Uncertainty Weighted Deep Ensemble to Enhance Protein Property Prediction

Alif Bin Abdul Qayyum[1] **Amir Hossein Rahmati**[1]
Xiaoning Qian[1,2,3] **Byung-Jun Yoon**[1,3]
[1]Department of Electrical and Computer Engineering, Texas A&M University
[2]Department of Computer Science and Engineering, Texas A&M University
[3]Computing and Data Sciences, Brookhaven National Laboratory

Abstract

Recent advances in machine learning (ML) have led to significant improvements in data-driven protein property prediction. While these ML models have demonstrated strong prediction performance on natural proteins, their practical utility still remains limited due to the absence of reliable uncertainty estimates for their predictions. In this work, we present **DUNE** (**D**eep **UN**certainty-weighted **E**nsemble), a method for incorporating predictive uncertainty into ML models, to achieve uncertainty-aware protein property prediction. We demonstrate how incorporating uncertainty estimates can enhance the overall predictive performance across three property prediction tasks; immunogenicity, toxicity and allergenicity. Experimental results show that our proposed DUNE outperforms existing ensemble based classification strategies.

1 Introduction

Recent advances in computational power and modeling has enabled not only accurate predictions of protein structures [29, 2, 35, 1] but also novel protein designs [9, 55]. The process of protein design typically involves changing the inherent properties and characteristics of proteins. Antibody design, for instance, frequently involves optimizing binding affinity for a target antigen to improve neutralization potency. Nevertheless, the inherent complexity of biological systems introduces the possibility of inadvertently altering undesirable properties during the design procedure.

Evaluating and monitoring protein properties are of significant importance, leading to substantial efforts in accurately and efficiently predicting safety-related protein properties. Especially, the advent of ML based protein structure prediction models, e.g. AlphaFold (AF), ESMFold, and protein language models (PLMs) [26, 47, 42, 35, 22] has inspired many scientists and engineers to use hidden representations of them to model relationships between proteins and their properties, since hidden representations are believed as rich enough features to capture the correlations. Examples include ToxDL [43], VenusVaccine [33], and AllergenAI [58], to name a few.

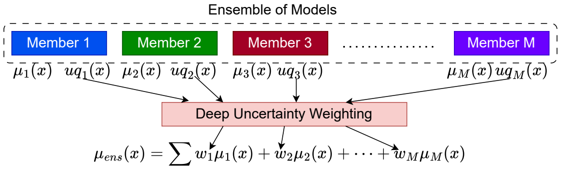

Figure 1: DUNE.

Machine learning (ML) models designed to predict protein properties are typically developed using a rigorous train/validation/test dataset split. This methodology involves training the model on the training set, subsequently tuning hyperparameters and performing initial performance assessments on the validation set, and finally evaluating generalization performance and robustness on a held-out test set. Despite these stringent evaluation protocols, concerns persist regarding the composition of

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference: Sampling Meets Learning.

these datasets, as highlighted by recent research [4]. A critical limitation, furthermore, is the general absence of uncertainty estimates from these predictors, which are indispensable for statistically sound decision-making in protein design and characterization.

Uncertainty quantification (UQ) is an active research area of studying how to quantify uncertainties in models, not limited to statistical models [13, 32], including physics-based models [57, 11, 48]. As models have grown larger and more nonlinear, specifically deep learning models, applications of UQ methods for classical statistical modeling [38, 20, 12] are computationally expensive and inaccurate.

Ensemble based prediction with ML models has been shown performing better than single learners. Further integration of uncertainty estimates into ensemble approaches can enable uncertainty-aware prediction for deep ensemble models. Also, most of the existing UQ metrics do not explicitly take into consideration the variations of predictions from probabilistic prediction models, necessitating a novel UQ metric specifically designed for probabilistic prediction models.

In this work, we explore these two aforementioned aspects:

- **DUNE:** We propose a novel methodology for protein property prediction, **D**eep **UN**certainty-weighted **E**nsemble, which integrates uncertainty estimation into deep ensemble models for enhanced protein property prediction.
- **CDiv:** We propose, **C**ross **Div**ergence, a novel uncertainty quantification metric specifically designed for deep probabilistic models that explicitly takes into account the variations in predictions, unlike many of the existing UQ metrics.

We showcase how incorporating uncertainty estimates can enhance predictive performance across three distinct property prediction tasks: immunogenicity, toxicity and allergenicity.

2 Method

We propose a deep ensemble classifier model that consists of several probabilistic binary classifier models, $\mathcal{M}_k$. The distinctiveness among these ensemble members lies in their representation of the input protein data. Specifically, each protein is characterized by hidden embeddings generated from a protein language model (PLM). To introduce diversity within the deep ensemble, we have employed five different PLMs, for each of the five ensemble members for its protein representations: ProstT5 [24], Ankh [15], ESM-2 [35], ProtTrans [14], and ESM-Cambrian [16]. This approach ensures that the individual models within the ensemble are exposed to different features of the proteins, potentially leading to a more robust and comprehensive collective prediction. In this study, we adopt **VenusVaccine** [33] model architecture for the backbone of each member of the ensemble.

2.1 DUNE

Each member $\mathcal{M}_k$ in the deep ensemble provides a predicted distribution, $P_{\mathcal{M}_k}(x) = \mathcal{N}(\mu_k(x), \sigma_k(x)^2)$, over positive class probability for protein data x where $\mu_k(x)$ is the mean and $\sigma_k(x)^2$ is the variance of the predicted distribution. To obtain the final prediction from the deep ensemble, we compute the weighted average of the individual predictions from each member within that ensemble.

$$\mu_{ens}(x) = \sum w_k(x)\mu_k(x) \tag{1}$$

The weights for each member in that ensemble are inversely proportional to the uncertainty, $\mathcal{UQ}_k(x)$, of the probabilistic prediction of that corresponding member:

$$w_k \propto \frac{1}{\mathcal{UQ}_k(x)} \tag{2}$$

We propose three different weighting schemes to determine the values of w_k as detailed below:

Unbiased Weighting: The weights, $w_k(x)$, can be set inversely proportional to the variance of predicted distribution:

$$w_k(x) \propto \frac{1}{\sigma_k^2(x)} \tag{3}$$

Negative Softmax Weighting: The weights, $w_k(x)$, are assigned proportionally to the negative softmax of the standard deviation of the predicted distribution:

$$w_k(x) \propto \exp(-c\sigma_k(x)) \tag{4}$$

KL-Divergence Weighting: Another weighting scheme is defined as follows:

$$w_k \propto D_{KL}(P_{\mathcal{M}_k}(x)||P_{unc}) \tag{5}$$

Here the weights, $w_k(x)$, are assigned proportionally to the KL-divergence between, $P_{\mathcal{M}_k}(x)$, the predicted distribution, and P_{unc}, a reference distribution. We use $P_{unc} = \mathcal{N}(0.5, \sigma^2_{unc})$ as the reference distribution. In our experiments, we set $\sigma_{\text{unc}} = 0.1$.

2.2 Cross Divergence

We model the prediction probability from an uncertain probabilistic binary classifier model following a Gaussian distribution $P_{unc} = \mathcal{N}(0.5, \sigma^2_{unc})$ for any binary classification problem. We evaluate the quantified uncertainty of each member in our deep ensemble model using the following equation for **CDiv** (**C**ross-**Div**ergence):

$$\begin{aligned} cd_k(x) = (y(x)\log(2\mu_k(x)) + (1-y(x))\log(2-2\mu_k(x))) \\ D_{KL}(\mathcal{N}(\mu_k(x), \sigma_k(x)^2)||\mathcal{N}(0.5, \sigma^2_{unc})) \end{aligned} \tag{6}$$

Here $y(x)$ denotes the true label for datapoint x.

The KL-divergence $D_{KL}(P_{\mathcal{M}_k}||P_{unc})$ quantifies the difference between two distributions $P_{\mathcal{M}_k}$ and P_{unc}, with low values indicating high similarity between the two distributions and vice versa. The multiplicative term, $y(x)\log(2\mu_k(x)) + (1-y(x))\log(2-2\mu_k(x))$, gives a positive value if $\mu_k(x)$ is at the correct side of the decision boundary, 0.5, whereas it gives a high negative value if $\mu_k(x)$ is at the incorrect side of the decision boundary. As a result, the cross-divergence term produces a high positive value if the predicted distribution has a mean close to the true label with a low variance, whereas a very high negative value if the predicted distribution has a mean close to the incorrect label with a low variance, indicating high value inferring low uncertainty and vice versa.

To evaluate the quantified uncertainty of an ensemble with M number of members, we take the average of the quantified uncertainty of each member of our ensemble:

$$cd(x) = \frac{1}{M}\sum_{k=1}^{M} cd_k(x) \tag{7}$$

This approach evaluates the quantified uncertainty of the deep ensemble by measuring the KL-divergence between the predicted distribution and the distribution of an uncertain binary classifier model for each classifier in that ensemble.

