Computational and Structural Biotechnology Journal 27 (2025) 1538–1549

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research article

ToxDL 2.0: Protein toxicity prediction using a pretrained language model and graph neural networks

Lin Zhu [a], Yi Fang [b,c], Shuting Liu [a], Hong-Bin Shen [b,c], Wesley De Neve [d,e], Xiaoyong Pan [b,c,*]

[a] School of Health Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[b] Department of Automation, Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China
[c] Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China
[d] Department for Electronics and Information Systems, IDLab, Ghent University, Ghent 9000, Belgium
[e] Department of Environmental Technology, Food Technology and Molecular Biotechnology, Center for Biotech Data Science, Ghent University Global Campus, Songdo, Incheon 305-701, South Korea

ARTICLE INFO

Keywords:
Protein toxicity
Graph Neural Network
Language models
Multi-modal deep learning

ABSTRACT

Motivation: Assessing the potential toxicity of proteins is crucial for both therapeutic and agricultural applications. Traditional experimental methods for protein toxicity evaluation are time-consuming, expensive, and labor-intensive, highlighting the requirement for efficient computational approaches. Recent advancements in language models and deep learning have significantly improved protein toxicity prediction, yet current models often lack the ability to integrate evolutionary and structural information, which is crucial for accurate toxicity assessment of proteins.

Results: In this study, we present ToxDL 2.0, a novel multimodal deep learning model for protein toxicity prediction that integrates both evolutionary and structural information derived from a pretrained language model and AlphaFold2. ToxDL 2.0 consists of three key modules: (1) a Graph Convolutional Network (GCN) module for generating protein graph embeddings based on AlphaFold2-predicted structures, (2) a domain embedding module for capturing protein domain representations, and (3) a dense module that combines these embeddings to predict the toxicity. After constructing a comprehensive toxicity benchmark dataset, we obtained experimental results on both an original non-redundant test set (comprising pre-2022 protein sequences) and an independent non-redundant test set (a holdout set of post-2022 protein sequences), demonstrating that ToxDL 2.0 outperforms existing state-of-the-art methods. Additionally, we utilized Integrated Gradients to discover known toxic motifs associated with protein toxicity. A web server for ToxDL 2.0 is publicly available at www.csbio.sjtu.edu.cn/bioinf/ToxDL2/.

1. Introduction

Proteins are essential biological molecules that play crucial roles in maintaining cellular functions and physiological processes, such as enzyme activity, gene expression regulation, and programmed cell death. The design and development of novel protein drugs have gained increasing prominence in biopharmaceutical research [1]. Therapeutic proteins offer distinct advantages over small molecules, including superior target specificity, enhanced tissue penetration, and intrinsic biological activity [2]. These attributes make them a preferred strategy for treating a broad range of diseases [3,4]. Similarly, in agriculture and food engineering, protein engineering has created new opportunities to improve crop traits and ensure food security. Genetically modified crops leverage modified or exogenous proteins to enhance yield, pest resistance, and herbicide tolerance. These advancements are largely driven by advanced gene-editing technologies like CRISPR, which have revolutionized agricultural practices [5].

Despite these advancements, ensuring the safety of novel synthesized proteins remains a critical challenge in both biomedicine and agriculture, especially in terms of toxicity. Protein toxicity evaluation is

* Corresponding author at: Department of Automation, Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail address: 2008xypan@sjtu.edu.cn (X. Pan).

https://doi.org/10.1016/j.csbj.2025.04.002
Received 26 December 2024; Received in revised form 31 March 2025; Accepted 1 April 2025
Available online 2 April 2025

particularly critical to prevent adverse effects in therapeutic and agricultural applications [6]. Traditional methods for evaluating the toxicity of proteins, peptides, and chemicals primarily rely on experimental toxicity assays. Although these methods are highly reliable, they are often labor-intensive, costly, and often involve ethical concerns related to animal testing [7]. These limitations highlight the need for more efficient *in silico* approaches to assess protein toxicity.

Recent advances in computational methods offer transformative solutions to the challenges of traditional protein toxicity evaluation. Inspired by the success of predictive algorithms in reducing clinical trial failures for small molecules, these computational methods complement experimental assays and streamline the evaluation process, accelerating therapeutic development and promoting the safe application of engineered proteins in both biomedicine and agriculture [8,9]. Currently, protein toxicity prediction methods can be categorized into three main categories: (a) sequence similarity-based approaches, (b) machine learning-based methods, and (c) deep learning-based models. Of them, sequence similarity-based methods rely on tools such as BLAST [10] to calculate the sequence similarity between a query protein and a database of proteins with known toxicity, leveraging homology to predict the toxicity of the query protein [11]. These approaches are straightforward and widely used. However, they suffer from several limitations: 1) The query protein must have homologous counterparts within the toxic protein database. 2) These methods often rely on global sequence similarity, even though toxicity is frequently determined by specific local domains or motifs. 3) Establishing appropriate similarity or e-value cutoffs is challenging, further limiting their applicability [12]. Therefore, sequence similarity-based methods are less effective at identifying toxic potential in novel proteins with significant sequence divergence, which underscore the need for more robust computational methods, such as machine learning, to enhance the accuracy and generalizability of protein toxicity predictions.

Machine learning-based methods have proven to be powerful tools for predicting protein toxicity with a high accuracy. These approaches typically follow a two-step process: first, protein features are extracted using various techniques based on primary sequences, physicochemical properties, and evolutionary profiles. Then, these features are input into classifiers, such as support vector machines (SVMs) [13] or random forests [14], to label proteins as "toxic" or "non-toxic". Several feature-based machine learning models have been developed for protein toxicity prediction, such as ClanTox [15], NNTox [16], ToxClassifier [17], ToxinPred [18], ToxinPred2 [19] and ToxinPred 3.0 [20]. ClanTox focuses on animal toxins, using boosted classifiers with 545-dimensional sequence-derived features to predict toxicity. It is especially effective for identifying short animal toxins based on primary protein sequences [15]. NNTox detects protein toxicity based on several gene ontology annotations, expanding its application beyond simple sequence features to incorporate functional annotations [16]. ToxClassifier uses traditional machine learning methods and common protein descriptors to predict venom toxins. It generates feature representations for proteins and applies classifiers to classify toxins based on these descriptors [17]. ToxinPred uses SVM to classify toxic peptides by analyzing their amino acid composition, dipeptide composition, and toxic motifs [18]. Furthermore, ToxinPred2 and ToxinPred 3.0, which are built upon the original ToxinPred method [19,20], incorporate a broader set of features in combination with hybrid or ensemble machine learning techniques to improve prediction accuracy for both peptides and proteins.

Despite their successes, these feature-based machine learning models have limitations. Their effectiveness is heavily dependent on the quality of hand-designed features, which often require expert knowledge and may not fully capture the complexity of the sequences and structures. Additionally, many of these models are specialized for specific types of toxins (e.g., animal, bacterial, or neurotoxins), limiting their applicability to a broader range of toxins.

To address the limitations of feature-based models, several deep learning-based methods have been developed for protein toxicity prediction. These models, including TOXIFY [21], ToxDL [12], ATSE [22], ToxIBTL [23], CSM_Toxin [24], and VISH-Pred [25], offer end-to-end solutions that eliminate the requirement for manual feature extraction by learning features directly from protein sequences. For instance, TOXIFY employs a gated recurrent unit (GRU) architecture to analyze physicochemical features extracted from protein sequences, enabling it to classify toxic proteins, particularly animal venom proteins [21]. Similarly, ToxDL combines convolutional neural networks (CNNs) to extract local sequence features with protein domain knowledge to predict protein toxicity, specifically for animal-origin proteins [12]. ATSE integrates evolutionary and topological structure information by employing graph neural networks (GNNs) with an attention mechanism to capture discriminative features for peptide toxicity prediction [22]. Additionally, ToxIBTL, an extension of ATSE, introduces the information bottleneck principle and utilizes transfer learning to enhance model effectiveness by transferring knowledge from proteins to peptides [23]. CSM-Toxin is a transformer-based model that predicts protein toxicity by fine-tuning the ProteinBERT [26] architecture on protein sequences. It leverages the attention mechanism to capture complex relationships between distant amino acids [24]. VISH-Pred is an ensemble framework that integrates fine-tuned ESM2 protein language models with machine learning classifiers like LightGBM and XGBoost to predict protein toxicity. It addresses class imbalance through undersampling techniques, offering a streamlined solution for toxicity prediction based solely on sequence data [25].

Although several deep learning-based methods have demonstrated promising effectiveness in protein toxicity prediction, none of the existing approaches explicitly incorporate spatial information [27,28]. Recently, Ebrahimikondori et al. introduced tAMPer [29], a model that combines sequence-derived features from the ESM2 protein language model with 3D structural data encoded as graphs, enhancing peptide toxicity prediction. tAMPer utilizes a hybrid architecture to integrate sequence and structural information, taking advantage of the 3D structures of peptides. However, tAMPer is primarily designed for peptides and lacks the generalization capability necessary for full protein toxicity prediction.

To address these limitations, we aimed to develop an enhanced model building upon our earlier work, ToxDL [12]. While ToxDL is widely utilized within the scientific community [30–32], it has certain constraints that necessitate improvement. Specifically, ToxDL was trained exclusively on animal proteins and relied solely on CNNs [33] to extract sequence information from one-hot encodings, thereby neglecting critical evolutionary information and structural context.

Leveraging the recent advances in protein structure prediction achieved by AlphaFold2 [27,34–37], we introduce ToxDL 2.0, a novel multimodal deep learning model designed specifically for protein toxicity prediction. ToxDL 2.0 integrates multiple types of information and comprises three key components: (1) a Graph Convolutional Network (GCN) module that generates protein graph embeddings by incorporating both evolutionary and structural data; (2) a domain embedding module that captures the average embeddings of all domains within a protein; and (3) a dense module that concatenates the graph embeddings and domain embeddings, utilizing a multilayer perceptron to predict toxicity probabilities. Furthermore, we apply Integrated Gradients to investigate the learned toxic motifs, which align well with known toxic domains.

