Báo cáo bài tập nhóm - CAFA6

Nguyễn Minh Hiển
Đại Học Công Nghệ
Đại Học Quốc Gia Hà Nội
Hà Nội, Việt Nam
22021106@vnu.edu.vn

Nguyễn Đức Hùng
Đại Học Công Nghệ
Đại Học Quốc Gia Hà Nội
Hà Nội, Việt Nam
22021109@vnu.edu.vn

Lôi Đình Nhất
Đại Học Công Nghệ
Đại Học Quốc Gia Hà Nội
Hà Nội, Việt Nam
22021152@vnu.edu.vn

***Abstract*—Dự đoán chức năng protein từ trình tự amino acid là một bài toán cơ bản và thiết yếu trong sinh học và y học. Bài toán này đặt ra những thách thức đáng kể do tính chất phức tạp của dữ liệu. Cụ thể, nó yêu cầu một mô hình có khả năng mở rộng để xử lý $31,466$ khái niệm Gene Ontology (GO) tiềm năng, nơi mỗi protein có thể được gán từ 1 đến hàng trăm chức năng khác nhau. Thách thức thứ hai là mất cân bằng dữ liệu cực độ, với số lượng các GO terms tuân theo phân bố lũy thừa, dẫn đến tỷ lệ mẫu dương tính và âm tính có thể đạt $1:100$ hoặc cao hơn. Hơn nữa, cấu trúc ontology của các thuật ngữ GO được sắp xếp theo Directed Acyclic Graph (DAG), đòi hỏi mô hình phải tôn trọng mối quan hệ cha-con. Cuối cùng, việc đánh giá sử dụng $\mathbf{F}_{\max}$ score trong cuộc thi CAFA-6 thay vì metrics thông thường yêu cầu tối ưu hóa ngưỡng threshold động cho từng ontology. Báo cáo này trình bày phương pháp tiếp cận đa mô hình (multi-modal) dựa trên Protein Language Models để giải quyết các thách thức trên. Mã nguồn: https://github.com/nmhienbn/CAFA-6**

***Index Terms*—Gene Ontology, Multi-modal, Protein Language Models**

I. Giới thiệu

Trong kỷ nguyên của sinh học phân tử và dữ liệu lớn (Big Data), protein được xem là những cỗ máy phân tử đảm nhiệm hầu hết các chức năng quan trọng của sự sống. Với sự phát triển vượt bậc của các công nghệ giải trình tự gen thế hệ mới (Next-Generation Sequencing - NGS), số lượng trình tự protein được giải mã đã tăng trưởng theo cấp số nhân, đạt hàng trăm triệu trình tự trong cơ sở dữ liệu UniProtKB. Tuy nhiên, việc xác định chức năng của các protein này bằng các phương pháp thực nghiệm truyền thống (*wet-lab*) lại tốn kém chi phí và mất rất nhiều thời gian. Sự chênh lệch này tạo ra một "khoảng trống kiến thức" (*knowledge gap*) ngày càng lớn, khiến việc khai thác toàn bộ tiềm năng của dữ liệu sinh học trở nên khó khăn. Để giải quyết vấn đề này, các phương pháp tính toán, đặc biệt là Học máy và Học sâu, đã trở thành công cụ thiết yếu để dự đoán chức năng protein tự động với độ chính xác cao.

Trong ngữ cảnh này, dự đoán chức năng protein là một bài toán phân loại đa nhãn cực lớn với nhãn được định nghĩa bởi hệ thống Gene Ontology (GO), bao gồm các khía cạnh chức năng phân tử (MF), Quá trình sinh học (BP), và thành phần tế bào (CC). Bài toán này đặt ra nhiều thách thức do không gian nhãn khổng lồ (hơn 30,000 loại), sự mất cân bằng dữ liệu nghiêm trọng theo quy luật Long-tail và cấu trúc phân cấp phức tạp của nhãn.

Báo cáo này tập trung vào giải quyết bài toán thông qua cuộc thi **Critical Assessment of protein Function Annotation (CAFA 6)** được tổ chức trên nền tảng **Kaggle**. CAFA là cuộc thi đánh giá khả năng dự đoán chức năng protein chính xác và khách quan nhất trong cộng đồng tin sinh học. Dự án của nhóm áp dụng các kỹ thuật tiên tiến, bao gồm việc sử dụng mô hình ngôn ngữ Protein (Protein Language Models) để trích xuất feature nhằm cải thiện đáng kể độ chính xác dự đoán.

II. Premilinaries

Bài toán dự đoán chức năng protein được định nghĩa như một bài toán **phân loại đa nhãn (multi-label classification)** trên không gian Gene Ontology:

Cho tập dữ liệu huấn luyện:

$$D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{train}}}$$

trong đó:

- $x_i \in \mathbb{R}^{d_m}$: embedding đa phương thức của protein i
- $y_i \in \{0,1\}^{|\mathcal{G}|}$: vector nhãn multi-hot encoding
- $d_m = 3,072$ (tổng chiều từ 3 embedding sources)
- $N_{\text{train}} = 82,404$ proteins
- $|\mathcal{G}| \in \{2,975; 8,245; 10,246\}$: số GO terms (tùy thuộc ontology)

Tập dữ liệu kiểm tra (không nhãn)

$$D_{\text{test}} = \{x_j\}_{j=1}^{N_{\text{test}}}$$

với $N_{\text{test}} = 224,309$ proteins

Mục tiêu: Học hàm ánh xạ $f_\theta : \mathbb{R}^{d_m} \to \mathbb{R}^{|\mathcal{G}|}$ tối ưu hóa:

$$\theta^* = \arg\max_\theta \mathbb{E}_{(x,y)\sim D_{\text{OOF}}} \left[F\text{-max}(\boldsymbol{y}, \sigma(f_\theta(x)))\right]$$

trong đó:

