Decoding Functional Multiplicity: Graph Learning Approaches to Multifunctional Proteins

Mattia Cervellini*†, Enrico De Santis*, Antonello Rizzi*, Alessio Martino†
*Department of Information Engineering, Electronics and Telecommunications
University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy
Email: {mattia.cervellini, enrico.desantis, antonello.rizzi}@uniroma1.it
†Department of AI, Data and Decision Sciences, LUISS University of Rome, Viale Romania 32, 00197 Rome, Italy
Email: amartino@luiss.it

***Abstract*—Proteins are central to virtually all biological processes, and their structural diversity underlies a wide range of functions. Among them, multifunctional proteins (i.e., those associated with multiple enzymatic activities) pose unique challenges for computational analysis. In this work, we apply a suite of machine learning and deep learning techniques to predict multifunctionality roles starting from protein 3D structures. Proteins are represented as residue contact networks, enabling the use of graph machine learning approaches. We explore feature-based methods grounded in simplicial complexes analysis alongside end-to-end Graph Neural Networks implementing recent message-passing schemes. Our models address the task of multi-label classification of the first level Enzyme Commission numbers of multifunctional proteins. Evaluation across a strictly multifunctional subset of the human proteome associated with repeated stratified validation demonstrates that structural graph representations effectively capture signals of multifunctionality, shedding light on how protein architecture encodes a diverse range of biochemical roles.**

***Index Terms*—Bioinformatics, Machine Learning, Complex Systems, Graphs, Graph Neural Networks, Multifunctional Proteins.**

I. Introduction

Proteins are fundamental biological macromolecules that orchestrate the majority of cellular processes. Their functional roles range from catalyzing biochemical reactions and transporting metabolites to mediating signaling pathways and regulating gene expression. While classical molecular biology often associates a protein with a dominant physiological role (defined via the associated Enzyme Commission number [1]), a substantial fraction of proteins are multifunctional (i.e., they can perform multiple distinct tasks depending on various factors) [2]. Such functional multiplicity is not merely a notation artifact, it is a biologically relevant property that contributes to proper cellular functioning.

From a structural perspective, proteins can be naturally described as complex systems of interconnected amino acid residues [3], [4]. Their behavior emerges from the interplay of many interacting components across multiple scales: atoms form residues, residues form secondary-structure motifs, motifs assemble into domains, and domains cooperate to create a three-dimensional architecture whose dynamics enable function. This multi-scale, interconnected organization motivates the use of representations that go beyond linear sequences or global descriptors, conversely capturing the higher-order organization of molecular structures. A principled way to model such interconnected organization is to represent proteins as graphs (or hypergraphs) whose nodes and edges encode structural entities and their relations [5]–[7].

Under these lenses, protein function prediction becomes a learning problem over complex relational data, where the goal is to infer functional labels from topological features that reflect the underlying organization of the molecule. Importantly, when proteins are multifunctional the learning task is intrinsically multi-label: each analyzed protein is associated with a set of functional annotations simultaneously.

Practically speaking, the focus of the work is to correctly predict the set of first level Enzyme Commission (EC) numbers (also known as main enzyme classes) associated with each multifunctional protein. Table I presents a brief summary of the possible first level EC numbers defined by the International Union of Biochemistry and Molecular Biology in 1961 [8], with Translocases (EC 7) only formalized in 2018.

The main contributions of this work can be summarized as follows:

- We cast the prediction of multifunctional proteins as a multi-label classification task on structure-derived protein contact networks where each protein is associated with multiple first-level EC annotations simultaneously.
- We rigorously compare simplicial embeddings, specialized graph kernels, and message-passing GNN within a common evaluation and hyperparameter-optimization framework.
- We empirically demonstrate that topology-aware structural representations encode relevant signals for multi-label functional annotation of proteins.

The remainder of the paper is structured as follows: in Section II we discuss related works; in Section III we describe the whole array of graph-based machine learning techniques for predicting the multifunctional role of proteins; in Section IV and V we describe the dataset in detail[1] and the computational

[1]All the data used for analysis can be freely downloaded from Protein Data Bank (PDB) at https://www.rcsb.org. Interested researchers can be provided with the list of analyzed proteins by asking the authors.

results, respectively. Finally, Section VI concludes the paper, drafting future lines of research.

TABLE I
MAIN EC CLASSES, NAMES, AND FUNCTIONS - SOURCE: ADAPTED FROM [9].

EC Class	Name	Function
EC 1	Oxidoreductases	Catalyze redox reactions by transferring electrons between substrates.
EC 2	Transferases	Transfer functional groups from one molecule (donor) to another (acceptor).
EC 3	Hydrolases	Catalyze hydrolytic cleavage of bonds by addition of water (e.g., ester, glycosidic bonds).
EC 4	Lyases	Cleave bonds by means other than hydrolysis or oxidation, often forming double bonds or rings.
EC 5	Isomerases	Catalyze intramolecular rearrangements, converting a molecule into one of its isomers.
EC 6	Ligases	Join two molecules by forming new bonds, typically coupled to ATP hydrolysis.
EC 7	Translocases	Catalyze the movement of ions or molecules across membranes or their separation within membranes.

II. RELATED WORKS

Classifying proteins into EC main classes has been extensively explored in scientific literature. Early machine learning works showed that the EC class can be predicted from sequence-derived physicochemical descriptors [10]–[12]. Soon after, sequence-based representations were complemented by more informative graph-based representations known as Protein Contact Networks (PCNs) [13]. In PCNs atomic elements of the molecule (i.e., residues) represent nodes of the network, while proximity or specific interactions are used to determine the presence of edges [13]–[15]. This representation explicitly encodes the three-dimensional topological interaction of proteins, capturing long-range residue contacts and global structural organization that are often inaccessible to linear sequence representations, yet crucial for biochemical function. Thanks to these characteristics, PCNs have been the basis for many studies on protein functioning in the early 2000s [5]–[7], while in more recent times they became the foundational block of various studies aimed at predicting the main enzymatic class of proteins [16]–[18].

