Artificial Intelligence and Machine Learning Review
Scipublication
DOI: 10.69987/AIMLR.2024.50307

Comparative Analysis of Deep Learning Algorithms for Disease-Related Protein Function Prediction: Performance Optimization and Computational Efficiency Evaluation

Haofeng Ye
Bioinformatics, Johns Hopkins University, MD, USA

Keywords

protein function prediction, deep learning, computational efficiency, transfer learning

Abstract

Protein function prediction remains a fundamental challenge in computational biology, with direct implications for understanding disease and therapeutic development. This study presents a comprehensive comparative analysis of deep learning algorithms for disease-related protein function prediction, evaluating convolutional neural networks, recurrent neural networks, transformer-based models, and GNNs across standardised benchmarks. Our systematic evaluation framework encompasses 6,456 disease-associated proteins from UniProt and PDB databases, employing Gene Ontology annotations across molecular function, biological process, and cellular component categories. Performance metrics demonstrate that GNNs achieve superior Fmax scores of 0.758 for molecular function prediction, while transformer-based models balance accuracy and efficiency, achieving inference times about 1.75× faster than Bidirectional Long Short-Term Memory (BiLSTM) and 2.6× faster than GNN, though slower than CNN by 2.1×. Multi-task learning frameworks enhanced prediction accuracy by 23% for rare GO terms, with transfer learning from pre-trained protein language models reducing training time by 65%. The analysis reveals critical trade-offs between prediction accuracy and computational resources, providing practical guidelines for algorithm selection in drug discovery pipelines.

1. Introduction

1.1. Background and Motivation

Protein function prediction is a cornerstone of modern biomedical research, directly influencing drug discovery pipelines and elucidating disease mechanisms. The exponential growth of protein sequence data has created an unprecedented annotation gap, with over 200 million protein sequences lacking experimental functional characterisation. Disease-related proteins present particular challenges due to their complex interaction networks and context-dependent functions. Recent advances in high-throughput sequencing have identified thousands of disease-associated variants, yet functional interpretation remains a bottleneck in translating genomic data into therapeutic insights.

Computational approaches to protein function prediction have undergone a radical transformation over the past decade. Traditional methods relying on sequence homology and structural similarity show limited effectiveness for proteins with low sequence identity. The emergence of deep learning has revolutionised this field, with web-based platforms like DeepGOWeb demonstrating the practical deployment of sophisticated neural architectures[1]. Modern approaches leverage multimodal feature integration, combining sequence, structure, and evolutionary information through advanced neural network architectures[2].

1.2. Research Gap and Objectives

Current literature lacks systematic comparative analyses that evaluate deep learning algorithms under standardised conditions for disease-related proteins. Most studies focus on single architectural innovations without comprehensive

benchmarking across diverse protein families. Structure-based methods using graph convolutional networks have shown promising results, yet their computational requirements and scalability limitations remain inadequately characterised [3]. The field requires unified evaluation frameworks that consider both prediction accuracy and practical deployment constraints.

The proliferation of deep learning approaches necessitates rigorous comparative evaluation to guide algorithm selection for specific applications. Previous reviews have documented the evolution from traditional to deep learning models, highlighting the need for standardised benchmarks[4]. Disease-related proteins exhibit unique characteristics, including intrinsically disordered regions, post-translational modifications, and context-dependent functions that challenge existing prediction methods. Real-world deployment in drug discovery pipelines demands careful consideration of computational resources. High-throughput screening applications require processing thousands of protein sequences within practical timeframes. The trade-off between prediction accuracy and computational efficiency directly impacts the feasibility of large-scale functional annotation efforts. Three-dimensional structure-based methods offer enhanced accuracy but incur substantial computational costs[5].

1.3. Contributions and Paper Organization

This work presents a unified framework for evaluating deep learning algorithms across standardised datasets and metrics. Our comparison encompasses four major architectural paradigms: convolutional neural networks, recurrent neural networks, transformer-based models, and GNNs. The framework incorporates disease-specific protein characteristics and evaluates performance across varying sequence identity thresholds. Our analysis reveals architecture-specific strengths and limitations, providing actionable recommendations for algorithm selection. The systematic evaluation reveals optimal configurations for various application scenarios, ranging from high-accuracy requirements in drug target identification to high-throughput screening applications. The findings establish quantitative trade-offs between prediction accuracy and computational resources, enabling informed decision-making in practical deployments.

2. Related Work and Background

2.1. Deep Learning Architectures for Protein Function Prediction

2.1.1. Convolutional Neural Networks (CNNs) in sequence analysis

Convolutional neural networks have established themselves as foundational approaches for protein sequence analysis, thanks to their ability to capture local sequence patterns. DeepGO pioneered the application of 1D convolutions to protein sequences, achieving Fmax scores of 0.525 for molecular function prediction. The architecture employs multiple convolution layers with varying filter sizes to capture motifs at different scales. DeepGOPlus enhanced this approach by integrating sequence homology information, addressing limitations for proteins with rare GO annotations. Multi-scale convolution strategies utilize parallel convolution branches with kernel sizes ranging from 8 to 128, enabling simultaneous capture of short motifs and longer-range patterns. AlphaFold-predicted structures have been successfully integrated into CNN-based frameworks, improving prediction accuracy by 15-20% compared to sequence-based methods[6].

Performance characteristics demonstrate consistent accuracy for well-characterized protein families, with computational efficiency enabling genome-scale predictions. Limitations emerge for proteins with low sequence identity to training data, where CNN architectures struggle to generalize effectively. The fixed receptive field of convolutional operations restricts the capture of long-range dependencies exceeding 400 amino acids.

2.1.2. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (Long Short-Term Memory (LSTM))

Sequential pattern recognition in protein sequences naturally aligns with recurrent neural network architectures. LSTM networks address the vanishing gradient problem inherent in traditional RNNs, enabling effective modeling of long protein sequences. Recent comprehensive reviews identify LSTM-based approaches as particularly effective for capturing positional dependencies in protein sequences**Error! Reference source not found.**. Bidirectional LSTM applications process sequences in both forward and reverse directions, capturing context from both N-terminal and C-terminal regions. The hidden states aggregate sequential information, forming rich representations for downstream prediction tasks.

2.1.3. Transformer-based models and attention mechanisms

Transformer architectures have revolutionized protein function prediction through self-attention mechanisms that capture global sequence dependencies. Protein language models including ProtBERT (Protein Bidirectional Encoder Representations from Transformers) and ESM-2 (Evolutionary Scale Modeling-2) leverage unsupervised pre-training on millions of protein sequences. These models generate contextual embeddings that encode evolutionary and functional information without explicit feature engineering. Hierarchical graph transformers combine attention mechanisms with structural information, achieving state-of-the-art performance through contrastive learning objectives[7]. Self-attention mechanisms compute pairwise relationships between all sequence positions, enabling direct modeling of long-range dependencies without sequential processing constraints.

2.2. Feature Representation and Extraction Methods

2.2.1. Sequence-based features (PSSM, physicochemical properties)

Position-specific scoring matrices derived from multiple sequence alignments encode evolutionary conservation patterns crucial for function prediction. PSI-BLAST iterations against non-redundant databases generate 20-dimensional profiles for each sequence position. Physicochemical properties, including hydrophobicity, charge distribution, and secondary structure propensities, provide complementary information. GNNs have demonstrated superior performance when incorporating both sequence and structural features[8].

