Pool PaRTI: A PageRank-based Pooling Method for Robust Protein Sequence Representation in Deep Learning

Alp Tartici
Stanford University
tartici@stanford.edu

Gowri Nayar
Stanford University
gnayar@stanford.edu

Russ B Altman
Stanford University
rbaltman@stanford.edu

Abstract

Motivation: Protein language models generate token-level embeddings for each residue, necessitating a method to pool these into a single vector representation of the entire protein. Traditional pooling methods often result in substantial information loss, impacting downstream task performance. We aim to develop a task-agnostic pooling method that preserves more information from token embeddings while offering biological interpretability.

Results: We introduce Pool PaRTI, a novel pooling method that leverages internal transformer attention matrices and PageRank to assign token importance weights. Pool PaRTI demonstrates statistically significant performance improvements across three diverse protein machine learning tasks, outperforming traditional pooling methods. It enhances accuracy, AUPRC, and MCC while offering interpretability by identifying biologically relevant regions without explicit structural training. Our approach is generalizable to encoder-containing protein language models.

Availability and Implementation: Pool PaRTI is implemented in Python with PyTorch and is available at `https://github.com/Helix-Research-Lab/Pool_PaRTI.git`

1 Introduction

In the rapidly evolving field of protein bioinformatics, the development of large language models for proteins (referred to as protein language models or PLMs) has marked a significant technological advancement, offering new pathways to understanding the complex nature of biological sequences [1]. Protein language models treat amino acids at different positions as distinct tokens and primarily produce embeddings for each of these tokens [1, 2]. Thus, for an input protein with (N) amino acids, the PLM generates (N) distinct token embedding vectors. While this token-level representation is highly informative, having N vectors per protein is computationally intensive and less practical for many downstream tasks that require a single, fixed-length representation per protein. However, the methods traditionally used for deriving a single sequence embedding vector from these token embeddings—such as sum pooling, mean pooling, max pooling, or the direct use of the [CLS] token (a special token added to the input sequence in some models to mark the beginning of the sequence)—can lead to significant information loss in compression. We hypothesize that these pooling methods may fail to capture the complex, multi-dimensional nature of biologically relevant information within the embedding, which is crucial for supporting predictive tasks in bioinformatics. Indeed, our experimental results demonstrate that protein sequence embeddings generated by these conventional methods frequently fall short in their ability to effectively support tasks such as protein subcellular localization prediction, protein interaction prediction, and functional annotation.

Preprint. Under review.

The potential shortfall of these traditional pooling techniques lies in their inability to capture the variability in functional importance of segments or individual tokens. Sum pooling and mean pooling treat all tokens equally, potentially diluting the influence of critical but less frequent tokens that may be vital for specific biological functions. Conversely, max pooling may amplify noise by giving undue importance to the most extreme values, potentially skewing the representation away from biologically meaningful interpretations. In BERT-style models, thanks to the next sentence prediction task, the [CLS] token is instrumental in learning an aggregate summary of texts, which is pivotal for tasks such as sentiment analysis or document classification [3]. However, in protein language models like ESM-2, the emphasis is not on predicting the continuity between two sequences but only on predicting masked amino acids through the Masked Language Model (MLM) objective function for a single sequence [2]. The absence of the next sequence prediction task in these models may limit the [CLS] token's capacity to aggregate and learn from the broader context of the sequence.

To address these potential limitations, we created a novel pooling method that leverages the internal attention matrices of transformer-based protein language models. These attention matrices, which are fundamental to the transformer architecture, contain rich information about the relationships between different tokens in the sequence. By utilizing this internal information, we aim to develop a more nuanced and biologically informed method for generating protein sequence embeddings.

Our proposed method, Pool PaRTI (**Pool**ing by **Pa**ge**R**ank **T**oken **I**mportance), combines insights from graph theory with the internal attention mechanisms of transformer models. This approach not only aims to preserve more of the complex biological information encoded in protein sequences but also offers a task-agnostic solution that can be generalized to various protein language models and downstream tasks.

In this paper, we present Pool PaRTI and demonstrate its effectiveness across three diverse protein-related tasks. We provide empirical evidence of its superior performance compared to traditional pooling methods, including sum pooling, mean pooling, max pooling, and the [CLS] token. Furthermore, we explore the biological interpretability enabled by Pool PaRTI, showing that it can identify functionally relevant regions in proteins without explicit structural training. This work represents a significant step towards more effective and biologically meaningful protein sequence representations, with potential implications for enhancing a wide range of bioinformatics discovery applications from protein functional annotation to variant effect prediction and possibly beyond to other domains of sequence analysis in machine learning.

The quest for more effective pooling methods in protein language models has led to various attempts to surpass the traditional max pooling, sum pooling, mean pooling, and the [CLS] token approaches. Despite these efforts, a common limitation persists: many of these new methods are tailored to specific tasks [4, 5], which restricts their generalizability and application across different types of bioinformatics analysis. An instance is seen in the DeepLoc 2.0 model [4], which for the protein localization task, learns a scalar attention vector. This vector is used with token embeddings to assign weights in weighted average pooling, highlighting biologically significant tokens, or amino acids, that are relevant to localization [4]. However, the task-specific nature of the scalar attention vector means that it is optimized primarily for localization and may not perform well in other contexts, such as protein-protein interaction prediction. This limitation underscores a critical challenge: task-specific models offer improvements in their designated areas but often fail to adapt to the diverse needs of broader protein sequence analysis.

Adding to the diversity of pooling techniques, NaderiAlizadeh and Singh have developed an optimal transport-based pooling method that treats each token's embedding as a sample from a probability distribution [5]. This method leverages sliced-Wasserstein distances to align these samples with a learnable reference set, generating a Euclidean embedding in the output space that encapsulates the entire protein [5]. This approach, however, relies heavily on task-specific trainable reference elements for constructing the reference probability distribution [5]. Such dependence on specific reference elements raises concerns regarding the bias that the reference selection may introduce and the ability of the generated embeddings to generalize across various tasks.

This uncertainty about the adaptability of task-specific embeddings to different predictive tasks underscores a critical limitation in the existing methodologies.