3 Experiments

Architecture: We followed **VenusVaccine** [33] as the backbone architecture for each member of the ensemble. To convert the backbone model into Bayesian Neural Network [28], we employ five probabilistic models: MC-Dropout [18], SVDKL [56], Laplace approximation [36], SWAG [37], and VBLL [21]. We altered the original VenusVaccine architecture by inserting an additional linear layer into the final MLP segment for DVBLL, LA, and SVDKL.

Datasets: We utilized 3 protein property datasets: (i) ImmunoDB [33], an immunogenicity database containing labeled antigens from bacterial, viral, and human sources, (ii) ToxDL 2.0 [60], a database containing labeled toxic and non-toxic proteins, (iii) SDAP 2.0 [41], adatabase containing labeled allergenic and non-allergenic proteins.

Setup: Each ensemble member $\mathcal{M}_k$ was independently optimized following Li et al. [33]. Separate ensembles were trained for each specific dataset. For MC-Dropout based implementations, we used a dropout rate of 0.1. The dimensions of the appended linear layers for LA, DVBLL and SVDKL are 64, 64 and 16 accordingly. We obtain the probabilistic prediction $P_{\mathcal{M}_k}$ through 64 MC sample predictions. We utilized deterministic VenusVaccine [33] model for the deterministic baselines.

Results: Table 1 shows the comparative prediction performances on immunogenic virus, independent toxicity and allergenicity datasets and Table 2 shows the UQ performances on all datasets. To compare the prediction performance we selected the following baselines: (i) Majority Voting, (ii) Soft Voting (Uniform Weighting), (iii) Performance Weighting [34], (iv) Single Deterministic Learner and (v) Uncertainty Voting [27]. For evaluating the prediction performance in Table 1, we only report the best performing combination of BNN and weighting starategy for DUNE and UVote according to accuracy metric. For Single Learners, we only report that PLM which obtained the best accuracy among the Single Learners.

Table 1: Results on protein property datasets. *MVote*, *SVote*, *PWeight* and *SL* refers to Majority Voting, Soft Voting, Performance Weighting and Single Learner methods accordingly. KLD denotes KL-Divergence weighting and NS denotes Negative-Softmax.

Dataset	Method	BNN	Weight Strategy	Accuracy(↑)	Precision(↑)	Recall(↑)	F1-Score(↑)	AUC-ROC(↑)
Virus	MVote	-	-	0.9232	0.9171	0.9330	0.9250	0.9805
	SVote	-	-	0.9345	0.9249	0.9479	0.9363	0.9809
	PWeight	-	-	0.9332	0.9268	0.9429	0.9348	0.9806
	SL(Ankh)	-	-	0.9131	**0.9380**	0.8957	0.9164	0.9582
	UVote	SVDKL	NS(c=5)	0.9320	0.9246	0.9429	0.9337	0.9650
	DUNE*	MCD	NS(c=5)	**0.9395**	0.9298	**0.9529**	**0.9412**	**0.9810**
Toxicity^I	MVote	-	-	0.9611	0.4378	0.7171	0.5436	0.9619
	SVote	-	-	0.9636	0.4603	**0.7237**	0.5627	**0.9637**
	PWeight	-	-	0.9664	0.4865	0.7105	0.5775	0.9630
	SL(ESM2)	-	-	0.9600	**0.7105**	0.4286	0.5347	0.9601
	UVote	DVBLL	KLD	0.9674	0.4974	0.6184	0.5513	0.9619
	DUNE*	DVBLL	KLD	**0.9698**	0.5269	0.6447	**0.5799**	0.9619
Allergenicity	MVote	-	-	0.7836	0.0353	**1.0000**	0.0682	0.9912
	SVote	-	-	0.9298	0.1014	**1.0000**	0.1841	0.9966
	PWeight	-	-	0.9204	0.0871	0.9545	0.1597	0.9951
	SL(ESMC)	-	-	0.8351	**1.0000**	0.0458	0.0876	0.9890
	UVote	DVBLL	NS(c=5)	0.6982	0.0256	**1.0000**	0.0499	0.9126
	DUNE*	DVBLL	NS(c=5)	**0.9802**	0.2857	**1.0000**	**0.4444**	**0.9997**

Table 2: Uncertainty Quantification Evaluation of DUNE.

UQ Metric	Model	Virus	Bacteria	Tumor	Toxicity^T	Toxicity^I	Allergenicity
CDiv(↑)	MCD	**6.6719**	0.5928	-3.6989	12.4150	9.8370	-62.6277
	SVDKL	3.5264	0.2320	0.0006	6.6525	6.1032	-10.3151
	DVBLL	5.6940	-5.5328	-0.6190	11.5257	10.3730	-29.3711
	LA	6.1067	0.4271	-1.8109	**13.4206**	**10.8600**	-46.1843
	SWAG	4.7570	**2.2087**	**1.6627**	8.5277	7.4408	**-3.4477**
ECE(↓)	MCD	**0.0060**	0.0964	0.1242	0.0057	0.0285	0.3302
	SVDKL	0.0470	0.1716	0.1782	0.0362	0.0540	0.3347
	DVBLL	0.0153	0.1118	0.1060	**0.0030**	**0.0185**	**0.2556**
	LA	0.0154	0.0940	0.1270	0.0031	0.0187	0.2878
	SWAG	0.0370	**0.0476**	**0.0742**	0.0182	0.0508	0.3662
NLL(↓)	MCD	**0.1788**	0.5159	0.5916	**0.0515**	0.1003	0.8755
	SVDKL	0.2140	0.5049	0.6917	0.0784	0.1160	0.7475
	DVBLL	0.2012	0.5624	0.5393	0.0534	**0.0860**	**0.5063**
	LA	0.2014	0.4784	0.5708	0.0551	0.0961	0.5701
	SWAG	0.1958	**0.4200**	**0.4960**	0.0600	0.1099	0.7281
Brier Score(↓)	MCD	**0.0496**	0.1345	0.1905	**0.0130**	0.0284	0.2414
	SVDKL	0.0544	0.1597	0.2493	0.0141	0.0270	0.2321
	DVBLL	0.0541	0.1359	0.1747	0.0141	**0.0236**	**0.1602**
	LA	0.0576	0.1353	0.1849	0.0142	0.0258	0.1890
	SWAG	0.0531	**0.1294**	**0.1655**	0.0147	0.0305	0.2524

Observations: DUNE outperformed the baselines at majority of the prediction metrics across all datasets, specifically at the Allergenicity dataset by large margins. Although DUNE lagged behind other methods in terms of precision and recall, those methods lagged in other metrics for all cases. Results also demonstrate that our proposed Cross-Divergence (CDiv) uncertainty quantification (UQ) metric generally aligns with other established UQ metrics, specially in Immuno-Virus, Bacteria and Tumor datasets, where MCD performed best in Immuno-Virus datasets at all metrics and SWAG performed best in Immuno-Bacteria and Tumor datasets. Although no one model obtained the best UQ performance across both Toxicity datasets; MCD, DVBLL and LA obtained consistent overall performances. CDiv's performance deviated the most from other metrics at the Allergenicity dataset.

4 Conclusion

Advancing data-driven and computationally intensive methods for evaluating protein properties is crucial for developing immunogenic therapeutics such as vaccines, where safety and efficacy are paramount. The rise of machine learning (ML) has opened new avenues in this field. When public health is at stake, uncertainty quantification (UQ) integrated in these AI/ML methods becomes vital, not just to gauge prediction reliability, but also to increase predictive performance. Our research introduces a novel methodology that enhances protein property prediction by integrating uncertainty estimates into deep ensemble models. We also propose a novel UQ metric specifically designed for evaluating probabilistic deep learning classifiers. Experimental evaluations indicate that integration of uncertainty estimates of probabilistic models into deep ensemble methods achieve superior performance than single learners and other deterministic deep ensemble approaches. Our approach focuses on epistemic uncertainty to achieve enhanced predictive performance. We recognize that addressing uncertainty in protein representation and its impact on prediction remains an important area for future research.

Funding Statement

This work has been supported by the Advanced Research Projects Agency for Health (ARPA-H) Award 1AY1AX000053-01.

References

[1] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

[2] M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[3] W. Braun and M. S. Venkatarajan. New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical-chemical properties. *Journal of Molecular Modeling*, 7(12):445–453, Dec. 2001. ISSN 0948-5023. doi: 10.1007/s00894-001-0058-5. URL `http://dx.doi.org/10.1007/s00894-001-0058-5`.

[4] M. Buttenschoen, G. M. Morris, and C. M. Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9): 3130–3139, 2024.

[5] B. Chen, X. Cheng, P. Li, Y.-A. Geng, J. Gong, S. Li, Z. Bei, X. Tan, B. Wang, X. Zeng, C. Liu, A. Zeng, Y. Dong, J. Tang, and L. Song. XTrimoPGLM: Unified 100b-scale pre-trained transformer for deciphering the language of protein. *bioRxiv*, July 2023.

[6] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020. URL `https://arxiv.org/abs/2003.10555`.

