i} = \frac{5t_i}{4}\sin^2(t_i), \mu_{z,i} = \frac{5}{3}\cos(t_i)$ for 20 fixed sensor locations, $t_{i=1\cdots 20}$, evenly spaced between $-\pi$ and π. Each experiment uses 30 training examples which is 30 samples of $(\mathbf{x}, \mathbf{y}, \mathbf{z})$. The covariance matrix Σ differs across experiments. Under the RBF kernel, it takes the form $\Sigma_{ij} = \frac{1}{2l^2}\exp(-(t_i - t_j)^2 + \mathbb{1}_{t_i = t_j}\sin(\frac{t_i}{3})^2$ where the latter term gives the heteroscedastic variance along the diagonal of the covariance matrix. Under the altered RBF kernel, the covariance matrix takes the form $\Sigma_{ij} = \frac{1}{2l^2}\exp(-(t_i - t_j)^2 + \mathbb{1}_{|t_i - t_j| > \pi}\frac{|t_i - t_j|}{5} + \mathbb{1}_{t_i = t_j}\sin(\frac{t_i}{3})^2$. The lengthscale, l, is also altered across experiments varying between values 0.01 and 2 in intervals as given in Table 5. For the test set, a single sample is used but with the 20 sensor locations chosen uniformly at random from the range $t \in [-\pi, \pi]$.

Model: The model is a feedforward network with leaky-relu activations. The trunk of the network has a single hidden linear layer. The mean head of the network has two additional hidden layers and the variance head has an additional four hidden layers. The output of the variance head is passed through a softplus activation.

Training: For each covariance matrix structure, we trained an ensemble of 10 models. They were trained for 300 epochs using ADAM with a weight decay parameter set to 0.0001. We used cosine annealing with warm restarts as a learning rate scheduler with a 50 epoch warm up time and a learning rate range of $(0.0001, 0.001)$. The batch size was set to 1

The predicted aleatoric estimates along the curve are shown in Figure 4 and compared to the known variance. Table 5 records the mean squared error of the aleatoric uncertainty estimates with the ground truth variance as described in Equation 8.

Across both kernel types, we see that the aleatoric estimates made on the train set resemble the ground truth variance more closely than the estimates made on the test set. In the test set, the effect of the lengthscale parameter does not exhibit a clear trend. However, in the train set there is a clear trend as the lengthscale varies. In both kernel types, Table 5 shows that larger lengthscale values in the underlying kernel are associated with worse aleatoric uncertainty predictions as they relate to variance. In Figure 4, the aleatoric uncertainty estimates made on the test set are more closely centered around the ground truth variance values. However, closer inspection on the RBF kernel case

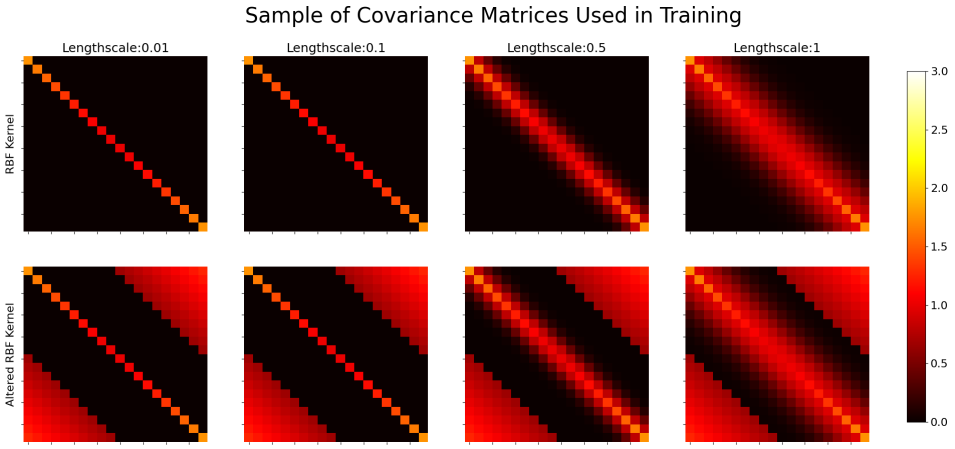

Figure 3: Heatmaps of covariance matrices used to generate the sets of training data. The top row is a sample of covariance matrices from a Gaussian process with an RBF kernel. The bottom row is a sum of the covariance matrices from the altered RBF kernel.