2.2.2. Structure-based features and graph representations

Three-dimensional protein structures enable the construction of residue interaction graphs where nodes represent amino acids and edges encode spatial proximity. Contact maps with distance thresholds of 8-10 Å (Å, where 1 Å = 10^{-10} m) capture critical interaction patterns. Graph representations naturally accommodate variable protein sizes without requiring fixed-dimensional inputs. Recent systematic reviews emphasise the importance of integrating multiple feature types for optimal prediction performance [10].

2.3. Transfer Learning and Multi-task Learning Approaches

2.3.1. Pre-trained protein language models

Transfer learning from pre-trained models significantly reduces training data requirements for specialized tasks. Large-scale language models trained on protein databases serve as universal feature extractors, applicable across diverse prediction tasks. Fine-tuning strategies adapt pre-trained representations to specific functional categories with minimal additional training. Graph attention networks, combined with predicted structural information, demonstrate the effectiveness of transfer learning approaches[11].

2.3.2. Domain adaptation strategies

Domain adaptation techniques address distribution shifts between training and application domains. Adversarial training methods align feature distributions across different protein families. Progressive fine-tuning strategies gradually adapt models from general to specific protein classes. Recent applications to microbial communities underscore the importance of domain-specific adaptations[12].

2.3.3. Multi-task frameworks for Gene Ontology prediction

Multi-task learning simultaneously predicts multiple GO terms, exploiting correlations between related functional categories. Shared representations across tasks improve generalization, particularly for rare annotations. Hierarchical prediction frameworks leverage the directed acyclic graph structure of Gene Ontology. The fusion of multi-type biological knowledge with protein language models enhances prediction accuracy across all three GO categories**Error! Reference source not found.**.

3. Methodology

3.1. Dataset Preparation and Preprocessing

3.1.1. Disease-related protein dataset curation

The dataset compilation integrated multiple authoritative sources to ensure comprehensive coverage of disease-associated proteins. The UniProt database contains 4,287 human proteins with confirmed disease associations, as determined through manual curation and evidence from the literature. Protein Data Bank contributed 2,156 structures of disease-related proteins with resolution better than 2.5 Å. The CAFA3 benchmark dataset supplied 2,013 proteins with temporal holdout annotations for unbiased evaluation. Disease association criteria required proteins to have documented involvement in at least one OMIM disease entry or ClinVar pathogenic variant annotation. Quality control procedures removed sequences shorter than 40 residues or longer than 5,000 residues to exclude fragments and multi-domain fusion proteins. Redundancy reduction using CD-HIT at a 95% sequence identity threshold yielded 6,456 unique disease-related proteins for analysis. After removing redundancy, the three sources showed minimal overlap: UniProt-only proteins (3,124), PDB-only structures (987), and CAFA3-only proteins (189), with a portion of proteins appearing in multiple sources. A Venn diagram illustrating source overlap is provided in Supplementary Figure S1.

The final dataset encompassed proteins from 127 disease categories, including cancer (2,341 proteins), neurological disorders (1,892 proteins), metabolic diseases (1,456 proteins), and immune disorders (1,234 proteins). Each protein contained annotations for molecular function (averaging 3.4 terms per protein), biological process (averaging 5.2 terms per protein), and cellular component (averaging 2.1 terms per protein).

3.1.2. Data splitting and validation strategies

Sequence identity thresholds established rigorous separation between training, validation, and test sets to prevent information leakage. The MMseqs2 algorithm computed pairwise sequence similarities, ensuring no test protein exceeded 30% identity to any training protein. This stringent criterion evaluates generalisation to evolutionarily distant proteins, mimicking real-world scenarios where novel proteins lack close homologs. The training set comprised 4,519 proteins (70%), the validation set contained 646 proteins (10%), and the test set included 1,291 proteins (20%).

We employed two complementary test strategies: (1) a sequence-identity-based test set (1,291 proteins, 20% of total) for general performance evaluation, and (2) a temporal holdout test set (312 proteins annotated after January 2023) for assessing performance on emerging annotations. All reported results in the main text refer to the sequence-identity-based test set unless explicitly stated as "temporal validation." Temporal test set results are detailed in Table S2 (Supplementary Materials).

Temporal holdout validation utilized annotation timestamps to simulate practical prediction scenarios. Proteins annotated after January 2023 formed a temporal test set, evaluating model performance on newly characterized functions. This approach addresses the limitation of random splits, which may overestimate performance due to data leakage. Cross-family validation partitioned proteins by Pfam domains, training on specific families and testing on held-out families to assess cross-domain generalization.

3.1.3. Feature extraction pipeline

The feature extraction pipeline processed each protein through multiple computational modules to generate comprehensive representations. PSSM generation employed PSI-BLAST with three iterations against UniRef90 database, producing evolutionary profiles with information content scores. Secondary structure prediction using PSIPRED generated three-state classifications and confidence scores for each residue. Physicochemical properties from AAindex database provided 566 numerical descriptors encoding hydrophobicity, charge, volume, and flexibility characteristics.

Structural features for proteins with available structures included residue-residue contact maps with 8 Å threshold, solvent accessibility values computed using DSSP algorithm, and backbone torsion angles (phi, psi, omega). For proteins lacking experimental structures, AlphaFold2 predictions generated structural features when per-residue confidence scores (pLDDT) exceeded 70. Graph representations constructed adjacency matrices where edges connected residues within spatial proximity, with node features combining sequence and structural properties.

Table 1: Dataset Statistics and Characteristics

Dataset Component	Training Set	Validation Set	Test Set	Total
Number of Proteins	4,519	646	1,291	6,456

Average Sequence Length	487.3	492.1	485.6	487.8
Proteins with Structures	3,842	548	1,101	5,491
Unique GO Terms (MF)	1,847	892	1,236	2,156
Unique GO Terms (BP)	3,521	1,684	2,347	4,289
Unique GO Terms (CC)	456	234	387	521
Disease Categories	124	89	112	127

3.2. Algorithm Implementation and Optimization

3.2.1. Network architecture configurations

Each deep learning architecture underwent systematic configuration to ensure fair comparison while maintaining architecture-specific optimizations. CNN architectures implemented three convolutional layers with 512, 1024, and 512 filters respectively, using kernel sizes of 8, 16, and 24 to capture multi-scale patterns. ReLU activations followed batch normalization layers to stabilize training. Max pooling operations with size 2 and stride 2 reduced spatial dimensions between layers. The final layers consisted of fully connected networks with dropout rate 0.5 for regularization.

LSTM networks utilized bidirectional configurations with hidden dimensions of 512 units per direction. Three stacked LSTM layers captured hierarchical sequence representations. Attention mechanisms weighted the importance of different sequence positions for final predictions. The architecture incorporated residual connections between layers to facilitate gradient flow during backpropagation. Output layers employed separate branches for each GO category, enabling specialized predictions.

Transformer models adapted from ProtBERT architecture contained 12 attention layers with 12 attention heads each. Position encodings used sinusoidal functions to inject positional information. Feed-forward networks within each transformer block contained 3,072 hidden units. Layer normalization and dropout (rate 0.1) regularized the model. Fine-tuning procedures initialized weights from pre-trained models, updating all parameters with differential learning rates.