To respect the biological relevance of multifunctional proteins, different researchers experimented with multi-label functional classification of complex protein structures thanks to a variety of different methods such as InterPro signatures [19], handcrafted features deriving from the position-specific scoring matrix [20], customizations of Pseudo Amino Acid Composition (PseAAC) descriptors [21], and modern deep representation learning approaches [22].

These approaches primarily rely on sequence-based or signature-based representations, leaving the potential of graph-based topological descriptors for multi-label enzyme classification largely unexplored. This work specifically addresses this gap under a rigorous experimental framework aimed at validating the discriminative capabilities encoded in strictly topological information regarding the morphological conformation of proteins. This work builds on the foundations of graphs and complex systems theory to evaluate and compare the performances of both topological embeddings and modern GNNs optimized to exploit the topological information encoded in PCNs.

III. METHODS

A. Proteins as PCNs

Protein structures were processed by creating PCNs following the same methodology proposed in [13], [16]–[18]. Starting from the spatial 3D atomic coordinates in the proteins, their Cα atoms were used as nodes in the resulting PCNs. Cα atoms were connected if the Euclidean distance among them was in the range [4–8]Å: the lower bound was set in order to discard trivial first-neighbour interactions along the chain of a protein while the upper bound was set to approximately two Van der Waals radii of Cα atoms [23]. It is important to note that all methods described in this paper use only topological information deriving from inter-residue interactions, not relative positioning in the 3D space (e.g., incidence angles).

B. Embedding via Simplicial Complexes

Simplicial complexes represent a concept from algebraic topology which has been widely explored in the domain of graph and hypergraph analysis [24]–[26]. A simplex of order k (k-simplex) is effectively a convex hull of $(k+1)$ points. Any non-empty subset of these points defines a face of the simplex, which is itself a simplex of lower order. Following this reasoning, a simplicial complex $\mathcal{S}$ can be defined as a group of simplices having the following two properties:

1) if $s \in \mathcal{S}$, every face in s is also included in $\mathcal{S}$
2) if $s_1, s_2 \in \mathcal{S}$, then $s_1 \cap s_2$ is a face of both s_1 and s_2

By combining this concept with PCNs it is possible to conceptualize an embedding technique based on simplicial complexes. Simplices of node-labels (i.e., amino acid names) can be used to create a *symbolic histogram* of the protein structure by counting how many times each simplex appears in the protein [27]. In other words, each PCN can be effectively embedded in the Euclidean space as a multi-set of its simplices.

Practically speaking, edges of PCNs were aggregated to create clique hypergraphs[2] in order to explicitly represent higher-order interactions that cannot be captured by the binary nature of PCNs.

Hypergraphs have been explored in literature to represent a great variety of complex systems (e.g., co-authorship networks, metabolic networks, brain functional networks, etc.)

[2]Namely, hypergraphs generated starting from a plain graph in which maximal cliques were substituted with hyper-edges of the same order [28].

due to their flexibility and expressive power [29]–[31]. In order to carry out such embedding procedure, it is however necessary to compute the entire dictionary of distinct simplices in the dataset which in this specific application had a dimensionality of $\sim 12,000$ entries.

The instance matrix $\mathbf{X}^{(S)}$ resulting from the embedding via simplicial complexes has shape $n \times |d|$ where n is the number of proteins in the dataset and d is the dictionary of distinct simplices in the dataset. Considering $c(\mathcal{H}_i, d_j)$ a function that counts how many times the simplex d_j appears in the clique hypergraph $\mathcal{H}_i$ the instance matrix $\mathbf{X}^{(S)}$ can be defined as

$$\mathbf{X}^{(S)}_{i,j} = c(\mathcal{H}_i, d_j)$$

C. INDVAL Scores

In order to mitigate the high dimensionality of the embedding presented in Section III-B, the INDVAL score was considered [32] as a model-agnostic feature selection criterion. The INDVAL score is a sensitivity and fidelity integrated evaluation originally proposed to identify the most characteristic species of a given environment. According to the INDVAL criterion, a species s is an indicator of an environment class E if it is both specific to E, meaning that its occurrences are largely concentrated in E compared to other classes, and faithful to E, meaning that it is present in a high proportion of the individual environments (sites) belonging to E. The indicator value combines these two aspects into a single quantitative score without requiring strict exclusivity or ubiquity. By drawing a parallelism by which one considers environmental classes as classification labels and individual species as substructures of a PCN, it is possible to restate the INDVAL criterion as follows: a graph substructure s acts as an indicator for a protein class E when it is both specific and faithful. Specificity reflects the degree to which the frequency of s is biased toward proteins in class E, while fidelity measures the fraction of PCNs in class E in which s occurs. The resulting indicator value provides a continuous measure of class association, allowing a set of sub-structures to be identified as relevant patterns.

With this approach it was possible to construct a restricted version of the embedding presented in section III-B where only the sub-structures with the best INDVAL scores are retained. Considering j a specific class, and i a specific simplex, the INDVAL score can be defined starting from these scores:

$$A_{i,j} = \frac{\text{\# structures with class } j \text{ having simplex } i}{\text{\# structures having simplex } i} \in [0,1]$$

$$B_{i,j} = \frac{\text{\# structures with class } j \text{ having simplex } i}{\text{\# structures exhibiting class } j} \in [0,1]$$

Combining $A_{i,j}$ (specificity) and $B_{i,j}$ (fidelity) the INDVAL score $I_{i,j}$ can be defined as $I_{i,j} = A_{i,j} \cdot B_{i,j} \cdot 100 \in [0, 100]$.