[7] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019. URL `https://arxiv.org/abs/1901.02860`.

[8] N. T. Daiki HIRATA. Ensemble learning in cnn augmented with fully connected subnetworks. *IEICE TRANSACTIONS on Information*, E106-D(7):1258–1261, July 2023. ISSN 1745-1361. doi: 10.1587/transinf.2022EDL8098.

[9] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. Wicky, A. Courbet, R. J. de Haas, N. Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL `https://arxiv.org/abs/1810.04805`.

[11] J. Dick, F. Y. Kuo, Q. T. Le Gia, and C. Schwab. Fast qmc matrix-vector multiplication. *SIAM Journal on Scientific Computing*, 37(3):A1436–A1450, 2015.

[12] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[13] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.

[14] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, and B. Rost. ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44 (10):7112–7127, Oct. 2022.

[15] A. Elnaggar, H. Essam, W. Salah-Eldin, W. Moustafa, M. Elkerdawy, C. Rochereau, and B. Rost. Ankh: Optimized protein language model unlocks general-purpose modelling, 2023. URL `https://arxiv.org/abs/2301.06568`.

[16] ESM Team. ESM Cambrian: Revealing the mysteries of proteins with unsupervised learning. `https://evolutionaryscale.ai/blog/esm-cambrian`, Dec. 2024. EvolutionaryScale Website.

[17] N. Ferruz, S. Schmidt, and B. Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nat. Commun.*, 13(1):4348, July 2022.

[18] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[19] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022.

[20] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6): 721–741, 1984. doi: 10.1109/TPAMI.1984.4767596.

[21] J. Harrison, J. Willes, and J. Snoek. Variational bayesian last layers. *arXiv preprint arXiv:2404.11599*, 2024.

[22] T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, page eads0018, 2025.

[23] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, and B. Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, 20(1):723, Dec. 2019.

[24] M. Heinzinger, K. Weissenow, J. Sanchez, A. Henkel, M. Mirdita, M. Steinegger, and B. Rost. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6(4):lqae150, 11 2024. ISSN 2631-9268. doi: 10.1093/nargab/lqae150. URL `https://doi.org/10.1093/nargab/lqae150`.

[25] S. Hellberg, M. Sjoestroem, B. Skagerberg, and S. Wold. Peptide quantitative structure-activity relationships, a multivariate approach. *Journal of Medicinal Chemistry*, 30(7):1126–1135, 1987. doi: 10.1021/jm00390a003. URL `https://doi.org/10.1021/jm00390a003`. PMID: 3599020.

[26] B. Hie, E. D. Zhong, B. Berger, and B. Bryson. Learning the language of viral evolution and escape. *Science*, 371(6526):284–288, 2021.

[27] Y. Jiang, V. S. F. Garnot, K. Schindler, and J. D. Wegner. Uncertainty voting ensemble for imbalanced deep regression. In *DAGM German Conference on Pattern Recognition*, pages 329–343. Springer, 2024.

[28] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17 (2):29–48, 2022. doi: 10.1109/MCI.2022.3155327.

[29] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[30] L. I. Kuncheva and J. J. Rodríguez. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 38(2):259–275, Dec. 2012. ISSN 0219-3116. doi: 10.1007/s10115-012-0586-6. URL `http://dx.doi.org/10.1007/s10115-012-0586-6`.

[31] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020. URL `https://arxiv.org/abs/1909.11942`.

[32] W. Lee, H. Kim, S. Hwang, A. Zanobetti, J. D. Schwartz, and Y. Chung. Monte carlo simulation-based estimation for the minimum mortality temperature in temperature-mortality association study. *BMC medical research methodology*, 17:1–10, 2017.

[33] S. Li, Y. Tan, S. Ke, L. Hong, and B. Zhou. Immunogenicity prediction with dual attention enables vaccine target selection. *arXiv preprint arXiv:2410.02647*, 2024.

[34] Y. Li and Y. Luo. Performance-weighted-voting model: An ensemble machine learning method for cancer type classification using whole-exome sequencing mutation. *Quantitative Biology*, 8(4):347–358, 2020. doi: https://doi.org/10.1007/s40484-020-0226-1. URL `https://onlinelibrary.wiley.com/doi/abs/10.1007/s40484-020-0226-1`.

[35] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[36] D. J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 05 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448. URL `https://doi.org/10.1162/neco.1992.4.3.448`.

[37] W. Maddox, T. Garipov, P. Izmailov, D. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning, 2019. URL `https://arxiv.org/abs/1902.02476`.

[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6): 1087–1092, 1953.

[39] I. D. Mienye and Y. Sun. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10:99129–99149, 2022. doi: 10.1109/ACCESS.2022.3207287.

[40] B. Naderalvojoud and T. Hernandez-Boussard. Improving machine learning with ensemble learning on observational healthcare data. In *AMIA Annual Symposium Proceedings*, volume 2023, page 521, 2024.

[41] S. S. Negi, C. H. Schein, and W. Braun. The updated structural database of allergenic proteins (sdap 2.0) provides 3d models for allergens and incorporated bioinformatics tools. *Journal of Allergy and Clinical Immunology: Global*, 2(4):100162, 2023. ISSN 2772-8293. doi: https://doi.org/10.1016/j.jacig.2023.100162. URL `https://www.sciencedirect.com/science/article/pii/S2772829323000875`.

[42] T. H. Olsen, I. H. Moal, and C. M. Deane. Ablang: an antibody language model for completing antibody sequences. *Bioinformatics Advances*, 2(1):vbac046, 2022.

[43] X. Pan, J. Zuallaert, X. Wang, H.-B. Shen, E. P. Campos, D. O. Marushchak, and W. De Neve. Toxdl: deep learning using primary structure and domain embeddings for assessing protein toxicity. *Bioinformatics*, 36(21):5159–5168, 2020.

[44] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL `https://arxiv.org/abs/1910.10683`.

[45] R. Rao, J. Meier, T. Sercu, S. Ovchinnikov, and A. Rives. Transformer protein language models are unsupervised structure learners. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=fylclEqgvgd`.

[46] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15): e2016239118, 2021. doi: 10.1073/pnas.2016239118. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2016239118`.

[47] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[48] H. Shin and M. Choi. Physics-informed variational inference for uncertainty quantification of stochastic differential equations. *Journal of Computational Physics*, 487:112183, 2023.

[49] M. Steinegger, M. Mirdita, and J. Söding. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods*, 16(7):603–606, July 2019.

[50] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

[51] A. Thuy and D. F. Benoit. Fast and reliable uncertainty quantification with neural network ensembles for industrial image classification. *Annals of Operations Research*, pages 1–27, 2024.

[52] M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, J. Lee, C. L. M. Gilchrist, J. Söding, and M. Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, 42(2):243–246, May 2023. ISSN 1546-1696. doi: 10.1038/s41587-023-01773-0. URL `http://dx.doi.org/10.1038/s41587-023-01773-0`.

[53] M. Varadi, D. Bertoni, P. Magana, U. Paramval, I. Pidruchna, M. Radhakrishnan, M. Tsenkov, S. Nair, M. Mirdita, J. Yeo, et al. Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic acids research*, 52(D1): D368–D375, 2024.

[54] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher, and N. F. Rajani. Bertology meets biology: Interpreting attention in protein language models, 2021. URL `https://arxiv.org/abs/2006.15222`.

[55] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[56] A. G. Wilson, Z. Hu, R. R. Salakhutdinov, and E. P. Xing. Stochastic variational deep kernel learning. *Advances in neural information processing systems*, 29, 2016.

[57] D. Xiu and G. E. Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.

[58] C. Yang, S. S. Negi, C. H. Schein, W. Braun, and P. Kim. Allergenai: a deep learning model predicting allergenicity based on protein sequence. *bioRxiv*, pages 2024–06, 2024.

[59] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020. URL `https://arxiv.org/abs/1906.08237`.

[60] L. Zhu, Y. Fang, S. Liu, H.-B. Shen, W. De Neve, and X. Pan. Toxdl 2.0: Protein toxicity prediction using a pretrained language model and graph neural networks. *Computational and Structural Biotechnology Journal*, 27:1538–1549, 2025. ISSN 2001-0370. doi: https://doi.org/10.1016/j.csbj.2025.04.002. URL `https://www.sciencedirect.com/science/article/pii/S2001037025001230`.

A Deep Uncertainty Weighted Ensemble

We briefly describe the three weighting approaches proposed in this work:

A.1 Unbiased Weighting

The weights, $w_k(x)$, can be set inversely proportional to the variance of predicted distribution:

$$w_k(x) \propto \frac{1}{\sigma_k^2(x)} \tag{8}$$

The predicted variance, σ_{ens}^2 of the ensemble model is $\sum w_k^2 \sigma_k^2$. Setting the weights w_k inversely proportional to corresponding member predicted variances makes σ_{ens}^2 a constant, resulting μ_{ens} being unbiased to any member's variations in predictions.

