Article

Generative and Contrastive Self-Supervised Learning for Virulence Factor Identification Based on Protein-Protein Interaction Networks

Yalin Yao , Hao Chen , Jianxin Wang [*] , Yeru Wang [*]

Posted Date: 3 June 2025

doi: 10.20944/preprints202506.0217.v1

Keywords: virulence factor identification; protein-protein interaction; graph neural network; self-supervised learning

Article

Generative and Contrastive Self-Supervised Learning for Virulence Factor Identification Based on Protein-Protein Interaction Networks

Yalin Yao [1,†], Hao Chen [1,†], Jianxin Wang [1,*] and Yeru Wang [2,*]

1 School of Information, Beijing Forestry University, Beijing 100083, China
2 Risk Assessment Division 1, China National Center for Food Safety Risk Assessment, Beijing 100022, China
* Correspondence: wangjx@bjfu.edu.cn (J.W.); wangyeru@cfsa.net.cn (Y.W.)
† These authors contributed equally to this work.

Abstract: Virulence factors (VFs), produced by pathogens, facilitate pathogenic microorganisms to invade, colonize, and damage the host cells. Accurate VF identification advances pathogenic mechanism understanding and provides novel anti-virulence targets. Existing models primarily utilize protein sequence features while overlooking the systematic protein-protein interaction (PPI) information, despite pathogenesis typically results from coordinated protein-protein actions. Moreover, a severe imbalance exists between virulence and non-virulence proteins, which causes existing models trained on balanced datasets by sampling to fail to incorporate proteins' inherent distributional characteristics, thus restricting generalization to real-world imbalanced data. To address these challenges, we propose a novel Generative and Contrastive self-supervised learning framework for Virulence Factor identification (GC-VF) that transforms VF identification into an imbalanced node classification task on graphs generated from PPI networks. The framework encompasses two core modules: the generative attribute reconstruction module learns attribute space representations via feature reconstruction, capturing intrinsic data patterns and reducing noise; the local contrastive learning module employs node-level contrastive learning to precisely capture local features and contextual information, avoiding global aggregation losses while ensuring node representations truly reflect inherent characteristics. Comprehensive benchmark experiments demonstrate that GC-VF outperforms baseline methods on naturally imbalanced datasets, exhibiting higher accuracy and stability, providing a potential solution for accurate VF identification.

Keywords: virulence factor identification; protein-protein interaction; graph neural network; self-supervised learning

1. Introduction

Virulence factors (VFs) are essential for pathogenesis, enabling bacteria to adhere to and invade host cells, utilize host cell nutrients for survival and reproduction, evade host immune defenses, and secrete toxins that destroy host cells. Antibiotics remain the primary therapeutic agents for bacterial infections, exerting their effects by either killing bacteria or inhibiting their growth and reproduction. For example, this can be accomplished by disrupting the bacterial cell wall and cell membrane, inhibiting bacterial DNA/RNA and protein synthesis, and blocking folate synthesis [1]. Since antibiotics directly suppress bacterial survival, large-scale antibiotic use accelerates the evolution of bacterial antibiotic resistance. Antibiotic resistance has become a significant global public health challenge [2]. To relieve the pressure stemming from the rapid evolution of bacterial resistance, relevant research has proposed anti-virulence strategies aimed at achieving therapeutic effects by targeting VFs to suppress the manifestation of bacterial virulence. These strategies include inhibiting bacterial toxins gene expression and toxins transmission, disrupting bacterial adhesion capabilities, and interfering with bacterial communication [3]. Therefore, identifying bacterial VFs and developing an in-depth understanding of their mechanisms of action provide potential targets for the development of anti-virulence therapeutic strategies.

Due to the importance of this problem, a large number of studies have emerged in the field of VF identification. Methods based on sequence similarity aim to obtain homologous sequences corresponding to a given query and determine the type of the given query based on the type of homologous sequences. However, due to the diversity of VFs, which are involved in various pathogenic processes such as adhesion, invasion, and toxicity, and exhibit species-specificity and host-specificity, many VFs may show only insignificant similarity to known protein sequences [4]. Moreover, pathogens continuously adapt to the environment during the evolutionary process, and their VFs are prone to alterations. Therefore, traditional sequence alignment methods such as BLAST [5] have limited performance in identifying diverse VFs and VFs with distant evolutionary relationships.

To address this challenge, methods based on predefined protein features have been proposed. These methods extract features such as protein sequences, physicochemical properties, and evolutionary information, and combine them with traditional machine learning models and deep learning models for VF identification. For example, VirulentPred [6] takes amino acid composition (ACC), dipeptide composition (DPC), higher-order dipeptide composition, and position-specific scoring matrixes (PSSMs) as inputs for the first layer of a two-stage cascade support vector machine (SVM). The results from the first layer are then cascaded to the second layer SVM classifier for training to generate the final classifier, which outperforms SVM classifiers based on single or multiple sequence features only in the first layer. MP3 [4] uses AAC and DPC as inputs for SVM, while leveraging the hidden Markov model (HMM) to analyze domain information in a local MiniPfam database constructed based on the Pfam database. The results from SVM and HMM are integrated according to specific rules, providing a new approach and promising approach for predicting virulence proteins in large-scale genomic or metagenomic datasets. PBVF [7] conducts relevant analysis and prediction of VFs by taking DPC and multiple sequence similarity-based features as the inputs for SVM. Regarding the critical issue of negative dataset selection, this research adopts the NExIGO method based on Gene Ontology (GO) annotations to construct the negative dataset. The research results indicate that direct sequence similarity is crucial in the identification of VFs, and thus this characteristic should be fully utilized to improve the accuracy of analysis and prediction. DeepVF [8] extracts features based on sequence, physicochemical properties, and evolutionary information as inputs for four classic traditional machine learning models. It maps protein sequences to numerical values in alphabetical order, generating 10-dimensional features based on different window lengths for three deep learning models. By employing a stacking strategy, DeepVF effectively combines these baseline models, significantly enhancing model performance, and providing an important reference for classification problems in bioinformatics. VF-Pred [9] is based on sequence features and physicochemical properties, and can successfully generate the sequence similarity features proposed in reference [7]. Additionally, it introduces the sequence alignment feature, Seg-Alignment, to capture the percentage of the best sequence alignment with the negative and positive datasets. These features are input into traditional machine learning models, and various ensemble methods including stacking, voting, and boosting are used together to enhance the classification performance. Experimental results show that the sequence alignment feature significantly improves the accuracy of the adopted machine learning algorithms. Methods based on predefined protein features usually rely on the knowledge of domain experts for design, and thus have certain limitations in feature selection, especially for manually extracted sequence features, which makes it difficult to comprehensively cover the potential information of protein sequences. Since the selection of predefined features is crucial to model performance, the model performance often highly depends on the quality of feature engineering.

With the rapid development of deep learning, natural language processing (NLP) has provided a new research perspective for protein representation learning [10,11]. Essentially, protein sequences can be viewed as a kind of "language". By leveraging the framework of language models, it is possible not only to capture the local patterns within the sequences but also to reveal their global connections. Compared with traditionally manually extracted features, protein features extracted based on language models are more likely to unearth the deep semantic information and complex patterns of protein sequences. The DTVF method proposed in reference [12] is based upon the protein sequence model ProtT5 [13], which is pre-trained on large-scale protein sequence data, as a protein feature extractor. It adopts a dual-channel architecture model, combining the long short-term memory network (LSTM) module with the convolutional neural network (CNN) module. And it introduces a dot-product self-attention layer into each module respectively. This model can enhance the accuracy and efficiency of VF identification. Another approach, GTAE-VF [14] aided with a graph transformer autoencoder for VF identification, leverages the ESM-2 [15] language model to obtain amino acid feature vectors and is the first to utilize three-dimensional structural information of protein predicted by ESMFold, finally transforming VF identification into a graph-level prediction task. The encoder-decoder framework integrates graph convolutional networks(GNNs) and Transformer structures, effectively capturing long-range correlations, thereby further improving predictive performance. Methods based on language models for protein features indicate that the introduction of high-quality features may help models better capture the relationship between VFs and sequence patterns. The transition from methods based on predefined features to those based on language model features marks a shift in VF identification models from traditional manual feature extraction to data-driven automated representation learning.

Most relevant studies primarily focus on the properties of proteins, including sequences, physicochemical properties, evolutionary information, and structural characteristics, to identify VFs. However, protein-protein interaction (PPI) network information has also been demonstrated to have potential in VF prediction. Reference [16] proposes a method to predict VFs based on the PPI network through the number, type, and interaction weight of neighbor nodes. Reference [17] further enriches the direct interaction neighbors of proteins into KEGG pathways, calculates the pathway enrichment scores, and uses the random forest model for prediction. However, these methods only utilize direct protein interactions and explicit features, such as the number and labels of neighbors, pathway enrichment scores, which are limited to shallow information within the direct neighborhood. They lack comprehensive modeling of broader contextual and deeper information on PPI networks and do not fully leverage the biological features of proteins themselves, such as sequence and structure. Moreover, explicit features rely on known protein functional annotations, and the models may be restricted by incomplete annotation information, making it difficult to comprehensively capture the implicit properties of unannotated proteins. Since the PPI network can be naturally modeled and mined using a graph structure, GNNs can be taken into account to address the above limitations.

GNNs can integrate and simultaneously deal with protein sequence features and interaction information, and effectively aggregate node information across network layers through message-passing mechanisms to capture deeper, non-linear features [18] Most GNNs are based on the homophily hypothesis [19–21], which essentially reflects the principle that "birds of a feather flock together" [22], meaning that interconnected nodes in a graph tend to belong to the same category. Under the homophily assumption, node representations will be smoothed through the aggregation process, with each node acquiring additional information from neighboring nodes that likely share the same label. We employ three commonly used homophily metrics: edge homophily [23,24], node homophily [25], and class homophily [26] to evaluate the homophily of three PPI networks: *Salmonella enterica* serovar Typhimurium LT2, *Campylobacter jejuni* NCTC, and *Staphylococcus aureus* NCTC 8325. In addition to comparing with three homophily graphs commonly used in node classification tasks, we also compare the homophily of PPI networks with the randomly generated networks. We make sure that the randomly generated networks be of the same size as the corresponding PPI networks, in order to eliminate the influence of network size differences and focus precisely on the homophily aspect during the comparison process. For each PPI network, 30 random networks are generated according to their node count, edge count, and category proportions. The three homophily metrics are calculated and subject to t-tests (p-values all less than 0.05), with the mean homophily metrics being retained. The evaluation results are shown in Figure 1.

