LoRADisPred: Parameter-Efficient Adaptation of Protein Language Models for Intrinsically Disordered Region Prediction

Md Wasi Ul Kabir *mkabir3@uno.edu*
Department of Computer Science
University of New Orleans, New Orleans, LA, USA

Md Tamjidul Hoque* *thoque@uno.edu*
Department of Computer Science
University of New Orleans, New Orleans, LA, USA

Abstract

Intrinsically disordered regions (IDRs) lack a stable three-dimensional structure yet are central to cellular signaling, regulation, and disease. Protein language models (PLMs) such as ESM-2 learn rich per-residue representations, but adapting these large models to downstream tasks by full fine-tuning is costly. We present LoRADisPred, a per-residue disorder predictor that adapts a frozen 650M-parameter ESM-2 backbone using Low-Rank Adaptation (LoRA), updating only 1.85% of the total model parameters. We augment the adapted backbone with a CRF prediction head using token-weighted layer pooling and class-weighted cross-entropy to handle the imbalance between ordered and disordered residues. We train on a homology-filtered DisProt-derived corpus, with training sequences filtered to remove homologs of both validation and test proteins at $\geq 25\%$ sequence identity and $\geq 80\%$ coverage, and evaluate on the CAID3-NOX benchmark. LoRADisPred achieves 0.884 AUC, 0.768 average precision, and 0.743 $F1_{max}$ on CAID3-NOX. It matches ESMDisPred-2PDB to within 0.001 AUC while surpassing it in average precision. These results demonstrate that parameter-efficient adaptation can closely approach the accuracy of full fine-tuning for disorder prediction.

1 Introduction

Proteins carry out most cellular functions, and their behavior is shaped by their structural organization. A substantial fraction of eukaryotic proteins, however, contains intrinsically disordered regions (IDRs) that do not fold into a single stable conformation. These regions are not functionless: they mediate molecular recognition, signaling, post-translational regulation, and the formation of biomolecular condensates, and their dysregulation is linked to cancer, neurodegeneration, and other diseases. Because IDRs by definition lack a fixed structure, the classical sequence-to-structure pipeline does not apply, and disorder must instead be inferred computationally from sequence. A long line of predictors has been developed for this task, ranging from biophysics-based scores such as IUPred3 (Erdős et al., 2021) to machine-learning methods such as flDPnn (Hu et al., 2021), with the Critical Assessment of protein Intrinsic Disorder prediction (CAID) providing community benchmarks (Necci et al., 2021; Del Conte et al., 2023).

Self-supervised protein language models (PLMs) such as the ESM family (Rives et al., 2021; Lin et al., 2023) have recently become a strong source of per-residue features, and embedding-based disorder predictors such as SETH (Ilzhöfer et al., 2022) and ADOPT (Redl et al., 2023) demonstrate their value for this task. Exploiting the largest PLMs by full fine-tuning is nonetheless expensive: updating every weight of a model with hundreds of millions of parameters requires substantial GPU memory and compute, which limits accessibility.

We address this with LoRADisPred, which adapts a frozen ESM-2 backbone using Low-Rank Adaptation (LoRA) (Hu et al., 2022). LoRA injects trainable low-rank matrices into the attention projections and leaves the pretrained

*Corresponding Author

weights unchanged, reducing the number of trainable parameters by roughly two orders of magnitude relative to full fine-tuning. Our contributions are: (i) a parameter-efficient per-residue disorder predictor built on a 650M ESM-2 model with LoRA; (ii) a per-residue CRF head with token-weighted layer pooling and class-weighted cross-entropy loss; (iii) probability calibration with temperature scaling and a validation-tuned threshold; and (iv) an evaluation on the CAID3 benchmark under a strict homology-separation protocol, with CAID2 used for validation, and released code and data scripts.