A.2 Negative Softmax Weighting

The weights, $w_k(x)$, are assigned proportionally to the negative softmax of the standard deviation of the predicted distribution:

$$w_k(x) \propto \exp(-c\sigma_k(x)) \tag{9}$$

The parameter c acts as a control parameter, modulating the influence of the member-specific weights, w_k. A higher value of c amplifies the contribution of members with greater predictive certainty, effectively giving more "mass" to their predictions within the ensemble. This mechanism allows the ensemble to prioritize models that are more confident in their predictions. Figure 2 summarizes the effect of c on weights for corresponding members in the ensemble, showing that high c amass higher proportion of total weights on more certain members of the ensemble whereas lower c value flattens the weights among all members in the ensemble. Effectively $c = \infty$ evolves into **Uncertainty Voting** method, and $c = 0$ evolves into **Soft Voting**. Negative softmax weighting balances between these two extremes with different values of c in the range of $(0, \infty)$.

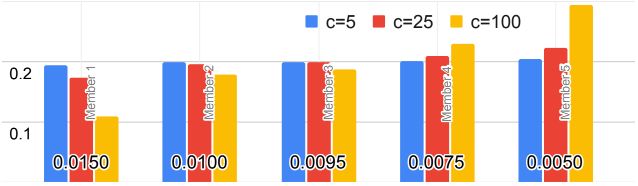

Figure 2: Negative Softmax Weighting. $x-$axis shows the standard deviations σ_k of predicted distributions by models within ensemble, and $y-$axis shows the weights w_k for different values of c.

A.3 KL-Divergence Weighting

Another weighting scheme is defined as follows:

$$w_k \propto D_{KL}(P_{\mathcal{M}_k}(x) || P_{unc}) \tag{10}$$

Here the weights, $w_k(x)$, are assigned proportionally to the KL-divergence between, $P_{\mathcal{M}_k}(x)$, the predicted distribution, and P_{unc}, a reference distribution. We use $P_{unc} = \mathcal{N}(0.5, \sigma_{unc}^2)$ as the reference distribution. This reference distribution is chosen based on the intuition that, in a classification setting, a model lacking knowledge about an input should assign approximately equal probability to each class. However, in rare cases where the input is inherently ambiguous, the model may confidently predict a balanced probability (e.g., 0.5 in binary classification), which should be reflected in a *low predictive variance*. In contrast, when the model is uncertain due to lack of evidence, it is expected not only to assign equal probabilities, but also to exhibit *high variance* in its prediction. Accordingly, the variance σ_{unc}^2 in the reference distribution $\mathcal{N}(0.5, \sigma_{\text{unc}}^2)$ controls the sensitivity of the proposed metric to uncertainty. In our experiments, we set $\sigma_{\text{unc}} = 0.1$. The reason for selecting $\sigma_{\text{unc}} = 0.1$ is the fact that under the probability distribution $\mathcal{N}(z; \mu = 0.5, \sigma = 0.1)$, the probability $P(0.0 \leq z \leq 1.0)$ is almost equal to 1. For a binary classification problem, any value for predicted positive class probability, $\hat{y}(x)$, outside the range of $[0, 1]$ does not make sense. Also, a σ_{unc} less than 0.1 would make the model less uncertain.

B Intuition behind Cross-Divergence Metric

The intuition behind the new Cross-Divergence metric comes from the fact that probabilistic models with high confidence should predict the distribution, $P_{\mathcal{M}_k}(x) = \mathcal{N}(\mu_k(x), \sigma_k(x)^2)$, with a mean $\mu_k(x)$ close to the true label $y(x)$ and with low variance. To quantify this, we evaluate the distance between predicted distribution and the distribution an uncertain model would provide. However, a model can predict incorrectly and with extreme confidence (mean close to the incorrect label and with a low variance). That's why we multiply the KL-divergence term with a multiplicative term that produces a positive value for correct predictions and a high negative value for incorrect prediction. Existing UQ metrics do not explicitly take into account the variations in predictions, which is an indication of the uncertainty of the model in its predictions. A highly certain probabilistic model is supposed to show low variations in its predictions. For example, most of the existing UQ metrics only take into account the predicted μ's , but not the predicted σ's. So if we have a probabilistic model that gives us the distribution $\mathcal{N}(0.8, 0.01)$, we would intuitively consider that this is more confident than a model which gives distribution of $\mathcal{N}(0.8, 0.05)$. However, most of existing UQ metrics would identify both models as equally confident.

C Behavior of an Uncertain Binary Classification Model

An uncertain binary classification model would behave in a way that its prediction would follow a Normal distribution $\mathcal{N}(0.5, \sigma_{\text{unc}})$ with a sufficiently high value for σ_{unc}. The reason for selecting $\sigma_{\text{unc}} = 0.1$ is the fact that under the probability distribution $\mathcal{N}(z; \mu = 0.5, \sigma = 0.1)$, the probability $P(0.0 \leq z \leq 1.0)$ is almost equal to 1. For a binary classification problem, any value for predicted positive class probability outside the range of [0,1] does not make any sense. Also, a σ_{unc} less than 0.1 would make the model less uncertain. That's why we model the predicted positive class probability from an uncertain model as $\hat{y}_{\text{unc}} \sim \mathcal{N}(0.5, 0.1)$. The objective of quantification of predictive uncertainty of a trained model is to identify how differently the trained model behaves from an uncertain model.

To further demonstrate this, we consider the perspective of *evidence theory*, particularly in *evidential deep learning*, where a similar intuition emerges. Assuming a symmetric Beta prior for binary classification with parameters $(a_1 = a_2 = a)$, the variance of the resulting Beta distribution becomes: Var $= \frac{a^2}{(2a)^2(2a+1)} = \frac{1}{4(2a+1)}$. This gives us a principled way to calibrate the degree of uncertainty: the smaller the a, the higher the variance, and thus the more "uncertain" the belief. This can then be mapped to a Gaussian approximation with a corresponding standard deviation and use it to define a reference distribution: $\mathcal{N}(0.5, \sigma_{\text{unc}})$.

So whether from a probabilistic (Gaussian) or evidential (Beta) viewpoint, the key idea is to define a *canonical uncertainty distribution* that is centered at 0.5 and sufficiently **broad** to represent maximal ignorance, and then use this distribution as a basis for quantifying how certain or uncertain our model predictions are.

D Uncertainty Quantification

All models and data are inherently imperfect. This imperfection primarily stems from two sources: the underlying assumptions made during model derivation and measurement errors present in the data collection process. Accurately assessing these uncertainties can significantly enhance the reliability of model predictions. Uncertainty quantification (UQ) aims to estimate the confidence in a Deep Neural Network (DNN) prediction, going beyond just its accuracy. However, quantifying these uncertainties is often non-trivial. UQ methods are typically problem-specific and can be computationally expensive to implement.

The most commonly utilized way to address model uncertainty is through a Bayesian neural network (BNN) [28]. A BNN accounts for parameter uncertainty by placing a prior distribution over its model parameters. The goal is then to infer the posterior distribution of these parameters, which provides a theoretical basis for understanding the model's inherent uncertainty.

To address the high computational and memory demands of Bayesian Neural Networks (BNNs), various approximation methods have been developed. In this work, we focus on five representative methods, namely MC-Dropout [18], SVDKL [56], Laplace approximation [36], SWAG [37], and VBLL [21], which we briefly summarize in the following:

Monte-Carlo Dropout (MC-DROPOUT) estimates uncertainty by interpreting stochastic forward passes as approximate Bayesian inference in deep Gaussian processes.

Stochastic Variational Deep Kernel Learning (SVDKL) combines deep networks with Gaussian processes using stochastic variational inference for scalable, flexible uncertainty modeling.

Laplace Approximation (LA) fits a Gaussian around the MAP estimate using the Hessian of the log-posterior to model uncertainty efficiently.

Stochastic Weight Averaging Gaussian (SWAG) uses SGD trajectories to approximate a Gaussian posterior over weights, enabling Bayesian model averaging with low overhead.

Variational Bayesian Last Layers (VBLL) maintains a posterior only over the last layer via a deterministic variational approach, yielding fast, sampling-free uncertainty estimates.

E Deep Ensemble

A deep ensemble model leverages the "wisdom of crowds" by combining the predictions of multiple individual deep neural networks (DNNs) [19, 39]. Instead of relying on a single, potentially overconfident, model, a deep ensemble trains several distinct DNNs, often with different random initializations, data subsets, or even architectures. Deep ensemble can achieve greater predictive accuracy, improved robustness, and more reliable uncertainty estimates by averaging or combining the outputs of these diverse member models than any single component model [8, 40]. This is because different models within the ensemble may capture distinct aspects of the data and make uncorrelated errors, leading to a more robust consensus. Deep ensembles are particularly effective for tasks requiring uncertainty quantification, as the variability in predictions across the ensemble members can provide a measure of confidence [51].