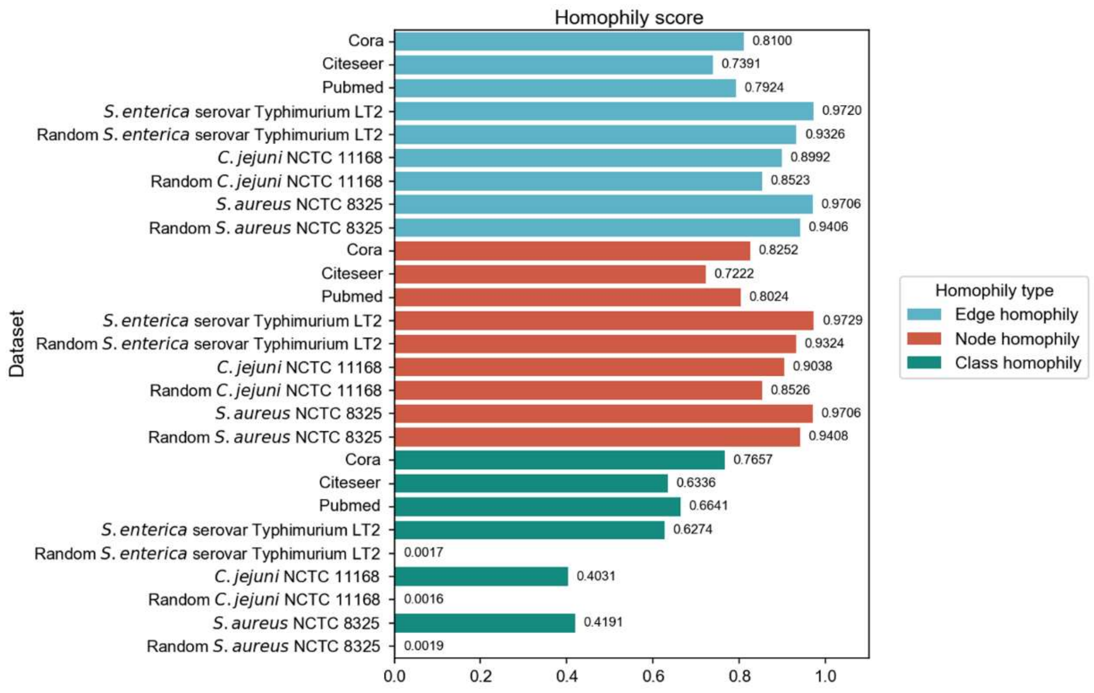


Figure 1. Comparison of homophily evaluation results across different graph types.

The original PPI networks exhibit higher node and edge homophily compared to commonly used homophily graphs for node classification tasks, although class homophily is relatively low. Overall, the homophily metrics of the original PPI networks are promising, suggesting that homophily-based GNNs are likely to effectively aggregate node information to obtain beneficial embeddings. Compared to random networks of the same scale, the original PPI networks show superior performance across all homophily metrics. The high node and edge homophily observed in both original PPI networks and random networks of the same size may reflect the impact of class imbalance in PPI networks, where dominant class nodes enhance these two metrics. The numerical advantage of dominant class nodes increases the probability of connections between nodes within this class. The difference in class homophily between original PPI networks and random networks indicates significant collaborative relationships among minority class nodes (virulence proteins), suggesting their coordinated participation in pathogenesis-related biological processes. Furthermore, based on the STRING [27] database, KEGG pathway enrichment analysis is performed on the interactions between virulence proteins, and the average local clustering coefficient among virulence proteins in these pathways is calculated, with results shown in Figure 2. The high average local clustering coefficient indicates that virulence proteins cooperatively participate in pathogenesis-related biological activities through tight interactions, involving multiple aspects such as bacterial secretion regulation, bacterial invasion of host cells, synthesis and assembly of bacterial structural components, and coordination of quorum sensing. The intimate interactions of virulence proteins in PPI networks facilitate learning virulence protein features through message passing.

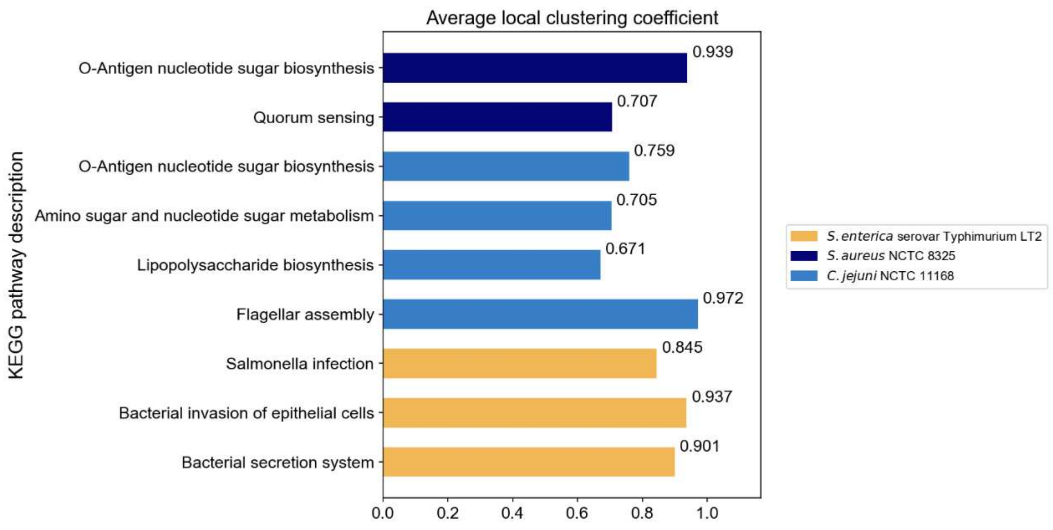


Figure 2. The average local clustering coefficient of virulence proteins in KEGG pathways.

The natural distribution of proteins in bacterial systems exhibits an inherent imbalance between virulence and non-virulence proteins, with virulence proteins constituting a minority class. Traditional modeling approaches have attempted to address this imbalance through various sampling techniques. Under-sampling strategies are commonly employed for strain-specific datasets to reduce the number of non-virulence proteins moderately (such as maintaining a ratio of 1:5 between the size of the negative dataset and the positive dataset), thereby mitigating class imbalance effects. Balanced sampling is performed for multi-strain datasets by randomly selecting negative samples to match the number of positive samples to construct a balanced dataset. Additionally, repeated learning of virulence proteins is performed to enhance model performance. However, these sampling approaches may inadvertently compromise the inherent diversity of the training data, potentially limiting the model's generalization capabilities. In imbalanced datasets, the majority class often dominates the learning process, causing the decision boundary to be biased towards the majority class. Self-supervised learning [28] has emerged as a promising alternative, offering the ability to extract effective representations from unlabeled data. This approach has the potential to capture more comprehensive and generalizable information that is inherent in the data itself, regardless of the skewed label distribution, demonstrating particular promise in addressing imbalanced classification challenges [29,30] Existing self-supervised graph learning methodologies can be categorized into three distinct paradigms [31]. The generative approach leverages intrinsic graph information as self-supervised signals, focusing on reconstructing specific components of the input data to develop robust graph data representations [32–34]. Contrastive methods emphasize the analysis of consistency between different views to extract essential features and structural patterns inherent in the data [35,36]. The predictive approach autonomously generates informative labels as supervision signals, addressing the relationships between data and their corresponding labels [37,38].

In the field of class-imbalanced node classification, multiple innovative self-supervised learning methods have been applied to meet the challenges. INS-GNN [39] implements generative self-supervised pre-training to reconstruct the origin graph structure, effectively mitigating the inherent label bias present in imbalanced datasets. This method further incorporates self-training for pseudo-label assignment to unlabeled nodes and utilizes self-supervised edge enhancement to modify the structural characteristics of minority nodes, thereby amplifying their influence in the learning process. The GraphMixup [40] improves the class-imbalanced node classification on graphs based on predictive self-supervised learning. It conducts semantic-level feature mixing by building a semantic relationship space and mixing edges while using an edge predictor trained via two context-based self-supervised tasks. Existing self-supervised learning approaches for addressing class-imbalanced graph node classification have been predominantly validated in conventional graph domains, such as social networks and citation networks, yet with limited exploration beyond these traditional contexts. This constraint is particularly noteworthy in the domain of bioinformatics, where complex graph structures such as PPI networks present unique challenges and opportunities that remain largely unexplored within the current methodological framework.

We present a novel framework for VF identification based on PPI networks, which integrates both generative and contrastive self-supervised learning strategies. Specifically, we construct local subgraphs centered on target proteins and generate multi-view representations through data augmentation strategies. These subgraph views are processed through GNNs to learn latent node representations. During the self-supervised learning phase, generative learning captures the inherent distributional characteristics of the data through node attribute reconstruction, while contrastive learning performs node-level comparisons between different views to effectively capture both local node features and contextual information. The learned latent representations are then utilized by a classifier for VF prediction. Experimental results on multiple real-world PPI network datasets demonstrate the effectiveness and robust performance of the new model we have proposed. The key contributions can be summarized as follows:

- We employ GNNs to identify VFs leveraging PPI networks. This approach inte-grates both topological information from PPI networks and protein sequence fea-tures. Notably, we pioneer in transforming the VF identification task into a class-imbalanced node classification problem within the graph domain.
- We propose a novel framework for VF identification that combines generative and contrastive self-supervised learning. Through attribute reconstruction and multi-view contrast, these two approaches work synergistically to enhance model performance in imbalanced classification tasks.

We conduct comprehensive experimental evaluations across three PPI datasets, benchmarking our approach against available VF identification baseline models. The experimental results consistently demonstrate that our method achieves notable performance across multiple evaluation metrics.

2. Materials and Methods

As illustrated in Figure 3, the GC-VF framework's overall architecture comprises three components: protein sequence encoding, generative and contrastive self-supervised learning, and VF probability prediction. During the protein sequence encoding phase, the model encodes amino acid sequence features through multiple layers of deep neural networks, transforming them into protein sequence representations that are conducive to subsequent tasks. Subsequently, leveraging PPI networks, the model employs a hierarchical sampling strategy to construct local subgraphs of target proteins. Through data augmentation techniques, multi-view features essential for contrastive learning are generated. These views undergo information aggregation via GNNs, which enhance node representations by capturing more effective features.

