FROM PROTEIN REPRESENTATIONS TO FUNCTIONAL HIERARCHIES: RESHAPING REPRESENTATION GEOMETRY FOR ENZYME CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Enzyme Commission (EC) classification assigns protein sequences to hierarchically organized functional classes and is important for biotechnology. Fine-grained EC classification remains challenging, and errors can compromise downstream functional annotation and enzyme discovery. Despite increasingly powerful protein representations, existing methods still insufficiently model the representation geometry of fine-grained EC classes: at the inter-class level, sibling classes can remain poorly separated; at the intra-class level, proteins within the same class can form heterogeneous distributions poorly captured by a single center. To address these challenges, we propose HiPro-EC, a hierarchy-guided multi-prototype framework that reshapes this geometry over frozen protein language model (PLM) representations. It reshapes inter-class structure by increasing sibling-class separation using the EC hierarchy, and models intra-class structure with multiple local prototypes to capture heterogeneous class distributions. Extensive experiments on the CARE benchmark show that HiPro-EC achieves the best performance at the finest EC level across all four official test sets, with consistent gains across multiple PLM backbones. Further analyses show clearer sibling-class separation and larger gains for heterogeneous EC classes, highlighting the value of modeling representation geometry for fine-grained EC classification.

1 INTRODUCTION

Enzyme function prediction aims to infer biochemical functions from protein sequences and is fundamental to large-scale protein annotation, enzyme discovery, and functional genomics (Minshull et al., 2005; Ariaeenejad et al., 2024; Lin et al., 2024). The Enzyme Commission (EC) system organizes enzyme functions into a four-level hierarchy with increasingly specific catalytic activities (McDonald et al., 2009). Fine-grained EC classification therefore requires distinguishing closely related functional classes, where proteins sharing the same higher-level class may still differ in catalytic activity or substrate preference. Effective discrimination of such subtle differences depends on informative protein representations. Recent protein language models (PLMs) have substantially improved sequence-based protein representations through large-scale pretraining (Rives et al., 2021). Despite this progress, resolving fine-grained enzyme functions remains challenging.

Protein language models are pretrained on large-scale unlabeled protein sequences and therefore capture multiple sources of biological variation rather than enzyme function alone. Their representations encode evolutionary and homology-related information (Liu et al., 2024). As a result, the representation space can reflect sequence ancestry and family relationships, whereas EC labels organize proteins according to catalytic function; the two organizations are related but not necessarily identical (Tian & Skolnick, 2003; Riziotis et al., 2025). For fine-grained EC classification, this mismatch can manifest in the geometry of EC classes at both the inter-class and intra-class levels: at the inter-class level, closely related sibling classes may remain insufficiently separated, while at the intra-class level, proteins sharing the same EC label may form heterogeneous distributions. As illustrated in Fig. 1(a), these two aspects jointly increase the difficulty of fine-grained discrimination, making proteins from heterogeneous classes more susceptible to misclassification into nearby sibling classes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

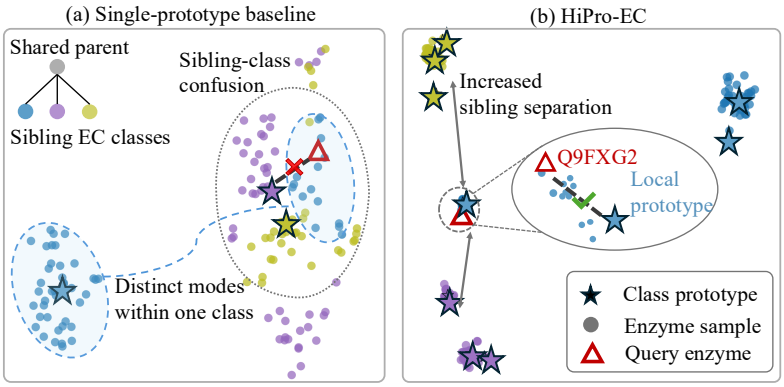

Figure 1: Visualization of enzyme representation spaces. (a) A single-prototype baseline exhibits sibling-class confusion and multiple modes within the same EC class. (b) HiPro-EC improves sibling separation and represents heterogeneous class structure with multiple local prototypes. The query enzyme (red triangle) is misclassified as the purple sibling class in (a), but correctly assigned to the blue target class in (b) by prototype similarity.

To address these challenges, we propose **HiPro-EC**, a hierarchy-guided multi-prototype framework that reshapes the geometry of fine-grained EC classes in the representation space. Hierarchy-Guided Geometry (HG) uses the EC hierarchy to initialize a projection that improves separation among related sibling classes. Multi-Prototype (MP) modeling represents each fine-grained EC class with multiple learnable local prototypes to capture heterogeneous intra-class structure. Together, these components reduce the tendency for proteins from heterogeneous classes to be misclassified as nearby sibling classes, improving fine-grained EC classification. The pretrained protein encoder remains frozen, and the framework requires only protein sequences. Extensive experiments on CARE (Yang et al., 2024a) show that HiPro-EC consistently outperforms strong baselines in fine-grained EC classification across diverse test sets and PLM backbones. Further analyses confirm improved sibling-class separation and larger gains for heterogeneous EC classes.

Our main contributions are summarized as follows:

- We introduce a hierarchy-guided geometry initialization that exploits hierarchical relationships among EC classes to improve geometric separation among closely related sibling classes.
- We incorporate multiple local prototypes to capture heterogeneous distributions within EC classes, allowing distinct local modes of the same function to be represented separately.
- Extensive evaluation on CARE shows that HiPro-EC achieves the best performance at the finest EC level across all four test sets, with gains that generalize across different PLM backbones.

2 Related Work

2.1 Enzyme Function Prediction

Early computational enzyme annotation mainly relied on sequence homology and profile-based methods, such as BLAST (Altschul et al., 1990) and profile HMMs (Eddy, 1998). Subsequent sequence-based predictors, including ECPred (Dalkiran et al., 2018), DeepEC (Ryu et al., 2019), and ProteInfer (Sanderson et al., 2023), introduced task-specific learning for enzyme function prediction. With the emergence of protein language models, general-purpose models such as ESM-2 (Lin et al., 2023), ProtT5 (Elnaggar et al., 2022), and ESM-C (Candido et al., 2026) have provided increasingly powerful protein sequence representations. CLEAN (Yu et al., 2023) builds on pretrained protein representations and uses contrastive learning for enzyme annotation, whereas EnzGFM (Wang et al., 2026) develops an enzyme-specific PLM through hierarchical pretraining on general protein and enzyme sequences. However, fine-grained EC classes do not form an independent flat label space, and effective discrimination also depends on how EC class structure is reflected in the representation space. In existing approaches, such structure is largely induced implicitly from training data

and learning objectives, which does not necessarily yield a representation space well suited to fine-grained EC discrimination. We therefore explicitly model EC class structure over frozen pretrained representations. Additional related methods are discussed in Appendix A.

2.2 Prototype-Based Class Representation Learning

Prototype-based learning represents classes using representative embeddings and performs prediction according to their similarity to query samples. Prototypical Networks (Snell et al., 2017) introduced the widely used single-prototype formulation. More flexible class representations have since been developed: CBC (Saralajew et al., 2025) uses component-based prototypes, DNC (Wang et al., 2023) represents each class with multiple sub-centroids, and ClusPro (Qu et al., 2025) constructs diversified prototypes through clustering. Related multi-prototype strategies have also been explored in semantic segmentation (Zhou et al., 2022), named entity recognition (Wu et al., 2023), and continual domain adaptation (Sun et al., 2025). Beyond intra-class modeling, structured label relationships have also been used to organize class representations for fine-grained classification (Li et al., 2022; Lim et al., 2026).

