# Advanced Heterogeneous Network-Based Graph Neural Network Framework for Predicting Anti-CRISPR Protein Sequences

Yeqiang Wang, Wenxiao Zhao, Yijun He, Jiale Li, and Rui Mao

***Abstract*—Anti-CRISPR proteins play a crucial role in bacterial-phage interactions by inhibiting the CRISPR/Cas system and thus enhancing phage survival. Accurately predicting these proteins is essential for understanding phage-host immune interactions and progressing CRISPR/Cas-based technologies. Current approaches primarily analyze proteins individually, which may overlook the intrinsic similarities and potential connections among protein sequences. This study introduces PACRGNN, a graph neural network framework that creates a heterogeneous protein network by integrating sequence and structural similarities, wherein nodes represent proteins and edges signify their relationships. By combining Graph Attention (GAT) and Graph Sample and Aggregation (GraphSAGE) layers, PACRGNN captures both local and global topological dependencies, while incorporating six protein feature categories to enrich node representations. PACRGNN achieves an accuracy of 0.9577, an F1-Score of 0.9572, and a PRAUC of 0.9876 on the validation set. The model demonstrated superior performance to existing methods on the independent test set derived from NCBI database (Jan.–Oct. 2024).**

***Index Terms*—Anti-CRISPR prediction, CRISPR/Cas system, graph neural networks, heterogeneous protein network.**

## I. Introduction

BACTERIA and phages engage in a constant, competitive struggle for survival [1]. Bacteria defend themselves against phage infection through the acquisition of an immune system, CRISPR/Cas, which enables them to recognize and defend against invading genetic elements [2], [3]. In response, phages have countered by evolving Anti-CRISPR (Acr) proteins that inhibit the bacterial CRISPR/Cas defenses [4], [5]. Recent study highlights phages' ability to finely regulate Acr proteins expression, balancing CRISPR inhibition with minimizing host cell toxicity, a critical factor for successful infection [6]. The CRISPR/Cas systems, celebrated for their breakthroughs in gene editing and potential in various scientific fields, nonetheless face challenges related to safety concerns such as off-target effects and cytotoxicity [7], [8]. In this context, Acr proteins emerge as natural inhibitors, offering strategies to enhance gene editing precision by specifically interacting with Cas protein [9]. The urgent need to identify and predict Acr proteins accurately is thus underscored, pressing for methods that enhance our understanding of bacterial-phage dynamics and bolster CRISPR/Cas system safety in gene editing applications.

Current methods for identifying Acr proteins can be broadly classified into two categories. The first category relies heavily on experimental evidence or biological knowledge, utilizing methods such as the "guilt by association" approach [10] or self-targeting spacer analysis [11]. While effective, these methods are constrained by their dependence on prior knowledge, limiting their ability to address the diverse functional range of Acr proteins [12], [13]. The second category encompasses machine learning (ML) and deep learning (DL) techniques, which can efficiently identify Acr proteins without prior biological knowledge. Among ML-based methods, AcRanker employed the XGBoost algorithm to construct 412-dimensional feature vectors by analyzing amino acid composition and dimer/trimer frequency counts. When tested on an independent dataset of 20 known Acr proteins, AcRanker successfully ranked 8 out of 13 Acrs located in phage or prophage regions within the top 10 positions. Additionally, using known Acr proteins as a training set, AcRanker identified two novel Acr proteins, AcrIIA20 and AcrIIA21 [14]. PaCRISPR adopted an ensemble learning strategy, leveraging PSI-BLAST to identify remote homologous proteins in the UniRef50 [15] database and generate Position-Specific Scoring Matrices (PSSMs). These matrices captured evolutionary features such as PSSM-composition, DPC-PSSM, PSSM-AC, and RPSSM. PaCRISPR then trained multiple Support Vector Machine (SVM) classifiers and integrated predictions using an ensemble framework, achieving an accuracy of 0.85, an F1-Score of 0.85, and an MCC of 0.70 on its independent test set [12]. AcrPred introduced an innovative two-step fusion

Received 18 December 2024; revised 12 February 2025; accepted 26 February 2025. This work was supported in part by the Key Research and Development Program of Shaanxi Province under Grant 2024NC-ZDCYL-05-06 and Grant 2024NC-ZDCYL-05-11 and in part by the National Innovation Training Program for College Students under Grant 202410712075. *(Corresponding author: Rui Mao.)*

Yeqiang Wang, Yijun He, Jiale Li, and Rui Mao are with the College of Information Engineering, Northwest A&F University, Xianyang 712100, China (e-mail: maorui@nwafu.edu.cn).

Wenxiao Zhao is with the College of Animal Science and Technology, Northwest A&F University, Xianyang 712100, China.

A user-friendly online platform (https://www.acrs.top) facilitates practical applications.

Datasets and source code are available at https://github.com/WangYeQianger/PACRGNN.

Digital Object Identifier 10.1109/JBHI.2025.3548463

TABLE I
SUMMARY OF THE RESEARCHES FOR ANTI-CRISPR PREDICTION

| Method | Year | Feature encoding scheme | Algorithm | Evaluation metrics | Webserver |
|---|---|---|---|---|---|
| AcRanker | 2020 | AAC, GDC, GTC | XGBoost | Rank | Decommissioned |
| PaCRISPR | 2020 | PSSM features, AAC, DPC | SVM | SN, SP, ACC, F1, MCC | Yes |
| DeepAcr | 2022 | PP, AAC, Sequencing features | LSTM, GRU, linear networks | ACC | No |
| AcrPred | 2023 | DPC, CTD, PSSM features | SVM | SN, SP, ACC, MCC, AUC | Yes |

1. Feature encoding scheme: AAC—amino acid composition; GDC—grouped dimer frequency counts; GTC—grouped trimer frequency counts; PSSM—Position-Specific Scoring Matrix; PP—protein properties; DPC—dipeptide composition; CTD—composition, transition and distribution;
2. Algorithm: XGBoost—extreme gradient boosting; SVM—support vector machine; GRU—gated recurrent unit; LSTM—long short-term memory;
3. Evaluation metrics: SN—sensitivity; SP—specificity; ACC—accuracy; F1—F1-Score; MCC—Matthews correlation coefficient; AUC—area under the curve;
4. Note: AcRanker webserver is now available on AcrHub platform.

strategy that combines six feature-encoding methods, including Dipeptide Composition (DPC), Composition-Transition-Distribution (CTD), and PSSM-based evolutionary features. This method balanced training subsets through down-sampling, built base classifiers using SVM, and optimizes feature sets through analysis of variance and incremental feature selection, resulting in an integrated predictor. AcrPred achieved an accuracy of 0.88 and a ROCAUC value of 0.95 on its independent test set [16]. Despite the promising performance of traditional ML approaches, DL approaches are better at fitting complex functions and capturing abstract concepts in data. This allows it to effectively capture the complex features of protein sequences. DL methods, such as DeepAcr, overcome these limitations by employing multiple neural network architectures, including Long Short-Term Memory (LSTM), linear networks, and Gated Recurrent Units (GRU). DeepAcr uses One-Hot encoding of amino acid sequences and protein properties for prediction, achieving accuracies of 0.96, 0.94 and 0.95 with LSTM, linear networks, and GRU, respectively, on a withheld dataset. Using this approach, researchers successfully identified AcrVIB1 [17]. Table I summarizes the key characteristics of these methods.

Despite advances in Acr protein prediction methods, significant limitations remain. A key issue is the traditional approach of considering proteins as isolated entities, which overlooks inherent similarities and potential connections, impeding a comprehensive understanding required for accurate Acr protein identification. To address this limitation and improve recognition of Acr proteins, it is crucial to construct a protein network that encompasses multiple proteins. This network-based approach aligns with the strengths of Graph Neural Networks (GNNs) [18]. Thus, a GNN-based node prediction framework, differentiated from non-graph deep learning [19], static graph-based machine learning [20], and intra-molecular geometric GNNs [21], is essential. This framework leverages GNNs to identify similarities and connections among Acr proteins, enhancing precision in Acr protein prediction within a heterogeneous protein network.

In response to these limitations, this study presents PACRGNN, an innovative model based on graph neural networks. PACRGNN represents multiple Acr proteins within a heterogeneous protein network, with edges indicating sequence and structural similarities, and nodes symbolizing proteins. This method effectively captures inherent dataset relationships. By employing representation learning via a graph neural network, PACRGNN utilizes GAT [22] and GraphSAGE [23] layers to abstract features and capture both local and global topological information. Additionally, PACRGNN integrates six categories of protein features: Composition-based Features, Evolutionary Features, Repetition and Distribution Features, Sequence Order and Correlation Features, Structure-related Features, and Large Language Model Features. This amalgamation provides a comprehensive initial representation for each node in the graph neural network. PACRGNN demonstrates excellent predictive capabilities and offers a user-friendly online platform, establishing itself as an effective tool for predicting Acr proteins. It is anticipated that PACRGNN will be a valuable resource for predicting Acr proteins, significantly advancing their discovery and potential applications in CRISPR-based therapeutic approaches.