The specificity of current approaches, while beneficial in narrow applications, points to a broader opportunity: the development of a pooling method that can transcend the limitations of task-specific tuning to offer wide-reaching utility across the diverse spectrum of tasks encountered in protein bioinformatics. To our knowledge, there is no existing literature that explores the use of internal model parameters for developing expressive, task-agnostic pooling methods.

While not directly addressing pooling methods, there is precedence for using PLM attention maps to interpret embeddings created by black-box models. Vig et al. [1] demonstrated significant correlations between attention patterns and well-characterized protein features, providing insights into what these models learn about protein structure and function. This approach to interpretability, while not focused on pooling, highlights the potential of leveraging internal model parameters for gaining biological insights from protein language models.

2 Systems and Methods

Pool PaRTI capitalizes on the internal attention matrices of transformer-based protein language models. These matrices provide a rich representation of the relationships between tokens in the sequence, offering a more nuanced view of token interactions than traditional pooling methods.

Pool PaRTI uses a weighted average of token embeddings to construct a comprehensive sequence embedding vector. This process involves leveraging the attention matrices across all layers of the transformer model. Unlike raw attention heads that only yield pairwise token importance values, Pool PaRTI conceptualizes the protein attention matrices as adjacency matrices in a graph model. To generate the source matrix for our graph, we perform pixel-wise max pooling across all layers of the internal attention matrices. This operation captures the strongest attention relationships across the entire model. Within the resulting graph, amino acid tokens are represented as nodes, and their interactions, determined by the pooled attention from the query to the key, serve as directed edges with corresponding weights (Figure 1). Upon constructing this graph, we run the PageRank algorithm on the graph to extract normalized node importance values, which directly correspond to token weights in weighted average pooling (Algorithm 1). We normalize the token weights to sum to one to ensure that the output sequence embedding is independent of the input sequence length.

We selected the PageRank algorithm for its effectiveness in graphs with weighted and directed edges, useful for delineating token significance within the complex hierarchy within protein sequences [6]. For a graph G with N nodes and E edges, the PageRank algorithm with a random surfer model has a space complexity of $O(N + E)$ [6] because at each iteration, the algorithm only stores the edge connectivity, the weights, if any, and the current node importance values. This level of space complexity is manageable within the context of the much larger space used by the foundation model itself. At each iteration, PageRank computes importance transfer along all the edges. Hence, the time complexity is $O(kE)$ for convergence over k iterations [6]. Given that the attention matrix from BERT style models typically results in a fully connected graph where E grows quadratically with N, the computational cost of applying PageRank directly on the original matrix would scale quadratically with N, the length of the protein sequence.

We validated Pool PaRTI's efficacy across three distinct protein-related tasks—multilabel subcellular localization, binary protein-protein interaction, and enzyme activity prediction—using consistent deep learning architectures. The architectural consistency ensures that observed performance enhancements are directly attributable to our pooling method.

For evaluation, we use the Matthews Correlation Coefficient (MCC) and Area Under the Precision-Recall Curve (AUPRC), chosen for their robustness in assessing binary classification performance across unbalanced datasets, alongside accuracy. For multi-label prediction, we also report the Jaccard index to evaluate performances. To

Algorithm 1 Pool PaRTI: Protein Sequence Embedding Calculation

Input: Protein sequence of length n
Output: Protein sequence embedding E_{seq}
Run a forward pass with the transformer-based protein language model to generate:
  $E_{tok} \in \mathbb{R}^{n \times d}$ token embeddings
  $A_l \in \mathbb{R}^{n \times n}$ attention matrices for each layer $l \in \{1, \ldots, L\}$
Perform pixel-wise max pooling across all attention matrices:
  $M_{att} = \max_{l \in \{1,\ldots,L\}} A_l$
Construct graph $G = (N, E)$ where:
  N: Set of nodes, each representing a token
  E: Set of edges, with weights $W \in \mathbb{R}^{n \times n}$ from M_{att}
Apply PageRank to G with:
  Damping factor $d = 0.85$
  Maximum iterations $= 100$
  Convergence tolerance $= 10^{-6}$
$\alpha_{imp} \in \mathbb{R}^n \leftarrow$ Extracted token importance weights from PageRank
Normalize α_{imp} such that $\alpha_{imp,i} \leftarrow \frac{\alpha_{imp,i}}{\sum_{j=1}^{n} \alpha_{imp,j}}$ for all $i \in \{1, \ldots, n\}$
Compute E_{seq} as:
  $E_{seq} = \sum_{i=1}^{n} \alpha_{imp,i} \cdot E_{tok,i}$ where $E_{tok,i} \in \mathbb{R}^d$
return E_{seq}

ensure the statistical significance of our results, we conducted experiments using 10 different random seeds for all methods under comparison. We report the mean and standard deviation for each condition, along with p-values, to demonstrate the statistical significance of our performance improvements.

3 Experiments and Results

We conducted a series of experiments to evaluate the performance of Pool PaRTI across three different protein-related tasks. We used the 650 million parameter version of the ESM-2 protein language model to generate token embedding due to its manageable size and state-of-the-art token representation. We designed each task to assess the pooling method's efficacy in retaining the expressive power of protein language models compared to the baselines of four traditional sequence embedding generation methods: [CLS] token, mean pooling, sum pooling, and max pooling.

In our first experiment, we aim to determine if a given protein sequence can be classified into any of eight possible subcellular locations, such as the nucleus, cell membrane, mitochondria, or endoplasmic reticulum, using the dataset split in [4]. This task is inherently a multilabel binary classification since a protein can localize to multiple cellular compartments simultaneously. Pool PaRTI significantly outperforms all four baseline methods in terms of accuracy, Matthews Correlation Coefficient (MCC) and Area Under the Precision-Recall Curve (AUPRC) metrics. Furthermore, it significantly outperforms all four baselines in Jaccard index, indicating its robust capability in extracting spatial information and handling the complexities of multilabel classification (Table 1).

The second experiment focuses on binary classification of proteins on whether they have an enzymatic activity or not, based Enzyme Commission (EC) numbers, which denote the type of enzymatic activity or lack thereof. To ensure robust model training and minimize data leakage, we used the gold standard ECPred40 dataset created by [7], which limited sequence identity across training, validation, and test datasets to 40%. We opted to use EC numbers over the Gene Ontology (GO) codes due to their robustness compared to GO term

prediction, which has significant redundancy and label noise. The models were trained on human proteins and tested separately on sequences from vertebrate and eukaryotic proteins. In predicting enzymatic activity for vertebrate proteins, Pool PaRTI again significantly outperforms the baselines in accuracy, MCC, and AUPRC metrics. While Pool PaRTI outperforms the baselines in AUPRC, accuracy, and MCC for the eukaryotic protein sequences too, the performance boost over mean pooling is slightly less statistically significant (Table 2).