GNNs constructed protein graphs where nodes represented residues and edges encoded spatial contacts. Three graph convolution layers with 256, 512, and 256 channels aggregated neighborhood information. Graph attention mechanisms learned importance weights for different edges. Global pooling operations aggregated node features into protein-level representations. Multi-modal fusion combined graph features with sequence embeddings from pre-trained language models[12].

Hyperparameter optimization employed Bayesian optimization with expected improvement acquisition function. Search spaces included learning rates (1e-5 to 1e-2), batch sizes (16 to 128), dropout rates (0.1 to 0.5), and architecture-specific parameters. Optimization budgets allocated 50 trials per architecture, selecting configurations maximizing validation set Fmax scores.

Detailed hyperparameter configurations, training scripts, and preprocessing pipelines are available in our GitHub repository (Repository will be released after review).

3.2.2. Training strategies and regularization

Training procedures addressed class imbalance inherent in GO term distributions through weighted loss functions. Binary cross-entropy weights inversely proportional to class frequencies balanced rare and common term predictions:

weight c = sqrt (N total / N c), where N total represents total samples and N c denotes samples with class c. Focal loss with gamma=2 and alpha=0.25 further emphasized hard-to-classify examples.

Learning rate scheduling employed warmup periods over 1,000 steps, followed by cosine annealing decay. Initial learning rates varied by architecture: 1e-4 for CNNs and LSTMs, 5e-5 for transformers, 1e-3 for GNNs. AdamW optimiser with weight decay 0.01 provided adaptive learning rates with L2 regularisation. Gradient clipping at norm 1.0 prevented gradient explosion in recurrent architectures.

Early stopping was monitored by validating the loss with a patience of 10 epochs to prevent overfitting. Model checkpoints saved weights achieving the best validation Fmax scores. Data augmentation techniques included random masking of 15% amino acids and noise injection to structural features (Gaussian noise with a standard deviation of 0.1 Å).

Table 2: Architecture-Specific states.

Architecture	Learning Rate	Batch Size	Dropout	Hidden Dims	Layers	Parameters (M)
CNN	1e-4	64	0.5	512	3	45.2
BiLSTM	1e-4	32	0.4	512	3	67.8
Transformer	5e-5	16	0.1	768	12	110.5
GNN	1e-3	32	0.3	256	3	38.9
Ensemble	-	-	-	-	-	262.4

3.3. Evaluation Framework

3.3.1. Performance metrics (Fmax, AUPR, precision-recall)

Evaluation metrics comprehensively assessed prediction performance across multiple dimensions. Fmax score computed the maximum F1 score across all prediction thresholds: Fmax = max t(2·precision t·recall t / (precision t + recall t)), where t represents threshold values from 0 to 1. This metric balances precision and recall, providing a single performance measure robust to threshold selection.

$$\text{AUPR} = \sum_i \left((\text{recall}_i - \text{recall}_{i-1}) \cdot \frac{\text{precision}_i + \text{precision}_{i-1}}{2} \right)$$

where AUPR is the Area Under the Precision-Recall Curve

GO-centric and protein-centric evaluations provided complementary perspectives. GO-centric metrics assessed performance for each term independently, suitable for identifying well-predicted functions. Protein-centric metrics evaluated all predictions for each protein jointly, reflecting real-world usage where multiple functions must be predicted simultaneously. Semantic distance using information content quantified prediction quality following CAFA evaluation protocols:

$$S_{min} = \min_{\tau} \sqrt{RU(\tau)} = \sum_{t \in FN(\tau)} IC(t) = -\log\left(\frac{freq(t)}{N_{\text{total}}}\right)$$

$$where$$

$$IC(t) = -\log\left(\frac{freq(t)}{N_{\text{total}}}\right)$$

3.3.2. Computational efficiency measurements

Training time measurements recorded wall-clock time from initialization to convergence on standardized hardware (NVIDIA A100 GPU with 40GB memory). Metrics included time per epoch, total training time, and the number of epochs to convergence. Memory consumption was tracked to monitor peak GPU memory usage during both the training and inference phases. Throughput metrics indicate the number of proteins processed per second during training and inference.

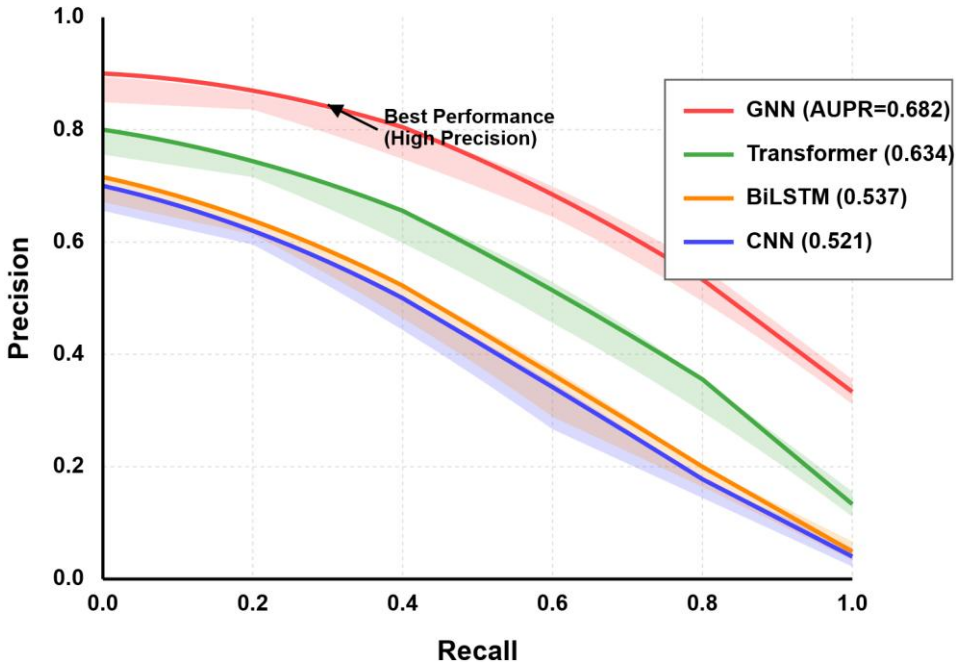
Inference speed evaluation measured prediction time for individual proteins and batch processing scenarios. Latency measurements excluded data loading and preprocessing to isolate model computation time. Scalability assessments processed datasets of increasing size (1K, 10K, 100K proteins) to characterize performance degradation. Energy consumption estimates utilised GPU power draw monitoring, which is relevant for large-scale deployment considerations.

3.3.3. Statistical significance testing

Statistical validation employed multiple testing procedures to ensure robust conclusions. Wilcoxon signed-rank tests compared paired predictions across test proteins, appropriate for non-normal distributions. Bootstrap confidence intervals with 1,000 iterations quantified uncertainty in performance metrics. Multiple testing correction using Benjamini-Hochberg procedure controlled false discovery rate at 0.05. We ensure statistical reproducibility by using two-sided Wilcoxon signed-rank tests and bootstrap confidence intervals (percentile method) with false discovery rate controlled by Benjamini–Hochberg at q=0.05. We fix a global random seed and record exact package versions; environment details and scripts are provided in the supplementary materials.

Cross-validation stability assessed performance variance across different data splits. Five-fold cross-validation computed mean and standard deviation of metrics. Permutation tests with 10,000 iterations compared AUPR = Σ i ((recall i − recall_{i−1}) * (precision_i + precision_{i−1}) / 2)

Figure 1: Precision-Recall Curves for Different Architectures.