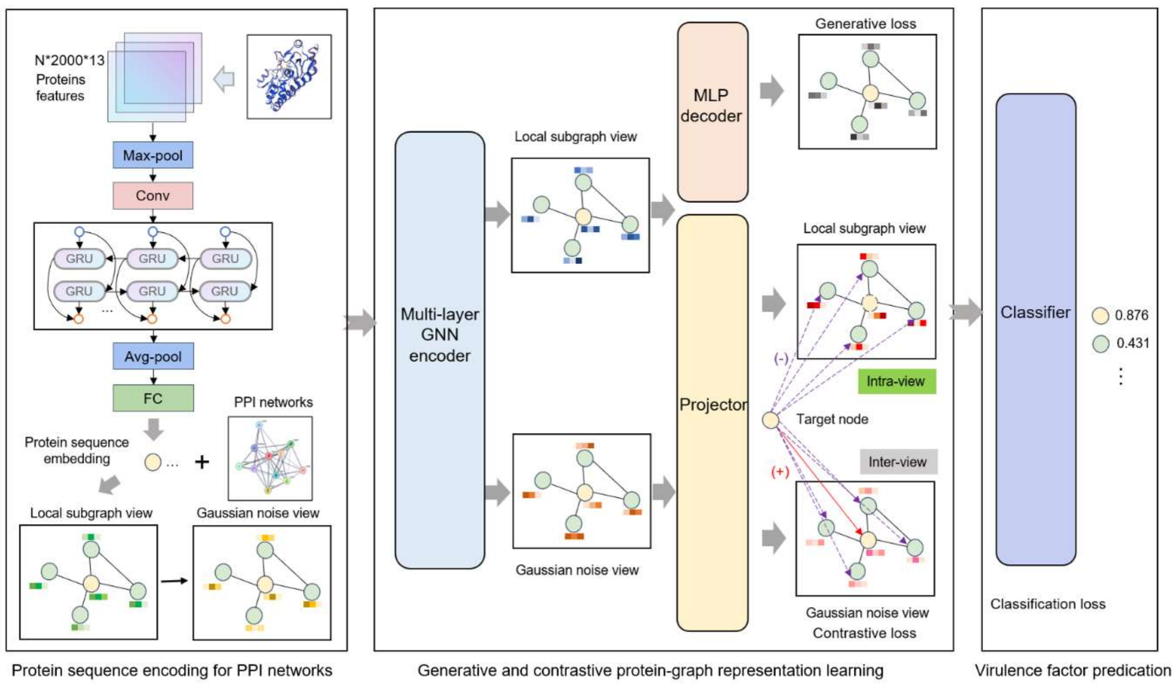


Figure 3. The overall framework of the GC-VF.

In the self-supervised learning module, generative learning captures the intrinsic feature distribution of proteins by node attribute reconstruction tasks, while contrastive learning focuses on optimizing discriminative features between nodes by minimizing feature distances between positive samples and maximizing those between negative samples. Combining these approaches enables the model to take account of both global distribution and local variations in the data. Node-level generative and contrastive objectives inhibit decision-making from being dominated by global structural patterns, thus more precisely capturing the inherent characteristics of individual nodes. Finally, the model inputs the learned node representations into a classifier and utilizes the optimized features for VF prediction.

2.1. Datasets

Since the proportion of VFs in most bacterial strains is relatively low, three species with a moderate proportion of VFs are selected for analysis: *Salmonella enterica* serovar Typhimurium LT2, *Campylobacter jejuni* NCTC 11168, and *Staphylococcus aureus* NCTC 8325. Protein sequence data and PPI network data for these three strains are obtained from the STRING database while corresponding VFs' information is retrieved from the VFDB database [41]. Following the combined score threshold settings of the STRING database, only interaction relationships with a combined score greater than or equal to 0.4 are retained. To achieve a more comprehensive understanding of the characteristics inherent in each dataset, we calculate the basic properties of the PPI networks for these three strains and the class imbalance ratio (defined as the ratio of the number of samples in the largest class to that in the smallest class) [42] for each dataset. The relevant data is summarized in Table 1. Specifically, the numbers of VFs in the three strains are 156, 130, and 97, respectively. Furthermore, the class imbalance ratios in the selected datasets are 27.53, 11.48, and 31.72, respectively, all of which are notably high, indicating significant class imbalance.

Table 1. Dataset Statistics.

Dataset	Nodes	Edges	VFs	Imbalance Ratio
S. enterica serovar Typhimurium LT2	4451	86605	156	27.53
C. jejuni NCTC 11168	1623	81710	130	11.48

S. aureus NCTC 8325	2847	79578	87	31.72

2.2. Protein Sequence Encoding

Currently, significant advances have been achieved in extracting protein features from amino acid sequences through the application of protein language models [43,44]. In particular, the extracting method of amino acid embedding proposed in reference [45] has demonstrated substantial improvements in PPI-related tasks. This method implements a dual-component embedding representation: the first leverages amino acid sequences to compute co-occurrence similarities via a pre-trained Skip-Gram model [46], while the second component employs one-hot encoding to capture electrostaticity and hydrophobicity similarities among amino acids [47]. Extending this amino acid embedding framework, Reference [48] develops a multi-layer deep neural network architecture to generate protein feature representations optimized for PPI network input. This architecture combines four types of layers, namely convolutional layers (Conv1d), pooling layers, bidirectional gated recurrent units (BiGRU), and fully connected layers (FC). In our present work, we utilize this combination of amino acid embedding method and multi-layer deep neural network architecture with an additional dropout layer to generate node feature representations that effectively characterize both global and local protein sequence properties. These refined feature embeddings establish a robust foundation for subsequent graph learning processes that incorporate PPI networks.

2.3. Generative and Contrastive Protein Representation Learning

2.3.1. Graph View Establishment

Based upon the protein sequence encoding module, which is responsible for transforming protein sequences into node feature embeddings essential for GNNs, we employ a hierarchical neighborhood sampling method to construct local subgraphs [21] to further mine the structural information within PPI networks. Specifically, within each batch iteration, we initially select target proteins as center nodes from the PPI network through sampling without replacement. Next, we construct local structures through a two-layer neighborhood sampling approach: randomly sampling K_1 nodes from first-order neighbors that have direct physical interactions with the center protein, and further sampling K_2 second-order neighbor nodes for each selected first-order neighbor, thereby constructing an original view centered on the target protein. The first-order neighbors capture direct interaction relationships, while the incorporation of the second-order neighbors facilitates the detection of potential indirect interaction patterns. Then, we introduce Gaussian-distributed random noise into the protein features of the original view, forming an augmented view. Subsequently, both the original and augmented views are processed as dual-stream inputs through the GNNs, enabling the aggregation of multi-scale neighborhood information and facilitating the learning of contextual protein representations within the PPI network.

2.3.2. Generative Attribute Reconstruction

The basic principle of an autoencoder is to map the input data into a latent representation space through an encoder, followed by the reconstruction of the original input through a decoder. Within the context of graph data, this architecture not only facilitates the learning of latent node representations but also enables the capture of intrinsic association patterns between nodes through reconstruction tasks in an unsupervised manner.

For node attribute reconstruction, we implement a strategic approach by not employing target node anonymization (setting features to zero), instead enabling the original attributes of target nodes to participate directly in the information aggregation process. This decision stems from the consideration that virulence proteins do not manifest strong homophily in their interaction patterns. Therefore, the non-anonymized reconstruction approach facilitates more precise capture of critical shared features between target nodes and their neighborhood nodes, while circumventing the introduction of irrelevant noise that might emerge from excessive reliance on neighborhood information.

The PPI network is formally defined as $G=(V,E)$, where V denotes the set of protein nodes and E represents the set of interaction relationships. For each target node $i\in V$, views G_i^{ori} and G_i^{aug} derive from sampling and feature augmentation procedures, respectively. Notably, the augmentation operation exclusively modifies node features while preserving structural consistency between the two views, yielding identical adjacency matrices.

The GNN architecture facilitates node representation updates through iterative neighbor information propagation, effectively mapping high-dimensional features onto a low-dimensional space. In the initial phase, the original subgraph G_i^{ori} of target node i is processed through a multi-layer GNN encoder, which can be formally represented as

$$\mathbf{h}_i^{(l)}=\mathrm{GNN}_{enc}\left(\mathbf{h}_i^{(l-1)},\mathbf{A}_i\right) \quad (1$$

where $\boldsymbol{h}_i^{(l)}$ represents the latent embedding of node i at the l-th layer of the GNNs. $\mathbf{A}_i$ is the adjacency matrix of the subgraph sampled for node i. The $\mathrm{GNN}_{enc}(\cdot)$ is a GNN encoder consisting of L layers. The multi-layer structure enables to gradually aggregate node information from more distant neighbors and effectively capture the multi-level collaborative patterns among proteins. Each layer of the GNN is defined as follows

$$\mathbf{h}_i^{(l)}=\mathrm{UPDATE}(\mathbf{h}_i^{(l-1)},\mathrm{AGGREGATE}(\{\mathbf{h}_j^{(l-1)}, for\ each\ j\in S(i)\})) \quad (2)$$

where $S(i)$ represents the set of adjacent nodes in the subgraph sampled for node i. In particular, at the input stage, $\mathbf{h}_i^{(0)}=\mathbf{x}_i^{ori}$, representing the initial encoded feature of the protein sequence of the node.

The architecture allows for various GNN implementations, including Graph Convolutional Network (GCN) [19], Graph Attention Network (GAT) [20], and Graph Isomorphism Network (GIN) [49], each employing distinct neighbor aggregation strategies. In this study, we employ GraphSAGE [21] as the backbone network due to its robust generalization capabilities, achieved mainly by training a set of aggregator functions rather than individual node embeddings. The specific formulation of GraphSAGE is as follows

$$\mathbf{h}_i^{(l)}=\sigma(\mathbf{W}\cdot\mathrm{CONCAT}(\mathbf{h}_i^{(l-1)},\mathrm{AGGREGATE}(e_{ji}\mathbf{h}_j^{(l-1)}, for\ each\ j\in S(i)))) \quad (3)$$

where $\sigma(\cdot)$ is the ReLU [50] activation function, $\mathbf{W}$ represents the weight matrix, and e_{ji} represents the weight of the edge between the node pair (j,i). The incorporation of edge weights enables the model to differentiate varying interaction strengths between protein pairs, reflecting their distinct collaborative importance in biological processes. The model firstly performs weighted aggregation of features from all neighboring nodes of node i in the subgraph and then concatenates the aggregated information with the feature of node i at the $(l-1)$-th layer to obtain the latent representation at the l-th layer.