Prototype-based representations have also been explored in biological prediction. MIMML (He et al., 2022) and CoraL (Li et al., 2023) use prototypical learning for peptide prediction, while BioBridge (Lian et al., 2025) employs dynamic prototype learning for drug–target interaction prediction. More recently, ProtoMol (Wang et al., 2025b) introduces prototype-guided representations for molecular property prediction. These studies demonstrate the value of explicitly modeling class representations, but this perspective remains less explored in fine-grained enzyme classification, where functional classes are both hierarchically related and internally diverse.

3 Methods

In this section, we present HiPro-EC for fine-grained EC classification. As illustrated in Fig. 2, HiPro-EC first initializes the projection space using Hierarchy-Guided Geometry (HG) (Section 3.1), and then models each fine-grained EC class with multiple local prototypes through Multi-Prototype (MP) representation learning (Section 3.2).

3.1 Hierarchy-Guided Geometry Initialization

In this section, we introduce Hierarchy-Guided Geometry (HG) to initialize the linear projection head using class statistics defined along the EC hierarchy. By aggregating parent–child variations across hierarchy levels, HG constructs a projection that emphasizes hierarchy-related differences while controlling within-class variation.

Hierarchical Variations. For a class c at hierarchy level l, let $p(c)$ denote its parent class. HG characterizes the functional variation of each class relative to its parent. Let $\mathbf{x}_i \in \mathbb{R}^d$ denote the feature representation of protein i. The centroid of class c is

$$\boldsymbol{\mu}_c^{(l)} = \frac{1}{|\mathcal{I}_c^{(l)}|} \sum_{i \in \mathcal{I}_c^{(l)}} \mathbf{x}_i, \tag{1}$$

where $\mathcal{I}_c^{(l)}$ denotes the set of training samples assigned to class c.

Let $p(c)$ denote the parent class of c. We characterize the hierarchical variation of class c by its offset from the corresponding parent:

$$\boldsymbol{\delta}_c^{(l)} = \boldsymbol{\mu}_c^{(l)} - \boldsymbol{\mu}_{p(c)}^{(l-1)}. \tag{2}$$

For EC1 classes, the root node is represented by the global mean of all training features. Thus, $\boldsymbol{\delta}_c^{(l)}$ describes the parent-to-child variation associated with class c.

Geometry Construction. Only classes whose parent has at least two observed child classes are included, since a parent with a single observed child provides no sibling-level discrimination. For

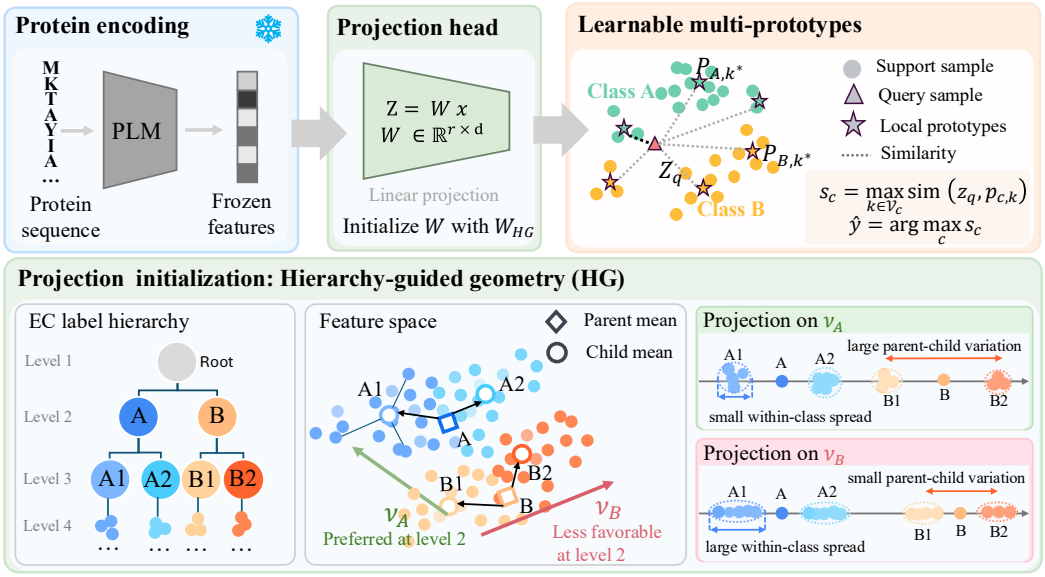

Figure 2: Overview of HiPro-EC. Hierarchy-Guided Geometry (HG) initializes the linear projection by favoring directions with large parent–child variation and small within-class dispersion. Multi-prototype representation models each EC4 class with multiple learnable local prototypes for similarity-based prediction.

each hierarchy level, the parent-relative offsets are aggregated into

$$\widetilde{\mathbf{S}}_l = \sum_{c \in \mathcal{C}_l} n_c \, \boldsymbol{\delta}_c^{(l)} \boldsymbol{\delta}_c^{(l)\top}, \tag{3}$$

where $\mathcal{C}_l$ denotes the included classes at level l, and n_c weights each child class by its training support.

To prevent hierarchy levels with larger scatter magnitudes from dominating the projection, we normalize each level-wise scatter by its trace before aggregation. The hierarchy-guided scatter matrix is defined as

$$\mathbf{S}_H = \sum_{l=1}^{L} \mathbf{S}_l, \qquad \mathbf{S}_l = \frac{\widetilde{\mathbf{S}}_l}{\mathrm{tr}(\widetilde{\mathbf{S}}_l)}. \tag{4}$$

Thus, $\mathbf{S}_H$ aggregates parent–child variations from all hierarchy levels on a comparable scale.

In parallel, we measure within-class variation at the finest hierarchy level. Let $\mathcal{C}_L$ denote the set of EC4 classes in the training set. The within-class scatter matrix is defined as

$$\mathbf{S}_W = \sum_{c \in \mathcal{C}_L} \sum_{i \in \mathcal{I}_c^{(L)}} \left(\mathbf{x}_i - \boldsymbol{\mu}_c^{(L)}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_c^{(L)}\right)^\top. \tag{5}$$

A small ridge term is added to $\mathbf{S}_W$ for numerical stability, yielding the regularized matrix $\mathbf{S}_W^{\mathrm{reg}}$ used below.

Thus, S_H captures hierarchy-related inter-class variation, whereas S_W captures within-class variation at the EC4 level.

Projection Initialization. Following the classical discriminant criterion (Fisher, 1936), we seek projection directions along which hierarchy-related variation is large while EC4 within-class variation remains small. Unlike conventional discriminant analysis, the between-class term is replaced by the hierarchy-guided scatter constructed from parent–child relations.

This objective can be written as the generalized Rayleigh quotient

$$\mathbf{v}^* = \arg\max_{\mathbf{v} \neq \mathbf{0}} \frac{\mathbf{v}^\top \mathbf{S}_H \mathbf{v}}{\mathbf{v}^\top \mathbf{S}_W^{\mathrm{reg}} \mathbf{v}}. \tag{6}$$

The numerator measures the hierarchical variation preserved along direction $\mathbf{v}$, whereas the denominator measures the corresponding within-class dispersion at the EC4 level. Therefore, directions with a larger ratio are preferred by HG, as illustrated on the right of Fig. 2.

The above objective leads to the generalized eigenvalue problem

$$\mathbf{S}_H \mathbf{v} = \gamma \mathbf{S}_W^{\text{reg}} \mathbf{v}. \tag{7}$$

We select the eigenvectors associated with the largest generalized eigenvalues and use them to initialize the linear projection matrix:

$$W_{\text{HG}} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r]^\top. \tag{8}$$

HG is used only to initialize the projection geometry. During subsequent training, the full projection matrix remains trainable and is jointly optimized with the prototype-based classifier under the standard classification objective.