Predictions for classification tasks from any deep ensemble model can be obtained through multiple different approaches. For the sake of comparative evaluation with existing deep ensembles, we choose four existing approaches as baselines. Brief descriptions of those are as follows:

(i) **Majority Voting**: This refers to the scenario where we take the verdict of the majority among members in an ensemble of deterministic classifier models [30].

(ii) **Soft Voting**: In another variant of voting based approaches, we take the verdicts of all members on class probabilities and take the average of them, more generally termed as *Soft Voting*, and also known as *Uniform Weighting*.

(iii) **Performance Weighting**: This strategy refers to weighted averaging of predictions from models of an ensemble based on the performance of the models on a held-out dataset, e.g. validation set [34]. This is typically a two-step optimization process: the parameters of the ensemble members are first optimized using the train set, and then member-specific weights are optimized based on each member's performance on a held-out validation dataset.

(iv) **Uncertainty Voting**: Model uncertainty is generally not accounted for in standard majority voting, soft voting or performance weighted schemes. *Uncertainty Voting* is a strategy where an ensemble's final decision is determined by the least uncertain member [27]. Typically, a Bayesian Neural Network (BNN) variant is used to estimate each model's uncertainty. The simplest way to identify the least uncertain model is by finding the one with the smallest variation in its probabilistic predictions.

F Protein Language Model

The application of large language models (LLMs)—originally transformative in natural language processing (NLP)—to protein sequences has led to the development of sophisticated protein language models (PLMs) [17, 46, 23]. This modeling approach treats amino acids as analogous to words and full protein sequences as sentences, enabling the use of language modeling techniques in a biological context [54, 45]. Typically trained in a self-supervised manner on large-scale amino acid datasets, these PLMs learn rich contextual representations of residues [46, 23]. As a result, they can function as general-purpose feature extractors for various protein analysis tasks, such as protein fold classification, binding site identification, sub-cellular localization, property and structure prediction.

In this work we take advantage of five most recent state-of-the-art (SOTA) PLMs to acquire protein encodings for use in VenusVaccine [33], the backbone in our experiments. Particularly we use ProstT5 [24], Ankh [15], ESM-2 [35], ProtTrans [14], and ESM-Cambrian [16]. Below, we provide a brief description of each.

ProstT5: This is a bilingual PLM, based on encoder-decoder T5 [44] model, that is fine-tuned with the goal to translate between amino acids and structural 3Di tokens—introduced by Foldseek [52]. Particularly, it is trained on a non-redundant subset of high confidence protein structures from AlphaFold structure Database (AFDB) [53]. The extracted sequence and structure embeddings from the encoder then can be used in downstream tasks.

Ankh: A family of encoder-only Protein Language Models (PLMs) designed for compute efficiency and strong generalization. It follows an ESM-2-like architecture, is pretrained on UniRef50 [50] using masked language modeling (MLM), and scales from 5M to 650M parameters. Despite its smaller size, Ankh matches or outperforms larger models like ESM-2 and xTrimoPGLM [5] by adhering to compute-optimal scaling laws. The authors show that performance does not scale linearly with size—well-optimized small models can be more effective than their larger counterparts.

ESM-2: A large-scale transformer-based Protein Language Model (PLM) trained with a masked language modeling (MLM) objective on evolutionary-scale sequence data, scaling up to 15 billion parameters. It directly predicts atomic-level 3D structures from primary sequences, bypassing the need for multiple sequence alignments (MSAs). As the model scales, structural information emerges implicitly in its embeddings, enabling accurate and fast structure prediction—up to 60x faster than AlphaFold2 [29]—while maintaining comparable resolution and confidence metrics such as pLDDT. These embeddings can be used with lightweight heads for downstream tasks, making ESM-2 a general-purpose backbone for structure prediction pipelines and allowing generalization across metagenomic and diverse protein families.

ProtTrans: This work investigates scaling protein language models using both autoregressive (Transformer-XL [7], XLNet [59]) and auto-encoding models (BERT [10], ALBERT [31], ELECTRA [6], T5 [44]) trained on UniRef and BFD datasets [49, 50]. These PLMs are pretrained in a self-supervised manner, reconstructing corrupted tokens from raw protein sequences where single amino acids act as input tokens. Embeddings—vector representations from the last hidden layer—are extracted and used as exclusive input to downstream models for tasks like secondary structure prediction, subcellular localization, and solubility classification. Auto-encoding models that leverage bidirectional context generally outperform uni-directional autoregressive models, highlighting the importance of capturing full contextual information in protein sequences. In our experiments, we used the ProtBert model for extracting embeddings.

ESM Cambrian: A generative PLM family developed alongside ESM-3 [22], designed to learn representations that capture the underlying biology of proteins. It improves upon ESM-2 by scaling up both training data and compute, and is available in 300M, 600M, and 6B parameter versions. Trained using a masked language modeling objective, ESM Cambrian learns biological structure and function from unlabeled protein sequences by capturing patterns shaped by evolution. These internal representations reflect the hidden variables driving amino acid selection, enabling broader generalization than models relying only on labeled structural or functional data.

G Experiments

G.1 Ensemble Member Architecture

We followed the Dual Attention mechanism based **VenusVaccine** model architecture proposed by Li et al. [33], as the backbone architecture for each member of the ensemble. The VenusVaccine model architecture uses a multi-modal approach for input protein data representation. The three modalities are:
(i) *Sequence Embedding:* We used pre-trained protein language model (PLM) to extract embeddings that represent protein sequences. For example, a protein sequence of length L processed by a pre-trained PLM yields an $L \times V$ representation, where each Amino Acid is represented by a V-dimensional vector. In our proposed ensemble model, each member uses a different PLM to extract

sequence embedding.

$$E_{\text{seq}} = PLM_{\text{emb}}(x_{seq}) \tag{11}$$

(ii) *Structural Embedding:* We use ESM3 [22] and FoldSeek [52] model to extract structural features from protein structures predicted by ESMFold [35].

$$x_{\text{structure}} = ESMFold(x_{\text{seq}}) \tag{12}$$

$$E_{\text{esm3}} = ESM3_{\text{emb}}(x_{\text{structure}}) \tag{13}$$

$$x_{\text{esm3}} = CrossAttention_{\theta}^{\text{esm3}}(E_{\text{seq}}, E_{\text{esm3}}) \tag{14}$$

$$E_{\text{foldseek}} = FoldSeek_{\text{emb}}(x_{\text{structure}}) \tag{15}$$

$$x_{\text{foldseek}} = CrossAttention_{\theta}^{\text{foldseek}}(E_{\text{seq}}, E_{\text{foldseek}}) \tag{16}$$

(iii) *Physicochemical Descriptors:* Five hand-crafted E-descriptors [3] and three Z-descriptors [25] offer an Amino Acid-level summary of key physicochemical properties, including hydrophobicity and secondary structure propensity.

$$E_{\text{ez}} = EZ_{\text{descriptor}}(x_{\text{seq}}) \tag{17}$$

$$x_{ez} = MLP_{\theta}^{ez}(CONCAT(E_{\text{seq}}, E_{\text{ez}})) \tag{18}$$

These features are then concatenated and passed though a MLP network to classify the label of the corresponding protein.

$$\hat{y} = f_{\theta}(CONCAT(E_{\text{seq}}, x_{\text{esm3}}, x_{\text{foldseek}}, x_{ez}) \tag{19}$$

For probabilistic variant of this architecture, we treat parameters of f_{θ} as a random variable to enable uncertainty estimation.

G.2 Datasets

We evaluate our models across three protein property datasets.

G.2.1 Immunogenicity

We utilized ImmunoDB [33], a comprehensive immunogenicity database containing 7,216 labeled antigens from bacterial, viral, and human sources. Each antigen is categorized as either immunogenic (positive) or non-immunogenic (negative). ImmunoDB was compiled through a meticulous process involving literature curation, database mining, and bioinformatics filtering. The majority of positive samples were sourced from previously published studies. To maintain high data quality, redundant sequences and samples from ambiguous regions were removed. This rigorous curation yielded three distinct subsets: Immuno-Virus, Immuno-Bacteria, and Immuno-Tumor. In total, this dataset provides **913/1562** positive/negative instances in Immuno-Bacteria, **2078/1886** in Immuno-Virus, and **300/477** in Immuno-Tumor.

G.2.2 Toxicity

We utilized ToxDL 2.0 [60] dataset, a comprehensive toxicity database containing labeled toxic and non-toxic proteins. This dataset is split into two half according to the date of collection. The first part consists of proteins collected before January 1, 2022:

(i) *Training Set:* containing **4,879** toxic and **9,637** non-toxic proteins.

(ii) *Validation Set:* containing **76** toxic and **837** non-toxic proteins.

(iii) *Test Set:* containing **110** toxic and **1,696** non-toxic proteins.

The second split of the data consists of proteins collected after the date of January 1, 2022:

(iv) *Independent Set:* containing **152** toxic and **4,547** non-toxic proteins.

The original dataset has few more labeled very long protein sequences, which we excluded from our utilized dataset due to resource constraints while predicting protein structures using ESMFold.