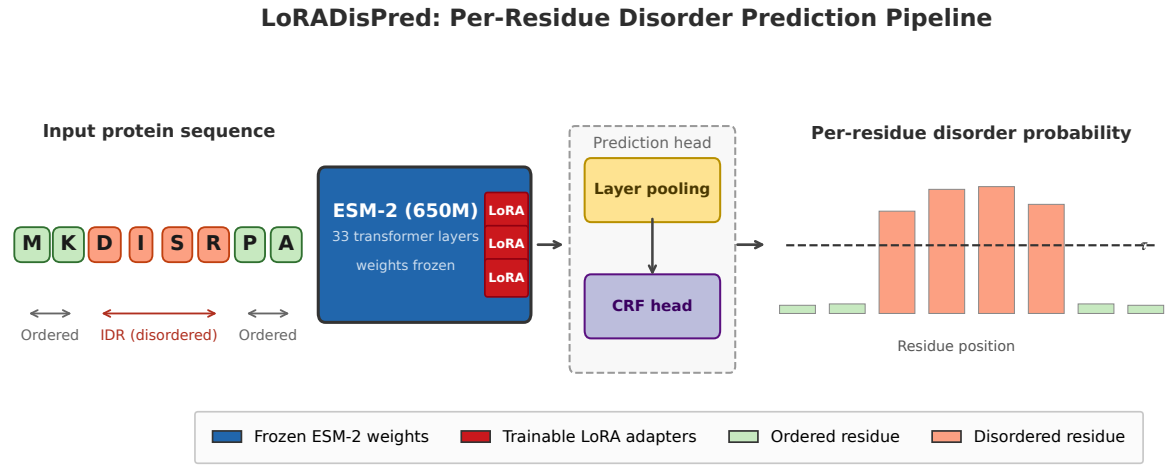

Figure 1: Overview of the LoRADisPred prediction pipeline. A protein sequence is passed through a frozen ESM-2 backbone (650M parameters) equipped with LoRA adapters injected at each attention layer, with only 1.85% of the total model parameters trained. Per-layer hidden states are aggregated by a token-weighted pooling module and passed to a CRF prediction head, producing per-residue disorder probabilities. A validation-selected threshold τ binarizes the output into ordered and disordered predictions.

More broadly, LoRADisPred follows a transfer-learning framework in which general-purpose protein representations are adapted to specialized residue-level prediction tasks. This paradigm is summarized in Figure 2.

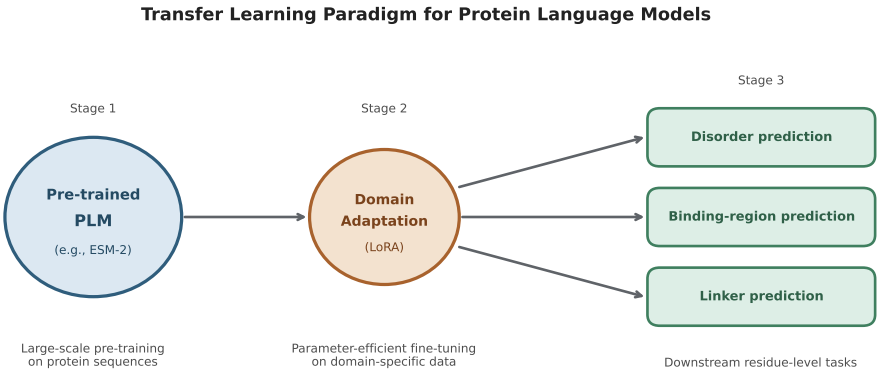

Figure 2: Transfer learning paradigm for protein language models. A large PLM pre-trained on millions of protein sequences is adapted to a target domain via parameter-efficient fine-tuning (LoRA), enabling application to multiple downstream per-residue prediction tasks. Disorder prediction is the focus of this work; binding-region and linker prediction represent natural future extensions.

2 Related Work

Early predictors relied on amino-acid propensities and biophysical reasoning; IUPred3 (Erdős et al., 2021) is a widely used representative. Supervised learning methods such as flDPnn (Hu et al., 2021) improved accuracy and were among the strongest entries in CAID (Necci et al., 2021). The CAID rounds standardized evaluation on curated DisProt-derived test sets and distinguish the NOX (disorder defined by the absence of experimental order) and PDB (disorder relative to resolved structure) reference definitions (Necci et al., 2021; Del Conte et al., 2023).

PLMs trained on large sequence databases (Rives et al., 2021; Lin et al., 2023) produce contextual per-residue embeddings that transfer to many tasks. SETH (Ilzhöfer et al., 2022) regresses continuous disorder from ESM embeddings, and ADOPT (Redl et al., 2023) predicts disorder with transformer representations; both typically use frozen embeddings with a trained head. LoRADisPred differs in that it *adapts* the backbone with LoRA rather than freezing it entirely, while keeping the trainable footprint small. LoRA (Hu et al., 2022) adds low-rank update matrices to selected weight matrices and trains only those, keeping the base model frozen. We use the PEFT implementation (Mangrulkar et al., 2022) within the Hugging Face Transformers ecosystem (Wolf et al., 2020). We additionally draw on conditional random fields (Lafferty et al., 2001) for structured per-residue output and temperature scaling (Guo et al., 2017) for calibration.

Among the strongest current methods on CAID3, ESMDisPred (Kabir et al., 2026) is an ESM-2–based CNN-Transformer architecture for intrinsically disordered protein prediction; variants differ in training reference definition (NOX vs. PDB) and head architecture, and the method also draws on DisPredict3.0 features as part of its input. DisPredict3.0 (Kabir & Hoque, 2024) combines ESM embeddings with flDPnn features in a LightGBM classifier and serves as a strong baseline on CAID benchmarks. Both are included in the CAID3 comparison (Table 2).