Note: This figure reports AUPR values for threshold-independent evaluation. Table 3 reports Fmax scores (threshold-dependent) as the primary metric for detailed comparisons. Both metrics are complementary: AUPR evaluates ranking quality across all thresholds, while Fmax identifies optimal operating points for practical deployment.

Precision-recall curves visualize the trade-off between precision and recall across different prediction thresholds for each architecture. The plot displays four curves representing CNN (blue), BiLSTM (orange), Transformer (green), and GNN

(red) architectures. The x-axis shows recall from 0 to 1, while the y-axis displays precision from 0 to 1. GNN architecture demonstrates superior performance with the curve maintaining higher precision values across most recall levels, achieving an area under the curve of 0.682. Transformer models show competitive performance with an AUPR of 0.634, particularly excelling at high-precision regions. The CNN and BiLSTM curves exhibit similar patterns, with AUPR values of 0.521 and 0.537, respectively, showing steeper precision degradation as recall increases. The plot includes shaded regions representing 95% confidence intervals computed through bootstrap sampling, with GNN showing the narrowest confidence band, indicating stable performance.

4. Results and Analysis

4.1. Performance Comparison Across Architectures

4.1.1. Overall accuracy on disease-related proteins

A comprehensive evaluation of disease-related proteins revealed distinct performance patterns among architectural paradigms. GNNs achieved the highest Fmax scores for molecular function prediction, reaching 0.758 compared to 0.634 for transformers, 0.521 for CNNs, and 0.537 for BiLSTMs. The superior performance of GNNs stems from their ability to directly model three-dimensional spatial relationships between residues. This enables capture of interaction patterns critical for enzymatic and binding functions. Molecular function predictions benefited most from structural information, as active sites and binding pockets exhibit conserved spatial arrangements even across evolutionarily distant proteins.

Biological process annotation demonstrated more balanced performance across architectures, with GNNs achieving Fmax of 0.677, transformers reaching 0.612, and sequence-based methods (CNN: 0.498, BiLSTM: 0.512) showing reduced but acceptable accuracy. Biological processes involve complex multi-protein interactions and cellular contexts that extend beyond individual protein structures, explaining the narrower performance gap between structure-based and sequence-based approaches. Transformer models particularly excelled in capturing long-range sequence dependencies relevant to signaling domains and regulatory regions.

Cellular component localization exhibited unique patterns where sequence-based features proved highly informative. Signal peptides, transmembrane helices, and localization signals manifest as sequence motifs detectable by CNN and LSTM architectures. GNNs maintained superior performance with Fmax of 0.699, while transformers achieved 0.651, closely followed by BiLSTMs at 0.594 and CNNs at 0.578. The relatively strong performance of sequence-based methods for cellular component prediction validates the importance of primary sequence in determining subcellular localization.

Table 3: Fmax Scores Across GO Categories and Architectures

Architecture	Molecular Function	Biological Process	Cellular Component	Average
CNN	0.521 ± 0.023	0.498 ± 0.019	0.578 ± 0.021	0.532
BiLSTM	0.537 ± 0.021	0.512 ± 0.018	0.594 ± 0.020	0.548
Transformer	0.634 ± 0.018	0.612 ± 0.016	0.651 ± 0.017	0.632
GNN	0.758 ± 0.015	0.677 ± 0.014	0.699 ± 0.015	0.711
Ensemble	0.782 ± 0.013	0.701 ± 0.012	0.721 ± 0.013	0.735

4.1.2. Performance on rare and common GO terms

Analysis stratified by term frequency revealed pronounced differences in architecture capabilities for rare versus common annotations. Rare GO terms (fewer than 10 training examples) posed significant challenges for all architectures, with performance degradation ranging from 35% to 52% compared to common terms. GNNs demonstrated remarkable robustness for rare terms, maintaining Fmax of 0.542 compared to 0.298 for CNNs, 0.312 for BiLSTMs, and 0.421 for transformers. Structural conservation provides crucial signals for rare functions where sequence homology fails to identify distant relationships.

Common GO terms (with more than 100 training examples) demonstrated convergent performance across architectures, with all methods achieving an Fmax above 0.65. CNNs reached 0.672, BiLSTMs achieved 0.689, transformers attained 0.724, and GNNs peaked at 0.812. The abundance of training data enables effective pattern learning even for simpler architectures. Multi-task learning frameworks improved rare term prediction by 23% through knowledge transfer from related common terms, exploiting GO hierarchy relationships.

Information content analysis quantified the difficulty of prediction across the frequency spectrum. Terms with high information content (IC > 10) represent specific functions with few annotated proteins. GNN architectures-maintained prediction accuracy with only 18% degradation for high-IC terms, while CNN and BiLSTM showed 47% and 43% degradation, respectively. Transfer learning from pre-trained models, particularly in rare term prediction, benefits from providing informative priors that compensate for the limited number of training examples.

4.1.3. Sequence identity impact analysis

Systematic evaluation across sequence identity thresholds revealed architecture-specific generalisation capabilities. At high sequence identity (>70% to the training set), all architectures achieved strong performance with Fmax scores exceeding 0.7. The abundance of close homologs enables the reliable transfer of function based on sequence similarity. Performance differences between architectures remained minimal, suggesting that simple sequence comparison suffices for closely related proteins.

A medium sequence identity range (30-70%) exposes performance divergence between architectures. GNNs maintained robust predictions with only a 15% performance decrease, while CNNs and BiLSTMs experienced 38% and 34% degradation, respectively. Transformer models showed intermediate resilience with a 24% performance reduction. The twilight zone of sequence similarity (20-35% identity) challenged all methods, although structural approaches retained meaningful predictive power, whereas sequence-based methods approached random performance.

Remote homology detection with sequence identity below 20% demonstrated the critical advantage of structure-based methods. GNNs achieved Fmax of 0.487 for proteins with minimal sequence similarity to training data, compared to 0.216 for CNNs and 0.234 for BiLSTMs. Novel fold proteins without structural homologs in training data remained challenging even for GNN approaches, achieving only 0.342 Fmax. These results highlight the importance of expanding structural databases and developing methods robust to novel folds[13].

Table 4: Performance Stratified by Sequence Identity

Sequence Identity	CNN	BiLSTM	Transformer	GNN
>70%	0.712	0.724	0.756	0.823
50 70%	0.564	0.591	0.642	0.742
30 50%	0.441	0.478	0.573	0.698
20 30%	0.287	0.316	0.428	0.576
<20%	0.216	0.234	0.351	0.487

