ESMDisPred: A Structure-Aware CNN–Transformer Architecture for Intrinsically Disordered Protein Prediction

Md Wasi Ul Kabir, Ayon Dey, Farzeen Nafees, Md Tamjidul Hoque*

Email: {mkabir3, adey, fnafees, thoque}@uno.edu

Department of Computer Science, University of New Orleans, New Orleans, LA, USA.

*Corresponding Author: thoque@uno.edu

Abstract

Intrinsically disordered proteins (IDPs) lack stable three-dimensional structures, yet play vital roles in key biological processes, including signaling, transcription regulation, and molecular scaffolding. Their structural flexibility presents significant challenges for experimental characterization and contributes to diseases such as cancer and neurodegenerative disorders. Accurate computational prediction of IDPs and IDRs is important for research and progress in drug discovery, structural biology, and protein engineering. In this study, we introduce ESMDisPred, a novel structure-aware disorder predictor that leverages the representational power of Evolutionary Scale Modeling-2 (ESM2) protein language models. ESMDisPred integrates sequence embeddings with structural information to deliver state-of-the-art prediction accuracy. Model performance is further enhanced through feature engineering strategies, including terminal residue encoding, statistical summarization, and sliding-window analysis. To capture both local sequence motifs and long-range dependencies, we designed a hybrid CNN–Transformer architecture that balances convolutional efficiency with the representational power of self-attention. On CAID3 benchmarks, our latest model achieves ROC-AUC 0.895, APS 0.778, and F1 max 0.759, outperforming recent methods under the official evaluation protocol. Our results highlight the importance of incorporating language model-derived structural awareness in protein disorder prediction.

Keywords: Intrinsically Disordered Proteins, Protein Language Models, Disorder Prediction, Machine Learning, Deep Learning, Bioinformatics.

1.1 Introduction

Intrinsically disordered regions (IDRs) play essential roles in various biological processes but remain difficult to characterize experimentally due to their lack of stable tertiary structure. This has led to a growing reliance on computational methods for IDR prediction [1, 2]. Recent advances in deep learning and protein language models (PLMs) [3, 4] have significantly improved the accuracy and scalability of such predictions. These models learn rich representations from protein sequences, capturing both evolutionary and structural information without the need for handcrafted features. Among the current state-of-the-art methods, DisPredict3.0 has demonstrated strong performance across multiple benchmarks. However, assessments from the CAID (Critical Assessment of protein Intrinsic Disorder) community challenge [5] reveal limitations in predictive performance, particularly in capturing subtle or transient disorder characteristics. This motivates the development of a more advanced and structure-aware prediction framework.

Several recent methods reflect similar efforts to push the boundaries of IDR prediction. For example, rawMSA-disorder, introduced by Mirabello and Wallner, utilizes raw multiple sequence alignments as input without relying on pre-computed features like PSSMs [6]. This architecture allows the network to learn disorder-relevant evolutionary patterns directly from the alignment data. DisoFLAG-IDR, developed by Pang and Liu, adopts a multi-task learning framework based on PLM embeddings and graph convolutional networks (GCNs). This model not only predicts disordered regions but also infers six functional classes of IDRs, including those involved in protein, DNA, RNA, lipid, and ion binding, as well as linker segments. The flDPnn3a model, part of the flDPnn series by the Kurgan Lab, leverages a combination of sequence-derived features and global protein attributes. It employs an ensemble learning strategy to predict both general disorder and four functional IDR types, including MoRFs and linkers [7].

Beyond the models discussed above, several complementary approaches shape today's IDR-prediction landscape. SPOT-Disorder2 and the flDPnn family remain strong deep-learning baselines; their profile/ensemble designs still compete on CAID despite the shift to single-sequence PLMs [7, 8]. Lightweight single-sequence tools—SETH, LMDisorder, and Metapredict-v3—enable fast proteome-scale screening with accuracy that rivals heavier profile-based pipelines [9]. Classical physics-inspired and consensus methods—IUPred3/AIUPred and MobiDB-lite—

remain popular for interpretability and long-IDR specificity [10-12]. Structure-proxy baselines from AlphaFold/AlphaFold3 (e.g., 1−pLDDT or RSA) offer quick heuristics that place mid-table on CAID, but they generally trail purpose-trained PLM models [13]. New entrants such as PUNCH2 and task-focused binders (e.g., IPA variants, DeepDISObind) show gains for specialized aims like disordered binding-site localization [14-16]. Overall, embedding-first single-sequence predictors now set the pace, while profile-based ensembles, physics-inspired estimators, and structure-derived proxies remain valuable for cross-validation, interpretability, or speed, depending on the use case.

Recent trends also favor lightweight models that maintain high accuracy while offering faster runtime. DisorderUnetLM, introduced by Kotowski et al., uses a U-Net convolutional architecture combined with embeddings from ProtT5. It demonstrates that single-sequence models can match or exceed profile-based methods in some contexts [17]. Another innovative approach is UdonPred-combined, which uniquely integrates continuous disorder signals from NMR-derived TriZOD data with binary disorder labels from DisProt. This combination allows the model to provide not only binary predictions but also nuanced, real-valued disorder propensities [18]. Finally, EBIND-protein focuses on predicting disordered binding regions, specifically in MoRFs that mediate protein-protein interactions. By leveraging PLM embeddings, EBIND-protein identifies binding-prone segments within IDRs [19].

In our earlier work, we introduced DisPredict3.0, the most recent iteration of the DisPredict series, which integrates evolutionary representations derived from protein language models to enhance the prediction of intrinsically disordered regions (IDRs) [20]. This approach achieved substantial performance gains over CAID-2018 methods, and secured the top ranking in CAID2 on the Disorder NOX dataset. Building on this foundation, we now present ESMDisPred, a structure-aware disordered protein predictor that leverages embeddings from the Evolutionary Scale Modeling-2 (ESM2) language model [3]. ESM2 has demonstrated remarkable success in protein structure prediction, and its structural awareness offers a powerful basis for modeling disordered regions with greater biological relevance. To further improve predictive capacity, we integrate targeted feature-engineering strategies—including terminal residue encoding, statistical summarization, and sliding-window analysis. To capture both local sequence motifs and long-

range dependencies, ESMDisPred employs a hybrid CNN–Transformer architecture, combining the efficiency of convolutional filters with the expressive power of self-attention mechanisms.

Comprehensive benchmarking on CAID3 datasets [21] demonstrates that ESMDisPred consistently outperforms existing state-of-the-art methods across multiple evaluation metrics, including AUC, F1 max, and average precision. These results underscore the value of incorporating language model-derived structural awareness into disorder prediction and establish ESMDisPred as a robust and scalable framework for proteome-wide analysis of intrinsically disordered proteins and their functional roles in disease.

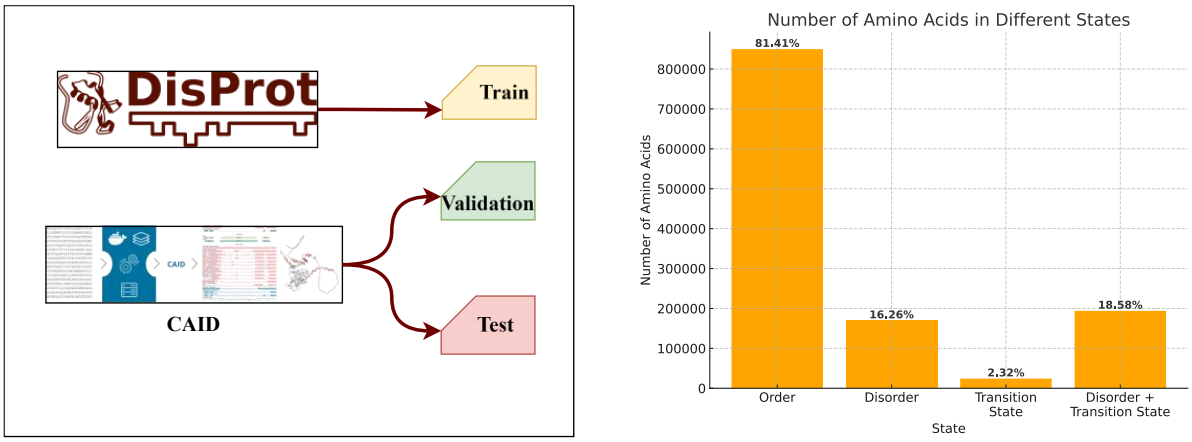
1.2 Materials and Methods

1.2.1 Dataset

We carefully curated the dataset from the DisProt database (prior to the 2023_12 release) [1, 2] to ensure no overlap between the training and testing sets, maintaining the integrity of the evaluation. The initial dataset was composed of 2,845 proteins from the latest DisProt release. To improve the quality of the dataset, we removed long protein sequences that could introduce noise or bias into the analysis. In order to further refine the dataset and avoid redundancy, we applied a 25% sequence identity cutoff, ensuring that similar protein sequences were excluded [22]. This process resulted in a final training set consisting of 2,020 proteins, which together account for a total of 1,043,829 amino acids. This rigorous filtering ensures that the training data is diverse and representative, while minimizing potential biases from highly similar sequences.

Figure 1 presents the distribution of amino acids in the training dataset across four structural states: Order, Disorder, Transition State, and their combination (Disorder + Transition State). The majority of residues, approximately 81.41%, are classified as ordered, while disordered and transition residues make up 16.26% and 2.32%, respectively. For model training and evaluation, we consider the annotation of disordered and transition states (totaling 18.58% of the dataset) as the positive class for prediction. This grouping acknowledges the functional and structural continuum between fully disordered and transitional regions. The imbalance between ordered and

non-ordered residues shows the importance of handling class imbalance effectively during model development.



(a) (b)

Figure 1. Overview of dataset partitioning and composition used for ESMDisPred model training and evaluation. (a) Illustration of dataset splitting into training, validation, and test subsets derived from the DisProt database and the CAID benchmark dataset. (b) Distribution of amino acids across different structural states (ordered, disordered, transition state) within the training dataset, highlighting the prevalence of ordered residues.

For validation, we utilized two subsets from the CAID2 dataset: Disorder NOX (210 sequences) and Disorder PDB (348 sequences). These subsets enable targeted evaluation of model performance across different aspects of protein disorder, including functional binding regions and flexible linkers. The final model was tested using the CAID3 benchmark test datasets [23].

1.2.2 Exploratory Data Analysis

Exploratory Data Analysis is an important step for designing Machine Learning models. In this section, we explore the dataset to find relevant information that would help us increase the model's accuracy. We begin by analyzing the distribution of protein sequences by length. Figure 2 illustrates the distribution of protein sequences based on their length. Each bar in the figure represents the number of protein sequences corresponding to a particular sequence length, providing an overview of the length variation within the dataset. The analysis reveals that most protein lengths fall below 2000 amino acids. To reduce class imbalance, we exclude these longer sequences from the training dataset.

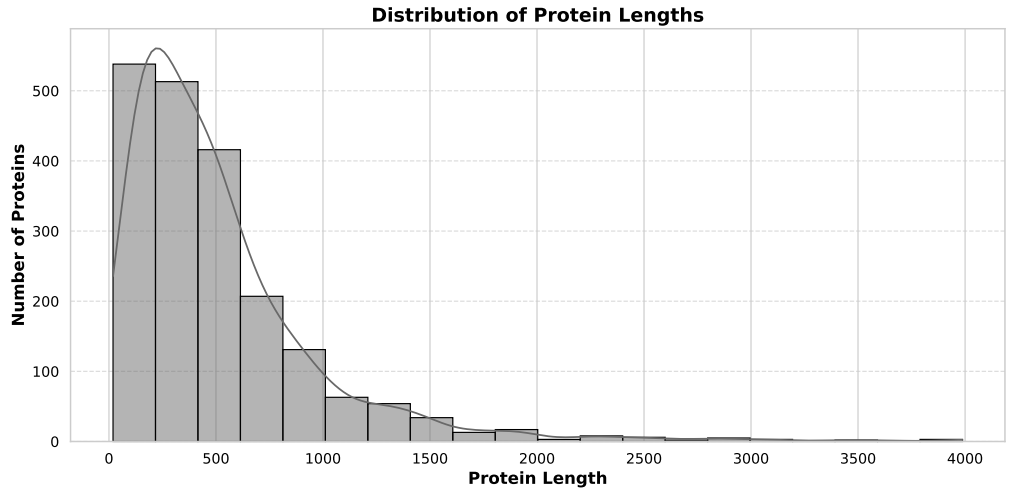


Figure 2. Distribution of protein sequences by length. The x-axis represents the length of the protein sequences (number of amino acids), and the y-axis shows the number of proteins corresponding to each sequence length. The majority of protein sequences in the dataset fall between 0 and 500 amino acids, with a sharp decline in the number of sequences as the length increases beyond 500 amino acids. A few sequences are longer, but they are much less frequent, extending up to 4000 amino acids.

We also explore the distribution of disordered regions in proteins from the training set (Figure 3). Most disordered residues are concentrated in the N- and C-terminal regions. This pattern suggests that intrinsic disorder is not randomly distributed but follows a trend. Such positional bias reflects underlying structural or functional constraints. Incorporating this context into predictive models by highlighting terminal regions can help improve disorder prediction. We discuss the effect of annotating terminal regions in feature selection and engineering section.