3 Materials and Methods

3.1 Datasets and homology separation

Training data were derived from the DisProt database (Aspromonte et al., 2024). We parsed the DisProt release into per-protein sequences and residue-level labels, marking residues inside annotated disorder regions as positive (1) and all other residues as negative (0); residues without a confident annotation are assigned an ignore label and excluded from the loss. This parsing yielded 3,199 sequences comprising 1,872,726 residues.

To prevent train/test leakage, we removed any training sequence homologous to the evaluation proteins. We pooled both CAID3 test sets (CAID3-NOX, CAID3-PDB) and the CAID2 validation sets (CAID2-NOX, CAID2-PDB) and searched the training sequences against this pool with MMseqs2 (Steinegger & Söding, 2017), removing every training sequence that aligned to any evaluation sequence at $\geq 25\%$ sequence identity with $\geq 80\%$ coverage. This step reduced the DisProt-derived set to 2,798 sequences (1,657,922 residues), which formed the final training set.

The CAID2 benchmark (CAID2-NOX: 210 sequences, 160,802 residues; CAID2-PDB: 348 sequences, 130,877 residues) was used as the held-out validation set for model selection and calibration. Final evaluation used the CAID3 benchmark under both reference definitions: CAID3-NOX (204 sequences) and CAID3-PDB (319 sequences). Sequences longer than the maximum training length were truncated during tokenization rather than discarded (Section 3.4). Table 1 summarizes the data.

3.2 Model architecture

LoRADisPred consists of three main components: a frozen ESM-2 backbone, LoRA adapters that provide task-specific adaptation, and a residue-level prediction head (Figure 3). The backbone is ESM-2 (`esm2_t33_650M_UR50D`; 33 transformer layers, hidden size 1280) (Lin et al., 2023). The pretrained ESM-2 weights are kept frozen during training.

LoRA (Hu et al., 2022) is applied to the self-attention projection matrices in each ESM-2 layer, including the query, key, value, and output/dense projections. Instead of updating an original ESM-2 weight matrix, LoRA learns a

Table 1: Dataset composition. CAID2 serves as the validation set for model selection and calibration; CAID3 is the held-out test set.

Set	Sequences	Residues
DisProt parsed (raw)	3,199	1,872,726
DisProt after homology filtering (training set)	2,798	1,657,922
CAID2-NOX (validation)	210	160,802
CAID2-PDB (validation)	348	130,877
CAID3-NOX (test)	204	201,036
CAID3-PDB (test)	319	320,844

small low-rank update:

$$W_{\text{adapted}} = W_{\text{frozen}} + \Delta W, \qquad \Delta W = \frac{\alpha}{r} BA. \tag{1}$$

Here, W_{frozen} is the original pretrained projection matrix, ΔW is the trainable LoRA update, r is the LoRA rank, α is the scaling factor, and A and B are the two small trainable matrices. This equation is included only to clarify the adaptation mechanism: the large pretrained matrix is not changed, and only the low-rank update is learned. For the selected model, we use rank $r = 2$, scaling $\alpha = 2$, and LoRA dropout 0.40. For a 1280-by-1280 attention projection, this adds 5,120 trainable LoRA parameters instead of updating 1,638,400 original parameters, reducing the trainable parameters for that projection by roughly 99.7%.

ESM-2 produces a residue representation at every transformer layer. Rather than using only the final layer, LoRADisPred combines information from all layers via token-weighted pooling:

$$\bar{h}_t = \sum_{\ell=0}^{L} a_{t,\ell} h_{t,\ell}, \qquad \sum_{\ell=0}^{L} a_{t,\ell} = 1. \tag{2}$$

In this notation, $h_{t,\ell}$ is the representation of residue t from layer ℓ, $a_{t,\ell}$ is the learned weight assigned to that layer for that residue, and $\bar{h}_t$ is the final pooled representation used for prediction. The constraint that the weights sum to one makes the pooled representation a weighted average of layer-specific residue features. This allows different residues to use information from different depths of ESM-2.

The pooled residue representation is passed to a linear-chain CRF head (Lafferty et al., 2001) that outputs an ordered/disordered probability for each residue. A CRF is used because adjacent residues in proteins are not independent; ordered and disordered residues often occur in contiguous segments.

3.3 Loss functions and class imbalance

Per-residue disorder prediction is class-imbalanced because ordered residues are more common than disordered residues. During training, residues without confident annotations are ignored and do not contribute to the loss. For annotated residues, the default objective is class-weighted cross-entropy:

$$\mathcal{L}_{\text{WCE}} = -\frac{1}{M} \sum_{t=1}^{M} w_{y_t} \log p_{t,y_t}. \tag{3}$$

Here, M is the number of annotated residues in the batch, y_t is the true label of residue t, p_{t,y_t} is the model probability assigned to the correct class, and w_{y_t} is the class weight. The disordered class receives a larger weight when it is under-represented, which discourages the model from simply favoring the majority ordered class.