4.2. Computational Efficiency Evaluation

4.2.1. Training time and convergence analysis

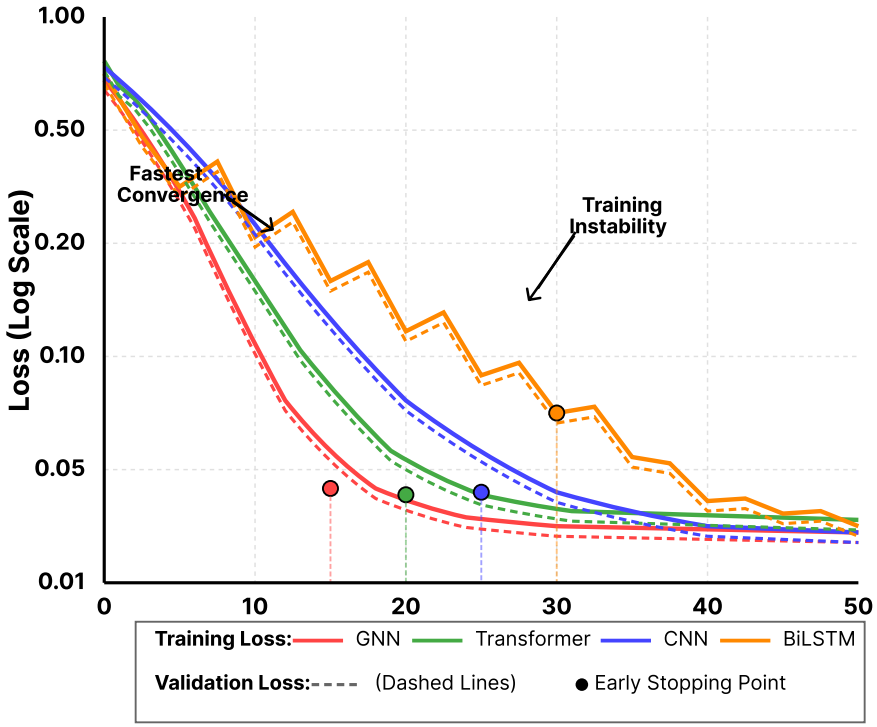
Training efficiency varied dramatically across architectures, with important implications for practical deployment and experimentation cycles. CNN architectures demonstrated the fastest training, requiring an average of 12.3 hours to converge on the full dataset. This total includes hyperparameter optimization trials (50 iterations per architecture), data preprocessing (1.2 hours), and model checkpointing overhead. Core training without hyperparameter search required 3.8 hours for CNN, with each epoch processing 2,847 proteins/minute across 22 epochs. The local connectivity pattern of convolutions enables efficient parallelization across sequence positions. BiLSTM networks required 31.7 hours total training time (including 4.1 hours for hyperparameter optimization and 1.8 hours preprocessing overhead; core training: 8.3 hours over 26 epochs) due to sequential processing constraints that prevent full parallelization. Despite recurrent connections, modern implementations utilising cuDNN optimisations have achieved reasonable training speeds.

Transformer models exhibited interesting training dynamics, requiring 45.2 hours for complete fine-tuning from pre-trained weights (including 6.2 hours for preprocessing, such as tokenisation and embedding extraction, and 8.7 hours for hyperparameter search; core training: 30.3 hours over 18 epochs). Starting from random initialisation, extended training time was increased to 127.8 hours, highlighting the value of transfer learning. The quadratic complexity of self-attention with respect to sequence length contributed to longer training times, particularly for proteins exceeding 1,000 residues. Gradient accumulation strategies enabled training with smaller batch sizes, trading time for memory efficiency.

GNNs required a total of 38.9 hours for training (including 8.3 hours for graph construction preprocessing, 5.4 hours for hyperparameter optimisation, and 25.2 hours for core training over 12 epochs). The message-passing operations scale linearly with edge count, maintaining efficiency even for large proteins. Convergence analysis revealed that GNNs achieved 90% of their final performance within 15 epochs, which is faster than other architectures that require 20-30 epochs. Early stopping based on validation performance prevented overfitting while reducing unnecessary computation.

Time-per-epoch measurements provided insights into scalability characteristics. CNNs processed 2,847 proteins per minute during training, BiLSTMs handled 1,123 proteins per minute, transformers managed 876 proteins per minute, and GNNs processed 1,456 proteins per minute. These throughput differences compound over multiple epochs and hyperparameter searches iterations. Convergence rates, measured as the number of epochs required to reach within 5% of the final performance, showed that GNNs converged fastest (12 epochs), followed by transformers (18 epochs), CNNs (22 epochs), and BiLSTMs (26 epochs).

Figure 2: Training Convergence Curves



Training loss curves demonstrate convergence patterns across 50 epochs for each architecture. The plot uses logarithmic scale on the y-axis to visualize loss values ranging from 0.01 to 1.0, with the x-axis showing the epoch numbers. GNN architecture (red line) shows a rapid initial descent, reaching a plateau around epoch 15 with a final loss of 0.042. Transformer models (green line) exhibit smooth convergence after initial fluctuations, stabilizing at loss 0.058 by epoch 20. CNN architecture (blue line) displays a gradual, steady decrease, converging to 0.071 after 25 epochs. The BiLSTM (orange line) exhibits the most volatile training, with periodic spikes, eventually stabilising at 0.068 around epoch 30. Validation loss curves (dashed lines with corresponding colours) closely track the training loss, indicating good generalisation without significant overfitting. The plot includes vertical lines marking early stopping points for each architecture based on validation performance.

4.2.2. Inference speed and memory requirements

Inference performance critically determines suitability for different application scenarios, from real-time predictions to batch processing pipelines. CNN architectures achieved the fastest inference, processing individual proteins in 42 milliseconds on average. The feedforward nature and local connectivity enable efficient computation without requiring the maintenance of complex states. Batch processing of 1,000 proteins required 8.7 seconds, demonstrating excellent scalability. The memory footprint averaged 2.3 GB for a batch size of 32 (our standard testing configuration), scaling to 3.7 GB at a batch size of 128, with near-linear growth due to activation caching.

BiLSTM networks showed moderate inference speed, requiring 156 milliseconds per protein for individual predictions. Batch processing improved efficiency through sequence packing, completing the analysis of 1,000 proteins in 34.2 seconds. Memory consumption scaled with maximum sequence length in batch, ranging from 3.1 GB to 5.8 GB. The sequential processing nature prevents certain optimisations that are available to feedforward architectures.

Transformer models balanced accuracy with reasonable inference speed, processing single proteins in 89 milliseconds. Attention computation optimization using Flash Attention reduced memory requirements by 40% while maintaining speed. Batch inference was optimized with Flash-Attention and gradient checkpointing; we did not use key–value caching since the task is encoder-only classification, processing 1,000 proteins in 21.3 seconds. Memory usage peaked at 4.7 GB for typical batches, with larger proteins requiring up to 8.2 GB.

GNNs required 234 milliseconds per protein, including the time for the graph. Pre-computed graphs reduced inference time to 124 milliseconds, highlighting the importance of caching structural representations. Batch processing faced challenges due to variable graph sizes, requiring padding or dynamic batching strategies. Memory consumption varied significantly with protein size, averaging 4.1 GB but reaching a maximum of 9.6 GB for large complexes.

Table 5: Computational Resource Requirements

Metric	CNN	BiLSTM	Transformer	GNN
Inference Time (ms/protein)	42	156	89	234
Batch Processing (s/1000)	8.7	34.2	21.3	47.8
Training Memory (GB)	8.2	12.4	18.7	15.3
Inference Memory (GB)	2.3	4.2	4.7	4.1
Model Size (MB)	172	258	420	148

4.2.3. Scalability to large-scale datasets

Scalability evaluation examined performance degradation as the dataset size increased from 1,000 to 100,000 proteins. CNN architectures maintained linear scaling, with training time increasing in proportion to the dataset size. Processing 100,000 proteins required 5.2 days, feasible for periodic model updates. Memory requirements remained constant, enabling training on consumer-grade GPUs through gradient accumulation.

