DEP: Dual-Path Embeddings for Protein Toxicity Classification

André Gambogi, Arthur Buzelin, Gabriela Miserani, Guilherme H. G. Evangelista, Pedro Bento, Yan Aquino, Samira Malaquias, Pedro Bacelar, Pedro Robles Dutenhefner, Wagner Meira Jr., Gisele L. Pappa

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

{andregambogi, arthurbuzelin, gabrielamiserani, guilherme.evangelista

pedro.bento, yanaquino, samiramalaquias,pedro.bacelar

meira, glpappa}@dcc.ufmg.br pedroroblesduten@ufmg.br

Abstract. *The reliance of protein predictors on computationally expensive 3D structures or MSAs severely limits large-scale screening. To overcome this limitation, we propose a lightweight, dual-path architecture operating exclusively on 1D sequence embeddings. Our model fuses representations from a Local-Hierarchical Path, designed to capture functional motifs, with a Global-Holistic Path that models long-range dependencies. Evaluated on a benchmark toxicity dataset, our model establishes a new state-of-the-art with an AUC-ROC of 0.966, surpassing complex models that require structural inputs. These results show that a well-designed, sequence-only approach can be a faster, more scalable, and even better-performing alternative to structure-based methods.*

1. Introduction

Predicting the biochemical properties of proteins is critical for the development of effective vaccines and therapeutic agents. Before a candidate molecule can move forward in development, it must meet strict safety and efficacy requirements, including being non-toxic and chemically stable. Historically, identifying failures in these properties occurred late in the development cycle, during expensive and time-consuming experimental validation. A single failed formulation could lead to wasted resources, adverse clinical reactions, or poor bioavailability, compromising years of research [Rappuoli et al. 2011] [Bento et al. 2025].

To mitigate these risks, bioinformatics has introduced computational tools capable of predicting protein properties *in silico*. These models allow researchers to flag unsuitable candidates based on their primary sequence, simplifying the design process by filtering out problematic molecules before laboratory testing. However, a growing trend has emerged: the most accurate models increasingly depend on computationally intensive inputs. State-of-the-art approaches often rely on 3D structural predictions from tools like AlphaFold2 or on evolutionary information derived from multiple sequence alignments (MSAs) [Jumper et al. 2021]. While these inputs enhance predictive power, they severely limit scalability, making such methods impractical for large-scale screening and inaccessible for newly sequenced proteins that lack structural or evolutionary annotations.

This dependency has created a critical research gap. Despite the widespread availability of powerful pre-trained protein language models (PLMs) like ESM, which encode rich functional and structural information directly within 1D sequence embeddings, their full potential remains underexplored. The field has largely overlooked the question of how far these embeddings can be pushed when paired with sophisticated, task-specific architectures. Rather than designing architectures to fully leverage this rich sequence-level information, the prevailing trend has been to supplement these embeddings with complex, external data, assuming that structural inputs are a prerequisite for top-tier performance.

In this work, we challenge this assumption. We introduce DEP (Dual-Path Embeddings for Protein), a lightweight and efficient framework that operates exclusively on 1D sequence embeddings, eliminating the need for 3D structures or MSAs. Our core innovation is a purpose-built, dual-path architecture designed to extract multi-scale features from the embeddings. A Local-Hierarchical Path uses specialized local-global attention to identify conserved functional motifs, while a parallel Global-Holistic Path employs standard self-attention to capture the long-range dependencies that define a protein's overall context. By fusing these complementary representations, our model learns a comprehensive feature set for highly accurate property prediction.