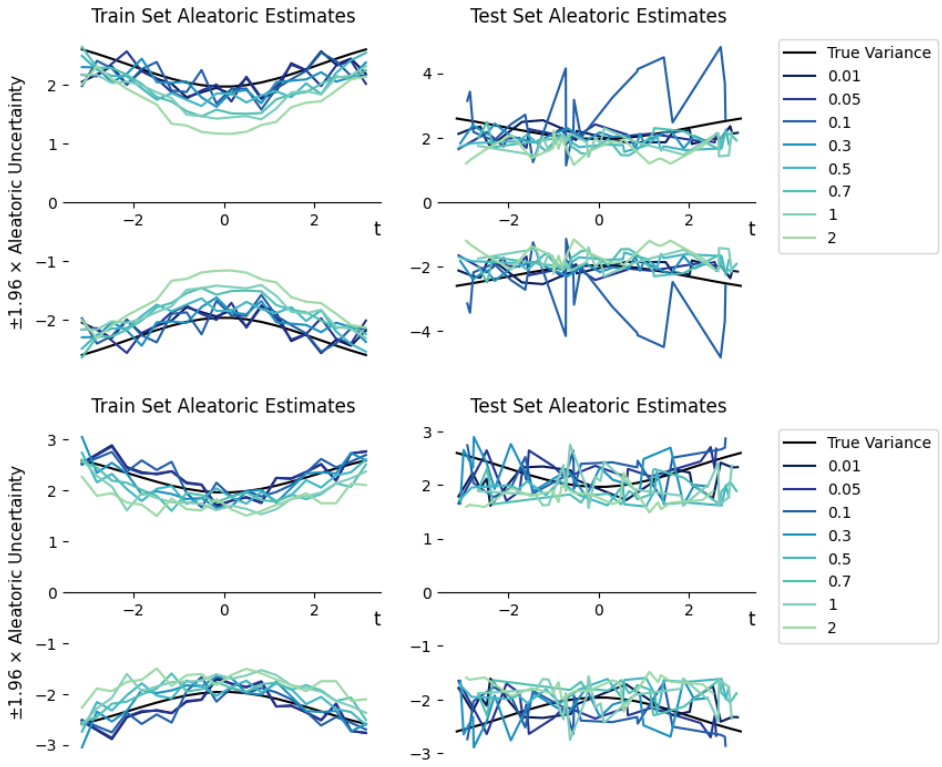

Figure 4: Plot comparing the aleatoric estimates made over sensor locations as the RBF lengthscale varies. The top row gives the results from the RBF kernel and the bottom row for the altered RBF kernel. The black line is the true variance. In the train set, aleatoric uncertainty estimates are closer in value to the ground truth variance as the lengthscale decreases but match the shape more closely as the lengthscale inscreases.

RBF Kernel			Altered RBF Kernel		
l	Train MSE	Test MSE	l	Train MSE	Test MSE
0.01	1.091	4.957	0.01	1.247	4.842
0.05	1.377	3.388	0.05	2.079	4.018
0.1	1.433	5.453	0.1	1.867	6.056
0.3	1.289	**2.131**	0.3	2.204	**3.288**
0.5	0.788	3.103	0.5	0.8385	3.997
0.7	1.456	3.170	0.7	1.441	4.115
1	0.719	4.480	1	0.9292	6.447
2	**0.558**	2.598	2	**0.7591**	4.226
Mean	1.089	3.66	Mean	1.421	4.624

Table 4: Table containing the mean squared error of predictions by the deep ensemble across lengthscale values and kernel function types on both the train and test set.

RBF Kernel			Altered RBF Kernel		
l	Train Var MSE	Test Var MSE	l	Train Var MSE	Test Var MSE
0.01	**0.1011**	**0.1252**	0.01	**0.0398**	0.1316
0.05	0.1058	0.1796	0.05	0.0417	0.1099
0.1	0.1091	0.3202	0.1	0.0437	**0.0796**
0.3	0.1072	0.1576	0.3	0.0577	0.1409
0.5	0.1116	0.2193	0.5	0.0900	0.2135
0.7	0.1348	0.1812	0.7	0.1097	0.1888
1	0.1656	0.2510	1	0.1366	0.2006
2	0.2368	0.3527	2	0.1936	0.2623
Mean	0.1340	0.2234	Mean	0.0890	0.1659

Table 5: Results of the experiment comparing the quality of aleatoric estimates across underlying covariance structures. The table records the mean squared error of the aleatoric uncertainty estimates against the ground truth variance on both the train and test sets.

show that the shape of the aleatoric uncertainty estimates made for larger lengthscale values more closely resemble that of the ground truth.