The main contributions of this work are summarized as follows:

1. We developed a GCN module that extracts protein graph embeddings by leveraging the strengths of pretrained language models and graph neural networks (GNNs).
2. We trained domain embeddings using the Skip-gram model, which learns distributed representations of domains based on domain co-occurrence patterns across a dataset consisting of 200,810,128 proteins and 45,151 domains.

3. We constructed four distinct datasets from UniProt release 2024_03 and ensured high dataset quality by applying the CD-HIT tool to remove redundant sequences.
4. We performed extensive experiments on both an original test set (comprising pre-2022 protein sequences) and an independent test set (a holdout set of post-2022 protein sequences), demonstrating that ToxDL 2.0 consistently outperformed state-of-the-art methods, including its predecessor ToxDL, with robust generalizability.

2. Materials and methods

In this section, we first present the datasets assembled for benchmarking. We then introduce the representations produced by the pre-trained protein language model and the domain embeddings used for generating protein features. Following this, we provide a detailed description of the ToxDL 2.0 network architecture. Finally, we outline the baseline methods, evaluation metrics, and implementation details of the ToxDL 2.0 model. A glossary of technical terms and acronyms, along with their definitions, is provided in the Supplementary Table S1.

2.1. Benchmark dataset construction

2.1.1. Data preparation

To train and evaluate the ToxDL 2.0 model, we employed a strategy similar to that used for original ToxDL to construct a new dataset for protein toxicity prediction. The dataset for this study was derived from UniProt release 2024_03 (released on May 29, 2024) [38] through specific keyword-based queries. Initially, we extracted previously reviewed toxic proteins (positive samples) using the query "(keyword: KW-0800) AND (reviewed: true)" and reviewed non-toxic, non-allergenic proteins (negative samples) using the query "NOT (keyword: KW-0800) AND NOT (keyword: KW-0020) AND (reviewed: true)". These queries resulted in 7776 positive samples and 562,857 negative samples, respectively.

Subsequently, protein sequences with nonstandard amino acids such as B, J, O, U, X, and Z were excluded to ensure unambiguous sequence interpretation and model compatibility. Additionally, proteins with identical amino acid sequences, regardless of UniProt ID or species origin, were removed to eliminate the redundancy. After this filtering process, the dataset was refined to 7385 toxic and 472,645 non-toxic protein sequences.

2.1.2. Benchmark dataset generation

Following previous studies [12,24,25], we further split the refined sequences based on their creation dates to generate an independent test set and a main dataset. A total of 385 toxic and 5247 non-toxic proteins, collected after January 1, 2022, were designated as an independent test set (i.e., a holdout set of post-2022 protein sequences). The remaining sequences, collected before January 1, 2022, were used to form a main dataset of pre-2022 protein sequences, which consists of 7000 toxic and 14,000 non-toxic protein sequences, where the non-toxic sequences were randomly selected from 467,398 negative samples to reduce the significant class imbalance ratio to 1:2.

To create the training set and the original test set, 80 % of the main dataset was randomly selected for model training, while the remaining 20 % was set aside for model testing. To reduce the sequence redundancy, we applied CD-HIT-2D [39] to remove sequences from the test set that shared more than 40 % sequence similarity with any sequence in the training set (with 40 % being the minimum threshold for CD-HIT-2D). Additionally, 10 % of the original training set was reserved as a

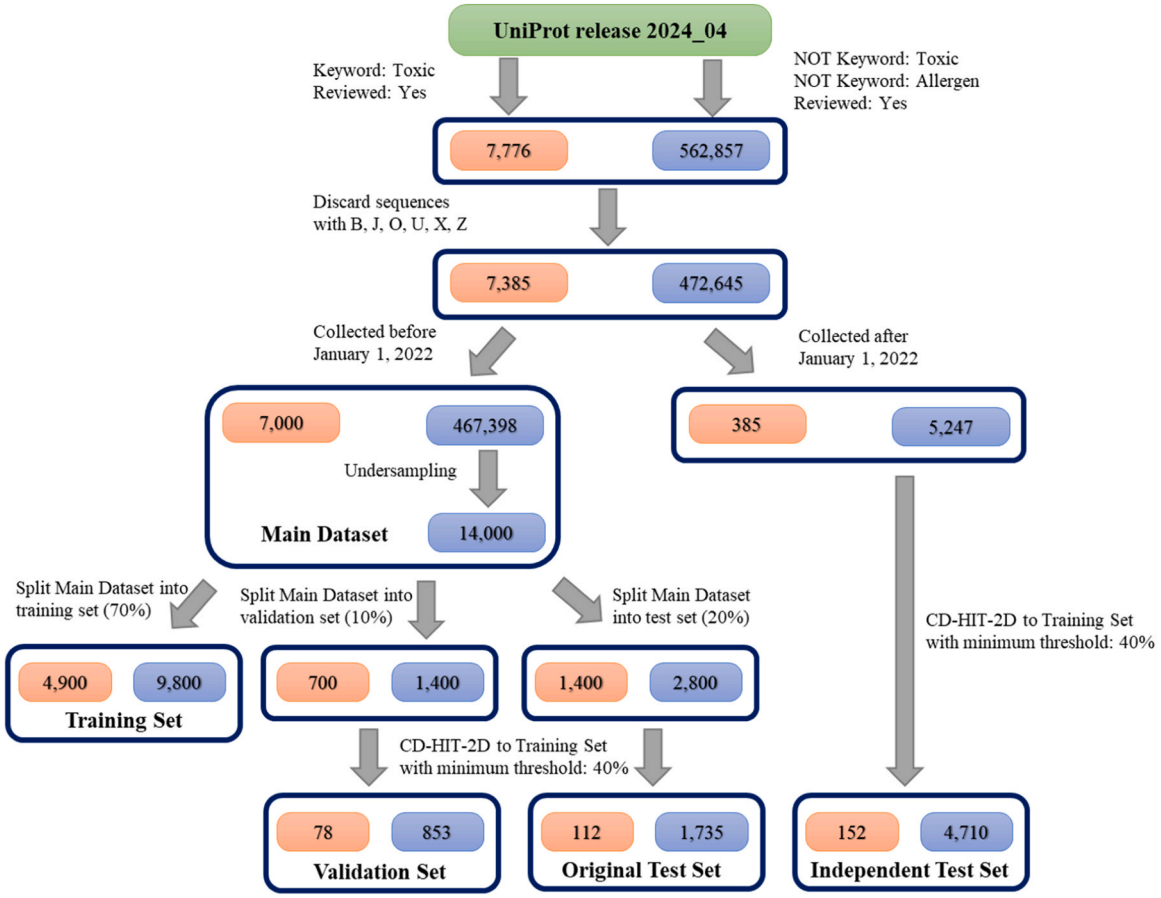

Fig. 1. Data generation pipeline. Data were collected from the UniProt database (release 2024_03) through a keyword search for toxic and non-toxic proteins. To ensure dataset quality and reduce redundancy, sequences with over 40 % similarity to any sequence in the training set were excluded using the CD-HIT-2D tool.

validation set, with homologous sequences similarly removed. This dataset generation process is illustrated in Fig. 1.

Finally, we constructed four distinct datasets, as outlined below:

a) Training Set: This set contains 4900 toxic and 9800 non-toxic protein sequences, selected randomly from the refined positive and negative samples.
b) Validation Set: This set comprises 78 toxic and 853 non-toxic protein sequences, ensuring no sequence redundancy with the training set.
c) Original Test Set: This set contains 112 toxic and 1735 non-toxic protein sequences, ensuring no sequence redundancy with the training set.
d) Independent Test Set: This set consists of 152 toxic and 4710 non-toxic protein sequences, collected after January 1, 2022, ensuring no sequence redundancy with the training set.

The validation set, original test set, and independent test set were generated by applying CD-HIT-2D to the training dataset, ensuring that no sequence shares more than 40 % similarity with any sequence in the training set. This process introduces an additional imbalance between the number of positive and negative samples, thereby creating a more realistic evaluation scenario. A summary of the four different datasets used for benchmarking is provided in Table 1.

2.2. Protein representation

2.2.1. Language model representation for proteins

Protein language models are able to encode evolutionary information within protein sequences and capture essential functional and structural properties [40,41]. To numerically represent protein sequences, we utilized amino acid embeddings generated by the pretrained ESM-2 language model [42]. ESM-2, a comprehensive model trained on UniRef50 protein sequences, effectively capturing both biochemical and co-evolutionary information. In this study, we employed the latest version, esm2_t33_650M_UR50D, which generates a 1280-dimensional feature vector for each residue in the protein sequences.

Definition 1: For each protein sequence, the embedding is denoted as $V = [v_1, v_2, \cdots, v_L]^T \in R^{L\times D}$, where v_i corresponds to the vector representation of the residue at position i. L represents the length of the protein sequence, and $D = 1280$, indicating that each residue is represented by a 1280-dimensional embedding derived from the ESM-2 (esm2_t33_650M_UR50D) model.