## II. MATERIALS AND METHODS

The overall workflow includes data collection, feature representation, heterogeneous protein network construction, and model prediction. Acr and non-Acr protein sequences were gathered from databases, with PDB structures from RCSB [24] and predictions for unknown proteins made using ESMFold [25]. Sequences were split into training, validation, and test sets. Protein features were extracted using methods like ESM-2 [25] and PRAS [26]. Similarities in sequence and structure were computed with the SW algorithm [27] and TM-align [28], integrated through SNF [29], and refined with KNN. The final network and node features were combined into a comprehensive heterogeneous network of Acr proteins and input into the PACRGNN architecture (Graph Neural Network model composed of GAT layer and GraphSAGE layers) for downstream analysis. The relevant process is illustrated in Fig. 1.

### A. Data Collection and Pre-Processing

In the comprehensive protein dataset we constructed, the training and validation sets comprised a total of 2008 protein sequences, including 1004 Acr protein sequences (positive samples) obtained through experimental validations and literature reports, and 1004 non-Acr protein sequences (negative samples). These sequences were allocated between the training set (80% ) and the validation set (20% ). The independent test set included 86 sequences, with an equal distribution of 43 positive and 43 negative samples. The specific distribution of the dataset are presented in Table II.

We curated the dataset from multiple authoritative databases, including Anti-CRISPRdb [30], PaCRISPR [12], AcrHub [31], and the unified Anti-CRISPR resource [32]. The dataset includes

TABLE II
STATISTICAL SUMMARY OF THE BENCHMARK DATASETS IN THIS STUDY

| Dataset | Positive | Negative | Total |
|---|---|---|---|
| Training | 803 | 803 | 1,606 |
| Validation | 201 | 201 | 402 |
| Independent test | 43 | 43 | 86 |

experimentally verified and literature-reported Acr protein sequences as positive samples. In total, we collected 2346 Acr sequences, which were processed using the CD-HIT tool [33] to retain 1004 sequences, all with sequence similarity below 90% . Negative samples were randomly selected from the PaCRISPR negative category dataset, consisting of 1004 non-Acr protein sequences derived from phages or bacterial mobile genetic elements (MGEs) in bacterial genera previously identified to harbor Acr proteins. These sequences range in length from 50 to 350 residues, with a sequence similarity of less than 40% .

The independent test set comprises newly released Acr proteins from NCBI [34], spanning January to October 2024, and unused negative samples from the PaCRISPR dataset. We applied the same similarity thresholds to the independent test set as to the training set: positive samples with less than 90% sequence similarity, and negative samples with less than 40% . We ensured that the similarity between test set samples and training/validation set samples adhered to the same criteria. To obtain structural information for the proteins, we prioritized PDB structure files determined through experimental methods from the RCSB database. For sequences with unknown structures, we used the deep learning model ESMFold through NVIDIA's NIM API for structure prediction.

## B. Protein Representation

*1) Feature Extraction:* Various protein features were extracted and calculated using Pfeature [35], POSSUM [36], PRAS [26], BioPython [37], and the open-source implementation by Cheng Chen et al. [38]. These tools collectively capture multidimensional information about protein sequences and structures. A summary of the categories and dimensions of the features collected in this study is presented in Table III, with a more detailed explanation provided in the Supplementary Table 1. Based on their emphasis on different aspects of protein sequences and structures, we categorized the features into six major classes: composition-based features, evolutionary features, repetition and distribution features, sequence order and correlation features, structure-related features and large language model features.

*2) ESM-2 Model Embedding:* To obtain richer sequence representations, we employed the ESM-2 protein language model (650 M parameter version) through NVIDIA's NIM API to generate protein embedding representations. ESM-2 is an encoder model based on the Bert architecture [39], specifically designed for processing amino acid sequences. The version we used contains 33 transformer layers, 20 attention heads, a hidden space dimension of 1280, and a total of 650 million parameters.

ESM-2 underwent extensive pre-training on the UniRef protein sequence database, learning latent patterns in sequences through a masked language modeling self-supervised learning strategy. This large-scale training enables the model to effectively capture evolutionary information and structural features within protein sequences.

By pre-training on a large and diverse dataset, ESM-2 has showcased exceptional feature extraction abilities, learning universal protein patterns and offering high-quality sequence representations for subsequent tasks. This makes ESM-2 especially suitable for the Acr protein classification task in this study. Notably, despite the relatively small size of our dataset (approximately 2000 sequences), the feature representations obtained from ESM-2's large-scale pre-training provide PACRGNN with stable, information-rich input features, effectively addressing the challenges of few-shot learning.

## C. Heterogeneous Protein Network Construction

To capture global correlations among Acr proteins comprehensively, we propose a heterogeneous protein network that integrates multi-modal similarity features. In this network, proteins are depicted as nodes, while edges are created by merging sequence and structural similarity information, explicitly modeling the inherent relationships between proteins. To address the limitations of single-source similarity metrics, we combined two complementary measures: (1) sequence similarity calculated using the Smith-Waterman algorithm [27] and (2) structural similarity derived from the TM-score [40] based on TM-align [28]. These two similarity matrices were then fused using Similarity Network Fusion (SNF) [29], implemented in SNFpy, to generate a comprehensive matrix that robustly quantifies protein relationships. The final network was constructed by connecting each protein to its K-nearest neighbors based on this fused similarity matrix.

*1) Smith-Waterman Score Sequence Similarity:* To identify local similarities between protein sequences, the Smith-Waterman algorithm (SW) was employed. This algorithm is particularly effective in identifying conserved regions within sequences, which are often functionally significant or related to the protein's active site, especially in regions critical for inhibiting the CRISPR/Cas system. During alignment, the BLOSUM62 scoring matrix [41] was used to score amino acid substitutions, while gap penalties were applied to account for insertions or deletions.

To standardize the comparison of similarities between protein pairs, we normalized the alignment scores. Following the approach of Zheng et al. [42], we first computed the alignment score $S(A,B)$ and self-alignment score $S(A,A)$ and $S(B,B)$ using the Smith-Waterman algorithm. The normalized Smith-Waterman similarity $NSW(A,B)$ was then calculated as follows:

$$NSW(A,B) = \frac{S(A,B)}{\sqrt{S(A,A)} \times \sqrt{S(B,B)}} \quad (1)$$

The formula for calculating the final similarity is as follows:

$$SimNSW(A,B) = \frac{NSW(A,B) + NSW(B,A)}{2} \quad (2)$$

TABLE III
FEATURE TYPES AND DIMENSIONS WITH BREAKDOWN

| Category | Type of Features | Dimension |
|---|---|---|
| Composition-based Features | AAC, DPC, CTDC, PCP | 489 |
| Evolutionary Features | PSSMs (AADP, MEDP, PSSS-AC, PSSM-comp, RPSSM) | 1550 |
| Repetition and Distribution Features | RRI, SER, CTDD | 235 |
| Sequence Order and Correlation Features | CTDT, PAAC, SOCN | 68 |
| Structure-related Features | B-factor, SSC | 9 |
| Large Language Model Features | ESM | 1280 |

Finally, a sequence similarity matrix $W_{seq}$ was constructed, where each element of the matrix represents the Smith-Waterman scoring similarity between a pair of sequences. Specifically, the element $W_{seq}(i,j)$ corresponds to the similarity score $SimNSW(S_i, S_j)$.

*2) TM-Score Structure Similarity:* In addition to sequence similarity, the three-dimensional structure of proteins is crucial for determining their function. Even when proteins show considerable sequence variations, structural similarities can uncover functional connections. By computing structure similarity for each protein pair, we construct a similarity matrix that reflects structure relationships. This matrix aids in building a heterogeneous protein network and predicting functions.

To assess the similarity between protein structures, we initially employ the TM-align algorithm to align each protein pair structurally. After obtaining the alignment results, we utilize TM-score to quantitatively evaluate the quality of the alignments. TM-score, a scoring function commonly used for protein structure alignment, is calculated using the formula provided below:

$$TM-score = \max\left(\frac{1}{L_N}\sum_{i=1}^{L_T}\frac{1}{1+\left(\frac{d_i}{d_0}\right)^2}\right) \tag{3}$$

Where $L_N$ denotes the length of the original protein structure, measured as the number of residues. $L_T$ represents the aligned residue length, and $d_i$ is the distance between the $i$-th pair of aligned residues. The constant $d_0$ is used for normalization, aiming to mitigate the impact of protein size variations.Using the TM-score, we construct a protein structure similarity matrix $W_{struct}$, where each element represents the minimum TM-score from the two alignments between the two protein structures.

*3) Fusion Similarity:* Since the functional information of proteins is influenced by both sequence and structure similarities, relying on a single similarity metric may not fully capture their characteristics. Therefore, we adopted the SNF method, which integrates Smith-Waterman score sequence similarity and TM-score structure similarity. The SNF method employs an iterative process to update the similarity matrix, ultimately generating a comprehensive matrix that merges these two similarities measures.

The SNF approach effectively integrates multiple similarity sources into a unified network. Sequence similarity reveals the evolutionary relationships between proteins, while structure similarity often directly reflects their function. By fusing these similarities using SNF, we construct a network that simultaneously incorporates both evolutionary and functional information. The SNF process is as follows:

First, we standardize the two similarity matrices:

$$P_{seq}(i,j) = \begin{cases} \frac{W_{seq}(i,j)}{2\sum_{k\neq i} W_{seq}(i,k)}, & j \neq i \\ \frac{1}{2}, & j = i \end{cases} \tag{4}$$

$$P_{struct}(i,j) = \begin{cases} \frac{W_{struct}(i,j)}{2\sum_{k\neq i} W_{struct}(i,k)}, & j \neq i \\ \frac{1}{2}, & j = i \end{cases} \tag{5}$$

Where $W_{seq}(i,j)$ and $W_{struct}(i,j)$ represent the initial sequence and structure similarities between two proteins, respectively. The standardized matrices $P_{seq}$ and $P_{struct}$ are then used in the subsequent iterative updates.