3.2 Multi-Prototype Representation Learning

In this section, we introduce Multi-Prototype (MP) representation learning to model intra-class heterogeneity in fine-grained EC classes. Instead of representing each EC4 class with a single global class center, MP uses multiple learnable local prototypes to capture different regions of the class distribution.

Local Prototype Construction. For each EC4 class c, let n_c denote the number of training samples in the class. We use a global prototype budget K and set the actual number of prototypes for class c as

$$K_c = \min(K, n_c). \tag{9}$$

Thus, all classes share the same maximum prototype budget, while low-support classes use fewer prototypes when necessary. We then maintain a set of K_c learnable prototypes:

$$\mathcal{P}_c = \{\mathbf{p}_{c,1}, \mathbf{p}_{c,2}, \ldots, \mathbf{p}_{c,K_c}\}. \tag{10}$$

where each prototype $p_{c,k}$ represents a local mode of class c in the projected feature space.

We initialize these prototypes by applying spherical K-means (Dhillon & Modha, 2001) independently to the HG-projected training features of each EC4 class. Because both clustering and subsequent prototype matching rely on angular similarity, this initialization is consistent with the cosine-based classification geometry. The prototypes remain learnable and are jointly optimized with the projection head during downstream training.

Prototype-Based Matching. Given a query protein with projected representation $\mathbf{z}_q$, we compute its cosine similarity to every prototype of class c:

$$s_{c,k} = \text{sim}\left(\mathbf{z}_q, \mathbf{p}_{c,k}\right), \tag{11}$$

where the similarity function is defined as

$$\text{sim}\left(\mathbf{z}, \mathbf{p}\right) = \frac{\mathbf{z}^\top \mathbf{p}}{\|\mathbf{z}\|_2 \|\mathbf{p}\|_2}. \tag{12}$$

Instead of averaging similarities over all prototypes, the score of class c is determined by its most compatible local prototype:

$$s_c = \max_{k \in \{1,\ldots,K_c\}} s_{c,k} = \max_{k \in \{1,\ldots,K_c\}} \text{sim}\left(\mathbf{z}_q, \mathbf{p}_{c,k}\right). \tag{13}$$

As shown in Fig. 2, only the locally best-matched prototype contributes to the class score. This max-based matching allows a query to interact with the most relevant mode of each EC class, rather than being compared against an averaged global representation or being influenced by unrelated local modes.

Training Objective. Let Y denote the set of valid EC4 labels for a protein. We optimize the projection head and prototypes using a multi-positive softmax cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{|Y|} \sum_{c \in Y} \log \frac{\exp(s_c)}{\sum_{j \in \mathcal{C}_L} \exp(s_j)}. \tag{14}$$

For single-label proteins, this reduces to the standard softmax cross-entropy loss. For multi-label proteins, all valid EC labels contribute equally to the objective.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on Task 1 of the CARE benchmark (Yang et al., 2024a), which evaluates enzyme function prediction under the four-level Enzyme Commission (EC) hierarchy. We use the sequence-redundancy-reduced training set containing approximately 28K proteins, constructed by clustering training sequences at 50% sequence identity to reduce highly similar training examples while preserving the functional label space. Models are evaluated on four official test sets covering different challenging scenarios for enzyme function prediction.

- $< 30\%$ **Identity.** This test set contains proteins with less than 30% sequence identity to the training set and evaluates generalization to remotely related sequences.
- 30–50% **Identity.** This test set contains proteins with 30–50% sequence identity to the training set and evaluates prediction under moderate sequence similarity.
- **Previously Misclassified (Price).** This test set consists of enzymes that were previously misannotated and later corrected based on experimental evidence, representing challenging cases for function prediction (Price et al., 2018).
- **Promiscuous.** This test set contains enzymes associated with multiple EC numbers and evaluates the ability to predict multiple valid functions for a single protein.

Implementation Details. We use ESM-2-650M (Lin et al., 2023) as the frozen protein encoder and obtain sequence-level representations by mean pooling. HiPro-EC uses an HG-initialized linear projection and learnable prototypes initialized by spherical K-means, with a maximum prototype budget of $K = 9$. The same configuration is used across all four evaluation settings; full implementation details are provided in Appendix B.2. HiPro-EC results in the main tables are averaged over three independent runs; standard deviations are reported in Appendix D.1.

Evaluation Metrics. Following CARE (Yang et al., 2024a), we use the official accuracy score, reported as a percentage (%), at all four EC hierarchy levels. For an EC annotation such as 1.2.3.4, Levels 1–4 correspond to 1.-.-.-, 1.2.-.-, 1.2.3.-, and 1.2.3.4, respectively. For each protein, we select the top-$|Y|$ predicted EC numbers, where $|Y|$ is the number of ground-truth annotations. At each level, the score is the fraction of ground-truth annotations matched by these predictions, averaged over all proteins in the test set.

Baselines. We compare HiPro-EC with representative sequence-based enzyme prediction methods, including BLASTp (Altschul et al., 1990), CLEAN (Yu et al., 2023), ESM-2 (Lin et al., 2023), ProtT5 (Elnaggar et al., 2022), ESM-C (Candido et al., 2026), and EnzGFM (Wang et al., 2026). We further adapt CBC (Saralajew et al., 2025), DNC (Wang et al., 2023), and ClusPro (Qu et al., 2025), which were originally developed for other classification settings and construct class representations using prototypes or centroids. For controlled analyses, we additionally use a single-prototype variant, where each EC4 class is represented by one central prototype under the same training objective as HiPro-EC. Unless otherwise specified, we refer to this variant as the baseline in the following analyses. All adapted baselines use the same frozen ESM-2 representations, CARE training split, and evaluation protocol as HiPro-EC. Detailed settings are provided in Appendix C.

Table 1: Accuracy score (%) comparison on the CARE low-identity test sets. Bold and underlined values denote the best and second-best results, respectively.

Method	$< 30\%$ **Identity**				30–50% **Identity**			
	L1 (x.-.-.-)	**L2** (x.x.-.-)	**L3** (x.x.x.-)	**L4** (x.x.x.x)	**L1** (x.-.-.-)	**L2** (x.x.-.-)	**L3** (x.x.x.-)	**L4** (x.x.x.x)
BLASTp	62.7	58.3	56.7	47.7	90.0	87.7	84.8	77.1
CLEAN	85.2	75.5	69.7	52.8	95.2	91.3	85.9	76.3
ESM-2	80.3	71.8	68.3	55.1	94.3	90.4	87.1	81.0
ProtT5	81.7	70.8	65.7	53.9	95.5	90.5	87.0	80.2
ESM-C	73.8	65.3	60.6	49.1	93.6	87.5	84.5	78.6
EnzGFM	85.0	75.7	70.1	53.2	94.6	91.3	86.1	75.4
CBC	75.5	64.8	61.1	48.8	93.6	87.3	83.9	77.3
DNC	86.1	77.8	<u>72.2</u>	<u>58.8</u>	**96.3**	<u>92.0</u>	<u>88.2</u>	80.2
ClusPro	<u>87.7</u>	<u>78.0</u>	<u>72.2</u>	58.1	95.7	91.6	<u>88.2</u>	<u>81.4</u>
HiPro-EC	**89.6**	**82.6**	**77.7**	**64.5**	<u>96.0</u>	**92.7**	**88.8**	**83.0**

Table 2: Accuracy score (%) comparison on the CARE Price and Promiscuous test sets. Bold and underlined values denote the best and second-best results, respectively.