After the creation of the INDVAL matrix $\mathbf{I}$ it is possible to select simplices which have at least one of their INDVAL scores (regardless of the class per se) above a certain user-defined threshold τ. The choice of τ is not straightforward and strongly depends on the specific dataset. Figure 1 presents the number of features included in the dataset for every possible τ. The vertical line in the figure represents the chosen threshold $\tau = 5$, chosen in accordance with the elbow rule heuristic [33], resulting in a dimensionality reduction of the dataset of $\sim 80\%$. This sharp feature selection will put to the test the effectiveness of INDVAL scores in selecting the most relevant sub-structures for the task at hand.

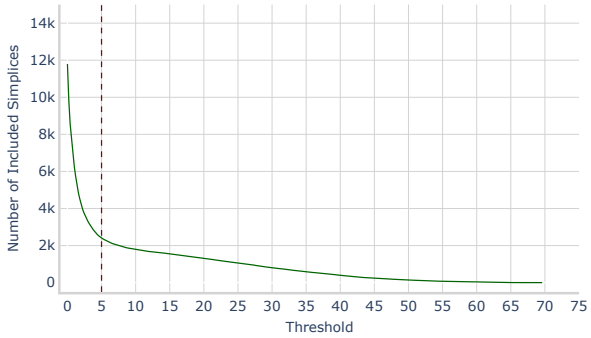

Fig. 1. Number of Features at Various Threshold Levels

D. (Hyper)Graph Kernels

Graph kernels are among the most widely used techniques for graph machine learning tasks [34]. The kernel methods presented in this section are adapted from [26] and can be constructed directly starting from the simplicial complexes interpretation of PCNs provided in Section III-B.

The *Histogram Cosine Kernel* (HCK) between any two PCNs can be computed as the cosine similarity between their respective *symbolic histogram* representations (rows in $\mathbf{X}^{(S)}$). Considering $\mathbf{x}_i = \mathbf{X}^{(S)}_{i,\cdot}$ the HCK between any two PCNs i and j can be defined as

$$\text{HCK}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{||\mathbf{x}_i||\ ||\mathbf{x}_j||} \in [0,1]$$

The *Weighted Jaccard Kernel* (WJK) is computed as the ratio between the intersection and the union of the two simplicial complexes by accounting their multi-set nature. Considering again d as the dictionary of all simplices represented in the *symbolic histograms* and $\mathbf{x}_i = \mathbf{X}^{(S)}_{i,\cdot}$ the WJK between any two PCNs i and j can be defined as

$$\text{WJK}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{k=1}^{|d|} \min\{\mathbf{x}_{i,k}, \mathbf{x}_{j,k}\}}{\sum_{k=1}^{|d|} \max\{\mathbf{x}_{i,k}, \mathbf{x}_{j,k}\}} \in [0,1]$$

The *Edit Kernel* (EK) computes the similarity among PCNs in relation to the minimum number of simplices that need to be inserted, removed or substituted in order to make the two associated simplicial complexes identical. Considering $\mathcal{S}_i$ the simplicial complex associated with the clique hypergraph of a specific protein and $e(\mathcal{S}_i, \mathcal{S}_j)$ an edit distance with unitary weights, the EK between any two PCNs i and j reads as

$$\text{EK}(\mathcal{S}_i, \mathcal{S}_j) = 1 - \frac{2 \cdot e(\mathcal{S}_i, \mathcal{S}_j)}{|\mathcal{S}_i| + |\mathcal{S}_j| + e(\mathcal{S}_i, \mathcal{S}_j)} \in [0,1]$$

E. Graph Neural Networks

Graph Neural Networks (GNNs) have proven highly effective for graph classification tasks, frequently achieving state-of-the-art performances. However, the large design space induced by different message-passing (MP) strategies, architectural depths, and pooling mechanisms makes exhaustive structural exploration computationally prohibitive. Moreover, deep MP stacks are known to suffer from over-smoothing, where node representations converge to near-identical vectors after multiple propagation steps [35].

To balance expressive power and computational feasibility, all evaluated GNN architectures were deliberately constrained to configurations with at most 5 message-passing layers, 5 layers in the classification head, and a maximum hidden dimensionality of 256 per individual node representation.

GNNs operated directly on PCNs without requiring additional feature engineering. Each node was associated with a single categorical attribute (i.e., its residue name) which was encoded either using one-hot vectors or via dense learnable embeddings. The choice between these encoding strategies was treated as a tunable architectural parameter.

Following input embedding, node representations were updated through MP layers augmented with residual connections to stabilize gradient flow. The architectural exploration focused on several well-established MP paradigms, namely GraphConv [36], GraphSAGE [37], GCN [38], GIN [39], and GATv2 [40]. Each candidate architecture employed a single MP strategy throughout the entire network and node representation dimensionality was kept constant across all MP layers after the initial projection. These two architectural choices were taken to guarantee a coherent structure (e.g., without bottlenecks) and a manageable hyperparameter space.

After MP, node representations were aggregated into a single latent graph representation via a pooling operation, either using standard permutation-invariant reductions (i.e., mean, max or sum) or via attentional aggregation [41], the latter assigning data-dependent importance weights over nodes. The pooled graph representation was finally processed by a standard Multi-Layer Perceptron to produce predictions on the label set of each protein.