- $\sigma(\cdot)$ là hàm sigmoid (element-wise)
- $F\text{-}max$ là metric đánh giá được định nghĩa bởi CAFA

III. Xử lý dữ liệu và xây dựng đặc trưng

Trong phần này, nhóm trình bày chi tiết quy trình xử lý dữ liệu và các chiến lược tạo đặc trưng được sử dụng trong bài toán dự đoán chức năng protein của cuộc thi CAFA 6. Do đặc thù của bài toán là phân loại đa nhãn với không gian nhãn rất lớn và phân bố nhãn mất cân bằng nghiêm trọng, việc thiết kế đặc trưng đóng vai trò then chốt trong việc nâng cao hiệu năng của mô hình.

A. Đặc trưng từ Sequence Embeddings

Đặc trưng từ embedding chuỗi protein là nguồn thông tin quan trọng nhất trong hệ thống. Chúng tôi tận dụng các mô hình Ngôn ngữ Protein (Protein Language Models – PLMs) hiện đại để mã hóa chuỗi amino acid thành các vector đặc trưng có khả năng biểu diễn thông tin về cấu trúc, tiến hóa và chức năng sinh học.

1) Các mô hình tiền huấn luyện: Trong quá trình thực nghiệm, chúng tôi thử nghiệm và đánh giá nhiều mô hình PLM khác nhau, bao gồm:

- **ESM2** (Meta AI) [1]:
  - `esm2_t48_15B_UR50D` (15 tỷ tham số),
  - `esm2_t36_3B_UR50D` (3 tỷ tham số),
  - `esm2_t33_650M_UR50D` (650 triệu tham số).
- **ESM1b [2]:** `esm1b_t33_650M_UR50S`.
- **Ankh** [3] (ElnaggarLab): `ankh-large`, `ankh3-large`.
- **ProtT5** [4] (Rostlab): `prot_t5_xl_uniref50`.
- **ProtBERT** [5] (Rostlab): `prot_bert_bfd`.

Kết quả thực nghiệm cho thấy việc kết hợp **ESM2-650M hoặc ESM2-3B** với **ProtT5-XL** mang lại hiệu quả dự đoán tốt nhất trong khi vẫn đảm bảo chi phí tính toán hợp lý.

2) Phương pháp trích xuất embedding: Cho một chuỗi protein gồm L amino acid:

$$S = (a_1, a_2, \ldots, a_L)$$

Sau khi đưa qua mô hình PLM, ta thu được chuỗi vector ẩn ở lớp cuối:

$$\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L), \quad \mathbf{h}_i \in \mathbb{R}^d$$

Chúng tôi sử dụng kỹ thuật **mean pooling** để thu được embedding cố định chiều:

$$\mathbf{e} = \frac{1}{L} \sum_{i=1}^{L} \mathbf{h}_i$$

Toàn bộ quá trình suy luận embedding được thực hiện với độ chính xác **FP16** nhằm giảm tiêu thụ bộ nhớ GPU và tăng tốc độ xử lý, đặc biệt quan trọng đối với các mô hình quy mô lớn như ESM2-15B.

B. Đặc trưng Taxonomy

Thông tin phân loại sinh học (taxonomy) cung cấp ngữ cảnh tiến hóa quan trọng, giúp mô hình phân biệt chức năng protein giữa các nhóm sinh vật khác nhau.

1) Đặc trưng loài (Species-level Features): Mỗi protein được gán một `NCBI Taxonomy ID`, thu thập từ tập huấn luyện hoặc từ phần header của file FASTA trong tập test. Các TaxID được xử lý như sau:

- Mã hóa bằng One-hot Encoding hoặc Label Encoding,
- Các loài hiếm (có tần suất thấp hơn ngưỡng xác định) được gộp vào nhóm *unknown*,
- Chỉ giữ lại top 30–50 loài phổ biến nhất để kiểm soát số chiều đặc trưng.

2) Đặc trưng phân cấp tiến hóa (Hierarchical Lineage): Để khai thác mối quan hệ tiến hóa giữa các loài, chúng tôi sử dụng công cụ `taxonkit` kết hợp với dữ liệu NCBI taxdump để trích xuất phả hệ đầy đủ gồm bảy cấp:

$$\begin{aligned}\text{Superkingdom} \rightarrow \text{Phylum} \rightarrow \text{Class} \rightarrow \text{Order} \rightarrow \text{Family} \\ \rightarrow \text{Genus} \rightarrow \text{Species}\end{aligned} \tag{1}$$

Trong đó, các cấp **Order**, **Family** và **Genus** được mã hóa riêng biệt bằng One-hot Encoding, giúp mô hình học được sự tương đồng chức năng giữa các protein có quan hệ tiến hóa gần.

C. Đặc trưng Protein–Protein Interaction (PPI)

Các tương tác protein–protein cung cấp thông tin chức năng bổ sung, phản ánh vai trò của protein trong mạng lưới sinh học.

1) Nguồn dữ liệu: Dữ liệu PPI được thu thập từ cơ sở dữ liệu **STRING phiên bản 12**, giới hạn ở các loài xuất hiện trong bộ dữ liệu CAFA 6. Việc ánh xạ từ STRING ID sang CAFA Protein ID được thực hiện thông qua UniProt Accession.