Our third task tests performance on predicting whether two given protein sequences interact one another, using gold standard data split taken from [8]. This binary classification task evaluates the method's ability to retain the expressive power regarding the complex biological interactions across sequences. Pool PaRTI significantly outperforms the [CLS] token, sum pooling, and max pooling methods in terms of accuracy, MCC, and AUPRC. Pool PaRTI outperforms mean pooling as well on all three metrics, albeit with less statistical significance (Table 3). This outcome highlights its efficacy in capturing nuanced biological interactions across sequences that are not solely dependent on the most extreme token features.

To showcase the reproducibility and generalizability of Pool PaRTI's performance, it's important to highlight that these enhancements are not attributable to isolated, optimal settings. By averaging runs conducted at 10 new random seeds, rather than relying on a single optimal configuration, we consistently observed performance gains. This methodological approach confirms that the improvements with Pool PaRTI are reproducible and robust, demonstrating that its effectiveness is not contingent on finding serendipitous hyperparameter combinations but is consistently achievable across varied settings.

Based on Tables 1-3, the closest rival of Pool PaRTI in head-to-head comparison is the mean pooling method. However, in all instances, Pool PaRTI has a higher average performance than the mean pooling method. In 8 of the 13 metrics, the Mann-Whitney U-test indicates performance superiority with significance of p-value less than 0.05. Four of the remaining five metrics indicate Pool PaRTI's performance gain has statisical significance of p-value under 0.1. These results demonstrate that Pool PaRTI has a clear advantage over its closest rival, the mean pooling method.

To further validate the generalizability of Pool PaRTI, we examined its performance with another protein language model, protBERT [9], in addition to ESM-2. We evaluated how well the sequence embeddings generated by different pooling methods align with SCOP (Structural Classification of Proteins) families. SCOP is a database that classifies protein structures hierarchically, with level 2 corresponding to broad structural and functional relationships [10, 11]. For both ESM-2 and protBERT, Pool PaRTI achieved significantly higher ratios of inter-cluster to intra-cluster distances compared to all baseline methods, indicating better preservation of information related to protein structure and function.

Pool PaRTI not only enhances performance in protein sequence representation but also offers a degree of interpretability, addressing a gap in existing pooling methods. While protein language models capture rich, contextualized information in their token embeddings, traditional pooling methods often lose much of this information when compressing to a single sequence embedding. Pool PaRTI, however, preserves more of this valuable information by discerning which parts of the sequence carry more useful information than others. To investigate the biological relevance of the importance weights assigned by our method, we conducted an in-depth analysis of several well-characterized proteins. This examination revealed a striking overlap between residues assigned higher importance weights by Pool PaRTI and known functionally or structurally significant regions within these proteins. A notable example is observed in Cytochrome C, a crucial component of the electron transport chain [12]. Pool PaRTI assigned higher importance weights to residues within the heme-binding regions, which are essential for the protein's electron transfer function [12] (Figure 2A). Similarly, in superoxide dismutase-1 (SOD1), an enzyme critical for cellular antioxidant defense [13], our method highlighted regions proximal to the zinc-binding domain (Figure 2B). This domain is integral to SOD1's catalytic activity and structural stability, and mutations therein are implicated in Amyothropic Lateral

Sclerosis (ALS) [14]. These findings are particularly significant as they demonstrate Pool PaRTI's capacity to identify biologically relevant regions without explicit training on structural or functional data. This emergent property suggests that our method captures intrinsic sequence features that correlate with functional importance, potentially offering new insights into protein structure-function relationships.

Regarding computational efficiency, we measured the time required to carry out Pool PaRTI on CPU (32 GB memory) as a function of sequence length, validating our theoretical reasoning about quadratic computational cost scaling (Figure 3, Appendix A.2). Our empirical analysis shows that Pool PaRTI does not introduce prohibitive computational overhead compared to other pooling methods at the input sizes relevant to human proteome. Model configurations for all three experiments are detailed in Appendix A.3

4 Discussion

Pool PaRTI is a novel approach to generating protein sequence embeddings that outperforms traditional pooling methods across a remarkable range of bioinformatics tasks. By leveraging the internal attention matrices of transformer-based protein language models, Pool PaRTI captures complex relationships between amino acids directly from sequence information, without relying on explicit structural data.

Our evaluation across three distinct protein-related tasks—subcellular localization, protein-protein interaction prediction, and enzyme classification—demonstrates the robustness and versatility of Pool PaRTI. These tasks represent diverse and important aspects of protein research. Each task relies on different biological properties and information encoded in protein sequences. Enzymatic activity prediction relies on the capability to encode and recognize local functional motifs, such as active sites and catalytic residues, embedded within sequences. This demonstrates the model's understanding of sequence-to-function relationships. Protein protein interaction prediction involves identifying patterns across multiple protein representations, while spatial analysis involves clusters of proteins. These three tests comprehensively evaluate our method's effectiveness across various protein-related prediction tasks and its ability to make accurate predictions at multiple scales. The method consistently achieves statistically significant improvements in key performance metrics, including accuracy, Matthews Correlation Coefficient (MCC), and Area Under the Precision-Recall Curve (AUPRC). Notably, Pool PaRTI outperforms all baseline methods, including mean pooling, max pooling, sum pooling, and the use of the [CLS] token, across all tasks and metrics.

The superior performance of Pool PaRTI is particularly evident in its handling of class imbalances, as demonstrated by the improved AUPRC scores. This capability is crucial for real-world applications in protein science, where imbalanced datasets are common. By effectively capturing and utilizing biologically significant patterns, even in underrepresented classes or interactions, Pool PaRTI addresses a critical challenge in protein bioinformatics.

Our analysis of Pool PaRTI's performance across two different protein language models provides additional evidence for its effectiveness. The SCOP family alignment results suggest that Pool PaRTI preserves more structural and functional information in sequence embeddings compared to baseline methods. This preservation was observed not only in ESM-2 but also in protBERT, indicating that Pool PaRTI's performance gain may extend to various transformer-based protein language models.