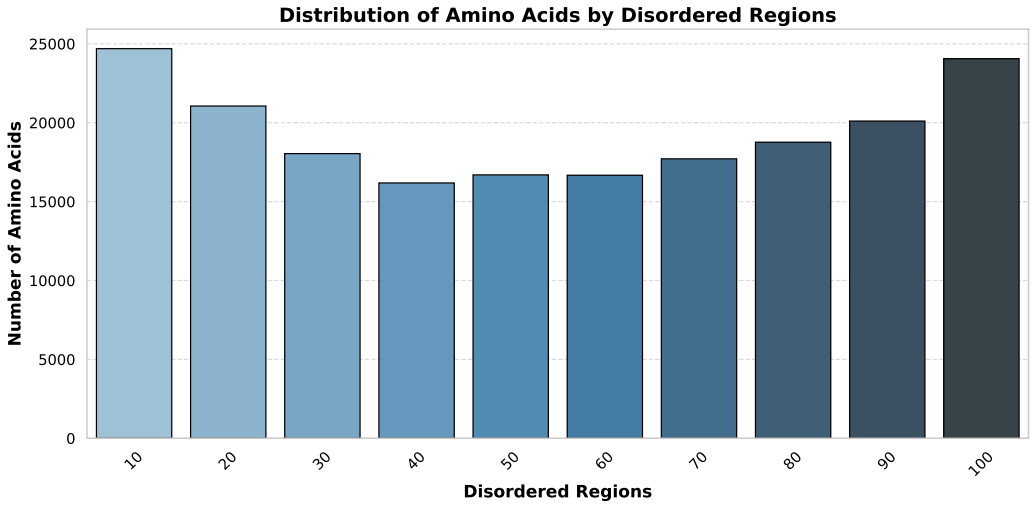


Figure 3. The distribution of disordered amino acids across protein sequence positions. The x-axis represents relative positions along the protein sequence, divided into 10 equal parts (from 10% to 100%). The y-axis indicates the total number of disordered amino acids observed in each segment across all proteins. The plot reveals that disordered amino acids are most frequently found at the N-terminal (10%) and C-terminal (100%) ends of proteins, while middle regions (30%–70%) contain fewer disordered residues. This suggests that intrinsic disorder is more prevalent at the sequence termini.

Figure 4 shows the distribution of proteins by the percentage of disordered regions. As expected, the majority of proteins contain a low proportion of disordered residues, indicating that intrinsic disorder is often confined to specific regions rather than spread throughout the protein. This supports the notion that disorder commonly plays a localized regulatory or interaction role within otherwise structured proteins. However, the distribution also reveals a distinct subset of proteins with a high percentage of disordered regions, including proteins that are almost entirely disordered. These fully disordered proteins likely perform specialized functions, such as signaling, molecular recognition, or forming dynamic complexes, where structural flexibility is essential. The presence of both partially and fully disordered proteins reflects the functional diversity of disorder in the proteome.

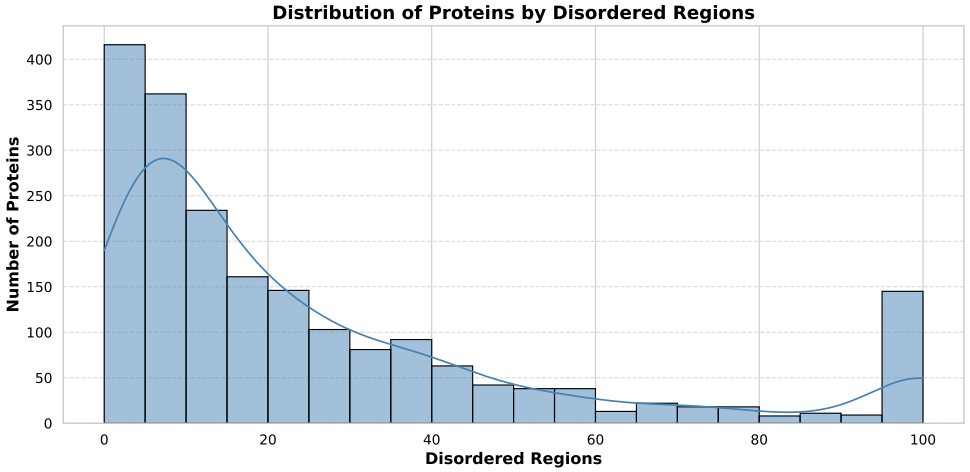


Figure 4. The distribution of proteins based on the percentage of disordered regions. The x-axis represents the percentage of disordered regions in proteins, while the y-axis shows the number of proteins in each percentage range. The majority of proteins have a low percentage of disordered regions (0-20%), with a sharp decline as the percentage of disorder increases. However, there is a noticeable peak at 100%, indicating that a substantial number of proteins are fully disordered. This distribution reflects that while most proteins have minimal disorder, fully disordered proteins also occur with some frequency in the dataset.

The insights obtained from the exploratory analysis guide the next stages of model development. Specifically, identifying the sequence length distribution helps us address class imbalance by filtering out extreme outliers, while the observed enrichment of disorder at the N- and C-termini highlights the importance of positional information. Based on these findings, we performed targeted feature selection and engineering to capture sequence length variability, terminal region characteristics, and disorder proportions. We discuss these steps in detail in the Feature Selection and Engineering section.

1.2.3 Machine Learning methods

We applied a broad spectrum of machine learning (ML) methods to model tabular data, spanning classical statistical approaches, ensemble tree algorithms, and deep learning architectures optimized for structured inputs. Each method contributed distinct strengths to our modeling pipeline. We began with linear discriminant analysis (LDA), which identifies linear feature combinations that maximize class separation by optimizing the ratio of inter-class to intra-class variance [24]. In parallel, we used a ridge classifier, a regularized linear model based on ridge regression. It applies an L2 penalty to reduce coefficient variance and improve generalization, particularly in high-dimensional spaces.

For instance-based learning, we used KNeighborsClassifier (KNN) [25]. This method classifies a sample based on the majority class of its nearest neighbors in feature space. Though simple, KNN offers strong performance in tasks prioritizing interpretability and low training cost. We also implemented several ensemble tree-based methods. RandomForestClassifier [26] constructs multiple decision trees using bootstrapped samples and randomly selected feature subsets. This bagging approach enhances generalization and reduces overfitting. ExtraTreesClassifier [27] builds an ensemble of completely randomized trees, introducing further randomness in the choice of split thresholds. This increases robustness to noise and accelerates training.

We evaluated boosting-based ensemble methods using four algorithms. AdaBoostClassifier [28], which combines weak learners in sequence, giving more weight to misclassified instances. XGBoost [29] improves classical gradient boosting through second-order optimization, regularization, and fast tree construction. LightGBM [30] incorporates innovations like Gradient-

based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to improve efficiency on large datasets. CatBoost [31] addresses categorical feature handling using ordered boosting and target statistics, reducing overfitting and bias.

In exploring deep learning techniques, we began with the CategoryEmbeddingModel [32], a multilayer perceptron that uses learnable embeddings for categorical features. This approach replaces one-hot encodings with dense, continuous vectors that the model updates during training. We evaluated TabTransformer [33], which applies transformer layers to contextualize categorical embeddings before combining them with numerical features in a fully connected network. We also used TabNet [34], which introduces a sequential attention mechanism for selecting relevant features at each decision step. TabNet's feature sparsity and interpretability make it effective for tabular tasks. DANet [35] incorporates an Abstract Layer to construct higher-order feature interactions, aided by shortcut paths from input to deeper layers to preserve original feature representations.

We also evaluated GANDALF [36], a deep learning model based on Gated Recurrent Units (GRUs), adapted for tabular data. It introduces the Gated Feature Learning Unit (GFLU), which performs implicit feature selection through gating mechanisms, enabling the model to prioritize informative variables during training and inference.

In addition, we designed a CNN–Transformer hybrid architecture for per-residue disorder prediction (Algorithm 2). This model combines convolutional layers for capturing local motifs with Transformer encoders for long-range sequence dependencies, making it well-suited for protein sequence analysis. A full description of this model is provided in Section 1.4.

This comprehensive set of models reflects the evolving strategies in tabular data modeling. Tree-based ensembles like RandomForest and ExtraTrees remain competitive with minimal preprocessing. Deep learning models such as TabNet, DANet, and GANDALF offer advanced feature abstraction and end-to-end learning capabilities. The performance of all models on the validation set is reported in the Results section.

1.2.4 Evaluation Metrics for Performance Assessment

Given the significant class imbalance in the disorder dataset, as illustrated in the Dataset section, we evaluate model performance using three key metrics: Area Under the Curve (AUC), Average Precision (AP), and the maximum F1 score (F1 max).

AUC (Area Under the ROC Curve): AUC measures the model's ability to rank positive instances higher than negative ones. It is computed as the area under the receiver operating characteristic (ROC) curve, which plots the true positive rate (TPR) against the false positive rate (FPR) (see Eq. (1)). The AUC value ranges from 0 to 1, where 1.0 indicates perfect discrimination and 0.5 represents random performance.

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(\mathrm{FPR})\ \mathrm{dFPR} \tag{1}$$

Average Precision Score (APS): Average Precision summarizes the precision-recall curve and is defined as the weighted mean of precisions at each threshold (see Eq. (2)) where recall increases.

$$\mathrm{AP} = \sum_k [R(k) - R(k-1)] \cdot P(k) \tag{2}$$

where, $P(k)$ and $R(k)$ denote the precision and recall at the k threshold. APS is informative for imbalanced datasets, where it reflects the model's ability to retrieve relevant instances.

F1 max (Maximum F1 Score): F1 score is the harmonic mean of precision and recall (see Eq. (3)):

$$\mathrm{F1} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

F1 max is the maximum F1 score obtained by varying the classification threshold t over a set of possible thresholds T (see Eq. (4)):

$$F1_{max} = \max_{t \in T} F\,1(t) \tag{4}$$

1.2.5 Protein Representation

We represent proteins using protein language models (PLMs)—deep learning models inspired by natural language processing (NLP) techniques. These models treat amino acid sequences as a form of "biological language," where sequences of residues carry implicit structural, functional, and evolutionary information. By training on large-scale protein databases such as UniProt, these models learn to capture sequence patterns, residue dependencies, and contextual representations without requiring explicit structural labels [3, 37]. At the core of most PLMs lies the transformer architecture, originally developed for NLP tasks. Models like ESM (Evolutionary Scale Modeling), ProtBert, TAPE, and MSA Transformer have demonstrated the ability to learn high-quality embeddings of protein sequences [4, 38, 39]. These embeddings encode features such as secondary structure, disorder, binding affinity, and functional domains.

Protein language models are powerful because they are unsupervised, scalable, and general-purpose. Once trained, their learned embeddings can be fine-tuned or directly applied to a wide range of biological prediction tasks, often outperforming traditional methods [4, 38]. Their interpretability is still under exploration, but attention maps and latent representations offer potential insights into biologically meaningful patterns. As PLMs continue to scale and integrate structural supervision (as seen in ESMFold and AlphaFold-style models) [39-41], they are expected to become central tools in computational biology, aiding in protein function annotation, design, and drug discovery [39, 40].

In this study, we extract embeddings from the ESM family of models, developed by Meta AI (formerly Facebook AI Research), which are large-scale Transformers trained on hundreds of millions of protein sequences using masked-language modeling. ESM embeddings have shown strong performance on tasks such as remote homology detection, structure prediction, and zero-shot classification. More recent ESM2 and ESMFold models push this further by integrating language modeling with structure prediction, achieving near AlphaFold-level performance in certain benchmarks [39-41]. MSA Transformer, another variant, incorporates multiple sequence alignments as input, allowing the model to directly learn from evolutionary relationships across homologous sequences [38].

1.3 Feature Selection, Engineering, and Post-processing

We integrated multiple types of features derived from state-of-the-art approaches, including protein language model embeddings, predicted structural representations, and sequence-based descriptors. To evaluate the relative importance of each feature type, we systematically removed or altered individual components and measured the resulting changes in predictive performance. This ablation analysis was essential for quantifying the specific contribution of each feature category to the overall model accuracy and robustness.

The feature engineering strategy employed in ESMDisPred involves collecting four key types of information to construct the final statistical feature set: (1) disorder predictions generated by DisPredict3.0, (2) protein sequence embeddings extracted from ESM2, (3) residue-level filtering to remove positions with missing structural data, and (4) terminal residue annotations encoding each residue's relative position within the sequence. These features are then transformed into statistical descriptors that capture both local and global sequence contexts, providing rich input representations that enhance the model's accuracy. We also analyze post-processing to improve the residue-level stability and interpretability, specifically evaluating probability-smoothing methods—moving average, exponential moving average, and Gaussian filtering. All performance evaluations of feature-engineering strategies and post-processing are reported on the validation set, ensuring a consistent basis for comparison. The following section discusses the effect of each feature engineering technique we experimented with.

1.3.1 Effect of Large Protein Sequences

We first explore the impact of excluding large proteins on the performance of the model. Figure 5 presents the impact of filtering out large proteins based on sequence length thresholds on model performance across APS, AUC, F1 max, and average improvement metrics. The base model, which includes all proteins regardless of size, serves as a reference point. Removing proteins longer than 2000 residues yields the most significant performance gains, with APS improving to over 11% and F1 max reaching around 7%, resulting in the highest average performance improvement among all tested conditions. Excluding proteins above 2500 residues also boosts performance, though slightly less effectively. Conversely, applying a more aggressive filter by

removing proteins longer than 1500 residues leads to weaker improvements, particularly in F1 max and AUC, suggesting potential loss of valuable sequence diversity. These findings indicate that eliminating extremely long proteins helps mitigate complexity or noise during model training.