Method	**Price**				**Promiscuous**			
	L1 (x.-.-.-)	**L2** (x.x.-.-)	**L3** (x.x.x.-)	**L4** (x.x.x.x)	**L1** (x.-.-.-)	**L2** (x.x.-.-)	**L3** (x.x.x.-)	**L4** (x.x.x.x)
BLASTp	77.7	77.7	68.9	33.8	82.7	77.8	72.8	68.0
CLEAN	<u>93.8</u>	86.1	80.6	40.6	86.2	79.8	75.6	63.2
ESM-2	89.6	<u>88.9</u>	<u>83.3</u>	35.8	90.0	<u>86.5</u>	<u>82.6</u>	74.1
ProtT5	91.7	<u>88.9</u>	81.9	34.4	<u>90.9</u>	86.3	<u>82.6</u>	73.2
ESM-C	85.4	79.9	75.7	29.9	89.3	83.8	78.6	71.3
EnzGFM	93.1	84.0	75.7	38.5	87.7	81.3	77.9	66.5
CBC	91.7	84.7	79.2	<u>44.1</u>	85.1	77.7	73.0	64.2
DNC	93.1	**89.6**	<u>83.3</u>	40.6	89.0	85.9	82.0	<u>75.6</u>
ClusPro	89.6	84.0	78.5	39.2	84.2	78.4	73.0	66.8
HiPro-EC	**97.2**	**89.6**	**84.0**	**45.3**	**92.5**	**89.0**	**86.6**	**80.7**

4.2 Main Results

Comparison with Existing Methods. Tables 1 and 2 compare HiPro-EC with existing sequence-based methods and adapted class-representation approaches. HiPro-EC achieves the best Level-4 performance across all four CARE test sets. Notably, the strongest competing Level-4 results are consistently obtained by class-representation methods such as DNC, ClusPro, and CBC, highlighting the importance of class representation for fine-grained EC classification. The further improvements of HiPro-EC over these generic approaches indicate that the specific design of class representations matters for fine-grained EC classification.

A further pattern is that the advantage of HiPro-EC becomes more evident at finer EC levels. On the $< 30\%$ Identity split, its margin over the strongest competing method increases from 1.9 percentage points at Level 1 to 5.7 at Level 4; on Promiscuous, it increases from 1.6 to 5.1 percentage points. The 30–50% Identity split shows a similar trend: HiPro-EC is 0.3 percentage points below the best baseline at Level 1 but 1.6 above it at Level 4. Price shows a different trend. This split contains previously misclassified enzymes, many of which may lie near activity cliffs (Yang et al., 2024a). All methods show a sharp drop from Level 3 to Level 4, indicating that exact EC4 assignment is particularly difficult. Even in this setting, HiPro-EC achieves the best Level-4 performance.

Generalization Across Protein Language Models. To examine whether HiPro-EC depends on a specific protein language model, we further evaluate it with ProtT5, ESM-C, and EnzGFM. Figure 3 compares HiPro-EC with the corresponding single-prototype baseline under the same training set-

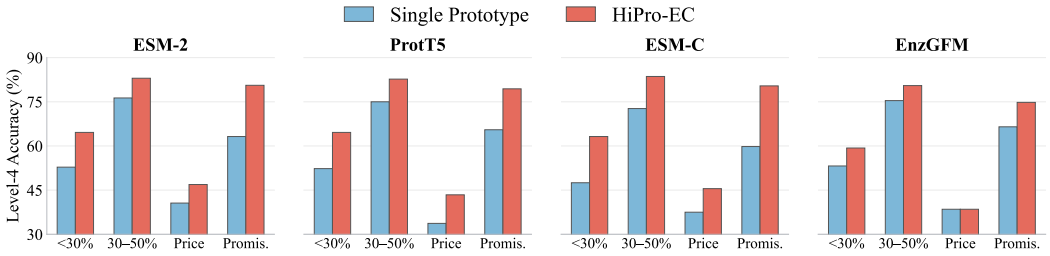

Figure 3: Level-4 accuracy score (%) across PLM backbones. For each backbone, HiPro-EC is compared with the corresponding single-prototype baseline under the same training setting.

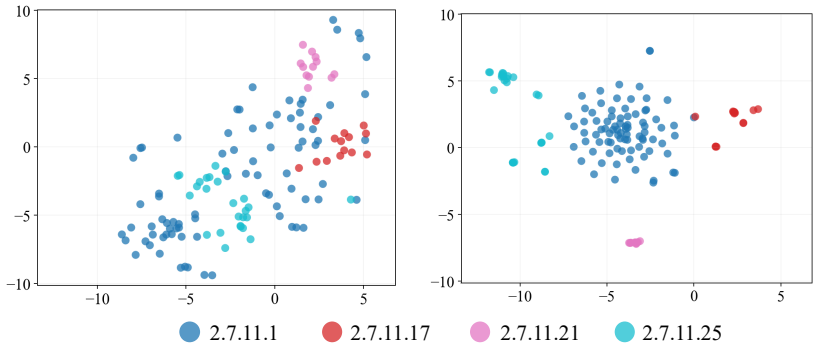

Figure 4: t-SNE visualization of sibling EC4 classes under the same EC3 parent (2.7.11.-). Compared with the baseline (left), HiPro-EC (right) yields clearer fine-grained class separation.

ting for each backbone. HiPro-EC improves Level-4 performance across nearly all backbone–split combinations, with particularly large gains on ESM-C. For the enzyme-specific EnzGFM, HiPro-EC improves three of the four splits while leaving Price unchanged. The smaller gains on EnzGFM may reflect its enzyme-oriented pretraining, which already encourages function-relevant representations and therefore leaves less room for downstream reorganization.

4.3 Ablation Study

We conduct the ablation study on the $< 30\%$ Identity test set to evaluate the contributions of hierarchy-guided geometry (HG) and multi-prototype (MP) modeling. The baseline removes both components and represents each EC4 class with a single central prototype under the same training objective as HiPro-EC. As shown in Table 3, removing either component leads to consistent performance degradation across the EC hierarchy. At Level 4, introducing MP improves the accuracy score from 52.8% to 63.4%, corresponding to a gain of 10.6 percentage points, while HG alone improves the score to 58.8%. Combining HG and MP further increases the Level-4 score to 64.5%, outperforming either component alone and demonstrating the benefit of integrating the two components.

Table 3: Ablation study results.

Variant	L1	L2	L3	L4
Full	**89.6**	**82.6**	**77.7**	**64.5**
w/o HG	88.1	81.0	75.8	63.4
w/o MP	87.5	78.9	73.4	58.8
w/o HG & MP	85.2	75.5	69.7	52.8

4.4 Representation Space Analysis

Sibling-Class Representation Geometry. To examine the geometry of closely related EC classes in the representation space, we visualize four sibling EC4 classes sharing the same EC3 parent (2.7.11.-) using t-SNE in Fig. 4. The baseline exhibits substantial overlap among these sibling classes, particularly between EC 2.7.11.1 and EC 2.7.11.25. In contrast, HiPro-EC yields clearer

Table 4: Multi-prototype gains across intra-class heterogeneity levels. Each cell shows single-prototype (Single) → multi-prototype (MP) accuracy score (%), with the change in percentage points in parentheses.

Heterogeneity	$<30\%$ **Identity**	30–50% **Identity**	**Price**	**Promiscuous**
Low	74.4 → 74.4 (+0.0)	89.6 → 87.8 (−1.8)	50.0 → 39.6 (−10.4)	78.8 → 73.5 (−5.3)
Medium	76.6 → 78.7 (+2.1)	91.6 → 92.6 (+1.0)	25.9 → 41.5 (+15.6)	79.4 → 82.4 (+3.0)
High	55.7 → 67.6 **(+11.9)**	72.9 → 87.7 **(+14.8)**	9.1 → 41.8 **(+32.7)**	53.0 → 73.4 **(+20.4)**

separation among the four EC4 classes. This visualization provides qualitative evidence that HiPro-EC reorganizes the representation space in a way that better separates fine-grained sibling classes. Additional visualizations covering multiple EC3 parent groups are provided in Appendix D.2.