The decoder transforms the encoder-generated node embeddings back to original node features, guiding the encoder toward more significant node representations while keeping capturing latent structural graph relationships. The decoder utilizes a multi-layer perceptron (MLP) with an architecture matching the encoder in terms of layer count and parameters, which can be denoted as

$$\hat{\mathbf{x}}_{\boldsymbol{i}}^{\boldsymbol{ori}}=\mathrm{MLP}_{dec}(\mathbf{h}_i^{(L)}) \quad (4)$$

where the MLP decoder implements two-layer architectures: $\mathrm{MLP}(\mathbf{h}_{\boldsymbol{i}})=\mathbf{W}^{(2)}\sigma(\mathbf{W}^{(1)}\mathbf{h}_{\boldsymbol{i}})$. Specifically, the decoder transforms the L-th layer node representation $\mathbf{h}_i^{(L)}$ into a reconstructed original encoded feature $\hat{\mathbf{x}}_{\boldsymbol{i}}^{\boldsymbol{ori}}$ through a series of non-linear transformations.

The generative self-supervised learning framework optimizes itself by minimizing the reconstruction error between decoder-generated and origin embeddings. The mean squared error (MSE) quantifies the reconstruction loss, which can be specifically written as

$$\mathcal{L}_{gen}=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{d}\parallel \mathbf{x}_i^{ori}-\hat{\mathbf{x}}_i^{ori}\parallel^2 \quad (5)$$

where d denotes the dimension of the original encoded feature $\mathbf{x}_i^{ori}$. Through minimization of reconstruction loss, the model learns to extract feature patterns shared between the target node and neighboring nodes during reconstruction.

2.3.3. Multi-View Local-Local Contrasting

Contrastive learning, an unsupervised learning method, facilitates the exploration of structural information and node relationships to learn feature similarities and differences, offering novel approaches for identifying potential virulence proteins. However, virulence proteins, within PPI networks, compared to their non-virulence counterparts, typically struggle to form stable subgraph features. This characteristic presents significant challenges for graph-level contrastive learning in extracting positive sample features from heavily-imbalanced virulence protein subgraphs. Pooling subgraph features as positive samples in contrastive learning may disproportionately reflect non-virulence protein patterns, potentially compromising virulence protein feature learning. To address this challenge, we implement a node-level (local-local) contrastive scheme enabling independent optimization of individual node features.

Specifically, the encoded features $\mathbf{x}_i^{ori}$ from original-view and $\mathbf{x}_i^{aug}$ from augmented-view of the target node i are processed through a shared GNN encoder $\mathrm{GNN}_{enc}(\cdot)$ to generate node embeddings $\mathbf{h}_i^{ori}$ and $\mathbf{h}_i^{aug}$. To enhance representation quality, we incorporate a projection head Projector(·), implementing a two-layer MLP that projects node features onto a new feature space. Within this space, positive sample pair features exhibit increased proximity while negative sample pairs maintain greater separation, thereby improving representation quality [51]. This process is formalized as

$$\mathbf{z}_i^{ori} = \mathrm{Projector}(\mathbf{h}_i^{ori}) \tag{6}$$

$$\mathbf{z}_i^{aug} = \mathrm{Projector}(\mathbf{h}_i^{aug}) \tag{7}$$

During node-level contrastive instance sampling, $\mathbf{z}_i^{ori}$ from the original view serves as the anchor, while $\mathbf{z}_i^{aug}$ from the augmented view as the positive sample. For negative sample selection, inspired by GRACE [52], we employ a mixed sampling strategy across and within views: node embeddings excluding node i in the original view are selected as negative samples to enhance intra-view relationship understanding, while nodes other than the positive sample in the augmented view are selected to improve inter-view discrimination capability.

The contrastive self-supervised learning objective simultaneously minimizes representation distance between similar samples while maximizing it between dissimilar samples. We employ InfoNCE [51] to compute contrastive loss, maximizing mutual information between random events, which is a practical and powerful tool for extracting shared data information. The InfoNCE loss can be formulated as

$$\mathcal{L}_{con} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\mathrm{sim}\left(\mathbf{z}_i^{ori}, \mathbf{z}_i^{aug}\right)/\tau\right)}{\exp\left(\mathrm{sim}\left(\mathbf{z}_i^{ori}, \mathbf{z}_i^{aug}\right)/\tau\right) + \mathrm{NSC}_i} \tag{8}$$

where $\mathrm{sim}(\mathbf{h}_v, \mathbf{h}_u)$ denotes the cosine similarity between the node features of v and u, calculated as $\frac{\mathbf{h}_v \mathbf{h}_u}{\|\mathbf{h}_v\|\|\mathbf{h}_u\|}$. The temperature coefficient τ modulates similarity distribution sharpness, with smaller values producing more concentrated distributions and larger values yielding smoother distributions. NSC_i represents negative sample contribution to the loss function, computed as

$$\mathrm{NSC}_i = \sum\nolimits_{j \in G_i^{ori}, j \neq i} \exp\left(\mathrm{sim}\left(\mathbf{z}_i^{ori}, \mathbf{z}_j^{ori}\right)/\tau + \sum\nolimits_{k \in G_i^{aug}, k \neq i} \exp\left(\mathrm{sim}\left(\mathbf{z}_i^{ori}, \mathbf{z}_k^{aug}\right)/\tau\right.\right. \tag{9}$$

where the first term aggregates similarity contributions from original-view negative samples to the anchor node, while the second term computes augmented-view negative sample contributions.

2.4. VF Prediction

The protein embeddings optimized through self-supervised modules achieve an integration of protein sequences, physicochemical properties, as well as contextual information derived from the PPI network. These embeddings are subsequently processed through a classifier to predict VFs. The classifier architecture incorporates a FC layer and a Dropout layer, the latter of which effectively mitigates overfitting. The final output is transformed into a probability value via the sigmoid activation function. The classifier can be mathematically expressed as

$$\hat{y}_i = \varphi(\mathbf{W} \cdot \mathrm{Dropout}(\mathbf{h}_i^{ori}) + b) \tag{10}$$

where $\mathbf{h}_i^{ori}$ denotes the protein embedding of node i learned by the GNN, $\mathbf{W}$ represents the FC layer weight matrix, b is the bias term, and φ represents the sigmoid activation function.

To address class imbalance in VF prediction, we implement Focal loss [53] for classification loss. The Focal loss dynamically adjusts sample weights, diminishing easily-classified sample contributions while amplifying difficult-to-classify sample importance, thereby enhancing minority class recognition capabilities. The focal loss calculation introduces an intermediate variable pt_i, which is a combination of the true label and the predicted probability, and its specific definition is as follows

$$pt_i = y_i\hat{y}_i + (1 - y_i)(1 - \hat{y}_i) \tag{11}$$

where $\hat{y}_i$ represents the predicted probability, and y_i is the true label. The complete form of the focal loss can be formulated as

$$\mathcal{L}_{cls} = -\frac{1}{N}\sum_{i=1}^{N}[\lambda(1 - pt_i)^{\gamma}\log(pt_i)] \tag{12}$$

where λ represents the class weight coefficient balancing positive and negative sample contributions, and γ denotes the focusing parameter controlling attention distribution between easy and difficult samples.

The final loss function integrates three essential components: reconstruction loss $\mathcal{L}_{gen}$, contrastive loss $\mathcal{L}_{con}$, and classification loss $\mathcal{L}_{cls}$. Self-supervised learning components are balanced by coefficients α and β

$$\mathcal{L} = \alpha\mathcal{L}_{gen} + \beta\mathcal{L}_{con} + \mathcal{L}_{cls} \tag{13}$$

The overall workflow of the proposed GC-VF framework is shown in Algorithm 1. Firstly, we sample a batch of protein nodes from the PPI network. For each target node, we generate its original view and augmented view, then input these two views into the shared GNN encoder to extract the embeddings of the target node. After that, we decode and restore the target node embeddings in the original view and calculate the reconstruction loss. Next, we input the target node embeddings from the two views into the projection head and project them onto a new embedding space. Subsequently, we adopt a mixed sampling strategy between and within views for node-level contrastive learning and calculate the contrastive loss. Finally, we combine the generation, contrastive, and classification tasks, train the model through joint optimization of multiple objectives, and perform predictions on proteins.

3. Results

3.1. Experimental Settings

In experimental implementation, a two-layer GraphSAGE is employed as the GNN backbone, utilizing LSTM as the aggregation function. Given that the protein sequence encoding module generates embeddings of 256 dimensions, we configure the GNN architecture with hidden dimensions of 128 and 64 for the first and second layers, respectively. The noise hyperparameter is configured at 0.1, and the InfoNCE loss temperature hyperparameter is set to 0.03. For the composite loss function, we assign a coefficient of 0.4 to the generative loss. As for the contrastive loss, its coefficient varies depending on the dataset, being set to either 0.6 or 0.2. The focal loss hyperparameters λ and γ are set to 0.25 and 2. The subgraph sampling is established at a size of (6, 6), resulting in a subgraph with the size of 37 nodes for each target node.

For performance evaluation, we employ two primary metrics: the Area Under the Receiver Operating Characteristic Curve (AUROC) and the Area Under the Precision-Recall Curve (AUPRC). The AUROC provides a comprehensive assessment of the model's discriminative capability between positive (majority) and negative (minority) samples, while the AUPRC specifically emphasizes performance on positive samples. To ensure a thorough evaluation, we incorporate additional supplementary metrics: sensitivity, specificity, F1-score, and Matthews Correlation Coefficient (MCC). Sensitivity is specifically utilized to evaluate positive sample identification capability, while specificity measures its discriminative ability for negative samples. The F1-score adeptly balances the accuracy and recall evaluations of positive class prediction, and the MCC comprehensively contemplates various prediction outcomes, rendering them particularly appropriate for the evaluation of imbalanced datasets.

Our model is trained using the Adam optimizer, with a learning rate of 0.001 and a weight decay coefficient of 0.0001. The datasets are partitioned into training, validation, and test sets in a 6:2:2 ratio. Throughout the training process, a total of 131 epochs are carried out, and the model's performance is scrutinized on the validation set every 10 epochs. The optimal model is then selected and archived. During the testing phase, the model is subjected to 100 trials on the test set, and the average of the results is adopted as the final performance indicator. To expedite and optimize the training process, the batch size is set to 300. The GC-VF model is executed on a NVIDIA GeForce RTX 3080 GPU.