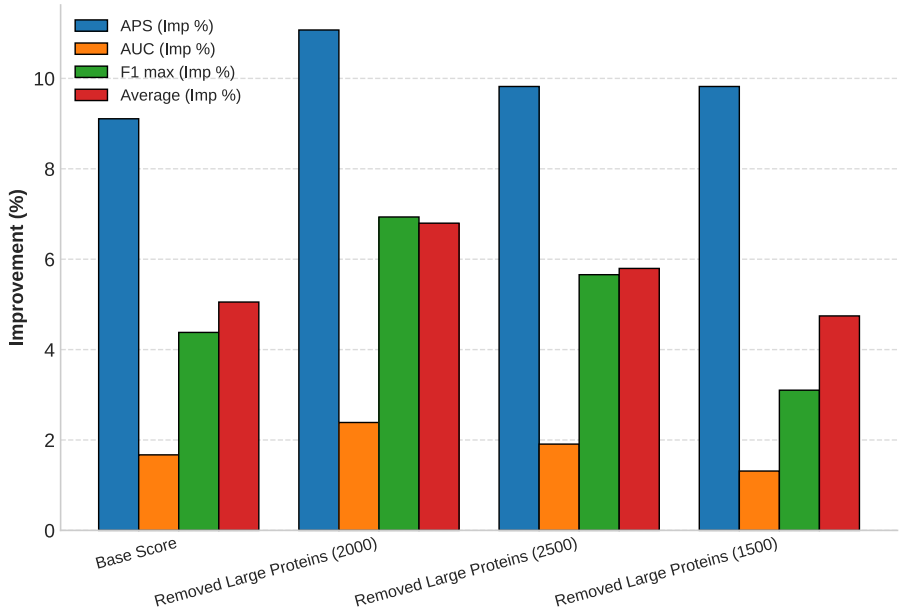


Figure 5. Impact of removing large proteins on various performance metrics. The percentage improvements in Average Precision Score (APS), Area Under Curve (AUC), maximum F1 score (F1 max), and the overall average performance are shown for different thresholds of protein length removal (1500, 2000, and 2500 amino acids) compared to the base model score.

1.3.2 Effect of ESM2 Statistics

We further analyzed the ESM2-derived statistical features—including metrics such as sequence length, mean, standard deviation, variance, and higher-order moments (e.g., skewness and kurtosis) to assess their contribution to predictive performance. These statistics provided complementary information that improved the model's robustness, particularly when combined with other features in an ensemble setting. Figure 6 illustrates the effect of incorporating ESM2-derived features on model performance across four evaluation metrics: APS, AUC, F1 max, and their average. The base model, which excludes ESM2 information, shows the lowest performance across all metrics. Incorporation of raw ESM2 embeddings leads to notable improvements, particularly in APS (~12.2%) and F1 max (~8%), demonstrating the utility of pretrained language model representations in capturing relevant protein sequence information. Further refinement

using ESM2 statistical features results in the highest observed gains, with APS nearing 16% improvement and F1 max exceeding 11%. These results highlight that not only the inclusion of ESM2 features, but also their statistical summarization, can significantly enhance the model's ability.

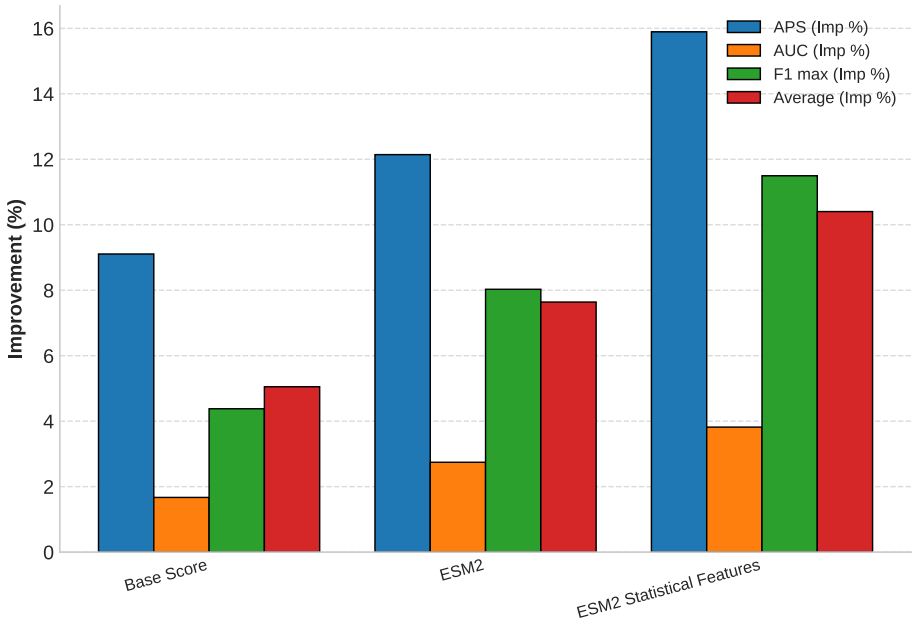


Figure 6. Performance improvements resulting from integrating ESM2 embeddings and ESM2-derived statistical features into the model. Shown are the percentage improvements in Average Precision Score (APS), Area Under Curve (AUC), maximum F1 score (F1 max), and the overall average performance compared to the base model score.

1.3.3 Effect of Terminal Values

We also investigated the influence of terminal regions on model performance. Exploratory data analysis revealed that disordered residues are predominantly concentrated in the N- and C-terminal regions of protein sequences. This observation motivated us to explicitly include terminal residue information in our feature set. Incorporating these regions led to improved predictive accuracy, particularly for proteins exhibiting high intrinsic disorder near sequence termini.

Figure 7 illustrates two encoding schemes applied to a protein sequence: a normalized scale from –1 to 1 (shown in blue) and a positive symmetric scale from 1 to 1 (shown in green). Each residue is assigned a numerical value based on its relative position, with central residues receiving values

near zero and terminal residues approaching the extremes. Normalized encoding provides a continuous, direction-aware representation ideal for terminal residue annotation. In contrast, the positive symmetric encoding reflects distance from the sequence center without directional sign, which may benefit models sensitive to relative rather than absolute position.

Protein Sequence	**Y**	**Y**	**P**	**K**	**S**	**G**	**T**	**G**	**K**
Pos-Neg Annotation	**-1**	**-0.75**	**-0.5**	**-0.25**	**0**	**0.25**	**0.5**	**0.75**	**1**
Positive Annotation	**1**	**0.75**	**0.5**	**0.25**	**0**	**0.25**	**0.5**	**0.75**	**1**

Figure 7. Example of annotating protein sequences for terminal residue encoding. Residues are annotated using two schemes: a normalized scale from –1 to 1 (blue), encoding direction and position, and a positive symmetric scale from 1 to 1 (green), encoding relative distance from the center. Central residues receive values near 0, while terminal residues approach the extremes. These encodings provide position-aware input features for modeling sequence context.

Figure 8 examines the impact of terminal residue annotations on predictive performance. The base model, which lacks any positional encoding of terminal residues, shows the lowest performance. The "Terminal Pos-Neg," which uses a symmetric encoding scheme ranging from –1 to 1 to highlight both N- and C-termini, yields the highest improvement in APS (~11.8%) and F1 max (~8%). This indicates that emphasizing both termini enhances the model's ability to capture biologically relevant positional cues. In contrast, the "Terminal Positive-Only," which encodes positional information with only positive values, provides a modest improvement over the base but underperforms compared to the symmetric scheme. By distinguishing both terminal directions, the model gains a clearer context, yielding higher predictive performance.

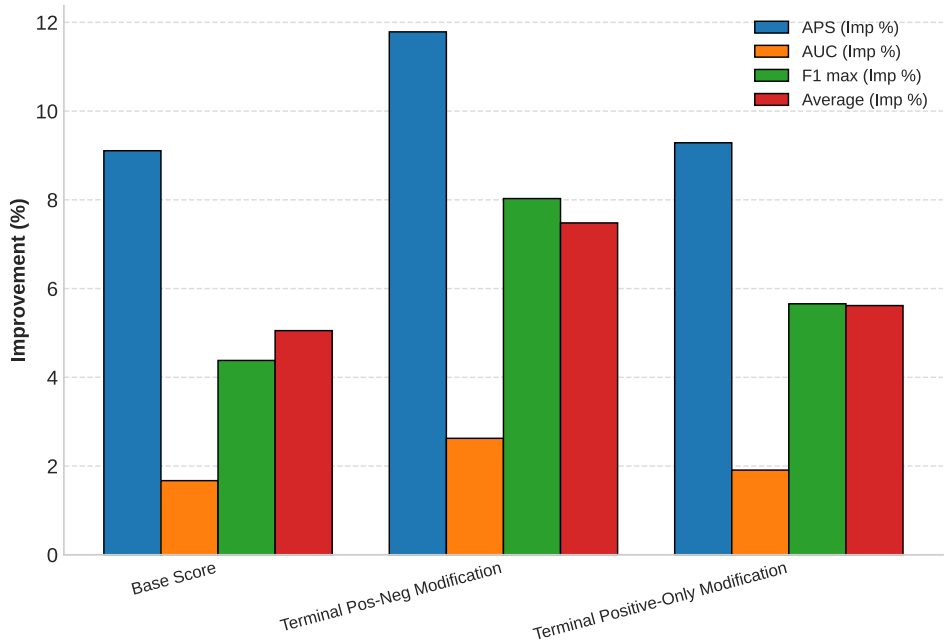


Figure 8. Effect of incorporating terminal residue annotations on model performance. The figure displays the percentage improvements in Average Precision Score (APS), Area Under Curve (AUC), maximum F1 score (F1 max), and overall average performance for two terminal modification scenarios: considering both positive and negative terminals (Pos-Neg) versus considering only positive terminals (Positive-Only).

1.3.4 Effect of Removing Missing Residues

We evaluated the impact of removing residues (amino acids) present in the sequence but lacking atomic coordinates in experimentally determined structures (e.g., Protein Data Bank (PDB) entries). These missing residues often result from unresolved regions due to intrinsic flexibility, truncation, or experimental limitations. Although commonly treated as disordered, their status is ambiguous and can introduce noise into training data. To address this, we removed proteins containing missing residues from the training dataset. This filtering led to cleaner training data. Figure 9 illustrates the effect of this filtering on performance metrics. Compared to the base model, removing proteins with missing residues led to a marked increase in predictive performance. Specifically, the Average Precision Score (APS) increased from ~9% to over 12.5%, indicating improved ranking of disordered residues. The average of all metrics also improved, rising from ~5% to ~6%. The changes in AUC and F1 max were more modest, suggesting that while ranking and recall improved, binary classification performance remained stable. These findings underscore

the importance of data quality in structure-informed learning. Including missing residues can introduce inconsistencies between sequence and structure, degrade feature reliability, and mislabel flexible but ordered regions. Filtering such cases improves learning signals and enhances generalization in disorder prediction models.

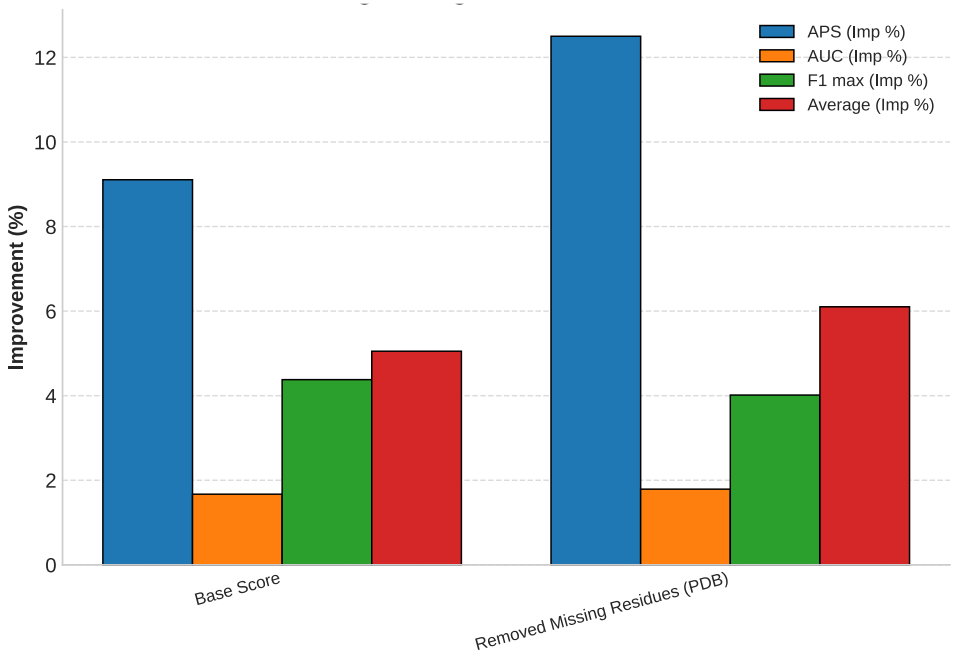


Figure 9. Effect of removing proteins with missing residues on model performance metrics. Missing residues refer to amino acids lacking structural coordinates in PDB data. The bar chart shows percentage improvements in Average Precision Score (APS), Area Under the Curve (AUC), maximum F1 score (F1 max), and the overall average performance after filtering proteins with incomplete structural data. Removal leads to cleaner training data.