F. Tested Classifiers

The embedding techniques presented in the previous sections have been compared using a variety of well-known machine learning classification models. After some early exploration, the choice of optimal candidates for the study was: (i) ℓ_1-Linear-SVM for its speed and feature selection capabilities [42], (ii) Kernelized ν-SVM for its ability to work also with pre-computed kernels and great flexibility [43], and (iii) Random Forest (RF) for its all-around good performances on a great variety of tasks [44]. Such classification algorithms were used on the embedding methods described in Sections III-B and III-C while Kernel methods described in Section III-D were modeled only via ν-SVM as it naturally supports pre-computed kernels thanks to the dual formulation of the SVM problem [45]. Given the multi-label setting, all classifiers were trained using a one-vs-rest strategy, i.e., a distinct binary classifier was trained independently for each label, and their outputs were combined to form the final multi-label prediction.

G. Performance Metrics

Given the multi-label nature of the task, let $\mathbf{y}_i \in \{0,1\}^C$ and $\hat{\mathbf{y}}_i \in \{0,1\}^C$ denote the ground-truth and predicted label vectors for sample i, with C labels and n samples. Model performance was evaluated using the following multi-label metrics:

- **Accuracy**: defined as

  $$\frac{1}{n}\sum\nolimits_{i=1}^{n} \mathbb{I}(\hat{\mathbf{y}}_i \overset{?}{=} \mathbf{y}_i), \quad \text{where} \quad \mathbb{I}(x) = \begin{cases} 1, \text{if } x \text{ is True} \\ 0, \text{otherwise} \end{cases}$$

  It measures the fraction of samples for which the entire label set is predicted correctly and is therefore a strict metric in the multi-label context.
- **Sample F1-score**: defined as

  $$\frac{1}{n}\sum\nolimits_{i=1}^{n} \frac{2|\hat{\mathbf{y}}_i \cap \mathbf{y}_i|}{|\hat{\mathbf{y}}_i| + |\mathbf{y}_i|}$$

  It captures per-sample agreement between predicted and true labels, balancing false positives and false negatives.

The *Sample F1-score* was selected as the primary performance metric and used as the validation objective for hyper-parameter optimization (see later Section III-H). This choice favors models that accurately recover complete label sets while also rewarding partial label overlap which is key for any optimizer.

The results of the best candidate combinations of classifiers and embedding strategies were analyzed further in a per-class fashion via F1-score and balanced accuracy, shown in addition to standard accuracy as being robust to imbalanced classes.

H. Hyperparameter Optimization and Data Splitting

Hyperparameter optimization was performed using multivariate Tree-Structured Parzen Estimators, a sequential Bayesian optimization method that models $p(x \mid y)$ rather than the typical $p(y \mid x)$, yielding a sampling strategy closely related to Expected Improvement and naturally supporting mixed and conditional search spaces [46], [47].

Model optimization and evaluation followed a five-run stratified resampling protocol. The dataset was partitioned into five mutually exclusive stratified training, validation and test sets (60%-20%-20% ratio) per run. Hyperparameter optimization was conducted independently for each run: for each candidate configuration, the model was trained on the training set and the objective function reads as the sample F1-score (see Section III-G) on the validation set. At the end of the optimization, the best candidate model is retrained on the training set and finally tested on the test set. To ensure fair, paired comparisons, the same fold assignments were fixed across all representation strategies and learning algorithms.

Table II presents a summary of the hyperparameter search spaces for each of classification algorithm.

TABLE II
SEARCH SPACES FOR CANDIDATE MODEL FAMILIES

GNNs		
Parameter Name	**Type**	**Values / Range**
Data Embedding Type	Cat	OHE, dense
Embedding Dimensions †	Int	[8, 256]
Activation Type	Cat	ReLU, LeakyReLU
Negative LeakyReLU Slope †	Float	[0.01, 0.3]
MP Strategy	Cat	GraphConv, Sage, GCN, GIN, GATv2
Num. of MP Layers	Int	[1, 5]
Normalization Type	Cat	Graph, GraphSize, Batch, Layer, None
Hidden Dimensions	Int	[64, 256]
Pooling Type	Cat	Mean, Sum, Max, Attention
After Pooling Norm.	Cat	Layer, None
Num. of MLP Layers	Int	[1, 5]
MLP Dropout	Float	[0, 0.5]
SAGE aggregation †	Cat	Mean, Max, LSTM
Num. of GAT Heads †	Int	{2, 4, 8}
Num. of GIN MLP Layers †	Int	[2, 3]
ν-SVM ♠		
Kernel Type	Cat	RBF, polynomial, sigmoid
ν	Float	(0, 1]
Gamma	Float	[1e-6, 1e+2]
Degree †	Int	[2, 7]
Scale	Cat	True, False
RF		
Num. of Trees	Int	[10, 2000]
Max Tree Depth	Int	[2, 20]
Max Features per Tree	Cat	sqrt, log2
Min Samples per Split	Int	[2, 300]
Min Samples per Leaf	Int	[1, 300]
Criterion	Cat	gini, entropy
ℓ_1-Lin-SVM		
C	Float	[1e-5, 1e+4]
scale	Cat	True, False

† denotes conditional hyperparameters (e.g., dependent on kernel, architecture choices, etc.).
♠ when ν-SVM was trained using a precomputed kernel (i.e., HCK, WJK and EK) the only optimizable hyper-parameter was ν

I. Summary of the Compared Approaches

Overall, the compared methodologies differ mainly in how they represent the topological information encoded in PCNs and in how such information is exploited for classification. Simplicial complexes embeddings transform each protein into an explicit vector of higher-order topological patterns. INDVAL feature selection allows for a representation focused on the most relevant substructures. Graph kernels, instead, avoid explicit embedding-based approaches by defining non-linear similarities between proteins directly from their simplicial organization. Finally GNNs operate end-to-end on the original PCNs, learning task-oriented representations directly from residue interactions without the need for a predefined topological dictionary.