2.2.2. Three-dimensional protein structures

The spatial structure of a protein is intricately linked to its function, making the integration of structural properties a promising approach for enhancing toxicity prediction [27,29]. To incorporate spatial interactions for each residue, we obtained the three-dimensional (3D) structures of proteins in PDB format from the RCSB Protein Data Bank (https://www.rcsb.org) [43]. For proteins in the benchmark set without known 3D structures, we obtained predicted 3D structures from the AlphaFold Protein Structure Database [35] or generated predictions using ColabFold (v1.5.2) [29,44], an open-source implementation of AlphaFold2 (v2.3.1) optimized for accessibility and computational efficiency. Following the tutorial at https://github.com/sokrypton/ColabFold, we deployed ColabFold on a GPU server, generating five structural models based on different parameter settings. These models were ranked according to their average pLDDT (predicted Local Distance Difference Test) score, which reflects the confidence of ColabFold in the accuracy of the predicted conformations. The model with the highest average pLDDT score was selected as the final predicted structure for each protein.

Definition 2: A protein graph, based on the 3D structure of a protein, is defined as $G = \{V, E\}$, where V is the set of nodes representing residues, and E is the set of edges representing interactions between residues. In this study, an edge is established between two amino acid nodes in the protein graph if the Euclidean distance between their alpha-carbon (Cα) atoms in 3D space is less than 8 angstroms (Å) [45,46].

Table 1
Overview of the different datasets used for model benchmarking.

Dataset	Number of positives	Number of negatives	Class imbalance ratio
Training Set	4900	9800	1:2
Validation Set	78	853	1:10
Original Test Set	112	1735	1:15
Independent Test Set	152	4710	1:30

2.2.3. Protein domain embeddings

Protein functions are often determined by specific local domains, which may co-occur and interact within a protein [47]. To incorporate domain-level information, we generated protein domain embeddings using the Skip-gram model [48], a widely recognized technique for creating word embeddings that capture the latent representations of words within a continuous vector space.

We first obtained all UniProtKB protein domains generated by InterProScan [49] from https://www.ebi.ac.uk/interpro/download/InterPro. This dataset contains InterPro entries and individual signatures that match UniProtKB proteins, consisting of 200,810,128 proteins and 45,151 domains. The Skip-gram model was then employed to learn distributed representations of domains based on the co-occurrence of domains within a defined context window.

Definition 3: For a set of proteins, the domain embeddings are denoted as $X = [x_1, x_2, \cdots, x_N]^T \in R^{N\times D}$, where x_i corresponds to the embedding for the i*th* protein. N represents the number of proteins in the set, and $D = 256$, indicating that each protein is represented by a 256-dimensional embedding derived from the Skip-gram model. For a protein containing multiple domains, the average domain embedding is used for x_i.

2.3. Network architecture of ToxDL 2.0

This study presents ToxDL 2.0, a multimodal deep learning framework designed for accurate protein toxicity prediction. By combining the strengths of pretrained language models and graph neural networks, ToxDL 2.0 achieves superior predictive performance in identifying protein toxicity. As illustrated in Fig. 2, the architecture of ToxDL 2.0 is comprised of three main components: a Graph Convolutional Network (GCN) module that extracts protein graph embeddings by integrating both evolutionary and structural information, a domain embedding module that generates an average embedding representing all domains within a protein, and a dense module that concatenates graph embeddings and domain embeddings to predict toxicity probabilities through a multilayer perceptron (MLP).

2.3.1. GCN module

Graph Neural Networks (GNNs) [50,51] have garnered significant attention in deep learning, leading to the development of numerous models for representing graph-structured data, particularly for their applications in computational biology. Among these, Graph Convolutional Networks (GCNs) [52] stand out for their ability to propagate information between nodes through convolutional operations. A typical GCN mainly consists of a node features matrix and an adjacency matrix, which together enable the network to effectively aggregate features from neighboring nodes. By leveraging both the intrinsic properties of individual nodes and their local graph structure, GCNs generate updated node representations that are highly effective for prediction tasks. This makes GCNs particularly suitable for applications involving graph-structured data [45].

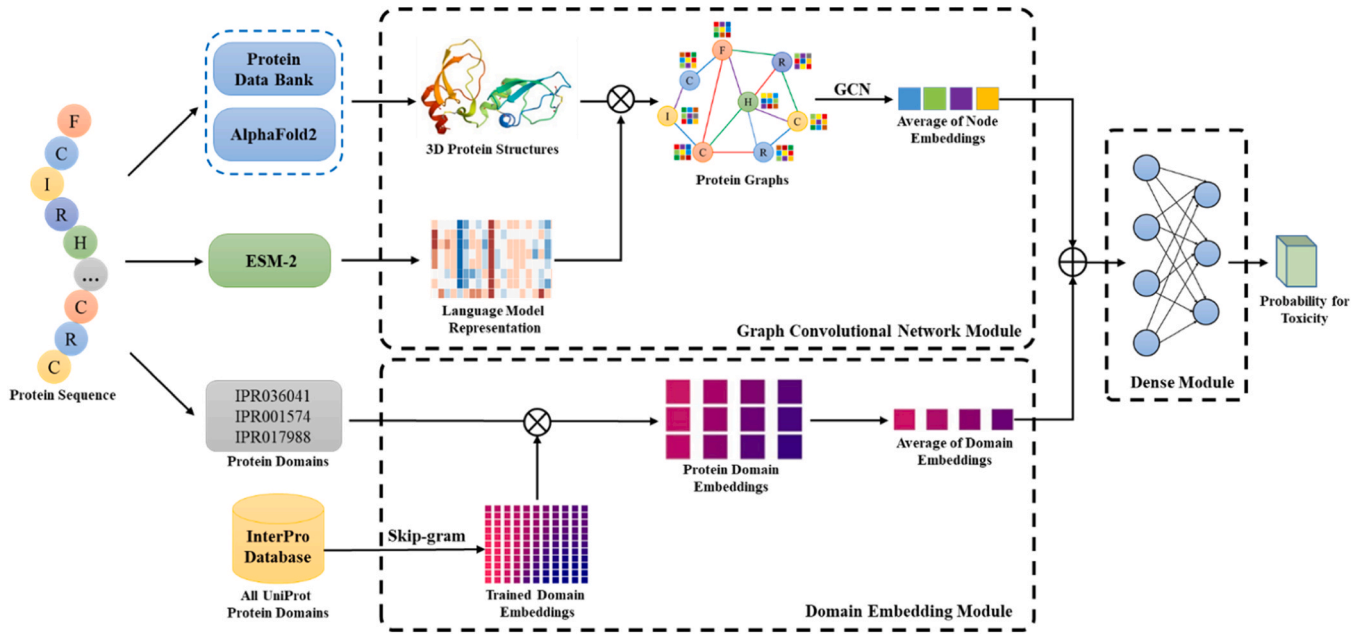

Fig. 2. The architecture of ToxDL 2.0. First, the GCN module processes protein graphs from protein 3D structures into latent representations. Simultaneously, the domain embedding module generates protein domain embeddings. Next, the outputs of these two modules are concatenated and passed through an MLP within the dense module to generate protein toxicity probabilities.

In this study, we employ GCNs to embed protein graphs into fixed-size, graph-level latent representations. A protein graph, which is based on the 3D structure of a protein, is defined as $G = \{V, E\}$, where V represents the set of nodes corresponding to residues, and E represents the set of edges indicating interactions between residues. These edges are defined if the distance between two residues falls below a specified threshold. We utilize the pretrained ESM-2 language model [42] to extract residue features, which are represented as $V = [v_1, v_2, \cdots, v_L]^T \in R^{L\times D}$, where L denotes the number of nodes and D is the dimensionality of the node embeddings. These features serve as the initial node embeddings for the GCN, denoted as $V^{(0)}$.

The GCN module is structured with classic GCN layers, where the embedding of a node $v_i^{(l)}$ at the l-th layer is generated through a message-passing mechanism, aggregating information from neighboring nodes and combining this information with the features of node $v_i^{(l-1)}$. This process can be mathematically formulated as:

$$v_i^{(l)} = ReLU\left(W^{(l)} \times \frac{\sum_{u\in N(v_i)} v_u^{(l-1)}}{|N(v_i)|} + B^{(l)} \times v_i^{(l-1)}\right) \quad (1)$$

where $N(v_i)$ denotes the set of neighbors of node v_i, $v_i^{(l-1)}$ is the node embedding from the (l-1)-th layer, and $W^{(l)}$ and $B^{(l)}$ are the learnable weight matrices.

To incorporate graph-level information, the final component of the GCN module is a pooling function that aggregates all node embeddings to construct a vector representation of the entire graph from the final GCN layer. We employ mean pooling, which calculates the protein graph embeddings by averaging all node embeddings as follows:

$$X_{GCN} = \frac{1}{|V|}\sum_{v_i\in V} v_i^{(L)} \quad (2)$$

where $|V|$ is the number of nodes, and $v_i^{(L)}$ is the embedding of node i from the final GCN layer.

For enhanced clarity, a detailed explanation of the embedding generation process from 3D structures is presented in Supplementary Note S1.

2.3.2. Domain embedding module

Word embeddings are a classic technique in natural language processing (NLP), representing words as continuous vectors in a multi-dimensional space, where the distance and direction between vectors reflect the similarity and relationships among corresponding words [48]. The Word2Vec algorithm, which can utilize either a continuous bag of words (CBOW) or a skip-gram architecture, is widely used for training word embeddings. In this study, we employ the skip-gram model to capture distributed representations of protein domains based on their InterPro annotations [53].

Inspired by the success of word embeddings in NLP, similar approaches have been increasingly adopted in computational biology. Here, we treat each domain as a word, each protein as a sentence, and the entire set of UniProtKB proteins as the corpus. Given a set of proteins, each containing a number of domains $d_1, d_2, \cdots, d_n$, the skip-gram model captures the representation of protein domains by leveraging co-occurrence information within a context window. This model maximizes the following objective function:

$$\frac{1}{N}\sum_{t=1}^{N}\sum_{-c\le j\le c,\ j\neq 0} \log p(d_{t+j}|\ d_t) \quad (3)$$

where N denotes the number of proteins, and c is the context window size.