1.3.5 Effect of Windowing

The windowing technique (or sliding window) is a widely used method that involves moving a fixed-length subsequence along the protein sequence to capture the local contextual information surrounding each residue [42-46]. Figure 10 illustrates the windowing technique on a protein sequence. In the left panel, the model processes the entire sequence as a whole, extracting features globally at each residue position. While this approach captures overall sequence information, it may miss important local patterns, particularly those associated with intrinsic disorder. The right panel demonstrates the windowing strategy, where the sequence is segmented into overlapping

windows (e.g., triplets such as V–K–G, K–G–L, etc.), and features are extracted for each window independently. This method enables the model to focus on local context and short-range dependencies, which are often critical for detecting regions of disorder. By aggregating the features from multiple overlapping windows, the model builds a richer, more localized representation of the sequence.

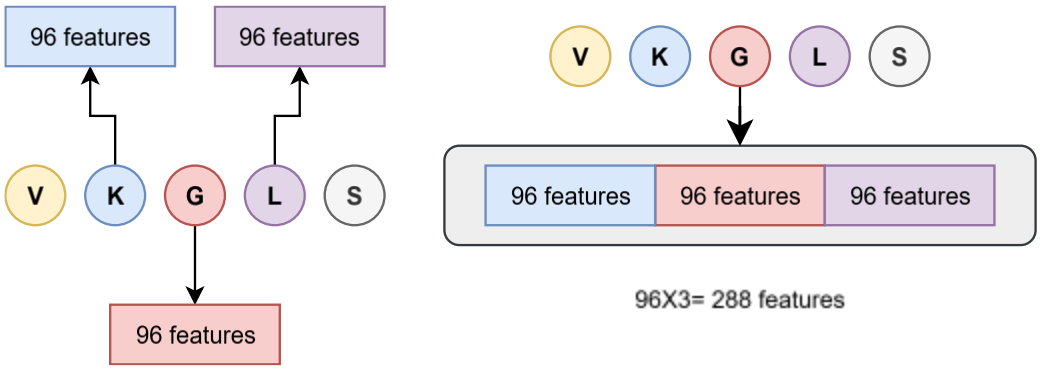


Figure 10. Illustration of window-based statistical feature extraction from protein sequences. Each amino acid residue in a sequence (e.g., VKGLS) is represented by a set of statistical features (e.g., 96-dimensional feature vectors) derived from a sliding window approach. The window encompasses neighboring residues to capture local context, resulting in a feature representation that encodes local physicochemical and contextual information.

In machine learning, the curse of dimensionality poses a significant challenge, as increasing the number of input features can lead to overfitting and degraded generalization. To mitigate this, we applied the windowing technique selectively to features where local context is most informative, such as disorder probabilities, and ESM2-derived statistical features. Figure 11 presents the impact of applying windowing techniques with varying window sizes on the performance of selected features. The base score, which does not use windowing, shows the lowest performance across all metrics. Among the tested window sizes, window size 7 yielded the highest overall improvement, particularly in APS (over 16%) and F1 max (11.5%). Window sizes 5 and 3 also demonstrated consistent performance gains, suggesting that smaller window sizes are effective in capturing local sequence patterns. However, the improvement in AUC was more modest across all windowing configurations, indicating that some metrics are more sensitive to local feature modeling than others.

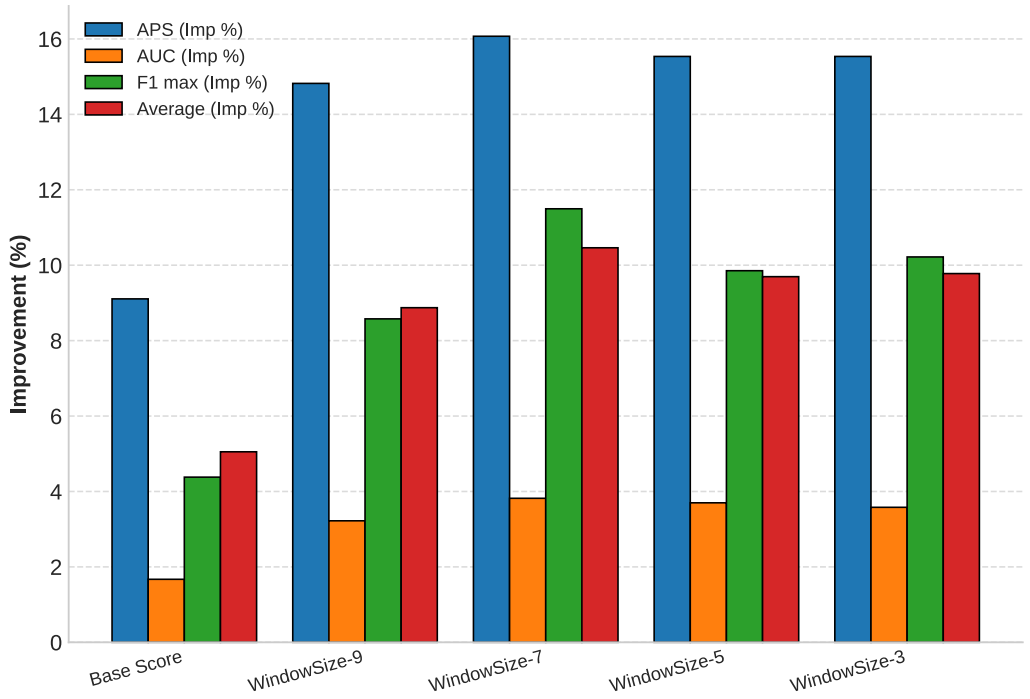


Figure 11. Impact of varying window sizes for ESM2-derived statistical features on model performance metrics. Percentage improvements in Average Precision Score (APS), Area Under Curve (AUC), maximum F1 score (F1 max), and overall average performance are compared across different window sizes (3, 5, 7, 9). The results indicate that optimal performance improvements vary with the chosen window size.

1.3.6 Effect of Smoothing techniques

To further enhance the stability and reliability of residue-level disorder predictions, we applied several smoothing techniques aimed at reducing local fluctuations in the predicted probability scores. These fluctuations often occur in boundary regions where the transition between ordered and disordered states is ambiguous, leading to inconsistent predictions. By smoothing the output scores, we were able to generate more coherent disorder profiles that better align with biologically meaningful regions. Figure 12 provides a residue-level plot of disorder probability predictions. The blue line shows the original raw probability values, which fluctuate considerably between residues, while the red line represents the smoothed probabilities. The smoothed curve offers a clearer and more interpretable trend, highlighting regions with high disorder probability (approaching 1.0) and areas of low disorder likelihood (closer to 0.2). This visual comparison confirms that smoothing not only reduces noise but also helps identify coherent disorder patterns within protein sequences.

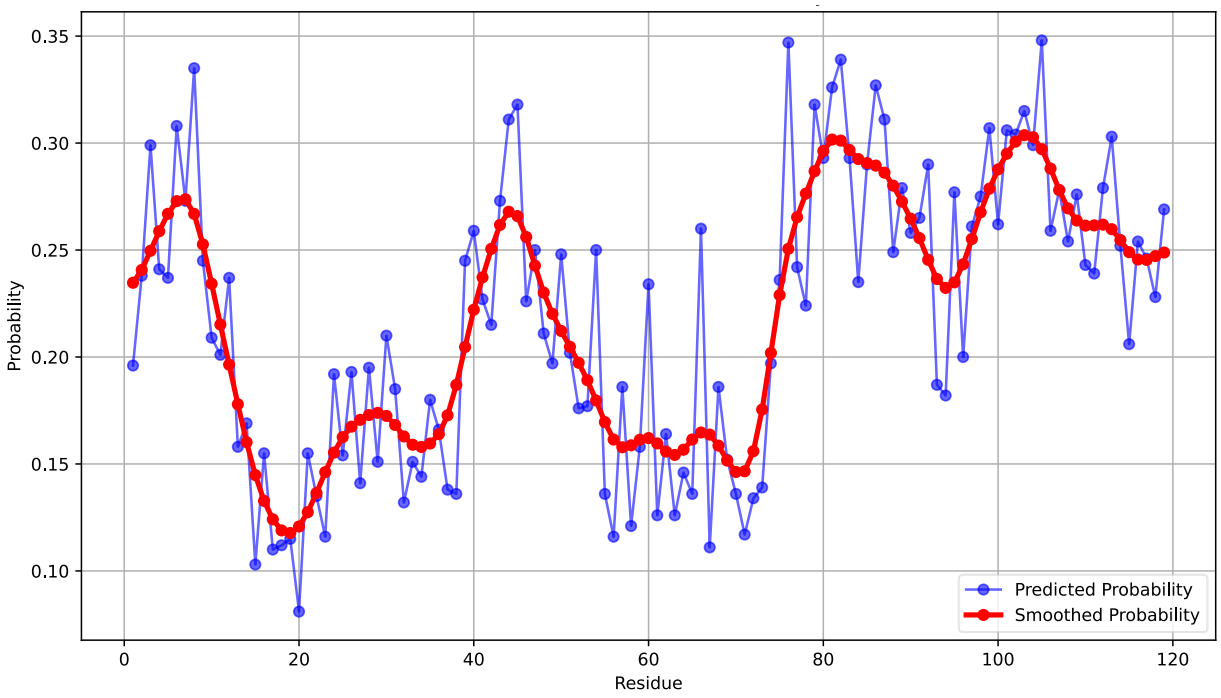


Figure 12. The effect of smoothing in disordered proteins. The x-axis represents the residue position, while the y-axis shows the probability of disorder. The blue line represents the original probability predictions for each residue, displaying considerable fluctuations. The red line represents the smoothed probability values, offering a clearer trend. The smoothed curve provides a more interpretable view of protein disorder prediction.

We evaluated the effects of moving average, exponential moving average, and Gaussian smoothing methods. As illustrated in Figure 13, all three smoothing techniques led to incremental improvements in ROC AUC performance when compared to the baseline (unsmoothed) predictions. The Gaussian smoothing method achieved the most significant gain, followed closely by the exponential moving average, indicating their effectiveness in capturing broader contextual patterns without overly distorting localized predictions. These results suggest that incorporating smoothing as a post-processing step can enhance the performance of disorder prediction models.

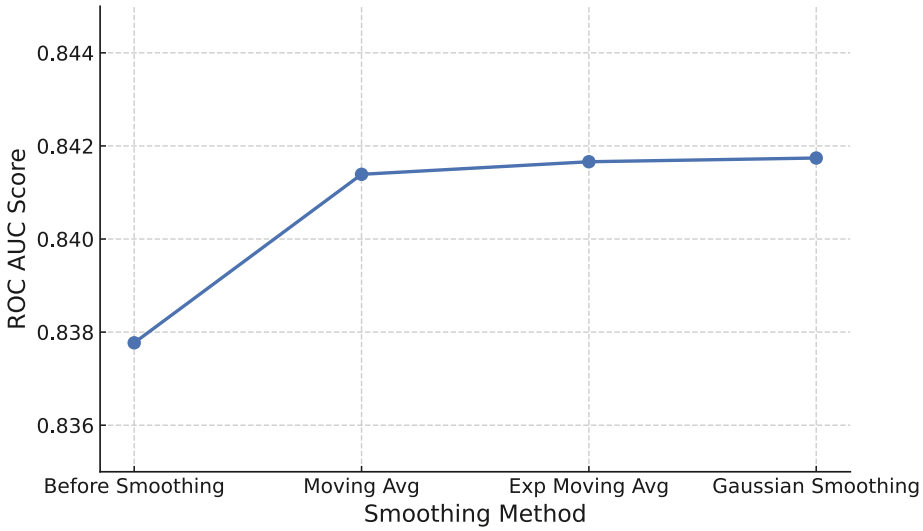


Figure 13. Impact of different smoothing techniques on ROC AUC performance. The ROC AUC scores before and after applying moving average, exponential moving average, and Gaussian smoothing methods are compared, demonstrating incremental improvements due to smoothing.

1.4 Architecture of ESMDisPred

We developed and trained four model variants within the ESMDisPred framework, focusing on different feature combinations, sequence representations, and algorithms. Specifically, three ESMDisPred-LightGBM variants were submitted to the CAID3 challenge, each designed to evaluate the contribution of distinct feature sets. ESMDisPred-1 utilized features from DisPredict3.0 and ESM1, ESMDisPred-2 incorporated both DisPredict3.0 and ESM1 features along with additional ESM2-derived representations, and ESMDisPred-2PDB further extended the configuration by adding structure-related features obtained through the replacement of missing residues in experimental structures. Across multiple benchmark datasets in the CAID3 challenge, these models consistently outperformed existing state-of-the-art tools in terms of accuracy, precision, and recall.

The first three variants share an identical architectural design but differ in their input feature compositions, allowing systematic assessment of how various feature combinations influence predictive performance. The ESMDisPred-LightGBM (Algorithm 1), architecture integrates pretrained protein language model embeddings with a LightGBM classifier to predict intrinsically disordered regions in protein sequences. The process begins with preprocessing protein sequences

to remove long or redundant entries. For each sequence, embeddings are generated using the ESM2 language model, which captures contextual and evolutionary features from raw amino acid sequences. These embeddings are transformed into statistical features such as mean, variance, and kurtosis and enriched with local context through sliding window techniques. Additionally, disorder-specific features are extracted using the DisPredict3.0 tool. The resulting features are then used to train a LightGBM model, which is optimized using cross-entropy loss and tuned hyperparameters (e.g., number of estimators). During inference, features from both ESM2 and DisPredict3.0 are computed for test sequences and passed into the trained LightGBM model to generate residue-level disorder probabilities. Post-processing techniques like smoothing are applied to refine the output. This architecture combines the rich representational power of deep protein embeddings with the gradient-boosted trees, resulting in a high-performing and scalable disorder prediction system.

Algorithm 1: ESMDisPred – Disorder Prediction using LightGBM (ESMDisPred-LightGBM)

Input:

- Protein sequence dataset $D = \{S_1, S_2,...,S_n\}$.