G.2.3 Allergenicity

We utilized SDAP 2.0 [41] dataset, a comprehensive allergenicity database containing labeled allergenic and non-allergenic proteins. We split this dataset into three distinct sets:

(i) *Training Set:* containing **7,191** allergenic and **7,068** non-allergenic proteins.

(ii) *Validation Set:* containing **2,397** allergenic and **2,356** non-allergenic proteins.

(iii) *Test Set:* containing **22** allergenic and **2,755** non-allergenic proteins, all from Cupin family.

This split was done following the experimental dataset setup conducted in AllergenAI [58].

G.3 Experimental Settings

Each ensemble member $\mathcal{M}_k$ was independently optimized using the technique proposed by Li et al. [33]. Separate ensembles were trained for each specific dataset:

- **Three** for the ImmunoDB dataset, one for each of the Immuno-Virus, Immuno-Bacteria, and Immuno-Tumor datasets.

- **One** for the toxicity dataset.

- **One** for the allergenicity dataset.

For dropout based implementations, we used a dropout rate of 0.1, and only applied it before the final classification head linear layer. For DVBLL, LA, and SVDKL, we altered the original VenusVaccine architecture by inserting an additional linear layer into the final MLP segment. Only this new layer functions as the probabilistic segment, to achieve consistent training behavior with minimal computational burden, and its dimensions for each model are:

LA	DVBLL	SVDKL
64	64	16

We obtain the probabilistic prediction $P_{\mathcal{M}_k}$ through 64 MC sample predictions. For the deterministic baselines, we utilized deterministic VenusVaccine [33] model with protein sequence embeddings from 5 different PLMs.

G.4 Experimental Results

All experiments conducted in this work can be summarized as follows:

- We compared our proposed **DUNE** approach with three deterministic ensemble baselines: *Majority Voting*, *Soft Voting* and *Performance Weighting*, one uncertainty based probabilistic ensemble baseline: *Uncertainty Voting*, and also the deterministic single learner. Section G.4.1 discusses obtained results.
- We compared the three uncertainty weighting schemes: (i) *KL-Div*, (ii) *Negative-softmax* and (iii) *Unbiased weighting*, discussed in Section A, and explained obtained results in Section G.4.2.
- We compared several different BNNs in our proposed DUNE approach, evaluating both their predictive and UQ performances in Sections G.4.3 and G.4.4.

G.4.1 Comparison with Baselines

Table 3 shows the comparative results on immunogenic virus, bacteria, tumor datasets, along with toxicity and allergenicity datasets. For DUNE and UVote approaches, we found that the combination of weighting strategy and BNN produced the highest accuracy. For the single learner method, we reported those PLM embeddings for different datasets that achieved the highest accuracy among five different PLM embeddings. Following the approach mentioned in *Performance Weighting* [34], we measured the member-specific weights by training a linear regression model on the validation dataset predictions.

Our proposed DUNE model outperformed the baselines at majority of the metrics across all datasets. Apart from test toxicity dataset, DUNE produced higher accuracy across all datasets. Even for test toxicity dataset, the difference in performance between the best performing PWeight and DUNE is very small. Also DUNE had higher AUC-ROC than PWeight method for this dataset. Although

Table 3: Results on protein property datasets. *MVote*, *SVote*, *PWeight* and *SL* refers to Majority Voting, Soft Voting, Performance Weighting and Single Learner methods accordingly. KLD denotes KL-Divergence weighting and NS denotes Negative-Softmax.

Dataset	Method	BNN	Weight Strategy	Accuracy(↑)	Precision(↑)	Recall(↑)	F1-Score(↑)	AUC-ROC(↑)
Virus	MVote	-	-	0.9232	0.9171	0.9330	0.9250	0.9805
	SVote	-	-	0.9345	0.9249	0.9479	0.9363	0.9809
	PWeight	-	-	0.9332	0.9268	0.9429	0.9348	0.9806
	SL(Ankh)	-	-	0.9131	**0.9380**	0.8957	0.9164	0.9582
	UVote	SVDKL	NS(c=5)	0.9320	0.9246	0.9429	0.9337	0.9650
	DUNE*	DROPOUT	NS(c=5)	**0.9395**	0.9298	**0.9529**	**0.9412**	**0.9810**
Bacteria	MVote	-	-	0.8286	0.7987	0.6839	0.7368	0.8794
	SVote	-	-	0.8327	0.8054	0.6897	0.7430	0.8883
	PWeight	-	-	**0.8488**	0.8153	0.7356	**0.7734**	0.8851
	SL(Ankh)	-	-	0.8306	0.6954	**0.7961**	0.7423	0.8616
	UVote	DROPOUT	KLD	0.8387	0.8052	0.7126	0.7561	0.8883
	DUNE*	DROPOUT	Unbiased	**0.8448**	**0.8170**	0.7184	0.7645	**0.8892**
Tumor	MVote	-	-	0.7436	0.6329	**0.8197**	0.7143	0.8336
	SVote	-	-	0.7500	0.6486	0.7869	0.7111	0.8483
	PWeight	-	-	0.7628	0.6765	0.7541	0.7132	0.8383
	SL(Ankh)	-	-	0.7692	**0.8852**	0.6506	**0.7500**	**0.8607**
	UVote	DVBLL	NS(c=5)	0.7756	0.6806	0.8033	0.7368	0.8214
	DUNE*	DVBLL	NS(c=25)	**0.7885**	0.7000	0.8033	0.7481	0.8373
Toxicity(Test)	MVote	-	-	0.9845	0.8534	**0.9000**	0.8761	0.9896
	SVote	-	-	0.9845	0.8596	0.8909	0.8750	0.9901
	PWeight	-	-	**0.9856**	**0.8684**	**0.9000**	**0.8839**	0.9896
	SL(Ankh)	-	-	0.9801	0.8273	0.8426	0.8349	0.9858
	UVote	DROPOUT	KLD	0.9845	0.8534	**0.9000**	0.8761	**0.9933**
	DUNE*	DROPOUT	KLD	0.9850	0.8673	0.8909	0.8789	0.9904
Toxicity(Independent)	MVote	-	-	0.9611	0.4378	0.7171	0.5436	0.9619
	SVote	-	-	0.9636	0.4603	**0.7237**	0.5627	**0.9637**
	PWeight	-	-	0.9664	0.4865	0.7105	0.5775	0.9630
	SL(ESM2)	-	-	0.9600	**0.7105**	0.4286	0.5347	0.9601
	UVote	DVBLL	KLD	0.9674	0.4974	0.6184	0.5513	0.9619
	DUNE*	DVBLL	KLD	**0.9698**	0.5269	0.6447	**0.5799**	0.9619
Allergenicity	MVote	-	-	0.7836	0.0353	**1.0000**	0.0682	0.9912
	SVote	-	-	0.9298	0.1014	**1.0000**	0.1841	0.9966
	PWeight	-	-	0.9204	0.0871	0.9545	0.1597	0.9951
	SL(ESMC)	-	-	0.8351	**1.0000**	0.0458	0.0876	0.9890
	UVote	DVBLL	NS(c=5)	0.6982	0.0256	**1.0000**	0.0499	0.9126
	DUNE*	DVBLL	NS(c=5)	**0.9802**	0.2857	**1.0000**	**0.4444**	**0.9997**

DUNE lagged behind other methods in terms of precision and recall, those methods lagged in other metrics for all cases. For allergenicity dataset, DUNE outperformed other baselines by a large margin across all metrics, except precision.

G.4.2 Comparative Uncertainty Weighting Approaches

Table 4 shows the comparative results on different weighting approaches. We used Dropout approach to convert the ensemble members into probabilistic models. KL-Divergence and Negative Softmax weighting with $c = 5$ performed in a similar manner and better than their counterparts in majority cases. For the Negative Softmax-based weighting, smaller values of the control parameter yielded better performance than larger values in majority cases. This aligns with expectations, as higher control parameter values bias the model towards UVote approach. The unbiased weighting provided better results in terms of 4 out the 6 metrics only at Immuno-Bacteria dataset.

G.4.3 Comparative Probabilistic Model Architectures for Members in Deep Ensemble

Apart from Dropout, we also tested other probabilistic deep model architectures. Table 5 shows the results for different probabilistic models.

In terms of predictive UQ performance, different BNNs exhibited varying strengths across different datasets. Overall, DROPOUT and DVBLL performed better than their counterparts at majority of metrics across all datasets. DROPOUT performed in a better manner across Immuno-Virus and test toxicity dataset, and DVBLL performed in a better manner in rest of the datasets.

G.4.4 Uncertainty Quantification Evaluation of Probabilistic Model Architectures for Members in Deep Ensemble

We evaluated different probabilistic deep model architectures for uncertainty quantification. Table 6 shows the results for different probabilistic models on several uncertainty quantification metrics, along with our proposed Cross-Divergence metric.

Table 4: Comparative results on different weighting approaches. KLD denotes KL-Divergence weighting and NS denotes Negative-Softmax weighting.