The skip-gram model defines the conditional probability p using the Softmax function as follows:

$$p\left(d_{t+j}|\ d_t\right) = \frac{\exp\left(v'^{T}_{d_{t+j}} v_{d_t}\right)}{\sum_{i=1}^{D}\exp\left(v_i'^{T} v_{d_t}\right)} \quad (4)$$

where D is the number of domains, and v_{d_t} and $v_{d_{t+j}}$ are the input and output vector representations of the domain d, respectively.

Once training is complete, each domain is represented by a continuous-valued vector. For a given protein P with domains d_P, the domain embeddings of the protein P are calculated as:

$$X_{domain}(P) = \frac{\sum_{d_i\in d_P}\phi(d_i)}{|d_P|} \quad (5)$$

where $\phi()$ is the domain embedding mapping obtained by the trained skip-gram model and $d_i$is the embedding of the i-th domain within the protein.

The final embedding vector corresponding to a given protein is computed by averaging the embeddings of all domains present in that protein. A comprehensive explanation of the methodology used to generate embeddings of domains can be found in Supplementary Note S2.

2.3.3. Dense module

After obtaining the output from the GCN module and the average embedding that represents all domains within the protein, we concatenate the protein graph embedding and the domain embedding to form a combined vector. This vector is then passed through a dense module that generates a toxicity probability. The dense module consists of a fully connected layer and a sigmoid output unit, which can be expressed as follows:

$$Y_{pred} = Sigmoid(ReLU((X_{GCN}||X_{domain})W + b)) \quad (6)$$

where $X_{GCN} \in R^{N\times 256}$ represents the output of the GCN module, and $X_{domain} \in R^{N\times 256}$ represents the output of the domain embedding module. $Y_{pred} \in R^{N\times 1}$ is the predicted result for N protein sequences. W denotes the weight matrix of the fully connected layer, b is the bias term, and $ReLU$ (Rectified Linear Unit) serves as the activation function.

The sigmoid function outputs a probability indicating the predicted likelihood of the protein sequence being toxic. If this probability exceeds 0.5, the protein sequence is classified as 'toxic'; otherwise, it is classified as 'non-toxic'.

To minimize the error between the predicted probability and the true toxicity label, we employ the Binary Cross Entropy loss function, which can be formally described as follows:

$$L_{BCE} = -[Y_{real} \times \log(Y_{pred}) + (1 - Y_{real}) \times \log(1 - Y_{pred})] \quad (7)$$

where Y_{pred} represents the predicted probability, and Y_{real} denotes the true label.

L_{BCE} quantifies the difference between the predicted distribution and the true data distribution by computing the binary cross-entropy between the predicted values and the ground truth, enabling ToxDL 2.0 to be effectively trained for accurate protein toxicity prediction.

2.4. Baseline methods

To evaluate the effectiveness of ToxDL 2.0, we compared it with several baseline models, including two machine learning-based methods (ToxinPred2 [19], ToxinPred 3.0 [20]) and three deep learning-based approaches (ToxDL [12], CSM-Toxin [24], tAMPer [29]). Unfortunately, we were unable to obtain toxicity predictions from ATSE [22] (accessible at http://server.malab.cn/ATSE) and ToxIBTL [23] (available at https://server.weigroup.net/ToxIBTL/Server.html) due to server inaccessibility. Similarly, the VISH-Pred model [25] (accessible at http://ec2-35–170-123–194.compute-1.amazonaws.com:7860/) could not generate predictions for our original test set and independent test set, likely due to service limitations. Therefore, this exclusion introduces a limitation in the comprehensiveness of our study and highlights the need for more accessible and reliable implementations of protein toxicity prediction models. Notably, as shown in earlier evaluations, ToxDL has outperformed BLAST-based and domain search-based methods. Therefore, our comparative analysis focused exclusively on machine learning and deep learning-based methods.

1. **ToxinPred2**: ToxinPred2 is an updated version of the original ToxinPred tool, specifically designed to predict the toxicity of peptides and small proteins. It incorporates BLAST and MERCI-based approaches, along with feature selection and machine learning techniques. For this study, we downloaded ToxinPred2 from https://github.com/raghavagps/toxinpred2 and generated predictions using the Random Forest model, as recommended by the authors.
2. **ToxinPred 3.0**: ToxinPred 3.0 is an enhanced version of ToxinPred, developed to enhance both the reliability and accuracy of peptide toxicity predictions. It employs a combination of machine learning and deep learning techniques, utilizing a hybrid model for better predictive performance. In this study, we applied the Extra Tree algorithm in conjunction with the hybrid approach, which integrates Extra Tree and MERCI. Predictions were generated using the default hyperparameters provided by the authors, and ToxinPred 3.0 can be accessed at https://github.com/raghavagps/toxinpred3.
3. **ToxDL**: ToxDL is a deep learning-based method for *in silico* protein toxicity prediction, leveraging both sequence information and protein domain knowledge. The model comprises a CNN module, a domain2vec module, and an output module that classifies proteins as toxic or non-toxic. We utilized the ToxDL model from https://github.com/xypan1232/ToxDL and re-trained it on our benchmark dataset using the default hyperparameters provided by the authors.
4. **CSM-Toxin**: CSM-Toxin is a novel method for estimating protein toxicity that encodes protein sequence information using ProteinBERT, a deep learning-based natural language model. We uploaded our test sequences to the CSM-Toxin online server (https://biosig.lab.uq.edu.au/csm_toxin/predict) and obtained the predicted results for both the original test set and the independent test set.
5. **tAMPer**: tAMPer is a multimodal, structure-aware deep learning model designed for peptide toxicity prediction. It integrates information from both the amino acid sequence and the three-dimensional structure of peptides to provide a comprehensive representation for toxicity prediction. We downloaded the original tAMPer model from its GitHub repository (https://github.com/bcgsc/tAMPer) and generated probability scores for our test sequences using the default configurations and model checkpoint provided by the authors.

2.5. Evaluation metrics

Similar to the previous studies [12,29], the effectiveness of the proposed ToxDL 2.0 model was evaluated against multiple baseline methods using four performance metrics: F1-score (F1), Matthews correlation coefficient (MCC), area under the receiver operating characteristic curve (auROC), and area under the precision-recall curve (auPRC). These metrics were calculated based on the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as follows:

$$\text{Precision (Prec)} = \frac{TP}{TP + FP} \quad (8)$$

$$\text{Recall (Rec)} = \frac{TP}{TP + FN} \quad (9)$$

$$\text{F1 - score (F1)} = 2 \times \frac{Prec \times Rec}{Prec + Rec} \quad (10)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (11)$$

where TN and TP represent the number of correctly predicted non-toxic and toxic proteins, respectively, while FN and FP denote the number of incorrectly identified non-toxic and toxic proteins, respectively.

The auPRC and auROC metrics were calculated without applying thresholds, providing a comprehensive view of the overall model performance. In contrast, the other metrics were computed using a predefined threshold to convert predicted scores into binary classifications, as suggested by the respective models. For imbalanced classification tasks, auPRC is a more objective and reliable metric [54].

2.6. Implementation details

ToxDL 2.0 was implemented using PyTorch [56] and the PyTorch Geometric library [57], and trained on a GPU. In the GCN module, amino acid embeddings for each sequence were generated using the pretrained ESM-2 model (esm2_t33_650M_UR50D), with each residue represented by a 1280-dimensional embedding. The GCN module consisted of four layers, with embedding dimensions of 512, 512, 512, and 256, respectively. In the domain embedding module, protein domain embeddings were learned by training a skip-gram model for 10 epochs, using a context window size of 5 and an embedding dimension of 256. The dense module comprised three fully connected layers, with a dropout rate of 0.5 applied to prevent overfitting. The Adam optimizer [58] was used with a learning rate of 1e-3, and a learning rate scheduler with a step size of 10 and a decay factor (gamma) of 0.1 was employed to adjust the learning rate throughout training. A batch size of 64 was utilized for model optimization.

To address the class imbalance, the traditional binary cross-entropy (BCE) loss function was replaced with Focal Loss [55], which is specifically designed to address the class imbalance by dynamically adjusting the loss weights. The model was trained for 20 epochs, with the best model selected based on the highest auPRC. The predictive performance of ToxDL 2.0 was evaluated on both the original test set and the independent test set, with results averaged over 10 experimental runs.

3. Experimental results

In this section, we first describe the application of class balancing techniques in ToxDL 2.0, followed by a comparative evaluation of its predictive performance against state-of-the-art methods on both the original and independent test sets. Additionally, we conduct an ablation study to quantify the contribution of individual model components. Furthermore, we present a case study identifying key motifs associated with protein toxicity, using Integrated Gradients-generated saliency maps.

3.1. Application of class balancing techniques in ToxDL 2.0

To demonstrate the advantages of ToxDL 2.0, we employed two distinct test sets. The original test set consists of 112 toxic and 1735 non-toxic protein sequences, randomly selected from UniProt entries uploaded before January 1, 2022. The independent test set, collected after January 1, 2022, contains 152 toxic and 4710 non-toxic protein sequences.

Given the significant class imbalance in both datasets, we replaced the traditional binary cross-entropy (BCE) loss function with Focal Loss [55], a method designed to dynamically adjust loss weights and mitigate the challenges of imbalanced datasets. We conducted comprehensive experiments using various loss functions on both test sets (see Supplementary Tables S2 and S3). For the independent test set, which exhibits a severe imbalance ratio of 1:30, Focal Loss led to improvements in key performance metrics: the F1-Score increased from 0.603 to 0.614, the MCC was improved from 0.592 to 0.610, the auROC was improved from 0.934 to 0.945, and the auPRC increased from 0.602 to 0.611. These results highlight the effectiveness of Focal Loss in handling extreme class imbalance, demonstrating its capacity to enhance model performance in challenging scenarios.