Output:

- Disorder probability scores for each amino acid.

  Step 1: Data Preprocessing

  For each protein sequence $S_i \in D$:

  - Filter out sequences longer than 2000 amino acids.
  - Remove redundant sequences using a 25% sequence identity cutoff.

  Step 2: Feature Extraction using ESM and DisPredict3.0

  For each sequence $S_i \in D$:

  - Obtain ESM2 embeddings: $F_i = \text{ESM2}(S_i)$.
  - Compute statistical features (mean, variance, skewness, kurtosis, etc.).
  - Apply optimal windowing for local context features.
  - Extract disorder-related features using DisPredict3.0.

  Step 3: Model Training using LightGBM

  - Define LightGBM model parameters:
    - Optimize number of estimators (e.g., via grid search).
  - Train LightGBM model using extracted features.

Step 4: Disorder Prediction

For each test sequence S_t:

- Extract ESM2 and DisPredict3.0 features:

  F_t = ESM2(S_t) + DisPredict3.0(S_t)

- Predict disorder probability:

  P_t = M(F_t), where M is the trained model.

Step 5: Post-processing

- Apply smoothing techniques to predicted disorder scores.
- Evaluate prediction performance using metrics such as AUC, F1 max, and Average Precision.
- Return disorder probability scores for each amino acid.

To further enhance predictive performance of intrinsically disordered regions (IDRs), we designed ESMDisPred-DNN, an advanced Transformer–CNN-based deep neural network. As outlined in Algorithm 2, this model employs a hybrid architecture that integrates convolutional neural networks (CNNs) and Transformer encoders to capture both local and long-range dependencies within protein sequences. The motivation for this design arises from the observation that disordered proteins often contain both short and long disordered regions, requiring a model capable of learning across multiple spatial scales. The CNN layers effectively identify local sequence motifs, while the Transformer layers capture contextual relationships over longer sequence spans, making the architecture particularly well-suited for per-residue disorder prediction.

In ESMDisPred-DNN, we formulate residue-level IDR prediction as a per-token binary classification problem over variable-length protein sequences. For a given sequence of length L each residue $t \in 1,2, \dots, L$ is represented by a standardized feature vector $f_t \in R^{d_i}$. All features are z-score normalized using statistics computed from the training split and subsequently applied to the validation and test sets to ensure consistent scaling. During training, mini-batches are padded to the length of the longest sequence within each batch. A key-padding mask M is applied to the attention logits so that padded positions neither attend to nor are attended by other residues; the same mask is also used in the loss function.

Each residue feature is projected to the model dimension and combined with fixed sinusoidal positional encodings (see Eq. (5)):

$$x_t = W_e f_t + b_e + p_t \tag{5}$$

where x_t represents the input at position t in the model's embedding space, $W_e \in R^{d_{model} \times d_i}$ and $b_e \in R^{d_{model}}$ are learnable linear projections that maps the input features from dimension d_i to the model's embedding dimension d_{model}. $p_t \in R^{d_{model}}$ represents the fixed sinusoidal positional encoding.

Before global attention, a lightweight, length-preserving convolutional stem injects local inductive bias via three depthwise 1D convolutions (kernels 5, 9, 15), each followed by a pointwise projection. We added convolutions layer so that the model can learn the short- and mid-range motifs. Sequence context is integrated by an eight-block pre-norm Transformer (LayerNorm → sublayer → residual). Within each block, queries (Q), keys (K), and values (V) are formed from the hidden states $H \in R^{L \times d_{model}}$ as shown in Eq. (6)).

$$Q = HW_Q, \quad K = HW_K, \quad V = HW_V \tag{6}$$

Where $W_Q, W_K, W_V \in R^{d_{model} \times d_k}$ are linear projection matrices produce the queries, keys, and values. Masked multi-head self-attention then computed as follows (see Eq. (7)):

$$\text{Attn}(H) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V \tag{7}$$

where d_k is the per-head key/query size and M injects the padding mask into the attention logits. Using pre-norm residual structure, block l updates are (see Eq. (8)):

$$\begin{aligned} H_{msa}^{(l)} &= MSA\left(LN\left(H^{(l-1)}\right)\right) + H^{(l-1)} \\ H^{(l)} &= FFN\left(LN\left(H_{msa}^{(l)}\right)\right) + H_{msa}^{(l)} \end{aligned} \tag{8}$$

where LN is LayerNorm, Multi-Head Self-Attention (MSA) concatenates attention heads. The output of this layer is a position-wise feed-forward network (FFN) (see Eq. (9)). We chose GELU as the activation function because it often outperforms ReLU and tanh in transformer architectures (e.g., BERT, GPT, ViT) [47, 48]. GELU also better models nonlinearity in continuous feature

spaces, especially for embeddings or attention outputs that are approximately Gaussian-distributed.

$$\mathrm{FFN}(u) = W_2\,\mathrm{GELU}(W_{1u} + b_1) + b_2 \tag{9}$$

For residue-wise classification, we apply dropout to the final hidden state h_t and a position-wise linear layer to obtain a logit z_t and a disorder probability $\widehat{p_t}$ (see Eq. (10)):

$$z_t = w^T Dropout(h_t) + b \qquad \widehat{p_t} = \sigma(z_t) = \frac{1}{1 + e^{-z_t}} \tag{10}$$

Training minimizes a masked binary cross-entropy loss that averages only over valid (non-padded) residues. During validation, macro per-protein ROC–AUC is computed by averaging AUC values across proteins in each epoch. Early stopping monitors this macro-AUC, and the best-performing checkpoint is restored after training. To improve calibration, the output probabilities are refined via Platt scaling [49] (see Eq. (11)), where the calibrated probabilities are:

$$\widehat{p_t^{\mathrm{cal}}} = \sigma(az_t + b) \tag{11}$$

where a, b learned by logistic regression. The calibrated probabilities are subsequently smoothed along the sequence using a Gaussian kernel with bandwidth $\sigma_g > 0$ and normalization constant $Z > 0$ (see Eq. (12)):

$$\widetilde{p_t} = \frac{1}{Z}\sum_{u=1}^{L} \exp\left(-\frac{(t-u)^2}{2\sigma_g^2}\right) \widehat{p_u^{\mathrm{cal}}} \tag{12}$$

Here, Z ensures proper normalization of the Gaussian weights, t denotes the target residue position, and u iterates over all sequence positions contributing to the smoothed estimate. The smoothing bandwidth σ_g controls how broadly neighboring residues influence the prediction.

This architectural choice balances between model complexity, training stability, and computational efficiency, which are essential for accurate residue-level predictions in long protein sequences. By combining a multi-kernel convolutional stem with Transformer-based global attention, the model effectively captures both local sequence motifs and long-range dependencies. Details of the model's optimal hyperparameter settings are discussed in the Hyperparameter Selection section.

Algorithm 2: Transformer-based Per-Residue Disorder Prediction (ESMDisPred-DNN)

Inputs: Variable-length protein sequences with per-residue features (Algorithm 1 - Step 2).

Outputs: Per-residue disorder probabilities.

Step 1: Pre-processing & Batching

- Standardize features by fitting the scaler on the training set and applying the same transform to validation and test set.
- Batch variable-length sequences by padding each batch to the longest sequence and building a padding mask, so padded tokens never affect attention, loss, or metrics.

Step 2: Model Forward (Per Token)

- Project residue features the model width with a linear layer followed by dropout.
- Add fixed sinusoidal positional encodings.
- Apply a depthwise-separable 1D convolutional stem with kernel sizes 5, 9, and 15 to capture short- and medium-range motifs.
- Run a Transformer encoder where each layer does:
  LayerNorm → multi-head self-attention (using the key-padding mask) + residual;
  LayerNorm → feed-forward network + residual.
- For each residue, apply a linear classifier to the final token embedding to produce a scalar logit, then apply a sigmoid to obtain the disorder probability.

Step 3: Loss & Optimization

- Use a masked binary cross-entropy computed only on real tokens; average over tokens within each protein (sequence), then average across the batch.
- Train with AdamW (with weight decay), gradient clipping, and a cosine learning-rate schedule with warm-up.

Step 4: Validation, Early Stopping, Checkpoint

- Validate each epoch and compute token-level macro per-protein AUC.
- Early-stop based on the macro-AUC: save when it improves, stop after patience is exceeded, and restore the best weights.

Step 5: Post-processing

- Fit Platt scaling on the validation set and apply it to validation and test predictions.
- Apply per-protein Gaussian 1D smoothing to reduce spurious spikes.

1.5 Experimental Results

This section presents a structured evaluation of protein disorder prediction models using CAID benchmarks. We begin with a validation set comparison. Models including LightGBM, CatBoost, XGBoost, Random Forest, and Logistic Regression are evaluated using APS, AUC, and F1 max. LightGBM consistently outperforms all others across these metrics. We then analyze the enhanced ESMDisPred-DNN architecture. By integrating protein language model features and deep sequential modeling, it surpasses the LightGBM-based variant in prediction accuracy. Comparative analyses against state-of-the-art predictors follow. ROC and Precision–Recall curves identify ESMDisPred–2PDB as the top-performing model. To assess the significance of model differences, we applied DeLong statistical tests. These confirmed that the observed performance gains are statistically significant. ESMDisPred demonstrates an optimal balance between accuracy and computational speed. Each subsection includes a corresponding figure to support the reported findings.

1.5.1 Model Comparison and Selection on the Validation Set

A comparative performance evaluation of several machine learning models such as LightGBM, CatBoost, XGBoost, Random Forest, and Logistic Regression was conducted on the Disorder NOX dataset from the CAID2 benchmark (Figure 14). The figure reports performance in terms of Average Precision Score (APS), Area Under the Curve (AUC), maximum F1 score (F1 max), and an overall average of these metrics, with percentage improvements calculated relative to the state-of-the-art (SOTA) model. Among all the methods, LightGBM shows the highest performance, achieving improvements of up to 7.5% in APS, 6.2% in AUC, 8.9% in F1 max, and 7.5% in the overall average performance. These results demonstrate LightGBM's superior capacity to capture complex relationships in protein sequence-derived features. Its gradient boosting framework, efficient handling of large-scale data, and robustness against overfitting make it particularly well-suited for the task of disordered region prediction. In contrast, while CatBoost and XGBoost offer competitive results, they fall short of LightGBM's consistency across all metrics. Logistic Regression and Random Forest, being more traditional models, lag significantly behind,

highlighting the advantages of modern gradient boosting techniques in high-dimensional bioinformatics applications.

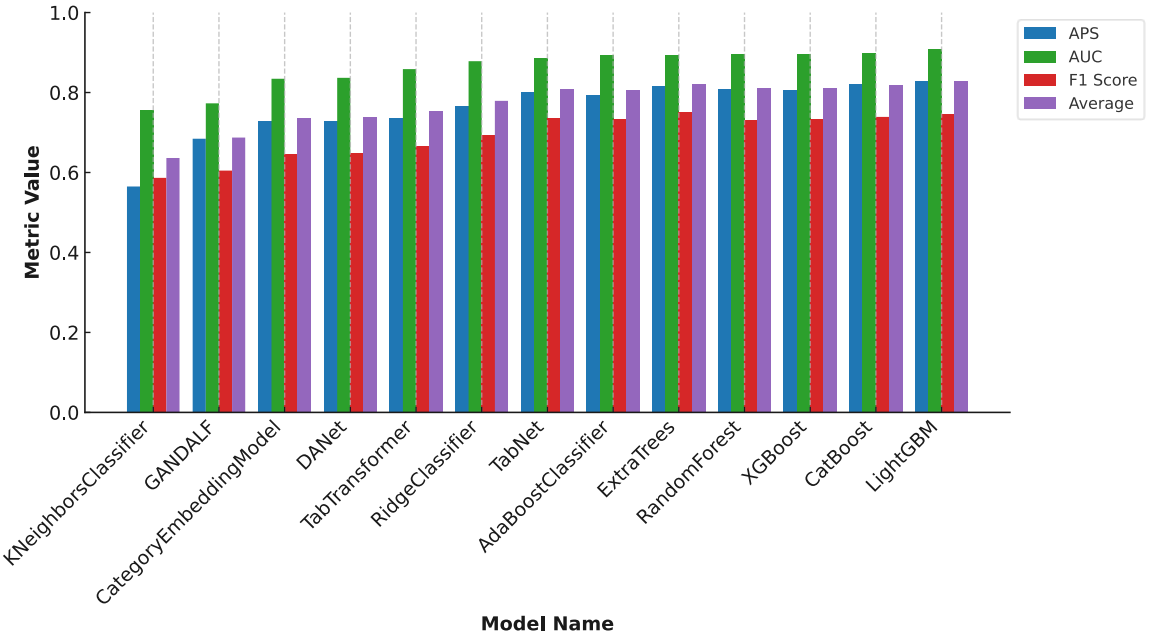


Figure 14. Performance comparison of various machine learning models (LightGBM, CatBoost, XGBoost, Random Forest, Logistic Regression) on the Disorder NOX Dataset (CAID2). The percentage improvements over the state-of-the-art (SOTA) model are shown for Average Precision Score (APS), Area Under Curve (AUC), maximum F1 score (F1 max), and overall average performance metrics, demonstrating that LightGBM achieves superior results among the tested methods.

For the ESMDisPred-LightGBM models, we focused on optimizing the number of estimators, which controls the number of sequential learners in the ensemble. This parameter was chosen because of its direct influence on model complexity and predictive performance in boosting-based algorithms. Increasing the number of estimators typically reduces bias and improves accuracy but can also increase the risk of overfitting and computational cost. Hence, identifying an optimal range is essential.