Dataset	Weighting	Accuracy(↑)	Precision(↑)	Recall(↑)	F1 Score(↑)	AUC ROC(↑)	NLL(↓)
Virus	KLD	0.9358	0.9231	**0.9529**	0.9377	0.9804	**0.1788**
	NS(c=5)	**0.9395**	**0.9298**	**0.9529**	**0.9412**	0.9810	0.1836
	NS(c=25)	0.9383	0.9296	0.9504	0.9399	**0.9812**	0.1794
	NS(c=100)	0.9370	0.9253	**0.9529**	0.9389	0.9807	0.1803
	Unbiased	0.9282	0.9159	0.9454	0.9304	0.9711	0.4163
Bacteria	KLD	0.8367	0.8039	0.7069	0.7523	**0.8896**	0.5159
	NS(c=5)	0.8327	0.8054	0.6897	0.7430	0.8883	**0.4800**
	NS(c=25)	0.8327	0.8054	0.6897	0.7430	0.8881	0.4880
	NS(c=100)	0.8407	0.8105	0.7126	0.7584	0.8881	0.5245
	Unbiased	**0.8448**	**0.8170**	**0.7184**	**0.7645**	0.8892	0.8412
Tumor	KLD	**0.7564**	**0.6716**	0.7377	0.7031	0.8378	0.5916
	NS(c=5)	0.7436	0.6400	**0.7869**	**0.7059**	**0.8475**	**0.4718**
	NS(c=25)	0.7308	0.6338	0.7377	0.6818	**0.8475**	0.4822
	NS(c=100)	0.7436	0.6522	0.7377	0.6923	0.8385	0.5734
	Unbiased	0.7500	0.6618	0.7377	0.6977	0.8264	1.7150
Toxicity(test)	KLD	**0.9850**	**0.8673**	0.8909	**0.8789**	0.9904	**0.0515**
	NS(c=5)	0.9845	0.8596	0.8909	0.8750	0.9902	0.0534
	NS(c=25)	0.9845	0.8596	0.8909	0.8750	0.9903	0.0518
	NS(c=100)	0.9839	0.8462	**0.9000**	0.8722	0.9908	0.0520
	Unbiased	0.9839	0.8462	**0.9000**	0.8722	**0.9931**	0.1631
Toxicity(independent)	KLD	**0.9645**	**0.4681**	**0.7237**	**0.5685**	0.9648	0.1003
	NS(c=5)	0.9634	0.4580	0.7171	0.5590	0.9634	0.0979
	NS(c=25)	0.9638	0.4619	0.7171	0.5619	0.9631	**0.0969**
	NS(c=100)	0.9636	0.4599	0.7171	0.5604	0.9646	0.1041
	Unbiased	0.9625	0.4496	0.7039	0.5487	**0.9683**	0.2620
Allergenicity	KLD	0.8740	0.0591	1.0000	0.1117	0.9985	0.8755
	NS(c=5)	**0.9240**	**0.0944**	1.0000	**0.1725**	0.9967	**0.7113**
	NS(c=25)	0.8945	0.0698	1.0000	0.1306	0.9971	0.7809
	NS(c=100)	0.8419	0.0477	1.0000	0.0911	**0.9988**	1.0359
	Unbiased	0.6572	0.0226	1.0000	0.0442	0.9132	5.9252

Table 5: Comparative results for different probabilistic ensemble member model.

Prediction Metric	Model	Virus	Bacteria	Tumor	Toxicity(Test)	Toxicity(Independent)	Allergenicity
Accuracy(↑)	DROPOUT	**0.9358**	0.8367	0.7564	**0.9850**	0.9645	0.8740
	SVDKL	0.9332	**0.8387**	0.6795	0.9845	0.9640	0.8905
	DVBLL	0.9332	0.8327	**0.7756**	0.9845	**0.9698**	**0.9503**
	LA	0.9232	0.8286	0.7500	0.9845	0.9668	0.9399
	SWAG	0.9345	0.8347	0.7500	0.9817	0.9608	0.9096
Precision(↑)	DROPOUT	0.9231	0.8039	0.6716	**0.8673**	0.4681	0.0591
	SVDKL	0.9227	**0.8092**	0.5965	0.8661	0.4612	0.0675
	DVBLL	**0.9375**	0.7725	**0.6857**	0.8596	**0.5269**	**0.1375**
	LA	0.9233	0.7572	0.6447	0.8661	0.4896	0.1164
	SWAG	0.9249	0.7738	0.6667	0.8235	0.4380	0.0806
Recall(↑)	DROPOUT	**0.9529**	0.7069	0.7377	**0.8909**	0.7237	1.0000
	SVDKL	0.9479	0.7069	0.5574	0.8818	0.6645	1.0000
	DVBLL	0.9305	0.7414	0.7869	**0.8909**	0.6447	1.0000
	LA	0.9256	**0.7529**	**0.8033**	0.8818	0.6184	1.0000
	SWAG	0.9479	0.7471	0.7213	**0.8909**	**0.7434**	1.0000
F1 Score(↑)	DROPOUT	**0.9377**	0.7523	0.7031	**0.8789**	0.5685	0.1117
	SVDKL	0.9351	0.7546	0.5763	0.8739	0.5445	0.1264
	DVBLL	0.9340	0.7566	**0.7328**	0.8750	**0.5799**	**0.2418**
	LA	0.9244	0.7550	0.7153	0.8739	0.5465	0.2085
	SWAG	0.9363	**0.7602**	0.6929	0.8559	0.5512	0.1492
AUC-ROC(↑)	DROPOUT	**0.9804**	0.8896	0.8378	0.9904	**0.9648**	0.9985
	SVDKL	0.9771	0.8750	0.7220	0.9899	0.9506	0.9989
	DVBLL	0.9762	**0.8916**	**0.8502**	0.9907	0.9619	**0.9996**
	LA	0.9749	0.8858	0.8380	0.9904	0.9591	0.9994
	SWAG	0.9792	0.8796	0.8373	**0.9932**	0.9633	0.9979

Table 6: Uncertainty Quantification Evaluation

UQ Metric	Model	Virus	Bacteria	Tumor	Toxicity(test)	Toxicity(independent)	Allergenicity
CDiv(↑)	DROPOUT	**6.6719**	0.5928	-3.6989	12.4150	9.8370	-62.6277
	SVDKL	3.5264	0.2320	0.0006	6.6525	6.1032	-10.3151
	DVBLL	5.6940	-5.5328	-0.6190	11.5257	10.3730	-29.3711
	LA	6.1067	0.4271	-1.8109	**13.4206**	**10.8600**	-46.1843
	SWAG	4.7570	**2.2087**	**1.6627**	8.5277	7.4408	**-3.4477**
ECE(↓)	DROPOUT	**0.0060**	0.0964	0.1242	0.0057	0.0285	0.3302
	SVDKL	0.0470	0.1716	0.1782	0.0362	0.0540	0.3347
	DVBLL	0.0153	0.1118	0.1060	**0.0030**	**0.0185**	**0.2556**
	LA	0.0154	0.0940	0.1270	0.0031	0.0187	0.2878
	SWAG	0.0370	**0.0476**	**0.0742**	0.0182	0.0508	0.3662
NLL(↓)	DROPOUT	**0.1788**	0.5159	0.5916	**0.0515**	0.1003	0.8755
	SVDKL	0.2140	0.5049	0.6917	0.0784	0.1160	0.7475
	DVBLL	0.2012	0.5624	0.5393	0.0534	**0.0860**	**0.5063**
	LA	0.2014	0.4784	0.5708	0.0551	0.0961	0.5701
	SWAG	0.1958	**0.4200**	**0.4960**	0.0600	0.1099	0.7281
Brier Score(↓)	DROPOUT	**0.0496**	0.1345	0.1905	**0.0130**	0.0284	0.2414
	SVDKL	0.0544	0.1597	0.2493	0.0141	0.0270	0.2321
	DVBLL	0.0541	0.1359	0.1747	0.0141	**0.0236**	**0.1602**
	LA	0.0576	0.1353	0.1849	0.0142	0.0258	0.1890
	SWAG	0.0531	**0.1294**	**0.1655**	0.0147	0.0305	0.2524

These results demonstrate that our proposed Cross-Divergence (CDiv) uncertainty quantification (UQ) metric generally aligns with other established UQ metrics, specially in Immuno-Virus, Bacteria and Tumor datasets, where DROPOUT performed best in Immuno-Virus datasets at all metrics and SWAG performed best in Immuno-Bacteria and Tumor datasets.

In test toxicity dataset, no one model provided optimum values across all four metrics. But in general, DROPOUT, DVBLL and LA achieved low ECE, NLL, Brier Score and high CDiv. In independent toxicity dataset, DVBLL yielded optimal results for Expected Calibration Error (ECE), Negative Log-Likelihood (NLL), and Brier Score, while LA performed best according to CDiv. However, in general all four metrics behaved quite coherently for both DVBLL and LA.