Next, the local similarity matrices are calculated to preserve the neighbourhood structure of each protein:

$$S_{seq}(i,j) = \begin{cases} \frac{W_{seq}(i,j)}{\sum_{k\in N_i} W_{seq}(i,k)}, & j \in N_i \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$S_{struct}(i,j) = \begin{cases} \frac{W_{struct}(i,j)}{\sum_{k\in N_i} W_{struct}(i,k)}, & j \in N_i \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Where $N_i$ represents the set of neighbours of protein $i$. These matrices are normalized only within each protein's local neighbourhood, thus preserving local information.

The local matrices $S_{seq}$ and $S_{struct}$ are then used to facilitate iterative updates, allowing the exchange of state matrix information derived from different similarity sources. The iterative update formulas for the two distinct similarity matrices are as follows:

$$P_{seq}^{t+1} = S_{seq} P_{struct}^{t} S_{seq}^{T} \tag{8}$$

$$P_{struct}^{t+1} = S_{struct} P_{seq}^{t} S_{struct}^{T} \tag{9}$$

The state matrices $P_{seq}$ and $P_{struct}$, derived from different data sources, gradually approach each other after multiple iterations, fusing information from both sequence and structure similarity. The final comprehensive similarity matrix is obtained by averaging the two resulting matrices.

$$P_c = \frac{P_{seq}^{f} + P_{struct}^{f}}{2} \tag{10}$$

The comprehensive similarity matrix integrates both sequence and structure information, providing a comprehensive representation of protein features that can be used for constructing protein networks.

*4) The Construction of a K-Nearest Neighbor Heterogeneous Protein Network:* Following the creation of the fusion similarity matrix, the K-nearest neighbor (KNN) algorithm was used to construct a heterogeneous protein network. This algorithm develops the network by connecting each protein node to its K most similar neighbors, with these connections reflecting varying degrees of sequence and structure similarity that potentially indicate functional associations. Using the KNN algorithm enables the network to highlight important similarity patterns, especially those where high similarity implies that Acr proteins are likely to have similar roles.

Subsequently, the KNN-constructed heterogeneous protein network was converted into an adjacency matrix, which, along with node features, served as input for the PACRGNN model. The PACRGNN model's GAT layer and GraphSAGE layer enable protein nodes to aggregate the feature information of neighboring nodes. As the network layer deepens, the node representation vector progressively integrates information from its local neighborhood, enhancing the model's ability to learn critical patterns of Acr protein sequences.

This feature aggregation method, as utilized in PACRGNN, holds substantial implications for biological understanding. In the domain of protein function prediction, proteins with similar sequences and structures often display similar functional traits. Thus, our model's hierarchical aggregation process effectively learns the association pattern between sequence, structure, and function. Then PACRGNN efficiently captures these essential feature patterns through its attention mechanism and feature aggregation.

## D. The Architecture of PACRGNN

The architectural framework of PACRGNN is illustrated in Fig. 2. The core function of this network is to aggregate features through one GAT [22] layer and four GraphSAGE [23] layers. Subsequently, nodes designated for prediction are sequentially integrated into the network for analysis and forecasting. The integration of various graph neural network layers in PACRGNN facilitates the rapid assimilation of vital protein sequences features via effective learning of features and network topology, thereby enhancing prediction performance. To emphasize critical features, a trainable feature selection layer has been introduced to enhance the model's prediction performance by adaptively adjusting the weights of input features. This layer consists of a learnable parameter vector, where each element represents the importance of a corresponding feature. During training, the model dynamically updates these coefficients to highlight important features and reduce the impact of noise.

*1) GAT Layer:* After constructing the heterogeneous protein network, which integrates comprehensive similarity and complex feature information of Acr proteins, the feature network is then initialized, then the initial feature matrix $h_0$ is fed into the graph neural network for processing. The GAT layer serves as the starting layer of the model, processing node features and dynamically adjusting weights between nodes. The output of the this layer is described as follows:

$$h_i^{(1)} = \sigma\left(\sum_{j\in N(i)} \alpha_{ij} W h_j^{(0)}\right) \tag{11}$$

Where $h_j^{(0)}$ represents the input feature of node $j$, while $h_i^{(1)}$ denotes the output feature of node $i$. Additionally, $\mathcal{N}(i)$ denotes the set of neighbor nodes of node $i$, $W$ signifies the learnable weight matrix, and $\sigma$ is the activation function. Finally, $\alpha_{ij}$ represents the attention coefficient between node $i$ and node $j$.

The attention coefficient $\alpha_{ij}$ is calculated using the following formula:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(a^T\left[Wh_i^{(0)} \parallel Wh_j^{(0)}\right]\right)\right)}{\sum_{k\in N(i)} \exp\left(\text{LeakyReLU}\left(a^T\left[Wh_i^{(0)} \parallel Wh_k^{(0)}\right]\right)\right)} \tag{12}$$

*2) GraphSAGE Layer:* Following the graph attention layer, the PACRGNN employs a series of GraphSAGE layers to extract the node features. GraphSAGE is a model designed to capture both local and global information between nodes using graph convolution operations. It learns node representations by sampling and aggregating the features of each node's neighbours. The computation in each GraphSAGE layer is defined as follows:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)\cup\{i\}} \frac{1}{\sqrt{d_i d_j}} W h_j^{(l)}\right) \tag{13}$$

Where $h_j^{(l)}$ represents the input features of node $j$ at layer $l$ and $h_i^{(l+1)}$ denotes the output features of node $i$ at layer $l+1$. $d_i$ and $d_j$ and represent the degrees of nodes $i$ and $j$, respectively. The core concept behind the GraphSAGE layer is updating the representation of each node by aggregating the features of its neighbors. This process involves several steps: Node $i$ aggregates the features of its neighbors and integrates the neighbor information into its own features through a weighted sum. The degree normalization factor $\frac{1}{\sqrt{d_i d_j}}$ is applied to adjust for differences in node degrees and prevent feature imbalance. A learnable weight matrix $W$ linearly transforms the aggregated features, and a nonlinearity is introduced through the activation function $\sigma$.

In the PACRGNN implementation, four GraphSAGE layers are employed, each followed by a LeakyReLU activation function and a Dropout mechanism to mitigate overfitting. Furthermore, the graph's adjacency matrix effectively captures structural relationships between nodes, enhancing the model's capacity to learn protein sequence.

*3) Prediction of Anti-CRISPR Proteins:* In this study, the prediction of Acr proteins is approached by individually inputting nodes into the constructed PACRGNN network. The model processes each node by receiving its initial feature vector, connecting it with the existing network, and leveraging the graph neural network for feature aggregation. This technique enables the extraction of more in-depth feature representations.

