SoCoBio: Predicting Biomolecular Functions via Hierarchical Socialized Coevolution

Anonymous authors
Paper under double-blind review

Abstract

Biomolecular function prediction assigns sequences to functional taxonomies whose data-rich ancestors branch into a long tail of data-scarce leaf functions. Previous methods do not treat vertical generalization, which carries evidence from ancestors to tail classes, and horizontal discrimination, which distinguishes a class from its most similar siblings, as distinct interactions. To tackle this, we introduce SoCoBio, a hierarchical multi-label classifier for biomolecules that treats each class prototype as a learner in the sense of socialized learning. Whereas static hierarchical models weight relatives by a fixed rule, SoCoBio learns two interactions among prototypes, namely a Vertical Controller that passes gated information from each parent to its children and a Horizontal Controller that then lets each class select messages from a few similar siblings. A class thus receives shared context before it weighs messages from its closest siblings. Viewing SoCoBio as a within-predictor special case of socialized learning, we derive a sufficient condition for a learned update to lower the multi-label loss and a per-class direction bound whose required positive support grows only logarithmically with the number of classes. As a complete predictor on peptide and enzyme benchmarks and an independent enzyme test set, SoCoBio attains the highest leaf-level and mean-level F1 among the compared methods, and matched retraining with a short budget on the largest benchmark shows that its leaf-F1 gain from interaction, concentrated on tail classes, requires both controllers.

1 Introduction

Knowing the functions of biomolecules, notably proteins and peptides, guides applications such as enzyme discovery and antimicrobial peptide screening (Santos et al., 2025; Wan et al., 2024), where alignment-based methods (e.g., Altschul et al., 1990; Remmert et al., 2011) and deep learning models (e.g., Lin et al., 2023; Jumper et al., 2021) are developed. Function prediction follows existing taxonomy, such as the four-level Enzyme Commission (EC) classification (Yu et al., 2023) or a hierarchy of peptide activities as the restriction of the predictor (Figure 1a). Such hierarchical multi-label annotation faces two challenges:

On one hand, fine-grained classification at deep hierarchical levels demands *sibling discrimination*, since function systems such as EC and Gene Ontology (GO) form trees or directed acyclic graphs in which sibling functions share most attributes of their parent. For instance, the serine endopeptidases trypsin (EC 3.4.21.4) and thrombin (EC 3.4.21.5) differ in substrate specificity, which can hinge on a few active-site residues (Le Bonniec & Esmon, 1991). Hierarchical multi-label classification (HMC) methods (Wehrmann et al., 2018; Giunchiglia & Lukasiewicz, 2020; Zhu et al., 2023) incorporate the hierarchy so that predictions agree across levels, often reading the taxonomy as a fixed prior (Figure 1c) that constrains the predictions but does not decide which relatives a data-scarce leaf should learn from.

On the other hand, learning from few examples in a long-tailed distribution demands robust *peer generalization*, as most specific functions have only a handful of labeled sequences, both in two enzyme benchmarks (Figure 1b) and among peptide activities such as targeted antimicrobial activity. Long-tailed learning methods (Xiao et al., 2021; Dong et al., 2023) transfer knowledge from head to tail classes but overlook the hierarchy, so a tail class may borrow from unrelated functions and be biased toward features it shares with data-rich classes, against the specificity required above.

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.

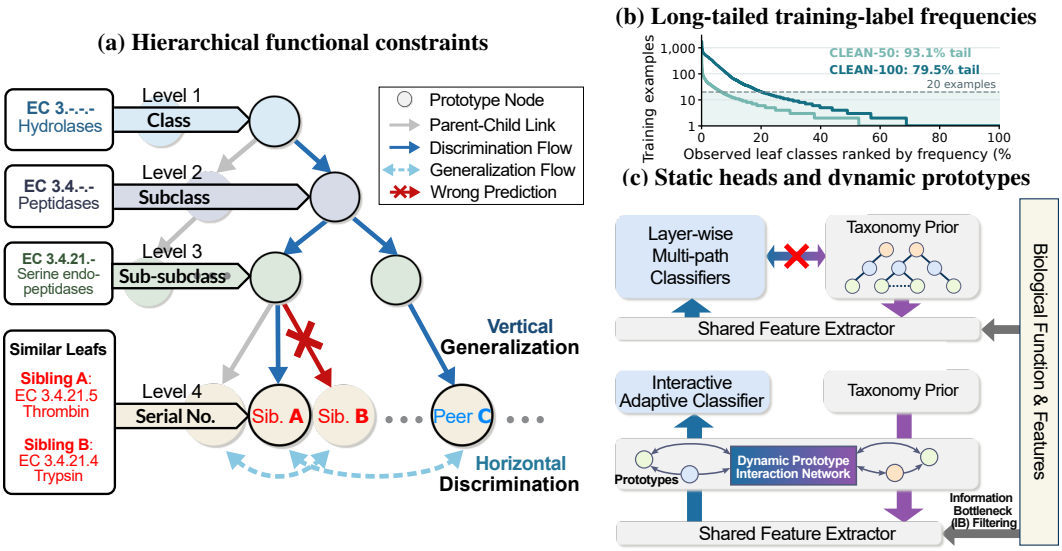

Figure 1: Hierarchical prediction motivates selective sharing. (a) Functional taxonomy and consistency constraints. (b) Long-tailed training-label frequencies on a log scale. (c) Static-head and dynamic-prototype architectures of prediction sharing.

Borrowing from the parent instead gives a tail class the broad activity it shares with its siblings but not the boundaries that separate them.

These two requirements expose a trade-off between generalization and discrimination (Zhou et al., 2020; Huang et al., 2024). However, existing hierarchy-aware message-passing approaches do not explicitly separate parent and sibling information according to their distinct roles in generalization and discrimination.

This choice is the question of *socialized learning*. In the society of learners, they select collaborators and regulate what they receive (Yao et al., 2024; Li et al., 2025), and whose theory ties the benefit of collaboration to how collaboration weights track the learners' losses (Yao et al., 2025). In this paper, we adopt this method to reformulate biomolecular function prediction as collaboration among class prototypes (Figure 1c).

Building on this view, we propose **SoCoBio**, **So**cialized **Co**evolution for **Bio**molecular function prediction, which realizes the two dimensions as a Vertical Controller and a Horizontal Controller acting on the prototypes of a hierarchical classifier (Figure 2). The Vertical Controller adaptively transfers ancestral information through gated parent–child calibration, whereas the Horizontal Controller learns sparse peer interactions through selective peer collaboration and gated aggregation.

An update of this kind need not help, because one prototype scores every sequence, and an update that helps the positives of a class can also raise its scores on negatives. In this paper, we find that moving a prototype toward its positive examples does not by itself lower the loss. The amount of such improvement, as provided in Theorem 3.1, 3.2 in Section 3.6, grows only logarithmically with the number of classes, which gives HMC a direction bound set by each class's own support. We see this as a valuable supplement to the theoretical foundations of socialized learning.

Our main contributions are threefold:

- We formulate hierarchical biomolecular function prediction as collaboration among class prototypes, a within-predictor special case of socialized learning in which the hierarchy decides whom each class may learn from rather than only constraining its outputs.
- We extend the analysis of socialized learning to learners that share one prediction head, deriving a sufficient condition for a learned update to lower the multi-label loss and a per-class direction bound set by the positive support of each class.
- Across extensive experiments of peptide and enzyme benchmarks and an independent enzyme test set, the proposed method (SoCoBio) attains the highest leaf- and mean-level F1 among the compared methods.

2 Related Work

Biomolecular function prediction. For peptides, most efforts address binary antimicrobial peptide (AMP) prediction (Li et al., 2022; Wan et al., 2024). Other works predict broader properties, such as activity against specific bacterial genera with Siamese networks (Losin & Veltri, 2021b) and signal peptides with parameter-efficient fine-tuning of ESM-2 (Zeng et al., 2024), or design peptide vaccines by evolutionary multi-objective optimization (Liu et al., 2024).

For enzymes, CLEAN (Yu et al., 2023) and CLIPZyme (Mikhael et al., 2024) use contrastive learning to align sequences with EC numbers or reaction schemes, and Prot2Text (Abdine et al., 2024) and ProtCLIP (Zhou et al., 2025) generate or align textual function descriptions. On GO, DeepGO-SE reasons with ontology axioms (Kulmanov et al., 2024) and PFresGO attends between proteins and GO terms (Pan et al., 2023), whereas DualGOFiller (Wang et al., 2025), MSRep (Luo & Luo, 2025), and GOBoost (Zhang et al., 2025b) address partial annotation and long-tailed labels. Hierarchy-aware enzyme predictors have also appeared, as EnzymeHunter trains a hierarchically aware contrastive objective (Cao et al., 2026) and HIT-EC encodes the EC levels one after another (Dumontet et al., 2026). In contrast, SoCoBio explicitly separates gated parent-to-child inheritance from learned sparse collaboration among sibling prototypes.

Hierarchical Multi-label Classification (HMC) and message passing. HMC predicts label sets that respect ancestor relations. HMCN combines local and global objectives, coherent HMC builds ancestor consistency into the output layer, HCAL aggregates child features into hierarchical prototypes, and SPUR unifies instance and hierarchy representations (Wehrmann et al., 2018; Giunchiglia & Lukasiewicz, 2020; Jiang et al., 2025; Yuan et al., 2024). Fine-grained classifiers such as FGVC and SCAD separate similar classes using granularity-aware representations and cross-level consistency (Wang et al., 2023b; Chang et al., 2021), metric-guided prototypes borrow structure from the hierarchy (Garnot & Landrieu, 2021), and rare HMC nodes have been addressed by imbalance weighting (Xu et al., 2026). GNN-based hierarchical methods pass information along graph edges, while graph attention can further assign adaptive weights to neighboring nodes (Velickovic et al., 2025). DeepGraphGO propagates protein features over interaction networks (You et al., 2021), GloEC passes input-dependent EC label states top-down and bottom-up with persistent trainable parameters (Huang et al., 2024), and HiAGM propagates label information in text classification (Zhou et al., 2020).

Socialized learning. Socialized learning studies whom a learner should learn from and how much to accept, with communication as its core mechanism, and its agents compensate for limited direct experience by collaborating with other agents (Yao et al., 2024). Socialized coevolution (SC, Yao et al. (2025)) lets models for different tasks collaborate and bounds the error of the collaborative system through the collaboration weights and the losses of the learners. Graphs Help Graphs (GHG, Li et al. (2025)) extends this view to graph data, letting graph learners exchange information and bounding the target risk by how compatible the exchanged sources are with the target distribution.