Across covariance matrix structures, results from the altered RBF kernel are often better in their estimations of ground-truth variance than those of the RBF kernel. In the case of the protein structure prediction problem, we would anticipate that the true covariance structure will more closely resemble that of the altered RBF kernel than the standard RBF kernel and with a larger lengthscale. This is because the atoms in the structure will have strong dependencies on the positions of neighbouring atoms, causing stronger values in the band of the covariance matrix. They will also be dependent on certain atoms that are far away in the sequence but close in physical space, causing stronger values in positions far from the band of the matrix [26]. Our results suggest that the covariance terms from non-neighbouring atoms will not hinder the ability of the model to produce good aleatoric uncertainty estimates but the dependencies between neighbouring atoms will reduce the quality of estimates made with only a diagonal covariance matrix.

A.2 Loss Function Adaptation

In order to estimate the aleatoric uncertainty of protein structure predictions, we alter the architecture of the Structure Module to give a variance estimate. We describe how we alter the loss function of the network to produce good variance estimates.

The likelihood-based method proposed by [21] involves modelling the output of a network as a Gaussian, parameterised by the mean head output, $\mu(x;\theta)$, and the variance head output, $\sigma(x;\theta)$, and minimising the negative log likelihood of this predicted distribution. The authors describe the learning of the heteroscedastic variance as a form of 'loss attenuation' as the variance is not learned to match a labelled variance but rather is learned to account for error in the mean prediction. The formula is given by

$$\mathcal{L}_{NLL} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2\sigma^2(x_i;\theta)}\|y_i - \mu(x_i)\|^2 + \frac{1}{2}\log\sigma^2(x_i;\theta), \tag{6}$$

where N is the number of observations, y_i is the true label for observation x_i, and μ, σ^2 are the mean and variance heads with network parameters, θ.

However, recent works have determined that the method proposed by Kendall and Gal can lead to a degradation in the quality of mean predictions [29, 22]. By jointly training a network to produce both a mean and variance output, the network is permitted to explain away a poor performance on some difficult data points by giving them a large variance and can therefore avoid improving the predictive performance of the mean head whilst maintaining a small loss. Networks trained in this manner exhibit a poor performance in mean predictions when compared to networks that predict just a mean output.

To allow a network to learn a heteroscedastic variance whilst maintaining 'faithful' estimates of the mean, Stirn et al. [22] suggest a separation of the mean head and the variance head by implementing a stop gradient between the variance head and the trunk of the network thereby allowing the variance head to take advantage of the features learned by the trunk network but not allowing the network to 'cheat' in its mean predictions. Secondly, they recommend scaling the gradient update of the mean head by the variance estimate to convert the learning procedure to a Newton gradient step as this removes an additional inversion the covariance matrix.

We adopt this method of variance prediction in our network as it is easy to integrate into the existing Structure Module network without having to compromise on mean predictive performance. The loss function that implements these two features takes the form of

$$\mathcal{L}_{ff} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|y_i - \mu(f_{trunk}(x_i))\|^2 - \log\mathcal{N}(y; \lfloor\mu(f_{trunk}(x_i))\rfloor, \Sigma(\lfloor f_{trunk}(x_i)\rfloor)), \tag{7}$$

where $\lfloor\cdot\rfloor$ denotes the stop gradient operation that prevents the flow of gradients through the argument and f_{trunk} is the trunk of the network that provides input to both the mean and variance heads.

We therefore alter the Structure Module loss function as follows. Throughout training, the model minimises the FAPE loss

$$\mathcal{L}_{FAPE} = \frac{c}{M \cdot N}\sum_{ij}\min\left(\epsilon_{clamp}, \sqrt{\|\hat{x}_{ij} - x_{ij}\|^2 + \epsilon}\right),$$

where x_{ij} is the position of atom x_j relative to frame T_i, i ranges over the number of backbone frames which is equal to the number of residues in the protein (N) and j ranges over the total number of atoms (M).. We add on to this a negative log likelihood term to give the loss of Equation 5,

$$\mathcal{L}_{UQ} = \mathcal{L}_{FAPE} + \frac{1}{2}\log|\Sigma(\lfloor s\rfloor)| + \frac{c}{2}(\hat{x} - x)^T\Sigma^{-1}(\lfloor s\rfloor)(\hat{x} - x).$$

A.3 Training Details

Data: The models are trained on 90% of the length 11 loops with 30% of the training set used as a hold out validation set for early stopping. For each loop sequence in the training set, the model is trained on all associated conformations for that loop meaning that the model will observe multiple targets for a given sequence input. The test set has been chosen to include loop sequences that are a range of minimum edit distances ([30]) away from the points in the training and validation sets to simulate varying degrees of epistemic uncertainty.