Evaluated on a benchmark toxicity prediction task, DEP establishes a new state-of-the-art with an Area Under the Receiver Operating Characteristic Curve (AUC-ROC) of 0.966. This result demonstrates that superior performance is achievable without relying on 3D models or evolutionary alignments, challenging the perceived dependency on computationally intensive tools like AlphaFold. The model's lightweight design directly translates into faster training and inference, removing the computational bottlenecks associated with structure prediction pipelines. This combination of superior accuracy, efficiency, and robustness makes our framework an ideal solution for real-world screening pipelines where speed and predictive power are paramount.[1]

2. Related Works

2.1. Classical Feature-Based Models

Early methods for protein toxicity relied heavily on handcrafted features such as amino acid composition, motif occurrence, and physicochemical descriptors. Tools like BTXpred and NTXpred [Saha and Raghava 2007a, Saha and Raghava 2007b] targeted narrow domains (e.g., bacterial and neurotoxins), using SVMs and BLAST-based filters to identify known toxic profiles. ClanTox [Naamati et al. 2009] expanded to venom proteins, applying boosting over sequence-derived features.

ToxinPred [Gupta et al. 2013] extended this approach to general toxicity prediction using dipeptide composition and motif-based features. Its successors, ToxinPred2 and ToxinPred3 [Sharma et al. 2022, Sharma and Raghava 2024], combined alignment-based similarity (BLAST), motif search (MERCI), and ensemble ML (e.g., Extremely Randomized Trees) with pre-trained embeddings such as ESM-2[Lin et al. 2023]. These models reached AUROC scores near 0.99 but remained dependent on curated training distributions and lacked mechanisms for handling novel or domain-shifted inputs.

[1]The source code and dataset are available at `https://github.com/Buzelin2/DEP`.

2.2. Sequence-Based Deep Learning Models

The limitations of feature-based models led to the adoption of deep learning for end-to-end learning from raw protein sequences. ToxDL [Pan et al. 2020] was among the first to apply convolutional architectures with domain embeddings, showing improved generalization over alignment-based approaches. More recently, CSM-Toxin [Morozov et al. 2023] leveraged fine-tuned ProteinBERT embeddings, achieving strong performance across diverse protein classes (MCC 0.66), though interpretability and structural awareness remained limited.

Newer approaches began exploiting larger pre-trained protein language models (PLMs) such as ESM-2. ToxDL 2.0 [Zhu et al. 2025] integrates ESM-2 representations with AlphaFold2-predicted 3D structures via graph neural networks (GCNs), and adds domain knowledge as input features. The model highlights toxic regions using Integrated Gradients, improving interpretability and generalization. However, it remains a static model, frozen during inference and unable to adapt to subtle distributional shifts.

2.3. Summary and Research Gap

Toxicity prediction has progressed from handcrafted pipelines to deep learning models that integrate structural and evolutionary information. However, most recent approaches depend heavily on AlphaFold2-derived structures or MSAs, resources that are computationally expensive, unavailable for newly sequenced proteins, and impractical at scale. Furthermore, existing models are typically static, once trained, they remain fixed at inference time. This lack of adaptability restricts their ability to handle distributional shifts, novel protein classes, or changes in the underlying biological context, limiting their robustness and generalization in dynamic or previously unseen scenarios.

In contrast, large pretrained protein language models (PLMs) such as ESM-C have demonstrated the ability to encode meaningful functional and structural information directly from 1D sequences, without relying on structural or evolutionary inputs. However, existing approaches have largely treated these embeddings as frozen inputs to static classifiers, failing to explore their full potential in dynamic or adaptive modeling settings.

This highlights a critical research gap: there is a need for lightweight, structure-free models that not only perform competitively but also adapt during inference to distributional shifts and novel input scenarios. To address this, we propose a dual-path architecture that processes frozen ESM-C embeddings through complementary local and global attention mechanisms.

3. Benchmark Dataset for Toxicity Prediction

To ensure a rigorous and directly comparable evaluation of our model, we adopted the pre-existing benchmark dataset for protein toxicity prediction established by recent state-of-the-art studies[Zhu et al. 2025]. Using a standardized benchmark and its prescribed data partitions is critical for academic integrity, as it guarantees that our model is assessed under the exact same conditions as competing methods, enabling a fair and transparent comparison of performance.

The benchmark was constructed to facilitate a robust assessment of model generalization. Sourced from the UniProt database, it incorporates two critical design features.