IV. DATA COLLECTION

The dataset construction started from the entirety of the human proteome ($\sim$ 70,000 proteins), whose structures were downloaded from PDB on March 1st, 2025 and parsed via specialized Python packages [48], [49]. The data ingestion pipeline then followed these steps: (i) retaining only proteins exhibiting multifunctional or moonlighting properties, i.e. presenting more than one first-level EC number; (ii) excluding proteins with missing resolution or with a resolution exceeding 3Å, since, given the residue interaction range of [4–8]Å, only highly detailed structures were considered informative for the analysis; (iii) excluding proteins showing evidently degenerate structures after PCN construction (e.g., a single residue or very distant residues); (iv) excluding all structures showing EC 7 due to their scarcity in the dataset (only 18 molecules); (v) Retaining non-enzymatic functionalities as explicit labels, as multifunctional proteins are known to concurrently perform both enzymatic and non-enzymatic physiological roles [2], [50]; (vi) retaining only the first molecular structure when multiple were available.

The final dataset was composed of 4030 multifunctional proteins whose label distribution can be observed in Table III.

TABLE III
EC NUMBER DISTRIBUTION.

EC	**Support**	**Prevalence ‡**
Non-Enz	3017	74.86%
EC 1	301	7.47%
EC 2	2396	59.45%
EC 3	1599	39.77%
EC 4	577	14.31%
EC 5	149	3.70%
EC 6	252	6.25%

‡ Percentages do not sum to 100% due to the multi-label nature of the dataset.

V. RESULTS

A. Overall Multi-Label Results

Table IV reports the test set performances for all combinations of representation strategies and classifiers (avg $\pm$ std across 5 runs).

Overall, all methods achieve satisfactory predictive performances, with Sample F1 ranging from 0.889 to 0.923 and Accuracy from 0.721 to 0.819. The strongest results are obtained by ν-SVM in combination with the WJK (Accuracy of 0.819 and Sample F1 of 0.923) closely matched by the GNN (Accuracy of 0.812 and Sample F1 of 0.921). In contrast, RF on the full simplicial embedding yields the lowest performance (Accuracy of 0.721 and Sample F1 of 0.889), consistent with the sensitivity of RF to high-dimensional feature spaces when many features are weakly informative [51].

INDVAL-based feature selection produces model-dependent effects. For RF, moving from the full simplicial embedding to INDVAL improves Accuracy by approximately 3.6%, with associated substantial computational and memory savings. For ℓ_1-Lin-SVM, INDVAL decreases Accuracy by approximately 2%, probably because ℓ_1 regularization already induces strong sparsity which is most effective in very high-dimensional regimes such as the full simplicial complexes embedding [52]. For ν-SVM, the effect of INDVAL is very minimal (less than

1% variation), indicating that this model is relatively robust to feature reduction while benefiting from reduced running times.

Finally, among the tested specialized kernels, WJK and EK perform similarly and consistently outperform HCK, bringing kernel learning in line with modern deep learning approaches. These results indicate that, despite the various practical advantages of GNNs (e.g., avoiding a dataset-dependent dictionary building and supporting incremental updates), SVMs equipped with appropriate kernels remain highly competitive for multi-functional protein classification.

TABLE IV
OVERALL MULTI-LABEL TEST SET RESULTS.

Classifier	Dataset	Accuracy	Sample F1
GNN	PCNs	0.812 ± 0.021	0.921 ± 0.011
ℓ_1-Lin-SVM	Simplicial	0.779 ± 0.020	0.908 ± 0.008
	INDVAL	0.764 ± 0.025	0.904 ± 0.010
ν-SVM	Simplicial	0.803 ± 0.020	0.914 ± 0.010
	INDVAL	0.797 ± 0.014	0.913 ± 0.006
	HCK	0.783 ± 0.017	0.910 ± 0.006
	WJK	0.819 ± 0.012	0.923 ± 0.007
	EK	0.811 ± 0.008	0.919 ± 0.009
RF	Simplicial	0.721 ± 0.021	0.889 ± 0.010
	INDVAL	0.747 ± 0.023	0.897 ± 0.012

B. Further Discussion on Per-class Results

Tables V–VIII present the detailed per-class performances (avg $\pm$ std across 5 runs) of the best candidate for each family of classification models, specifically: (i) GNNs, (ii) ℓ_1-Lin-SVM on the full simplicial embedding, (iii) ν-SVM on the WJK and, (iv) RF on the INDVAL filtered dataset.

All methods seem to exhibit satisfactory performances in detecting Non-Enzymatic functionalities of the proteins (Balanced Accuracy > 0.898 for all candidates). This indicates that the topological characteristics that identify such functions are clearly encoded in all representation strategies employed in this work. Non-Enzymatic functions are also the most prevalent in the dataset which aligns perfectly with the high discriminative power expressed by the classifiers.

Oxidoreductases (EC 1), Lyases (EC 4) and Isomerases (EC 5) achieve strong per-class performance despite their low prevalence in the dataset (7.47%, 14.31%, and 3.70%, respectively – see Table III). The combination of high Balanced Accuracy and F1 regardless of the sparsity of the labels suggests that these functional classes may be associated with more distinctive or less overlapping topological patterns, enabling reliable classification even with limited training examples. Lyases in particular exhibit the strongest results in the study, with a minimum balanced accuracy of 0.955.