Training: For each of the five models in the ensemble, the networks are trained using rectified RADAM which is a variant of ADAM that is more robust to the choice of learning rate [31]. The learning rate is adjusted by a cosine annealing with warm restarts scheduler with a 50 epoch warm up and learning rate values in the range of $[0.0001, 0.001]$. We train with a batch size of 32. We train for a maximum of 1000 epochs per training step and use early stopping with a patience of 60 epochs in the first stage of training and 75 epochs during finetuning. The FAPE loss, bond ideality loss, and clash loss weightings are all set to 1.

Feature	Description
`aa_loop`	The sequence of amino acids that refer to the loop structure.
`n_conformations`	The number of occurrences of this loop structure in proteins stored in the PDB.
`codes`	The four letter PDB codes that uniquely identify the proteins each conformation occurs in e.g *6rpt*.
`chains`	The identifiers of the protein chains in which the conformations occur.
`resis`	The residue numbers identifying the loop in the chain.
`n_conformations_.5`	The number of distinct conformations remaining after collapsing clusters within a 0.5Å RMSD of one another using agglomerative clustering.
`n_conformations_1.25`	The number of distinct conformations remaining after collapsing clusters within a 1.25Å RMSD of one another using agglomerative clustering.
`n_conformations_2`	The number of distinct conformations remaining after collapsing clusters within a 2Å RMSD of one another using agglomerative clustering.

Table 6: Names and descriptions of the features recorded in the loops dataset.

First the mean head of the network is trained with the standard FAPE loss and auxiliary losses and then the variance head is trained separately. We are able to do this due to the addition of the stop gradient in the variance loss function. The hyperparameters chosen for these models were chosen to be the same as the original model in the Structure Module as these values were carefully tuned to allow the model to make optimal predictions. The exceptions to this are the maximum number of epochs, which was reduced from 5000 to 1000, and reducing the patience of the model in early stopping. The reason for such decisions is because we expect that the problem of predicting a loop structure will be simpler than that of a whole protein as it is a much shorter length. Therefore, no hyperparameter tuning was performed.

A.4 Loops Dataset

The loops dataset consists of 68,390 amino acid sub-sequences that refer to loop regions that bridge beta sheets in protein structures. After removing entries with inconsistencies in the data in the form of missing sections, we have 60,240 total sub-sequences which is a reduction of 8,150 entries from the initial dataset.

For each subsequence, there exists a number of conformations of the loop that are observed in whole protein structures. Each conformation is an extract of whole protein structure, differing from one another only in the anchoring residues (two residues on either side of the residues common to all conformations). Figure 5 demonstrates an example of a set of conformations that refer to the same subsequence of amino acids within different proteins. As can be seen in Figure 6, the number of conformations for a given loop amino acid sequence is heavily positively skewed. The number of conformations per sequence ranges from one to 8031. The proportion of loop amino acid sequences that have five conformations or fewer is over a half at 54.3%, therefore there will be a limited number of conformations available for many loops. In total, there are 1,267,554 conformations.

A number of features in the loops dataset are concerned with the number of distinct conformations remaining after combining conformations within an RMSD threshold. These values provide us with a baseline indication of loop flexibility as we interpret the loop structures that retain a larger number of conformations at larger thresholds as having more conformational diversity and therefore more flexibility.

B Appendix Results

B.1 Further Results of Flexibility Estimation

The main body of the text contains examples of estimations of protein structure flexibility estimations that correspond with pLDDT values or empirical. variance estimates. Here, we provide examples of

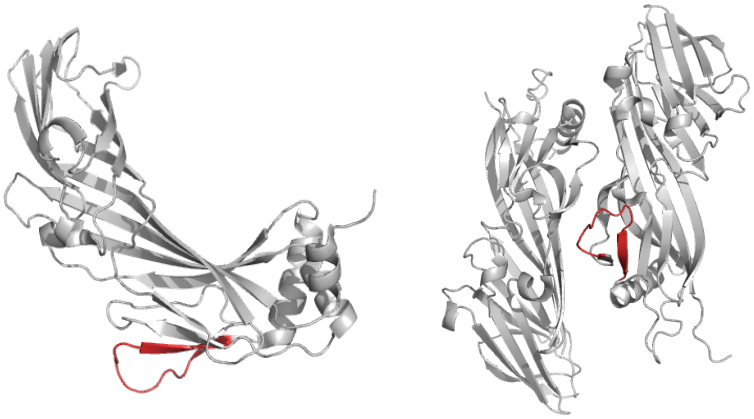

Figure 5: Example of the two conformations stored for the loop with amino acid sequence DNTQNDANTKE. The loops are highlighted in red. Images are generated by PyMOL [**?**].