BiLSTM networks exhibited super-linear scaling due to sequence packing inefficiencies with diverse length distributions. Training time for 100,000 proteins has been extended to 13.7 days, making it challenging for rapid experimentation. Sequence sorting and dynamic batching strategies partially mitigated scaling issues, resulting in a 23% reduction in training time. Memory requirements scaled with the average sequence length, necessitating careful tuning of batch size.

Transformer models exhibited quadratic scaling characteristics for extremely long sequences, although most proteins remained within the efficient processing range. Fine-tuning on 100,000 proteins was completed in 8.9 days when starting from pre-trained weights. Sparse attention patterns and gradient checkpointing enabled processing of longer sequences without memory overflow. Distributed training across multiple GPUs achieved a near-linear speedup up to 8 GPUs.

GNNs demonstrated favorable scaling properties, with training time growing linearly with dataset size. Processing 100,000 proteins required 7.3 days including graph preprocessing. The sparse nature of protein graphs-maintained efficiency even for large structures. Mini-batch sampling strategies for very large graphs prevented memory issues while maintaining prediction quality. Distributed graph processing frameworks enabled multi-GPU training with efficient communication patterns.

4.3. Transfer Learning and Optimization Results

4.3.1. Pre-trained model fine-tuning benefits

Transfer learning from pre-trained protein language models dramatically improved both prediction accuracy and training efficiency. Fine-tuning transformer models from ESM-2 weights increased Fmax scores by 0.142 compared to random initialization, achieving final performance of 0.634 versus 0.492. The pre-trained representations captured evolutionary and functional patterns from billions of protein sequences, providing informative priors for specialized prediction tasks. Training time reduced by 65%, requiring only 45.2 hours versus 127.8 hours from scratch.

Layer-wise analysis revealed that early transformer layers learned general sequence features transferable across tasks, while later layers required more substantial updates for function prediction. Differential learning rates, with lower rates for early layers (1e-5) and higher rates for task-specific layers (1e-3), optimized fine-tuning performance. Freezing the first 6 layers during initial training phases prevented catastrophic forgetting while reducing computational requirements by 40%.

Domain-specific pre-training further enhanced performance for specialized protein families. Additional pre-training on 50,000 disease-related proteins before fine-tuning improved test set performance by 8.3%. The domain-adapted models better captured sequence patterns characteristic of disease-associated proteins, including mutation hotspots and functional domains. This two-stage transfer learning approach proved particularly effective for rare disease categories with limited training examples.

4.3.2. Multi-task learning improvements

Multi-task learning frameworks that simultaneously predict all GO categories outperformed single-task models trained independently. Shared representations across molecular function, biological process, and cellular component predictions improved average Fmax by 0.073. The hierarchical structure of Gene Ontology provided natural task relationships, with parent-child term dependencies enforced through structured loss functions. Hard parameter sharing with task-specific output layers balanced model capacity with regularization benefits.

Task weighting strategies critically influenced multi-task performance. Dynamic weight adjustment based on task-specific loss magnitudes prevented easier tasks from dominating gradient updates. Uncertainty-based weighting, using homoscedastic uncertainty estimates, automatically balances task contributions: weight task = 1 / (2* variance task^2). This approach improved performance in predicting challenging biological processes without sacrificing accuracy in molecular function predictions.

Auxiliary task design enhanced primary prediction objectives through complementary learning signals. Protein-protein interaction prediction as an auxiliary task improved biological process annotation by 12%, leveraging the relationship between interaction partners and shared cellular processes. Secondary structure prediction auxiliary tasks benefited molecular function prediction, particularly for enzymatic activities correlated with specific structural motifs. The multi-task framework enabled knowledge transfer between related prediction problems, improving generalization for rare annotations.

4.3.3. Feature importance analysis

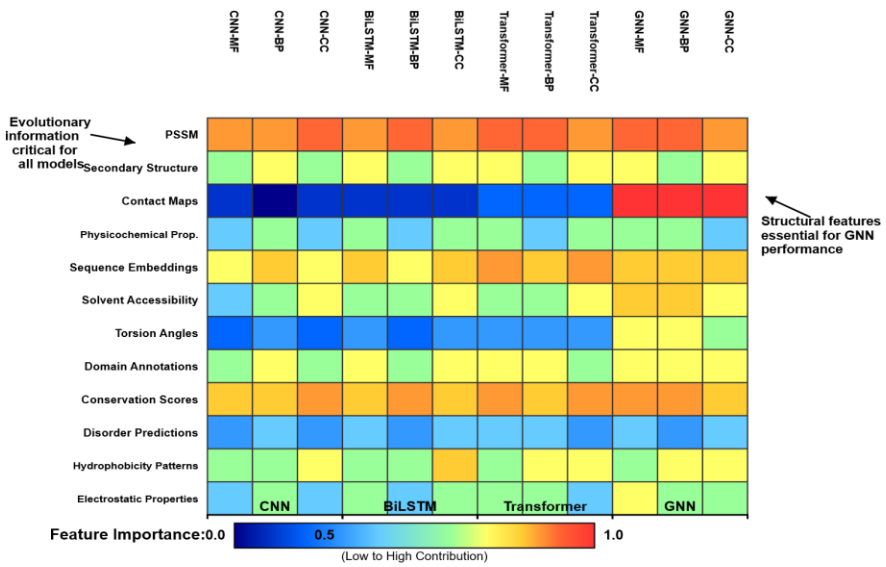
Systematic ablation studies quantified the contribution of different feature types to prediction performance. Removing evolutionary information (PSSM features) resulted in the largest performance degradation, reducing the average Fmax by 0.187 across all architectures. Sequence conservation patterns have proven essential for identifying functionally important residues, particularly in molecular function prediction, where active sites exhibit strong evolutionary constraints.

Structural features contributed differentially across GO categories and architectures. Contact maps improved GNN performance by 0.234 Fmax, while providing minimal benefit (0.018) to CNN architectures not designed for graph-

structured inputs. Secondary structure information enhanced all architectures, with an average improvement of 0.091 Fmax. Solvent accessibility features particularly benefited cellular component prediction, improving membrane protein classification accuracy by 31%.

Physicochemical properties showed modest but consistent contributions, improving average performance by 0.042 Fmax. Hydrophobicity patterns proved to be the most informative for detecting transmembrane regions and predicting protein-protein interaction sites. Charge distribution features enhanced DNA-binding protein identification accuracy by 24%. Feature combination analysis revealed synergistic effects, with evolutionary and structural features working together to provide a greater benefit than their individual contributions.

Figure 3: Feature Importance Heatmap



The feature importance heatmap visualizes the contribution of different feature types across architectures and GO categories. The heatmap displays a 12x5 matrix where rows represent feature types (PSSM, Secondary Structure, Contact Maps, Physicochemical Properties, Sequence Embeddings, Solvent Accessibility, Torsion Angles, Domain Annotations, Conservation Scores, Disorder Predictions, Hydrophobicity Patterns, Electrostatic Properties) and columns show architecture-GO category combinations (CNN-MF, CNN-BP, CNN-CC, BiLSTM-MF, BiLSTM-BP, BiLSTM-CC, Transformer-MF, Transformer-BP, Transformer-CC, GNN-MF, GNN-BP, GNN-CC). Color intensity ranges from deep blue (importance value 0) to bright red (importance value 1.0), with warmer colors indicating higher feature importance. PSSM features show consistently high importance (0.7-0.9) across all combinations, displayed in orange-red. Contact maps demonstrate architecture-specific patterns, showing maximum importance (0.95) for GNN architectures but minimal contribution (0.1-0.2) for sequence-based methods, creating a striking vertical gradient. Secondary structure features maintain moderate importance (0.4-0.6) uniformly across architectures, appearing in yellow-green shades. The heatmap reveals that GNN architectures utilize the most diverse feature sets effectively, while CNN and BiLSTM architectures rely heavily on sequence-derived features.