Transferases (EC 2) and Hydrolases (EC 3) are the second and third most prevalent classes in the dataset respectively. The classification performances on them are however weaker compared to both Non-Enzymatic and Lyases functionalities, suggesting that the topological patterns distinguishing them may either be weak or partially shared with other biological functionalities.

Ligases (EC 6) proved to be the most challenging to correctly classify, they are also among the least represented classes (6.25% of the dataset). Only ℓ_1-Lin-SVM was able to achieve a Balanced Accuracy above 0.8 on EC 6, whereas other methods—that tend to perform better overall—are not able to correctly discriminate EC 6. This suggests that the topological signals for Ligases are subtle and better captured when feature selection is enforced through the sparsity-inducing ℓ_1 penalty. This behavior may also reflect partial overlap between EC 6 and other labels in the dataset.

Across the four selected candidates, GNNs, ν-SVM with the WJK, and ℓ_1-Lin-SVM on the full simplicial embedding provide the most balanced performance profile across classes, achieving consistently high scores on the majority labels while maintaining strong robustness on rare classes such as EC 1, 5 and 6. RF on the INDVAL-filtered embedding constitutes a competitive lightweight baseline, but shows the most pronounced degradation on low-prevalence classes, indicating that its decision boundaries benefit less from topological insights when training signal is scarce. Overall, these patterns suggest a trade-off between balanced performances on most classes and the ability to discriminate sparse and overlapping classes.

Given the strong performance and relative robustness of GNN-based models across classes, their architectural configurations were further analyzed to understand whether they exhibit consistent patterns across data splits. GNN hyperparameter search converged toward a narrow set of high-performing architectural choices, despite being optimized independently on each split. This consistency suggests stable design preferences related to biological reality rather than split-specific overfitting. Specifically, configurations using explicit one-hot node features are repeatedly selected, indicating that preserving discrete amino-acid identity is advantageous in this setting. The best models typically employ expressive message-passing operators with high depth (around four layers) and relatively wide hidden representations (approximately 200 dimensions), concretizing a regime in which residual connections help maintain stable optimization and limit over-smoothing. At the activation level, LeakyReLU is consistently preferred, while global pooling is more variable, with sum and mean aggregation appearing most often. Normalization emerges as a recurring stabilizer, with graph or batch normalization within message passing and standard layer normalization after pooling selected in the majority of best configurations. Finally, the classification heads are generally shallow and paired with high dropout, suggesting that performance is primarily driven by the learned node representations and interactions rather than by a complex classifier head.

VI. CONCLUSIONS AND FUTURE PROSPECTS

This paper investigated whether protein structures represented as PCNs carry sufficient topological signal to support accurate prediction of physiological roles in the multi-label setting of multifunctional proteins. Besides its foundations in topological analysis and graph representation learning, the methodologies explored in the paper involved concepts from

TABLE V
GNN PER-CLASS TEST SET RESULTS.

Class	Accuracy	F1	Bal. Accuracy
Non-Enz	0.946 ± 0.011	0.964 ± 0.007	0.920 ± 0.016
EC 1	0.979 ± 0.003	0.851 ± 0.020	0.893 ± 0.032
EC 2	0.889 ± 0.017	0.908 ± 0.013	0.881 ± 0.020
EC 3	0.917 ± 0.014	0.894 ± 0.019	0.911 ± 0.017
EC 4	0.992 ± 0.003	0.970 ± 0.011	0.976 ± 0.009
EC 5	0.989 ± 0.003	0.826 ± 0.056	0.856 ± 0.040
EC 6	0.965 ± 0.012	0.668 ± 0.118	0.785 ± 0.071

TABLE VI
ℓ_1-LIN-SVM ON SIMPLICIAL EMBEDDING PER-CLASS TEST SET RESULTS.

Class	Accuracy	F1	Bal. Accuracy
Non-Enz	0.925 ± 0.008	0.950 ± 0.006	0.906 ± 0.014
EC 1	0.980 ± 0.004	0.856 ± 0.035	0.898 ± 0.032
EC 2	0.881 ± 0.014	0.900 ± 0.012	0.875 ± 0.016
EC 3	0.909 ± 0.009	0.885 ± 0.014	0.904 ± 0.013
EC 4	0.991 ± 0.003	0.967 ± 0.010	0.976 ± 0.010
EC 5	0.988 ± 0.003	0.828 ± 0.033	0.897 ± 0.022
EC 6	0.967 ± 0.007	0.708 ± 0.065	0.819 ± 0.036

TABLE VII
ν-SVM WITH WJK PER-CLASS TEST SET RESULTS.

Class	Accuracy	F1	Bal. Accuracy
Non-Enz	0.950 ± 0.005	0.967 ± 0.003	0.921 ± 0.009
EC 1	0.981 ± 0.006	0.852 ± 0.052	0.878 ± 0.039
EC 2	0.897 ± 0.010	0.913 ± 0.008	0.892 ± 0.011
EC 3	0.919 ± 0.015	0.897 ± 0.021	0.915 ± 0.018
EC 4	0.992 ± 0.003	0.971 ± 0.011	0.972 ± 0.011
EC 5	0.991 ± 0.002	0.865 ± 0.038	0.882 ± 0.030
EC 6	0.967 ± 0.010	0.672 ± 0.118	0.775 ± 0.062

TABLE VIII
RF ON INDVAL EMBEDDING PER-CLASS TEST SET RESULTS.