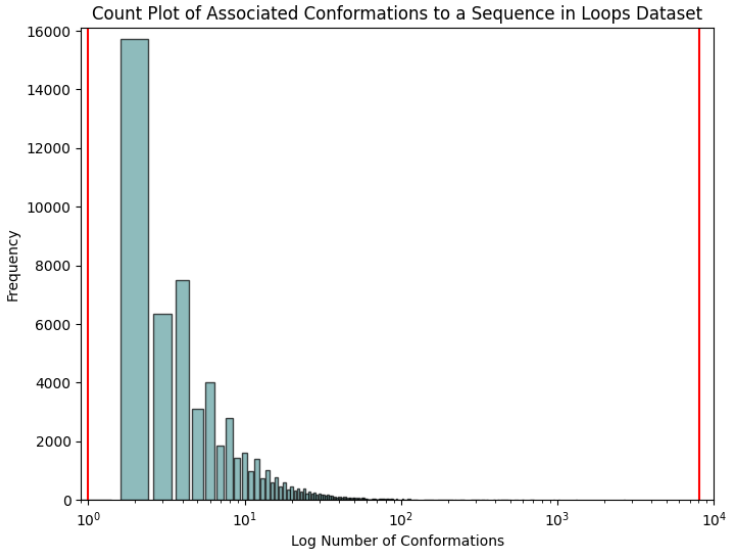

Figure 6: Count plot of how many conformations exist for a given loop amino acid sequence in the loops dataset. The red vertical lines denote the minimum and maximum log number of conformations associated with a sequence.

estimates of flexibility estimates in Figures 7, 8 that do not correlate with existing flexibility measures nor the empirical variance.