Algorithm 1. The key algorithm for the proposed GC-VF framework

```
Input: PPI graph $G = (V, E)$, Initial protein embeddings $\mathbf{x}_i$, Maximum
number of training epochs $T$, Batch size $B$
Output: VF prediction probability $\hat{y}_i$
1:  for each training epoch $t \in \{1,2,\dots,T\}$ do:
2:    Randomly divide the protein nodes $V$ into batches of size $B$.
3:    for each batch $b = \{v_1, v_2, \dots, v_B\}$ do:
4:      for each node $v_i$ in $b$ do:
5:        Randomly sample a second-order subgraph of $v_i$ as $S_i^{ori}$, and
          generate $S_i^{aug}$ by adding Gaussian noise to the features.
6:        Compute the protein embeddings $\mathbf{h}_i^{ori}$ and $\mathbf{h}_i^{ori}$ from the
          GNN encoder using the embeddings of both views, $\mathbf{x}_i^{ori}$ and
          $\mathbf{x}_i^{aug}$, via Eq. (3).
7:        Perform attribute reconstruction on $\mathbf{h}_i^{ori}$ via Eq. (4) to obtain
          the reconstructed protein attribute $\hat{\mathbf{x}}_i^{ori}$.
8:        Calculate the reconstruction loss $\mathcal{L}_{gen}$ using Eq. (5) between
          $\mathbf{x}_i^{ori}$ and $\hat{\mathbf{x}}_i^{ori}$.
9:        Project the GNN-encoded target node embeddings from the
          $G_i^{ori}$ and $G_i^{aug}$ by the projection head to obtain latent
          embeddings via Eq. (6) and Eq. (7), respectively.
10:       Perform contrastive learning using $\mathbf{z}_i^{ori}$ and $\mathbf{z}_i^{aug}$ to
          compute contrastive loss $\mathcal{L}_{con}$ via Eq. (8).
11:       Calculate the classification loss $\mathcal{L}_{cls}$ for VF prediction via
          Eq. (12).
12:       Update the model parameters by backpropagating the total loss
          $\mathcal{L}$ via Eq. (13).
13:     end for
14:   end for
15: end for
16: Predict the virulence factor probability $\hat{y}_i$ for node $v_i$ via Eq. (11).
```

3.2. Methods Comparison

To comprehensively evaluate method performance in VF prediction, we conduct experiments on three representative bacterial strain datasets: *S. enterica* serovar Typhimurium LT2, *C. jejuni* NCTC 11168, and *S. aureus* NCTC 8325. These datasets exhibit VF ratios of 3.5%, 8.0%, and 3.1% respectively, reflecting the characteristic class imbalance encountered in real-world applications.

For comparative analysis, we select four representative baseline methods: BLAST [5], which is predicated on sequence similarity; the classical machine learning approach, VirulentPred 2.0 [54]; as well as DeepVF [8], a hybrid method combining traditional machine learning and deep learning; and DTVF [12], which is based on deep learning. The BLAST implementation involves constructing a local sequence database, wherein test sequences are queried and predictions are made based on the highest-similarity matches within training samples. Considering the evaluations of VirulentPred 2.0 and DeepVF, the default parameter configurations provided by their respective web platforms are adopted. When implementing DTVF, ProtT5 is initially utilized to generate representations of protein sequences, followed by the employment of its pre-trained method for prediction.

In performance evaluation, we employ multiple complementary metrics. The AUROC baseline of 0.5 represents random classifier performance, while AUPRC baselines vary with positive sample proportions, corresponding to 0.035, 0.08, and 0.031 for the three datasets respectively. The experimental results presented in Table 2 demonstrate that our proposed GC-VF method exhibits notable overall performance.

Analysis of experimental results reveals that while certain baseline methods have achieved high accuracy on specific datasets, such apparent advantages may mask underlying classification limitations. These methods often exhibit high specificity, showing a tendency to classify samples into the majority negative class. In scenarios with substantial class imbalance, while this tendency might yield favorable accuracy and specificity metrics, the actual discriminative capability remains limited. Thus, comprehensive evaluation necessitates consideration of multiple metrics, particularly sensitivity and AUPRC.

Notably, we observe that some methods often struggle to maintain adequate specificity while attempting to enhance sensitivity. For instance, VirulentPred 2.0, despite demonstrating high sensitivity, achieves a low F1-score, indicating that increased recall comes at the cost of precision, resulting in numerous false positive predictions. This observation underscores the importance of balancing various performance metrics in VF prediction and highlights the GC-VF framework's capability in effectively harmonizing these competing objectives.

Table 2. Performance comparison of different methods for VF prediction.

Dataset	Method	Accuracy	Sensitivity	Specificity	F1-score	MCC	AUPRC	AUROC
S. enterica serovar Typhimurium LT2	BLAST	0.9560	0.4839	0.9731	0.4348	0.4145	0.3440	
	VirulentPred 2.0	0.6979	**0.9677**	0.6881	0.1829	0.2552	0.0989	
	DeepVF	0.9008	0.3548	0.9208	0.2018	0.1788	0.2593	0.7942
	DT-VF	0.9651	0.3548	0.9871	0.4151	0.4038	0.2761	0.8678
	GC-VF	**0.9941**	0.9058	**0.9973**	**0.9140**	**0.9119**	**0.9572**	**0.9972**
C. jejuni NCTC 11168	BLAST	0.8738	0.3462	0.9197	0.3051	0.2388	0.3356	0.8738
	VirulentPred 2.0	0.4831	**0.9231**	0.4448	0.2222	0.2025	0.1228	
	DeepVF	0.6440	0.7308	0.6364	0.2484	0.2045	0.4510	0.6695
	DT-VF	0.6800	0.8077	0.6689	0.2877	0.2679	0.1817	0.7622
	GC-VF	**0.9588**	0.6877	**0.9824**	**0.7276**	**0.7107**	**0.7365**	**0.9482**
S. aureus NCTC 8325	BLAST	0.9579	0.3529	0.9765	0.3333	0.3122	0.3440	
	VirulentPred 2.0	0.4614	**1.0000**	0.4448	0.0997	0.1528	0.0525	
	DeepVF	0.5554	0.8750	0.5456	0.1041	0.1427	0.4670	0.7021
	DT-VF	0.9035	0.4706	0.9168	0.2254	0.2250	0.1678	0.8171
	GC-VF	**0.9889**	0.7524	**0.9961**	**0.8005**	**0.7990**	**0.8107**	**0.9460**

3.3. Analysis of Graph-Based Approaches

We conduct comprehensive evaluation of various graph methods on three bacterial strain datasets, categorized into two primary classes: graph learning and graph construction. To ensure experimental validity, consistent datasets and hyperparameter settings are maintained across all methods.

Among the graph learning methods, we primarily evaluate three prominent GNN architectures: GraphSAGE, GCN, and GAT. GraphSAGE aims to train aggregator functions and has multiple aggregator architectures, including the Mean aggregator, LSTM aggregator, and Pooling aggregator. Moreover, GraphSAGE-GCN, a convolutional variant of GraphSAGE, is an extended inductive version of GCN.

In terms of graph construction methods, we explore several graph construction strategies: the PPI networks, the Cosine Similarity Graph (CSG), and the BLAST Similarity Graph (BSG). The CSG model computes the cosine similarity between protein feature vectors and constructs the graph using a threshold of 0.5. The BSG model, on the other hand, calculates protein sequence similarities via BLAST and retains edges with an E-value below 20. In addition, we set a baseline method without using a graph structure (MLP-NS), which directly inputs the embeddings encoded from protein sequences into an MLP classifier. Notably, both the CSG and BSG models utilize GraphSAGE-LSTM for graph learning.

Figures 4(a) and 4(b) present comparative AUROC and AUPRC metrics across the three bacterial strain datasets. In the experiments PPI of graph-based GNN variant, GraphSAGE models consistently outperform GCN and GAT, with superior performance from LSTM and Pooling aggregators. The LSTM aggregator achieves optimal performance, primarily due to its distinctive sequence processing mechanism. By establishing continuous dependency chains between nodes, it effectively captures complex neighbor relationships which might enhance the GNNs' expressive capacity in neighbor feature aggregation, enabling more comprehensive node-dependency modeling. Meanwhile, the Pooling aggregator, by adaptively identifying key features from neighbors, focuses attention on important information for VF prediction.

Experimental results reveal significant performance degradation when excluding PPI network structure. While both CSG and BSG approaches based on sequence similarity capture certain protein relationships, they inadequately express complex biological interactions. This characteristic underscores the PPI network's advantage in capturing protein functional relationships: sequence similarity alone does not equate to functional similarity. For instance, proteins with similar sequences may not participate in identical biological pathways, whereas PPI graphs better reflect proteins' synergistic effects through real biological interactions. Furthermore, even advanced sequence-based models (e.g., CNN, GRU) underperform graph-based methods. This suggests that graph structures have more potential to capture protein feature relationships, compensating for the limitations of sequence-only models.