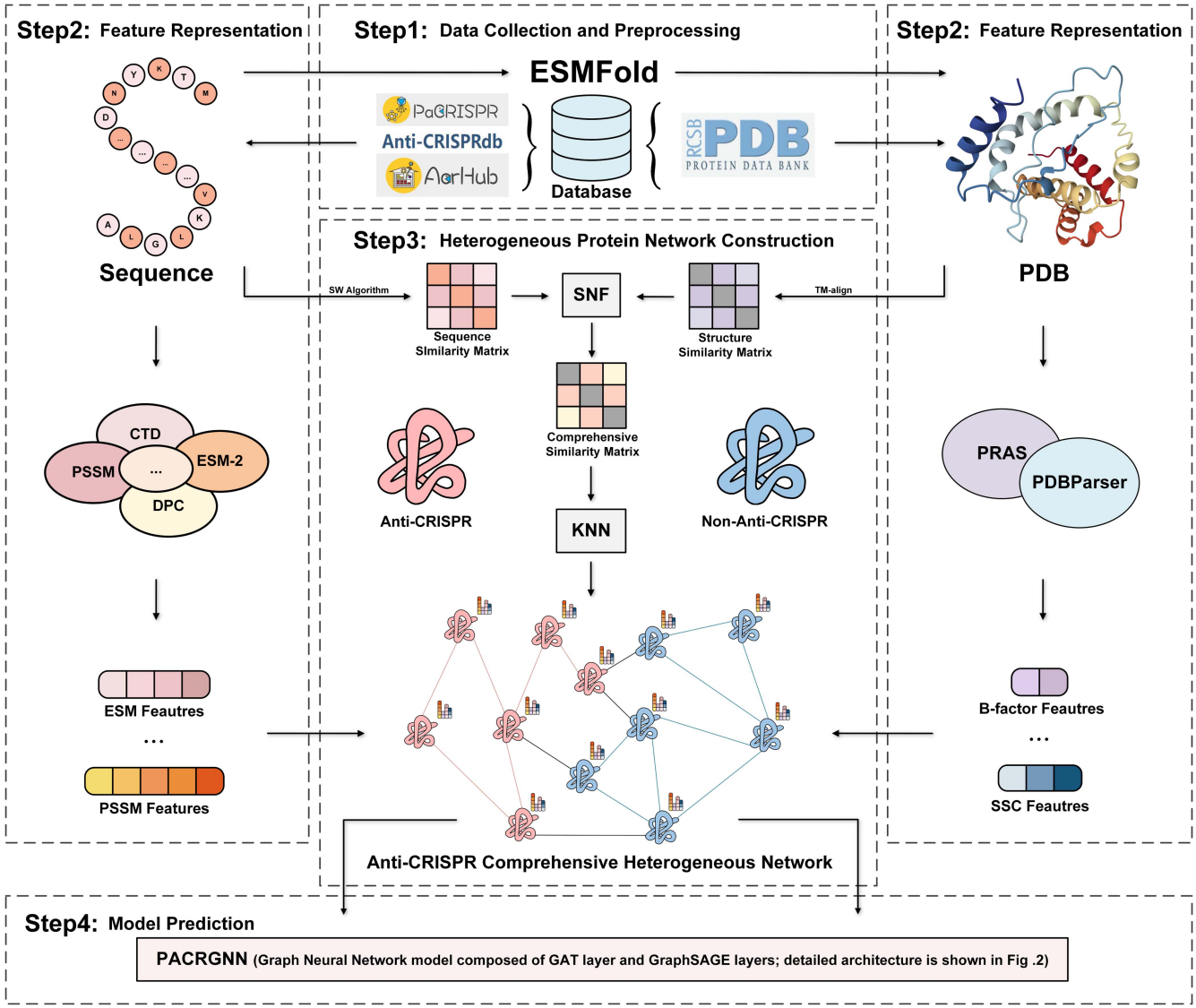


Fig. 1. The overall workflow. The workflow comprises four main components: 1. Data Collection: Protein sequences are gathered from relevant databases, with those that have experimentally resolved PDB structures retrieved from the RCSB database. ESMFold is employed to predict structures of unknown proteins. 2. Feature Representation: Protein representations are generated using various methods, including the pre-trained ESM-2 model, PRAS, and additional protein feature extraction techniques. 3. Heterogeneous Protein Network Construction: Sequence similarity is calculated through the Smith-Waterman algorithm, while structural similarity is assessed using TM-align. These similarity matrices are integrated using the Similarity Network Fusion (SNF) method, which involves iterative updates, local neighborhood smoothing, and cross-source information exchange to produce a comprehensive similarity matrix. Following SNF integration, k-Nearest Neighbors (KNN) is applied to further refine the heterogeneous protein network and ensure connectivity. 4. Model Prediction: The final network and node features are integrated into a comprehensive heterogeneous network of Acr proteins, which is then input into the PACRGNN architecture—a Graph Neural Network model composed of GAT and GraphSAGE layers—for downstream prediction of Anti-CRISPR proteins.

As a result, the node's feature vector includes contextual information from neighboring nodes, highlighting node importance and interrelationships within the network and thus improving prediction accuracy. Finally, these aggregated feature vectors are transformed into prediction outcomes via a linear layer.

$$y = W^{(final)} h^{(final)} + b^{(final)} \tag{14}$$

Where the output, represented by the variable $y$, is determined by the learnable weight and bias of the final layer, represented by $W^{(final)}$ and $b^{final}$, respectively. GAT layer allows PACRGNN to capture node interdependencies effectively, enhancing Acr protein prediction performance.

Model optimization employs binary cross-entropy loss with L2 regularization to minimize prediction errors and prevent overfitting. A cosine annealing learning rate schedule with warm restarts [43], combined with the Adam optimizer [44], enhances global optimization by periodically reducing the learning rate. Dropout [45] further improves robustness and mitigates overfitting.

A comprehensive evaluation of PACRGNN's performance in Acr protein prediction employs metrics such as accuracy,

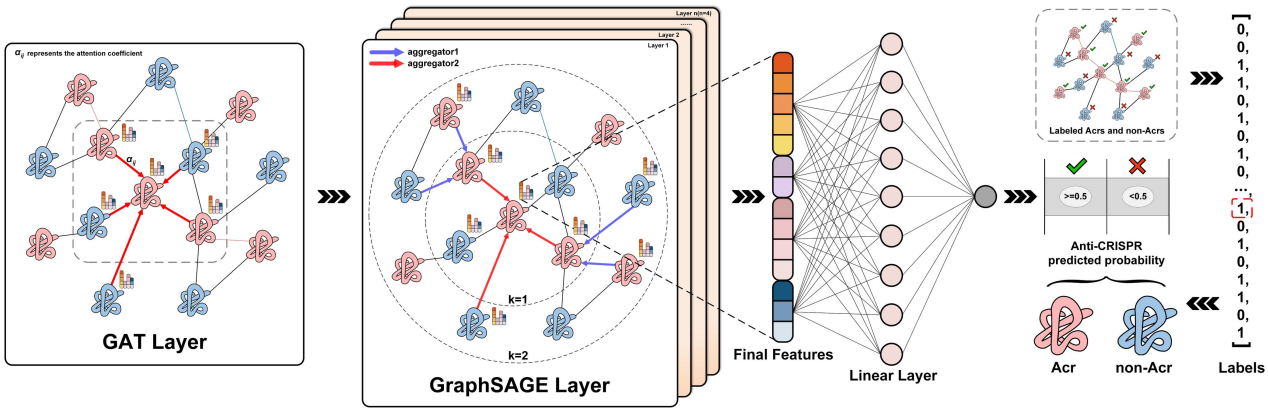


Fig. 2. Architecture of the PACRGNN model for Acr recognition and prediction. The model takes a heterogeneous protein network as input and is structured with three key components: 1. The GAT layer, which calculates the attention weights between protein nodes to capture the node relationships; 2. The GraphSAGE layer (n layers in total, n = 4), which learns node representations by aggregating neighbor information through multiple iterations; 3. The linear layer, which converts the final features into the predicted probability of Acr protein classification and determines whether it is an Acr protein based on a threshold of 0.5.

recall, F1, and area under the PR curve (PRAUC), as well as Matthews Correlation Coefficient (MCC), G-Mean, and area under the ROC curve (ROCAUC). These metrics offer an extensive analysis of the model's classification effectiveness. Accuracy is generally regarded as a broad indicator of model performance. However, in the realm of binary classification tasks, especially within biological research, recall and F1 assume greater significance. Recall is critical for ensuring comprehensive identification of true Acr proteins by minimizing false negatives, which are particularly detrimental in biological detection tasks. The F1, on the other hand, provides a balanced measure of precision and recall, proving invaluable in optimizing both the sensitivity and accuracy of Acr protein identification.

Moreover, PRAUC becomes essential in scenarios where the trade-off between precision and recall is pronounced. This metric offers a more detailed assessment than ROCAUC, especially beneficial when confronting imbalanced datasets or when threshold settings vary.

These methods and evaluation metrics offer targeted guidance for the model training process, enabling iterative refinements to reduce loss and improve both the accuracy and the robustness of predictions. The framework is not solely focused on enhancing prediction outcomes but also aims at bolstering model interpretability. This approach provides deeper insight into how different features influence predictions, fostering a better understanding of feature significance in predictive models.

## III. Results and Discussion

### A. Comprehensive Ablation Studies

*1) Network Construction Analysis:* This study investigated the impact of various parameter settings on model performance during the heterogeneous protein network construction process, focusing on two main factors: similarity source selection and the K-nearest neighbor parameter. Fig. 3(a) illustrates the prediction performance of PACRGNN under various similarity selection schemes. The results indicate that integrating sequence and structure similarity yields superior performance, with PACRGNN (SNF) achieving an accuracy of 0.9577, an F1 of 0.9572, and a PRAUC of 0.9876. In contrast, solely using sequence similarity results in an accuracy of 0.8234, an F1 of 0.8360, and PRAUC of 0.8401, whereas solely relying on structure similarity leads to an accuracy of 0.9104, an F1 of 0.9147, and PRAUC of 0.8825. These findings underscore the complementary nature of sequence and structure information in Acr protein prediction, where sequence homology indicates evolutionary relationships, and structure similarity often aligns with function. Combining both types of information significantly enhances feature representation.

Fig. 3(b) demonstrates that the optimal performance of PACRGNN occurs when the K value is set at 400. Specifically, at K = 400, the model exhibits peak performance, whereas deviations to smaller values (e.g., K = 300) or larger values (e.g., K = 500) result in performance declines. This suggests a need for balancing network information transfer and computational efficiency in choosing the K value. These insights not only confirm that the optimal configuration for the subsequent graph neural network model involves the integration of sequence and structure similarity with an optimal K value of 400, but also highlight the advantage of this approach over traditional methods such as AcRanker [[14]], PaCRISPR [[12]], AcrPred [[16]], and DeepAcr [17], which primarily rely on isolated sequence or evolutionary features. This integrated heterogeneous protein network construction approach is better at capturing the relationships between proteins.

From a biological standpoint, this integrated approach more effectively characterizes the intrinsic relationships and functional attributes of Acr proteins, suggesting that the heterogeneous protein network construction strategy introduced herein offers novel insights into understanding the action mechanism of Acr proteins and underscores the considerable potential of meshing network science with deep learning for protein function prediction.