Intra-Class Heterogeneity. To examine when multi-prototype (MP) modeling is most beneficial, we group EC4 classes into low-, medium-, and high-heterogeneity subsets based on feature-space dispersion; the exact definition is provided in Appendix D.3. As shown in Table 4, high-heterogeneity classes are consistently harder to classify and benefit most from MP. Across the four test sets, MP improves this group by 11.9–32.7 percentage points, substantially larger than the gains observed for the lower-heterogeneity groups. Compact low-heterogeneity classes benefit little and can occasionally degrade. The low- and medium-heterogeneity groups show less consistent differences, partly because limited class support can restrict the reliable estimation of multiple local modes; this effect is further analyzed in Appendix D.4. These results support our motivation that multiple local prototypes are most useful when a single prototype is insufficient to represent a heterogeneous class distribution.

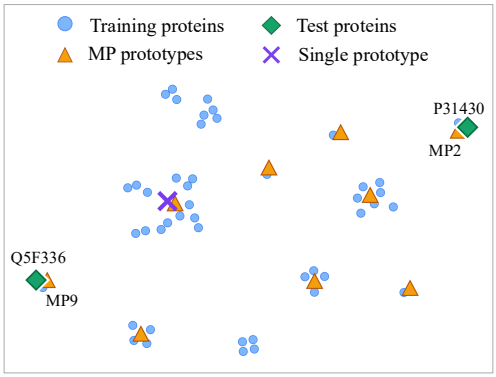

Figure 5: Case study on EC 3.5.2.6.

Case Study. We further examine EC 3.5.2.6, which is assigned to the high-heterogeneity group by the analysis above. Its 46 training proteins form 20 clusters at 30% sequence identity using MMseqs2 (Steinegger & Söding, 2017), reflecting considerable sequence diversity within the class. Figure 5 visualizes the representation space of this class, where the training proteins span several regions and the two test proteins, Q5F336 and P31430, fall in different parts of the class distribution. In the single-prototype baseline, one global prototype must summarize this dispersed distribution, and both test proteins are located away from it in the visualization. The baseline misses the target EC label for both proteins. In contrast, MP provides local prototypes near the two queries in different regions, and correctly recovers the target label for both. This case shows that MP can more flexibly represent proteins sharing the same enzymatic function despite substantial sequence diversity.

5 Conclusion

We present HiPro-EC, a hierarchy-guided multi-prototype framework for fine-grained EC classification that reorganizes EC class geometry at both the inter-class and intra-class levels. Hierarchy-guided geometry improves discrimination among closely related classes, while multiple local prototypes capture heterogeneous modes within individual classes. Experiments on CARE show consistent improvements across the EC hierarchy and evaluation settings, with particularly clear gains in fine-grained discrimination and heterogeneous EC classes; these gains also generalize across multiple PLM backbones. Overall, our results highlight the importance of organizing downstream representation space, beyond improving pretrained representations, for EC classification.

AI USE STATEMENT

We used an LLM (ChatGPT) to assist with language editing, improve the clarity of presentation, and support minor visual refinement of figures. It was also used to discuss the presentation and wording of already-computed experimental results. No generative AI tools were used to generate scientific data, implement experiments, or modify experimental results. All scientific claims, analyses, figures, and conclusions were verified by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

Shohreh Ariaeenejad, Javad Gharechahi, Mehdi Foroozandeh Shahraki, Fereshteh Fallah Atanaki, Jian-Lin Han, Xue-Zhi Ding, Falk Hildebrand, Mohammad Bahram, Kaveh Kavousi, and Ghasem Hosseini Salekdeh. Precision enzyme discovery through targeted mining of metagenomic data. *Natural Products and Bioprospecting*, 14(1):7, 2024.

Nicolas Buton, François Coste, and Yann Le Cunff. Predicting enzymatic function of protein sequences with attention. *Bioinformatics*, 39(10):btad620, 2023.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Z. Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akın, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language modeling materializes a world model of protein biology. *bioRxiv*, 2026.

Alperen Dalkiran, Ahmet Sureyya Rifaioglu, Maria Jesus Martin, Rengul Cetin-Atalay, Volkan Atalay, and Tunca Doğan. ECPred: a tool for the prediction of the enzymatic functions of protein sequences based on the EC nomenclature. *BMC Bioinformatics*, 19(1):334, 2018.

Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

Louis Dumontet, So-Ra Han, Jun Hyuck Lee, Tae-Jin Oh, and Mingon Kang. Trustworthy prediction of enzyme commission numbers using a hierarchical interpretable transformer. *Nature Communications*, 17:1146, 2026.

Sean R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022.

Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 (2):179–188, 1936.

So-Ra Han, Mingyu Park, Sai Kosaraju, JeungMin Lee, Hyun Lee, Jun Hyuck Lee, Tae-Jin Oh, and Mingon Kang. Evidential deep learning for trustworthy prediction of enzyme commission number. *Briefings in Bioinformatics*, 25(1):bbad401, 2024.

Wenjia He, Yi Jiang, Junru Jin, Zhongshen Li, Jiaojiao Zhao, Balachandran Manavalan, Ran Su, Xin Gao, and Leyi Wei. Accelerating bioactive peptide discovery via mutual information-based meta-learning. *Briefings in Bioinformatics*, 23(1):bbab499, 2022.

Yiran Huang, Yufu Lin, Wei Lan, Cuiyu Huang, and Cheng Zhong. GloEC: a hierarchical-aware global model for predicting enzyme function. *Briefings in Bioinformatics*, 25(5):bbae365, 2024.

Gi Bae Kim, Ji Yeon Kim, Jong An Lee, Charles J. Norsigian, Bernhard O. Palsson, and Sang Yup Lee. Functional annotation of enzyme-encoding genes using deep learning with transformer layers. *Nature Communications*, 14:7370, 2023.

Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. Deep hierarchical semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1246–1257, 2022.

Zhongshen Li, Junru Jin, Wenjia He, Wentao Long, Haoqing Yu, Xin Gao, Kenta Nakai, Quan Zou, and Leyi Wei. CoraL: interpretable contrastive meta-learning for the prediction of cancer-associated ncrna-encoded small peptides. *Briefings in Bioinformatics*, 24(6):bbad352, 2023.

Xiaoqing Lian, Tianxu Lv, Jie Zhu, Shiyun Nie, Hang Fan, Guosheng Wu, Yunjun Ge, Hong Xu, Xiaoting Wang, Lihua Li, Xiangxiang Zeng, and Xiang Pan. Inductive-associative meta-learning pipeline with human cognitive patterns for unseen drug-target interaction prediction. *Advanced Science*, 12(35):e06404, 2025.

Seoha Lim, Jinmyeong Kim, Jieun Kim, and Sung-Bae Cho. Hierarchical prototype learning for semantic segmentation. In *International Conference on Learning Representations*, 2026.

Baohui Lin, Xiaoling Luo, Yumeng Liu, and Xiaopeng Jin. A comprehensive review and comparison of existing computational methods for protein function prediction. *Briefings in Bioinformatics*, 25(4):bbae289, 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan Dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Wei Liu, Ziye Wang, Ronghui You, Chenghan Xie, Hong Wei, Yi Xiong, Jianyi Yang, and Shanfeng Zhu. PLMSearch: Protein language model powers accurate and fast sequence search for remote homology. *Nature Communications*, 15:2775, 2024.