2) Đặc trưng đồ thị: Từ đồ thị PPI, chúng tôi trích xuất các đặc trưng mạng lưới sau:

- Degree Centrality,
- Betweenness Centrality,
- Closeness Centrality,
- PageRank,
- Node2Vec embeddings học từ các bước đi ngẫu nhiên trên đồ thị.

Các đặc trưng này giúp mô hình nắm bắt cả vai trò cục bộ lẫn tầm quan trọng toàn cục của protein trong mạng tương tác.

D. Đặc trưng cấu trúc và chú thích

1) Thông tin cấu trúc 3D: Cấu trúc không gian ba chiều của protein mang nhiều thông tin quan trọng liên quan trực tiếp đến chức năng. Tuy nhiên, việc khai thác dữ liệu từ các công cụ như AlphaFold hoặc FoldSeek ở quy mô lớn đòi hỏi chi phí tính toán rất cao. Do giới hạn về tài nguyên, các đặc trưng này chưa được tích hợp trong phiên bản hiện tại của hệ thống.

2) BLAST: BLAST được sử dụng để tìm các trình tự tương đồng nhằm suy luận chức năng gián tiếp. Tuy nhiên, qua thực nghiệm, các đặc trưng dựa trên BLAST không mang lại cải thiện đáng kể so với sequence embeddings, nên không được sử dụng trong mô hình cuối cùng.

E. Đặc trưng từ UniProtKB

UniProtKB cung cấp các annotation đã được kiểm chứng, bao gồm:

- Chú thích chức năng (GO terms, enzyme classification),
- Đặc trưng trình tự (domain, motif),
- Tính chất lý hóa (khối lượng phân tử, điểm đẳng điện).

Các đặc trưng này được mã hóa bằng One-hot hoặc TF–IDF. Kết quả cho thấy UniProtKB giúp cải thiện rõ rệt hiệu suất đối với các GO terms hiếm, đặc biệt khi kết hợp với sequence embeddings.

F. Chiến lược kết hợp đặc trưng

Cấu hình mang lại hiệu quả tốt nhất bao gồm:

1) Sequence embeddings (kết hợp ESM2-3B và ProtT5-XL),
2) Đặc trưng taxonomy phân cấp (Order, Family, Genus),
3) Chú thích chức năng từ UniProtKB.

IV. Phương pháp

Nhóm đề xuất kiến trúc ensemble lai kết hợp giữa Mạng Nơ-ron (Deep Neural Network) và Gradient Boosting. Kết quả dự đoán cuối cùng được tối ưu hóa thông qua chiến lược hợp nhất có trọng số, tận dụng tính bổ trợ giữa hai dòng mô hình khác biệt.

A. Trích xuất feature để huấn luyện

Mỗi protein P được biểu diễn bởi một tập hợp các vector nhúng được trích xuất từ ba Mô hình Ngôn ngữ Protein (pLMs) tiền huấn luyện khác nhau. Gọi X là biểu diễn đặc trưng của protein:

$$X = \{\mathbf{x}_{ankh}, \mathbf{x}_{esm}, \mathbf{x}_{t5}\} \tag{2}$$

Trong đó:

- $\mathbf{x}_{ankh} \in \mathbb{R}^{768}$: Vector từ mô hình Ankh.
- $\mathbf{x}_{esm} \in \mathbb{R}^{1280}$: Vector từ ESM-2.
- $\mathbf{x}_{t5} \in \mathbb{R}^{1024}$: Vector từ ProtT5.

B. Xử lý nhãn

Để giảm thiểu nhiễu và tập trung vào các nhãn quan trọng, nhóm thiết kế thuật toán chọn lọc nhãn dựa trên Information Accretion (IA).

1) Thuật toán Chọn lọc Nhãn: Điểm số ưu tiên $S(t)$ cho mỗi nhãn GO t được tính toán như sau:

$$S(t) = \text{Freq}(t) \times IA(t) \tag{3}$$

Trong đó $\text{Freq}(t)$ là tần suất của nhãn t và $IA(t) = -\log_2(P(t|Pa(t)))$ là lượng tin mà nhãn t cung cấp so với nhãn cha $Pa(t)$. Tập nhãn mục tiêu $\mathcal{T}_{target}$ được chọn gồm $K = 10,000$ nhãn có điểm số cao nhất:

$$\mathcal{T}_{target} = \{t \in \mathcal{T} \mid \text{rank}(S(t)) \leq K\} \tag{4}$$

File IA của các nhãn được ban tổ chức cung cấp.

2) Làm trơn Nhãn (Label Smoothing): Nhóm áp dụng kỹ thuật làm trơn nhãn với $\epsilon = 0.1$ để điều chỉnh vector nhãn mục tiêu $\mathbf{y}$:

$$\mathbf{y}_{smooth} = (1 - \epsilon) \cdot \mathbf{y} + \frac{\epsilon}{K} \tag{5}$$

Kỹ thuật này giúp mô hình giảm sự tự tin quá mức vào các nhãn bị nhiễu.