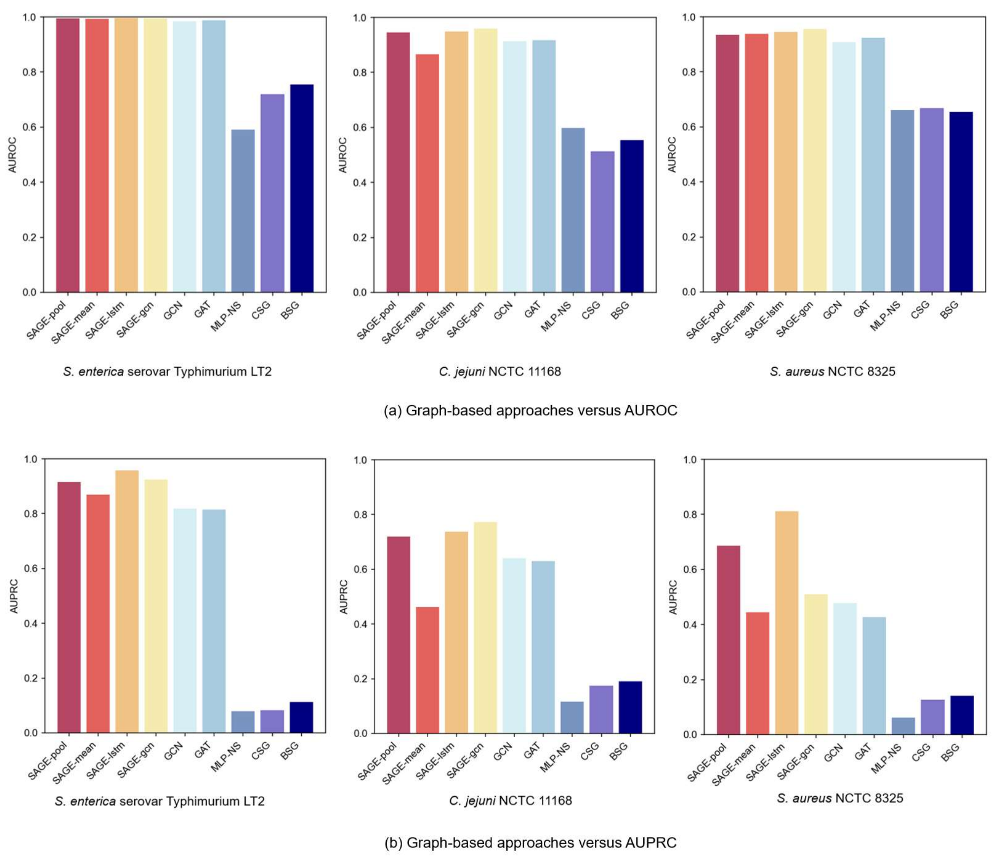


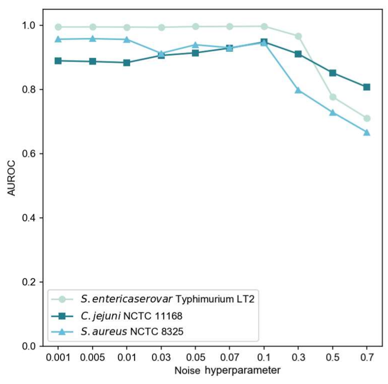
Figure 4. Performance comparison of graph learning and construction approaches.

3.4. Analysis of Graph-Based Approaches

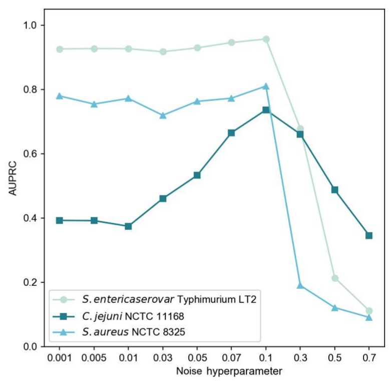
We conduct comprehensive experiments to evaluate how various key hyperparameters influence our proposed framework's performance. The analysis focuses on four critical hyperparameters: noise hyperparameter, temperature hyperparameter, subgraph sampling size, and loss function balance factors. Below we present our detailed findings.

In our contrastive learning setup, we generate an augmented view by applying Gaussian noise to the original node features. The noise, following a standard normal distribution, is controlled by a hyperparameter that determines the perturbation magnitude. We test noise hyperparameter values throughout the set {0.001, 0.005, 0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0.5, 0.7} and evaluate performances using AUROC and AUPRC, as illustrated in Figures 5(a) and 5(b). Our findings reveal that moderate noise levels improve model performance, though the effect varies from dataset to dataset. Excessive noise (above 0.1) significantly degrades performance by disrupting feature integrity. Through extensive testing, we determine that a noise hyperparameter of 0.1 provides an optimal balance between data diversity and stability across datasets.

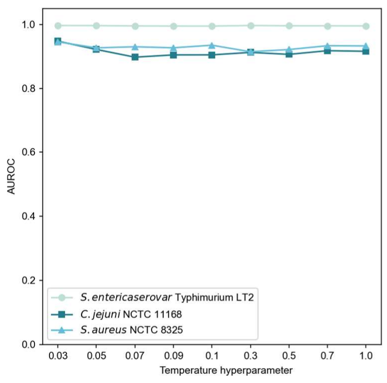
The temperature hyperparameter τ in the InfoNCE loss governs the model's ability to distinguish between positive and negative samples. Our experiments, spanning τ values within the set {0.03, 0.05, 0.07, 0.09, 0.1, 0.3, 0.5, 0.7, 1.0}, reveal dataset-specific sensitivities to this hyperparameter (Figures 5(c) and 5(d)). When setting $\tau = 0.01$, an abnormal situation occurs where the model outputs NaN (Not a Number) values. This indicates that the selected value for this parameter is rather diminutive, failing to support stable operations of the model. Consequently, in subsequent experiments regarding the setting of this hyperparameter, we initiate the range from 0.03 to avoid such instabilities and further explore the optimal configuration. Based on extensive testing, we finally select $\tau = 0.03$ as the optimal value, delivering consistent and superior performance across most datasets.



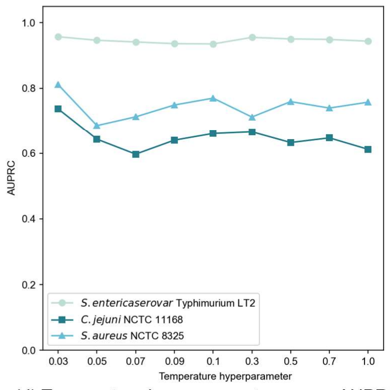
(a) Noise hyperparameter versus AUROC



(b) Noise hyperparameter versus AUPRC



(c) Temperature hyperparameter versus AUROC



(d) Temperature hyperparameter versus AUPRC

Figure 5. Sensitivity analysis of noise and temperature hyperparameters on GC-VF performance.

We implement a two-layer subgraph sampling strategy, denoted as (K_1, K_2), where $K_1, K_2 \in \{2, 3, 4, 5, 6, 7\}$ and $K_2 \leq K_1$. Results shown in Figure 6 indicate that subgraph size particularly affects AUPRC, while AUROC remains more stable. Larger first-layer sampling (K_1) generally enhances performance, suggesting better capture of local information characteristics. Similarly, increased second-layer sampling (K_2) improves AUPRC, highlighting the importance of second-order neighborhood information in modeling contextual relationships. Balancing performance and computational efficiency, we establish (K_1, K_2) = (6,6) as optimal sampling hyperparameters.

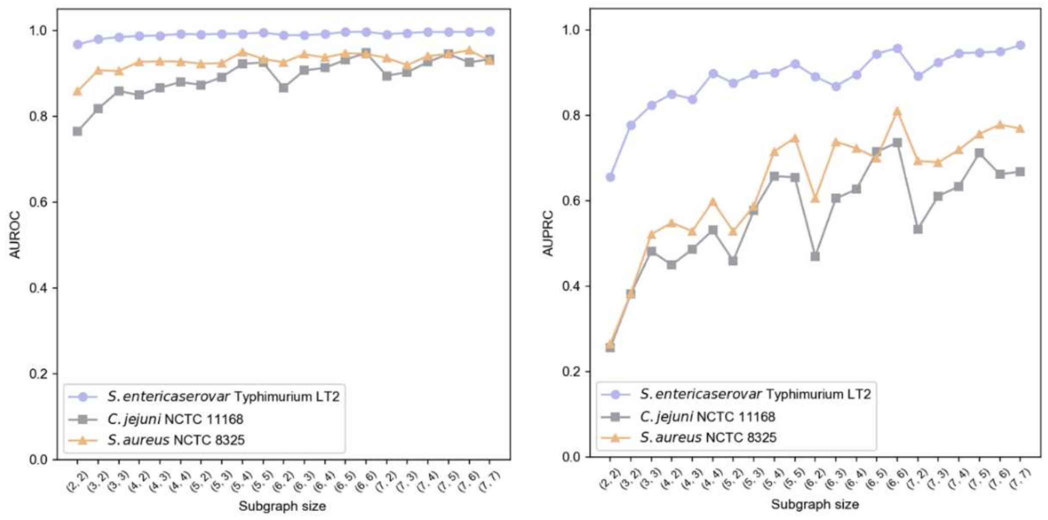
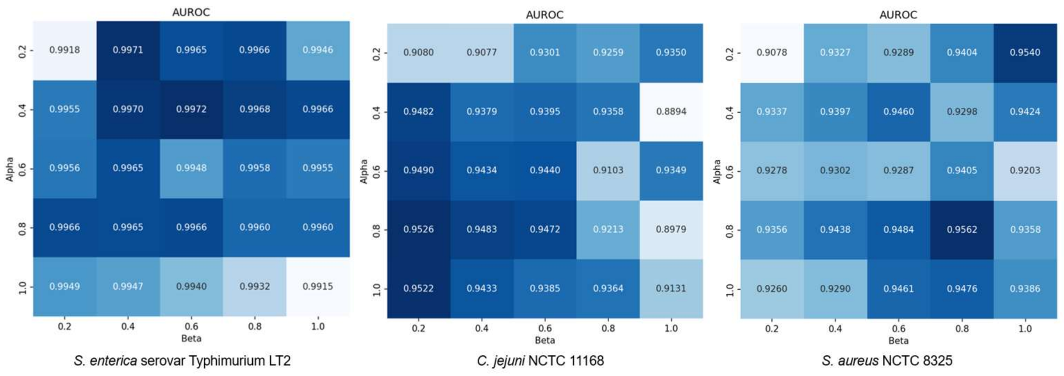
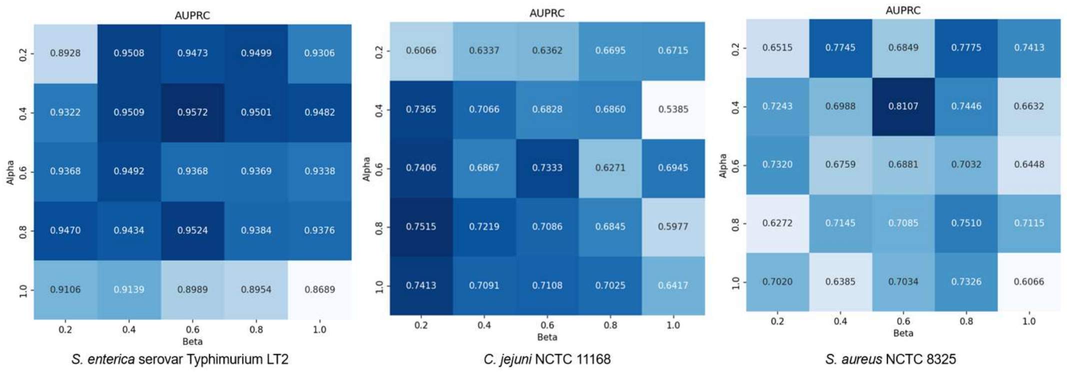


Figure 6. Impact of subgraph sampling size on GC-VF performance.