Figure 15 presents the average performance improvement as a function of the number of estimators. The model achieved its highest improvement at around 100 estimators, marking the optimal balance between performance and training efficiency. Beyond this point, performance showed minor fluctuations and no consistent improvement, eventually declining slightly after 1500

estimators. This pattern suggests that increasing the number of estimators beyond a moderate range leads to diminishing returns and may even reduce overall performance due to overfitting or increased variance.

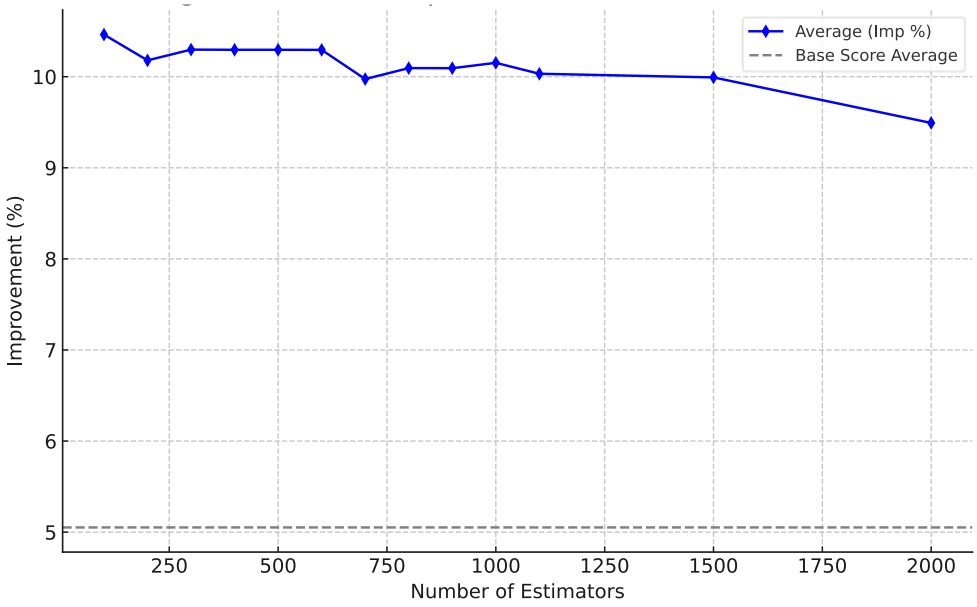


Figure 15. Average performance improvement as a function of the number of estimators in the LightGBM model.

For the ESMDisPred-DNN model, we explored hyperparameters to understand how capacity, regularization, and post-hoc calibration affect generalization and performance. A grid search was conducted over the following ranges (Table 1): projection width (256–512), attention heads (8–16), Transformer depth (4–8 layers), and feed-forward width (768–2560). These were paired with dropout rates (0.20–0.30), weight decay (0.01–0.05), and learning rates (1e−4–3e−4). Each model was trained with early stopping based on the macro AUC metric. To compute macro AUC, we first calculated the AUC for each protein sequence individually, based on residue-level disorder probabilities, and then averaged these values across all proteins. This per-protein macro averaging ensures that each protein contributes equally to the evaluation, regardless of sequence length or class imbalance. Such an approach provides a more reliable measure of the model's generalization ability across diverse proteins, preventing bias toward longer or majority-class (ordered) sequences. We selected macro AUC for early stopping because it better reflects balanced

predictive performance across all proteins and disorder categories. The final performance was assessed using AUC metrics on the validation set.

Table 1. Hyperparameter search space with minimum and maximum values, along with the number of unique sampled values used during model optimization.

Hyperparameter	Min	Max	Unique values
Projection Dimension	256	512	4
Number of Attention Heads	8	16	4
Number of Layers	4	8	3
Feed-Forward Dimension	768	2560	10
Dropout Rate	0.2	0.3	3
Learning Rate	0.0001	0.0003	3
Weight Decay	0.01	0.05	3

Across 35 independent training runs (Table 2), model performance consistently clustered within a narrow range, indicating that the prediction task is robust to moderate architectural and regularization variations. Table 2 summarizes all 35 ESMDisPred-DNN configurations sampled during the grid search, along with their corresponding validation AUC values.

Table 2. Summary of all 35 ESMDisPred-DNN training runs. Each row represents a unique hyperparameter configuration sampled during the grid search, with corresponding validation performance metrics.

Calibrate	Dropout	Lr	Projection Dimension	FF Dimension	No. of head	No. of Layers	Weight Decay	Validation AUC
Platt	**0.2**	**0.0002**	**512**	**2048**	**8**	**8**	**0.05**	**0.952**
isotonic	0.2	0.0001	256	1024	8	6	0.01	0.948
Platt	0.2	0.0002	512	1536	8	8	0.02	0.948
isotonic	0.2	0.0001	256	1024	8	6	0.02	0.948
isotonic	0.3	0.0001	512	2560	16	4	0.05	0.948
isotonic	0.3	0.0001	320	960	10	6	0.05	0.947
isotonic	0.25	0.0003	384	1920	8	8	0.01	0.947
none	0.2	0.0001	256	1024	8	8	0.05	0.947
none	0.2	0.0002	384	768	8	8	0.02	0.946
Platt	0.3	0.0002	512	2048	16	6	0.05	0.946
isotonic	0.25	0.0002	384	1536	12	6	0.02	0.946
isotonic	0.3	0.0002	320	1600	10	4	0.05	0.945
none	0.25	0.0002	256	1024	8	6	0.05	0.945
none	0.3	0.0001	512	1536	16	4	0.02	0.945
Platt	0.25	0.0002	320	1280	10	4	0.02	0.945

isotonic	0.2	0.0001	512	2560	16	6	0.02	0.945
Platt	0.25	0.0001	512	2048	16	6	0.01	0.945
none	0.3	0.0001	384	1152	8	4	0.02	0.944
none	0.3	0.0003	320	960	10	8	0.02	0.944
none	0.2	0.0003	512	2560	8	4	0.01	0.944
none	0.3	0.0001	256	1280	8	4	0.01	0.944
Platt	0.25	0.0003	256	768	8	6	0.02	0.943
isotonic	0.3	0.0001	512	2560	16	6	0.05	0.943
Platt	0.2	0.0002	384	768	8	8	0.02	0.943
none	0.3	0.0002	512	2560	8	4	0.05	0.943
none	0.2	0.0003	320	1280	10	4	0.05	0.942
isotonic	0.2	0.0002	512	2560	16	8	0.01	0.942
Platt	0.2	0.0003	384	1920	8	4	0.02	0.941
isotonic	0.3	0.0001	256	768	8	4	0.05	0.941
Platt	0.3	0.0001	512	2560	16	8	0.02	0.941
none	0.2	0.0003	512	2560	8	6	0.05	0.940
none	0.25	0.0002	384	1536	12	6	0.02	0.940
isotonic	0.25	0.0003	512	2560	8	4	0.05	0.940
Platt	0.2	0.0002	512	1536	16	6	0.01	0.939
Platt	0.25	0.0002	256	1280	8	4	0.02	0.935

Based on the experimental results, the selected model configuration for validation was a 512-dimensional, 8-head, 8-layer Transformer with feed-forward width 2048, dropout 0.2, learning rate 2×10^{-4}, weight decay 0.05, and Platt calibration for our final training model. This configuration achieved a validation AUC of 0.952, along with a test AUC of 0.895, and test Average Precision (AP) ≈ 0.778. Figure 16 illustrates the comparison of calibration methods (none, isotonic, and Platt scaling) [49, 50], showing the relationship between validation AUC and test AUC. Each point represents a model configuration, highlighting how calibration impacts generalization performance and stability across model variants.

Figure 16. Comparison of calibration methods (none, isotonic, Platt scaling) showing the relationship between validation AUC and test AUC. Each point represents a model configuration, highlighting how calibration impacts generalization performance.

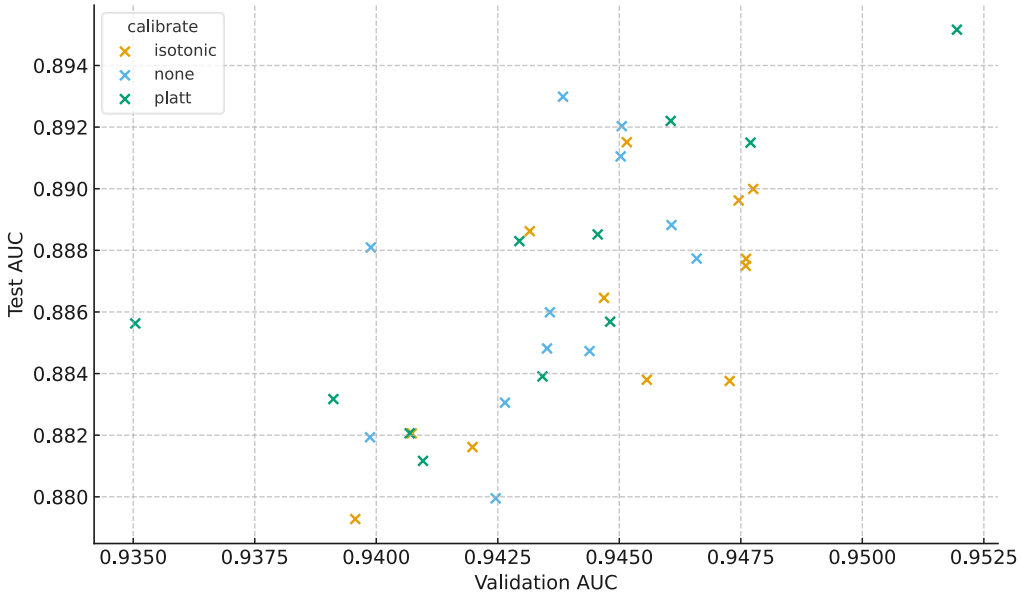


Figure 17 illustrates the comparative performance of the ESMDisPred-DNN and ESMDisPred-LightGBM models across four evaluation metrics: Average Precision Score (APS), Area Under the Curve (AUC), maximum F1 score (F1 max), and the overall average performance. The results clearly demonstrate the superior performance of the ESMDisPred-DNN architecture. Unlike the LightGBM variant, ESMDisPred-DNN employs a CNN–Transformer hybrid model, as described in Section 1.4. This design leads to significant improvements in predictive accuracy and robustness, particularly in detecting subtle or context-dependent disordered regions.

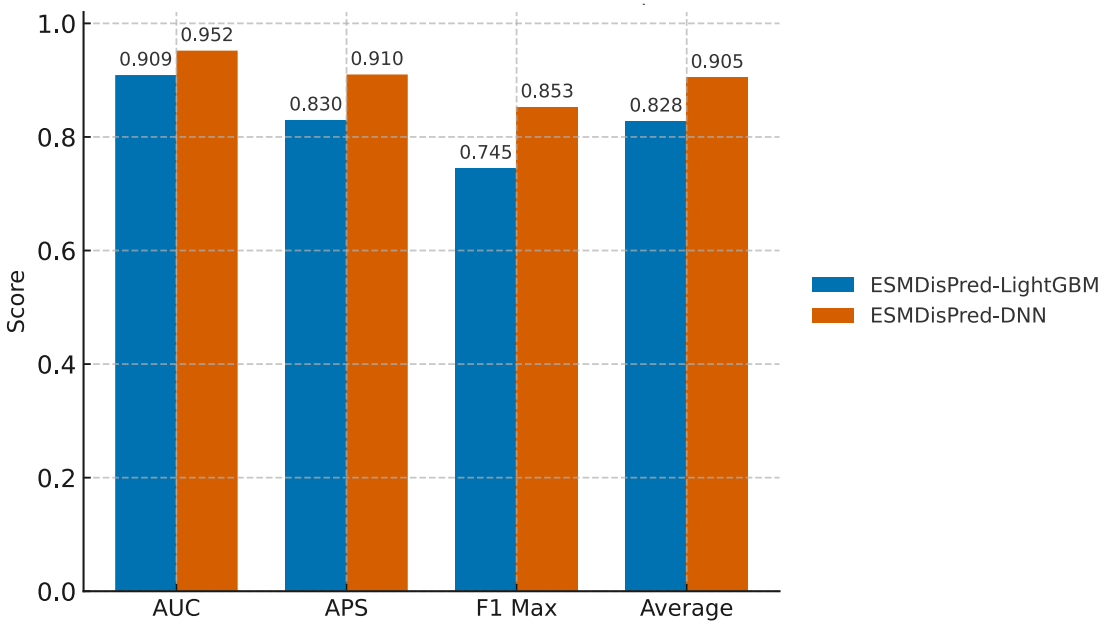


Figure 17. Comparison of predictive performance between ESMDisPred-LightGBM and ESMDisPred-DNN across four evaluation metrics (AUC, APS, F1 Max, and Average).

1.5.2 Comparison with Existing Methods

To assess the performance of ESMDisPred, we compared it with several state-of-the-art disorder prediction tools. Table 3 and Figure 18 provide a comparative evaluation of ESMDisPred models against other top-performing predictors in CAID3 challenges. The top panel of Figure 19 presents the Receiver Operating Characteristic (ROC) curves, where ESMDisPred-DNN achieves the highest AUC (0.8952), surpassing all other methods, including ESMDisPred-2PDB (AUC = 0.8855), ESMDisPred-1 (AUC = 0.8763), and ESMDisPred-2 (AUC = 0.8723). The bottom panel shows the Precision–Recall (PR) curves, where ESMDisPred-DNN again leads with a PR AUC of 0.7778, outperforming ESMDisPred-2PDB (0.7542), ESMDisPred-1 (0.7450), ESMDisPred-2 (0.7431), and other state-of-the-art predictors such as DisoFLAG-IDR (0.7139). These results confirm not only the strong discrimination ability of the ESMDisPred-LightGBM family but also the clear superiority of ESMDisPred-DNN. Importantly, ESMDisPred-DNN is a new model developed after CAID. While it was not submitted to the challenge, its performance on the validation and independent test set demonstrates that it extends the strengths of the feature-based LightGBM models with a more expressive CNN–Transformer hybrid architecture.