3 Methodology

3.1 Problem Formulation

Let $\mathcal{D} = \{(x_n, Y_n)\}_{n=1}^{N}$ be a dataset of biological sequences x_n with label sets Y_n. The labels are drawn from K function classes $\mathcal{V} = \{1, \ldots, K\}$ organized by a taxonomy, and every label set, whether used for training, evaluation, or calibration, is closed under ancestors, so a class in Y_n implies all of its ancestors. EC numbers and the peptide vocabulary form trees, so each non-root class i has one parent $p(i)$ and its siblings are the other children of $p(i)$ and peers are all $p(\cdot)$'s with identical tree depth, whereas a taxonomy with several parents per class, such as GO, would need a rule for aggregating parent messages. We treat each class as a socialized learner that maintains a class prototype $P_i \in \mathbb{R}^d$, and its parent and selectively chosen sibling classes (Section 3.4). The update of P_i is therefore driven not only by the sequences labeled with i but also by the messages of these collaborators. An encoder gives features $f(x) \in \mathbb{R}^d$, and we write $\bar{v} = v/\|v\|$. Each class is predicted separately and evaluated before any ancestor closure, so hierarchical consistency is measured rather than imposed (Section 4.2).

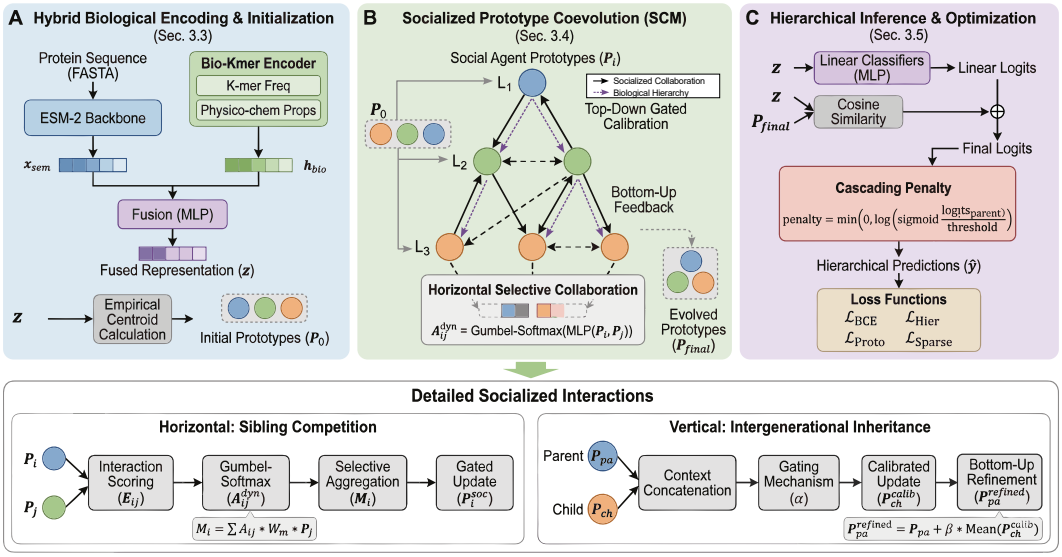

Figure 2: Diagram of SoCoBio, which proceeds from sequence encoding and prototype initialization (A), through parent and sibling collaboration (B), to hierarchical prediction (C).

3.2 Overall Architecture

The full design in Figure 2(a) comprises four components, namely Sequence Encoding and Biological Feature Extraction, which uses ESM-2 representations augmented with biological features, the Prototype Initialization Module, which builds an initial prototype for every class, the Socialized Collaboration Module, which updates the prototypes through the two controllers, and the Hierarchical Prediction Decoder, which scores sequences against the updated prototypes. The encoder pools the final-layer ESM-2 states (Lin et al., 2023) with learned attention weights rather than a plain token average, so that positions relevant to function can receive more weight.

3.3 Biological Encoding and Initialization Modules

We employ ESM-2 as the backbone to extract deep semantic representations from biological sequences. For an input sequence $S = (a_1, \ldots, a_L)$, let $H \in \mathbb{R}^{L \times d_{\text{esm}}}$ denote the final-layer token representations. Rather than naive token averaging, we use weighted attention pooling to emphasize functionally relevant motifs:

$$x_{\text{sem}} = \sum_{t=1}^{L} \alpha_t H_t, \quad \text{where} \quad \alpha_t = \mathsf{Softmax}_t\big(w^\top \tanh(W_a H_t + b_a)\big)\,. \tag{1}$$

To complement the contextual semantics with explicit biochemical information, we further extract two types of biological features. First, we compute normalized 3-mer frequencies over all $20^3 = 8000$ standard amino-acid combinations, yielding $v_{\text{kmer}} \in \mathbb{R}^{8000}$. Second, we construct a 4-D physicochemical vector v_{phy} containing net charge, average hydropathy, aromaticity, and normalized sequence length. The concatenated biological feature $v_{\text{raw}} = [v_{\text{kmer}}; v_{\text{phy}}] \in \mathbb{R}^{8004}$ is compressed by an MLP into $h_{\text{bio}} \in \mathbb{R}^{128}$.

Finally, we fuse the semantic and biological representations through NN with ELU activations as $z := \mathcal{F}_{\text{fusion}}([x_{\text{sem}} \oplus h_{\text{bio}}]) \in \mathbb{R}^{600}$. The fused representation z serves as the input to subsequent prototype initialization, socialized collaboration, and hierarchical prediction.

To align the initial prototypes with the learned feature manifold, we initialize each class prototype using its empirical feature centroid. Before training, we freeze the sequence encoder and feature-fusion network and perform a forward pass over the training set $\mathcal{D}_{\text{train}}$. For the k-th class at hierarchy level l, we aggregate the fused representations of all training samples annotated with that class and normalize their centroid:

$$S_{l,k} = \sum_{x_i \in \mathcal{D}_{\text{train}}} \mathbb{I}(y_{i,l,k} = 1)\, z_i, \quad N_{l,k} = \sum_{x_i \in \mathcal{D}_{\text{train}}} \mathbb{I}(y_{i,l,k} = 1), \quad P_{l,k}^{(0)} = \frac{S_{l,k}/N_{l,k}}{\|S_{l,k}/N_{l,k}\|_2}, \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function. This initialization places each prototype near the feature cluster of its class and provides a data-informed starting point for subsequent socialized prototype collaboration.

3.4 Socialized Collaboration Module

SoCoBio treats each class prototype as a socialized learner whose representation is jointly shaped by direct optimization and structured collaboration. Each prototype $P_i \in \mathbb{R}^d$ is a persistent learnable parameter updated by the global objective, while the Vertical and Horizontal Controllers introduce ancestral inheritance and selective peer interaction, respectively.

Vertical Controller. For a non-root class v_i with parent v_p, we form the parent–child context $C_{i,p} = [P_i \parallel P_p]$ and adaptively transfer ancestral information through a gated calibration:

$$g_i^V = \sigma\big(\mathrm{MLP}^V_{\mathrm{gate}}(C_{i,p})/\tau_V\big), \quad P_i^V = \mathrm{LN}\big(P_i + g_i^V \odot \mathrm{MLP}_{\mathrm{fusion}}(C_{i,p})\big). \tag{3}$$

The gate controls how strongly a child incorporates its parent context, providing shared ancestral information while retaining class-specific features.

Horizontal Controller. After vertical inheritance, each class selectively interacts with peers at the same hierarchical level. For a peer pair (v_i, v_j), we learn an interaction score and obtain a sparse binary relation through hard Gumbel–Softmax:

$$E_{ij} = \mathrm{MLP}_{\mathrm{rel}}(W_{\mathrm{src}} P_i^V + W_{\mathrm{tgt}} P_j^V), \qquad A_{ij}^{\mathrm{dyn}} = \mathsf{Gumbel\text{-}Softmax}_{\mathrm{hard}}(E_{ij}, \tau_H) \in \{0,1\}. \tag{4}$$

Selected peer messages are aggregated and adaptively incorporated into the prototype:

$$\begin{aligned} M_i &= \sum_{j:\, \ell(j)=\ell(i)} A_{ij}^{\mathrm{dyn}} W_{\mathrm{msg}} P_j^V, \quad g_i^H = \sigma\big(\mathrm{MLP}^H_{\mathrm{gate}}([P_i^V \parallel M_i])\big), \\ P_i^J &= \mathrm{LN}\big(P_i^V + g_i^H \odot M_i\big). \end{aligned} \tag{5}$$

3.5 Hierarchical Prediction and Training Objective

Given the fused sequence representation z and the final prototype P_i^J, SoCoBio combines a linear classifier with prototype similarity. For class i at hierarchy level $\ell(i)$,

$$\ell_i = \lambda_{\ell(i)}(w_i^\top z + b_i) + (1 - \lambda_{\ell(i)}) \frac{\bar{z}^\top \bar{P}_i^J}{\tau}, \qquad \hat{y}_i = \sigma(\ell_i), \tag{6}$$

where $\bar{v} = v/\|v\|$, $\tau > 0$ is the prototype temperature, and $\lambda_\ell = \sigma(\rho_\ell)$ is a learnable level-specific mixing weight. During training, we minimize

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{BCE}} + \lambda_1 \mathcal{L}_{\mathrm{Hier}} + \lambda_2 \mathcal{L}_{\mathrm{Proto}}, \text{ where } \mathcal{L}_{\mathrm{Hier}} = \sum_{i \in \mathcal{V}} \sum_{j \in \mathrm{Child}(i)} \max(0, \hat{y}_j - \hat{y}_i). \tag{7}$$

Here, $\mathcal{L}_{\mathrm{Hier}}$ and $\mathcal{L}_{\mathrm{Proto}}$ penalize parent-child violations and encourage samples to approach their corresponding prototypes while maintaining discriminative margins between related classes, separately.

3.6 When Does Prototype Collaboration Help?

SoCoBio is a within-predictor special case of the socialized-learning setting (Yao et al., 2024), with class prototypes as learners, the parent and candidate siblings as collaborators, and gates and attention as collaboration strengths. Because these learners share one encoder, prediction head, and loss, the bounds of SC and GHG (Yao et al., 2025; Li et al., 2025) do not cover them, so we extend the analysis and ask whether a learned update lowers the shared loss and whether a class has enough positives to judge the direction of its update. Proofs are provided in Appendix E.