Finally, we explore how the coefficients α and β in loss function (13) affect model performance by modulating the influence of generative and contrastive self-supervised modules. Given that self-supervised learning serves an auxiliary role to classification, we constrain both α and β to (0, 1]. We set the coefficients in a range of {0.2, 0.4, 0.6, 0.8, 1.0} and conducted a comprehensive exploration of different combinations within this range. Results in Figure 7 show that small coefficients ($\alpha = 0.2$, $\beta = 0.2$) lead to suboptimal performance due to insufficient feature learning, while large values ($\alpha = 1.0$, $\beta = 1.0$) compromise classification capability by overemphasizing self-supervised tasks. The *C. jejuni* NCTC 11168 dataset shows particular sensitivity to the generative loss coefficient α, performing best with high α and low β values ($\beta = 0.2$). Conversely, the *S. enterica* serovar Typhimurium LT2 dataset exhibits performance degradation at $\alpha = 1.0$, suggesting dataset-specific dependencies on different self-supervised learning modules. Based on these findings, we standardize α at 0.4 while allowing β to vary by dataset: $\beta = 0.2$ for *S. aureus* NCTC 8325 and $\beta = 0.6$ for the other datasets, achieving optimal cross-dataset performance.



(a) Loss coefficients versus AUROC



(b) Loss coefficients versus AUPRC

Figure 7. GC-VF performance variation with different α and β combinations.

3.5. Ablation Study

To systematically evaluate the GC-VF framework's key components, we conduct a series of ablation experiments. We test several variant models by removing specific modules: the generative attribute reconstruction module (GC-VF w/o Gen), the local contrastive learning module (GC-VF w/o Con), and the self-supervised learning module (GC-VF w/o SL). Additionally, we evaluate variants of the GraphSAGE encoder without edge weights (GC-VF w/o EW) and with binary cross-entropy loss replacing focal loss (GC-VF w/ BCE). Table 3 presents the comparative results across all three datasets.

Our experiments reveal that the complete GC-VF model, incorporating both generative and contrastive learning for self-supervision, achieves notable performance across all datasets. The ablation studies demonstrate that removing any self-supervised module results in performance degradation. Notably, the combined implementation of both modules produces performance improvements exceeding their individual contributions, confirming positive and effective synergy between these self-supervised strategies. However, we observe distinct variations in module contributions across different datasets. For the *S. enterica* serovar Typhimurium LT2 and *C. jejuni* NCTC 11168 datasets, the removal of the generative module results in minimal impact, suggesting contrastive learning's dominance in feature extraction. Conversely, for the *S. aureus* NCTC 8325 dataset, the attempt to remove the generative module significantly degrades performance across multiple metrics, underlining its crucial role in feature learning. These variations suggest that bacterial strain datasets possess unique feature distribution patterns, leading to differential responses to various self-supervised learning strategies.

The analysis of edge weights reveals their importance in model performance. Removing edge weights substantially degrades results, primarily because traditional feature aggregation methods fail to capture the nuanced intensity differences in PPI networks. Our statistical analysis of PPIs across the three datasets reveals distinctive patterns in interaction strengths: virulence protein interactions average a combined score of 0.7556, virulence to non-virulence protein interactions average 0.6037, and non-virulence protein interactions 0.6428. These findings indicate stronger interaction patterns between proteins of the same type, among which virulence proteins exhibit the strongest interaction patterns. This biological insight justifies our integration of edge weights into the framework. By incorporating these differentiated edge weights into the GNNs, our model effectively prioritizes strong interaction connections, particularly the high-intensity interactions between virulence proteins, during feature aggregation. This approach not only enhances the model's prediction capabilities but also aligns with established biological characteristics of PPI networks.

Furthermore, the implementation of focal loss has proved effective in addressing class imbalance, even with default hyperparameter settings. The comparative analysis shows that replacing focal loss with binary cross-entropy loss results in performance degradation, highlighting focal loss's beneficial role in managing data imbalance challenges.

Table 3. Ablation experiment results of GC-VF key modules across three datasets.

Dataset	Model	Accuracy	Sensitivity	Specificity	F1-score	MCC	AUPRC	AUROC
S. enterica serovar Typhimurium LT2	GC-VF w/o Con	0.9906	0.8458	0.9959	0.8625	0.8596	0.9106	0.9929
	GC-VF w/o Gen	0.9922	0.8632	0.9969	0.8855	0.8831	0.9151	0.9889
	GC-VF w/o SL	0.9894	0.8055	0.9961	0.8414	0.8380	0.8754	0.9813
	GC-VF w/o EW	0.9873	0.8084	0.9938	0.8168	0.8132	0.8689	0.9915
	GC-VF w/ BCE	0.9775	0.7100	0.9872	0.6886	0.6823	0.7333	0.9824
	GC-VF	**0.9941**	**0.9058**	**0.9973**	**0.9140**	**0.9119**	**0.9572**	**0.9972**
C. jejuni NCTC 11168	GC-VF w/o Con	0.8803	0.6046	0.9042	0.4482	0.4087	0.3591	0.8688
	GC-VF w/o Gen	0.9447	0.5323	0.9806	0.6070	0.5928	0.5921	0.8748
	GC-VF w/o SL	0.9287	0.4519	0.9701	0.5065	0.4937	0.4833	0.8323
	GC-VF w/o EW	0.9325	**0.7012**	0.9526	0.6249	0.6000	0.6013	0.9365
	GC-VF w/ BCE	0.9268	0.4646	0.9670	0.5061	0.4873	**0.8428**	0.4646
	GC-VF	**0.9588**	0.6877	**0.9824**	**0.7276**	**0.7107**	0.7365	**0.9482**
S. aureus NCTC 8325	GC-VF w/o Con	0.9856	0.6612	0.9955	0.7327	0.7401	0.7457	**0.9535**
	GC-VF w/o Gen	0.9806	0.7382	0.9880	0.6942	0.6877	0.6613	0.9296
	GC-VF w/o SL	0.9827	0.6835	0.9919	0.7035	0.7038	0.7301	0.9379
	GC-VF w/o EW	0.9841	0.6759	0.9936	0.7172	0.7141	0.6978	0.9342
	GC-VF w/ BCE	0.9867	0.7124	0.9951	0.7610	0.7585	0.7538	0.9013
	GC-VF	**0.9889**	**0.7524**	**0.9961**	**0.8005**	**0.7990**	**0.8107**	0.9460

4. Discussion

In this paper, we propose GC-VF, a novel VF identification framework based on PPI networks. GC-VF innovatively combines generative and contrastive self-supervised learning to enhance imbalanced node classification. The generative module generates robust node representations via feature reconstruction, preserving essential protein characteristics, and the contrasting module enhances feature discrimination by maximizing/minimizing similarities of positive/negative instances, boosting the model's protein classification ability. Unlike methods relying solely on sequence features, GC-VF exploits systemic PPI networks information, aligning with the biological mechanism that pathogenesis arises from coordinated protein interactions. Experiments on three bacterial-strain datasets validate its effectiveness in VF identification, offering insights for bacterial pathogenesis research and targeted therapeutic development.

However, PPI network-based approaches encounter two critical challenges: (1) strong coupling between protein features and network context, leading to transfer-learning difficulties across unseen networks; (2) hyperparameter optimization challenges arising from different network structures and data distributions, necessitating automated adaptive tuning to improve model efficiency and adaptability.

In response to the aforementioned limitations, we propose several directions for future research. First, we aim to explore transfer-learning techniques, such as Domain-Adversarial Neural Network (DANN)[55] and Conditional Domain-Adversarial Neural Network (CDAN)[56] on graphs, which leverage adversarial training to align domains by learning shared feature representations between source and target networks, thereby improving model generalization on new PPI networks. Second, to address hyperparameter selection, we propose employing Bayesian optimization and random search for automatic exploration of the hyperparameter space. Additionally, we plan to leverage hyperparameters tuned on similar datasets to quickly fine-tune the model for different PPI network datasets.

Author Contributions: Conceptualization, Y.Y. and H.C.; methodology, J.W. and Y.W.; software, Y.Y.; validation, H.C., J.W. and Y.W.; formal analysis, H.C.; investigation, Y.W.; resources, J.W. and Y.W.; data curation, Y.Y.; writing—original draft preparation, Y.Y. and H.C.; writing—review and editing, J.W. and Y.W.; visualization, H.C.; supervision, J.W. and Y.W.; project administration, J.W.; funding acquisition, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, Grant Number: 32302241; National Key Research and Development Program of China, grant number: 2024YFF1106705.