Table 3. Performance comparison of protein intrinsically disordered region (IDR) predictors on the test set, reported as AUC, average precision, and maximum F1 (higher is better). ESMDisPred-DNN attains the best overall performance.

Methods	AUC	Average Precision	F1 max
ESMDisPred-DNN	**0.895**	**0.778**	**0.759**
ESMDisPred-2PDB	0.885	0.754	0.749
ESMDisPred-1	0.876	0.745	0.720
ESMDisPred-2	0.872	0.743	0.724
DisoFLAG-IDR	0.868	0.714	0.671
DisorderUnetLM	0.862	0.702	0.647
flDPnn3a	0.860	0.700	0.668
UdonPred-combined	0.854	0.668	0.671
DisPredict3.0	0.850	0.666	0.658
rawMSA	0.850	0.671	0.641

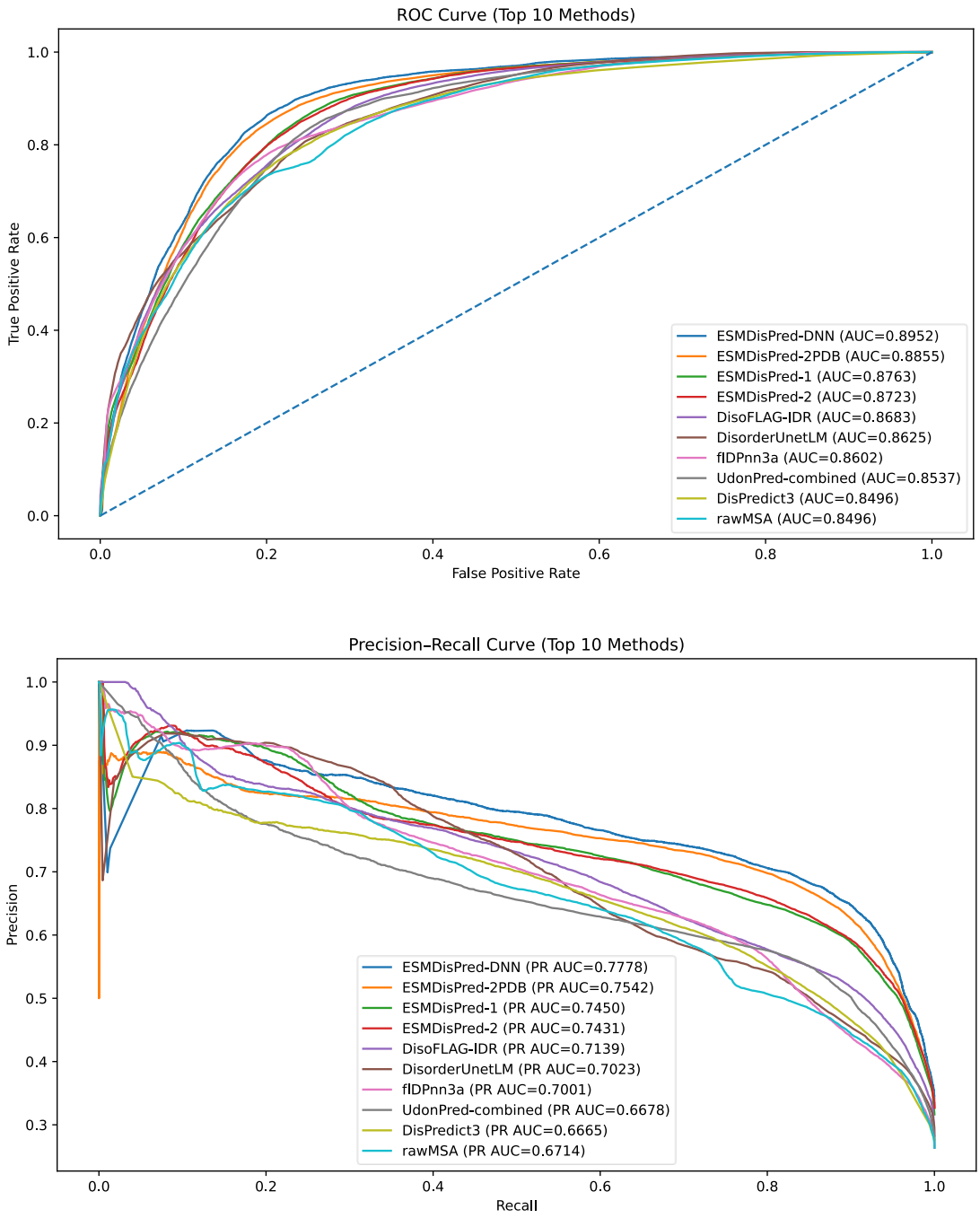


Figure 18. Comparison of ESMDisPred models against other top-performing predictors on the Disorder NOX dataset (CAID2). (Top) Receiver Operating Characteristic (ROC) curves showing model discrimination capabilities, with Area Under Curve (AUC) values for each method. (Bottom) Precision-Recall (PR) curves comparing predictive performance in terms of precision and recall.

To further analyze whether our model's improvement is statistically significant compared to other approaches, we performed pairwise comparisons using the DeLong test. The DeLong test is a

statistical method used to compare the area under the ROC curve (AUC) between two correlated classifiers [51]. It is particularly useful in assessing whether one prediction method performs significantly better than another, based on their true positive and false positive rates. The test assumes that AUCs follow a normal distribution and computes confidence intervals and p-values to determine statistical significance.

We compared the top 10 methods using pairwise DeLong tests for correlated ROC curves and summarized the outcomes in a significance grid (Figure 19). After Benjamini–Hochberg FDR correction ($\alpha = 0.05$), blue cells indicate the row method's AUC is significantly higher than the column method, red indicates significantly lower, and white denotes no significant difference. ESMDisPred-DNN was the top performer, achieving statistically significant gains over every comparator. A second tier included ESMDisPred-2PDB and DisorderUnetLM, which were not significantly different from each other but outperformed most remaining approaches. Within the ESMDisPred family, ESMDisPred-2 exceeded ESMDisPred-1, while DisoFLAG-IDR did not differ significantly from either. flDPnn3a lagged DisorderUnetLM, and UdonPred-combined and DisPredict3 were largely indistinguishable yet underperformed relative to the leaders. The rawMSA baseline was significantly worse than all other methods. Notably, all results except the ESMDisPred variants were collected from the CAID3 challenge submissions. This statistical analysis reinforces the AUC-based rankings observed in the ROC curves and provides rigorous support for performance differences between state-of-the-art disorder predictors.

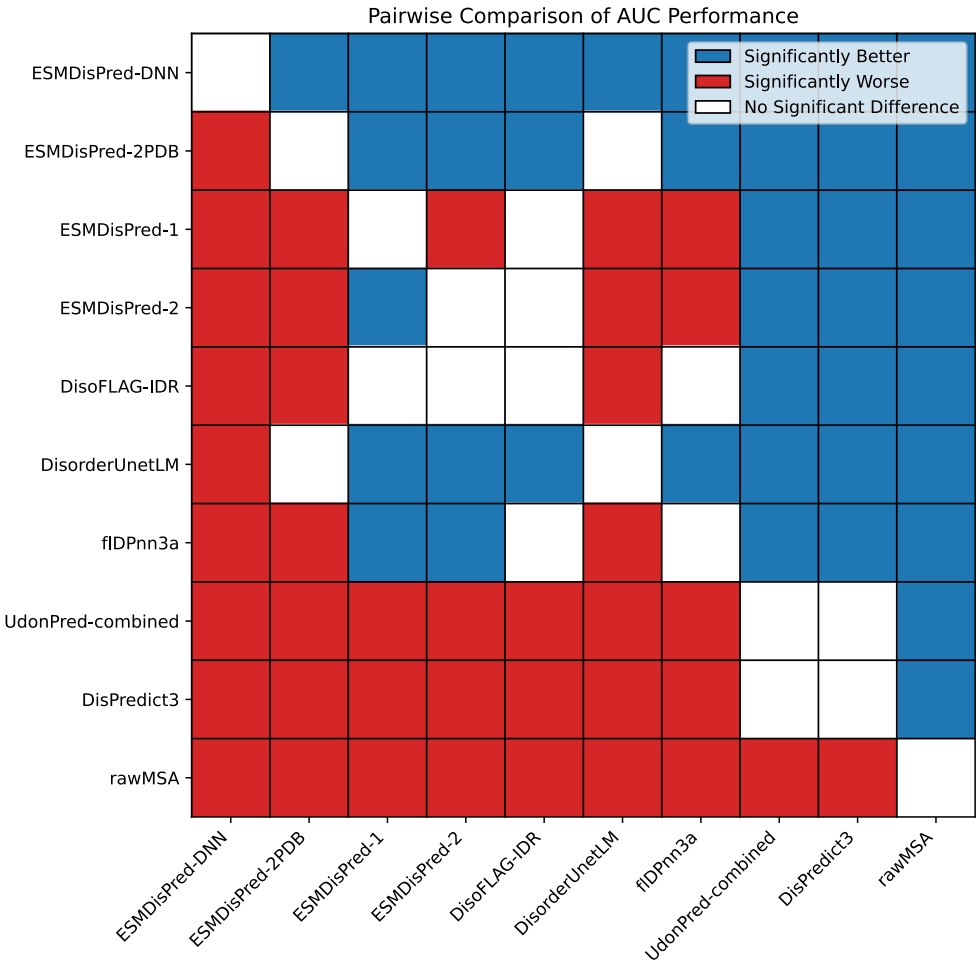


Figure 19. DeLong confidence interval comparison for the top 10 disorder prediction methods from the CAID (Critical Assessment of Intrinsic Disorder) results. Pairwise comparisons are based on ROC AUC scores. Blue cells indicate that the method on the row is statistically significantly better than the method on the column ($p < 0.05$), red indicates significantly worse performance, and white indicates no significant difference. The DeLong test was applied at the 95% confidence level using results from the official CAID challenge.

1.5.3 Computational Complexity

ESMDisPred is designed to deliver high predictive accuracy while maintaining computational efficiency, making it suitable for real-world, high-throughput applications. We ran experiments in containers—Docker locally and Apptainer on LONI HPC (QB4)—with fixed package versions for reproducibility. We set deterministic seeds (NumPy, PyTorch). Training and inference were

performed on LONI HPC nodes with 32-core Intel Xeon Platinum 8358, 512 GB RAM, and NVIDIA A100 with 40 GB memory.

When benchmarked on the CAID3 dataset comprising 232 protein sequences, ESMDisPred achieved an average runtime of ~95 seconds per protein. This efficiency is further enhanced when the model is deployed on GPU hardware, substantially reducing inference time and enabling large-scale proteomic analyses. As shown in Figure 20, ESMDisPred achieves a strong balance between runtime and AUC performance, positioning itself among the top-performing predictors. Compared to other state-of-the-art models such as rawMSA or flDPnn3b, which require significantly longer runtimes, ESMDisPred maintains competitive accuracy with substantially faster execution. This favorable trade-off between speed and predictive quality underscores the utility of ESMDisPred in proteome-wide disorder prediction pipelines, particularly in settings where both accuracy and scalability are critical.

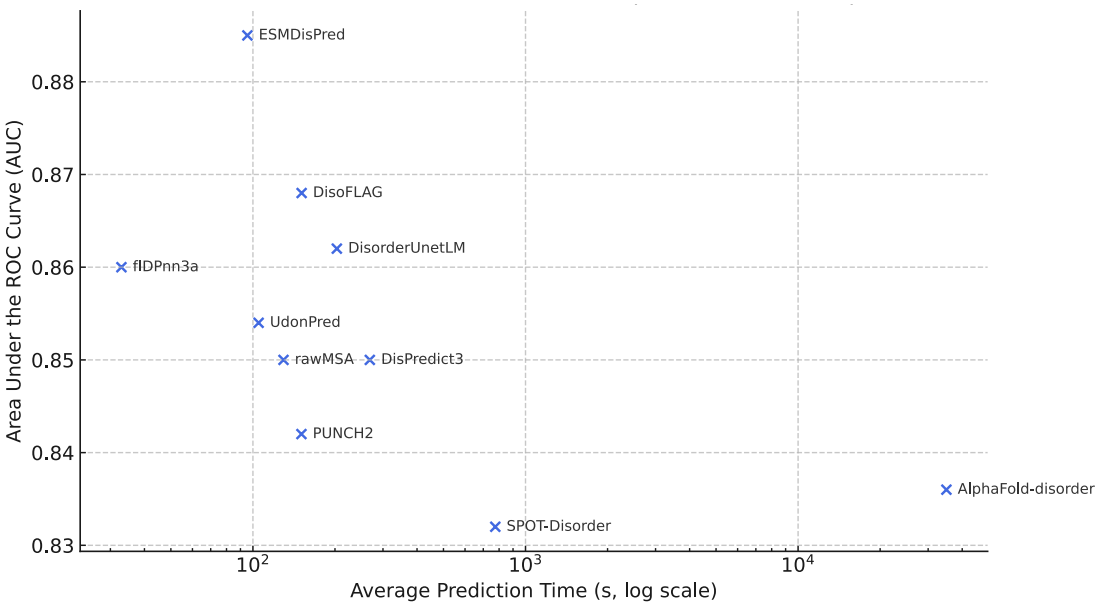


Figure 20. This scatter plot compares the average prediction time (log scale, in seconds) and area under the ROC curve (AUC) for ten leading protein disorder predictors evaluated in the CAID3 challenge. ESMDisPred achieves the highest AUC with relatively low runtime, indicating strong performance in both accuracy and efficiency. In contrast, AlphaFold-disorder demonstrates high computational cost (35,000 s) with moderate AUC. Labels next to each data point identify the predictors.