The CRF head is used, the model optimizes the likelihood of the full residue-label sequence rather than treating every residue independently. Instead of giving each residue a completely independent decision, the CRF combines residue-level emission scores with learned transition scores between adjacent labels:

$$S(\mathbf{y}) = \sum_{t=1}^{T} e_{t,y_t} + \sum_{t=2}^{T} \psi_{y_{t-1},y_t}. \tag{4}$$

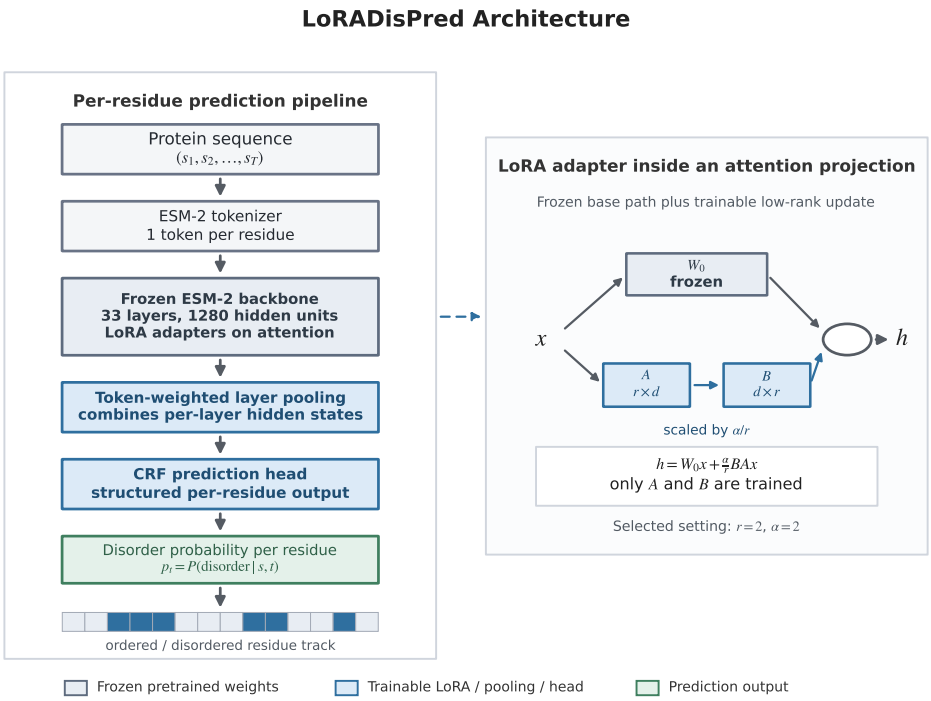

Figure 3: LoRADisPred architecture: a frozen ESM-2 backbone with LoRA adapters on the attention projections; per-layer hidden states combined by the token-weighted pooling module; and a CRF head producing per-residue disorder probabilities.

Here, $S(\mathbf{y})$ is the score of one possible label sequence, e_{t,y_t} is the neural-network score for assigning label y_t to residue t, and ψ_{y_{t-1},y_t} is the learned score for transitioning from the previous residue label to the current residue label. This encourages biologically plausible predictions, such as contiguous ordered or disordered segments, while still allowing transitions when supported by the sequence evidence.

3.4 Training configuration

We fine-tune with the Hugging Face Trainer (Wolf et al., 2020). Training sequences are tokenized without special tokens so token positions correspond one-to-one with residues. Based on the CAID2-NOX validation hyperparameter sweep, the selected configuration used a maximum training length of 1,500 residues and a maximum test length of 5,000 residues. We trained for 15 epochs with batch size 3, LoRA rank $r = 2$, LoRA scaling $\alpha = 2$, LoRA dropout 0.4, and learning rate 2×10^{-4}. The best checkpoint was selected by CAID2-NOX validation ROC-AUC.

3.5 Calibration

Because raw softmax outputs are not always well calibrated, we apply temperature scaling (Guo et al., 2017) on the validation set. Temperature scaling rescales the logits before converting them to probabilities:

$$\hat{p}_{t,c} = \text{softmax}(z_{t,c}/T), \tag{5}$$

where $z_{t,c}$ is the logit for residue t and class c, T is the learned temperature, and $\hat{p}_{t,c}$ is the calibrated probability. Temperature scaling changes the confidence values but does not change the ranking of residues. Therefore, ranking-based metrics such as AUC and average precision remain unchanged. The temperature value is selected on the validation set by minimizing negative log-likelihood over a grid from 0.5 to 3.0.