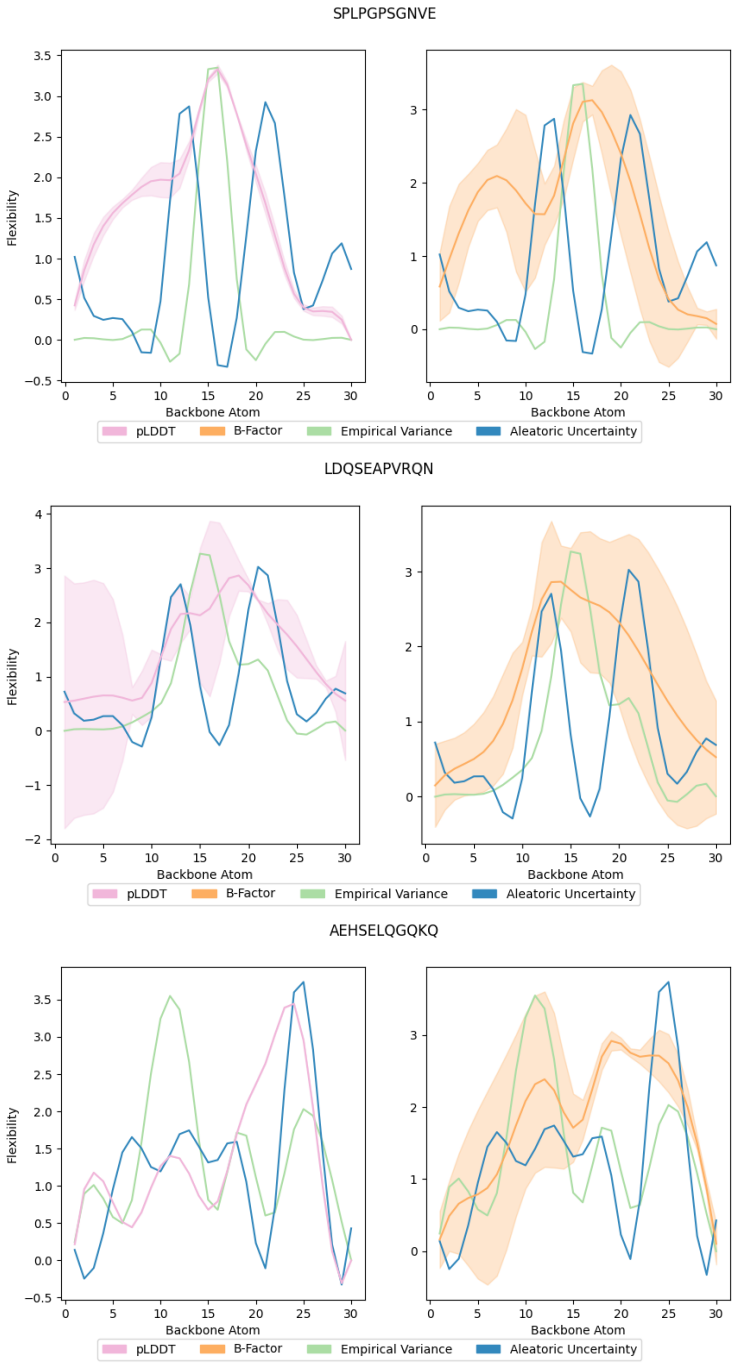

Figure 7: Figures of first set of examples of flexibility estimates made by model that do not imitate alternate flexibility estimation methods.

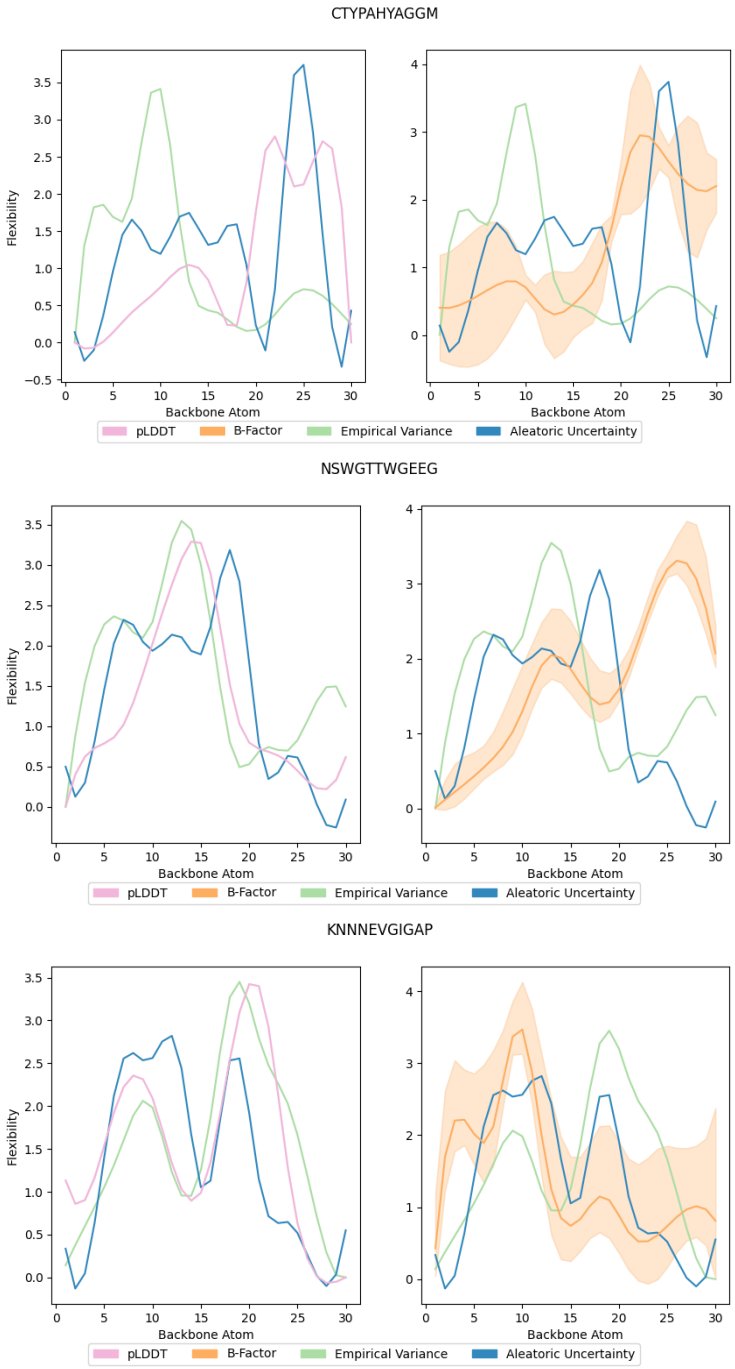

Figure 8: Figures of second set of examples of flexibility estimates made by model that do not imitate alternate flexibility estimation methods.

B.2 Impact of Epistemic Uncertainty

Heteroscedastic aleatoric uncertainty estimates are made dependent on a data input, as such if the model has a poor understanding on how the mean function behaves at this input it is likely the case that the aleatoric uncertainty estimates will not be truly reflective of the noise in the underlying data. We expect that high epistemic uncertainty will affect the quality of the aleatoric uncertainty estimates and so the epistemic uncertainty of a prediction should be considered when evaluating how well flexibility relates to aleatoric uncertainty.