1.6 Conclusions

In this work, we introduce ESMDisPred, a structure-aware framework for predicting intrinsically disordered regions in proteins. It integrates fine-tuned embeddings from ESM2 with domain-specific features such as terminal annotations, window-based representations, and structural filters. By employing a CNN-Transformer hybrid model, ESMDisPred effectively captures both local sequence motifs and long-range dependencies. This leads to consistent improvements over traditional machine learning algorithms. Empirical results on benchmark datasets, particularly the CAID3 Disorder NOX set, demonstrate substantial gains in APS, AUC, and F1 max. This underscores the robustness and reliability of the approach in identifying both fully and partially disordered regions. Post-processing strategies, including Platt scaling, Gaussian smoothing, and hyperparameter optimization, further refine predictions. They improve stability in noisy or ambiguous cases. Beyond performance, the study highlights the value of combining transformer-based protein language models with biologically relevant structural context, which significantly enhances predictive accuracy and generalization. Future work could leverage AlphaFold-predicted structures, where available, as lightweight structural cues—using per-residue confidence (pLDDT) and predicted aligned error (PAE) to flag flexible segments. However, important challenges remain, especially in modeling the dynamic nature of disorder, in which residues may transition between ordered and disordered states under different conditions or when structural confidence is low. Furthermore, the interpretability of ML models can be utilized through attention visualization and feature attribution, which will be critical for uncovering sequence motifs that drive disorder. Overall, this work highlights the potential of ESMDisPred as a reliable and practical tool for advancing our understanding of protein disorder.

Code availability

The ESMDisPred webserver is available at https://bmll.cs.uno.edu. The code and data related to the development of ESMDisPred can be found here https://github.com/wasicse/ESMDisPred

Conflict of Interest

The authors declare no conflicts of interest.

Funding and Acknowledgments

Research reported in this publication was supported by an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under grant number P20GM103424-21. We acknowledge support from the Louisiana Optical Network Infrastructure (LONI) for access to high-performance computing resources.

References

[1] M. C. Aspromonte *et al.*, "DisProt in 2024: improving function annotation of intrinsically disordered proteins," *Nucleic Acids Res,* vol. 52, no. D1, pp. D434-D441, Jan 5 2024, doi: 10.1093/nar/gkad928.

[2] M. Kumar *et al.*, "The Eukaryotic Linear Motif resource: 2022 release," *Nucleic Acids Res,* vol. 50, no. D1, pp. D497-D508, Jan 7 2022, doi: 10.1093/nar/gkab975.

[3] A. Rives, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences,* vol. 118, no. 15, 2021.

[4] A. Elnaggar *et al.*, "ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning," (in eng), *IEEE Trans Pattern Anal Mach Intell,* vol. 44, no. 10, pp. 7112-7127, Oct 2022, doi: 10.1109/TPAMI.2021.3095381.

[5] A. Del Conte *et al.*, "CAID prediction portal: a comprehensive service for predicting intrinsic disorder and binding regions in proteins," *Nucleic Acids Res,* vol. 51, no. W1, pp. W62-W69, Jul 5 2023, doi: 10.1093/nar/gkad430.

[6] C. Mirabello and B. Wallner, "rawMSA: End-to-end Deep Learning using raw Multiple Sequence Alignments," (in en), *PLOS ONE,* vol. 14, no. 8, p. e0220182, 2019/08/15/ 2019, doi: 10.1371/journal.pone.0220182.

[7] G. Hu *et al.*, "flDPnn: Accurate intrinsic disorder prediction with putative propensities of disorder functions," (in en), *Nat Commun,* vol. 12, no. 1, p. 4438, Jul 21 2021, doi: 10.1038/s41467-021-24773-7.

[8] J. Hanson, K. K. Paliwal, T. Litfin, and Y. Zhou, "SPOT-Disorder2: Improved Protein Intrinsic Disorder Prediction by Ensembled Deep Learning," *Genomics Proteomics Bioinformatics,* vol. 17, no. 6, pp. 645-656, Dec 2019, doi: 10.1016/j.gpb.2019.01.004.

[9] Y. Song, Q. Yuan, S. Chen, K. Chen, Y. Zhou, and Y. Yang, "Fast and accurate protein intrinsic disorder prediction by using a pretrained language model," *Brief Bioinform,* vol. 24, no. 4, p. bbad173, Jul 20 2023, doi: 10.1093/bib/bbad173.

[10] G. Erdos, M. Pajkos, and Z. Dosztanyi, "IUPred3: prediction of protein disorder enhanced with unambiguous experimental annotation and visualization of evolutionary conservation," *Nucleic Acids Res,* vol. 49, no. W1, pp. W297-W303, Jul 2 2021, doi: 10.1093/nar/gkab408.

[11] G. Erdos and Z. Dosztanyi, "AIUPred: combining energy estimation with deep learning for the enhanced prediction of protein disorder," *Nucleic Acids Res,* vol. 52, no. W1, pp. W176-W181, Jul 5 2024, doi: 10.1093/nar/gkae385.

[12] X. Zhang, R. M. Blumenthal, and X. Cheng, "DNA-binding proteins from MBD through ZF to BEN: recognition of cytosine methylation status by one arginine with two conformations," *Nucleic Acids Res,* vol. 52, no. 19, pp. 11442-11454, Oct 28 2024, doi: 10.1093/nar/gkae832.

[13] C. J. Wilson, W. Y. Choy, and M. Karttunen, "AlphaFold2: A Role for Disordered Protein/Region Prediction?," *Int J Mol Sci,* vol. 23, no. 9, p. 4591, Apr 21 2022, doi: 10.3390/ijms23094591.

[14] D. Meng and G. Pollastri, "PUNCH2: Explore the strategy for intrinsically disordered protein predictor," *PLoS One,* vol. 20, no. 3, p. e0319208, 2025, doi: 10.1371/journal.pone.0319208.

[15] N. Malhis, "Probabilistic Annotations of Protein Sequences for Intrinsically Disordered Features (IPA)," *bioRxiv*, 2025, doi: 10.1101/2024.12.18.629275.
[16] F. Zhang, B. Zhao, W. Shi, M. Li, and L. Kurgan, "DeepDISOBind: accurate prediction of RNA-, DNA- and protein-binding intrinsically disordered residues with deep multi-task learning," *Brief Bioinform*, vol. 23, no. 1, p. bbab521, Jan 17 2022, doi: 10.1093/bib/bbab521.
[17] M. Kotowski and J. Wojciechowski, "DisorderUnetLM: Lightweight U-Net Architecture with Language Model Embeddings for Protein Disorder Prediction," *Computers in Biology and Medicine*, 2024.
[18] S. Schlensok and H. Gohlke, "UdonPred: Protein Disorder Prediction from NMR-Derived TriZOD Scores and Deep Learning," *bioRxiv*, 2023.
[19] T. Zhang and L. Kurgan, "EBIND: Prediction of Binding-Prone Intrinsically Disordered Regions Using Deep Embeddings," *bioRxiv*, 2024.
[20] M. W. U. Kabir and M. T. Hoque, "DisPredict3.0: Prediction of intrinsically disordered regions/proteins using protein language model," *Applied Mathematics and Computation*, vol. 472, p. 128630, 2024/07/01/ 2024, doi: 10.1016/j.amc.2024.128630.
[21] M. Mehdiabadi, A. Del Conte, M. V. Nugnes, M. C. Aspromonte, S. C. E. Tosatto, and D. Piovesan, "Critical Assessment of Protein Intrinsic Disorder Round 3 - Predicting Disorder in the Era of Protein Language Models," *Proteins*, vol. 93, no. 8, p. e70045, Aug 26 2025, doi: 10.1002/prot.70045.
[22] M. Steinegger and J. Soding, "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets," *Nat Biotechnol*, vol. 35, no. 11, pp. 1026-1028, Nov 2017, doi: 10.1038/nbt.3988.
[23] "CAID Challenge Results," *CAID*, 2025/07/29 2025. [Online]. Available: https://caid.idpcentral.org/challenge/results.
[24] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55-67, 1970, doi: 10.1080/00401706.1970.10488634.
[25] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, pp. 175-185, 1992.
[26] T. K. Ho, "Random decision forests," presented at the Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, Montreal, Que., Canada, 1995.
[27] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3-42, 2006.
[28] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," presented at the Proceedings of the Second European Conference on Computational Learning Theory, 1995.
[29] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system.," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (KDD '16. New York, NY, USA: ACM, 2016, pp. 785-794.
[30] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.
[31] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," *arXiv preprint* 2018.
[32] C. Guo and F. Berkhahn, "Entity Embeddings of Categorical Variables," *CoRR*, vol. abs/1604.06737, 2016. [Online]. Available: https://arxiv.org/abs/1604.06737.
[33] X. Huang, A. Khetan, M. Cvitkovic, and Z. S. Karnin, "TabTransformer: Tabular Data Modeling Using Contextual Embeddings," *CoRR*, vol. abs/2012.06678, 2020. [Online]. Available: https://arxiv.org/abs/2012.06678.
[34] S. O. Arik and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning," (in en), *arXiv*, 2019/08/20/ 2019. [Online]. Available: https://arxiv.org/abs/1908.07442v5.

[35] J. Chen, K. Liao, Y. Wan, D. Z. Chen, and J. Wu, "DANets: Deep Abstract Networks for Tabular Data Classification and Regression," arXiv:2112.02962, 2022.
[36] M. Joseph and H. Raj, "GANDALF: Gated Adaptive Network for Deep Automated Learning of Features," arXiv:2207.08548, 2024/01/10, 2024.
[37] B. Faezov and R. L. Dunbrack Jr, "PDBrenum: A webserver and program providing Protein Data Bank files renumbered according to their UniProt sequences," *Plos one*, vol. 16, no. 7, p. e0253411, 2021.
[38] R. Rao *et al.*, "MSA Transformer," *bioRxiv*, 2021, doi: 10.1101/2021.02.12.430858.
[39] Z. Lin *et al.*, "Evolutionary-scale prediction of atomic-level protein structure with a language model," (in eng), *Science*, vol. 379, no. 6637, pp. 1123-1130, Mar 17 2023, doi: 10.1126/science.ade2574.
[40] J. Jumper, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, pp. 583-589, 2021.
[41] M. Varadi *et al.*, "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models," *Nucleic Acids Res*, vol. 50, no. D1, pp. D439-D444, Jan 7 2022, doi: 10.1093/nar/gkab1061.
[42] D.-T. H. Chang, H.-B. Shen, and K.-C. Chou, "Predicting the protein–protein interactions using primary sequence information," *BMC Bioinformatics*, vol. 11, no. Suppl 1, p. S3, 2009, doi: 10.1186/1471-2105-11-S1-S3.
[43] M. Li, Y. Zhang, and T. Chen, "A Novel Approach for Predicting Disordered Regions in Proteins Using a Sliding Window," *Computational and Structural Biotechnology Journal*, vol. 10, no. 17, pp. 30-39, 2014, doi: 10.1016/j.csbj.2014.02.005.
[44] Y. Jiang, R. Zhao, W. Zhang, and X. Liu, "Computational methods for protein subcellular localization prediction," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 2186-2198, 2021, doi: 10.1016/j.csbj.2021.03.040.
[45] M. W. Kabir, D. M. Alawad, A. Mishra, and M. T. Hoque, "TAFPred: Torsion Angle Fluctuations Prediction from Protein Sequences," *Biology*, vol. 12, no. 7, doi: 10.3390/biology12071020.
[46] A. Mishra, M. W. U. Kabir, and M. T. Hoque, "diSBPred: A machine learning based approach for disulfide bond prediction," *Comput Biol Chem*, vol. 91, p. 107436, Apr 2021, doi: 10.1016/j.compbiolchem.2021.107436.
[47] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.
[48] H. Fang, J.-U. Lee, N. S. Moosavi, and I. Gurevych, "Transformers with Learnable Activation Functions," in *Findings of the Association for Computational Linguistics: EACL 2023*, 2023, pp. 2382-2398, doi: 10.18653/v1/2023.findings-eacl.181.
[49] B. Zadrozny and C. Elkan, "Transforming Classifier Scores into Accurate Multiclass Probability Estimates," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, New York, NY, USA, 2002: ACM, pp. 694-699, doi: 10.1145/775047.775151. [Online]. Available: https://dl.acm.org/doi/10.1145/775047.775151
[50] A. Niculescu-Mizil and R. Caruana, "Predicting Good Probabilities with Supervised Learning," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, New York, NY, USA, 2005: ACM, pp. 625-632, doi: 10.1145/1102351.1102430. [Online]. Available: https://dl.acm.org/doi/10.1145/1102351.1102430
[51] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach," *Biometrics*, vol. 44, no. 3, pp. 837-45, Sep 1988, doi: 10.2307/2531595.