Class	Accuracy	F1	Bal. Accuracy
Non-Enz	0.941 ± 0.008	0.962 ± 0.005	0.898 ± 0.014
EC 1	0.974 ± 0.003	0.791 ± 0.031	0.829 ± 0.021
EC 2	0.887 ± 0.015	0.903 ± 0.013	0.887 ± 0.017
EC 3	0.892 ± 0.016	0.851 ± 0.024	0.873 ± 0.018
EC 4	0.987 ± 0.005	0.953 ± 0.018	0.955 ± 0.016
EC 5	0.986 ± 0.001	0.765 ± 0.025	0.812 ± 0.016
EC 6	0.960 ± 0.008	0.562 ± 0.106	0.705 ± 0.048

algebraic topology (i.e., simplicial complexes), quantitative ecology (i.e., the original INDVAL score formulation), kernel methods (i.e., graph kernels) and deep learning (i.e., GNNs). Such a wide array of techniques was combined with a rigorous data analysis framework, highlighted by the shared repeated data splitting and by the extensive Bayesian hyperparameter optimization. The ensemble of these factors allowed for true like-for-like methodological comparison in the topological graph analysis domain applied to bioinformatics. The study rigorously compared (i) graph embedding strategies, (ii) specialized graph kernels, and (iii) end-to-end learning directly from the topological structure of the proteins (i.e., GNNs); highlighting strengths and weaknesses of each approach while demonstrating the potentiality of topology-driven representations of complex systems such as proteins.

Experimental findings empirically showed how simplicial-based specialized graph kernels and end-to-end neural graph learning appear as the most promising methodologies for correct functional annotation of multifunctional proteins. While the analyzed graph kernels are based on strongly sample-dependent statistics (i.e., the dictionary of available simplices) they are able to maintain a clear biological relevance as each of the pivotal features is clearly defined and rooted in the topology of the molecules. End-to-end GNNs, on the other hand, excel in scalability and adaptability, it would in fact be possible to easily extend the produced models to different variations of the dataset (e.g., multifunctional proteins coming from different organisms) possibly without the need of full retraining. The price to pay for such versatility (the only sample dependent statistic needed is the number of distinct amino acids, which is constant) is, as in most cases involving deep learning approaches, the lack of clear connection between the model results and the phenomenon of interest which may tilt the analysis towards a black-box solution.

Overall this paper demonstrates how topology-aware representations of proteins encode sufficient information for accurate multifunctional annotation. The results are strengthened by methodological breadth and variance-aware resampling. However, some limitations should be recognized: (i) the dataset was restricted to human proteins, (ii) the classification task was explored only at the first level of the EC classification, and (iii) topological information was limited to inter-residue interaction without geometric information (i.e., how residues are relatively positioned in the space) or multi-level analysis of higher-order structures (e.g., secondary structures, domains).

A final axis of expansion could be the exploration of model explainability paradigms known in literature to retrieve concrete insights on which signals are determinant for protein functionality annotation: although embedding-based methods, especially if equipped with the INDVAL thresholding, are able to retain informative simplices, a fully biological analysis of the resulting simplices is still on our research agenda.

ACKNOWLEDGMENTS

A.M. has been partially supported by the project "NextGRAAL: Next-generation algorithms for constrained GRAph visuALization" funded by MUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2022 - Grant ID 2022ME9Z78.

REFERENCES

[1] E. C. Webb, *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes.*, 6th ed. Academic Press, 1992.

[2] C. J. Jeffery, "Multifunctional proteins: examples of gene sharing," *Annals of medicine*, vol. 35, no. 1, pp. 28–35, 2003.

[3] G. Bagler and S. Sinha, "Network properties of protein structures," *Physica A: Statistical Mechanics and its Applications*, vol. 346, no. 1-2, pp. 27–33, 2005.

[4] W. Yan, J. Zhou, M. Sun, J. Chen, G. Hu, and B. Shen, "The construction of an amino acid network for understanding protein structure and function," *Amino acids*, vol. 46, no. 6, pp. 1419–1439, 2014.

[5] L. H. Greene and V. A. Higman, "Uncovering network systems within protein structures," *Journal of molecular biology*, vol. 334, no. 4, pp. 781–791, 2003.

[6] N. T. Doncheva, K. Klein, F. S. Domingues, and M. Albrecht, "Analyzing and visualizing residue networks of protein structures," *Trends in biochemical sciences*, vol. 36, no. 4, pp. 179–182, 2011.

[7] E. Estrada, "Universality in protein residue networks," *Biophysical journal*, vol. 98, no. 5, pp. 890–900, 2010.

[8] International Union of Biochemistry. Commission on Enzymes, *Report of the Commission on Enzymes*, ser. I.U.B. Symposium Series, Vol. 20. Oxford: Pergamon Press, 1961.

[9] A. G. McDonald and K. F. Tipton, "Enzyme nomenclature and classification: the state of the art," *The FEBS journal*, vol. 290, no. 9, pp. 2214–2231, 2023.

[10] M. des Jardins, P. D. Karp, M. Krummenacker, T. J. Lee, and C. A. Ouzounis, "Prediction of enzyme classification from protein sequence without the use of sequence similarity." in *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, vol. 5, 1997, pp. 92–99.

[11] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of molecular biology*, vol. 330, no. 4, pp. 771–783, 2003.

[12] ——, "Predicting enzyme class from protein structure without alignments," *Journal of Molecular Biology*, vol. 345, no. 1, pp. 187–199, 2005.

[13] L. Di Paola, M. De Ruvo, P. Paci, D. Santoni, and A. Giuliani, "Protein contact networks: an emerging paradigm in chemistry," *Chemical reviews*, vol. 113, no. 3, pp. 1598–1613, 2013.

[14] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. suppl_1, pp. i47–i56, 2005.