First, a temporal split separates proteins based on their deposition date (before and after January 1, 2022), creating an independent test set of entirely novel sequences unavailable during the development of prior models. Second, the dataset enforces a maximum sequence identity of 40% between the training and test partitions using CD-HIT-2D. This measure is essential to mitigate information leakage and test the model's ability to learn meaningful biological patterns rather than simply recognizing homologous sequences.

The final dataset structure provided by the benchmark comprises distinct training, validation, and test partitions, with the independent test set containing 152 toxic and 4,710 non-toxic proteins. Our strict adherence to this rigorous, pre-established evaluation protocol is fundamental to the validity of our results. The dataset is also provided on our GitHub repository.

4. Methodology

Our approach to predicting protein properties based on their sequence is centered on a novel deep learning architecture that processes rich vector representations of protein amino acid sequences. The methodology is divided into two phases: first, the generation of contextualized embeddings from amino acid sequences using the pre-trained ESM-C 300M [ESM Team 2024] protein language model, and second, a hierarchical classification model that employs a dual-path, local and global attention mechanism to make a final prediction.

4.1. Input Representation and Pre-processing

The starting point for our analysis is the primary structure of the protein, represented by its amino acid sequence. To capture the complex biological and structural semantics embedded within these sequences, we leverage a transfer learning approach. We use the pre-trained ESM-C 300M protein language model to transform each amino acid sequence into a sequence of high-dimensional embeddings.

Formally, a protein represented by an amino acid sequence S is first tokenized into a sequence of its constituent tokens, $T = (t_0, t_1, ..., t_L)$. In this sequence, t_0 represents the special classification token `[CLS]`, a standard component in Transformer-based models like BERT or ESM. This token is prepended to the input and is specifically designed to act as a summary, aggregating information from the entire sequence into a single vector after processing. The remaining tokens $(t_1, ..., t_L)$ correspond to the L amino acids of the protein. The token sequence T is then processed by the ESM model, represented by Φ_{ESM}, which maps T into a sequence of output vectors H_{raw}:

$$H_{raw} = \Phi_{ESM}(T) \quad , \quad H_{raw} \in \mathbb{R}^{(L+1)\times D}$$

where $D = 960$ is the dimensionality of the ESM embedding space.

To handle the naturally variable lengths of protein sequences and enable efficient, parallelized batch processing on hardware like GPUs, it is necessary to standardize all model inputs into fixed-size tensors. We therefore define a maximum sequence length, N_{seq}, and apply a standard padding strategy. For any given protein, the body of its embedding sequence (all tokens except the initial `[CLS]`) is padded with zero vectors if it is shorter than N_{seq},. This procedure ensures a uniform input shape across all samples in a

batch. The final input tensor X for our model is built by prepending the original `[CLS]` token's embedding to the standardized sequence body. This procedure results in a final tensor X with fixed dimensions:

$$X \in \mathbb{R}^{N \times D}$$

where $N = N_{seq} + 1$ is the standardized total sequence length (including the CLS token). This tensor X, which encapsulates both global protein information (via the CLS token) and detailed sequential features, serves as input to our Dual-Path Transformer Classifier.

4.2. Overall Model Architecture

The proposed architecture, which we term DEP (Dual-path Embeddings for Protein), is engineered to effectively capture features at multiple scales from the input protein embeddings. Fundamentally, DEP can be viewed as a hybrid model, designed to leverage the distinct advantages of both Convolutional Neural Networks (CNNs) and Transformers. It uses convolutional layers in its initial stages to efficiently capture local motifs, while employing Transformer-based attention mechanisms to model long-range dependencies.

It features a dual-path architecture that processes the sequence representation in parallel to extract complementary information:

1. **The Local-Hierarchical Path:** This path operates on the sequence body, denoted as $X_{\text{body}} \in \mathbb{R}^{(N-1) \times D}$, which corresponds to the full input embedding sequence excluding the `[CLS]` token. Its purpose is to model local interactions among amino acids, such as structural motifs (e.g., α-helices, β-sheets) and functional patterns. It hierarchically aggregates these fine-grained features to produce a robust local representation vector, z_{local}.
2. **The Global-Holistic Path:** In parallel, this path processes the complete embedding sequence, $X \in \mathbb{R}^{N \times D}$. It employs a global self-attention mechanism to capture long-range dependencies between all amino acids, which are crucial for defining the protein's overall fold and function. The final state of the initial token from this path serves as the global summary vector, z_{global}.