The binary decision threshold used to convert disorder probabilities into ordered/disordered labels is also selected only on the validation set. Specifically, we choose the threshold that maximizes validation F1 and apply that same threshold unchanged to the test sets. The threshold is never tuned on test data.

3.6 Evaluation metrics

All metrics are computed over annotated residues only. We report ROC-AUC, average precision (AP), and maximum F1 score ($F1_{max}$). Let p_t be the predicted disorder probability for residue t, and let the annotated residues be divided into disordered residues $\mathscr{P}$ and ordered residues $\mathscr{N}$.

ROC-AUC measures whether disordered residues tend to receive higher disorder probabilities than ordered residues. One intuitive way to write it is:

$$\mathrm{AUC} = \frac{1}{|\mathscr{P}||\mathscr{N}|} \sum_{i \in \mathscr{P}} \sum_{j \in \mathscr{N}} \mathbb{1}[p_i > p_j]. \tag{6}$$

In words, AUC is the fraction of all disordered–ordered residue pairs for which the disordered residue receives the higher predicted disorder score. A value of 1.0 indicates perfect ranking, while 0.5 corresponds to random ranking.

Average precision summarizes the precision–recall curve and is especially useful for imbalanced residue-level prediction. After sorting residues by predicted disorder probability, AP can be written as:

$$\mathrm{AP} = \sum_k (R_k - R_{k-1})\, P_k. \tag{7}$$

Here, P_k and R_k are the precision and recall after applying the kth threshold along the ranked list. AP is high when the model retrieves many true disordered residues early in the ranked list while maintaining high precision.

F1 combines precision and recall at a particular threshold:

$$\mathrm{F1} = \frac{2\,\mathrm{Precision}\,\mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{8}$$

Because different predictors may use different probability scales, we report $F1_{max}$, the maximum F1 obtained by sweeping possible thresholds on the predicted disorder probabilities. Higher values indicate better performance for all three metrics.

4 Results

4.1 Performance on CAID3-NOX

Table 2 compares LoRADisPred against leading intrinsic disorder predictors on the CAID3-NOX benchmark under the official blind evaluation protocol. LoRADisPred achieves 0.884 AUC, 0.768 Average Precision, and 0.743 $F1_{max}$ on the CAID3-NOX test set when trained on the homology-filtered DisProt-derived training set, ranking among the top methods while updating only 1.85% of the total model parameters via LoRA adaptation. It matches ESMDisPred-2PDB to within 0.001 AUC while surpassing it in average precision.

4.2 Parameter efficiency

LoRADisPred trains only the LoRA update matrices, the pooling module, and the small prediction head, leaving the ESM-2 backbone frozen. Adapting the four attention projections at rank $r = 2$ across 33 layers introduces on the order of a few million LoRA parameters; together with the head this is less than 2% of the full model, so full fine-tuning would update roughly two orders of magnitude more parameters. Table 3 reports the exact counts logged during training.

Figure 4 further illustrates the reduction in trainable parameters relative to full fine-tuning.

Table 2: Performance comparison of leading intrinsic disorder predictors on the CAID3 benchmark. ROC-AUC, Average Precision, and maximum F1 score ($F1_{max}$) are reported under the official CAID3 blind evaluation protocol. Methods are ordered by ROC-AUC. The highest value in each column is in **bold**.

Method	AUC	Avg. Precision	$F1_{max}$
ESMDisPred-2PDB	**0.885**	0.754	**0.749**
LoRADisPred (ours)	0.884	**0.768**	0.743
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

Table 3: Parameter efficiency of LoRADisPred relative to full fine-tuning of ESM-2 (`esm2_t33_650M_UR50D`). Values from the training log line "Trainable params: X / Y (Z%)".

	Trainable params	% of total
Full fine-tuning (all weights)	663,324,225	100%
LoRADisPred (LoRA + head)	12,283,564	1.85%