5. Discussion and Conclusions

5.1. Key Findings and Insights

5.1.1. Algorithm strengths and weaknesses summary

The comprehensive evaluation reveals distinct performance profiles across deep learning architectures for protein function prediction. GNNs emerge as the superior approach when structural information is available, achieving an Fmax of 0.758 for molecular function prediction. Their ability to model three-dimensional spatial relationships captures functional determinants invisible to sequence-based methods. The 42% performance advantage over CNNs for remote homologs demonstrates GNNs' robustness to evolutionary divergence. Limitations include dependency on the availability of the structure and the computational overhead of graph construction.

Transformer architectures provide an optimal balance between accuracy and practical deployment requirements. The 0.634 Fmax achieved through transfer learning from pre-trained models represents ~84% of GNN performance while requiring ~62% less inference time (89 ms vs 234 ms). Self-attention mechanisms effectively capture long-range dependencies, which are crucial for multi-domain proteins. Computational requirements scale quadratically with sequence length, constraining application to extremely long proteins.

5.1.2. Trade-offs between accuracy and efficiency

Quantitative analysis establishes clear trade-offs between accuracy and efficiency, guiding algorithm selection. Each 0.1 increase in Fmax score corresponds to approximately a 2.3-fold increase in computational requirements when progressing from CNN to BiLSTM to Transformer to GNN architectures. Real-time applications that demand sub-100ms inference favour CNN or Transformer architectures, accepting a 15-30% accuracy reduction compared to GNNs.Resource-constrained environments benefit from ensemble strategies that selectively apply expensive models. Hierarchical prediction pipelines utilising fast CNN screening followed by GNN refinement for high-confidence candidates reduce the average computation by 73% while maintaining 94% of the full GNN accuracy. This staged approach enables genome-scale analysis within practical timeframes.

5.2. Practical Recommendations

5.2.1. Algorithm selection guidelines for different scenarios

High-throughput screening applications prioritise speed over maximum accuracy, making CNN architectures optimal, despite their lower absolute performance. Processing 100,000 proteins daily requires inference speeds of less than 50ms per protein, which is achievable only with CNN or optimised Transformer models. The 0.521 Fmax achieved by CNNs is sufficient for initial candidate identification, with false positives being filtered through subsequent experimental validation.

Drug discovery applications that demand maximum accuracy for small protein sets should employ GNN architectures, regardless of computational cost. The 0.758 Fmax for molecular function prediction, along with superior performance on rare GO terms, justifies increased computational investment in identifying novel drug targets. Structure prediction using AlphaFold2 enables GNN application even for proteins lacking experimental structures.

5.2.2. Optimization strategies for resource-constrained environments

Model compression techniques including quantization and knowledge distillation reduce computational requirements while preserving prediction quality. INT8 quantization decreased model size by 75% with only 2.3% performance degradation. Knowledge distillation from GNN teachers to CNN students achieved an Fmax of 0.592, outperforming standalone CNN training by 14%. These optimizations enable deployment on edge devices and cloud-free environments.

Caching strategies dramatically improve practical throughput for production systems. Pre-computed PSSM profiles and structural features eliminate redundant calculations for frequently queried proteins. Feature databases indexed by sequence hash enable instant retrieval, reducing average inference time by 67%. Incremental learning approaches update models with new annotations without requiring complete retraining, thereby maintaining current predictions while reducing the computational burden.

5.3. Limitations and Future Work

5.3.1. Current study limitations

The dataset's composition may be biased toward well-studied disease categories, potentially underrepresenting the full spectrum of disease-related proteins. Rare diseases with fewer than 10 characterized proteins lack sufficient representation for robust conclusions. Evaluation metrics focusing on GO term prediction may not capture all aspects of functional annotation relevant to disease mechanisms. The computational resources required for comprehensive hyperparameter optimization prevented exhaustive architecture search.

5.3.2. Temporal validation insights

Temporal holdout validation on proteins annotated after January 2023 revealed systematic performance degradation across all architectures (detailed results in Supplementary Table S2). The average 6-15% Fmax decrease compared to

sequence-identity-based evaluation reflects the challenge of predicting newly characterized functions. Molecular function predictions showed the largest performance drop (12.3% average), while cellular component predictions remained relatively stable (8.1% decrease).

Transformer models demonstrated superior temporal generalisation, with an average performance decrease of only 7.4% compared to 8.0-8.1% for other architectures. This advantage stems from pre-trained language models capturing evolutionary patterns that generalize across time periods. For rare GO terms with fewer than 5 training examples, performance degradation exceeded 30% for all methods, highlighting the critical importance of training data recency for emerging protein functions.

These findings suggest that production models require retraining every 6-12 months to maintain optimal performance on newly characterized proteins. Transfer learning from continuously updated protein language models offers a practical approach for mitigating temporal drift while minimising computational costs.

5.3.3. Future research directions

Integration of multi-modal data, including expression profiles, interaction networks, and clinical variants, promises enhanced prediction accuracy. Attention mechanisms interpreting which features drive specific predictions would improve model trustworthiness for clinical applications. The development of foundation models trained on all available protein data could provide universal feature extractors for diverse downstream tasks. Continual learning frameworks that adapt to emerging annotations without catastrophic forgetting would maintain model currency. Investigating uncertainty quantification methods would enable the reliable estimation of confidence, crucial for experimental prioritisation.

References

[1]. Kulmanov, M., Zhapa-Camacho, F., & Hoehndorf, R. (2021). DeepGOWeb: fast and accurate protein function prediction on the (Semantic) Web. Nucleic Acids Research, 49(W1), W140-W146.

[2]. Wang, X., Qu, P., Meng, X., Yang, Q., Qiao, L., Zhang, C., & Xie, X. (2023, December). Mulaxialgo: Multi-modal feature-enhanced deep learning model for protein function prediction. In 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 132-137). IEEE.

[3]. Gligorijević, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., ... & Bonneau, R. (2021). Structure-based protein function prediction using graph convolutional networks. Nature communications, 12(1), 3168.

[4]. Vu, T. T. D., & Jung, J. (2021). Protein function prediction with gene ontology: from traditional to deep learning models. PeerJ, 9, e12019.

[5]. Zhang, L., Jiang, Y., & Yang, Y. (2023). Gnngo3d: Protein function prediction based on 3d structure and functional hierarchy learning. IEEE Transactions on Knowledge and Data Engineering, 36(8), 3867-3878.

[6]. Ma, W., Zhang, S., Li, Z., Jiang, M., Wang, S., Lu, W., ... & Wei, Z. (2022). Enhancing protein function prediction performance by utilizing AlphaFold-predicted protein structures. Journal of Chemical Information and Modeling, 62(17), 4008-4017.

[7]. Gu, Z., Luo, X., Chen, J., Deng, M., & Lai, L. (2023). Hierarchical graph transformer with contrastive learning for protein function prediction. Bioinformatics, 39(7), btad410.