[15] A. R. Atilgan, P. Akan, and C. Baysal, "Small-world communication of residues and significance for protein dynamics," *Biophysical journal*, vol. 86, no. 1, pp. 85–91, 2004.

[16] A. Martino, E. Maiorino, A. Giuliani, M. Giampieri, and A. Rizzi, "Supervised approaches for function prediction of proteins contact networks from topological structure information," in *Scandinavian Conference on Image Analysis*. Springer, 2017, pp. 285–296.

[17] E. De Santis, A. Martino, A. Rizzi, and F. M. Frattale Mascioli, "Dissimilarity space representations and automatic feature selection for protein function prediction," in *2018 International joint conference on neural networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[18] M. Cervellini and A. Martino, "A machine learning approach for physiological role prediction in protein contact networks: a large-scale analysis on the human proteome," *bioRxiv*, 2026.

[19] L. De Ferrari, S. Aitken, J. van Hemert, and I. Goryanin, "Enzml: multi-label prediction of enzyme classes using interpro signatures," *BMC bioinformatics*, vol. 13, no. 1, p. 61, 2012.

[20] Y. Che, Y. Ju, P. Xuan, R. Long, and F. Xing, "Identification of multifunctional enzyme with multi-label classifier," *PloS one*, vol. 11, no. 4, p. e0153503, 2016.

[21] H.-L. Zou and X. Xiao, "Classifying multifunctional enzymes by incorporating three different models into chou's general pseudo amino acid composition," *The Journal of membrane biology*, vol. 249, no. 4, pp. 551–557, 2016.

[22] Z. Zou, S. Tian, X. Gao, and Y. Li, "mldeepre: multi-functional enzyme function prediction with hierarchical multi-label deep learning," *Frontiers in genetics*, vol. 9, p. 714, 2019.

[23] E. Maiorino, A. Rizzi, A. Sadeghian, and A. Giuliani, "Spectral reconstruction of protein contact networks," *Physica A: Statistical Mechanics and its Applications*, vol. 471, pp. 804–817, 2017.

[24] A. Patania, G. Petri, and F. Vaccarino, "The shape of collaborations," *EPJ Data Science*, vol. 6, no. 1, p. 18, 2017.

[25] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie, "Random walks on simplicial complexes and the normalized hodge 1-laplacian," *SIAM Review*, vol. 62, no. 2, pp. 353–391, 2020.

[26] A. Martino and A. Rizzi, "(Hyper)graph kernels over simplicial complexes," *Entropy*, vol. 22, no. 10, p. 1155, 2020.

[27] A. Martino, A. Giuliani, and A. Rizzi, "(Hyper)Graph Embedding and Classification via Simplicial Complexes," *Algorithms*, vol. 12, no. 11, 2019.

[28] A. Zomorodian, "Fast construction of the vietoris-rips complex," *Computers & Graphics*, vol. 34, no. 3, pp. 263 – 271, 2010, shape Modelling International (SMI) Conference 2010.

[29] M. Cervellini, B. Sinaimeri, C. Matias, and A. Martino, "Comparing the ability of embedding methods on metabolic hypergraphs for capturing taxonomy-based features," *bioRxiv*, 2026.

[30] E. Estrada and J. A. Rodríguez-Velázquez, "Subgraph centrality and clustering in complex hyper-networks," *Physica A: Statistical Mechanics and its Applications*, vol. 364, pp. 581–594, 2006.

[31] C. Matias, "A statistical perspective on higher-order interactions modeling," *arXiv preprint arXiv2603.28273*, 2026.

[32] M. Dufrêne and P. Legendre, "Species assemblages and indicator species: the need for a flexible asymmetrical approach," *Ecological monographs*, vol. 67, no. 3, pp. 345–366, 1997.

[33] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.

[34] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," *Journal of Machine Learning Research*, vol. 11, no. 40, pp. 1201–1242, 2010.

[35] T. K. Rusch, M. M. Bronstein, and S. Mishra, "A survey on oversmoothing in graph neural networks," *arXiv preprint arXiv:2303.10993*, 2023.

[36] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4602–4609.

[37] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[38] T. Kipf, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[39] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[40] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.

[41] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, "Graph matching networks for learning the similarity of graph structured objects," in *International conference on machine learning*. PMLR, 2019, pp. 3835–3845.

[42] J. Zhu, S. Rosset, R. Tibshirani, and T. Hastie, "1-norm support vector machines," *Advances in neural information processing systems*, vol. 16, 2003.

[43] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[44] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[45] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[46] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[47] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[48] S. Raschka, "Biopandas: Working with molecular structures in pandas dataframes," *The Journal of Open Source Software*, vol. 2, no. 14, jun 2017.

[49] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski *et al.*, "Biopython: freely available python tools for computational molecular biology and bioinformatics," *Bioinformatics*, vol. 25, no. 11, p. 1422, 2009.

[50] M. Mani, C. Chen, V. Amblee, H. Liu, T. Mathur, G. Zwicke, S. Zabad, B. Patel, J. Thakkar, and C. J. Jeffery, "Moonprot: a database for proteins that are known to moonlight," *Nucleic acids research*, vol. 43, no. D1, pp. D277–D282, 2015.

[51] N. Dessì, G. Milia, and B. Pes, "Enhancing random forests performance in microarray data classification," in *Conference on Artificial Intelligence in Medicine in Europe*. Springer, 2013, pp. 99–103.

[52] B. Peng, L. Wang, and Y. Wu, "An error bound for l1-norm support vector machine coefficients in ultra-high dimension," *Journal of Machine Learning Research*, vol. 17, no. 233, pp. 1–26, 2016.