Xiang Ma, Parnal Joshi, Iddo Friedberg, and Qi Li. How not to be seen: predicting unseen enzyme functions using contrastive learning. *Bioinformatics*, 42(Supplement_1):btag215, 2026.

Andrew G. McDonald, Sinéad Boyce, and Keith F. Tipton. ExplorEnz: the primary source of the IUBMB enzyme list. *Nucleic Acids Research*, 37(suppl_1):D593–D597, 2009.

Jeremy Minshull, Jon E. Ness, Claes Gustafsson, and Sridhar Govindarajan. Predicting enzyme function from protein sequence. *Current Opinion in Chemical Biology*, 9(2):202–209, 2005.

Morgan N Price, Kelly M Wetmore, R Jordan Waters, Mark Callaghan, Jayashree Ray, Hualan Liu, Jennifer V Kuehl, Ryan A Melnyk, Jacob S Lamson, Yumi Suh, et al. Mutant phenotypes for thousands of bacterial genes of unknown function. *Nature*, 557(7706):503–509, 2018.

Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. Learning clustering-based prototypes for compositional zero-shot learning. In *International Conference on Learning Representations*, 2025.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Ioannis G. Riziotis, Jenny C. Kafas, Gabriel Ong, Neera Borkakoti, António J. M. Ribeiro, and Janet M. Thornton. Paradigms of convergent evolution in enzymes. *The FEBS Journal*, 292(3): 537–555, 2025.

Jae Yong Ryu, Hyun Uk Kim, and Sang Yup Lee. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. *Proceedings of the National Academy of Sciences*, 116(28):13996–14001, 2019.

Theo Sanderson, Maxwell L. Bileschi, David Belanger, and Lucy J. Colwell. ProteInfer, deep neural networks for protein functional inference. *eLife*, 12:e80942, 2023.

Sascha Saralajew, Ashish Rana, Thomas Villmann, and Ammar Shaker. A robust prototype-based network with interpretable RBF classifier foundations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20273–20282, 2025.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4077–4087, 2017.

Yidong Song, Qianmu Yuan, Sheng Chen, Yuansong Zeng, Huiying Zhao, and Yuedong Yang. Accurately predicting enzyme functions through geometric graph learning on ESMFold-predicted structures. *Nature Communications*, 15:8180, 2024.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017.

Haopeng Sun, Yingwei Zhang, Lumin Xu, Sheng Jin, Ping Luo, Chen Qian, Wentao Liu, and Yiqiang Chen. Unsupervised continual domain shift learning with multi-prototype modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10131–10141, 2025.

Weidong Tian and Jeffrey Skolnick. How well is enzyme function conserved as a function of pairwise sequence identity? *Journal of Molecular Biology*, 333(4):863–882, 2003.

Karel van der Weg, Erinc Merdivan, Marie Piraud, and Holger Gohlke. TopEC: prediction of enzyme commission classes by 3D graph neural networks and localized 3D protein descriptor. *Nature Communications*, 16:2737, 2025.

Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, Martin Steinegger, et al. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42:243–246, 2024.

Chong Wang, Mengyao Li, Shaolei Geng, Weidong Li, Xuezhi Zhou, Yu Guang Wang, Yi Yu, Tianyun Wang, and Yiqing Shen. An enzyme-specific protein language model for catalytic property prediction. *Nature Communications*, 17:8760, 2026.

Duolin Wang, Mahdi Pourmirzaei, Usman L. Abbas, Shuai Zeng, Negin Manshour, Farzaneh Esmaili, Biplab Poudel, Yuexu Jiang, Qing Shao, Jin Chen, and Dong Xu. S-PLM: Structure-aware protein language model via contrastive learning between sequence and structure. *Advanced Science*, 12(5):2404212, 2025a.

Wenguan Wang, Cheng Han, Tianfei Zhou, and Dongfang Liu. Visual recognition with deep nearest centroids. In *International Conference on Learning Representations*, 2023.

Yingxu Wang, Kunyu Zhang, Jiaxin Huang, Nan Yin, Siwei Liu, and Eran Segal. ProtoMol: enhancing molecular property prediction via prototype-guided multimodal learning. *Briefings in Bioinformatics*, 26(6):bbaf629, 2025b.

Shuhui Wu, Yongliang Shen, Zeqi Tan, Wenqi Ren, Jietian Guo, Shiliang Pu, and Weiming Lu. MProto: Multi-prototype network with denoised optimal transport for distantly supervised named entity recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2361–2374, 2023.

Kun Xie, Peng Zhou, Xingyi Zhang, Wei Liu, Peilin Zhao, Sibo Wang, and Biaobin Jiang. PoinnCARE: Hyperbolic multi-modal learning for enzyme classification. In *International Conference on Learning Representations*, 2026.

Jason Yang, Ariane Mora, Shengchao Liu, Bruce J. Wittmann, Anima Anandkumar, Frances H. Arnold, and Yisong Yue. CARE: A benchmark suite for the classification and retrieval of enzymes. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

Yuxin Yang, Abby Jerger, Song Feng, Zixu Wang, Christina Brasfield, Margaret S. Cheung, Jeremy Zucker, and Qiang Guan. Improved enzyme functional annotation prediction using contrastive learning with structural inference. *Communications Biology*, 7(1):1690, 2024b.

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023.

Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2582–2593, 2022.

A Detailed Related Work

Beyond the representative sequence-based methods discussed in the main text, several studies have explored alternative architectures and enzyme-specific representation learning. DeepECtransformer (Kim et al., 2023) extends sequence-based EC prediction with a Transformer architecture, while EnzBert (Buton et al., 2023) and EnzPlacer (Ma et al., 2026) further investigate representations tailored to enzyme function prediction. A separate line of work incorporates structural or other biological information beyond protein sequence. Foldseek (van Kempen et al., 2024) enables efficient structure-based protein comparison, while CLEAN-Concat (Yang et al., 2024b), S-PLM (Wang et al., 2025a), GraphEC (Song et al., 2024), and TopEC (van der Weg et al., 2025) exploit structural information in different forms for enzyme function prediction. PoinnCARE (Xie et al., 2026) further integrates sequence, three-dimensional structure, active-site information, and the EC hierarchy. These approaches demonstrate the value of complementary biological information, but structural and active-site inputs may be unavailable, incomplete, or prediction-dependent. Our sequence-only formulation avoids these additional requirements and instead focuses on better organizing EC classes in the downstream representation space for fine-grained EC classification.

Hierarchy-aware enzyme classification has also been explored in several forms. ECPICK (Han et al., 2024) introduces hierarchical layers that combine global and level-specific prediction flows. GloEC (Huang et al., 2024) further represents the EC taxonomy as a directed graph and models dependencies among EC labels with a hierarchy-aware graph encoder, while HIT-EC (Dumontet et al., 2026) combines global prediction with level-specific modules aligned with the four-level EC hierarchy. These methods incorporate the EC hierarchy through hierarchical prediction or label-relation modeling, while the geometry among EC classes is still learned implicitly. Because prediction-level objectives constrain the final outputs rather than the relative positions of EC classes in the representation space, the learned geometry is not explicitly required to preserve hierarchical relationships. As a result, closely related sibling classes may still remain insufficiently separated. We instead use the hierarchy to directly construct the projection geometry for fine-grained discrimination.

B Experimental Details

B.1 Dataset Details

All experiments are conducted on the official splits of CARE Task 1 (Yang et al., 2024a). We use the 50% identity-reduced training set adopted in our main experiments, which contains 28,316 proteins after removing redundant training sequences. The corresponding test sets are kept unchanged from the benchmark. Table 5 reports the number of proteins and distinct EC annotations in each split.