The output representations from both paths, denoted z_{local} and z_{global}, are subsequently concatenated and fed into a final classification head, which performs the binary prediction (e.g., toxic/non-toxic). This consensus approach allows the model to ground its decision on both detailed local patterns and the broader protein context. Figure 1 provides a schematic overview of the architecture.

4.3. Local-Hierarchical Path

The local-hierarchical path is designed to extract and progressively abstract fine-grained features from the protein's amino acid sequence. This process begins with a convolutional "stem" that captures multi-scale local patterns. The features are then projected into the model's main feature space and processed by a series of transformer blocks that apply our specialized local and global attention mechanism.

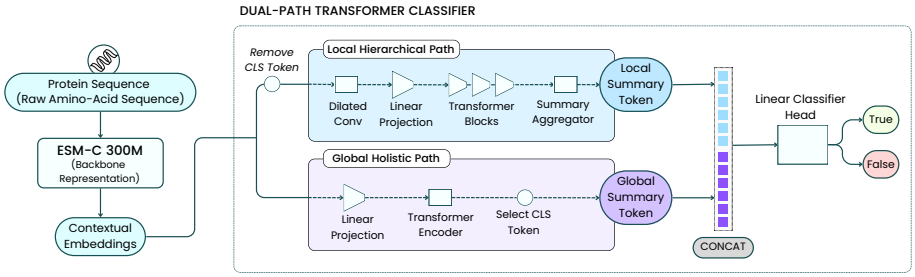

Figure 1. Overall pipeline of the proposed method for protein property prediction. The architecture leverages contextualized embeddings from ESM-C and employs a dual-path Transformer classifier to generate binary predictions.

4.3.1. Initial Feature Extraction: The Convolutional Frontend

The local-hierarchical path begins with a convolutional frontend, a module designed to perform an efficient, preliminary feature extraction. This frontend processes the sequence body $X_{\text{body}} \in \mathbb{R}^{(N-1)\times D}$ by first permuting the tensor to align with the channel-first format expected by 1D convolutional layers. It is then processed by a stack of three `Conv1d` layers with a kernel size of 3 and increasing dilation rates ($d = 1, 2, 4$). Each convolution is followed by a GELU activation function. This structure allows the model to capture features from varying local contexts efficiently.

Let the output of this frontend be X_{features}. These features, which retain the original sequence length, are then projected from the input embedding dimension D to the model's main hidden dimension, D_{model}, via a linear layer:

$$X_{\text{proj}} = X_{\text{features}} W_{\text{proj}} + b_{\text{proj}} \quad , \quad X_{\text{proj}} \in \mathbb{R}^{(N-1)\times D_{\text{model}}}$$

This projected tensor, X_{proj}, serves as the input to the subsequent transformer blocks.

4.3.2. Transformer Block with Local-Global Attention (LGA)

The projected feature tensor, X_{proj}, is then processed by a series of L Transformer Blocks. Each block applies a specialized local-global attention mechanism followed by a feed-forward network, illustrated in Figure 2.

A key feature of this path is its hierarchical nature. Each block progressively reduces the sequence length, creating a more abstract feature representation at every step. Therefore, the sequence length, denoted by M, is not fixed; it shrinks as data flows through the pipeline. Let the input to a given block be $X_{\text{in}} \in \mathbb{R}^{M\times D_{\text{model}}}$. The block will output a tensor with a new, shorter sequence length, M', calculated as $M' = \lfloor \frac{M-w}{s} \rfloor + 1$, where w is the window size and s is the stride. This output tensor then becomes the input for the subsequent block. This progressive summarization is fundamental to how the model shifts its focus from fine-grained details to broader structural patterns.