One of the most striking features of Pool PaRTI is its interpretability. Our analysis of well-characterized proteins, such as Cytochrome C and superoxide dismutase-1 (SOD1), reveals a strong correlation between residues assigned higher importance weights by Pool PaRTI and known functionally or structurally significant regions. This emergent property, achieved without explicit training on structural or functional data, suggests that Pool PaRTI captures intrinsic sequence features that correlate with functional importance, offering new insights into protein structure-function relationships.

Traditional pooling methods, such as mean pooling, max pooling, and sum pooling lack this interpretability. These approaches aggregate token embeddings into a single vector without providing any indication of which sequence regions contribute most significantly to the final representation. In contrast, Pool PaRTI's assignment of importance weights to individual residues allows for a more nuanced understanding of how different regions of a protein sequence influence the overall representation. This interpretability feature of Pool PaRTI opens up new avenues for protein analysis. It may aid in the identification of potential functional sites, guide experimental design for mutagenesis studies, or provide valuable insights into protein engineering applications. Furthermore, the ability to highlight potentially important regions without relying on structural data makes Pool PaRTI particularly valuable for studying proteins with unknown structures or intrinsically disordered regions.

The task-agnostic nature of Pool PaRTI is a key advantage. Unlike methods that require task-specific training or optimization, our approach generates informative sequence embeddings without the need for task-specific adjustments. By employing PageRank instead of more complex graph learning algorithms, we deliberately avoid parameter optimization tailored to specific downstream tasks. This choice enables Pool PaRTI to create general-purpose token importance weights, enhancing its versatility across various protein analysis tasks. This design makes Pool PaRTI particularly valuable for exploratory analyses and for applications in data-scarce domains where task-specific training data may be limited. Pool PaRTI operates independently from the neural network's training phases, allowing it to be integrated with any transformer-based protein language model. This feature facilitates its application without the need for additional training or alterations to the original model architecture, further reinforcing its task-agnostic nature.

From a computational perspective, our analysis shows that Pool PaRTI does not introduce prohibitive overhead compared to other pooling methods, particularly for sequence lengths relevant to the human proteome. This efficiency, coupled with its superior performance in supporting downstream tasks, makes Pool PaRTI a practical choice for large-scale protein analysis research. Pool PaRTI may enhance the performance in many other protein analysis tasks which are beyond the scope of this study.

There are several limitations to Pool PaRTI. First, we primarily tested the method using two encoder-only protein language models, and while the use of attention matrices should theoretically allow application to other transformer-based models, we have not demonstrated its application to other PLMs. Second, although our analysis shows that Pool PaRTI is computationally efficient for typical protein sequence lengths, its performance for extremely long sequences may warrant further investigation. While Pool PaRTI offers improved interpretability compared to traditional pooling methods, the biological significance of the assigned importance weights may require careful analysis in some contexts, especially for proteins with multiple modes of interactions. Lastly, we have not demonstrated the generalizability of Pool PaRTI in other non-protein sequence analysis domains, such as natural language processing. This remains a theoretical possibility that has not yet been empirically tested.

5 Conclusion

We introduced Pool PaRTI, a novel approach to generating protein sequence embedding from individual token embeddings. By combining attention matrices from transformer-based protein language models with the PageRank algorithm, Pool PaRTI captures the complex hierarchy within protein sequences. It assigns importance weights to amino acids based on their global context, creating sequence embeddings that reflect both local and global interactions. This approach effectively distills the rich information from token-level embeddings into an expressive sequence representation.

6 Main tables and figures

Table 1: The performances in subcellular localization prediction task averaged over 10 random seeds.

	Pooling				
Metric	[CLS] token	Mean pooling	Max pooling	Sum pooling	Pool PaRTI
Acc. avg	0.872±0.012***	0.873±0.001***	0.843±0.012***	0.772±0.084***	**0.905±0.004**
MCC avg	0.256±0.104***	0.306±0.014***	0.198±0.118***	0.082±0.166***	**0.572±0.017**
AUPRC avg	0.330±0.094***	0.415±0.030***	0.195±0.074***	0.158±0.011***	**0.493±0.010**
Jaccard index	0.450±0.187***	0.520±0.008***	0.239±0.096***	0.245±0.135***	**0.639±0.006**

***p < 0.01, **p < 0.05, *p < 0.10. p values correspond to the null hypothesis that the respective pooling method has the same performance as Pool PaRTI, tested by Mann-Whitney U test.

Table 2: The performances in binary prediction of enzymatic activity averaged over 10 random seeds.

	Pooling				
Metric	[CLS] token	Mean pooling	Max pooling	Sum pooling	Pool PaRTI
Acc. (euk)	0.780±0.046***	0.844±0.026*	0.817±0.028***	0.795±0.033***	**0.872±0.031**
MCC (euk)	0.600±0.070***	0.707±0.035*	0.663±0.044***	0.637±0.044***	**0.737±0.048**
AUPRC (euk)	0.924±0.008***	0.953±0.003***	0.928±0.010***	0.937±0.008***	**0.964±0.004**
Acc. (vert)	0.831±0.056**	0.838±0.042**	0.828±0.054**	0.805±0.049***	**0.872±0.027**
MCC (vert)	0.700±0.090**	0.712±0.055**	0.704±0.081**	0.667±0.054***	**0.774±0.044**
AUPRC (vert)	0.969±0.008***	0.982±0.004***	0.980±0.007***	0.962±0.007***	**0.991±0.002**

***p < 0.01, **p < 0.05, *p < 0.10. p values correspond to the null hypothesis that the respective pooling method has the same performance as Pool PaRTI, tested by Mann-Whitney U test.

Table 3: The performances in protein-protein interaction prediction averaged over 10 random seeds.

	Pooling				
Metric	[CLS] token	Mean pooling	Max pooling	Sum pooling	Pool PaRTI
Accuracy	0.627±0.013***	0.642±0.005*	0.591±0.006***	0.627±0.005***	**0.646±0.007**
MCC	0.260±0.018***	0.285±0.008*	0.186±0.011***	0.254±0.011***	**0.291±0.017**
AUPRC	0.686±0.010***	0.697±0.005	0.628±0.007***	0.671±0.009***	**0.699±0.006**

***p < 0.01, **p < 0.05, *p < 0.10. p values correspond to the null hypothesis that the respective pooling method has the same performance as Pool PaRTI, tested by Mann-Whitney U test.