The training label space contains 7, 73, 245, and 4,960 classes at EC Levels 1–4, respectively, illustrating the rapid increase in label granularity toward the finest level. The $< 30\%$ Identity, 30–50% Identity, and Price sets contain one target EC annotation per protein, whereas proteins in the Promiscuous set may be associated with multiple valid EC numbers.

B.2 Implementation and Hyperparameters

Feature Extraction. We use the pretrained ESM-2-650M model as the protein encoder and keep all encoder parameters frozen throughout training. For each protein, residue-level representations from the final encoder layer are mean-pooled over residue tokens, excluding special tokens, to obtain a 1280-dimensional sequence-level representation. Protein sequences longer than 1,022 residues are truncated before feature extraction. The extracted representations are cached before downstream training.

HG and Prototype Initialization. The pooled protein representations are mapped to a 512-dimensional space by a linear projection layer initialized with HG. The projected training features are then used to initialize the multi-prototype classifier by applying spherical K-means independently within each EC4 class. For class c, the number of prototypes is $K_c = \min(K, n_c)$, and all initialized prototypes remain learnable during downstream training. Detailed initialization settings are summarized in Table 6.

Table 5: Dataset statistics for the CARE Task 1 splits used in this work.

Split	**# Proteins**	**# Unique ECs**
Training (50% identity-reduced)	28,316	4,960
$< 30\%$ Identity	432	333
30–50% Identity	560	389
Price	148	56
Promiscuous	209	384

Table 6: Hyperparameter settings used for HiPro-EC.

Hyperparameter	Value
Projection dimension	512
HG within-class regularization	1×10^{-3}
Spherical K-means iterations	100
Spherical K-means tolerance	1×10^{-5}
Maximum prototypes K	9
Projection learning rate	1×10^{-5}
Prototype learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Batch size	256
Epochs	200

Optimization. We jointly optimize the trainable projection parameters and class prototypes using AdamW, with separate learning rates for the projection parameters and prototypes. Training is performed using the same optimization configuration across all four CARE evaluation splits. Detailed optimization hyperparameters are provided in Table 6.

C Baseline Details

This section provides additional details on the compared baselines, including brief descriptions of the methods and the implementation protocols used to obtain their results.

C.1 Baseline Methods

We compare HiPro-EC with representative sequence-based enzyme prediction methods and prototype- or centroid-based classification methods.

BLASTp (Altschul et al., 1990) identifies homologous proteins through local sequence alignment and transfers functional annotations based on sequence similarity. Under the CARE protocol, predictions are obtained from similarities between query proteins and proteins in the training set.

CLEAN (Yu et al., 2023) learns an enzyme function-oriented embedding space through contrastive learning, organizing proteins according to their functional relationships. EC prediction is performed by comparing query proteins with EC representations in the learned space.

ESM-2 (Lin et al., 2023) is an encoder-only Transformer protein language model pretrained with masked language modeling on large-scale protein sequence corpora. It provides general-purpose protein representations without explicit EC supervision.

ProtT5 (Elnaggar et al., 2022) is a T5-based protein language model pretrained on large-scale protein sequence corpora. Its contextual sequence representations can be transferred to downstream protein prediction tasks.

ESM-C (Candido et al., 2026) is a recent protein language model in the ESM family designed for scalable and efficient protein representation learning. It retains bidirectional masked sequence modeling while using an updated model architecture and training strategy.

EnzGFM (Wang et al., 2026) is an enzyme-specific protein language model based on a hybrid Mamba–Transformer architecture. Its hierarchical pretraining progressively adapts the model from general protein sequences to enzyme-specific sequence distributions. We use the 650M-parameter variant in our experiments.

CBC (Saralajew et al., 2025) is a Classification-by-Components framework that formulates prediction through the probabilistic compatibility between input representations and class-specific components. It belongs to the family of prototype-based networks and emphasizes interpretable component-based decision making.

DNC (Wang et al., 2023) revisits nearest-centroid classification in a deep representation space. It discovers sub-centroids from training samples to characterize the latent distribution of each class and performs non-parametric classification according to the distance between a query and these sub-centroids.

ClusPro (Qu et al., 2025) was originally proposed for compositional zero-shot learning. It performs non-parametric within-class clustering in the embedding space to discover local sub-clusters, and dynamically updates the corresponding prototypes using the samples assigned to each cluster. The resulting prototypes are further used in prototype-based contrastive learning and decorrelation objectives.

C.2 Baseline Implementation

All baselines are reproduced in our experimental environment using the official CARE training and test splits (Yang et al., 2024a). Whenever official implementations are available, we follow the original codebases and preserve the method-specific training and inference procedures, while adapting the input and output interfaces to the CARE enzyme classification task.

For BLASTp (Altschul et al., 1990), the CARE training proteins are used to construct the reference database. Each test protein is searched against this database. For a protein with $|Y|$ ground-truth EC annotations, we take the top-$|Y|$ retrieved training proteins and use their EC labels as the predicted label set.

For CLEAN (Yu et al., 2023), we follow the official implementation and train the function-oriented representation model using the CARE training split. Prediction follows the original CLEAN inference procedure based on the distance between query representations and EC representations in the learned embedding space.

For the protein language model baselines, including ESM-2 (Lin et al., 2023), ProtT5 (Elnaggar et al., 2022), ESM-C (Candido et al., 2026), and EnzGFM (Wang et al., 2026), the pretrained encoders are frozen. Residue-level representations are mean pooled to obtain protein-level embeddings, which are then fed into an MLP classifier trained on the CARE training split. The same downstream classification protocol is used across all PLM baselines.

For CBC (Saralajew et al., 2025), DNC (Wang et al., 2023), and ClusPro (Qu et al., 2025), we use frozen ESM-2-650M representations as the common protein input and adapt their original prototype- or centroid-based classification mechanisms to EC prediction. CBC retains its component-based decision mechanism, DNC constructs class-specific sub-centroids for nearest-centroid classification, and ClusPro dynamically identifies within-class sub-clusters and updates the corresponding prototypes during training. Using the same ESM representations provides a controlled comparison among different class-representation strategies.

D Additional Analyses

D.1 Run-to-Run Variability

To assess the sensitivity of HiPro-EC to random seeds, we repeat the main experiments over three independent runs. Table 7 reports the mean and standard deviation across these runs. The two low-identity splits and the Promiscuous split show consistently small variations, with standard deviations below 0.5 percentage points across all hierarchy levels.

Table 7: Run-to-run variability of HiPro-EC over three independent runs. Results are reported as mean ± standard deviation (%).