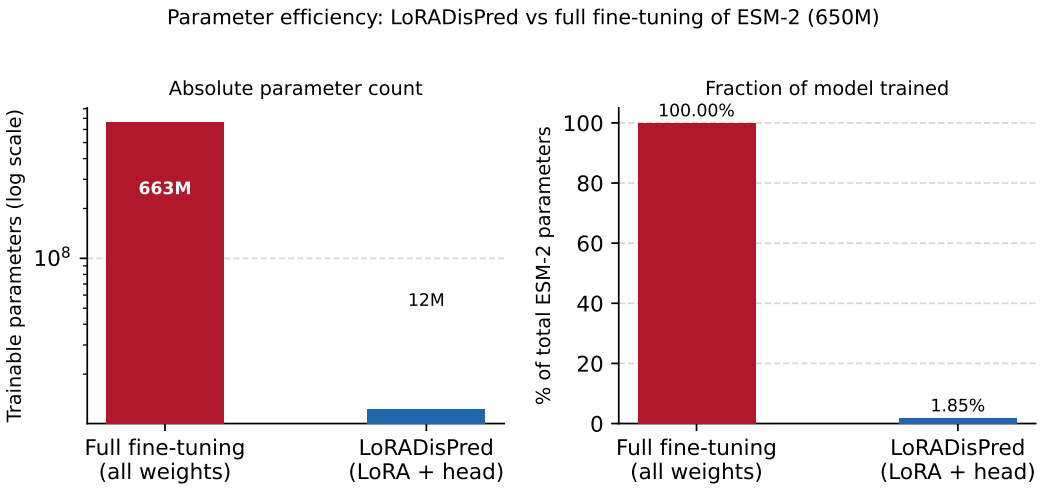

Figure 4: Parameter efficiency of LoRADisPred versus full fine-tuning of ESM-2 (650M parameters). Left: absolute trainable parameter count on a log scale. Right: trainable parameters as a percentage of the total model. LoRADisPred updates only 1.85% of the total model parameters, including LoRA adapters, the pooling module, and the prediction head.

4.3 Hyperparameter sweep

We evaluated 1,011 MLflow-tracked runs, of which 1,003 completed successfully. The sweep varied LoRA rank, dropout, learning rate, and maximum training sequence length. The best model was selected based on CAID2-NOX validation ROC-AUC, with the selected configuration among the top-performing settings. The selected configuration used LoRA rank $r = 2$, scaling $\alpha = 2$, dropout 0.4, learning rate 2×10^{-4}, maximum training length 1,500, maximum test length 5,000, batch size 3, and 15 epochs.

Table 4: Best LoRADisPred configuration selected by CAID2-NOX validation ROC-AUC.

Hyperparameter	Value
Backbone	`esm2_t33_650M_UR50D`
Epochs	15
Batch size	3
Maximum training length	1,500
Maximum test length	5,000
LoRA scaling α	2
LoRA dropout	0.4
LoRA rank r	2
Learning rate	2×10^{-4}

4.4 Hyperparameter sensitivity

Table 5 summarizes the effect of each hyperparameter on CAID2-NOX validation ROC-AUC across the 1,003 completed sweep runs. Validation ROC-AUC varied narrowly across most hyperparameter settings (range 0.817–0.829), indicating the model is relatively robust to these choices once LoRA adaptation is in place. Maximum training length showed the clearest trend: maxlen 1,000 outperformed 1,500 and 1,730 in both maximum and mean ROC-AUC. Rank differences were small, with $r \in \{2, 4\}$ achieving the highest individual maxima. Dropout values between 0.1 and 0.5 showed minimal variation in mean ROC-AUC. Learning rates 2×10^{-4} and 3×10^{-4} were nearly tied, with performance declining at higher rates.

Table 5: Hyperparameter sensitivity on CAID2-NOX validation ROC-AUC across 1,003 completed sweep runs. For each hyperparameter, all finished runs with that value are included regardless of other settings. **Bold** marks the selected value.

Hyperparameter	Value	Max ROC-AUC	Mean ROC-AUC	n
LoRA rank r	1	0.827	0.825	54
	2	**0.828**	0.823	120
	4	0.829	0.823	116
	8	0.828	0.825	50
	16	0.828	**0.825**	45
LoRA dropout	0.1	0.828	0.824	96
	0.2	0.828	0.824	78
	0.3	0.828	0.824	74
	0.4	**0.828**	**0.824**	74
	0.5	0.829	0.823	63
Learning rate	$\mathbf{2\times10^{-4}}$	**0.828**	**0.825**	94
	3×10^{-4}	0.829	0.824	93
	4×10^{-4}	0.827	0.825	49
	5.7×10^{-4}	0.827	0.822	101
	6×10^{-4}	0.825	0.824	48
Max train length	1,000	0.829	**0.825**	281
	1,500	**0.823**	0.820	90
	1,730	0.822	0.818	14

Figure 5 visualizes these trends across the main hyperparameters varied in the sweep.