*2) Evaluation of Model Architecture Components:* To comprehensively evaluate the effectiveness of the PACRGNN model architecture, experiments analyzed the impact of the number

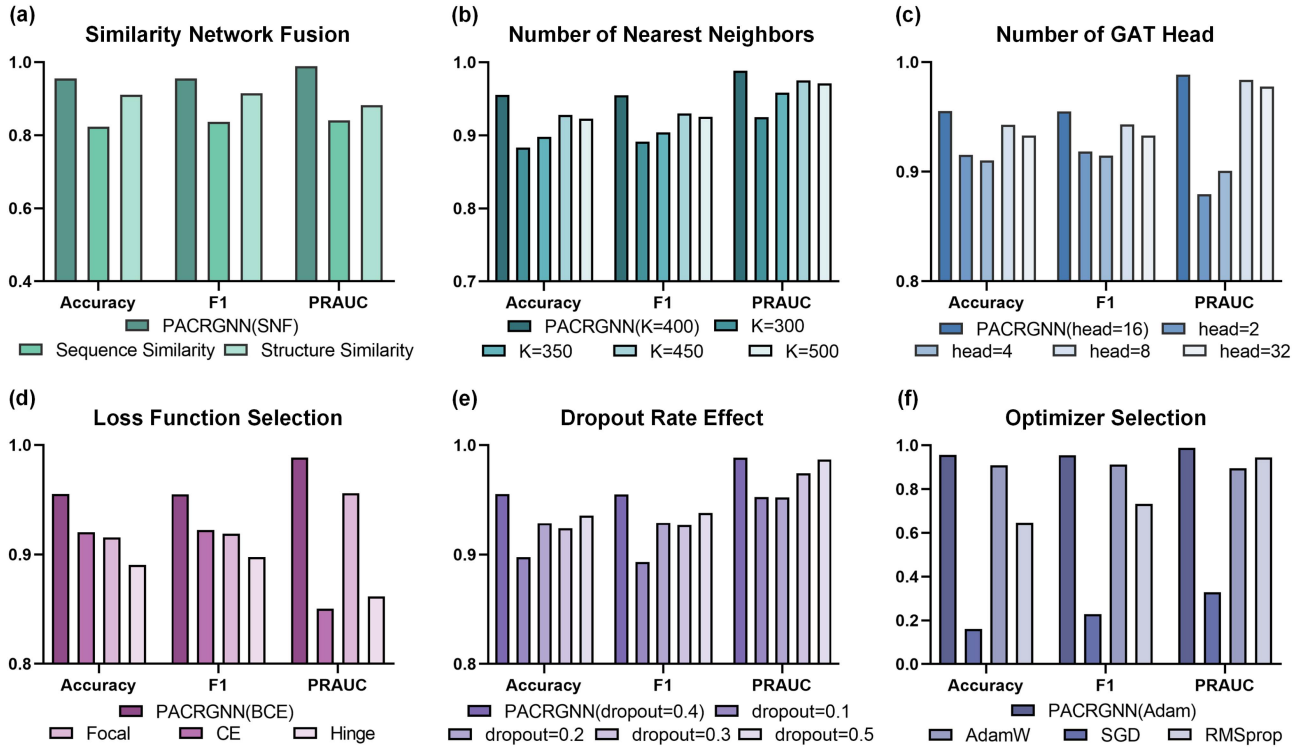


Fig. 3. Performance analysis of PACRGNN with different hyperparameter settings. The six subfigures show the impact of different model configurations: (a) performance comparison of different similarity fusion strategies, including sequence similarity, structure similarity, and the comprehensive similarity after SNF fusion; (b) the impact of different numbers of K-nearest neighbors (K = 300–500) on model performance; (c) the impact of different numbers of GAT heads (head = 2,4,8,16,32) on the prediction results; (d) comparison of different loss functions (BCE, CE, Focal and Hinge loss); (e) the impact of different dropout rates (0.1–0.5) on model robustness; (f) performance evaluation of different optimizers (Adam, AdamW, SGD and RMSprop).

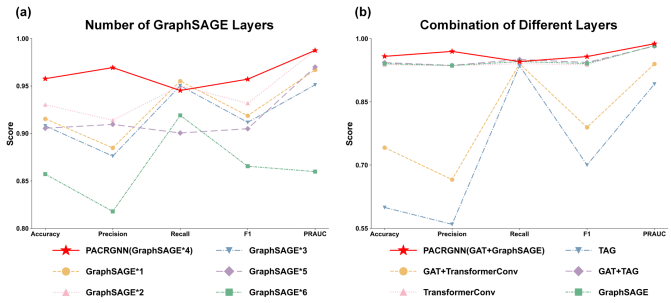


Fig. 4. The impact of neural network architecture on model performance. (a) shows the impact of different numbers of GraphSAGE layers (layers 1–6) on performance metrics. (b) compares the performance of different layer combinations, including PACRGNN (GAT+GraphSAGE), GraphSAGE, TAG, GAT+TransformerConv, TransformerConv, and GAT+TAG.

of GraphSAGE layers and various layer combination schemes on prediction performance. As shown in Fig. 4(a), the number of GraphSAGE layers was varied from 1 to 6. The model achieved optimal performance across multiple metrics (accuracy, precision, recall, F1, and PRAUC) with 4 GraphSAGE layers. Increasing the number of layers beyond this point neither improved performance nor resulted in significant gains, and, in some cases, even caused a decline. This suggests that a 4-layer GraphSAGE architecture provides an optimal balance by expanding the receptive field while maintaining computational efficiency. It effectively captures multiscale features of the heterogeneous protein network while mitigating the risk of overfitting.

Fig. 4(b) compares the performance of different neural network layer combinations. The introduction of a GAT layer significantly enhanced PACRGNN's prediction performance, highlighting the critical role of the attention mechanism in graph representation learning. The GAT layer adaptively focuses on important nodes and edges and selectively fuses information from different neighbors, enabling the model to extract key features related to protein functions more effectively. However, adding layers such as TAGConv (TAG) [46] or TransformerConv [47], or combining multiple convolutional layers, did not yield additional performance improvements.

The PACRGNN architecture demonstrated superior performance in Acr protein prediction compared to existing deep learning models. By leveraging the strengths of graph neural networks, particularly the combination of GAT and GraphSAGE layers, PACRGNN effectively captures the complex interactions and structural information within the heterogeneous protein network. This capability enables the model to learn more expressive and informative representations for Acr proteins compared to methods such as DeepAcr, AcrNET, and PreAcrs [17], [48], [49]. Biologically, this approach provides a framework to model intricate functional associations among proteins and identify critical

patterns and regions that elucidate Acr protein roles, offering a biologically meaningful tool for studying these proteins within the context of biological networks.

*3) Analysis of Model-Specific Parameters:* To further optimize the performance of PACRGNN, this study explores the effects of several key parameters on the model performance, including the number of GAT attention heads, loss function selection, dropout rate setting, and optimizer selection.

To optimize the performance of PACRGNN, this study investigates the impact of several crucial parameters, including the number of GAT attention heads, the choice of loss function, the setting of dropout rates, and the selection of optimizers. Fig. 3(c) illustrates the correlation between the number of GAT attention heads and model performance. The results indicate that the optimal configuration of 16 attention heads outperforms other settings across all metrics. This suggests that 16 heads are sufficient to capture essential correlation patterns among nodes effectively without introducing unnecessary computational redundancy. Consequently, the GAT layer in this study is configured with a 16-head attention mechanism. Fig. 3(d) evaluates different loss functions and identifies the BCE Loss as the most effective for binary classification tasks.

Fig. 3(e) assesses the influence of the dropout technique on the model's generalization ability. The findings demonstrate that a moderate dropout rate of 0.4 optimally mitigates overfitting and enhances performance across various metrics. In contrast, a higher dropout rate of 0.5 degrades performance, likely due to excessive loss of information. Therefore, a dropout rate of 0.4 is implemented after each graph neural network layer in PACRGNN. Fig. 3(f) explores the convergence effects of different optimizers, showing that the Adam optimizer—in comparison to others such as SGD and RMSprop [50] provides superior performance due to its adaptive learning rate adjustments and gradient correction capabilities, and is thus chosen as the standard optimizer for PACRGNN.

## B. Feature Contribution Analysis

To gain a deeper understanding of the impact that various features have on model performance, a comprehensive feature importance analysis and a series of ablation experiments were conducted. Feature importance was determined by calculating the gradients of the model output with respect to each input feature, highlighting how changes in each feature affect the model's predictions. Fig. 5(a) displays the importance scores of individual features, where higher scores indicate greater influence on the model. Fig. 5(b) illustrates the variations in model performance upon the removal of individual features, providing further insight into their relative importance.

The feature importance analysis results reveal that sequence order-related features (SOCN) possess the highest scores, followed by B-factor and MEDP (Fig. 5(a)). This suggests that sequence order, structure and evolutionary information are critical for the recognition of Acr proteins. Conversely, composition-based features such as PCP, DPC and SER exhibit relatively low scores, possibly because the information they convey is subsumed by other, more dominant features.

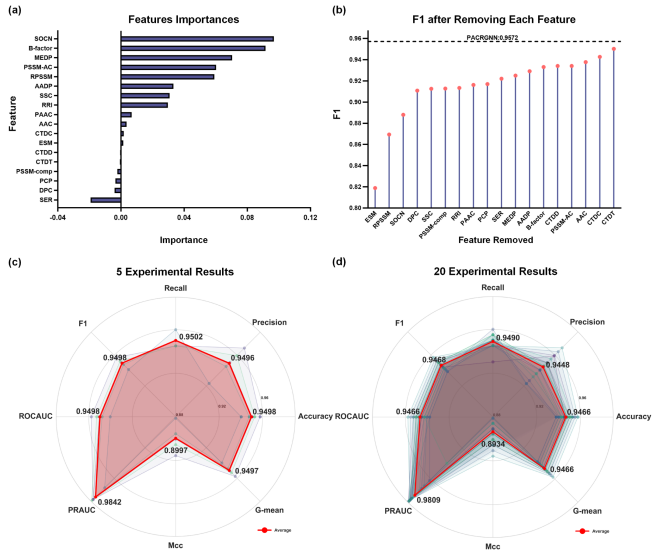


Fig. 5. Feature analysis and model stability assessment. The figure contains four parts: (a) importance scores of different protein features; (b) changes in F1 after removing individual features; (c) visualization of model performance metrics for 5 experimental runs; (d) visualization of performance metrics for 20 experimental runs.