We compare the joint pass with each comparator $A \in \{0, V, H\}$ on n calibration sequences drawn i.i.d. from the target distribution and independently of training and model selection, with class weights $\omega_i \geq 0$ that sum to one and are fixed before calibration. The passes share all parameters, features, and candidates, so they differ only in their unit prototypes. Write $d_i^A = u_i^J - u_i^A$, $z = \bar{f}$,

$q_{ki}^A = \ell_i^J(x_k) - \ell_i^A(x_k)$ for the score change, $r_{ki}^A = y_{ki} - \sigma(\ell_i^A(x_k))$ for the residual, and κ_{ki}^A for the largest value of $\sigma(t)(1-\sigma(t))$ between the two scores, and define

$$\widehat{\mathcal{A}}_A = \frac{1}{n}\sum_{k,i} \omega_i r_{ki}^A q_{ki}^A, \quad \widehat{\mathcal{C}}_A = \frac{1}{2n}\sum_{k,i} \omega_i (q_{ki}^A)^2 \kappa_{ki}^A, \quad B_A = \sum_i \omega_i c_i \|d_i^A\|. \tag{8}$$

The residual correction $\widehat{\mathcal{A}}_A$ is positive when the score changes move toward the labels, the curvature cost $\widehat{\mathcal{C}}_A$ charges for a finite step, and B_A bounds the paired loss difference of any single sequence. Let R_A be the population weighted BCE of pass A, assumed finite, and $\widehat{\Delta}_A$ the empirical comparator-minus-joint loss difference.

Theorem 3.1 (BCE gain of a learned joint update)**.** *For any labeled sample, $\widehat{\mathcal{A}}_A - \widehat{\mathcal{C}}_A \le \widehat{\Delta}_A \le \widehat{\mathcal{A}}_A$. Under independent calibration, with probability at least $1-\delta/2$, simultaneously for $A \in \{0, V, H\}$,*

$$R_A - R_J \ge \widehat{\mathcal{A}}_A - \widehat{\mathcal{C}}_A - B_A\sqrt{\frac{2\log(6/\delta)}{n}}. \tag{9}$$

On this event, a positive right-hand side implies a lower BCE risk than comparator A.

Theorem 3.2 (Update direction under positive support)**.** *Under independent calibration, with probability at least $1-\delta/2$, simultaneously over supported classes and $A \in \{0, V, H\}$, $L_i^A > 0$ implies that μ_i^{dir} exists and*

$$\|u_i^A - \mu_i^{\mathrm{dir}}\|^2 - \|u_i^J - \mu_i^{\mathrm{dir}}\|^2 \ge 2\|d_i^A\| L_i^A > 0. \tag{10}$$

For any observed margin $\bar{T}_i^A \ge \gamma > 0$, a support of $n_i \ge 8\log(6K/\delta)/\gamma^2$ suffices for a gain of at least $\gamma\|d_i^A\|$. Moreover,

$$\epsilon_{(r)} \le \sqrt{\frac{2r^\alpha \log(6K/\delta)}{C_\alpha}} = O\left(\sqrt{\frac{r^\alpha \log(K/\delta)}{C_\alpha}}\right), \tag{11}$$

so every supported leaf class of rank $r \le \min\{K_+, \lfloor (C_\alpha\gamma^2/[8\log(6K/\delta)])^{1/\alpha}\rfloor\}$ with observed margin at least γ gains at least $\gamma\|d_i^A\|$, and this support does not involve the total sample size n once the margin γ is observed.

Remarks. Coherent HMC proves that its outputs never violate the hierarchy (Giunchiglia & Lukasiewicz, 2020). Theorem 3.2 adds a post hoc statistical statement at every class with calibration positives and a nonzero update, with an allowance set by that class's own support and a cost logarithmic in K for all classes at once. Since $\bar{T}_i^A \le 1$, classes with at most $2\log(6K/\delta)$ positives, for example fewer than 27 for the $K = 5{,}574$ classes of CLEAN-100 at $\delta = 0.05$, cannot be certified. Moreover, the certification frontier widens with C_α and narrows only logarithmically with K. Thus, the tail classes with the fewest positives cannot be judged from their own examples, which is the gap that evidence from relatives is meant to fill.

4 Experiments

4.1 Experimental Setup

Datasets and preprocessing. We evaluate methods on two datasets: 1) **Therapeutic Peptide.** We construct the Peptide benchmark by aligning therapeutic peptide sequences from PepBenchmark (Zhang et al., 2025a) with the fine-grained functional taxonomy of Peptipedia v2.0 (Cabas-Mora et al., 2024). The resulting benchmark contains a three-level hierarchy with $3, 15$, and 54 functional categories. We follow the standardized PepBenchmark splitting procedure to construct the training, validation, and test sets. 2) **Enzyme Commission (EC).** Following CLEAN (Yu et al., 2023), we evaluate on CLEAN-50 and CLEAN-100 to study enzyme-function prediction at different data scales. Both datasets follow the four-level EC taxonomy. CLEAN-50 provides a relatively low-resource setting, whereas CLEAN-100 contains approximately 220k sequences for full-scale evaluation. We additionally evaluate the model trained on CLEAN-100 on the independent CLEAN test set, which combines three external enzyme datasets and is used only for out-of-distribution evaluation. Detailed split statistics, class counts, and label-frequency distributions are provided in Appendix B.

Method	Level	Peptide			CLEAN-50			CLEAN-100		
		Acc	HL	F1	Acc	HL	F1	Acc	HL	F1
F-MLP	L	61.42	34.20	28.96	0.00	0.944	0.00	73.07	0.143	61.90
	M	68.19	64.90	52.43	6.37	36.530	4.52	55.68	26.963	62.85
S-MLP	L	63.01	34.10	33.15	72.63	0.303	55.98	92.76	0.041	78.38
	M	70.02	64.00	57.72	84.80	4.019	74.79	96.57	0.658	91.39
FGVC	L	61.64	33.20	25.60	36.30	0.617	14.61	94.63	0.032	81.05
	M	69.41	64.40	54.77	72.51	6.955	61.72	97.18	0.717	91.20
HIFD2	L	60.73	33.80	23.35	20.36	1.949	26.77	32.26	2.400	13.61
	M	69.16	69.60	53.10	50.58	18.940	49.51	74.30	1.800	60.39
SCAD	L	60.00	39.50	29.32	56.52	0.442	36.41	91.04	0.050	69.31
	M	68.92	67.10	55.92	81.27	4.455	67.48	95.75	0.663	87.30
HMCN	L	70.55	30.40	42.41	23.96	0.722	9.83	78.07	0.112	44.44
	M	77.17	54.70	65.78	64.29	9.258	52.26	91.52	1.567	80.49
SPUR	L	67.12	34.70	39.02	62.58	0.387	47.01	85.50	0.080	60.59
	M	73.36	64.80	61.80	76.13	8.013	67.69	93.22	1.512	83.82
LLAMP	L	65.30	36.80	25.09	–	–	–	–	–	–
	M	71.46	68.80	54.90	–	–	–	–	–	–
AMPlify	L	62.33	36.10	35.93	–	–	–	–	–	–
	M	69.86	66.30	58.13	–	–	–	–	–	–
CLEAN	L	–	–	–	21.89	0.900	10.27	58.73	0.200	16.15
	M	–	–	–	63.99	5.900	57.04	86.52	1.100	69.50
ProtoHMC	L	51.83	37.70	20.09	54.99	0.476	28.55	87.61	0.071	56.29
	M	63.09	69.70	45.61	83.95	**3.071**	69.83	95.66	0.531	83.91
SoCoBio	L	**73.74**	**30.00**	**43.37**	**76.26**	**0.279**	**59.55**	**95.53**	**0.027**	**82.79**
	M	**79.30**	**51.20**	**66.23**	**88.49**	3.284	**78.70**	**98.07**	**0.465**	**93.56**

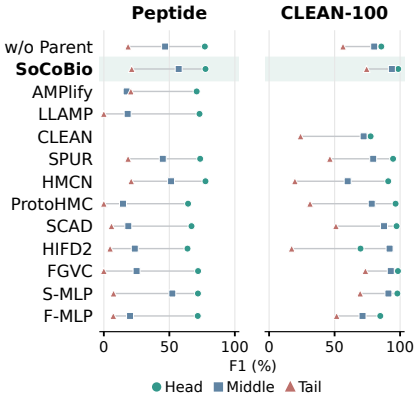

Figure 3: (Above) F1 of benchmark predictors grouped by training support; Peptide follows its predefined strata.

Table 1: (Left) Performance at the leaf (L) and mean (M) levels. Accuracy and F1 are percentages. HL is in 10^{-3} units.

Compared methods. The competitors of SoCoBio are in four categories. 1) **Simple ESM-2 baselines** include F-MLP (Flat-MLP) and S-MLP (Stratified-MLP). 2) **Hierarchical learning methods** include HMCN (Giunchiglia & Lukasiewicz (2020)) and SPUR (Yuan et al. (2024)). 3) **Fine-grained classifiers** include FGVC (Chang et al. (2021)), HIFD2 (Chen et al. (2022)), and SCAD (Wang et al. (2023a)). 4) **Biological sequence predictors** include LLAMP (Losin & Veltri (2021a)), AMPlify (Li et al. (2022)), and CLEAN (Yu et al. (2023)). LLAMP and AMPlify are evaluated only on the Peptide benchmark, whereas CLEAN is evaluated only on the enzyme benchmarks due to their domain-specific designs. We additionally report ProtoHMC, our prototype-only reference model, separately from the published baselines. Implementations of these methods are provided in Appendix B.2.

Evaluation metrics. We report Accuracy, macro-F1, and Hamming Loss (HL) using a probability threshold of 0.5. Metrics are reported at the **leaf level** and as the **mean across hierarchy levels**. Because HL is small in large label spaces, the main tables report it in units of 10^{-3}. We additionally report strict-path consistency, defined as the fraction of predictions containing exactly one class at each level and forming a valid root-to-leaf path. Complete definitions are provided in Appendix B.2.

4.2 Comparison and Ablation Studies

Benchmark comparison. Table 1 reports single-run results, consistent with the prediction records used in the subsequent sequence-level and resampling analyses; complete level-wise results are provided in Appendix C. SoCoBio achieves the highest leaf-level and mean-level F1 on all three benchmarks. On CLEAN-100, SoCoBio exceeds FGVC by 1.74 leaf-F1 points, while on CLEAN-50 it exceeds S-MLP by 3.57 points. Paired resampling of $2{,}000$ sequence-cluster replicates yields 95% percentile ranges of $[1.02, 2.15]$ and $[1.83, 3.79]$, respectively. At the sequence level, SoCoBio alone predicts all four CLEAN-100 hierarchy levels correctly for 538 sequences, compared with 175 sequences uniquely corrected by FGVC.