C. Kiến trúc mạng nơ-ron

1) Khối Mã hóa (Encoder Block): Mỗi vector đặc trưng đầu vào $\mathbf{x}_j$ (với $j \in \{\text{Ankh}, \text{ESM}, \text{ProtT5}\}$) đi qua một Khối Mã hóa riêng biệt. Mục tiêu của khối này là điều chỉnh và chiếu các đặc trưng từ không gian ban đầu về một không gian tiềm ẩn có chiều thấp hơn ($\mathbb{R}^{512}$), đồng thời cân bằng sự khác biệt về phân phối giữa các nguồn.

a) Chuẩn hóa Lớp Đầu vào: Bước đầu tiên là áp dụng Layer Normalization để cân bằng biên độ (amplitude) của các embedding. Điều này đặc biệt quan trọng khi kết hợp embedding của các pLMs có phân phối giá trị khác nhau.

$$\hat{\mathbf{x}}_j = \frac{\mathbf{x}_j - \mu_j}{\sigma_j + \delta} \cdot \gamma + \beta \tag{6}$$

Trong đó μ_j và σ_j là giá trị trung bình và độ lệch chuẩn được tính trên toàn bộ chiều của vector $\mathbf{x}_j$, còn γ và β là các tham số học được (scale và bias).

b) Biến đổi Tuyến tính: Sau khi chuẩn hóa, $\hat{\mathbf{x}}_j$ tiếp tục được đưa qua mạng nơ-ron 3 lớp (với số chiều lần lượt là 2048, 1024, và 512) để trích xuất các đặc trưng bậc cao. Quá trình tính toán tại mỗi bước l tuân theo công thức:

$$\mathbf{h}^{(l)} = \text{Dropout}(\text{GELU}(\text{BN}(\mathbf{W}^{(l)} \cdot \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)})))$$

với $\mathbf{h}^{(0)} = \hat{\mathbf{x}}_j$.

- $\mathbf{W}^{(l)} \cdot \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}$: Phép chiếu tuyến tính (Linear Projection) từ chiều d_{l-1} sang d_l.
- $\text{BN}(\cdot)$: Batch Normalization được sử dụng để ổn định quá trình huấn luyện bằng cách giữ cho đầu ra của mỗi lớp có phân phối ổn định.
- $\text{GELU}(\cdot)$: Hàm kích hoạt Gaussian Error Linear Unit được chọn:
$$\text{GELU}(x) = x \cdot \Phi(x) \tag{7}$$
với $\Phi(x)$ là hàm phân phối tích lũy Gauss chuẩn.
- $\text{Dropout}(\cdot)$: Với xác suất dropout_rate $= 0.4$ để giảm thiểu overfitting.

Đầu ra cuối cùng của Khối Mã hóa là $\mathbf{h}_j = \mathbf{h}^{(L)}$, với chiều là 512.

2) Hợp nhất Đặc trưng và SE-Block: Sau khi các vector đặc trưng từ các nhánh mã hóa ($\mathbf{h}_{ankh}, \mathbf{h}_{esm}, \mathbf{h}_{t5}$) được tạo ra, chúng được hợp nhất bằng phép nối (concatenation) để tạo thành vector tổng hợp $\mathbf{H}_{fused} \in \mathbb{R}^{D_{total}}$:

$$\mathbf{H}_{fused} = \mathbf{h}_{ankh} \oplus \mathbf{h}_{esm} \oplus \mathbf{h}_{t5} \tag{8}$$

với D_{total} là tổng số chiều đầu ra của các Encoder Block (ở đây là $3 \times 512 = 1536$).

Để xử lý nhiễu và bất đồng nhất thông tin giữa các nguồn, nhóm sử dụng khối Squeeze-and-Excitation (SE-Block). SE-Block bao gồm hai giai đoạn:

a) Excitation (Kích thích): Giai đoạn này học một vector trọng số $\mathbf{z} \in \mathbb{R}^{D_{total}}$ thông qua một mạng nơ-ronvới cấu trúc nút cổ chai (bottleneck architecture) với tỷ lệ giảm chiều $r = 4$:

$$\mathbf{z} = \sigma\left(\mathbf{W}_2 \cdot \text{ReLU}\left(\mathbf{W}_1 \cdot \mathbf{H}_{fused}\right)\right) \tag{9}$$

Trong đó:

- $\mathbf{W}_1 \in \mathbb{R}^{\frac{D_{total}}{r} \times D_{total}}$: Ma trận trọng số thực hiện phép giảm chiều.
- $\mathbf{W}_2 \in \mathbb{R}^{D_{total} \times \frac{D_{total}}{r}}$: Ma trận trọng số thực hiện phép khôi phục chiều.
- $\text{ReLU}(\cdot)$: Hàm kích hoạt.
- $\sigma(\cdot)$: Hàm Sigmoid để tạo ra các trọng số $\mathbf{z}$ nằm trong khoảng $[0, 1]$.

Vector $\mathbf{z}$ biểu thị mức độ quan trọng của từng đặc trưng trong $\mathbf{H}_{fused}$.