The DVBLL model performed best at allergenicity dataset in terms of ECE, NLL and Brier Score. However, SWAG model performed best in terms of CVdiv. Also, all models showed high uncertainty at allergenicity dataset compared to their UQ perfromances across other datasets.

G.5 Discussion

Key observations from the experimental results are summarized below:

- **DUNE Outperforms Baselines**: Table 3 demonstrates that DUNE outperforms all baselines in most of the datasets across majority of the metrics, especially performs much better than most of the baselines in the allergenicity dataset where most of the baselines perform poorly.
- **KL-Divergence and Negative Softmax with Low c are Better Weighting Choices**: Table 4 demonstrates KL-Divergence and Negative Softmax with smaller c performed better than their counterparts. Superior performance of Negative Softmax with low c indicates that having too much belief on one single model results in suboptimal performance.
- **DROPOUT & DVBLL Performs Better Predictive Performance**: Table 5 shows the predictive performances of different BNNs. Overall, DROPOUT and DVBLL performed better in terms of predictive performance.
- **Cross-Divergence's Coherent Behavior with Existing UQ Metrics**: Table 6 confirms that our proposed metric, Cross-Divergence, behaves consistently with established UQ metrics, indicating its capability for robust UQ performance evaluation.
- **High Uncertainty in Allergenicity Dataset**: Table 6 demonstrates that all models show high uncertainty in allergenicity dataset according to all UQ metrics.
- **Low Precision or Low Recall in Allergenicity Dataset**: All experimental results show that almost all models show quite low precision or recall value for the allergenicity dataset. This indicates that due to most models' bias toward the positive class and the highly imbalanced nature of the test allergenicity dataset, all methods produce low precision, except the single learner. The single learner however performed poorly in terms of all other metrics.

H Related works

Protein Property Prediction: The growing fields of protein engineering and therapeutic development demand precise and efficient ways to characterize crucial protein properties, especially those affecting safety and efficacy. The ability of a protein to elicit an undesired immune response (immunogenicity) is a critical concern in the development of biopharmaceuticals, vaccines, and gene therapies. To address this, VenusVaccine offers a deep learning solution that uses a dual attention mechanism to combine pretrained latent vector representations of protein sequences and structures [33].

Equally important is predicting protein toxicity, particularly for proteins used in therapeutic or industrial applications. Proteins can be toxic through various means, including direct cell damage, disrupting physiological processes, or accumulating to harmful levels. ToxDL 2.0 tackles this with a new multimodal deep learning model that integrates evolutionary and structural information from a pre-trained language model and AlphaFold2 [60].

Furthermore, with more new proteins appearing in food, pharmaceuticals, and industrial products, assessing their potential to cause allergies is vital for public health. AllergenAI provides a new AI-based tool to quantify this allergenic potential based solely on protein sequences, setting it apart from previous tools that also used physicochemical properties and sequence homology [58].

While machine learning has revolutionized safety and efficacy related protein property prediction, there's a notable gap in research concerning the application of uncertainty quantification (UQ) in this field. Integrating uncertainty quantification into existing data-driven protein property prediction methods requires new research to boost their predictive performance.

Deep Ensemble Meets UQ: To improve classification performance with deep learning models, there's been a trend toward using ensemble methods, which allow individual members to specialize in predictions for sparser data regions. UVOTE is a recently proposed ensemble approach designed to tackle imbalanced regression problems [27]. It integrates recent advancements in probabilistic deep learning, and its core mechanism involves deriving the final prediction from the least uncertain member of the deep learning model ensemble.

The integration of UQ into deep ensemble methods for safety and efficacy related protein property prediction, however, remains largely unexplored, highlighting a significant need for novel research endeavors.

I Uncertainty Evaluation Metrics

The three established UQ metrics we utilized in this work are Expected Calibration Error (ECE), Negative Log-Likelihood (NLL), and the Brier Score, which have been commonly employed in the literature. ECE and Brier scores are considered to assess a model's calibration, indicating how well its predicted probabilities align with the true likelihood of events, while NLL is mostly regarded as an indicator of overconfidence, revealing when a model is overly certain about its predictions, even if they are incorrect.

I.1 Expected Calibration Error (ECE)

ECE partitions predictions into M equally-spaced bins based on their prediction confidence, ECE can be calculated as,

$$ECE = \sum_{m=1}^{M} \frac{B_m}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \tag{20}$$

with N indicating the size of the dataset, and $\text{acc}(B_m) = 1/|B_m| \sum_{i \in B_m} \mathbb{I}(\tilde{y}_i = y_i)$ and $\text{conf}(B_m) = 1/|B_m| \sum_{i \in B_m} \hat{y}_i$ the average accuracy and confidence in bin B_m with size $|B_m|$ accordingly. Here, $\tilde{y}$ is the predicted label and $\hat{y}$ is the predicted class probability. For, binary classification problem, $\tilde{y} = \mathbb{I}(\hat{y} \geq \theta_d)$, with θ_d as the decision threshold, usually 0.5 for binary classification problem.

I.2 Brier Score

For a binary classification, this metric is evaluated as:

$$Brier = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{21}$$

Here y denotes the true label and $\hat{y}$ denotes the predicted positive class probability.

I.3 Negative Log-Likelihood

Negative log-likelihood is computed as the negative log-probability assigned to the true label,

$$NLL(x) = -y(x)\log(\hat{y}(x)) - (1 - y(x))\log(1 - \hat{y}(x)) \tag{22}$$

When a model is overconfident in an incorrect prediction, it assigns a high probability to the wrong class. Therefore, the log loss becomes very large, that results in a high NLL.

J Ablation Studies with Protein Structural Features

Table 7 shows the comparative results on ablation studies with protein structural features. The reported results were obtained using Dropout approach to convert the ensemble members into probabilistic models and following KL-Divergence based weighting approach.

Table 7: Reuslts on ablation studies with protein structural features.

Dataset	ESM3	FoldSeek	Accuracy(↑)	Precision(↑)	Recall(↑)	F1 Score(↑)	AUC ROC(↑)	NLL(↓)
Virus	✓	✓	**0.9358**	0.9231	**0.9529**	**0.9377**	**0.9804**	**0.1788**
	×	✓	0.9282	**0.9282**	0.9305	0.9294	0.9767	0.1965
	✓	×	0.9257	0.9257	0.9280	0.9269	0.9732	0.2099
	×	×	0.9207	0.9126	0.9330	0.9227	0.9753	0.2073
Bacteria	✓	✓	0.8367	**0.8039**	0.7069	0.7523	0.8896	0.5159
	×	✓	0.8347	0.7805	0.7356	0.7574	0.8906	0.4606
	✓	×	0.8306	0.7679	**0.7414**	0.7544	**0.8923**	**0.4527**
	×	×	**0.8407**	0.7950	0.7356	**0.7642**	0.8837	0.4777
Tumor	✓	✓	0.7564	0.6716	**0.7377**	0.7031	0.8378	0.5916
	×	✓	0.7564	0.6769	0.7213	0.6984	0.8485	0.4733
	✓	×	**0.7821**	**0.7143**	**0.7377**	**0.7258**	**0.8595**	**0.4568**
	×	×	0.7051	0.6087	0.6885	0.6462	0.8142	0.5154
Toxicity(test)	✓	✓	**0.9850**	0.8673	**0.8909**	**0.8789**	0.9904	**0.0515**
	×	✓	0.9823	**0.8824**	0.8182	0.8491	**0.9926**	0.0526
	✓	×	0.9839	0.8716	0.8636	0.8676	0.9908	0.0552
	×	×	0.9806	0.8319	0.8545	0.8430	0.9899	0.0576
Toxicity(independent)	✓	✓	0.9645	0.4681	**0.7237**	0.5685	0.9648	0.1003
	×	✓	**0.9694**	**0.5217**	0.6316	0.5714	**0.9653**	**0.0900**
	✓	×	0.9651	0.4722	0.6711	0.5543	0.9593	0.0981
	×	×	0.9651	0.4741	**0.7237**	**0.5729**	0.9648	0.0935
Allergenicity	✓	✓	0.8740	0.0591	1.0000	0.1117	0.9985	0.8755
	×	✓	0.8653	0.0556	1.0000	0.1053	0.9913	0.8608
	✓	×	0.8931	0.0690	1.0000	0.1290	0.9999	0.8419
	×	×	**0.9525**	**0.1429**	1.0000	**0.2500**	**1.0000**	**0.5032**

These results present intriguing observations regarding the utility of protein structural information in property prediction. Contrary to common belief, incorporating protein structural information did not consistently lead to improved predictive performance. However, drawing definitive conclusions about the universal utility of protein structural information for property prediction is complex. It's important to note that our experiments exclusively utilized ESMFold for structure prediction, a choice that prioritizes computational speed over accuracy compared to AlphaFold. The backbone VenusVaccine architecture employed in this study integrates sequence and structural features via an attention mechanism; more sophisticated integration approaches might yield enhanced predictive performance. Additionally, other factors may contribute to sub-optimal performance when using predicted protein structural information.