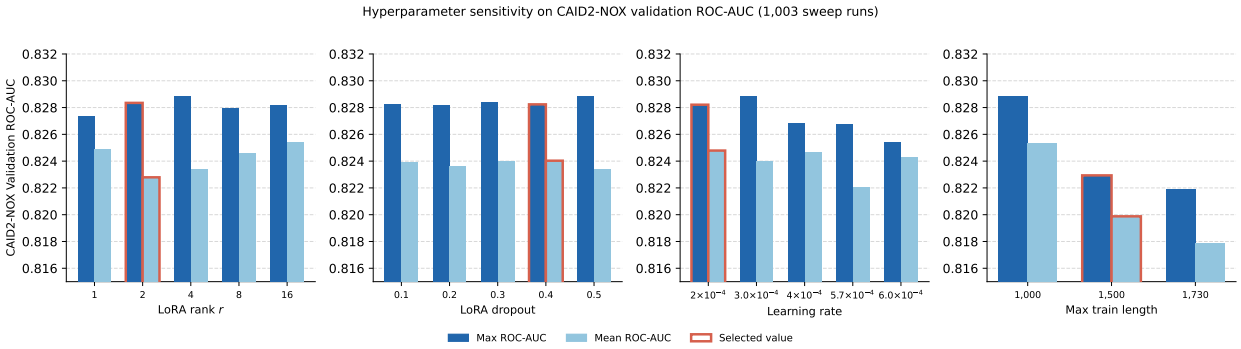

Figure 5: Hyperparameter sensitivity on CAID2-NOX validation ROC-AUC across 1,003 completed sweep runs. Each panel varies one hyperparameter while pooling all values of the remaining hyperparameters. Blue bars show the maximum ROC-AUC achieved; light blue bars show the mean. Red outlines mark the selected value for the final model.

4.5 Computational resources

We logged system-level resource usage during the hyperparameter sweep using MLflow system metrics. Across completed runs with available resource logs, peak GPU memory usage reached 99.1%, corresponding to approximately 99.6 GB of GPU memory. The maximum recorded GPU utilization was 100%, and the maximum GPU power draw was 399.8 W. The mean evaluation runtime was 26.6 seconds, with a mean throughput of 16.4 samples per second. The mean training time was 56.5 minutes and the longest recorded training run was 229.1 minutes. These measurements show that LoRADisPred avoids full fine-tuning of the 650M-parameter ESM-2 backbone while still requiring high-memory GPU hardware for long-sequence training and evaluation.

Table 6: Computational-resource usage recorded during the LoRADisPred hyperparameter sweep. Values summarize runs with available MLflow system metrics.

Resource metric	Mean	Maximum
GPU memory usage (%)	32.4	99.1
GPU memory usage (GB)	32.6	99.6
GPU utilization (%)	5.1	100.0
GPU power draw (W)	103.8	399.8
CPU utilization (%)	9.3	39.1
System memory usage (GB)	3.5	9.7
Evaluation runtime (s)	26.6	48.8
Evaluation samples/s	16.4	21.9
Training time (min)	56.5	229.1

5 Discussion

The hyperparameter sweep shows that validation ROC-AUC varies narrowly across rank values (0.823–0.825 mean), indicating that ESM-2 representations require only a modest task-specific adjustment for disorder prediction. Ranks $r \in \{2, 4\}$ achieved the highest individual run maxima while higher ranks performed comparably in mean ROC-AUC, consistent with the interpretation that a low-rank update is sufficient to capture the task-relevant signal and that increasing capacity beyond rank 4 does not provide a systematic benefit.

The choice of CRF prediction head reflects the sequential structure of intrinsic disorder. Disordered and ordered regions tend to form contiguous segments, so the transition scores learned by the CRF can encode a preference for persistent labels and penalize isolated flips between states. This is especially valuable under class imbal-

ance, where an independent per-residue classifier may scatter spurious disordered predictions throughout predominantly ordered stretches. The combination of a CRF head with class-weighted loss thus addresses both the sequential and distributional challenges of the task.

LoRADisPred matches ESMDisPred-2PDB to within 0.001 AUC while surpassing it in average precision, despite updating only 1.85% of the total model parameters. The near-identical AUC suggests that the ranking of disordered versus ordered residues is largely determined by ESM-2's pretrained representations, and that full fine-tuning provides little additional discriminative benefit for this task. The superior average precision of LoRADisPred indicates better retrieval of disordered residues at the top of the ranked list, which may reflect the class-imbalance handling or the CRF head's tendency to produce more contiguous, higher-confidence disordered segments. Several limitations should be noted: only the 650M ESM-2 backbone was evaluated; the primary results are reported under the NOX reference definition; and alternative parameter-efficient adaptation strategies such as IA3 or prefix tuning were not compared.

6 Conclusions

LoRADisPred shows that a large protein language model can be adapted to per-residue disorder prediction by training a small fraction of its parameters with LoRA, rather than fine-tuning the full backbone. The model uses token-weighted layer pooling, a CRF head for structured sequence prediction, and class-weighted cross-entropy to handle label imbalance, producing calibrated probabilities through temperature scaling and a validation-selected threshold. Evaluated on the CAID2 and CAID3 benchmarks under a strict homology-separation protocol, the method offers a practical, resource-efficient route to applying large PLMs in structural bioinformatics.

Although LoRADisPred performs competitively on CAID3-NOX, the present study focuses on a single ESM-2 backbone size and binary disorder prediction under the NOX reference definition. Future work should extend the evaluation to additional disorder definitions, continuous disorder scores, independent benchmarks beyond CAID, and related per-residue tasks such as binding-region or linker prediction. A systematic comparison of LoRA with other parameter-efficient adaptation methods is also a natural next step.