In addition to these metrics, confusion matrices for both test sets were presented in Supplementary Tables S4 and S5, and per-class performance metrics, including Precision, Recall, and F1-Score, were detailed in Supplementary Tables S6 and S7. These results provided an in-depth breakdown of the model's predictions and offered a comprehensive evaluation of the impact of the class imbalance.

To further mitigate the impact of the class imbalance during model selection, we prioritized auPRC as the primary evaluation metric, rather than the validation loss. The auPRC is a more informative measure for imbalanced datasets, as it directly reflects the model's ability to identify the minority class [54]. Supplementary Figure S1 displays the training and validation loss curves (up to 20 epochs), with training/validation loss and auPRC for the validation set. As the number of epochs increases, the training loss steadily decreases, while the validation loss drops sharply and stabilizes after epoch 11, indicating the convergence without model overfitting. The auPRC reaches its peak at around epoch 11. Notably, a strong negative correlation ($r = -0.90$) is observed between the validation loss and auPRC, reinforcing that our approach effectively balances the model generalization to the minority class.

3.2. Performance comparison with baseline methods

To provide a more comprehensive evaluation of ToxDL 2.0's performance, we first calculated the F-max scores, which represent the highest F1-score achieved by varying the classification thresholds incrementally between 0.01 and 1.0, with the optimal threshold selected based on the maximum F1 value. As shown in Supplementary Tables S8 and S9, ToxDL 2.0 consistently outperforms the baseline methods in terms of F-max on both the original and independent test sets, further validating its superior performance for protein toxicity prediction.

Next, we conducted a comparative analysis to assess the effectiveness of ToxDL 2.0 against existing baseline methods. As shown in Table 2, ToxDL 2.0 significantly outperformed all baseline methods on the original test set, achieving an F1-score of 0.878, MCC of 0.869, auROC of 0.992, and auPRC of 0.891. To ensure an objective comparison, we extended our evaluation to the independent test set. As presented in Table 3, ToxDL 2.0 maintained its superior performance across all metrics, with an F1-score of 0.614, MCC of 0.610, auROC of 0.945, and auPRC of 0.611. Notably, ToxDL 2.0 outperformed the second-best method, CSM-Toxin, with a relative improvements of 8.2 % in auROC and 11.5 % in auPRC, underscoring its robust efficacy, particularly on the independent test set.

In Table 3, CSM-Toxin ranks the second to ToxDL 2.0, with an F1-score of 0.574, MCC of 0.566, auROC of 0.873, an auPRC of 0.548. Notably, deep learning-based methods (e.g., ToxDL and tAMPer) consistently outperformed traditional machine learning approaches such as ToxinPred2 (RF) and ToxinPred 3.0 (ET/Hybrid), demonstrating the superiority of deep learning for toxicity prediction. Moreover, it is worth noting that ToxinPred2 outperformed its updated version, ToxinPred 3.0, while tAMPer exhibited slightly weaker predictive performance compared to other deep learning-based models. This discrepancy could be attributed to the limited generalization ability of ToxinPred 3.0 and tAMPer for longer proteins, as these models were originally optimized for peptide-based toxicity prediction. Compared to tAMPer,

Table 2
Performance comparison of ToxDL 2.0 and other methods on the original test set.

Method	F1-score	MCC	auROC	auPRC
ToxinPred2 (RF)	0.795	0.790	0.986	0.805
ToxinPred 3.0 (ET)	0.612	0.587	0.816	0.463
ToxinPred 3.0 (Hybrid)	0.619	0.595	0.832	0.512
ToxDL (re-trained)	0.794 (±0.018)	0.780 (±0.019)	0.978 (±0.002)	0.744 (±0.032)
CSM_Toxin	0.848	0.842	0.985	0.886
tAMPer	0.803	0.795	0.926	0.825
ToxDL 2.0	**0.878** (±0.014)	**0.869** (±0.013)	**0.992** (±0.002)	**0.891** (±0.014)

Note: To account for randomness during the training of deep learning methods, we report the average results along with standard deviation for ToxDL and ToxDL 2.0 after conducting each experiment 10 times. Baseline methods (ToxinPred2, ToxinPred 3.0, CSM-Toxin, tAMPer) were evaluated using pre-trained models, yielding consistent results across repeated runs. Bold indicates that the method is the best among the compared methods.

Table 3
Performance comparison of ToxDL 2.0 and other methods on the independent test set.

Method	F1-score	MCC	auROC	auPRC
ToxinPred2 (RF)	0.438	0.421	0.864	0.339
ToxinPred 3.0 (ET)	0.422	0.404	0.822	0.242
ToxinPred 3.0 (Hybrid)	0.435	0.415	0.833	0.258
ToxDL (re-trained)	0.501 (±0.019)	0.503 (±0.018)	0.888 (±0.008)	0.450 (±0.026)
CSM_Toxin	0.574	0.566	0.873	0.548
tAMPer	0.509	0.496	0.857	0.445
ToxDL 2.0	**0.614** (±0.025)	**0.610** (±0.022)	**0.945** (±0.006)	**0.611** (±0.020)

Note: To account for randomness during the training of deep learning methods, we report the average results along with standard deviation for ToxDL and ToxDL 2.0 after conducting each experiment 10 times. Baseline methods (ToxinPred2, ToxinPred 3.0, CSM-Toxin, tAMPer) were evaluated using pre-trained models, yielding consistent results across repeated runs. Bold indicates that the method is the best among the compared methods.

ToxDL 2.0 integrates protein domain embeddings, which substantially enhance its predictive accuracy, demonstrating the effectiveness of leveraging domain-level information for protein toxicity prediction.

When comparing ToxDL 2.0 with its predecessor, ToxDL, significant improvements were observed across both test sets. As shown in Table 2, ToxDL 2.0 improved the auROC of ToxDL from 0.978 to 0.992 (a relative gain of 1.2 %) and increased auPRC from 0.744 to 0.891 (a 19.8 % relative increase). Similarly, in Table 3, ToxDL 2.0 achieved a higher auROC of 0.945 (up from 0.888, a 6.4 % increase) and improved auPRC from 0.450 to 0.611 (a relative gain of 35.8 %) on the independent test set. These improvements are attributed to the graph embeddings generated by the GCN module in ToxDL 2.0, which effectively incorporates both evolutionary and structural information. In contrast, ToxDL relied primarily on CNNs to extract high-level sequence information from one-hot encodings.

Finally, we present the precision-recall curves for ToxDL 2.0 and the baseline methods in Figs. 3 and 4. Although all models showed performance degradation on the independent test set, likely due to the increased imbalance and complexity of the data, ToxDL 2.0 maintained a substantial advantage in auPRC, outperforming the baseline methods by a significant margin. Given that auPRC is particularly effective for evaluating the performance on imbalanced datasets, these curves further highlight ToxDL 2.0's superiority in protein toxicity prediction.

3.3. Ablation studies on ToxDL 2.0

ToxDL 2.0 incorporates a GCN module and a domain embedding module, both preceding the final output layer. To assess the contribution of each module, we performed an ablation study on both the original test set and the independent test set. Additionally, we performed the ablation by removing the GCN module entirely and replacing the protein domain embeddings with averaged ESM residue embeddings. In this approach, per-residue embeddings were extracted from the pretrained ESM-2 language model, and the mean of these embeddings was computed to generate a fixed-size protein-level embedding, which was fed into the MLP for protein toxicity prediction. The experimental results were summarized in Table 4.

As shown in Table 4, domain embeddings-only model, which excludes the GCN module, resulted to a significant drop in performance. On the original test set, the auROC decreased by 2.6 %, and the auPRC by 23.7 %. On the independent test set, the auROC and auPRC decreased by 4.8 % and 25.1 %, respectively. The substantial decrease in auPRC is particularly significant, as this metric is particularly suitable for imbalanced classification tasks [54]. These results highlight the role of the GCN module in capturing critical evolutionary and spatial information from protein sequences and structures. Similarly, removing the domain embedding module also resulted in a slight decline in predictive performance. For GCN-w-ESM-only model, the auROC drops from 0.992 to 0.981 and the auPRC drops from 0.891 to 0.874 on the original test set. For the independent test set, the auROC decreased from 0.945 to 0.913, and the auPRC dropped from 0.611 to 0.530. The results suggest that the domain embedding module effectively preserves essential domain-level information associated with protein toxicity.

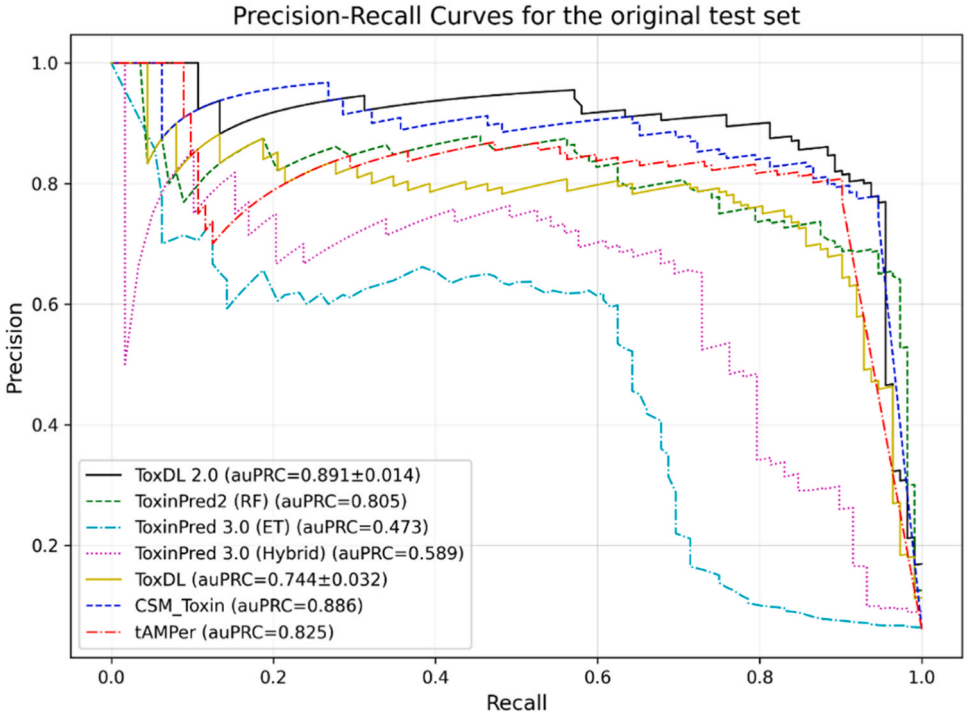

Fig. 3. Precision-Recall curves of proposed ToxDL 2.0 and baseline methods for the original test set.