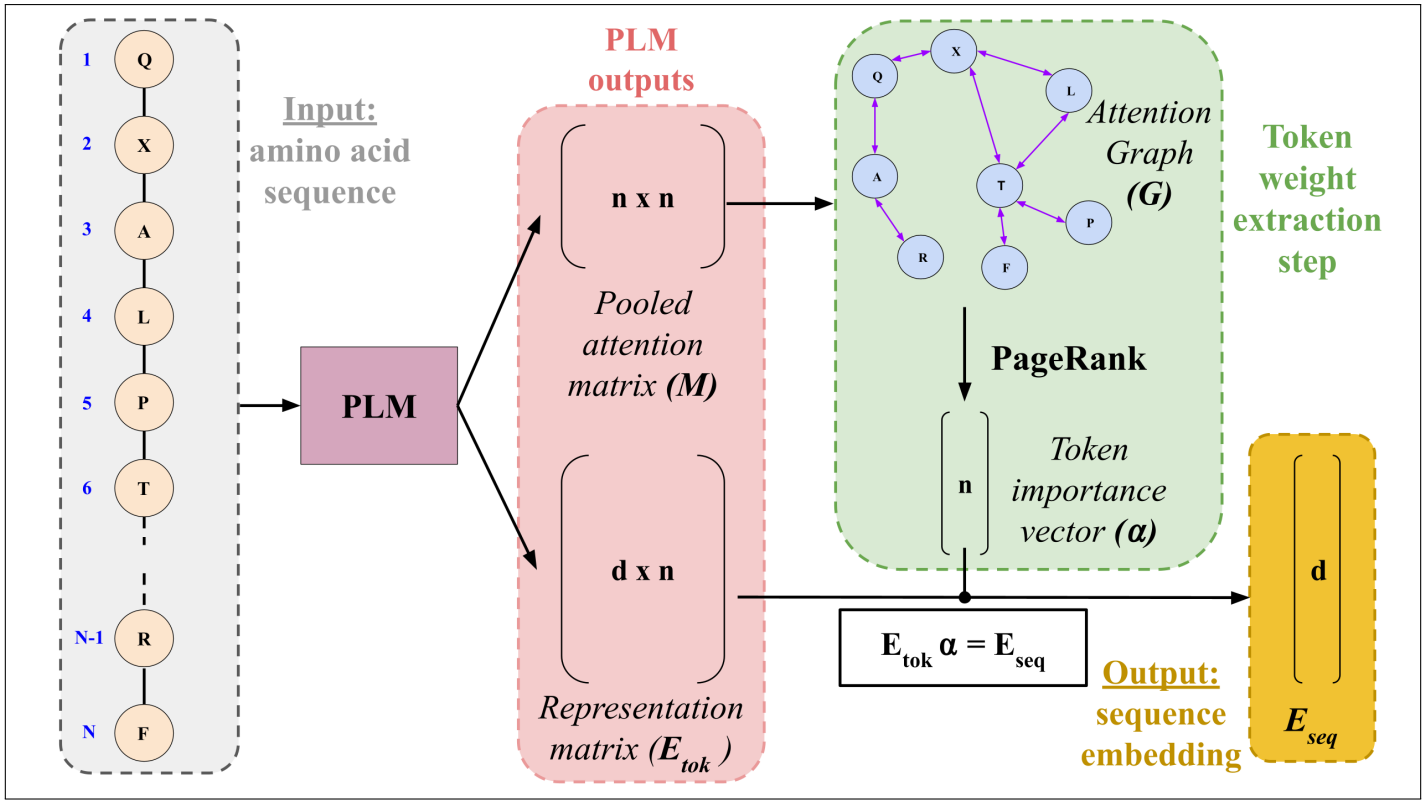

Figure 1: Graphical Summary of the Pool PaRTI Algorithm. The input is an amino acid sequence of length N (left), and the output is a d-dimensional sequence representation vector, E_{seq} (bottom right). Pool PaRTI determines this representation by calculating the weighted average of the token embeddings E_{tok} for each row. The key insight is obtaining the N-dimensional weighting vector α, which is derived by running the PageRank algorithm on a sparse graph. This graph is generated from the attention matrices inherent to all transformer-based architectures.