7 Computational Environment and Reproducibility

Experiments were performed in isolated container-based software environments to reduce variation across systems and improve reproducibility. Both Docker and Apptainer definitions were maintained for the experimental pipeline, enabling the same training and evaluation workflow to be executed on local and high-performance computing infrastructure. Model training and evaluation were carried out on Louisiana Optical Network Infrastructure (LONI) compute nodes with NVIDIA A100 GPUs.

For each run, configuration files and execution metadata were saved together with the corresponding model outputs. The recorded information includes the selected hyperparameters, random seed, software environment, and evaluation settings. Random seeds were fixed across experiments whenever deterministic execution was supported by the underlying libraries. Resource-usage information was also collected during the sweep, including GPU memory usage, GPU utilization, CPU utilization, and runtime statistics.

8 Data Availability

The datasets used in this study were constructed from publicly available protein-disorder resources. Training labels were derived from the DisProt 2024 release (Aspromonte et al., 2024) and processed into residue-level binary disorder labels as described in the Methods section. The reported evaluation was performed on the CAID3-NOX benchmark test set.

Processed data files, non-redundant training splits, and sequence-identity filtering outputs are available through the project repository: `https://github.com/wasicse/LORADisPred`. The original CAID benchmark data can be accessed from the CAID website: `https://caid.idpcentral.org/`.

9 Code Availability

The implementation of LoRADisPred is publicly available at: `https://github.com/wasicse/LORADisPred`. The repository contains the data-processing scripts, training and evaluation code, model-configuration files, trained adapter weights, and container files required to reproduce the reported experiments.

Documentation in the repository describes the software dependencies, hardware environment, and execution commands used for training and evaluation. The associated web server is available at: `https://bmll.cs.uno.edu/`.

10 Conflict of Interest

The authors declare no conflicts of interest.

11 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 2P20GM103424-24. Additional support from LA EPSCoR through the NSF EPSCoR SURE Program (NSF Cooperative Agreement No. OIA-2437963) and the Louisiana Optical Network Infrastructure (LONI) for access to high-performance computing resources are gratefully acknowledged.

References

Maria Cristina Aspromonte et al. DisProt in 2024: improving function annotation of intrinsically disordered proteins. *Nucleic Acids Research*, 52(D1):D434–D441, 2024.

Alessio Del Conte et al. Critical assessment of protein intrinsic disorder prediction (CAID) – round 2. *Proteins: Structure, Function, and Bioinformatics*, 2023. CAID Round 2.

Gábor Erdős, Mátyás Pajkos, and Zsuzsanna Dosztányi. IUPred3: prediction of protein disorder enhanced with unambiguous experimental annotation and visualization of evolutionary conservation. *Nucleic Acids Research*, 49(W1):W297–W303, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pp. 1321–1330, 2017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Gang Hu, Akila Katuwawala, Kui Wang, Zhonghua Wu, Sina Ghadermarzi, Jianzhao Gao, and Lukasz Kurgan. flDPnn: Accurate intrinsic disorder prediction with putative propensities of disorder functions. *Nature Communications*, 12(1):4438, 2021.

Dagmar Ilzhöfer, Michael Heinzinger, and Burkhard Rost. SETH predicts nuances of residue disorder from protein embeddings. *Frontiers in Bioinformatics*, 2:1019597, 2022.

Md Wasi Ul Kabir and Md Tamjidul Hoque. DisPredict3.0: Prediction of intrinsically disordered regions/proteins using protein language model. *Applied Mathematics and Computation*, 472:128630, 2024. doi: 10.1016/j.amc.2024.128630.

Md Wasi Ul Kabir, Ayon Dey, Farzeen Nafees, and Md Tamjidul Hoque. ESMDisPred: A structure-aware CNN-transformer architecture for intrinsically disordered protein prediction. *bioRxiv*, 2026. doi: 10.64898/2026.01.22.701204. Preprint.

John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, pp. 282–289, 2001.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

Marco Necci, Damiano Piovesan, CAID Predictors, DisProt Curators, and Silvio C. E. Tosatto. Critical assessment of protein intrinsic disorder prediction. *Nature Methods*, 18(5):472–481, 2021.

Istvan Redl, Carlo Fisicaro, Oliver Dutton, Falk Hoffmann, Louie Henderson, Benjamin M. J. Owens, Matthew Heberling, Emanuele Paci, and Kamil Tamiola. ADOPT: intrinsic protein disorder prediction through deep bidirectional transformers. *NAR Genomics and Bioinformatics*, 5(2):lqad041, 2023.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob 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.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, et al. Transformers: State-of-the-art natural language processing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pp. 38–45, 2020.