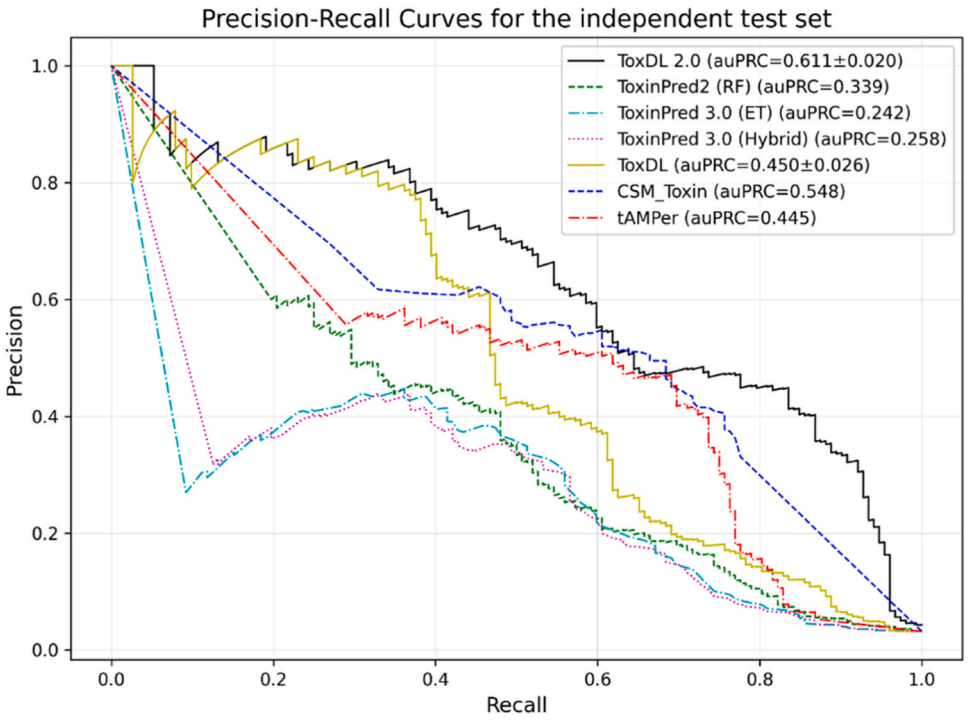

Fig. 4. Precision-Recall curves of proposed ToxDL 2.0 and baseline methods for the independent test set.

Table 4
Performance evaluation of ablation studies on ToxDL 2.0.

Dataset	Model	F1-score	MCC	auROC	auPRC
Original Test Set	domain embeddings-only model	0.640 (±0.007)	0.644 (±0.008)	0.966 (±0.002)	0.654 (±0.028)
	ESM embeddings-only model	0.829 (±0.010)	0.821 (±0.011)	0.979 (±0.001)	0.870 (±0.020)
	GCN-w-ESM-only model	0.839 (±0.016)	0.829 (±0.017)	0.981 (±0.001)	0.874 (±0.022)
	ToxDL 2.0	**0.878** (±0.014)	**0.869** (±0.013)	**0.992** (±0.002)	**0.891** (±0.014)
Independent Test Set	domain embeddings-only model	0.302 (±0.009)	0.345 (±0.010)	0.897 (±0.004)	0.360 (±0.025)
	ESM embeddings-only model	0.526 (±0.029)	0.525 (±0.023)	0.904 (±0.004)	0.512 (±0.019)
	GCN-w-ESM-only model	0.541 (±0.038)	0.544 (±0.028)	0.913 (±0.004)	0.530 (±0.020)
	ToxDL 2.0	**0.614** (±0.025)	**0.610** (±0.022)	**0.945** (±0.006)	**0.611** (±0.020)

Furthermore, a comparison of the GCN-w-ESM-only model with both the domain embeddings-only and ESM embeddings-only models reveals that the GCN-w-ESM-only model consistently outperforms the other models across all evaluation metrics on both test sets. This result reinforces the importance of the GCN module in protein toxicity prediction. The superior predictive performance of the GCN module can be attributed to its ability to generalize more effectively to unseen proteins by leveraging evolutionary and spatial information. Specifically, on the independent test set, the GCN-w-ESM-only model showed relative improvements over the ESM embeddings-only model, with increases of 2.8 % in F1-score, 3.6 % in MCC, 1.0 % in auROC, and 3.5 % in auPRC. For the domain embeddings-only model, these improvements were even more significant, with increases of 79.1 %, 57.7 %, 1.2 %, and 47.2 %, respectively. These results further demonstrate that relying solely on local domain embeddings or ESM embeddings is insufficient for accurate toxicity prediction, as it neglects critical global sequence information and structural context.

In summary, the results of the ablation study indicate that combining the GCN and domain embedding modules in ToxDL 2.0 yields the best overall predictive performance, benefiting from the unique contributions of each module in protein toxicity prediction.

3.4. Case study

To demonstrate the effectiveness of ToxDL 2.0, we present a case study involving a protein (UniProt AC: P0DQR5) from the independent test set. This protein, also referred to as PNTX_XIBTU, is classified as a putative neurotoxin from the venom of *Xibalbanus tulumensis*, a blind cave-dwelling remipede native to marine environments. ToxDL 2.0 accurately predicted P0DQR5 as being toxic, assigning it a high toxicity probability of 0.996. To further investigate this prediction, we employed Integrated Gradients [56] to generate saliency maps, which enabled the identification of critical toxic motifs within the sequence.

As illustrated in Fig. 5A, P0DQR5 aligns well with the Pfam PF02819 entry in the InterPro database [53], which classifies it within the spider toxin family for neurotoxic activity. Specifically, the region spanning amino acids 34–68 in P0DQR5 corresponds to the Pfam PF02819 domain. Fig. 5B presents a saliency map for the positions 34–68 of this protein sequence, where amino acids contributing significantly to the toxicity prediction are highlighted with larger letters. Additionally, Fig. 5C shows the Profile Hidden Markov Model (HMM) of Pfam PF02819, demonstrating a high similarity between the toxic motif detected by ToxDL 2.0 and the conserved features of the PF02819 domain. In particular, cysteine (C) residues frequently occur and make substantial contributions to the toxicity score.

Additionally, in Supplementary Figures S4, S5, and S6, we provide detailed analyses of the toxic motifs identified by ToxDL 2.0 for three additional proteins: Q8T0W8, A0A8K1YTT5, and P0DM30. These case studies further demonstrate the capacity of ToxDL 2.0 to accurately identify critical toxic motifs *in silico*, highlighting its potential applications in protein engineering and drug discovery. By pinpointing critical mutations within toxic motifs, ToxDL 2.0 enables efficient screening of