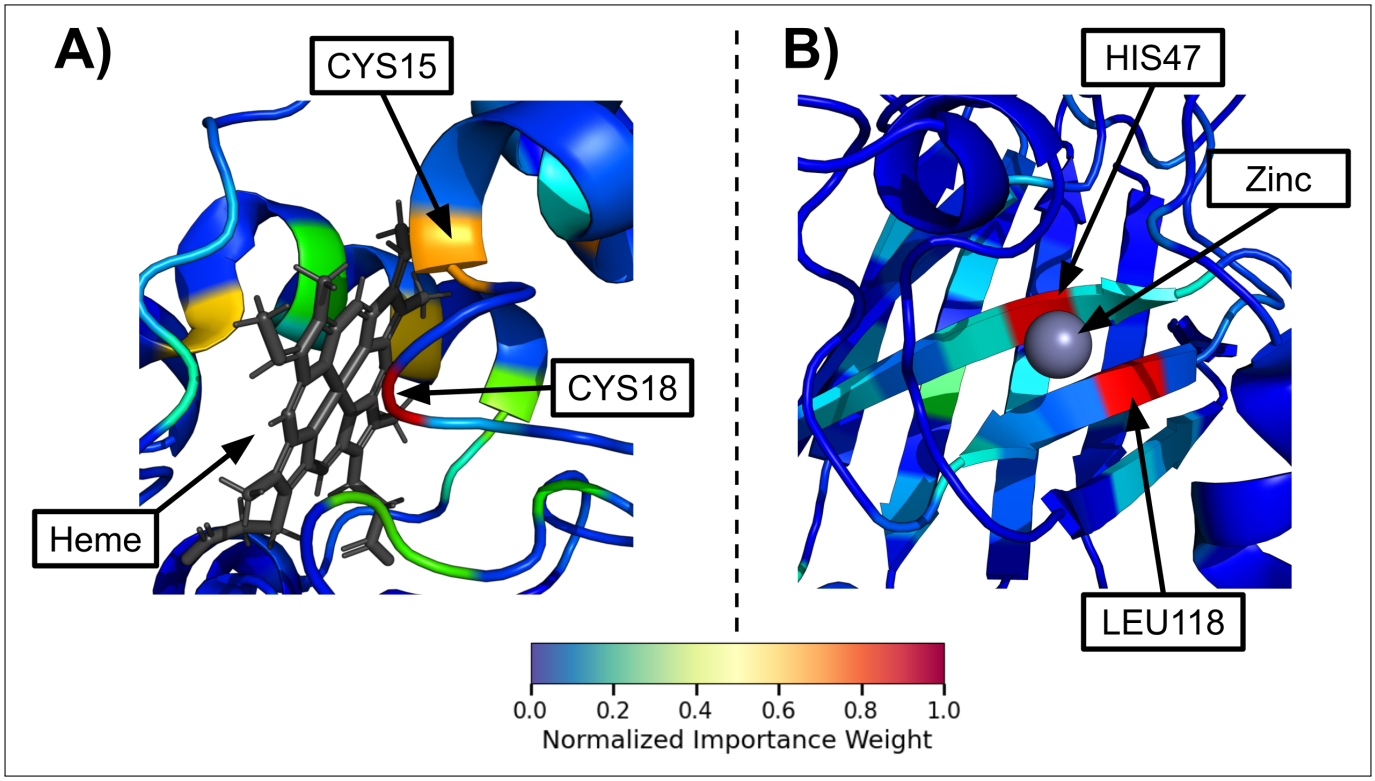

Figure 2: The normalized importance values Pool PaRTI assigns to the residues in the vicinity of **A)** the Heme-binding domain of Cytochrome C, **B)** the Zinc-binding domain of SOD1.

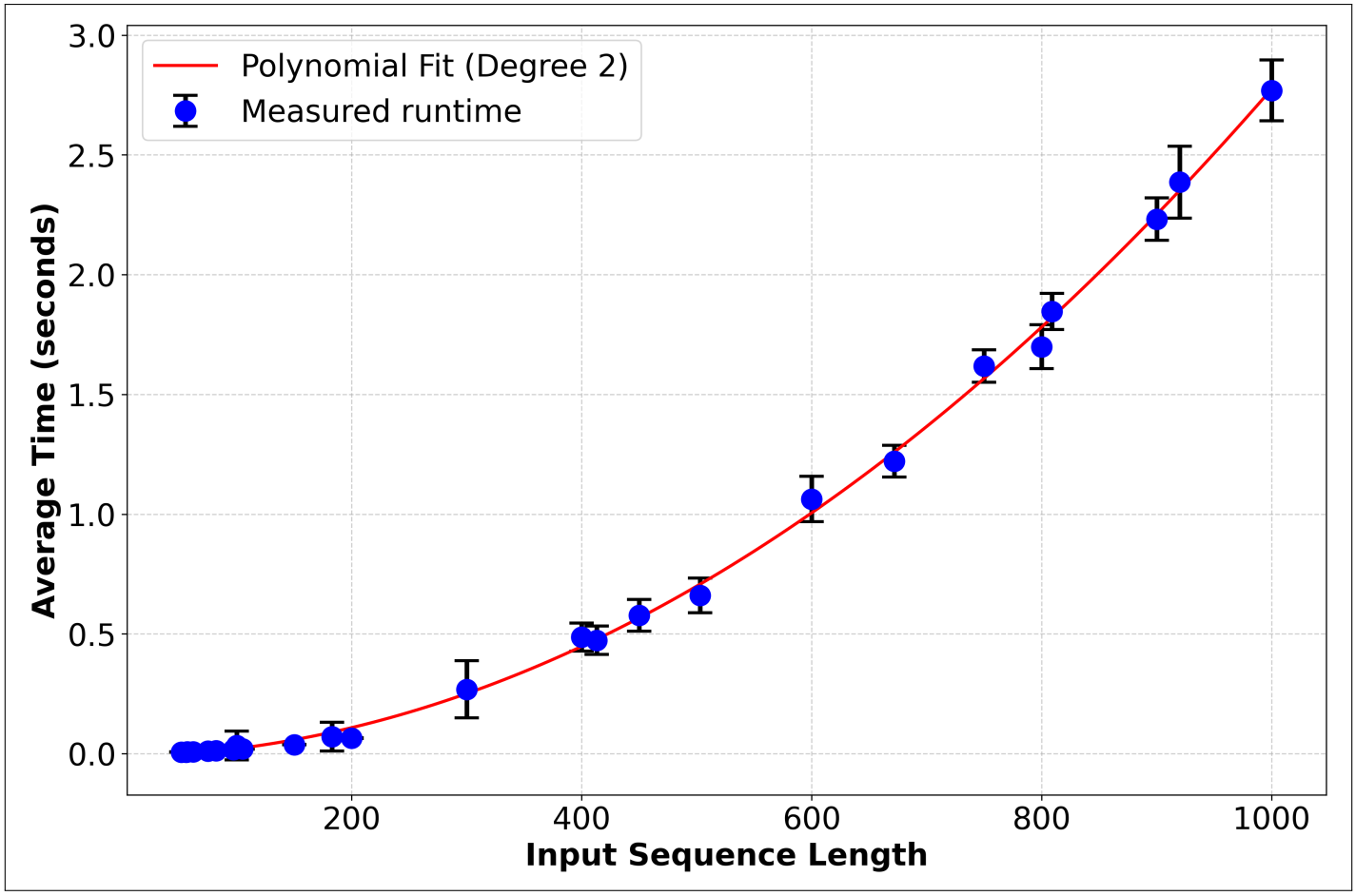

Figure 3: Measurements of the algorithm runtime follows near quadratic scaling with an empirical polynomial degree of 2.068.

7 Data and code availability

Our PyTorch implementation of Pool PaRTI and the experimental task models are available at `https://github.com/Helix-Research-Lab/Pool_PaRTI.git`. The data we used in these tasks. For the subcellular localization prediction task, we used the data annotation and splits provided by Thumuluri et al [4]. For the protein-protein interaction prediction tasks, we used the data annotation and gold standard split provided by Bernett et al. [8]. For the enzyme classification task, we used the gold standard ECPred40 dataset created by Buton et al. [7]. We obtained the SCOP labels from the SCOP database [10, 11] with API access.