Component and controller ablations. Figure 4 evaluates the major components of SoCoBio, including the contrastive objective, biological features, consistency constraint, prototype modeling, and the two socialized controllers. Within the controller ablations, we denote disabled interaction, Vertical-only interaction, Horizontal-only interaction, and the complete joint interaction as 0, V, H, and J, respectively. Thus, V removes the Horizontal Controller, H removes the Vertical Controller, while J is the complete SoCoBio model.

Removing either controller reduces predictive performance on the primary benchmarks, while disabling both interactions causes a substantially larger degradation, particularly on the two CLEAN benchmarks. The joint configuration therefore benefits from complementary vertical inheritance and horizontal peer collaboration rather than either interaction alone. The remaining component ablations further show that the gains are not attributable to a single auxiliary module.

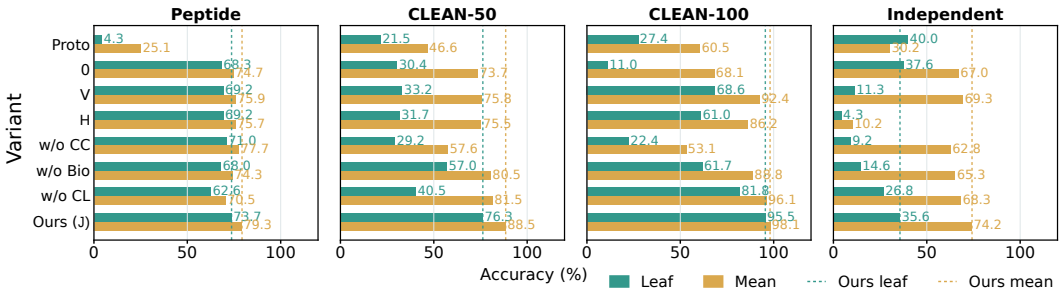

Figure 4: Component and controller ablations of SoCoBio on leaf- and mean-level accuracy.

Table 2: Single-run strict-path consistency (%).

Method	Peptide	CL-50	CL-100	Gen.	Method	Peptide	CL-50	CL-100	Gen.
F-MLP	74.20	0.00	28.09	2.60	w/o CL	88.81	43.54	84.45	19.38
S-MLP	76.94	68.91	88.00	39.97	w/o Bio	89.50	56.28	65.22	24.05
FGVC	71.92	36.06	89.95	41.18	w/o CC	73.52	24.15	21.66	6.23
HIFD2	68.26	13.07	11.37	3.29	w/o Parent	92.01	33.63	56.89	1.73
SCAD	73.52	57.79	86.82	27.85	w/o Peer	84.25	36.02	63.30	18.51
HMCN	88.58	23.99	76.17	18.17	w/o SC	84.25	34.10	70.51	7.44
SPUR	87.90	57.68	82.10	30.45	Proto-legacy	4.57	26.52	22.36	1.77
LLAMP	78.77	–	–	–					
AMPlify	80.82	–	–	–	ProtoHMC	62.56	59.02	85.76	23.70
CLEAN	–	22.21	35.80	33.39	**SoCoBio**	**92.69**	**76.33**	**91.63**	**46.19**

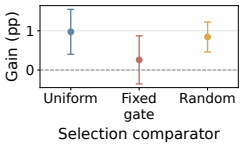

Figure 5: Peer-selection ablations on CLEAN-100; error bars show run-to-run standard deviation.

Selective peer interaction and hierarchy validity. To isolate how the Horizontal Controller selects same-level peers, we match initialization and training budget and compare the learned hard Gumbel selection with uniform peer aggregation and random peer selection, together with a fixed-gate variant. The results are reported in Figure 5 and Table 2. Uniform and random peer selection reduce CLEAN-100 leaf F1 by 0.97 and 0.84 points, respectively, whereas replacing the learned gate with a fixed gate changes performance by only 0.26 points. The learned peer-selection mechanism therefore contributes more strongly than the gate parametrization in this controlled comparison (Figure 5).

Table 2 reports strict-path consistency for the same representative runs used in the corresponding benchmark analyses. Before postprocessing, SoCoBio reaches 91.63% on CLEAN-100 compared with 89.95% for FGVC, and 46.19% on the independent set compared with 41.18% for FGVC. Ancestor closure leaves leaf F1 unchanged, indicating that the predictive gain cannot be explained by post-hoc hierarchy completion alone.

4.3 Evaluation under Different Settings

Long-tail prediction. Aggregate performance can obscure errors on data-scarce classes. In Figure 1, we group leaf classes by their training support. On CLEAN-100, head, middle, and tail classes contain at least 100, 20–99, and 1–19 training examples, respectively. SoCoBio achieves a tail F1 of 74.49, compared with 73.52 for FGVC and 69.52 for S-MLP, with moderate improvement on other groups.

Independent-test generalization. We further evaluate the model trained on CLEAN-100 on the independent CLEAN test set (Table 3). SoCoBio improves leaf F1 over FGVC by 1.39 points and mean-level F1 over S-MLP by 3.98 points. Particularly, across the constituent external subsets, SoCoBio obtains leaf F1 scores of 43.36 on New-392 and 20.00 on Price-149, compared with 41.02 and 15.99 for FGVC, respectively. FGVC performs better on the 37-sequence halogenase subset (28.32 versus 12.01), indicating that the improvement is not uniform across all external subsets.

4.4 Empirical Analysis of the Theoretical Results

Measured gain and the theoretical bound. Figure 6(a,b) shows the BCE gain of the joint pass over each alternative. At the final checkpoint against disabled interaction, a residual correction of 11.340×10^{-6} and a curvature cost of 3.546×10^{-6} give the lower bound 7.794×10^{-6} of Theorem 3.1, and eight segments bracket the measured 8.922×10^{-6} within $[8.792, 9.049] \times 10^{-6}$ (Figure 7a). At the intermediate checkpoint, the joint update instead raises test loss against all three alternatives while lowering direction error, the mean squared distance from a unit prototype to its positives' mean direction (Figure 6(a,c)), and against disabled interaction its correction of 2.775×10^{-6} is below the realized second-order term of 3.242×10^{-6}. On the validation split,

Table 3: Independent-test generalization on CLEAN. Accuracy and F1 are percentages; HL is in 10^{-3} units.

Method	L1			L2			L3			L4			Mean		
	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1
F-MLP	17.47	163.87	23.56	11.94	27.99	12.05	18.86	10.72	12.10	12.46	3.94	14.56	15.18	51.63	15.57
S-MLP	92.73	14.34	94.72	82.18	6.15	70.63	74.57	4.39	68.05	30.80	3.91	30.50	70.07	7.19	65.98
FGVC	93.94	11.86	93.08	83.74	5.69	65.98	74.05	**4.11**	64.26	33.39	3.62	34.47	71.28	6.32	64.45
HIFD2	74.05	47.90	70.57	50.17	17.60	41.63	28.89	10.30	27.47	1.38	4.90	1.66	38.62	20.20	35.33
SCAD	94.46	13.84	93.69	79.93	7.56	63.19	66.61	6.17	58.19	24.05	3.85	21.91	66.26	7.85	59.25
HMCN	86.85	28.67	88.71	76.82	8.89	63.30	63.67	5.82	52.73	11.42	4.34	10.38	59.69	11.93	53.78
SPUR	88.24	23.48	91.81	75.26	8.88	62.61	64.71	6.27	58.28	22.49	3.72	20.36	62.67	10.59	58.26
CLEAN	92.56	17.30	92.88	80.28	7.20	66.12	65.57	6.00	59.03	18.34	3.90	19.98	64.19	8.60	59.50
ProtoHMC	94.81	12.61	93.44	87.54	4.90	73.89	73.53	4.99	63.80	16.44	4.10	16.07	68.08	6.65	61.80
Ours	**95.16**	**10.13**	**96.92**	**89.45**	**4.23**	**74.42**	**76.64**	4.45	**72.65**	**35.64**	**3.35**	**35.86**	**74.22**	**5.54**	**69.96**

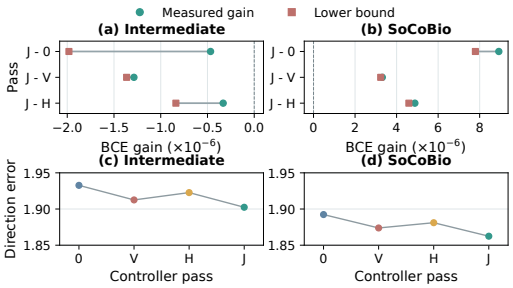

Figure 6: (a,b) Empirical BCE gains and lower bounds for joint interaction. (c,d) Prototype-direction errors.

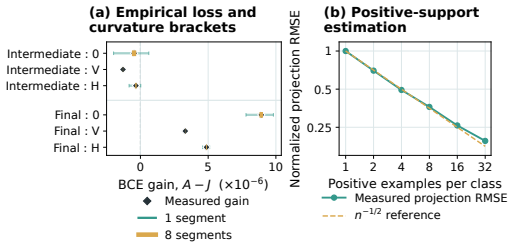

Figure 7: (a) Comparator-minus-joint BCE with 1&8 segment brackets. (b) Projection-estimation RMSE as positive support increases with $n^{-1/2}$ reference trend.

the same update lowers BCE against the disabled and peer-only passes, so the sign of these small differences depends on the sample. We find that moving prototypes toward their positives does not by itself lower the multi-label loss, and a rise requires the realized second-order term to exceed the correction.

Positive support. The allowance ϵ_i of Theorem 3.2 shrinks as $n_i^{-1/2}$ by construction, so we check whether the actual error follows this rate. On a fixed set of 135 classes at the final checkpoint, the normalized RMSE of the projection estimate falls to 0.195 at 32 positives, close to the $n^{-1/2}$ reference of 0.177 (Figure 7b). The worst-case allowance is much wider than this error, which is why few classes reach a positive direction endpoint.

Relations across hierarchy branches. Although messages pass only from parents to children and among peers, joint training moves the stored prototypes of leaf classes under different L3 parents slightly apart in all five matched CLEAN-100 seeds, whereas the updated prototypes move slightly closer in four. Among pairs whose classes both have test positives, however, we find that only 22 of the 100 pairs that approach each other most exceed their initial similarity, and none of these also improves F1 at an endpoint in at least three of five seeds, so linking a relation to biology and prediction needs independent biological evidence and targeted interventions. Appendix D.7 reports the full groups and the verified EC mapping.