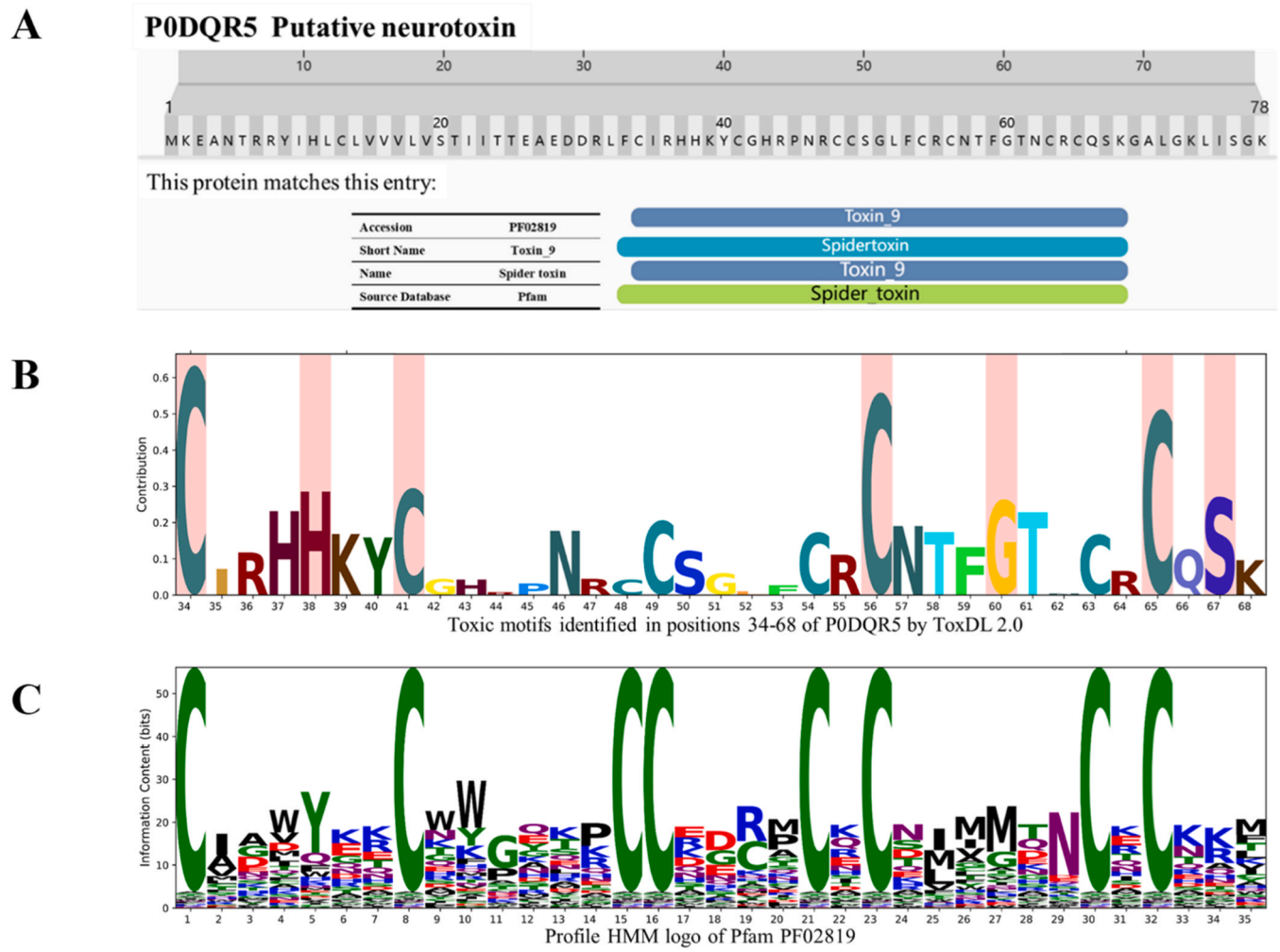

Fig. 5. Identification of toxic motifs in P0DQR5 by ToxDL 2.0. (A) Alignment of P0DQR5 with the Pfam PF02819 entry in the InterPro database. (B) Toxic motifs identified in positions 34–68 of P0DQR5 by ToxDL 2.0, with significant residues emphasized in larger font sizes. (C) Profile HMM logo of Pfam PF02819, clearly aligned with the toxic motifs shown in (B).

mutated sequences, allowing researchers to rapidly assess the impact of specific mutations on protein toxicity. This functionality offers considerable promise for guiding the design of novel, non-toxic proteins. While Integrated Gradients is well-suited for analyzing sequential data, other interpretable methods, such as SHAP (Shapley Additive Explanations) [57] and Grad-CAM (Gradient-weighted Class Activation Mapping) [58], could provide complementary insights into model interpretability. Future studies integrating these techniques could offer a more comprehensive understanding of the important features driving toxicity predictions.

4. Discussion and conclusions

Proteins are fundamental for cellular processes, playing crucial roles in both health and disease. Protein-based therapeutics have emerged as powerful tools for treating a wide array of conditions, including cancer, diabetes, and neurodegenerative disorders. Conversely, toxins, commonly proteins, peptides, or small chemical molecules, contribute to pathogenicity in plants, animals, and microbes. Traditional toxicity assays, while reliable, are labor-intensive and costly. In contrast, *in silico* approaches provide a faster, more cost-effective means of toxin classification, revealing the underlying toxicity mechanisms and facilitating the identification of candidate non-toxic protein drugs. The adoption of artificial intelligence and data-driven computational techniques has significantly improved the accuracy and efficiency of protein toxicity prediction, making these methods an essential tool for the rapid screening of non-toxic proteins.

In this study, we introduce ToxDL 2.0, a novel multimodal deep learning model for protein toxicity prediction that integrates both evolutionary and structural information derived from a pretrained language model and AlphaFold2. ToxDL 2.0 performs robustly across both our original test set and independent test set, demonstrating its effectiveness in protein toxicity prediction. However, there is still room for further improvement. A key challenge is the class imbalance in protein toxicity datasets, where toxic proteins are often underrepresented compared to non-toxic proteins, affecting the ability of the model to generalize across diverse classes. Additionally, while ToxDL 2.0 performs well in binary toxicity classification, further advancements are needed to address the multiclass toxicity prediction problem, as toxicity profiles can differ across species and contexts, such as between cell types. We introduce the proposed framework for handling multiclass toxicity prediction in Supplementary Note S3.

Future work will focus on refining the model by leveraging advanced Graph Convolutional Networks (GCNs) to more effectively capture the topological features of AlphaFold2-predicted structures. Additionally, we aim to address the challenges posed by class imbalance and multiclass toxicity data. Moreover, incorporating explainable AI (XAI) techniques [59], such as Graph Neural Additive Networks (GNAN) [60], will improve model interpretability, providing deeper insights into the specific features of protein sequences and structures that contribute to

toxicity predictions. In addition, the future research could explore the integration of SHAP [57] and Grad-CAM [58] to complement the insights provided by Integrated Gradients. Combining these interpretable techniques can further enhance model transparency and aid in the refinement of protein toxicity predictions.

In summary, ToxDL 2.0 represents a powerful tool for protein toxicity prediction, combining evolutionary and structural information with deep learning techniques. Future advancements in addressing class imbalance, multiclass toxicity, and model interpretability will further enhance its predictive accuracy and usability. ToxDL 2.0 provides a valuable resource for both biomedicine and agriculture, and the availability of a supporting web server at www.csbio.sjtu.edu.cn/bioinf/ToxDL2/ ensures easy access to this tool for further experimentation and application.

CRediT authorship contribution statement

Fang Yi: Software, Resources. **Liu Shuting:** Writing – original draft, Data curation. **Shen Hong-Bin:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **De Neve Wesley:** Writing – review & editing, Supervision, Conceptualization. **Zhu Lin:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation. **Pan Xiaoyong:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Funding

This work is supported by the National Natural Science Foundation of China (No. 62473257, 62073219), and the Science and Technology Commission of Shanghai Municipality (24ZR1435300, 24510714300), Shanghai Key Laboratory of Forensic Medicine and Key Laboratory of Forensic Science.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.csbj.2025.04.002.

Data availability

ToxDL 2.0 is publicly available through a user-friendly webserver at http://www.csbio.sjtu.edu.cn/bioinf/ToxDL2. A detailed description of the webserver implementation can be found in Supplementary Note S4. The source code for ToxDL 2.0 is openly available on GitHub at http://github.com/shzhulin/ToxDL2. The repository includes detailed documentation and instructions for setting up the required environment, ensuring accessibility and ease of use for researchers. The benchmark dataset, sourced from the UniProt database, can be downloaded at http://www.csbio.sjtu.edu.cn/bioinf/ToxDL2/Data.htm. Protein structure data used in this study were retrieved from the RCSB PDB database (http://www.rcsb.org/downloads/), while predicted structures were obtained from the AlphaFold Protein Structure Database (http://www.alphafold.ebi.ac.uk/).