8 Acknowledgements

We thank the Altman Lab members and Dr. Brian Hie for useful discussions. We performed the method development and analysis on the Sherlock cluster, and we thank Stanford University and Stanford Research Computing Center for the computational resources. A.T is supported by Smith Fellowship (Stanford Graduate Fellowship), G.N. is supported by NLM Training Grant 5T15007033-40 and NIH NLM 1F31LM014646. R.B.A. is supported by NIH GM102365 and Chan Zuckerberg Biohub.

9 Conflict of Interest

The authors declare no competing interests.

10 Funding

Smith Fellowship (Stanford Graduate Fellowship) NLM Training Grant 5T15007033-40 NIH NLM 1F31LM014646. NIH GM102365 Chan Zuckerberg Biohub

References

[1] Jesse Vig et al. *BERTology Meets Biology: Interpreting Attention in Protein Language Models*. preprint. Bioinformatics, June 28, 2020. DOI: 10.1101/2020.06.26.174417. URL: http://biorxiv.org/lookup/doi/10.1101/2020.06.26.174417 (visited on 07/25/2023).

[2] Zeming Lin et al. "Evolutionary-scale prediction of atomic-level protein structure with a language model". In: *Science* 379.6637 (Mar. 17, 2023), pp. 1123–1130. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.ade2574. URL: https://www.science.org/doi/10.1126/science.ade2574 (visited on 07/25/2023).

[3] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Version Number: 2. 2018. DOI: 10.48550/ARXIV.1810.04805. URL: https://arxiv.org/abs/1810.04805 (visited on 05/21/2024).

[4] Vineet Thumuluri et al. "DeepLoc 2.0: multi-label subcellular localization prediction using protein language models". In: *Nucleic Acids Research* 50 (W1 July 5, 2022), W228–W234. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkac278. URL: https://academic.oup.com/nar/article/50/W1/W228/6576357 (visited on 01/13/2024).

[5] Navid NaderiAlizadeh and Rohit Singh. *Aggregating Residue-Level Protein Language Model Embeddings with Optimal Transport*. preprint. Bioinformatics, Jan. 31, 2024. DOI: 10.1101/2024.01.29.577794. URL: http://biorxiv.org/lookup/doi/10.1101/2024.01.29.577794 (visited on 02/09/2024).

[6] Michael Brinkmeier. "PageRank revisited". In: *ACM Transactions on Internet Technology* 6.3 (Aug. 2006), pp. 282–301. ISSN: 1533-5399, 1557-6051. DOI: 10.1145/1151087.1151090. URL: https://dl.acm.org/doi/10.1145/1151087.1151090 (visited on 05/13/2024).

[7] Nicolas Buton, François Coste, and Yann Le Cunff. "Predicting enzymatic function of protein sequences with attention". In: *Bioinformatics* 39.10 (Oct. 3, 2023). Ed. by Alfonso Valencia, btad620. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btad620. URL: https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btad620/7329097 (visited on 01/22/2024).

[8] Judith Bernett, David B Blumenthal, and Markus List. "Cracking the black box of deep sequence-based protein–protein interaction prediction". In: *Briefings in Bioinformatics* 25.2 (Jan. 22, 2024), bbae076. ISSN: 1467-5463, 1477-4054. DOI: 10.1093/bib/bbae076. URL: https://academic.oup.com/bib/article/doi/10.1093/bib/bbae076/7621029 (visited on 04/15/2024).

[9] Ahmed Elnaggar et al. "ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (Oct. 1, 2022), pp. 7112–7127. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2021.3095381. URL: https://ieeexplore.ieee.org/document/9477085/ (visited on 08/27/2024).

[10] Antonina Andreeva et al. "The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures". In: *Nucleic Acids Research* 48 (D1 Jan. 8, 2020), pp. D376–D382. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkz1064. URL: https://academic.oup.com/nar/article/48/D1/D376/5625529 (visited on 09/27/2024).

[11] Antonina Andreeva et al. "SCOP2 prototype: a new approach to protein structure mining". In: *Nucleic Acids Research* 42 (D1 Jan. 2014), pp. D310–D314. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkt1242. URL: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkt1242 (visited on 09/27/2024).

[12] John C. Reed. "Cytochrome c: Can't Live with It—Can't Live without It". In: *Cell* 91.5 (Nov. 1997), pp. 559–562. ISSN: 00928674. DOI: 10.1016/S0092-8674(00)80442-0. URL: https://linkinghub.elsevier.com/retrieve/pii/S0092867400804420 (visited on 08/20/2024).

[13] Pamela Milani et al. "SOD1 Transcriptional and Posttranscriptional Regulation and Its Potential Implications in ALS". In: *Neurology Research International* 2011 (2011), pp. 1–9. ISSN: 2090-1852, 2090-1860. DOI: 10.1155/2011/458427. URL: http://www.hindawi.com/journals/nri/2011/458427/ (visited on 08/20/2024).

[14] Sai V. Seetharaman et al. "Disrupted Zinc-Binding Sites in Structures of Pathogenic SOD1 Variants D124V and H80R". In: *Biochemistry* 49.27 (July 13, 2010), pp. 5714–5725. ISSN: 0006-2960, 1520-4995. DOI: 10.1021/bi100314n. URL: https://pubs.acs.org/doi/10.1021/bi100314n (visited on 08/20/2024).

[15] Pierre Baldi et al. "Assessing the accuracy of prediction algorithms for classification: an overview". In: *Bioinformatics* 16.5 (May 1, 2000), pp. 412–424. ISSN: 1367-4811, 1367-4803. DOI: 10.1093/bioinformatics/16.5.412. URL: https://academic.oup.com/bioinformatics/article/16/5/412/192336 (visited on 05/22/2024).

[16] Brendan Juba and Hai S. Le. "Precision-Recall versus Accuracy and the Role of Large Data Sets". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.1 (July 17, 2019), pp. 4039–4048. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v33i01.33014039. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5193 (visited on 05/22/2024).