5 Conclusion

Tail biomolecular functions could borrow evidence from related classes, and SoCoBio gives each class prototype a bounded message from its parent and lets it choose which siblings to hear, following the question that socialized learning asks of collaborating learners. As a complete predictor, SoCoBio gives the highest leaf F1 and mean-level F1 among the compared predictors on peptide and enzyme benchmarks and an independent enzyme test set. Matched retraining with a short budget on the largest benchmark shows that the leaf-F1 gain from interaction, concentrated on tail classes, needs both the Vertical and the Horizontal Controller and learned sibling selection. On CLEAN-100 tail classes, the complete predictor makes fewer multi-leaf predictions and fewer errors outside the true parent than FGVC while confusing siblings about as often. Our analysis gives a sufficient condition for a learned update to lower the multi-label loss and the positive support that suffices to judge its direction. Population-level certificates with more power on tail classes, homology-separated holdouts, hierarchies with several parents per class, and biologically verified prototype relations remain open.

AI USE STATEMENT

Generative AI tools were used to assist with language revision and manuscript polishing, and to support the development and checking of mathematical arguments and proofs. In particular, AI assistance was used to improve the presentation of the paper and to help formulate, refine, and verify parts of the theoretical analysis. All AI-assisted mathematical arguments and manuscript revisions were reviewed and verified by the authors, who take responsibility for the final content.

REPRODUCIBILITY STATEMENT

The appendices provide implementation details, dataset splits, evaluation metrics, controller configurations, fixed-checkpoint procedures, and the assumptions and proofs underlying the theoretical analysis. We also retain the prediction records, measurement outputs, figure-generation scripts, and numerical checks used to produce the reported results. Code and supporting artifacts will be released upon publication.

REFERENCES

Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. Prot2text: Multimodal protein's function generation with gnns and transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 10757–10765, 2024.

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.

Gabriel Cabas-Mora, Anamaría Daza, Nicole Soto-García, Valentina Garrido, Diego Alvarez, Marcelo A. Navarrete, Lindybeth Sarmiento-Varón, Julieta Haydee Sepúlveda Yáñez, Mehdi D. Davari, Frédéric Cadet, Álvaro Olivera-Nappa, Roberto Uribe-Paredes, and David Medina-Ortiz. Peptipedia v2.0: a peptide sequence database and user-friendly web platform. a major update. *Database: The Journal of Biological Databases and Curation*, 2024.

Guoxin Cao, Jian Ouyang, Xiangyi Xiong, Changle Liu, Yi Zhang, Siqi Yang, Tieliu Shi, and Jun Wu. EnzymeHunter: Achieving fine-grained enzyme function prediction with a hierarchically aware contrastive learning framework. *Patterns*, 7(7):101567, 2026. doi: 10.1016/j.patter.2026.101567.

Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your "flamingo" is my "bird": Fine-grained, or not. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11471–11480, 2021.

Jingzhou Chen, Peng Wang, Jian Liu, and Yuntao Qian. Label relation graphs enhanced hierarchical residual network for hierarchical multi-granularity classification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4848–4857, 2022.

Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. Lpt: Long-tailed prompt tuning for image classification. *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

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. doi: 10.1038/s41467-026-68727-3.

Vivien Sainte Garnot and Loic Landrieu. Leveraging class hierarchies with metric-guided prototype learning. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2021.

Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9662–9673. Curran Associates, Inc., 2020.

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. doi: 10.1093/bib/bbae365.

Ruobing Jiang, Mengzhe Liu, Haobing Liu, and Yanwei Yu. Hierarchy-consistent learning and adaptive loss balancing for hierarchical multi-label classification. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1190–1199. ACM, 2025.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Maxat Kulmanov, Francisco J. Guzmán-Vega, Paula Duek Roggli, Lydie Lane, Stefan T. Arold, and Robert Hoehndorf. Protein function prediction as approximate semantic entailment. *Nature Machine Intelligence*, 6:220–228, 2024. doi: 10.1038/s42256-024-00795-w.

B F Le Bonniec and C T Esmon. Glu-192—-gln substitution in thrombin mimics the catalytic switch induced by thrombomodulin. *PNAS*, 88(16):7371–7375, 1991.

Chenkai Li, Darcy Sutherland, S Austin Hammond, Chen Yang, Figali Taho, Lauren Bergman, Simon Houston, René L Warren, Titus Wong, Linda MN Hoang, et al. Amplify: attentive deep learning model for discovery of novel antimicrobial peptides effective against who priority pathogens. *BMC genomics*, 2022.

Jialu Li, Yu Wang, Pengfei Zhu, Wanyu Lin, Xinjie Yao, and Qinghua Hu. Graphs help graphs: Multi-agent graph socialized learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 59325–59352. Curran Associates, Inc., 2025.

Zeming Lin, Halil Akin, Roshan Rao, Brian L. 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*, pp. 1123–1130, 2023.

Dan-Xuan Liu, Yi-Heng Xu, and Chao Qian. Peptide vaccine design by evolutionary multi-objective optimization. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, pp. 6967–6975, 2024.

Lauren Losin and Daniel Veltri. Exploring target specificity of antimicrobial peptides through deep learning embeddings. In *Proceedings of the 12th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB)*, 2021a.

Lauren Losin and Daniel Veltri. Exploring target specificity of antimicrobial peptides through deep learning embeddings. *Proceedings of the 12th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB)*, 2021b.

Jiaqi Luo and Yunan Luo. Learning maximally spanning representations improves protein function annotation. In *Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pp. 420–423, 2025.

Peter G. Mikhael, Itamar Chinn, and Regina Barzilay. Clipzyme: Reaction-conditioned virtual screening of enzymes. In *Proceedings of the 41th International Conference on Machine Learning (ICML)*, 2024.

Tong Pan, Chen Li, Yue Bi, Zhikang Wang, Robin B. Gasser, Anthony W. Purcell, Tatsuya Akutsu, Geoffrey I. Webb, Seiya Imoto, and Jiangning Song. Pfresgo: an attention mechanism-based deep-learning approach for protein annotation by integrating gene ontology inter-relationships. *Bioinformatics*, 39(3):btad094, 2023.

Michael Remmert, Andreas Biedermann, Thomas Beck, Andrei N Lupas, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2011.

Clelton A. Santos, Mariana A. B. Morais, Fernanda Mandelli, Evandro A. Lima, Renan Y. Miyamoto, Paula M. R. Higasi, Evandro A. Araujo, Douglas A. A. Paixao, Joaquim M. Junior, Maria L. Motta, Rodrigo S. A. Streit, Luana G. Morao, Claudio B. C. Silva, Lucia D. Wolf, Cesar R. F. Terrasan, Nathalia R. Bulka, Jose A. Diogo, Felipe J. Fuzita, Felippe M. Colombari, Camila R. Santos, Priscila T. Rodrigues, Daiane B. Silva, Sacha Grisel, Juliana S. Bernardes, Nicolas Terrapon, Vincent Lombard, Antonio J. C. Filho, Bernard Henrissat, Bastien Bissaro, Jean-Guy Berrin, Gabriela F. Persinoti, and Mario T. Murakami. A metagenomic 'dark matter' enzyme catalyses oxidative cellulose conversion. *Nature*, 639:1076–1083, 2025.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2025.

Fangping Wan, Felix Wong, James J. Collins, and Cesar de la Fuente-Nunez. Machine learning for antimicrobial peptide identification and design. *Nature Reviews Bioengineering*, 2(5):392–407, 2024.

Rui Wang, Cong Zou, Weizhong Zhang, Zixuan Zhu, and Lihua Jing. Consistency-aware feature learning for hierarchical fine-grained visual classification. In *Proceedings of the 31st ACM International Conference on Multimedia (ACMMM)*, pp. 2326–2334, 2023a.

Rui Wang, Cong Zou, Weizhong Zhang, Zixuan Zhu, and Lihua Jing. Consistency-aware feature learning for hierarchical fine-grained visual classification. In *ACM International Conference on Multimedia (ACMMM)*, 2023b.

Shaojun Wang, Hancheng Liu, Weiqi Zhai, and Shanfeng Zhu. Dualgofiller: A dual-channel graph neural network with contrastive learning for enhancing function prediction in partially annotated proteins. In *Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pp. 49–67, 2025.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo C. Barros. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 5075–5084, 2018.

Lin Xiao, Xiangliang Zhang, Liping Jing, Chi Huang, and Mingyang Song. Does head label help for long-tailed multi-label text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 14103–14111, 2021.

J. Xu, J. Gillis, A. Sharma, B. Misiuk, N. Brown, and T. Trappenberg. Improving detection of rare nodes in hierarchical multi-label learning, 2026. arXiv preprint.

Xinjie Yao, Yu Wang, Pengfei Zhu, Wanyu Lin, Jialu Li, Weihao Li, and Qinghua Hu. Socialized learning: Making each other better through multi-agent collaboration. In *Proceedings of the 41th International Conference on Machine Learning (ICML)*, 2024.

Xinjie Yao, Yu Wang, Pengfei Zhu, Wanyu Lin, Ruipu Zhao, Zhoupeng Guo, Weihao Li, and Qinghua Hu. Socialized coevolution: Advancing a better world through cross-task collaboration. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 71780–71797. PMLR, 13–19 Jul 2025.

Ronghui You, Shuwei Yao, Hiroshi Mamitsuka, and Shanfeng Zhu. Deepgraphgo: graph neural network for large-scale, multispecies protein function prediction. *Bioinformatics*, 37(Supplement_1): i262–i271, 2021.

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.

Zixuan Yuan, Hao Liu, Haoyi Zhou, Denghui Zhang, Xiao Zhang, Hao Wang, and Hui Xiong. Self-paced unified representation learning for hierarchical multi-label classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

Shuai Zeng, Duolin Wang, Lei Jiang, and Dong Xu. Prompt-based learning on large protein language models improves signal peptide prediction. In *Annual International Conference on Research in Computational Molecular Biology*, 2024.

Jiahui Zhang, Rouyi Wang, Kuangqi Zhou, Tianshu Xiao, Lingyan Zhu, Yaosen Min, and Yang Wang. Pepbenchmark: A standardized benchmark for peptide machine learning. In *International Conference on Learning Representations (ICLR)*, 2025a.

Lei Zhang, Yang Wang, Xiao Chen, Jie Hou, Dong Si, Rui Ding, Bo Jiang, Hailey Ledenko, and Renzhi Cao. GOBoost: leveraging long-tail gene ontology terms for accurate protein function prediction. *Bioinformatics*, 41, 2025b.