Within each block, the process begins with a standard layer normalization applied to the input:

$$X_{\text{norm}} = \text{LayerNorm}(X_{\text{in}})$$

The core of each block is the windowed, local-global attention mechanism [Buzelin et al. 2025], which computes queries (Q) from local segments of the sequence, while keys (K) and values (V) are derived from the entire sequence.

Local Windowed Query (Q) Generation. To capture fine-grained local motifs, queries are generated from overlapping windows of the normalized input, X_{norm}. Given a window size w and a stride s, we define a set of M' overlapping windows, where $M' = \lfloor \frac{M-w}{s} \rfloor + 1$. For each window $i \in \{1, \ldots, M'\}$, a segment of the input is passed through a 1D convolution (`Conv1d`) and then averaged along its length to produce a single query vector, $Q^{(i)}$. The collection of these vectors forms the final query tensor:

$$Q = [Q^{(1)}, Q^{(2)}, \ldots, Q^{(M')}] \quad , \quad Q \in \mathbb{R}^{M' \times D_{\text{model}}}$$

Global Key (K) and Value (V) Generation. In contrast, the key and value tensors are computed globally to provide a complete context for the local queries. The entire normalized sequence, X_{norm}, is passed through two separate 1D convolutional layers to produce the key and value tensors:

$$K = \text{Conv1D}_K(X_{\text{norm}}) \quad , \quad K \in \mathbb{R}^{M \times D_{\text{model}}}$$
$$V = \text{Conv1D}_V(X_{\text{norm}}) \quad , \quad V \in \mathbb{R}^{M \times D_{\text{model}}}$$

Attention Computation. We then apply a standard multi-head scaled dot-product attention mechanism. The query, key, and value tensors are split across H heads. For each head h, the output is computed as:

$$\text{Attention}(Q_h, K_h, V_h) = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_h}}\right) V_h$$

where $d_h = D_{\text{model}}/H$. The outputs from all heads are concatenated. A residual connection is added with the original query tensor Q, resulting in the attention output $Y_{\text{attn}} \in \mathbb{R}^{M' \times D_{\text{model}}}$.

Residual Connection and Feed-Forward Network. Due to the reduction in sequence length from M to M', a standard residual connection from X_{in} is not possible. Instead, we create a parallel residual path by applying a max-pooling layer to X_{norm} to match the new sequence length, followed by a `Conv1d` with a kernel size of 1. This is added to the attention output:

$$X_{\text{intermediate}} = Y_{\text{attn}} + \text{Conv1D}(\text{MaxPool1D}(X_{\text{norm}}))$$

Finally, this intermediate representation is passed through a second residual connection consisting of a layer normalization and a position-wise feed-forward network (MLP). The MLP is composed of two linear layers with a GELU activation. To foster hierarchical feature learning, the inner dimensionality of the MLP, D_{MLP}, expands at each block i: $D_{\text{MLP}}^{(i)} = D_{\text{MLP, base}} \times 2^i$.

$$X_{\text{out}} = X_{\text{intermediate}} + \text{MLP}(\text{LayerNorm}(X_{\text{intermediate}}))$$

The final tensor $X_{\text{out}} \in \mathbb{R}^{M' \times D_{\text{model}}}$ serves as the input to the next Transformer Block.

4.3.3. Local Feature Aggregation

After the feature sequence has been processed by the final Transformer Block, the resulting tensor, $X_{\text{final\_local}} \in \mathbb{R}^{M_f \times D_{\text{model}}}$, must be condensed into a single vector representation. For this purpose, we employ a summary aggregation module.

This module introduces a learnable summary token, $t_{\text{summary}} \in \mathbb{R}^{1 \times D_{\text{model}}}$, which is prepended to the sequence $X_{\text{final\_local}}$. The augmented sequence is then passed through a standard `TransformerEncoderLayer`. This final self-attention operation allows the summary token to gather and integrate the most salient information from all the feature vectors in the sequence.

The output state of this summary token is then selected as the definitive feature vector for the local-hierarchical path:

$$z_{\text{local}} = \text{AttentionAggregator}(X_{\text{final\_local}}) \quad , \quad z_{\text{local}} \in \mathbb{R}^{D_{\text{model}}}$$

This vector, z_{local}, encapsulates the most important hierarchical and local-contextual information discovered by this path.