To further substantiate the contributions of these features, removal experiments were carried out, detailed in Fig. 5(b). Initially, a baseline model was established, which achieved its peak F1 score of 0.9572 when all features were integrated, as indicated by the dashed line in Fig. 5(b). Sequentially removing features revealed that omitting the ESM embedding led to a significant decline in performance, highlighting its critical role in learning sequence representations. Additionally, the removal of features such as RPSSM and SOCN also resulted in notable performance drops, underscoring the importance of sequence order and evolutionary information in the feature set. These findings demonstrate the synergistic interplay between pre-trained language model-derived features and traditional features in enhancing model performance. In contrast, removing features like AAC, CTDC, and CTDD resulted in only marginal changes to the model's performance. This indicates that their contribution to the model's predictive power is relatively minimal. One possible reason is that the data they provide may be redundantly captured by more complex features. For instance, AAC and DPC might overlap since both describe amino acid composition, but DPC offers a more nuanced view by incorporating amino acid pairings. Similarly, CTDC and CTDD may have a reduced effect because the composition and distribution details they provide are already well-represented by higher-order features in the model.

Additionally, this study conducted comprehensive permutation and combination experiments using the six major categories of collected features, resulting in a total of 63 experimental setups (Supplementary Table 2). Our findings indicate that no feature combination surpasses the predictive power of using all six categories together. This underscores that incorporating all features provides a comprehensive representation of the underlying data, significantly enhancing the model's predictive accuracy. These experiments also reinforce our understanding of

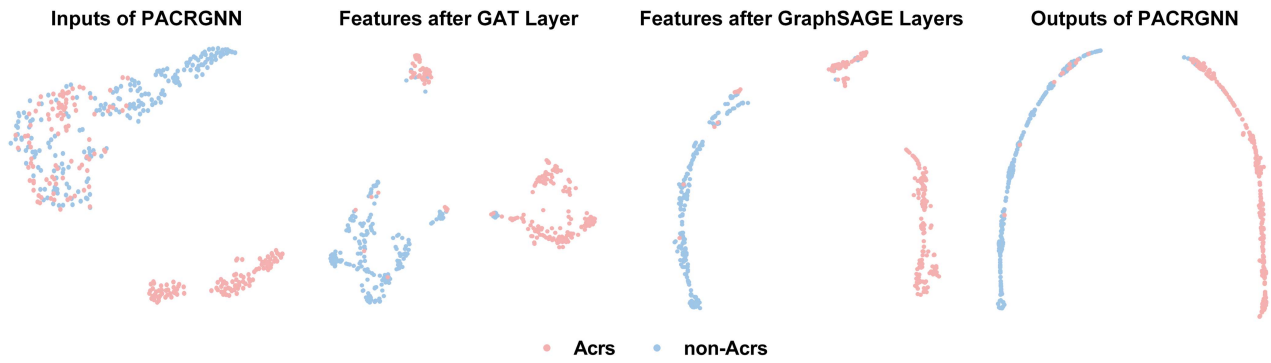


Fig. 6. Scatter plot of t-SNE visualization of node features in different network layers. From left to right: original feature distribution, feature distribution after GAT layer, feature distribution after tow GraphSAGE layers, and feature distribution in the final classification layer. The red pentagrams and blue dots represent the positive sample Acrs and negative sample non-Acrs, respectively.

each feature's relative importance and complementarity within the model. Admittedly, as the dataset size increases, integrating all six categories of protein features to construct a heterogeneous protein network poses challenges regarding computational resources and inference times.

This study's feature importance analysis and ablation experiments emphasize the robustness of PACRGNN's feature selection strategy, particularly its integration of the advanced ESM-2 pre-trained language model, distinguishing it from AcrNET, which utilize the ESM-1b model [51]. The synergy between ESM embeddings and diverse traditional features, including sequence order-related, evolutionary, and amino acid composition characteristics, equips PACRGNN to capture a broad spectrum of attributes essential for accurately identifying Acr proteins. This method not only elevates the accuracy of Acr protein predictions but also provides insightful revelations concerning the biological mechanisms and evolutionary relationships of these proteins by efficiently learning semantic information, high-level representations, and both local and global patterns, as well as evolutionary conservation and physicochemical properties.

### C. Performance Metrics and Model Validation

To comprehensively evaluate the performance of PACRGNN, this study implemented a rigorous experimental setup and analyzed the results in detail. The models were trained using NVIDIA A40, 3090, and 4090 GPUs. For parameter settings, the learning rate scheduler was configured with an initial cycle step $T_0 = 200$, a cycle multiplication factor $T_mult = 2$, and a minimum learning rate $\eta_{min} =$ 5e-6, striking a balance between training efficiency and model convergence. The input-output dimensions of each layer were fine-tuned to improve feature extraction and propagation. The model underwent five and twenty independent training and validation, each for 5000 epochs, with parameter updates performed using the Adam optimizer. key metrics, such as loss values, were recorded, and the training instance with the best performance was selected as the benchmark.

To assess the feature learning ability of PACRGNN, t-SNE dimensionality reduction scatter plots were used to visualize and analyze the outputs from different network layers (Fig. 6). The plots reveal the distribution trends of positive and negative samples, which become increasingly distinct as the network deepens. Particularly, after the GAT layer and multilayer GraphSAGE processing, the separation of positive and negative samples into distinct regions of the feature space becomes more pronounced, demonstrating PACRGNN's effectiveness in extracting discriminative features for Acr protein.

Fig. 5(c) and (d) illustrate the results of multiple PACRGNN runs on the validation set. The eight evaluation metrics were comprehensively examined across five and twenty independent runs. The results indicate that PACRGNN consistently delivers robust performance and stability. Fig. 5(d) shows the results of five runs, while Fig. 5(a) presents those from twenty runs. Regardless of the number of runs, the model exhibits minimal performance variation, with all evaluation metric variances across 20 runs falling below 0.0002. This highlights the stability and reliability of the model. The benchmark model, selected from the experiment with the best performance, achieved excellent results on the validation set: an accuracy of 0.9577, precision of 0.9694, recall of 0.9453, F1 of 0.9572, ROCAUC of 0.9853, PRAUC of 0.9876, MCC of 0.9157, and G-Mean of 0.9576.

These results demonstrate that PACRGNN achieves outstanding and robust performance in Acr protein prediction tasks. While some variation across training runs may arise due to parameter initialization, the model consistently exhibits excellent learning and generalization capabilities. The stability and consistency of results from multiple runs further validate the reliability and effectiveness of the PACRGNN method, establishing a solid foundation for its practical application in Acr protein identification and prediction tasks.

To objectively assess PACRGNN's performance advantages, it was compared against several existing methods (Table IV). The results indicate that PACRGNN significantly outperforms competing models across all evaluation metrics. For example, PACRGNN achieves an accuracy of 0.9577, approximately 7.4 percentage points higher than the next best method, AcrPred. Furthermore, PACRGNN demonstrates an exceptional balance between precision and recall, achieving an F1 of 0.9572, far surpassing other models. While HMM [52], [53] achieves a precision of 1.0000, its recall is extremely low (0.0766), resulting in an F1 far inferior to that of PACRGNN. These findings confirm PACRGNN's ability to accurately identify true positive samples

TABLE IV
COMPARE THE PERFORMANCE OF DIFFERENT METHODS ON VALIDATION SETS

| Method | TP | FP | TN | FN | Accuracy | Precision | Recall | F1 | MCC | G-Mean | ROCAUC | PRAUC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PACRGNN** | **190** | 6 | 195 | **11** | **0.9577** | 0.9694 | **0.9453** | **0.9572** | **0.9157** | **0.9576** | **0.9853** | **0.9876** |
| PaCRISPR | 74 | 26 | 175 | 127 | 0.6194 | 0.7400 | 0.3682 | 0.4917 | 0.2762 | 0.5662 | 0.7726 | 0.7581 |
| AcrPred | 167 | 13 | 188 | 34 | 0.8831 | 0.9278 | 0.8308 | 0.8766 | 0.7704 | 0.8815 | 0.9690 | 0.9662 |
| DeepAcr | 183 | 130 | 71 | 18 | 0.6318 | 0.5847 | 0.9104 | 0.7121 | 0.3175 | 0.5671 | 0.7215 | 0.7034 |
| AcRanker | 70 | 41 | 160 | 131 | 0.5721 | 0.6306 | 0.3483 | 0.4487 | 0.1614 | 0.5265 | – | – |
| HMM | 8 | **0** | **201** | 193 | 0.5199 | **1.0000** | 0.0398 | 0.0766 | 0.1425 | 0.1995 | – | – |

TABLE V
COMPARISON OF MODEL PERFORMANCE ON AN INDEPENDENT TEST SET (ACRS PUBLISHED ON NCBI IN 2024)

| Method | TP | FP | TN | FN | Accuracy | Precision | Recall | F1 | MCC | G-Mean | ROCAUC | PRAUC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PACRGNN** | **28** | **0** | **43** | **15** | **0.8256** | **1.0000** | **0.6512** | **0.7887** | **0.6948** | **0.8069** | 0.8275 | **0.8806** |
| PaCRISPR | 22 | 4 | 39 | 21 | 0.7093 | 0.8462 | 0.5116 | 0.6377 | 0.4557 | 0.6812 | **0.8880** | 0.8802 |
| AcrPred | 21 | 19 | 24 | 22 | 0.5233 | 0.5250 | 0.4884 | 0.5060 | 0.0466 | 0.5221 | 0.8524 | 0.8589 |
| DeepAcr | **28** | 13 | 30 | **15** | 0.6744 | 0.6829 | **0.6512** | 0.6667 | 0.3492 | 0.6740 | 0.6993 | 0.7523 |
| AcRanker | 17 | 3 | 40 | 26 | 0.6628 | 0.8500 | 0.3953 | 0.5397 | 0.3853 | 0.6064 | – | – |
| HMM | 2 | **0** | **43** | 41 | 0.5233 | **1.0000** | 0.0465 | 0.0889 | 0.1543 | 0.2157 | – | – |

while minimizing false positives, solidifying its position as a state-of-the-art method for Acr protein prediction.