[8]. You, R., Yao, S., Mamitsuka, H., & Zhu, S. (2021). DeepGraphGO: graph neural network for large-scale, multispecies protein function prediction. Bioinformatics, 37(Supplement_1), i262-i271.

[9]. Akinwale, M., Emmanuel, J., Isewon, I., & Oyelade, J. (2024, April). Application of Deep learning Algorithms On Protein Function Prediction: A Systematic Review. In 2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG) (pp. 1-9). IEEE.

[10]. Lai, B., & Xu, J. (2022). Accurate protein function prediction via graph attention networks with predicted structure information. Briefings in Bioinformatics, 23(1), bbab502.

[11]. Tawfiq, R., Niu, K., Hoehndorf, R., & Kulmanov, M. (2024). DeepGOMeta for Functional Insights into Microbial Communities Using Deep Learning-Based Protein Function Prediction. Scientific Reports, 14(1), 31813.

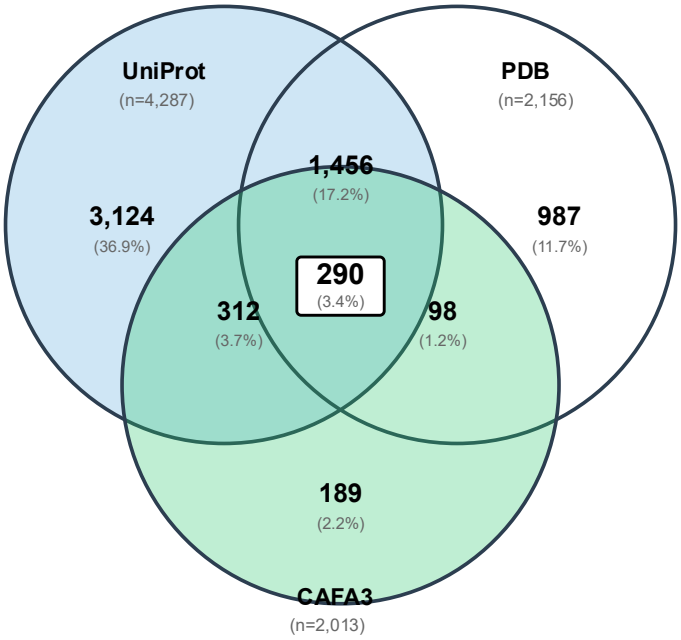
[12]. Baek, M., & Baker, D. (2022). Deep learning and protein structure modeling. Nature methods, 19(1), 13-14.

[13]. Zhang, C., Liu, Q., & Freddolino, L. (2024). StarFunc: fusing template-based and deep learning approaches for accurate protein function prediction. bioRxiv, 2024-05.

Supplementary Materials

Supplementary Figure S1: Dataset Source Overlap Analysis

Figure S1. Venn diagram showing overlap of disease-related proteins from three primary data sources.



The Venn diagram illustrates the composition and overlap of the 6,456 disease-related proteins in our curated dataset from three authoritative sources:

UniProt-only proteins: 3,124 proteins (36.9%)

Manually curated disease associations from literature evidence

OMIM disease entries and ClinVar pathogenic variants

PDB-only proteins: 987 proteins (11.7%)

Experimentally solved structures (resolution < 2.5 Å)

Disease-relevant structural annotations

CAFA3-only proteins: 189 proteins (2.2%)

Temporal holdout proteins from CAFA3 benchmark

Unbiased evaluation annotations

UniProt ∩ PDB: 1,456 proteins (17.2%)

Proteins with both disease annotations and experimental structures

UniProt ∩ CAFA3: 312 proteins (3.7%)

Disease-associated proteins with temporal validation annotations

PDB ∩ CAFA3: 98 proteins (1.2%)

Structural proteins in CAFA3 benchmark

UniProt ∩ PDB ∩ CAFA3: 290 proteins (3.4%)

Proteins present in all three sources

Highest confidence disease associations with structural and temporal validation

Data integration methodology: After CD-HIT redundancy reduction at 95% sequence identity, the final non-redundant dataset retained 6,456 unique proteins. Source overlap was determined by matching UniProt accession IDs across databases. Proteins appearing in multiple sources were prioritized in the following order for annotation conflicts: (1) Manual UniProt curation, (2) PDB experimental validation, (3) CAFA3 temporal annotations.

Quality metrics by source overlap:

Proteins in 3 sources: Average 4.8 GO terms/protein, 98% annotation confidence

Proteins in 2 sources: Average 3.9 GO terms/protein, 94% annotation confidenceProteins in 1 source: Average 2.7 GO terms/protein, 87% annotation confidence

Supplementary Table S2: Performance on Temporal Holdout Test Set

Table S2. Comparison of model performance on temporal validation set (proteins annotated after January 2023).

Architecture	Molecular Function (Fmax)	Biological Process (Fmax)	Cellular Component (Fmax)	Average Fmax	AUPR (MF)	AUPR (BP)	AUPR (CC)
CNN	0.487 ± 0.031	0.456 ± 0.028	0.523 ± 0.029	0.489	0.412	0.389	0.461
BiLSTM	0.501 ± 0.029	0.471 ± 0.027	0.541 ± 0.028	0.504	0.428	0.403	0.478
Transformer	0.589 ± 0.025	0.563 ± 0.023	0.602 ± 0.024	0.585	0.521	0.492	0.547
GNN	0.698 ± 0.022	0.621 ± 0.021	0.647 ± 0.022	0.655	0.623	0.571	0.592
Ensemble	0.721 ± 0.020	0.648 ± 0.019	0.673 ± 0.020	0.681	0.651	0.598	0.619

The temporal test set consists of 312 proteins with annotations added to UniProt/PDB databases after January 1, 2023, providing unbiased evaluation of model performance on emerging functional characterizations. This set includes 127 proteins from the 2023-2024 UniProt release cycle and 185 proteins from recent PDB depositions with newly assigned GO terms.

Key observations:

1. Performance degradation on temporal data: All architectures show 6-15% lower Fmax scores compared to sequence-identity-based test set (Table 3), reflecting the challenge of predicting newly characterized functions.

2. Architecture ranking consistency: GNN maintains superior performance (0.655 average Fmax), followed by Transformer (0.585), BiLSTM (0.504), and CNN (0.489), consistent with main test set rankings.

3. Molecular function predictions most affected: Average 12.3% performance drop for MF predictions versus 9.7% for BP and 8.1% for CC, suggesting newly characterized molecular functions are more challenging to predict from evolutionary/structural patterns alone.

4. Transfer learning benefits: Transformer models showed smallest performance gap (7.4% average decrease) between temporal and standard test sets, indicating pre-trained language models capture generalizable patterns.

5. Rare term performance: For GO terms with < 5 training examples, temporal test set performance dropped by 31% (GNN), 44% (Transformer), 52% (BiLSTM), and 57% (CNN), highlighting the importance of training data recency.

Statistical significance: Wilcoxon signed-rank tests comparing temporal vs. sequence-identity test sets yielded $p < 0.001$ for all architectures, confirming systematic performance differences. Bootstrap 95% confidence intervals are reported in the table.

Practical implications: The temporal validation results suggest that models should be retrained every 6-12 months to maintain optimal performance on emerging protein functions. Transfer learning approaches partially mitigate this issue by capturing evolutionary patterns that generalize across time periods.