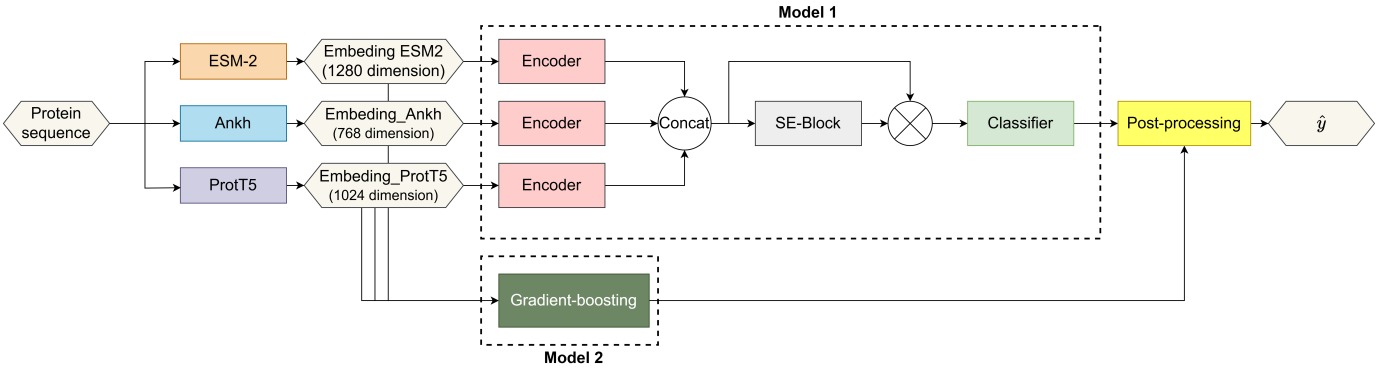

Fig. 1. Sơ đồ Pipeline Đa phương thức và Hợp nhất. Quá trình xử lý bắt đầu bằng việc trích xuất các vector nhúng đa phương thức từ ba mô hình ngôn ngữ protein hàng đầu (ESM-2, Ankh, ProtT5). Các vector này được chuẩn hóa và thu nhỏ chiều qua các Encoder riêng biệt, sau đó được hợp nhất. Khối SE-Block đóng vai trò là cơ chế chú ý thích nghi, hiệu chỉnh các kênh đặc trưng trước khi đưa vào Classifier. Kết quả của luồng Học sâu này được tinh chỉnh và sau đó hợp nhất (Ensemble) với đầu ra của luồng Học máy độc lập, sử dụng mô hình Gradient Boosting, nhằm tận dụng sự đa dạng của mô hình và tối ưu hóa kết quả dự đoán chức năng protein ($\hat{\mathbf{y}}$) cuối cùng.

b) Scale (Hiệu chỉnh): Dựa trên Vector trọng số $\mathbf{z}$ thu được, vector đặc trưng tổng hợp ban đầu được tái định trọng số thông qua phép nhân theo phần tử (element-wise multiplication) để tạo ra biểu diễn tinh chỉnh $\mathbf{H}_{refined}$:

$$\mathbf{H}_{refined} = \mathbf{H}_{fused} \odot \mathbf{z} \tag{10}$$

Phép nhân element-wise ($\odot$) này hoạt động như một bộ lọc thông minh, gcho phép mô hình nhấn mạnh các đặc trưng mang thông tin ngữ nghĩa quan trọng ($\mathbf{z}_i \approx 1$) và đồng thời ức chế các đặc trưng nhiễu hoặc dư thừa ($\mathbf{z}_i \approx 0$).

3) Đầu Phân loại (Classification Head): Sau khi được tinh chỉnh bởi khối SE-Block, vector đặc trưng $\mathbf{H}_{refined}$ được đưa vào module phân loại (Classification Head). Module này được thiết kế dưới dạng mạng nơ-ron đa tầng (MLP) nhằm ánh xạ các đặc trưng tổng hợp vào không gian nhãn mục tiêu $\mathcal{T}_{target}$. Kiến trúc cụ thể bao gồm các giai đoạn xử lý sau:

a) Chuẩn hóa Đầu vào: Để đảm bảo tính ổn định của phân phối dữ liệu trước khi đi vào các tầng ẩn, vector đầu vào $\mathbf{H}_{refined} \in \mathbb{R}^{D_{total}}$ được xử lý qua lớp Batch Normalization 1D:

$$\mathbf{z}^{(0)} = \text{BN}_{1D}(\mathbf{H}_{refined}) \tag{11}$$

b) Các Tầng Ẩn (Hidden Layers): Mạng sử dụng hai lớp ẩn liên tiếp với kích thước cố định $d = 512$ nhằm tăng cường khả năng mô hình hóa các mối quan hệ phi tuyến phức tạp. Tại mỗi lớp l (với $l \in \{1, 2\}$), quá trình biến đổi đặc trưng được thực hiện như sau:

$$\mathbf{z}^{(l)} = \text{Dropout}\left(\text{GELU}\left(\mathbf{W}^{(l)} \cdot \mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}\right)\right) \tag{12}$$

Cấu trúc này thực hiện phép nén chiều dữ liệu theo lộ trình $\mathbb{R}^{D_{total}} \to \mathbb{R}^{512} \to \mathbb{R}^{512}$, giúp cô đọng các thông tin ngữ nghĩa bậc cao trước khi phân loại.

c) Tầng Đầu ra (Output Layer): Lớp cuối cùng thực hiện phép chiếu tuyến tính từ không gian ẩn sang không gian nhãn có kích thước $K = |\mathcal{T}_{target}|$, tạo ra vector logits $\hat{\mathbf{y}} \in \mathbb{R}^K$:

$$\hat{\mathbf{y}} = \mathbf{W}_{out} \cdot \mathbf{z}^{(2)} + \mathbf{b}_{out} \tag{13}$$

Trong giai đoạn huấn luyện và suy luận, vector $\hat{\mathbf{y}}$ sẽ được chuyển đổi thành xác suất dự đoán thông qua hàm kích hoạt Sigmoid: $P = \sigma(\hat{\mathbf{y}})$.

4) Hàm mất mát và Huấn luyện: Để giải quyết vấn đề mất cân bằng lớp nghiêm trọng (Imbalance Data Problem) trong bài toán phân loại đa nhãn (Multi-label Classification), nhóm sử dụng Hàm mất mát Weighted Binary Cross-Entropy with Logits Loss ($\mathcal{L}_{\text{WBCE}}$). Hàm này kết hợp trọng số dương w_{pos} được tính dựa trên Information Accretion của từng nhãn GO, giúp tăng cường sự chú ý của mô hình đối với các nhãn hiếm và quan trọng.

$$\mathcal{L}_{\text{WBCE}} = [w_{pos,c} \cdot y_{ic} \log(\sigma(\hat{y}_{ic})) + (1 - y_{ic}) \log(1 - \sigma(\hat{y}_{ic}))] \tag{14}$$

Trong đó $\sigma(\cdot)$ là hàm Sigmoid, y_{ic} là nhãn mục tiêu, và $\hat{y}_{ic}$ là Logit dự đoán.