Hanjing Zhou, Mingze Yin, Wei Wu, Mingyang Li, Kun Fu, Jintai Chen, Jian Wu, and Zheng Wang. Protclip: Function-informed protein multi-modal learning. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 22937–22945, 2025.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1106–1117. Association for Computational Linguistics, 2020.

He Zhu, Chong Zhang, Junjie Huang, Junran Wu, and Ke Xu. Hitin: Hierarchy-aware tree isomorphism network for hierarchical text classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Appendices A and B specify the implementation and the experimental setup, Appendices C to D support Sections 4.2 to 4.4 in turn, Appendix E proves the results of Section 3.6, and Appendix F compares them with prior theory.

A Implementation Details

This appendix summarizes the implementation of SoCoBio used throughout Section 4, including the notation for fixed controller passes, training and fixed-checkpoint evaluation, and computational cost.

A.1 Notation

Table 4 summarizes the notation shared by the fixed-controller analyses and the theoretical results. The four passes $A \in \{0, V, H, J\}$ evaluate one checkpoint with both controllers disabled, only the Vertical Controller enabled, only the Horizontal Controller enabled, or both enabled, respectively.

Table 4: Notation for SoCoBio and its four fixed controller passes.

Symbol	Meaning
$\mathcal{G} = (\mathcal{V}, \mathcal{E}), K$	Label hierarchy and its K function nodes.
P_i, u_i^A	Prototype of class i and its unit direction in pass A.
$z(x), \bar{z}(x)$	Fused sequence representation and its normalized direction.
d_i^A	Actual displacement $u_i^J - u_i^A$.
$\ell_i^A, r_{ki}^A, q_{ki}^A$	Class logit, label minus comparator probability, and joint-minus-comparator logit change.
$\widehat{\mathcal{A}}_A, \widehat{\mathcal{C}}_A, B_A$	Residual correction, curvature cost, and absolute paired-contrast bound in Theorem 3.1.
$\bar{T}_i^A, \epsilon_i, L_i^A$	Mean feature projection along the update, sampling allowance, and lower endpoint in Theorem 3.2.
n, n_i	Number of calibration proteins and positive support of class i.
C_α, K_+	Observed rank–support envelope and number of leaves with positive calibration support.
R_A	Population weighted BCE of pass A.
$a_i(X), c_i$	Shared readout contribution and fixed prototype-branch scale.
ω_i, κ_{ki}^A	Fixed class weight and maximum logistic curvature on the logit-update interval.
$m_i, \mu_i^{\mathrm{dir}}$	Population positive-class feature mean and its direction when nonzero.
δ	Failure-probability budget fixed before calibration.

In Appendix E.6, m_i denotes the distance of a logit-update interval from zero rather than the positive-class feature mean, and δ_i denotes the corresponding logit increment. All other symbols follow Table 4.

A.2 Training and Fixed-Checkpoint Algorithms

Algorithm 1 summarizes SoCoBio training and the fixed-controller evaluation used in Sections 4.2 and 4.4. Training uses the joint interaction J, whereas fixed-checkpoint evaluation freezes all learned parameters and evaluates the same checkpoint under the four controller configurations 0, V, H, and J.

A.3 Computational Cost

We measure forward latency and peak allocated GPU memory on an NVIDIA A100-SXM4-80GB in FP32 with TF32 disabled. Controller-head timings use cached sequence representations, while the end-to-end measurements additionally include sequence encoding, tokenization, data transfer, thresholding, and CPU outputs. Table 5 reports median latency over repeated measurements.

With cached representations at batch size 256, joint interaction increases head latency from 1.209 to 5.169 ms. When sequence encoding is included, its cost dominates: at padded length 1,024, encoder-only and encoder-plus-head latency are 312.498 and 317.472 ms, respectively.

Algorithm 1: SoCoBio training and fixed-controller evaluation

Require: Fused representations $z(x)$, prototypes P, hierarchy, model parameters, mode
if training **then**
  $P^V \leftarrow V(P)$
  $P^J \leftarrow H(P^V)$
  Compute logits using Eq. (6) with P^J
  Compute $\mathcal{L}_{\text{total}}$ using Eq. (7)
  Update all trainable parameters by backpropagation
else
  Freeze the checkpoint
  $P^0 \leftarrow P, \quad P^V \leftarrow V(P), \quad P^H \leftarrow H(P), \quad P^J \leftarrow H(P^V)$
  for $A \in \{0, V, H, J\}$ **do**
    Compute logits using Eq. (6) with P^A
  end for
  Return the four fixed-pass predictions
end if

Table 5: Forward latency and peak GPU memory of SoCoBio components.

Component	Batch	Padded tokens	Latency (ms)	Memory (MiB)
Head 0	32	–	1.046	636.6
Head V	32	–	5.496	1320.3
Head H	32	–	4.805	1303.3
Head J	32	–	4.846	1320.3
Head 0	256	–	1.209	648.4
Head V	256	–	5.130	1321.5
Head H	256	–	5.125	1304.6
Head J	256	–	5.169	1321.5
Head 0	512	–	1.518	674.9
Head V	512	–	5.417	1323.2
Head H	512	–	5.403	1306.2
Head J	512	–	5.440	1323.2
Peer selection	0	–	53.305	659.1
Encoder	8	128	29.516	672.4
Encoder + head	8	128	33.860	1324.5
Warm sequence path	8	128	45.592	1324.5
Encoder	8	512	126.560	1034.6
Encoder + head	8	512	131.100	1334.9
Warm sequence path	8	512	143.393	1334.9
Encoder	8	1024	312.498	2088.5
Encoder + head	8	1024	317.472	2088.5
Warm sequence path	8	1024	331.069	2088.6

B Datasets and Evaluation Protocols

This appendix supports the experimental setup of Section 4. It reports the split sizes and label frequencies and defines the metrics.

B.1 Data Splits and Label Frequencies

Table 6 gives the split sizes and training-label statistics of Peptide, CLEAN-50, and CLEAN-100. The Peptide hierarchy has three levels, and both CLEAN benchmarks share the four-level EC hierarchy with 7, 72, 253, and 5,242 classes per level, so $K = 5{,}574$. Training covers 33 of the 54 Peptide leaf classes, 4,209 of the 5,242 EC leaf classes on CLEAN-50, and all 5,242 on CLEAN-100. The CLEAN independent test set of 578 enzymes is used only for evaluation.

Figure 8 plots the training support of every observed leaf class in decreasing order. Many leaf classes have fewer than 20 training sequences, so the frequency analyses of Figure 1 group classes by training

Table 6: Dataset split sizes, observed leaf classes, and training-support statistics.

Dataset	Total	Train	Val	Test	# Classes	Avg. Supp.	Tail Ratio (%)
Peptide	4,403	3,525	440	438	33	106.8	66.7%
CLEAN-50	29,430	23,895	2,780	2,755	4,209	6.1	93.1%
CLEAN-100	222,220	182,196	22,578	17,446	5,242	36.9	79.5%

support. The direction analysis of Appendix D instead counts the positive evaluation examples of each class, because that count determines how well the class mean is estimated, and the two counts are therefore reported separately.

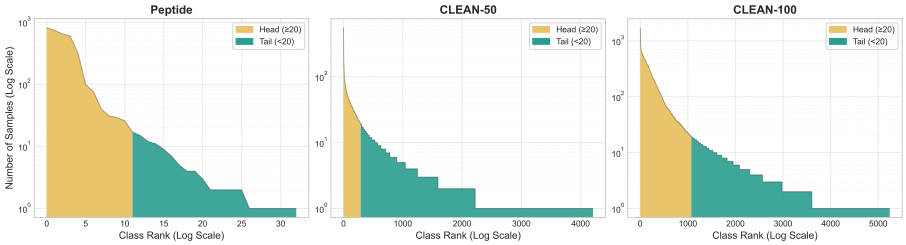

Figure 8: Training-label frequency distributions of Peptide, CLEAN-50, and CLEAN-100; tail classes have fewer than 20 training examples.

B.2 Metrics

For class i at one hierarchy level,

$$F1_i = \frac{2TP_i}{|\text{true}_i| + |\text{pred}_i|},$$

and macro-F1 averages $F1_i$ over the evaluated class set. Tables 1 and 3 use the union of classes appearing in the ground truth or predictions of each method. For analyses requiring a common class set, we instead average over leaf classes with at least one positive test example: 1,129 classes on CLEAN-50 and 2,214 on CLEAN-100. This set is fixed across methods and training subsets. Accuracy is the percentage of evaluated sequences whose predicted label set exactly matches the ground-truth label set at a given hierarchy level. Hamming loss is the fraction of incorrect binary label decisions over all evaluated sequences and classes at a hierarchy level. Tables 1 and 3 report HL in units of 10^{-3}, whereas the level-wise tables in Appendix C report the unscaled values.

Strict-path consistency is the fraction of predictions that contain exactly one class at every hierarchy level and form a valid root-to-leaf path before postprocessing. Ancestor closure, analyzed in Appendix C.4, is a separate criterion that permits multi-label outputs and empty levels.

C Complete Benchmark Analyses

This appendix provides the level-wise benchmark results, paired resampling analyses, class-frequency and sequence-level diagnostics, and hierarchy-consistency analyses supporting Sections 4.2 and 4.3.

C.1 Level-Wise Results

Tables 7 to 9 report accuracy, Hamming loss, and macro-F1 at every hierarchy level and averaged over levels. The level-wise view shows where methods differ within the hierarchy. On CLEAN-50, for example, ProtoHMC has higher accuracy than SoCoBio at the first three levels, whereas SoCoBio has higher F1 at the second to fourth levels and higher leaf accuracy.

Table 7: Performance comparison on Level 1, Level 2, Level 3, and Mean-level metrics.