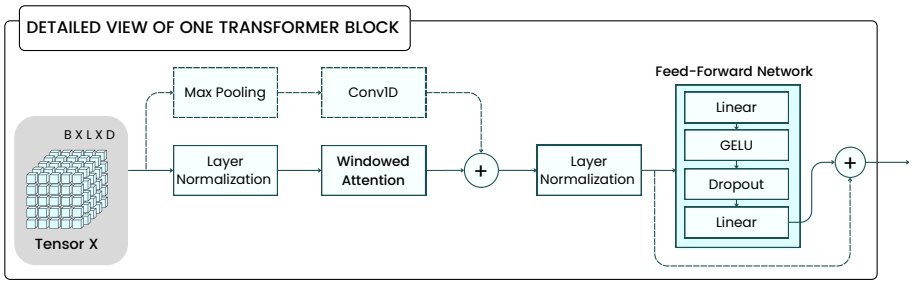

Figure 2. Detailed architecture of a single Transformer Block, featuring Windowed Attention, residual connections, and an MLP.

4.4. The Global-Holistic Path

Operating in parallel to the local path, the global-holistic path is responsible for capturing long-range dependencies and forming a holistic understanding of the entire protein. Unlike the local path, it processes the complete input tensor $X \in \mathbb{R}^{N \times D}$, including the `[CLS]` token.

The process begins with a linear projection that maps the input features from the original embedding dimension D to the model's hidden dimension, D_{model}:

$$X'_{\text{global}} = XW_{\text{global\_proj}} + b_{\text{global\_proj}}$$

The resulting tensor, X'_{global}, is then fed into a standard `TransformerEncoder`. This encoder is composed of a stack of L_{global} identical layers, where each layer performs global self-attention followed by a position-wise feed-forward network. This allows every token in the sequence to directly attend to every other token, capturing the overall protein context.

Let the output of the Transformer Encoder be Y_{global}. A final layer normalization is applied:

$$Y'_{\text{global}} = \text{LayerNorm}(Y_{\text{global}})$$

The definitive representation for the global path, $z_{\text{global}} \in \mathbb{R}^{D_{\text{model}}}$, is then extracted by taking the final hidden state corresponding to the first token of the sequence (the original `[CLS]` token). This vector serves as a comprehensive summary of the protein's global properties.

4.5. Final Classification Module

The final stage of the architecture integrates the specialized representations learned by the two parallel paths to produce a single prediction. This is achieved through a combination mechanism that combines the local and global feature vectors.

The two output vectors, z_{local} from the local-hierarchical path and z_{global} from the global-holistic path, are first concatenated to form a final, comprehensive feature vector, z_{final}:

$$z_{\text{final}} = [z_{\text{local}}; z_{\text{global}}] \quad , \quad z_{\text{final}} \in \mathbb{R}^{2 \times D_{\text{model}}}$$

This concatenated vector leverages both the fine-grained, contextualized motifs and the holistic properties of the protein.

Finally, z_{final} is passed through a simple linear layer that acts as the classification head, mapping the high-dimensional feature vector to a single logit. This logit is then passed through a sigmoid function, σ, to yield the final probability score $p \in [0, 1]$, which represents the model's confidence in the property being predicted (e.g., toxicity).

$$p = \sigma(W_{\text{classifier}} z_{\text{final}} + b_{\text{classifier}})$$

4.6. Experimental Settings

All experiments were run on a single NVIDIA RTX 4090 GPU. Models were trained for 10 epochs using the AdamW optimizer (initial learning rate $= 8 \times 10^{-5}$, with the PyTorch default weight decay of 10^{-2}). We used a batch size of 16 protein sequences and applied a sliding window of length 50 with a stride of 4.