5) Thông số Huấn luyện và Điều chỉnh: Mô hình được huấn luyện trên các tham số được tối ưu hóa sau:

- Số lượng Nhãn Mục tiêu (K): $K = 10,000$ nhãn GO có điểm $S(t)$ cao nhất.
- Tỷ lệ Tập Validation: 15% dữ liệu được tách ngẫu nhiên cho tập kiểm định (Validation Set).
- Epochs: 50.
- Kích thước Lô (Batch Size): 128.
- Thuật toán Tối ưu hóa: AdamW với hệ số giảm trọng số weight_decay $= 0.01$.
- Tốc độ Học (Learning Rate): Khởi tạo ở 2×10^{-4}.
- Bộ Lập lịch Tốc độ Học: ReduceLROnPlateau được sử dụng để giảm tốc độ học còn 0.5 lần sau mỗi 3 epochs nếu Loss trên tập Validation không cải thiện.

D. Gradient Boosting

Nhóm sử dụng mô hình Gradient Boosting Machine (GBM) từ thư viện `Py-Boost`, một thư viện tối ưu hóa trên GPU cho bài toán Phân loại Đa Nhãn (Multi-Label Classification) với hiệu suất tính toán vượt trội.

1) Cấu hình huấn luyện: Để xử lý hiệu quả $K \approx 5000$ nhãn, mô hình sử dụng hàm mất mát Binary Cross-Entropy (BCE):

$$\mathcal{L} = -\sum_{k=1}^{K} [y_{i,k} \log(\hat{y}_{i,k}) + (1 - y_{i,k}) \log(1 - \hat{y}_{i,k})]$$

Đặc biệt, kỹ thuật Random Projection Sketching với kích thước chiếu $S = 3$ được áp dụng để giảm đáng kể nhu cầu bộ nhớ VRAM khi tính toán gradient, cho phép huấn luyện mô hình với số lượng nhãn lớn.

2) Siêu Tham số Huấn luyện: Các siêu tham số chính được cấu hình như sau:

- Số lượng Cây tối đa (NTREES): $10,000$.
- Tốc độ Học (LR): 0.01.
- Độ sâu tối đa (MAX_DEPTH): 6.
- Dừng sớm (ES): 200 vòng.
- Regularization: $\lambda_{\text{L2}} = 1$ (L2 regularization).
- Tối ưu Bộ nhớ: Kích thước bin histogram (‘max_bin‘) được đặt là 64.

E. Chiến lược Hợp nhất

Kết quả từ hai mô hình mạnh nhất—Mạng Nơ-ron Đa phương thức (Model A, NN) và Gradient Boosting (Model B, GBDT)—được kết hợp thông qua một chiến lược hợp nhất có trọng số và thưởng đa dạng (Diversity Bonus) để tối đa hóa hiệu suất.

1) Hợp nhất có Trọng số (Weighted Averaging): Điểm tin cậy cuối cùng S_{final} được tính toán bằng cách sử dụng trọng số không đồng đều, ưu tiên cho Mô hình A (NN) do có hiệu suất cao hơn trên tập kiểm định:

$$S_{base}(\mathbf{x}) = W_A \cdot S_A(\mathbf{x}) + W_B \cdot S_B(\mathbf{x}) \tag{15}$$

Với $W_A = 0.8$ và $W_B = 0.2$.

2) Thưởng Đồng thuận: Nhóm áp dụng một cơ chế thưởng đồng thuận để tăng cường độ tin cậy cho những dự đoán mà cả hai mô hình độc lập đều đưa ra. Thưởng B được áp dụng nếu cả hai điểm tin cậy S_A và S_B đều lớn hơn 0:

$$B(\mathbf{x}) = \begin{cases} \text{DIVERSITY\_BONUS} & \text{nếu } S_A(\mathbf{x}) > 0 \text{ và } S_B(\mathbf{x}) > 0 \\ 0 & \text{ngược lại} \end{cases} \tag{16}$$

Với DIVERSITY_BONUS $= 0.05$. Điểm cuối cùng trước khi điều chỉnh nhiễu là:

$$S_{interim} = \text{Clip}\left(S_{base} + B(\mathbf{x}), 1.0\right) \tag{17}$$

3) Áp dụng Nhiễu và Lọc Cuối cùng: Để phá vỡ thế hòa (tie-breaking) giữa các dự đoán có điểm số sát nhau, một nhiễu ngẫu nhiên $\epsilon \in [-\text{NOISE\_LEVEL}, \text{NOISE\_LEVEL}]$ được thêm vào:

$$S_{final} = \text{Clip}\left(S_{interim} + \epsilon, 0, 1\right) \tag{18}$$

với NOISE_LEVEL $= 0.0005$. Cuối cùng, chỉ các dự đoán có $S_{final} \geq$ CONFIDENCE_THRESHOLD (với CONFIDENCE_THRESHOLD $= 0.001$) mới được giữ lại và sắp xếp theo thứ tự giảm dần của S_{final} trước khi ghi vào file nộp.

F. Post processing

Sau khi ensemble, áp dụng kỹ thuật Hậu xử lý (Post-processing) sử dụng cơ sở tri thức ngoài (UniProt KB [6], [7]) trên kết quả ensemble tốt nhất.