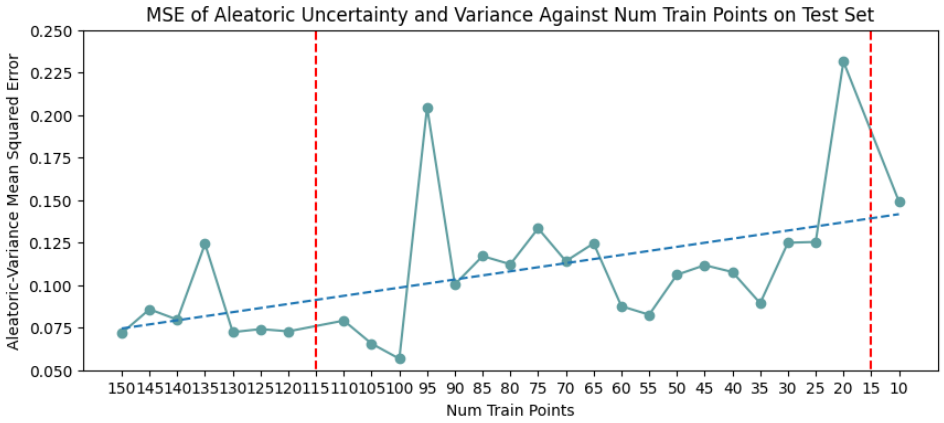

Figure 9: Line plot denoting the trend of the squared error of the estimated aleatoric uncertainty and value of the true variance function, averaged across test points, against the number of training points in the simple regression task. The blue dotted line is the line of best fit. Fewer training points indicates a higher epistemic uncertainty. We observe a trend of reduced quality of aleatoric uncertainty estimates as the epistemic uncertainty increase. The red dotted lines indicate where outliers are removed. At 115 train points, the MSE was 1.1291 and at 15 train points the MSE was 0.7783.

We perform a two-tailed hypothesis test to determine if there exists a relationship between epistemic uncertainty and the quality of the aleatoric uncertainty estimates. We perform a simple regression task with an ensemble of neural networks on a toy dataset with heteroscedastic noise. We vary the epistemic uncertainty by varying the number of training samples and assess the quality of the estimated aleatoric uncertainty through the squared error of the estimate and the ground-truth variance function.

$$\text{MSE}(\hat{\Sigma}, \Sigma) = \frac{1}{N}\sum_{i=1}^{N}(\hat{\sigma}_i - \sigma_i)^2, \tag{8}$$

where $\Sigma = \{\sigma_i\}_{i=1}^N$ is the ground truth variance at each of the N training points and $\hat{\Sigma}$ are the aleatoric uncertainty estimates calculated from the variance predictions of the ensemble of models.

Data: The data is sampled from the function $f(x) = \sin(x^2) + x\cos(2x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sin^2(x) + \frac{x}{8} + 0.01)$. The distribution of the data therefore has an increasing amount of variance in the noise the further from the origin. The number of training points differs across the 29 experiment runs, ranging from $n = 10$ to $n = 150$ in increments of 5. The test set consists of 150 samples from the function.

Model: The model is a feedforward network with 5 hidden layers and relu activations in the trunk of the network and an extra hidden layer in the variance head of the network. The output of the fifth hidden layer is passed to the mean head and the variance head. The mean head consists of a single linear layer. The variance head consists of two linear layers where the output is passed through a softplus activation to ensure positivity.

Training: We trained an ensemble of 100 models per experiment. Each were trained independently with different seeds. The training algorithm we used was stochastic gradient descent to minimise the loss function given in Equation 7 with a learning rate of 0.01 and a batch size of 20 (or 10 in the case of 10 training points) over 800 epochs.

As we cannot make an assumption of normality on the quality of the aleatoric uncertainty estimate and the number of training samples, we use the Spearman's rank correlation coefficient as the test statistic for a one-tailed test.

H_0 : The mean squared error of the aleatoric uncertainty and number of training points are not monotonically related.

H_1 : The mean squared error of the aleatoric uncertainty and number of training points have a positive monotonic relationship.

The mean squared error between aleatoric uncertainty estimates and ground truth variance are plotted in Figure 9 with the two outlier values removed. The correlation coefficient for this test (including the outliers) is 0.5631 with corresponds to a p-value of 0.0007 for $n = 29$. Therefore at the 0.1% significance level, there is strong evidence to suggest a positive correlation. This can be clearly seen in Figure 10 where the aleatoric uncertainty estimated for test set predictions are plotted along the bottom row. The aleatoric uncertainty estimates better match the trend in ground truth variance as the number of training samples increase. Near the origin, the estimates indicate small variance with the values growing as the x values increase in magnitude.