5. Experiments and Results

For a rigorous comparison, we adopt the exact training, validation, and test splits from the current state-of-the-art, ToxDL 2.0 [Zhu et al. 2025]. We note that the ToxDL 2.0 study reports threshold-dependent metrics (e.g., F1-score) using a threshold optimized on its test set—a practice known to inflate performance and hinder fair comparison. Consequently, we prioritize the threshold-independent Area Under the Receiver Operating Characteristic Curve (AUC-ROC) for our primary evaluation. As shown in Table 1, our model sets a new state-of-the-art with an AUC-ROC of 0.966, substantially outperforming ToxDL 2.0 (0.945). Crucially, this result is achieved using only 1D sequence embeddings, foregoing the need for computationally expensive 3D structures or multiple sequence alignments.

Table 1. Performance comparison (AUC-ROC) of our model against prior methods on the independent test set.

Method	AUC-ROC
ToxinPred2 (RF)	0.864
ToxinPred 3.0 (ET)	0.822
ToxinPred 3.0 (Hybrid)	0.833
CSM-Toxin	0.873
tAMPer	0.857
ToxDL	0.888
ToxDL2	0.945
Our Model	**0.966**

To further assess practical performance, we evaluate threshold-dependent metrics. For a fair comparison, the classification threshold for our model and ToxDL 2.0 was optimized strictly on the validation set. The results, presented in Table 2, show that DEP achieves a superior F1-score (0.626 vs. 0.606). This improvement is driven by a gain in precision (0.516 vs. 0.492) while maintaining a competitive recall (0.798 vs. 0.790).

Table 2. Performance comparison between ToxDL2 and DEP across multiple metrics.

Model	Precision	Recall	F1-score
ToxDL2	0.4918	0.7895	0.6061
DEP	0.5158	0.7976	0.6262

5.1. Ablation Study

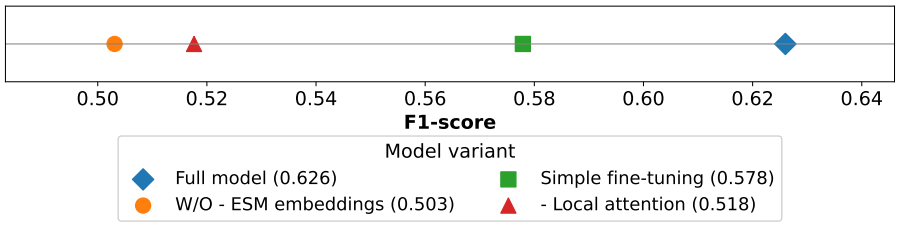

Figure 3. One-dimensional scatter plot of F1-scores for the full model and three ablation variants.

To quantify the contribution of each key component, we trained four model variants *de novo*, using identical hyper-parameters and data splits to those detailed in Section 4. Their F1-scores on the independent test set are shown in Figure 3.

The complete dual-path architecture, which couples frozen ESM-C embeddings with Local–Global Attention (LGA), attains an F1 of **0.626**. When the pretrained embeddings are replaced by a randomly initialised, learnable token matrix, performance drops sharply to 0.503, a relative loss of 20%. Collapsing the network to a single global transformer encoder and fine-tuning *all* ESM-C parameters recovers part of the gap, reaching an F1 of 0.578; nevertheless, this remains 8% below the full system, indicating that

parameter-efficient adaptation combined with our dual-path design is more effective than a monolithic fine-tuning strategy. Finally, removing the hierarchical LGA blocks while retaining global self-attention yields an F1 of 0.518, highlighting the essential role of the local path in capturing short-range motifs that drive toxic activity.

6. Conclusion

We introduced a novel dual-path architecture for protein toxicity prediction, engineered to extract multi-scale features from static embeddings. Our core architectural innovation is a Local-Hierarchical Path that performs progressive feature abstraction using a specialized Local-Global Attention (LGA) mechanism, designed to identify conserved functional motifs. In parallel, a Global-Holistic Path employs standard self-attention to capture the long-range dependencies that define a protein's overall context. The fusion of these complementary representations provides a robust foundation for toxicity classification.