V. Thực nghiệm và kết quả

A. Độ đo

Hiệu năng của mô hình dự đoán được đánh giá dựa trên chỉ số F-measure tối đa có trọng số (F_{max}). Chỉ số này được tính toán từ độ chính xác (precision) và độ phủ (recall), với trọng số dựa trên lượng tin (Information Content - $IC(\cdot)$) hoặc lượng tích lũy thông tin (Information Accretion - $IA(\cdot)$) của từng thuật ngữ chức năng. Trọng số $IA(f)$ phản ánh mức độ đặc hiệu của nhãn f trong cấu trúc phân cấp Gene Ontology, trong đó các nhãn hiếm gặp hoặc nằm sâu trong cây phả hệ sẽ có trọng số cao hơn.

Quá trình đánh giá được thực hiện độc lập trên ba phân nhóm bản thể: Molecular Function (MFO), Biological Process (BPO) và Cellular Component (CCO). Kết quả cuối cùng là trung bình cộng (arithmetic mean) của các chỉ số F_{max} đạt được trên ba phân nhóm này, được tổng hợp từ ba kịch bản kiểm thử dựa trên mức độ kiến thức tiên nghiệm về protein: *no-knowledge*, *limited-knowledge*, và *partial-knowledge*.

Công thức tổng quát cho độ đo F_{max} được định nghĩa như sau:

$$F_{max} = \max_{\tau} \left(\frac{2 \cdot pr(\tau) \cdot rc(\tau)}{pr(\tau) + rc(\tau)} \right) \tag{19}$$

Trong đó, $pr(\tau)$ và $rc(\tau)$ lần lượt là precision và recall có trọng số tại ngưỡng quyết định τ.

B. Kết quả

Trong khi mô hình đơn lẻ hoặc mô hình ensemble cơ sở (Base MLP) không có cơ chế chú ý thậm chí còn kém hiệu quả hơn các mô hình đơn lẻ (0.243), việc tích hợp khối Squeeze-and-Excitation đã chứng minh tính tối quan trọng của nó, thúc đẩy kết quả nhảy vọt lên $\mathbf{0.304}$, khẳng định hiệu quả của cơ chế chú ý trong việc lọc nhiễu giữa các không gian đặc trưng đa nguồn. Kết quả cuối cùng của nhóm đạt $\mathbf{0.358}$, xếp hạng 175 trong cuộc thi tính đến thời điểm 16/12/2025.

TABLE I
Kết quả Nghiên cứu Bóc tách (Ablation Study) và Hợp nhất

Mô hình/Chiến lược	**(Fmax)**
MLP(Ankh)	0.253
MLP(ESM-2)	0.268
MLP(ProtT5)	0.261
MLP(Ankh + ESM-2 + ProtT5)	0.243
Tailored-NN(Ankh + ESM-2 + ProtT5)	**0.304**
Gradient-Boosting(Ankh + ESM-2 + ProtT5)	0.288
Ensemble(Tailored-NN + Gradient-Boosting)	**0.308**
Ensemble(Tailored-NN + Gradient-Boosting) + Data collection	**0.358**

VI. Kết quả theo tuần

A. Tuần 11

Đọc hiểu đề bài, EDA

B. Tuần 12

- Thử nghiệm huấn luyện MLP đơn giản với embedding `esm2-8m` (0.12).
- Chuyển sang sử dụng embedding `ankh-base`, hiệu suất cải thiện rõ rệt (0.204).
- Áp dụng chiến lược chọn nhãn Top-k theo công thức Freq $\times$ IA kết hợp trọng số hàm loss theo IA (0.223).
- **Quyết định:** Nhóm chốt sử dụng chiến thuật xử lý nhãn dựa trên IA làm cấu hình mặc định cho dự án.

C. Tuần 13

- Nâng cấp backbone lên `esm2-650m`, train MLP đơn giản (0.223).
- Thử nghiệm chỉ lấy Top-20 nhãn có xác suất dự đoán cao nhất (0.243).
- Kết hợp Top-20 suy luận với việc lọc Top-10k nhãn theo Freq $\times$ IA (0.268).
- **Quyết định:** Nhóm chốt đây là chiến thuật suy luận và chọn nhãn mặc định.
- Đánh giá hiệu quả các embedding khác trên MLP: `ankh-base` (0.253), `esm2-3B` (0.26), `prot-t5` (0.261).
- Thử nghiệm các pretrain model kích thước lớn: `ankh-large` (0.24), `esm2-15B` (0.256).
- **Kết luận:** Mô hình pretrain càng lớn không đồng nghĩa kết quả tốt hơn (có thể do nhiễu hoặc khó hội tụ), nhóm loại bỏ các model quá khổ này.
- Hợp nhất (Concat) `ankh-base` và `esm2-650m`, train MLP đơn giản (0.28).
- Khảo sát miền nhãn trên mô hình hợp nhất:
  - Tăng lên Top-20k nhãn (0.274) $\rightarrow$ Kết quả giảm do nhiễu.
  - Lọc bỏ nhãn xuất hiện dưới 3 lần (0.271) $\rightarrow$ Kết quả giảm, chứng tỏ các nhãn hiếm (few-shot) vẫn mang thông tin quan trọng.
- Chuyển từ MLP sang kiến trúc NN đề xuất và tinh chỉnh tham số, kết quả cải thiện rõ rệt theo từng bước ($0.253 \rightarrow 0.287 \rightarrow 0.303 \rightarrow 0.304$).
- Thử nghiệm hợp nhất 4 nguồn (thêm `prot-bert`) trên kiến trúc NN (0.258) $\rightarrow$ Hiệu suất giảm.
- **Quyết định:** Chỉ sử dụng 3 nguồn embedding tối ưu nhất: `prot-t5`, `esm-650m` và `ankh-base`.
- Thực hiện đánh giá chéo 5-fold (Cross-validation) trên mô hình 3 nguồn đã chốt (0.278).