Level	L1-level			L2-level			L3-level			Mean-level		
	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1
F-MLP	80.82	0.1065	83.41	62.33	0.0539	44.91	61.42	0.0342	28.96	68.19	0.0649	52.43
S-MLP	82.42	0.1043	83.75	64.61	0.0537	56.25	63.01	0.0341	33.15	70.02	0.0640	57.72
FGVC	81.51	0.1073	83.59	65.07	0.0525	55.12	61.64	0.0332	25.60	69.41	0.0644	54.77
HIFD2	82.83	0.1192	83.41	63.93	0.0558	52.54	60.73	0.0338	23.35	69.16	0.0696	53.10
SCAD	82.88	0.1012	84.71	63.88	0.0605	53.73	60.00	0.0395	29.32	68.92	0.0671	55.92
HMCN	86.76	0.0875	86.77	74.20	0.0461	**68.15**	70.55	0.0304	42.41	77.17	0.0547	65.78
SPUR	83.79	0.1050	83.33	69.18	0.0548	63.06	67.12	0.0347	39.02	73.36	0.0648	61.80
LLAMP	84.02	0.1050	84.36	65.07	0.0646	55.26	65.30	0.0368	25.09	71.46	0.0688	54.90
AMPlify	82.88	0.1058	84.03	64.38	0.0571	54.43	62.33	0.0361	35.93	69.86	0.0663	58.13
ProtoHMC	81.28	0.1134	82.41	56.16	0.0580	34.32	51.83	0.0377	20.09	63.09	0.0697	45.61
SoCoBio	**87.67**	**0.0807**	**88.19**	**76.48**	**0.0429**	67.13	**73.74**	**0.0300**	**43.37**	**79.30**	**0.0512**	**66.23**

Table 8: Performance comparison on Level 1, Level 2, Level 3, Level 4, and Mean-level metrics.

Level	L1-level			L2-level			L3-level			L4-level			Mean-level		
	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1
F-MLP	22.43	0.122582	16.69	2.14	0.016897	1.14	0.91	0.005697	0.25	0.00	0.000944	0.00	6.37	0.036530	4.52
S-MLP	93.87	0.012186	94.87	87.95	0.002559	76.50	84.75	0.001028	71.82	72.63	0.000303	55.98	84.80	0.004019	74.79
FGVC	87.73	0.022868	90.85	83.81	0.003195	74.32	82.21	0.001139	67.09	36.30	0.000617	14.61	72.51	0.006955	61.72
HIFD2	73.32	0.056572	79.19	58.51	0.012177	48.13	50.13	0.005062	43.94	20.36	0.001949	26.77	50.58	0.018940	49.51
SCAD	94.01	0.013845	93.88	89.47	0.002441	72.78	85.08	0.001093	66.84	56.52	0.000442	36.41	81.27	0.004455	67.48
HMCN	85.77	0.029868	88.26	77.75	0.004570	62.54	69.69	0.001873	48.41	23.96	0.000722	9.83	64.29	0.009258	52.26
SPUR	87.08	0.025927	89.32	79.56	0.004124	70.70	75.32	0.001616	63.71	62.58	0.000387	47.01	76.13	0.008013	67.69
CLEAN	87.89	0.014800	90.06	77.82	0.005700	70.86	68.38	0.002500	56.98	21.89	0.000900	10.27	63.99	0.005900	57.04
ProtoHMC	**96.41**	**0.009489**	**95.77**	**93.65**	**0.001630**	81.43	**90.74**	**0.000690**	73.56	54.99	0.000476	28.55	83.95	**0.003071**	69.83
SoCoBio	95.68	0.010318	95.47	92.49	0.001773	**84.11**	89.55	0.000765	**75.66**	**76.26**	**0.000279**	**59.55**	**88.49**	0.003284	**78.70**

Table 9: Performance comparison on Level 1, Level 2, Level 3, Level 4, and Mean-level metrics.

Level	L1-level			L2-level			L3-level			L4-level			Mean-level		
	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1	Acc	HL	F1
F-MLP	49.50	0.096101	61.76	46.81	0.009109	63.52	53.35	0.002501	64.21	73.07	0.000143	61.90	55.68	0.026963	62.85
S-MLP	99.02	0.001982	99.25	97.78	0.000422	95.58	96.73	0.000187	92.34	92.76	0.000041	78.38	96.57	0.000658	91.39
FGVC	98.97	0.002244	99.01	97.78	0.000431	92.90	97.31	0.000162	91.85	94.63	0.000032	81.05	97.18	0.000717	91.20
HIFD2	98.99	0.002500	97.53	89.64	0.001700	75.76	76.33	0.000900	54.67	32.26	0.002400	13.61	74.30	0.001800	60.39
SCAD	99.12	0.001842	99.26	97.24	0.000508	92.08	95.62	0.000251	88.54	91.04	0.000050	69.31	95.75	0.000663	87.30
HMCN	97.50	0.005101	98.12	96.03	0.000755	94.65	94.48	0.000301	84.75	78.07	0.000112	44.44	91.52	0.001567	80.49
SPUR	97.66	0.004807	98.24	95.44	0.000837	90.67	94.28	0.000327	85.78	85.50	0.000080	60.59	93.22	0.001512	83.82
CLEAN	98.44	0.003300	98.83	96.00	0.000700	89.95	92.90	0.000400	73.08	58.73	0.000200	16.15	86.52	0.001100	69.50
ProtoHMC	99.21	0.001597	99.36	98.45	0.000301	93.29	97.37	0.000156	86.69	87.61	0.000071	56.29	95.66	0.000531	83.91
SoCoBio	**99.32**	**0.001523**	**99.40**	**98.87**	**0.000226**	**97.06**	**98.57**	**0.000085**	**95.00**	**95.53**	**0.000027**	**82.79**	**98.07**	**0.000465**	**93.56**

C.2 Paired Test-Set Resampling

Figure 9 evaluates the variability of SoCoBio–baseline F1 differences using paired sequence-cluster resampling. Each of the $2{,}000$ replicates samples identical-sequence clusters with replacement and evaluates both methods on the same resampled test set.

On CLEAN-100, the leaf-F1 difference between SoCoBio and FGVC is 1.74 points with a 95% percentile range of $[1.02, 2.15]$; on CLEAN-50, the difference from S-MLP is 3.57 points with a range of $[1.83, 3.79]$. These are percentile ranges of paired resampled differences rather than confidence intervals centered on the observed difference. Because each method's evaluated class set is fixed from the full test split before resampling, the percentile range need not contain the full-test difference.

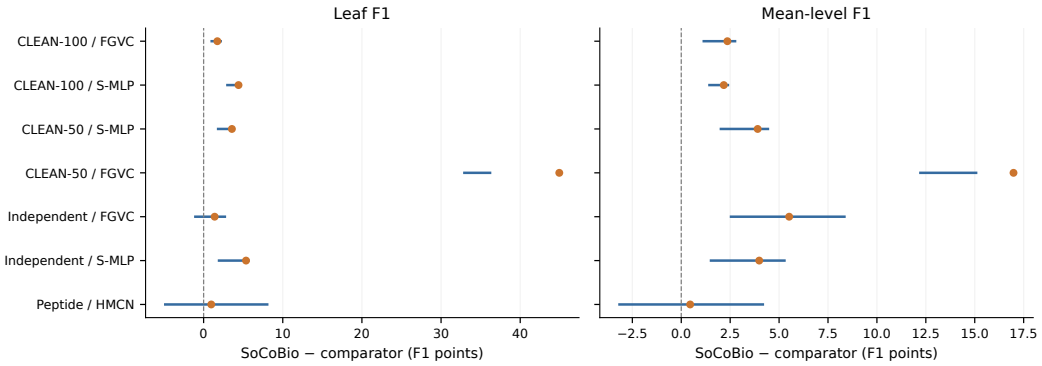

Figure 9: Paired sequence-cluster resampling of SoCoBio-minus-comparator F1 differences; points show full-test differences and segments show 95% percentile ranges.

C.3 Frequency Groups, Sibling Confusion, and Prediction Cases

This subsection supports the long-tail analysis of Section 4.3 and the sequence-level comparisons associated with Section 4.2. It reports class-frequency results, leaf-error categories, and representative CLEAN-100 prediction cases.

Frequency groups. Table 10 gives the group F1 plotted in Figure 1. On CLEAN-100, SoCoBio reaches 74.49 on tail classes against 73.52 for FGVC and 69.52 for S-MLP, and it also leads on head and middle classes (98.59 and 93.73 against 98.37 and 92.89 for FGVC). On Peptide, the tail F1 of SoCoBio, 21.30, exceeds HMCN at 20.88 and AMPlify at 20.63.

For all datasets, head, middle, and tail contain classes with at least 100, 20–99, and 1–19 training examples, respectively, and are evaluated over fixed test-supported class sets.

Table 10: Macro-F1 (%) by training-frequency group on Peptide and CLEAN-100.

	Peptide			CLEAN-100		
Method	Head	Middle	Tail	Head	Middle	Tail
F-MLP	71.68	20.00	7.22	84.88	71.33	51.57
S-MLP	71.83	<u>52.27</u>	7.41	97.92	91.10	69.52
FGVC	71.93	25.00	0.00	<u>98.37</u>	<u>92.89</u>	<u>73.52</u>
HIFD2	63.86	23.63	4.77	69.80	91.98	17.24
SCAD	66.91	18.68	5.85	97.30	87.73	51.03
HMCN	<u>77.58</u>	51.32	<u>20.88</u>	90.93	59.99	19.71
SPUR	73.54	45.00	18.52	94.65	79.47	46.42
LLAMP	73.02	18.18	0.00	–	–	–
AMPlify	70.85	17.42	20.63	–	–	–
CLEAN	–	–	–	77.49	72.16	24.01
ProtoHMC	64.29	14.62	0.00	96.55	78.35	31.25
Horizontal-only (H)	77.09	46.75	18.52	85.64	80.22	56.43
SoCoBio	**77.60**	**57.14**	**21.30**	**98.59**	**93.73**	**74.49**

Sequence-level agreement. On the 17,446 aligned CLEAN-100 test sequences, SoCoBio and FGVC both exactly match all annotated hierarchy levels on 16,093 sequences. SoCoBio alone is correct on 538, FGVC alone on 175, and neither on 640. Overall, SoCoBio has fewer label errors on 762 sequences and more on 300, while the remaining cases tie in error count. The test set contains multi-leaf examples, so exact label-set agreement is distinct from strict-path consistency.

Sibling confusion. Table 11 partitions single-leaf predictions into exact, sibling, cross-parent, empty, and multi-leaf outcomes. Among the 1,440 tail examples, SoCoBio predicts the exact leaf for

1,109 sequences versus 1,055 for FGVC, while sibling confusions remain similar (81 versus 76). Its additional exact predictions mainly coincide with fewer cross-parent (14 versus 32) and multi-leaf (34 versus 83) outputs. Thus, the tail improvement does not arise from increased confusion among true siblings.

Sibling denotes one incorrect leaf under the true L3 parent, cross-parent denotes one incorrect leaf under another L3 parent, empty denotes no predicted leaf, and multi-leaf denotes multiple predicted leaves.

Table 11: Leaf-level prediction outcomes on the deduplicated CLEAN-100 test set.