Our model establishes a new state-of-the-art on the benchmark toxicity dataset, outperforming recent methods like ToxDL 2.0 that depend on AlphaFold2-generated 3D structures and evolutionary alignments. This result is a central contribution of our work, demonstrating that superior predictive performance can be achieved without any reliance on computationally expensive 3D structure predictions or multiple sequence alignments. We conclude that the synergy between our local path, adept at isolating toxic motifs, and our global path, which contextualizes these motifs within the full protein, is sufficiently powerful to extract the necessary predictive signals from 1D sequence embeddings alone.

The lightweight, sequence-only nature of our model makes it not only highly accurate but also significantly more efficient and readily deployable for the large-scale screening of protein candidates, where invoking structural prediction pipelines would be a major bottleneck. Future work will focus on extending this successful architecture to other property prediction tasks to test its generalizability and on leveraging attention weights to enhance the biological interpretability of its predictions. Ultimately, our work presents a powerful paradigm: designing sophisticated, purpose-built architectures is a highly effective strategy for unlocking the full potential of protein language model embeddings, challenging the prevailing notion that complex structural inputs are a prerequisite for top-tier performance.

References

Bento, P., Aquino, Y., Buzelin, A., Rigueira, P. B., Gambogi, A., Porfírio, L. G., Doria, I., Anunciação, S., Mendes, G., Minardi, R., Paim, A. A., Pappa, G. L., da Fonseca, F., and Meira Jr., W. (2025). A machine learning-guided approach for a multi-epitope hiv vaccine design. *Revista Eletrônica de Iniciação Científica em Computação*, 23(1):118–123.

Buzelin, A., Dutenhefner, P. R., Rezende, T., Porfirio, L. G., Bento, P., Aquino, Y., Fernandes, J., Santana, C., Miana, G., Pappa, G. L., Ribeiro, A., and Jr, W. M. (2025). A cnn-based local-global self-attention via averaged window embeddings for hierarchical ecg analysis.

ESM Team (2024). Esm cambrian: Revealing the mysteries of proteins with unsupervised learning.

Gupta, S., Kapoor, P., Chaudhary, K., Gautam, A., and Kumar, R. G. P. S. (2013). Toxinpred: a web server for the prediction of toxic peptides and proteins. *Nucleic Acids Research*, 41(W1):W196–W203.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130.

Morozov, V., Rodrigues, C. H. M., and Ascher, D. B. (2023). Csm-toxin: A web-server for predicting protein toxicity. *Pharmaceutics*, 15(2):431.

Naamati, G., Winter, E., and Linial, M. (2009). Clantox: a classifier of animal toxins. *Nucleic Acids Research*, 37(Web Server issue):W602–W607.

Pan, X., Zuallaert, J., Wang, X., Shen, H.-B., Campos, E. P., Marushchak, D. O., and Neve, W. D. (2020). Toxdl: deep learning using primary structure and domain embeddings for assessing protein toxicity. *Bioinformatics*, 36(21):5159–5168.

Rappuoli, R., Mandl, C. W., Black, S., and Gregorio, E. D. (2011). Vaccines for the twenty-first century society. *Nature Reviews Immunology*, 11(12):865–872.

Saha, S. and Raghava, G. P. (2007a). Btxpred: Support vector machine-based method for predicting bacterial toxins. *BMC Bioinformatics*, 8:463.

Saha, S. and Raghava, G. P. (2007b). Ntxpred: A svm-based method for predicting neurotoxins. *BMC Bioinformatics*, 8:463.

Sharma, N., Devi, N. L., Jain, S., and Raghava, G. P. (2022). Toxinpred2: an improved method for predicting toxicity of proteins. *Briefings in Bioinformatics*, 23(5):bbac174.

Sharma, N. and Raghava, G. P. (2024). Toxinpred 3.0: A deep learning-based model for peptide and protein toxicity prediction. Manuscript accessed via Elsevier; exact citation pending journal confirmation.

Zhu, L., Fang, Y., Liu, S., Shen, H.-B., Neve, W. D., and Pan, X. (2025). Toxdl 2.0: Protein toxicity prediction using a pretrained language model and graph neural networks. *Computational and Structural Biotechnology Journal*, 27:1538–1549.