Test Set	L1	L2	L3	L4
$< 30\%$ Identity	89.58 ± 0.46	82.56 ± 0.35	77.70 ± 0.27	64.51 ± 0.13
30–50% Identity	95.95 ± 0.10	92.74 ± 0.27	88.75 ± 0.18	82.98 ± 0.27
Price	97.22 ± 0.00	89.58 ± 1.20	84.03 ± 1.20	45.25 ± 2.23
Promiscuous	92.52 ± 0.26	89.01 ± 0.12	86.62 ± 0.05	80.68 ± 0.28

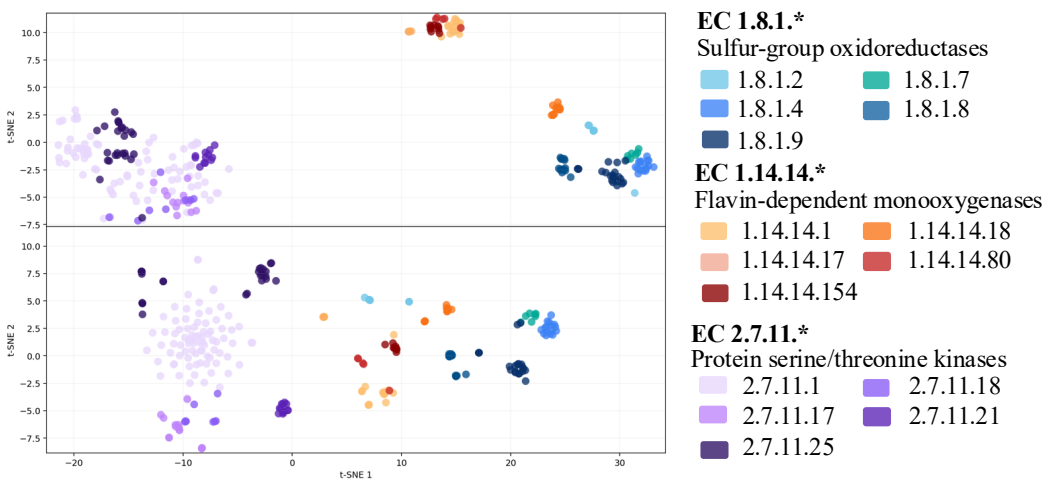

Figure 6: Additional t-SNE visualization of EC4 classes from three EC3 parent groups. Colors denote sibling EC4 classes within each parent group. The baseline is shown at the top and HiPro-EC at the bottom. While the baseline exhibits stronger coarse separation between parent groups, HiPro-EC yields clearer organization among fine-grained sibling EC4 classes.

A different pattern is observed on the Price split. While the Level-1 accuracy score is identical across all three runs (97.22 ± 0.00), the variation gradually becomes more pronounced at finer levels, reaching 45.25 ± 2.23 at Level 4. This suggests that the run-to-run variation on Price is primarily associated with fine-grained discrimination rather than coarse functional recognition. Despite this increased sensitivity at Level 4, the overall trends remain consistent across random seeds.

D.2 Additional Representation Visualizations

To complement the sibling-class visualization in the main text, Fig. 6 jointly visualizes EC4 classes from three different EC3 parent groups. In the baseline representation, the EC3 parent groups appear broadly separated, while several sibling EC4 classes within the same parent remain overlapping or poorly organized. HiPro-EC exhibits a different geometry: the separation between higher-level parent groups becomes less pronounced, whereas the fine-grained EC4 subclasses form clearer and more compact local structures.

This pattern is consistent with the goal of HiPro-EC, which is not to maximize already-sufficient coarse-level separation, but to emphasize the distinctions required for fine-grained EC prediction.

D.3 Intra-Class Heterogeneity

We quantify intra-class heterogeneity at the EC4 level using the dispersion of training proteins around a single class center. Importantly, the heterogeneity statistics are computed exclusively from the training set and do not use any test samples. We measure heterogeneity in the hierarchy-guided projection space before task-specific training, such that the analysis reflects the intrinsic class distribution under the HG initialization rather than a geometry induced by the learned multi-prototype classifier.

Table 8: Heterogeneity grouping based on the tertiles of the CARE training classes.

Group	Tertile	Cutoff
Low	Bottom third	≤ 0.0286
Medium	Middle third	$(0.0286, 0.0748]$
High	Top third	> 0.0748

Table 9: Effect of training support on multi-prototype modeling for the $< 30\%$ Identity split. Δ denotes the change in accuracy score from Single to MP, measured in percentage points.

Training support	Single	MP	Δ
1	38.27	35.80	-2.47
2–5	57.58	60.61	$+3.03$
6–9	56.60	69.81	$+13.21$
10–50	67.23	84.87	$+17.64$
> 50	73.47	95.92	$+22.45$

For an EC4 class c, let $\{\mathbf{z}_i\}_{i=1}^{N_c}$ denote the ℓ_2-normalized representations of its N_c training proteins in the HG projection space. We first define a normalized single-center representation for class c as

$$\mathbf{p}_c = \frac{\sum_{i \in c} \mathbf{z}_i}{\left\| \sum_{i \in c} \mathbf{z}_i \right\|_2}. \tag{15}$$

The intra-class heterogeneity score is then defined as the mean cosine distance from the training representations to this class center:

$$h_c = \frac{1}{N_c} \sum_{i \in c} \left(1 - \mathbf{z}_i^\top \mathbf{p}_c\right). \tag{16}$$

A smaller h_c indicates that proteins of the class are tightly concentrated around a common direction, whereas a larger h_c indicates a more dispersed class distribution. Thus, h_c measures the average dispersion around a single class center; it should not be interpreted as statistical variance or as the distance between multiple prototypes.

We compute intra-class heterogeneity only for EC4 classes with at least two training proteins ($N_c \geq 2$), since dispersion cannot be meaningfully estimated from a singleton class. The eligible classes are then divided into low-, medium-, and high-heterogeneity groups according to the tertiles of the heterogeneity distribution over the CARE training set. The resulting empirical cutoffs are summarized in Table 8. All four test splits use the same training-derived grouping, so no test samples are involved in determining the heterogeneity thresholds.

These cutoffs define relative heterogeneity groups within CARE and are not intended as universal thresholds for protein class heterogeneity.

D.4 Effect of Class Support

To examine whether the benefit of multi-prototype (MP) modeling is affected by the amount of training data available for each EC4 class, we further stratify samples in the $< 30\%$ Identity split according to class support, defined as the number of training proteins associated with the ground-truth EC4 class. The results are summarized in Table 9.

MP provides little benefit for extremely low-support classes and even degrades performance for singleton classes. In contrast, its advantage increases markedly with class support, reaching $+22.45$ percentage points for classes with more than 50 training proteins. This trend suggests that multiple prototypes are most effective when sufficient training examples are available to characterize distinct local patterns within a class. It also shows that class support is an important factor underlying classification difficulty and should therefore be considered alongside intra-class heterogeneity.

Table 10: Sensitivity to the maximum number of prototypes K. Level-4 accuracy scores are reported on the four CARE test sets. Bold and underlined values denote the best and second-best results, respectively.

K	$< 30\%$ **Identity**	30–50% **Identity**	**Price**	**Promiscuous**
1	63.2	<u>82.1</u>	35.1	78.8
3	<u>65.0</u>	81.6	38.5	79.1
5	64.5	<u>82.1</u>	38.5	79.4
7	**65.2**	**83.0**	**48.3**	79.7
9	64.6	**83.0**	<u>46.9</u>	<u>80.6</u>
11	64.6	**83.0**	**48.3**	**81.0**
13	64.8	**83.0**	43.4	79.9

D.5 Sensitivity to the Number of Prototypes

We further examine the sensitivity of HiPro-EC to the maximum number of prototypes K assigned to each EC4 class. We vary K while keeping all other training settings unchanged. For classes with fewer than K training samples, the actual number of prototypes remains $K_c = \min(K, n_c)$. Note that the $K = 1$ setting uses one learnable prototype per EC4 class, whereas the baseline used in the controlled analyses represents each class by its mean prototype computed from the training samples.

As shown in Table 10, performance is generally lower when only a small number of prototypes is available, with the single-prototype setting showing the clearest degradation, particularly on the challenging Price split. This suggests that making the class representative learnable alone does not fully account for the benefit of MP, and that multiple local prototypes can provide additional gains when a single representative is insufficient to capture the class distribution.

Performance improves substantially over the single-prototype setting and remains relatively stable for moderate prototype budgets ($K = 7$–11). Beyond this range, increasing K does not yield consistent further improvements, suggesting that a moderate prototype budget is sufficient. We therefore use $K = 9$ as the default setting, which lies within this stable range.