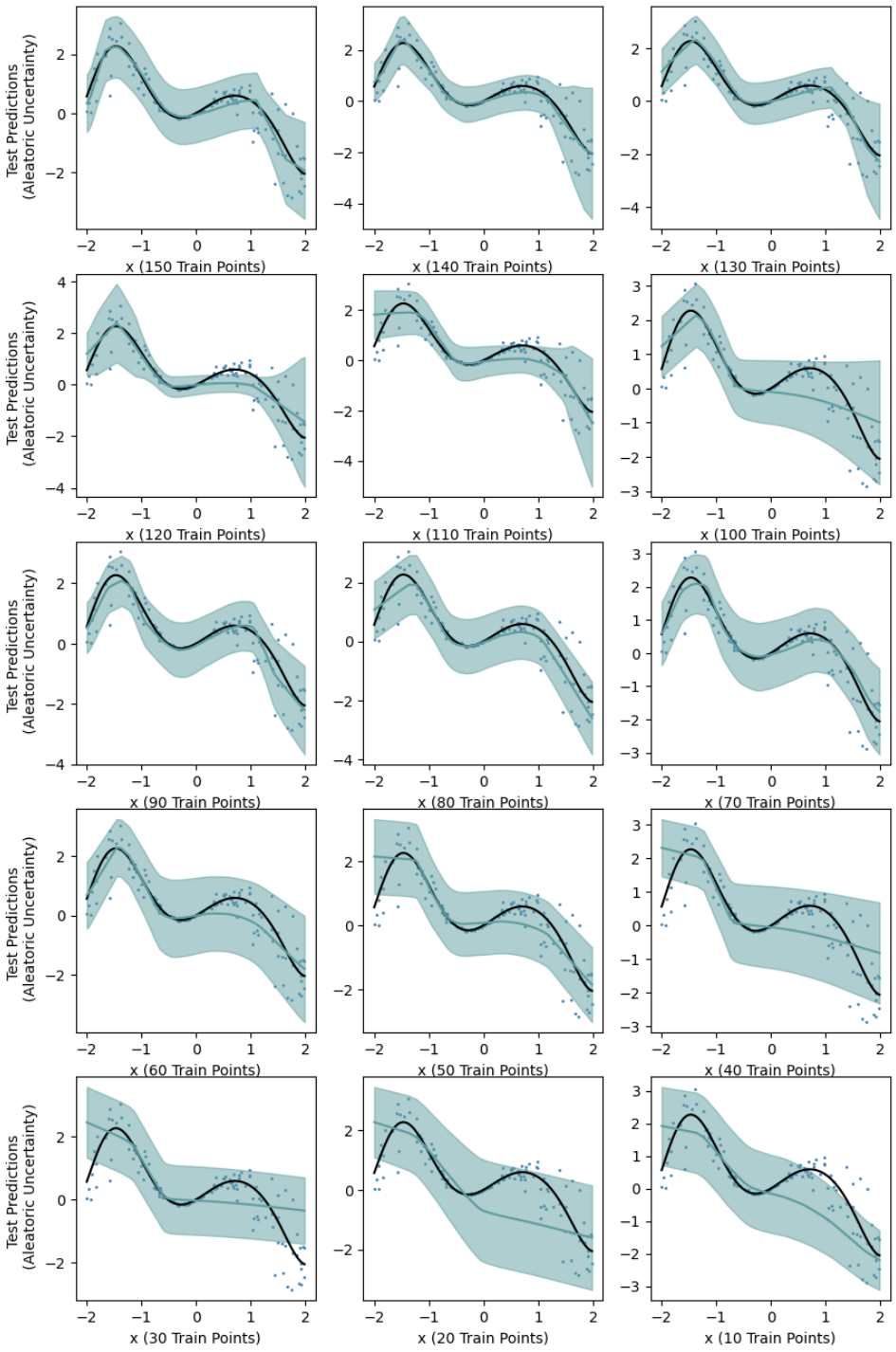

Figure 10: The predictive performance of the model in the simple regression task as the number of training points vary. The plots indicated the mean prediction (blue line) and the true mean (black line) with shaded regions giving the aleatoric uncertainty. As epistemic uncertainty increases, the aleatoric uncertainty estimates become a poorer estimate of ground truth variance.