Data Availability Statement: The source code of this work can be found at https://github.com/Aysiling/GC-VF (available from 24 February, 2025). The virulence factor data for *S. enterica* serovar Typhimurium LT2 were obtained from the VFDB at https://www.mgc.ac.cn/cgi-bin/VFs/compvfs.cgi?Genus=Salmonella (accessed on 24 October 2024). The virulence factor data for *C. jejuni* NCTC 11168 were obtained from VFDB at https://www.mgc.ac.cn/cgi-bin/VFs/compvfs.cgi?Genus=Campylobacter (accessed on 24 October 2024). The virulence factor data for *S. aureus* NCTC 8325 were obtained from VFDB at https://www.mgc.ac.cn/cgi-bin/VFs/compvfs.cgi?Genus=Staphylococcus (accessed on 24 October 2024). The protein sequence data for *S. enterica* serovar Typhimurium LT2 were downloaded from the STRING database at https://stringdb-downloads.org/download/protein.sequences.v12.0/99287.protein.sequences.v12.0.fa.gz (accessed on 24 October 2024). The protein sequence data for *C. jejuni* NCTC 11168 were obtained from https://stringdb-downloads.org/download/protein.sequences.v12.0/192222.protein.sequences.v12.0.fa.gz (accessed on 24 October 2024). The protein sequence data for *S. aureus* NCTC 8325 were obtained from https://stringdb-downloads.org/download/protein.sequences.v12.0/93061.protein.sequences.v12.0.fa.gz (accessed on 24 October 2024). The PPI data for *S. enterica* LT2 were retrieved from https://stringdb-downloads.org/download/protein.links.v12.0/99287.protein.links.v12.0.txt.gz (accessed on 24 October 2024). The PPI data for *C. jejuni* NCTC 11168 were obtained from https://stringdb-downloads.org/download/protein.links.v12.0/192222.protein.links.v12.0.txt.gz (accessed on 24 October 2024). The PPI data for *S. aureus* NCTC 8325 were obtained from https://stringdb-downloads.org/download/protein.links.v12.0/93061.protein.links.v12.0.txt.gz (accessed on 24 October 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Muteeb, G.; Rehman, M. T.; Shahwan, M.; Aatif, M. Origin of antibiotics and antibiotic resistance, and their impacts on drug development: A narrative review. *Pharmaceuticals* **2023**, 16 (11), 1615.
2. Dance, A. Five ways science is tackling the antibiotic resistance crisis. *Nature* **2024**, 632 (8025), 494-496.
3. Dehbanipour, R.; Ghalavand, Z. Anti-virulence therapeutic strategies against bacterial infections: recent advances. *Germs* **2022**, 12 (2), 262.
4. Gupta, A.; Kapil, R.; Dhakan, D. B.; Sharma, V. K. MP3: a software tool for the prediction of pathogenic proteins in genomic and metagenomic data. *PloS One* **2014**, 9 (4), e93907.
5. Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **1990**, 215 (3), 403-410.
6. Garg, A.; Gupta, D. VirulentPred: a SVM based prediction method for virulent proteins in bacterial pathogens. *BMC Bioinformatics* **2008**, 9, 1-12.
7. Rentzsch, R.; Deneke, C.; Nitsche, A.; Renard, B. Y. Predicting bacterial virulence factors–evaluation of machine learning and negative data strategies. *Brief. Bioinform.* **2020**, 21 (5), 1596-1608.
8. Xie, R.; Li, J.; Wang, J.; Dai, W.; Leier, A.; Marquez-Lago, T. T.; Akutsu, T.; Lithgow, T.; Song, J.; et al. DeepVF: a deep learning-based hybrid framework for identifying virulence factors using the stacking strategy. *Brief. Bioinform.* **2021**, 22 (3), bbaa125.
9. Singh, S.; Le, N. Q. K.; Wang, C. VF-Pred: Predicting virulence factor using sequence alignment percentage and ensemble learning models. *Comput. Biol. Med.* **2024**, 168, 107662.
10. Ofer, D.; Brandes, N.; Linial, M. The language of proteins: NLP, machine learning & protein sequences. *Comput. Struct. Biotechnol. J.* **2021**, 19, 1750-1758.
11. Ferruz, N.; Höcker, B. Controllable protein design with language models. *Nat. Mach. Intell.* **2022**, 4 (6), 521-532.
12. Sun, J.; Yin, H.; Ju, C.; Wang, Y.; Yang, Z. DTVF: A User-Friendly Tool for Virulence Factor Prediction Based on ProtT5 and Deep Transfer Learning Models. *Genes* **2024**, 15 (9), 1170.
13. Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rehawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M.; et al. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 44 (10), 7112-7127.
14. Li, G.; Bai, P.; Chen, J.; Liang, C. Identifying virulence factors using graph transformer autoencoder with ESMFold-predicted structures. *Comput. Biol. Med.* **2024**, 170, 108062.
15. Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **2023**, 379 (6637), 1123-1130.
16. Wan, X. F.; Xu, D. Computational methods for remote homolog identification. *Curr. Prot. Pept. Sci.* **2005**, 6 (6), 527-546.
17. Cui, W.; Chen, L.; Huang, T.; Gao, Q.; Jiang, M.; Zhang, N.; Zheng, L.; Feng, K.; Cai, Y.; Wang, H. Computationally identifying virulence factors based on KEGG pathways. *Mol. Biosyst* **2013**, 9 (6), 1447-1452.
18. Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; 1263-1272.
19. Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017; 2713-2726.
20. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, Canada, 30 April–3 May 2018; 2920-2931.
21. Hamilton, W. L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; 1025-1035.

22. McPherson, M.; Smith-Lovin, L.; Cook, J. M. Birds of a Feather: Homophily in Social Networks. *Annu. Rev. Sociol.* **2001**, 27 (1), 415-444.
23. Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Steeg, G. V.; Galstyan, A. Mixhop: Higher-order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; 21-29.
24. Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; Koutra, D. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In Proceedings of the 34th Conference on Neural Information Processing Systems, Vancouver, Canada, 6–12 December 2020; 7793-7804.
25. Pei, H.; Wei, B.; Chang, K. C. C.; Yang, B.; Lei, Y. GEOM-GCN: GEOMETRIC GRAPH CONVOLUTIONAL NETWORKS. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; 10247-10258.
26. Hamilton, W. L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; 1025-1035.
27. Szklarczyk, D.; Kirsch, R.; Koutrouli, M.; Nastou, K.; Mehryary, F.; Hachilif, R.; Gable, A. L.; Fang, T.; Doncheva, N. T.; Pyysalo, S.; et al. The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Res.* **2023**, 51 (D1), D638-D646.
28. Ericsson, L.; Gouk, H.; Loy, C. C.; Hospedales, T. M. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Process. Mag.* **2022**, 39 (3), 42-62.
29. Yang, Y.; Xu, Z. Rethinking the Value of Labels for Improving Class-Imbalanced Learning. In Proceedings of the 34th Conference on Neural Information Processing Systems, Vancouver, Canada, 6–12 December 2020; 19290-19301.
30. Liu, H.; HaoChen, J. Z.; Gaidon, A.; Ma, T. Self-supervised Learning is More Robust to Dataset Imbalance. In Proceedings of the 10th International Conference on Learning Representations, Virtual Conference, 25–29 April 2022.
31. Wu, L.; Lin, H.; Tan, C.; Gao, Z.; Li, S. Z. Self-Supervised Learning on Graphs: Contrastive, Generative, or Predictive. *IEEE Trans. Knowl. Data Eng.* **2023**, 35 (4), 4216-4226.
32. You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph Contrastive Learning with Augmentations. In Proceedings of the 34th Conference on Neural Information Processing Systems, Vancouver, Canada, 6–12 December 2020; 5812-5823.
33. Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; Hjelm, R. D. Deep Graph Infomax. In Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
34. Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; Tang, J. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Conference, 23–27 August 2020; 1150-1160.
35. You, Y.; Chen, T.; Wang, Z.; Shen, Y. When Does Self-supervision Help Graph Convolutional Networks? In Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, 13–18 July 2020; 10871-10880.
36. Hu, Z.; Dong, Y.; Wang, K.; Chang, K. W.; Sun, Y. GPT-GNN: Generative Pre-Training of Graph Neural Networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Conference, 23–27 August 2020; 1857-1867.
37. Sun, K.; Lin, Z.; Zhu, Z. Multi-Stage Self-Supervised Learning for Graph Convolutional Networks on Graphs with Few Labeled Nodes. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; 5892-5899.
38. Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; Huang, J. Self-Supervised Graph Transformer on Large-Scale Molecular Data. In Proceedings of the 34th Conference on Neural Information Processing Systems, Vancouver, Canada, 6–12 December 2020; 12559-12571.

39. Juan, X.; Zhou, F.; Wang, W.; Jin, W.; Tang, J.; Wang, X. INS-GNN: Improving Graph Imbalance Learning with Self-supervision. *Inf. Sci.* **2023**, 637, 118935.
40. Wu, L.; Xia, J.; Gao, Z.; Lin, H.; Tan, C.; Li, S. Z. GraphMixup: Improving Class-Imbalanced Node Classification by Reinforcement Mixup and Self-supervised Context Prediction. In Proceedings of the 2022 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Grenoble, France, 19–23 September 2022; 519-535.
41. Liu, B.; Zheng, D.; Zhou, S.; Chen, L.; Yang, J. VFDB 2022: A General Classification Scheme for Bacterial Virulence Factors. *Nucleic Acids Res.* **2022**, 50 (D1), D912-D917.
42. Zhang, Y.; Kang, B.; Hooi, B.; Yan, S.; Feng, J. Deep Long-Tailed Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, 45 (9), 10795-10816.
43. Wang, Y.; You, Z. H.; Yang, S.; Li, X.; Jiang, T. H.; Zhou, X. A High Efficient Biological Language Model for Predicting Protein-Protein Interactions. *Cells* **2019**, 8 (2), 122.
44. Slam, S. A.; Heil, B. J.; Kearney, C. M.; Baker, E. J. Protein Classification Using Modified N-Grams and Skip-Grams. *Bioinformatics* **2018**, 34 (9), 1481-1487.
45. Chen, M.; Ju, C. J. T.; Zhou, G.; Chen, X.; Zhang, T.; Chang, K. W.; Zaniolo, C.; Wang, W. Multifaceted Protein-Protein Interaction Prediction Based on Siamese Residual RCNN. *Bioinformatics* **2019**, 35 (14), i305-i314.
46. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 27th Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; 3111-3119.
47. Shen, J.; Zhang, J.; Luo, X.; Zhu, W.; Yu, K.; Chen, K.; Li, Y.; Jiang, H. Predicting Protein–Protein Interactions Based Only on Sequences Information. *Proceedings of the National Academy of Sciences* **2007**, 104 (11), 4337-4341.
48. Lv, G.; Hu, Z.; Bi, Y.; Zhang, S. Learning Unknown from Correlations: Graph Neural Network for Inter-novel-protein Interaction Prediction. In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), Montreal, Canada, 19–26 August 2021; 3677-3683.
49. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? In Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019; 9104-9120.
50. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; 315-323.
51. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the 37th International Conference on Machine Learning, Virtual Conference, 13–18 July 2020; 1597-1607.
52. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Deep Graph Contrastive Representation Learning. In Proceedings of the 37th International Conference on Machine Learning Workshop on Graph Representation Learning and Beyond, Virtual Conference, 13–18 July 2020.
53. Lin, T. Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; 2980-2988.
54. Sharma, A.; Garg, A.; Ramana, J.; Gupta, D. VirulentPred 2.0: An Improved Method for Prediction of Virulent Proteins in Bacterial Pathogens. *Protein Sci.* **2023**, 32 (12), e4808.
55. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **2016**, 17 (59), 1-35.
56. Long, M.; Cao, Z.; Wang, J.; Jordan, M. I. Conditional Adversarial Domain Adaptation. In Proceedings of the 32nd Conference on Neural Information Processing Systems, Montreal, Canada, 2–8 December 2018; 1647-1657.