### D. Performance Analysis on Independent Test set

To further validate the generalization ability of PACRGNN, a comprehensive performance evaluation was conducted on an independent test set published by NCBI in 2024. PACRGNN combined the original training set with the validation set to obtain a new training set, totaling 2008 sequences. Its performance on the independent test set was compared with several mainstream prediction methods, including PaCRISPR, AcrPred, DeepAcr, AcRanker, and HMM. The detailed results are presented in Table V.

The results show that PACRGNN consistently outperforms the comparison methods on the independent test set. Notably, PACRGNN achieves an accuracy of 0.8256, significantly higher than the other methods. Of particular importance is PACRGNN's precision rate of 1.0, indicating the absence of false positive predictions (FP = 0), and a recall rate of 0.6512 (TP = 28, FN = 15), demonstrating a favorable balance between precision and recall. This is reflected in an F1 of 0.7887, which is substantially higher than those achieved by other methods.

In terms of model stability metrics, PACRGNN achieved the highest values, with an MCC of 0.6948 and a G-Mean of 0.8069. Other machine learning methods tested on the independent set, such as AcrPred, DeepAcr, and AcRanker, achieved accuracy rates of 0.5233, 0.6744, and 0.6628, respectively.

These results conclusively demonstrate the superior predictive performance of PACRGNN in practical scenarios, particularly when handling novel Acr protein sequences, including previously unseen Acr protein variants. The model's ability to maintain high accuracy and reliability across diverse datasets underscores its robustness as a computational tool.

### E. Case Study

To further evaluate the performance of our Acr protein prediction model, PACRGNN, we conducted a case study involving four Acr proteins from an independent test dataset, representing three Acr families: AcrVA, AcrF, and AcrIIA. As shown in Fig. 7, these Acr proteins exhibit distinctive structural characteristics. Fig. 7(a)–(d) display the three-dimensional structures of the four Acr proteins, accompanied by secondary structure strip maps below each structural representation. These Acr proteins possess diverse mechanisms of action: AcrVA2 induces mRNA degradation by recognizing and binding to the N-terminal polypeptide of Cas12a [54]. AcrF3 forms a homodimer that locks PaCas3 in an ADP-bound state, obstructs the entrance to the DNA-binding tunnel, and masks the linker region and C-terminal domain, thereby preventing Cascade complex binding and PaCas3 recruitment [55]. AcrIIA4 inhibits

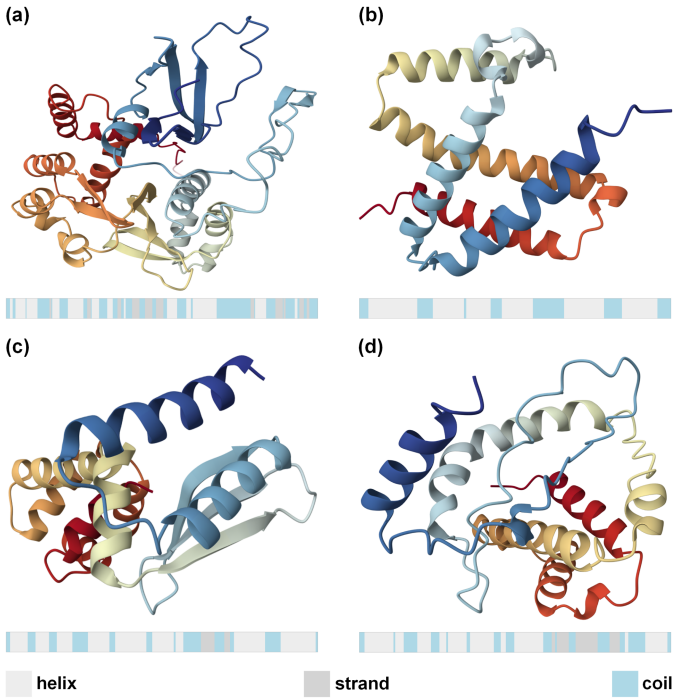


Fig. 7. The three-dimensional structures of four typical Acr proteins and their secondary structure distributions are presented herewith. The proteins are: (a) AcrVA2; (b)AcrF3; (c) AcrIIA4; and (d) AcrIIA6.

TABLE VI
CASE STUDY RESULTS FOR ANTI-CRISPR PROTEIN PREDICTIONS

| ID | Accession | Name | PACRGNN | PaCRISPR | AcrPred | DeepAcr | AcRanker | HMM |
|---|---|---|---|---|---|---|---|---|
| 0001 | WP_394950075.1 | AcrVA2 | 0.9975, ✓ | 0.3013, × | 0.2900, × | 0.0000, × | −, × | −, × |
| 0005 | WP_394595690.1 | AcrF3 | 1.0000, ✓ | 0.8443, ✓ | 0.7800, ✓ | 1.0000, ✓ | −, ✓ | −, × |
| 0017 | XFO04509.1 | AcrIIA4 | 1.0000, ✓ | 0.8463, ✓ | 0.3900, × | 0.7581, ✓ | −, ✓ | −, × |
| 0021 | WP_373110496.1 | AcrIIA6 | 1.0000, ✓ | 0.8818, ✓ | 0.5050, ✓ | 0.6256, ✓ | −, × | −, × |

This table summarizes prediction scores/results (✓ for correct prediction, × for incorrect prediction) across prevail methods and PACRGNN for Acr proteins in case study . For AcRanker, − corresponds to a threshold of -5. For HMM, − indicates the absence of a prediction probability.

DNA substrate binding and cleavage by interacting with the SpyCas9-sgRNA complex, occupying the PAM duplex binding site and blocking the RuvC active pocket [56]. AcrIIA6 functions through allosteric inhibition of St1Cas9 by binding to an allosteric site, altering its conformational dynamics, reducing DNA-binding affinity, and ultimately preventing DNA interaction within cells [57].

As demonstrated in Table VI, PACRGNN yielded accurate predictions for all four cases. Notably, for AcrVA2, only PACRGNN correctly identified it as an Acr protein with a high confidence score of 0.9975, while other methods like PaCRISPR (0.3013), AcrPred (0.2900), and DeepAcr (0.0000) failed to make accurate predictions. Furthermore, PACRGNN demonstrated consistent superior performance in identifying AcrVA2 homologs from different species, highlighting its robust capability in handling sequence diversity. For AcrIIA4 and AcrIIA6, PACRGNN achieved higher prediction probabilities compared to other methods. In the case of AcrF3, both PACRGNN and DeepAcr reached high accuracy levels, significantly outperforming other approaches.

### F. Webserver Development

To promote broader utilization of PACRGNN, we developed a user-friendly online prediction platform. This platform, accessible at https://www.acrs.top, allows researchers to effortlessly apply PACRGNN to their own protein sequence data, offering a practical tool for further research in antibiotic resistance and protein function prediction.

## IV. CONCLUSION

This study introduces PACRGNN, a novel graph neural network approach for Acr protein prediction. By leveraging a heterogeneous protein network, PACRGNN effectively captures the intricate relationships among Acr proteins and fully utilizes multiple protein features. This innovative method not only enhances predictive accuracy but also opens new avenues in computational biology and CRISPR/Cas research. Furthermore, PACRGNN offers researchers an online platform, facilitating accessible and efficient analysis. The robust performance of PACRGNN underscores its utility as a powerful tool for bioinformatics research and the discovery of therapeutically relevant Acr proteins.

The PACRGNN model shows strong performance in predicting Acr proteins but encounters methodological limitations related to computational efficiency when applied to large-scale networks. Constructing heterogeneous protein networks demands significant memory and processing resources, resulting in extended training and inference times as the dataset size grows. To improve scalability, future work will focus on algorithmic optimizations to reduce computational overhead while maintaining predictive accuracy.