A Appendix / Supplemental Material

A.1 Generalizability of Pool PaRTI

Table S4: The ratio of the weighted average inter-cluster distance to weighted average intra-cluster distance averaged over 20 random subsampling iterations based on the second level of SCOP classes.

	Pooling				
Model	[CLS] token	Mean pooling	Max pooling	Sum pooling	Pool PaRTI
ESM-2 650M	2.077±0.012***	2.236±0.016***	2.219±0.018***	2.250±0.036***	**2.315±0.026**
protBERT	1.856±0.017***	2.029±0.023**	1.990±0.013***	1.978±0.021***	**2.044±0.016**

***p < 0.01, **p < 0.05, *p < 0.10. p values correspond to the null hypothesis that the respective pooling method has the same performance as Pool PaRTI, tested by Mann-Whitney U test.

A.2 Empirical analysis of Pool PaRTI runtime scaling

To formally assess the empirical scaling of runtime for the Pool PaRTI algorithm, we executed the algorithm on 25 protein sequences, repeating each run 15 times. We then applied a logarithmic transformation to both the runtime data and the corresponding sequence lengths. A linear regression model was fitted to the log-transformed data, yielding a slope of 2.068, which aligns with our theoretical reasoning in Section 2. This slope indicates the exponent in the power-law relationship between the algorithm's runtime and the input size, providing a quantitative measure of the algorithm's scaling behavior with respect to input size.

A.3 Deep Learning Models and Configurations

A.3.1 Subcellular Localization Prediction Task

Description of the model

- 2 x (linear layer + leaky ReLU + dropout)
- Linear layer
- Input residual connection
- Sigmoid

Fixed configurations

- Initial learning rate: 0.01
- Max number of epochs: 1000
- Early stopping patience epochs: 50
- Optimizer: AdamW
- Learning rate scheduler: ReduceLROnPlateau
- Optimizer learning rate patience epochs: 10
- Learning rate reduction ratio: 0.1
- Weight Initialization: Xavier normal
- Random seed: 42

- Gradient clip value: 5.0
- batch size: 64

Hyperparameter optimization space

- Weight decay: [0.01, 0.1, 0.2]
- Dropout rate: [0.15, 0.25]
- Slope of Leaky ReLU: [0.01, 0.1]
- Exponent for imbalance penalty taming: [0.75, 1, 1.25]

A.3.2 Enzymatic Activity Prediction Task

Description of the model

- 1 x (linear layer + batch norm + leaky ReLU + dropout)
- 1 x (linear layer + linearly transformed residual connection + leaky ReLU)
- Linear layer

Fixed configurations

- Hidden dimensions: 512
- Max number of epochs: 50
- Early stopping patience: 10
- Optimizer: AdamW
- Learning rate scheduler: ReduceLROnPlateau
- Optimizer learning rate patience epochs: 4
- Leaky ReLU slope: 0.1
- Random seed: 42
- Weight initialization: Kaiming normal
- Gradient clip max value: 1.0
- batch size: 64

Hyperparameter optimization space

- Initial learning rate: [0.001, 0.01]
- Weight decay: [0.01, 0.1]
- Dropout: [0.05, 0.15, 0.25]
- Reduction in learning rate ratio: [0.2, 0.5]
- Exponent in class imbalance loss: [0.75, 1, 1.25, 2]

A.3.3 Protein-Protein Interaction Prediction Task

Description of the model

- 1 x (linear layer + batch norm + leaky ReLU + dropout)
- 1 x (linear layer + linearly transformed residual connection + leaky ReLU)

- Linear layer

Fixed configurations

- Hidden dimensions: 1024
- Max number of epochs: 40
- Early stopping patience: 9
- Optimizer: AdamW
- Learning rate scheduler: ReduceLROnPlateau
- Optimizer learning rate patience epochs: 4
- Random seed: 42
- Weight initialization: Kaiming normal
- Gradient clip max value: 2.0
- batch size: 32

Hyperparameter optimization space

- Initial learning rate: [0.001, 0.01]
- Weight decay: [0.01, 0.1, 0.2]
- Dropout: [0.05, 0.15, 0.25]
- Reduction in learning rate ratio: [0.2, 0.5]
- Leaky ReLU slope: [0.01, 0.1, 0.2]

A.4 Metrics

For single label and multilabel binary classifications, we employed different metrics to assess model predictive performances under the influence of the input embeddings resulting from alternative pooling methods.

A.4.1 Accuracy

Accuracy is the simplest and most commonly used metric for evaluating classification models. It is defined as the ratio of correctly predicted instances to the total instances. Mathematically, accuracy is expressed as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

While accuracy is intuitive and easy to understand, it can be misleading for imbalanced datasets, as it does not differentiate between the types of errors (false positives and false negatives).

A.4.2 Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) is a measure of the quality of binary (two-class) classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is defined as:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2}$$

where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

MCC returns a value between -1 and 1. A coefficient of 1 represents a perfect prediction, 0 represents a random prediction, and -1 indicates total disagreement between prediction and observation [15].

A.4.3 Area Under the Precision-Recall Curve (AUPRC)

The Area Under the Precision-Recall Curve (AUPRC) is a performance measurement for classification problems at various threshold settings, especially useful for imbalanced datasets. The Precision-Recall curve plots precision (positive predictive value) against recall (sensitivity) for different threshold values. Precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

The AUPRC provides a single scalar value to summarize the curve, with higher values indicating better performance. It is particularly useful when the positive class is rare, as it focuses on the performance of the positive class [16].

These metrics together provide a comprehensive evaluation framework for our models, allowing us to capture different aspects of performance, particularly in the presence of class imbalance and multi-label scenarios.

A.5 Compute resources for experiments

We ran all computational experiments and pooling algorithms on Tesla V100-SXM2-16GB GPUs housed in the internal Sherlock cluster. For each task, the hyperparameter optimization experiments were limited to two days on GPU. On the same GPUs, we conducted unpublished preliminary experiments in developing the Pool PaRTI algorithm. We generated the precomputed ESM-2 token embeddings on NVIDIA A100-PCIE-40GB GPUs, also housed in the internal Sherlock cluster. Token embedding generation through the ESM-2 650M model took 70 GPU hours. We have computed token embeddings once for each sequence and performed several different pooling operations on the precomputed token embeddings before feeding the sequence embeddings as inputs to the respective models.