References

[1] Bruno BJ, Miller GD, Lim CS. Basics and recent advances in peptide and protein drug delivery. Ther Deliv 2013;4(11):1443–67. https://doi.org/10.4155/tde.13.104.
[2] Vlieghe P, Lisowski V, Martinez J, Khrestchatisky M. Synthetic therapeutic peptides: science and market. Drug Discov Today 2010;15(1-2):40–56. https://doi.org/10.1016/j.drudis.2009.10.009.
[3] Sundaram B, Pandian N, Mall R, et al. NLRP12-PANoptosome activates PANoptosis and pathology in response to heme and PAMPs. Cell 2023;186(13):2783–801. https://doi.org/10.1016/j.cell.2023.05.005. e2720.
[4] Frattini, Pagnotta V, Tala SM, et al. A metabolic function of FGFR3-TACC3 gene fusions in cancer. Nature 2018;553(7687):222–7. https://doi.org/10.1038/nature25171.
[5] Bernabe-Orts JM, Casas-Rodrigo I, Minguet EG, et al. Assessment of Cas12a-mediated gene editing efficiency in plants. Plant Biotechnol J 2019;17(10):1971–84. https://doi.org/10.1111/pbi.13113.
[6] Usmani SS, Bedi G, Samuel JS, et al. THPdb: database of FDA-approved peptide and protein therapeutics. PLoS One 2017;12(7):e0181748. https://doi.org/10.1371/journal.pone.0181748.
[7] Duracova M, Klimentova J, Fucikova A, Dresler J. Proteomic methods of detection and quantification of protein toxins. Toxins (Basel) 2018;10(3). https://doi.org/10.3390/toxins10030099.
[8] Zhang MQ, Wilkinson B. Drug discovery beyond the 'rule-of-five. Curr Opin Biotechnol 2007;18(6):478–88. https://doi.org/10.1016/j.copbio.2007.10.005.
[9] Liu X, Wu F, Ji Y, Yin L. Recent advances in anti-cancer protein/peptide delivery. Bioconjug Chem 2019;30(2):305–24. https://doi.org/10.1021/acs.bioconjchem.8b00750.
[10] Altschul SF, Madden TL, Schaffer AA, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25(17):3389–402. https://doi.org/10.1093/nar/25.17.3389.
[11] Negi SS, Schein CH, Ladics GS, et al. Functional classification of protein toxins as a basis for bioinformatic screening. Sci Rep 2017;7(1):13940. https://doi.org/10.1038/s41598-017-13957-1.
[12] Pan X, Zuallaert J, Wang X, et al. ToxDL: deep learning using primary structure and domain embeddings for assessing protein toxicity. Bioinformatics 2021;36(21):5159–68. https://doi.org/10.1093/bioinformatics/btaa656.
[13] Cortes C, Vapnik V. Support-vector networks. Mach Learn 1995;20:273–97. https://doi.org/10.1007/BF00994018.
[14] Geurts P, Ernst D, Wehenkel. LJMl. Extremely randomized trees. Mach Learn 2006;63:3–42. https://doi.org/10.1007/s10994-006-6226-1.
[15] Naamati G, Askenazi M, Linial M. ClanTox: a classifier of short animal toxins. Nucleic Acids Res 2009;37(2):W363–368. https://doi.org/10.1093/nar/gkp299.
[16] Jain A, Kihara D. NNTox: gene ontology-based protein toxicity prediction using neural network. Sci Rep 2019;9(1):17923. https://doi.org/10.1038/s41598-019-54405-6.
[17] Gacesa R, Barlow DJ, Long PF. Machine learning can differentiate venom toxins from other proteins having non-toxic physiological functions. PeerJ Comput Sci 2016;2:e90. https://doi.org/10.7717/peerj-cs.90.
[18] Gupta S, Kapoor P, Chaudhary K, et al. In silico approach for predicting toxicity of peptides and proteins. PLoS One 2013;8(9):e73957. https://doi.org/10.1371/journal.pone.0073957.
[19] Sharma N, Naorem LD, Jain S, Raghava GPS. ToxinPred2: an improved method for predicting toxicity of proteins. Brief Bioinform 2022;23(5). https://doi.org/10.1093/bib/bbac174.
[20] Rathore AS, Choudhury S, Arora A, Tijare P, Raghava GPS. ToxinPred 3.0: An improved method for predicting the toxicity of peptides. Comput Biol Med 2024;179:108926. https://doi.org/10.1016/j.compbiomed.2024.108926.
[21] Cole TJ, Brewer MS. TOXIFY: a deep learning approach to classify animal venom proteins. PeerJ 2019;7:e7200. https://doi.org/10.7717/peerj.7200.
[22] Wei L, Ye X, Xue Y, Sakurai T, Wei L. ATSE: a peptide toxicity predictor by exploiting structural and evolutionary information based on graph neural network and attention mechanism. Brief Bioinform 2021;22(5). https://doi.org/10.1093/bib/bbab041.
[23] Wei L, Ye X, Sakurai T, Mu Z, Wei L. ToxIBTL: prediction of peptide toxicity based on information bottleneck and transfer learning. Bioinformatics 2022;38(6):1514–24. https://doi.org/10.1093/bioinformatics/btac006.
[24] Morozov V, Rodrigues CHM, Ascher DB. CSM-toxin: a web-server for predicting protein toxicity. Pharmaceutics 2023;15(2). https://doi.org/10.3390/pharmaceutics15020431.
[25] Mall R, Singh A, Patel CN, Guirimand G, Castiglione F. VISH-Pred: an ensemble of fine-tuned ESM models for protein toxicity prediction. Brief Bioinform 2024;25(4). https://doi.org/10.1093/bib/bbae270.
[26] Brandes N, Ofer D, Peleg Y, Rappoport N, Linial M. ProteinBERT: a universal deep-learning model of protein sequence and function. Bioinformatics 2022;38(8):2102–10. https://doi.org/10.1093/bioinformatics/btac020.
[27] Peng Z, Wang W, Han R, Zhang F, Yang J. Protein structure prediction in the deep learning era. Curr Opin Struct Biol 2022;77:102495. https://doi.org/10.1016/j.sbi.2022.102495.
[28] Zhang J, Qian J, Zou Q, Zhou F, Kurgan L. Recent advances in computational prediction of secondary and supersecondary structures from protein sequences. Methods Mol Biol 2025;2870:1–19. https://doi.org/10.1007/978-1-0716-4213-9_1.
[29] Ebrahimikondori H, Sutherland D, Yanai A, et al. Structure-aware deep learning model for peptide toxicity prediction. Protein Sci 2024;33(7):e5076. https://doi.org/10.1002/pro.5076.
[30] Parthiban S, Vijeesh T, Gayathri T, Shanmugaraj B, Sharma A, Sathishkumar R. Artificial intelligence-driven systems engineering for next-generation plant-derived biopharmaceuticals. Front Plant Sci 2023;14:1252166. https://doi.org/10.3389/fpls.2023.1252166.

[31] Singh S, Kaur N, Gehlot A. Application of artificial intelligence in drug design: a review. Comput Biol Med 2024;179:108810. https://doi.org/10.1016/j.compbiomed.2024.108810.
[32] Kumar N, Srivastava R. Deep learning in structural bioinformatics: current applications and future perspectives. Brief Bioinform 2024;25(3). https://doi.org/10.1093/bib/bbae042.
[33] Li Z, Liu F, Yang W, Peng S, Zhou J. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst 2022;33(12):6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827.
[34] Zeng Y, Wei Z, Yuan Q, et al. Identifying B-cell epitopes using AlphaFold2 predicted structures and pretrained language model. Bioinformatics 2023;39(4). https://doi.org/10.1093/bioinformatics/btad187.
[35] Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold. Nature 2021;596(7873):583–9. https://doi.org/10.1038/s41586-021-03819-2.
[36] Ismi DP, Pulungan R, Afiahayati. Deep learning for protein secondary structure prediction: pre and post-AlphaFold. Comput Struct Biotechnol J 2022;20:6271–86. https://doi.org/10.1016/j.csbj.2022.11.012.
[37] Hu W, Ohue M. SpatialPPI: three-dimensional space protein-protein interaction prediction with AlphaFold Multimer. Comput Struct Biotechnol J 2024;23:1214–25. https://doi.org/10.1016/j.csbj.2024.03.009.
[38] UniProt C. UniProt: the Universal Protein Knowledgebase in 2025. Nucleic Acids Res 2024. https://doi.org/10.1093/nar/gkae1010.
[39] Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics 2012;28(23):3150–2. https://doi.org/10.1093/bioinformatics/bts565.
[40] Liu W, Wang Z, You R, et al. PLMSearch: protein language model powers accurate and fast sequence search for remote homology. Nat Commun 2024;15(1):2775. https://doi.org/10.1038/s41467-024-46808-5.
[41] Liu W, Wang Z, You R, et al. Author correction: PLMSearch: protein language model powers accurate and fast sequence search for remote homology. Nat Commun 2024;15(1):7766. https://doi.org/10.1038/s41467-024-52177-w.
[42] Lin Z, Akin H, Rao R, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science 2023;379(6637):1123–30. https://doi.org/10.1126/science.ade2574.
[43] Berman HM, Battistuz T, Bhat TN, et al. The protein data bank. Acta Crystallogr D Biol Crystallogr 2002;58(Pt 6 1):899–907. https://doi.org/10.1107/s0907444902003451.
[44] Mirdita M, Schutze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M. ColabFold: making protein folding accessible to all. Nat Methods 2022;19(6):679–82. https://doi.org/10.1038/s41592-022-01488-1.
[45] Wang Y, Xia Y, Yan J, Yuan Y, Shen HB, Pan X. ZeroBind: a protein-specific zero-shot predictor with subgraph matching for drug-target interactions. Nat Commun 2023;14(1):7861. https://doi.org/10.1038/s41467-023-43597-1.
[46] Xie Z, Xu J. Deep graph learning of inter-protein contacts. Bioinformatics 2022;38(4):947–53. https://doi.org/10.1093/bioinformatics/btab761.
[47] Menichelli C, Gascuel O, Brehelin L. Improving pairwise comparison of protein sequences with domain co-occurrence. PLoS Comput Biol 2018;14(1):e1005889. https://doi.org/10.1371/journal.pcbi.1005889.
[48] Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. Proc 26th Int Conf Neural Inf Process Syst 2013;26(2):3111–9. https://doi.org/10.48550/arXiv.1310.4546.
[49] Jones P, Binns D, Chang HY, et al. InterProScan 5: genome-scale protein function classification. Bioinformatics 2014;30(9):1236–40. https://doi.org/10.1093/bioinformatics/btu031.
[50] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Trans Neural Netw 2009;20(1):61–80. https://doi.org/10.1109/TNN.2008.2005605.
[51] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 2021;32(1):4–24. https://doi.org/10.1109/TNNLS.2020.2978386.
[52] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint. 2016;arXiv(1609.02907) 2016:1–14. https://doi.org/10.48550/arXiv.1609.02907.
[53] Blum M, Chang HY, Chuguransky S, et al. The InterPro protein families and domains database: 20 years on. Nucleic Acids Res 2021;49(D1):D344–54. https://doi.org/10.1093/nar/gkaa977.
[54] Hancock JT, Khoshgoftaar TM, Johnson JM. Evaluating classifier performance with highly imbalanced big data. J Big Data 2023;10(1):1–42. https://doi.org/10.1186/s40537-023-00724-5.
[55] Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. Proc IEEE Int Conf Comput Vis 2017;1(1):2980–8. https://doi.org/10.1109/ICCV.2017.324.
[56] Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. Proc 34th Int Conf Mach Learn 2017;70:3319–28. https://doi.org/10.48550/arXiv.1703.01365.
[57] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. Proc 31st Int Conf Neural Inf Process Syst 2017;30(1):4768–77. https://doi.org/10.48550/arXiv.1705.07874.
[58] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. Int J Comput Vis 2020;128(1):336–59. https://doi.org/10.1007/s11263-019-01228-7.
[59] Tjoa E, Guan C. A survey on explainable artificial intelligence (XAI): toward medical XAI. IEEE Trans Neural Netw Learn Syst 2021;32(11):4793–813. https://doi.org/10.1109/TNNLS.2020.3027314.
[60] Bechler-Speicher M., Globerson A., Gilad-Bachrach R. The Intelligible and Effective Graph Neural Additive Networks. arXiv preprint. 2024;arXiv(2406.01317):1-20. https://doi.org/10.48550/arXiv.2406.01317.