## REFERENCES


[1] K. L. Maxwell, "The anti-CRISPR story: A battle for survival," *Mol. Cell*, vol. 68, pp. 8–14, Oct. 2017.
[2] K. S. Makarova et al., "Evolution and classification of the CRISPR–Cas systems," *Nature Rev. Microbiol.*, vol. 9, pp. 467–477, 2011.
[3] S. J. Labrie, J. E. Samson, and S. Moineau, "Bacteriophage resistance mechanisms," *Nature Rev. Microbiol.*, vol. 8, pp. 317–327, 2010.
[4] J. E. Samson, A. H. Magadán, M. Sabri, and S. Moineau, "Revenge of the phages: Defeating bacterial defences," *Nature Rev. Microbiol.*, vol. 11, pp. 675–687, 2013.
[5] N. Jia and D. J. Patel, "Structure-based functional mechanisms and biotechnology applications of anti-CRISPR proteins," *Nature Rev. Mol. Cell Biol.*, vol. 22, pp. 563–579, 2021.
[6] N. Birkholz et al., "Phage anti-CRISPR control by an RNA- and DNA-binding helix–turn–helix protein," *Nature*, vol. 631, pp. 670–677, 2024.
[7] M. M. Álvarez, J. Biayna, and F. Supek, "TP53-dependent toxicity of CRISPR/Cas9 cuts is differential across genomic loci and can confound genetic screening," *Nature Commun.*, vol. 13, Aug. 2022, Art. no. 4520.
[8] H. Manghwar et al., "CRISPR/Cas systems in genome editing: Methodologies and tools for sgRNA design, off-target evaluation, and strategies to mitigate off-target effects," *Adv. Sci.*, vol. 7, no. 6, 2020, Art. no. 1902312.
[9] N. D. Marino, R. Pinilla-Redondo, B. Csörgő, and J. Bondy-Denomy, "Anti-CRISPR protein applications: Natural brakes for CRISPR-Cas technologies," *Nature Methods*, vol. 17, pp. 471–479, May 2020.
[10] A. Pawluk et al., "Inactivation of CRISPR-Cas systems by anti-CRISPR proteins in diverse bacterial species," *Nature Microbiol.*, vol. 1, Jun. 2016, Art. no. 16085.
[11] B. J. Rauch et al., "Inhibition of CRISPR-Cas9 with bacteriophage proteins," *Cell*, vol. 168, pp. 150–158, Jan. 2017.
[12] J. Wang et al., "PaCRISPR: A server for predicting and visualizing anti-CRISPR proteins," *Nucleic Acids Res.*, vol. 48, pp. W348–W357, Jul. 2020.
[13] S. Hwang and K. L. Maxwell, "Meet the anti-CRISPRs: Widespread protein inhibitors of CRISPR-Cas systems," *CRISPR J.*, vol. 2, pp. 23–30, 2019.
[14] S. Eitzinger et al., "Machine learning predicts new anti-CRISPR proteins," *Nucleic Acids Res.*, vol. 48, pp. 4698–4708, May 2020.
[15] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, and C. H. Wu, and UniProt Consortium, "Uniref clusters: A comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, pp. 926–932, Mar. 2015.
[16] F. Y. Dao et al., "Acrpred: A hybrid optimization with enumerated machine learning algorithm to predict Anti-CRISPR proteins," *Int. J. Biol. Macromolecules*, vol. 228, pp. 706–714, Feb. 2023.
[17] K. G. Wandera et al., "Anti-crispr prediction using deep learning reveals an inhibitor of Cas13b nucleases," *Mol. Cell*, vol. 82, no. 14, pp. 2714–2726, 2022.
[18] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[19] Z.-A. Huang, P. Hu, L. Hu, Z.-H. You, K. C. Tan, and Y.-A. Huang, "Toward multilabel classification for multiple disease prediction using gut microbiota profiles," *IEEE Trans. Neural Networks Learn. Syst.*, Sep. 12, 2024.

[20] L. Wong, L. Wang, Z.-H. You, C.-A. Yuan, Y.-A. Huang, and M.-Y. Cao, "GKLOMLI: A link prediction model for inferring miRNA–lncRNA interactions by using Gaussian kernel-based method on network profile and linear optimization algorithm," *BMC Bioinf.*, vol. 24, May 2023, Art. no. 188.

[21] J. Zheng, H.-C. Yi, and Z.-H. You, "Equivariant 3D-conditional diffusion model for de novo drug design," *IEEE J. Biomed. Health Inform.*, vol. 29, no. 3, pp. 1805–1816, Mar. 2025.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018, *arXiv:1710.10903*.

[23] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1025–1035.

[24] S. Bittrich et al., "RCSB protein data bank: Efficient searching and simultaneous access to one million computed structure models alongside the PDB structures enabled by architectural advances," *J. Mol. Biol.*, vol. 435, Jul. 2023, Art. no. 167994.

[25] Z. Lin et al., "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, pp. 1123–1130, Mar. 2023.

[26] O. S. Nnyigide, T. O. Nnyigide, S.-G. Lee, and K. Hyun, "Protein repair and analysis server: A web server to repair PDB structures, add missing heavy atoms and hydrogen atoms, and assign secondary structures by amide interactions," *J. Chem. Inf. Model.*, vol. 62, no. 17, pp. 4232–4246, 2022.

[27] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J. Mol. Biol.*, vol. 147, pp. 195–197, Mar. 1981.

[28] Y. Zhang and J. Skolnick, "TM-align: A protein structure alignment algorithm based on the TM-score," *Nucleic Acids Res.*, vol. 33, pp. 2302–2309, Apr. 2005.

[29] B. Wang et al., "Similarity network fusion for aggregating data types on a genomic scale," *Nature Methods*, vol. 11, pp. 333–337, 2014.

[30] C. Dong et al., "Anti-CRISPRdb: A comprehensive online resource for anti-CRISPR proteins," *Nucleic Acids Res.*, vol. 46, pp. D393–D398, Jan. 2018.

[31] J. Wang et al., "AcrHub: An integrative hub for investigating, predicting and mapping anti-CRISPR proteins," *Nucleic Acids Res.*, vol. 49, pp. D630–D638, Jan. 2021.

[32] J. Bondy-Denomy et al., "A unified resource for tracking anti-CRISPR names," *CRISPR J.*, vol. 1, pp. 304–305, 2018.

[33] W. Li and A. Godzik, "CD-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences," *Bioinformatics*, vol. 22, pp. 1658–1659, Jul. 2006.

[34] National Center for Biotechnology Information (NCBI), [Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; [1988] – [cited 2024 20, Nov.]. [Online]. Available: https://www.ncbi.nlm.nih.gov/

[35] A. Pande et al., "Pfeature: A tool for computing wide range of protein features and building prediction models," *J. Comput. Biol.*, vol. 30, pp. 204–222, 2023.

[36] J. Wang et al., "POSSUM: A bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles," *Bioinformatics*, vol. 33, pp. 2756–2758, Sep. 2017.

[37] P. J. A. Cock et al., "Biopython: Freely available Python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, pp. 1422–1423, Jun. 2009.

[38] C. Chen et al., "Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier," *Comput. Biol. Med.*, vol. 123, Aug. 2020, Art. no. 103899.

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[40] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins, Struct., Function, Bioinf.*, vol. 57, no. 4, pp. 702–710, 2004.

[41] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proc. Nat. Acad. Sci.*, vol. 89, pp. 10915–10919, Nov. 1992.

[42] X. Zheng et al., "Fusing multiple protein-protein similarity networks to effectively predict lncRNA-protein interactions," *BMC Bioinf.*, vol. 18, Oct. 2017, Art. no. 420.

[43] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," May 2017, *arXiv:1608.03983*.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.

[46] J. Du, S. Zhang, G. Wu, J. M. F. Moura, and S. Kar, "Topology adaptive graph convolutional networks," Feb. 2018, *arXiv:1710.10370*.

[47] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," in *Proc. Int. Joint Conf. Artif. Intell. Org.*, 2021, pp. 1548–1554.

[48] Y. Li et al., "AcrNet: Predicting anti-CRISPR with deep learning," *Bioinformatics*, vol. 39, 2023, Art. no. btad259.

[49] L. Zhu, X. Wang, F. Li, and J. Song, "Preacrs: A machine learning framework for identifying anti-CRISPR proteins," *BMC Bioinf.*, vol. 23, Oct. 2022, Art. no. 444.

[50] T. Tieleman, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Networks Mach. Learn.*, vol. 4, no. 2, 2012, Art. no. 26.

[51] A. Rives et al., "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proc. Nat. Acad. Sci.*, vol. 118, no. 15, 2021, Art. no. e2016239118.

[52] S. R. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, pp. 755–763, 1998.

[53] S. R. Eddy, "Accelerated profile HMM searches," *PLoS Comput. Biol.*, vol. 7, 2011, Art. no. e1002195.

[54] N. D. Marino et al., "Translation-dependent downregulation of CAS12A mRNA by an anti-CRISPR protein," *BioRxiv*, 2023.

[55] X. Wang et al., "Structural basis of Cas3 inhibition by the bacteriophage protein AcrF3," *Nature Struct. Mol. Biol.*, vol. 23, pp. 868–870, Sep. 2016.

[56] H. Yang and D. J. Patel, "Inhibition mechanism of an anti-CRISPR suppressor AcrIIA4 targeting SpyCas9," *Mol. Cell*, vol. 67, pp. 117–127, Jul. 2017.

[57] O. Fuchsbauer et al., "Cas9 allosteric inhibition by the anti-CRISPR protein AcrIIA6," *Mol. Cell*, vol. 76, pp. 922–937, Dec. 2019.