D. Tuần 14

- Thử nghiệm thay thế hàm mất mát bằng Asymmetric Loss (ASL), kết quả tốt nhất chỉ đạt mức khiêm tốn (0.292).
- Tinh chỉnh sâu kiến trúc (thay Batch Norm bằng Layer Norm, điều chỉnh Dropout/Hidden dims), kết quả bão hòa tại ngưỡng (0.304).
- **Nhận định:** Đây có thể là giới hạn trích xuất đặc trưng của các pretrain model do buộc phải cắt cụt (truncate) các chuỗi protein quá dài (chỉ xử lý được 1024/2048 axit amin đầu).
- Triển khai Gradient Boosting (Py-Boost) với Top-5000 nhãn (0.288). Không thể huấn luyện trên Top-10000 nhãn do giới hạn phần cứng.
- Thử nghiệm giải pháp chia nhỏ nhãn (Label Chunking) để khắc phục phần cứng, nhưng hiệu suất suy giảm.
- **Kết luận:** Việc học đồng thời (joint learning) giúp mô hình khai thác mối tương quan giữa các nhãn tốt hơn so với việc chia tách.
- Thực hiện Ensemble kết hợp giữa mô hình Deep Learning tối ưu (0.304) và Gradient Boosting (0.288), đạt kết quả cao nhất toàn dự án (0.308).

E. Tuần 15

- Thử nghiệm tích hợp thông tin phân loại học (Taxonomy): Mã hóa one-hot cho Top-100 loài phổ biến nhất và kết hợp với vector embedding `prot-t5`, `esm-650m` và `ankh-base`.
- **Kết quả:** Hiệu suất suy giảm so với baseline.
- **Nguyên nhân:** Nhóm xác định hiện tượng overfitting do chênh lệch phân phối dữ liệu lớn (tập Train/Val chỉ chứa 1.000 loài, trong khi tập Test chứa >8.000 loài).
- Thử nghiệm chiến lược "Chia để trị" theo khía cạnh sinh học (MF, BP, CC) cho cả NN và Gradient Boosting.
- **Kết quả:** Hiệu suất thấp hơn so với việc huấn luyện gộp toàn bộ nhãn.
- **Kết luận:** Củng cố giả thuyết rằng việc học đồng thời (Joint Learning) là thiết yếu để mô hình khai thác mối tương quan ẩn giữa các nhóm chức năng, ngay cả khi chúng thuộc các khía cạnh (aspect) khác nhau.
- Áp dụng kỹ thuật Hậu xử lý (Post-processing) sử dụng cơ sở tri thức ngoài (UniProt KB [6], [7]) trên kết quả ensemble tốt nhất (0.308). **Kết quả cuối cùng:** Cao nhất toàn bộ quá trình thực nghiệm (0.358).

VII. Kết luận

Báo cáo này đã trình bày một framework kết hợp các kỹ thuật học sâu tiên tiến và phương pháp ensemble để giải quyết bài toán dự đoán chức năng protein. Đóng góp cốt lõi của nhóm là xây dựng Kiến trúc Mạng Nơ-ron Đa phương thức song song, khai thác đồng thời các vector đặc trưng mạnh mẽ từ **Ankh, ESM-2, và ProtT5**. Đặc biệt, việc tích hợp khối **Squeeze-and-Excitation (SE-Block)** giúp mô hình vượt qua điểm yếu của MLP cơ sở và tăng chỉ số Fmax lên đáng kể (từ 0.243 lên $\mathbf{0.304}$). Hơn nữa, nhóm đã sử dụng Hàm mất mát Weighted BCE và thuật toán chọn lọc nhãn IA để xử lý sự mất cân bằng dữ liệu một cách hiệu quả. Cuối cùng, chiến lược Hợp nhất Nâng cao (Ensemble) với mô hình Gradient Boosting (Py-Boost) đã đẩy kết quả dự đoán tổng thể lên $\mathbf{0.308}$ (và 0.338 với dữ liệu bổ sung). Kết quả này mở ra hướng nghiên cứu tiềm năng trong việc tối ưu hóa tích hợp đa phương thức cho các bài toán tin sinh học.

References

[1] Z. Lin *et al.*, "Language models of protein sequences at the scale of evolution enable accurate structure prediction," *bioRxiv*, 2022.

[2] A. Rives *et al.*, "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences," *Proc. Natl. Acad. Sci. U.S.A. (PNAS)*, 2019, doi: 10.1101/622803.

[3] A. Elnaggar *et al.*, "Ankh: Optimized protein language model unlocks general-purpose modelling," *arXiv preprint arXiv:2301.06568*, 2023.
[4] M. Heinzinger *et al.*, "Bilingual language model for protein sequence and structure," *NAR Genomics and Bioinformatics*, vol. 6, no. 4, pp. lqae150, 2024, doi: 10.1093/nargab/lqae150.
[5] N. Brandes *et al.*, "ProteinBERT: a universal deep-learning model of protein sequence and function," *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, 2022, doi: 10.1093/bioinformatics/btac020.
[6] UniProt Consortium, "Index of /pub/databases/GO/goa/old/UNIPROT," Available online: https://ftp.uniprot.org/pub/databases/GO/goa/old/UNIPROT, [Accessed: Dec. 16, 2025].
[7] Gene Ontology Consortium, "Guide to GO Evidence Codes," Available online: https://geneontology.org/docs/guide-to-go-evidence-codes/, [Accessed: Dec. 16, 2025].