	All 16,502 single-leaf sequences					1,440 tail sequences				
Method	Exact	Sibling	Cross-parent	Empty	Multi-leaf	Exact	Sibling	Cross-parent	Empty	Multi-leaf
F-MLP	12,170	97	383	3,144	708	687	31	70	566	86
S-MLP	15,408	136	62	625	271	977	81	30	250	102
FGVC	15,704	125	66	433	174	1,055	76	32	194	83
SCAD	15,168	162	54	1,021	97	783	101	27	494	35
HMCN	13,086	165	18	3,088	145	250	72	10	1,069	39
SPUR	14,231	233	85	1,408	545	599	103	34	537	167
SoCoBio	15,830	132	37	394	109	1,109	81	14	202	34

Prediction cases. Figure 10 shows representative corrected, regressed, and unchanged CLEAN-100 predictions from the aligned SoCoBio and FGVC outputs. We restrict the examples to tail classes with 1–19 training examples and exactly one ground-truth and predicted label per level, and choose a deterministic representative from each outcome group. The corrected case has leaf support 12, whereas the regressed and unchanged cases have support 5 and 4, respectively.

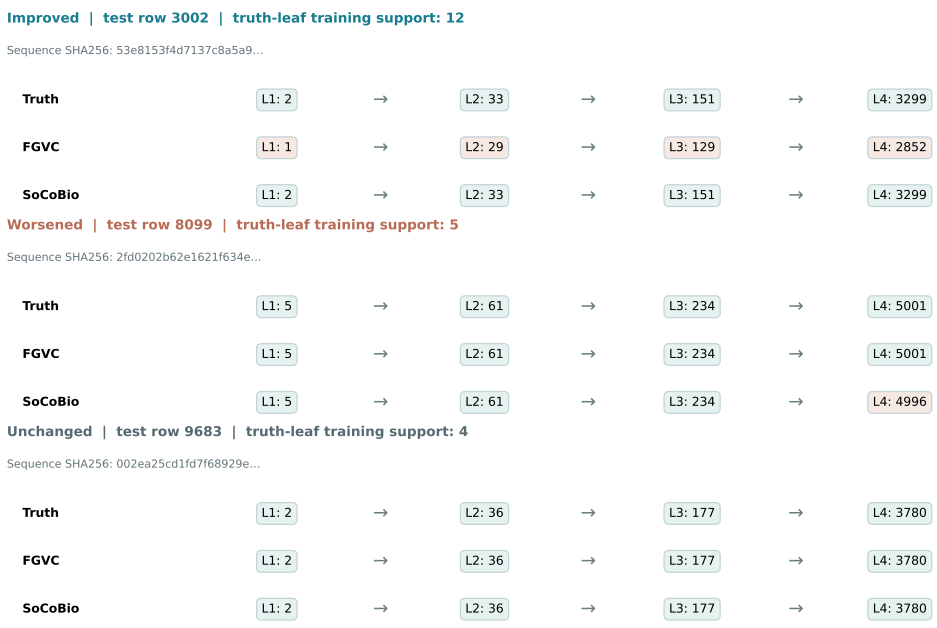

Figure 10: Representative corrected, regressed, and unchanged CLEAN-100 predictions for SoCoBio and FGVC.

C.4 Strict-Path Consistency and Ancestor Closure

Table 12 reports run-to-run variability in strict-path consistency for representative CLEAN-50 and CLEAN-100 comparisons. The CLEAN-50 gap between SoCoBio and S-MLP remains substantially larger than the observed variability.

Table 12: Strict-path consistency across repeated runs, reported as mean ± standard deviation (%).

Dataset	Method	Strict-path consistency
CLEAN-50	S-MLP	68.91 ± 1.51
	FGVC	36.06 ± 0.55
	SoCoBio	76.33 ± 0.47
CLEAN-100	SoCoBio	91.63 ± 0.06

Table 13 separates structural validity from predictive performance. Ancestor closure makes every prediction ancestor-closed while leaving leaf predictions unchanged. SoCoBio is already 99.68% ancestor-closed before postprocessing, and closure changes its mean-level F1 only from 93.56 to 93.59 while leaving leaf F1 at 82.79. Thus, structural closure alone does not account for the predictive improvement.

Table 13: Effect of ancestor closure on hierarchical consistency and predictive performance on CLEAN-100.

	Closed (%)		Mean-level F1		
Method	Raw	After	Raw	After	Leaf F1
F-MLP	47.52	100.00	62.85	74.51	61.90
S-MLP	97.64	100.00	91.39	91.56	78.38
FGVC	98.05	100.00	91.20	91.48	81.05
SCAD	97.73	100.00	87.30	88.03	69.31
HMCN	98.76	100.00	80.49	80.74	44.44
SPUR	96.90	100.00	83.82	83.85	60.59
ProtoHMC	99.25	100.00	83.91	84.05	56.29
SoCoBio	99.68	100.00	93.56	93.59	82.79

D Fixed-Checkpoint Analysis

This appendix supports Section 4.4 by analyzing fixed SoCoBio checkpoints without retraining. We evaluate the empirical quantities in Theorems 3.1 and 3.2, study sensitivity to interaction strength and positive support, and examine peer selection, controller gates, and prototype geometry.

D.1 Residual Correction and Local Curvature

Table 14 decomposes the BCE gain of the joint pass over each comparator A into the terms of Theorem 3.1. The residual correction $\widehat{\mathcal{A}}_A$ measures how far the joint update corrects the residual error of the comparator, and the curvature cost $\widehat{\mathcal{C}}_A$ is the one-segment cost of the theorem, so $\widehat{\mathcal{A}}_A - \widehat{\mathcal{C}}_A$ is its lower bound. The realized second-order term equals the residual correction minus the measured gain, which is the exact integral remainder, so it lies between zero and the curvature cost in every row. The global bound $Q/8$ replaces the local curvature by its maximum $1/4$ and exceeds the curvature cost by more than three orders of magnitude in every row, so the local calculation in Table 15 is the informative one for these fixed updates.

For the final checkpoint, all three lower bounds are positive, and the joint pass lowers BCE against each comparator. At the final checkpoint against disabled interaction, the residual correction of 11.340×10^{-6} minus the curvature cost of 3.546×10^{-6} gives the lower bound 7.794×10^{-6} quoted in Section 4.4. For the intermediate checkpoint, all three measured gains are negative, and against disabled interaction its residual correction of 2.775×10^{-6} is below the realized second-order term of 3.242×10^{-6}. The eight-segment calculation narrows the final $J - 0$ interval from

Table 14: BCE-gain decomposition from Theorem 3.1 at the intermediate and final CLEAN-100 checkpoints, in units of 10^{-6}.

Checkpoint	Comparator	Residual correction	Curvature cost	Realized second-order term	BCE gain	Global bound $Q/8$
Intermediate	0	2.775	4.757	3.242	−0.467	15300.687
Intermediate	V	−0.773	0.592	0.514	−1.287	3345.804
Intermediate	H	1.133	1.971	1.466	−0.332	6344.941
SoCoBio	0	11.340	3.546	2.418	8.922	31569.247
SoCoBio	V	3.765	0.530	0.454	3.311	6673.002
SoCoBio	H	5.816	1.225	0.936	4.880	12297.548

Table 15: Deterministic empirical bounds on the BCE gain, in units of 10^{-6}, using one- and eight-segment curvature approximations.

Checkpoint	Comparator	Gain	Lower $m = 1$	Upper $m = 1$	Lower $m = 8$	Upper $m = 8$
Intermediate	0	−0.467	−1.982	0.618	−0.625	−0.315
Intermediate	V	−1.287	−1.365	−1.224	−1.295	−1.278
Intermediate	H	−0.332	−0.838	0.024	−0.385	−0.282
SoCoBio	0	8.922	7.794	9.817	8.792	9.049
SoCoBio	V	3.311	3.235	3.376	3.302	3.320
SoCoBio	H	4.880	4.591	5.111	4.847	4.912

$[7.794, 9.817] \times 10^{-6}$ to $[8.792, 9.049] \times 10^{-6}$ around the measured gain of 8.922×10^{-6}. The local bounds therefore separate the favorable final updates from the unfavorable intermediate ones, and every measured gain lies within its lower and upper bounds.

D.2 Sensitivity to Interaction Strength

Figure 11 evaluates how the fixed-checkpoint conclusions change when the magnitude of the learned controller update is rescaled at inference time, while all trained parameters remain frozen. For each interaction strength, we recompute the joint-pass BCE gain, the one-segment lower bound of Theorem 3.1, and the residual correction against 0, V, and H. As the interaction becomes stronger, residual correction alone does not determine the sign of the BCE gain. For example, at an interaction scale of 0.10, the $J - 0$ gain remains positive although its lower bound becomes negative, while the $J - V$ gain becomes negative despite positive residual correction. At 0.20, the $J - 0$ gain is also negative. These results illustrate that the first-order correction must be balanced against the finite-step curvature cost.

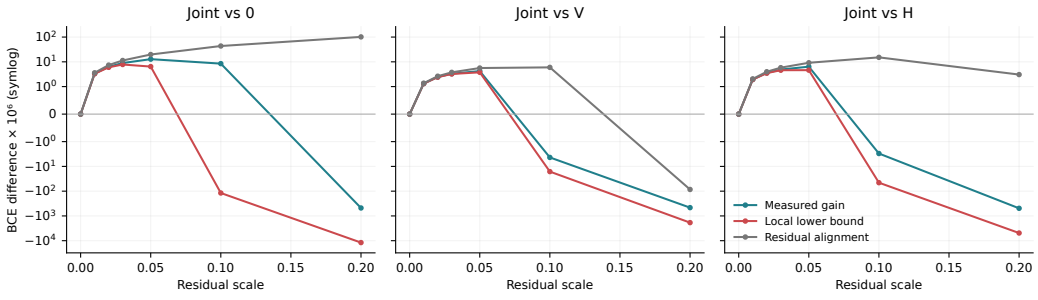

Figure 11: Sensitivity of the CLEAN-100 fixed-checkpoint BCE gain and its local lower bound to interaction strength.

D.3 Prediction Loss and Prototype Direction

Table 16 shows why direction error and prediction loss need separate analysis. On CLEAN-100, the final checkpoint lowers both, whereas the intermediate checkpoint lowers direction error while raising BCE. The CLEAN-50 checkpoint shows the opposite separation, as BCE improves while direction error rises. The CLEAN-50 row uses the $2{,}780$ sequences of the CLEAN-50 validation split, on which $1{,}424$ classes have positives, and is not pooled with the CLEAN-100 test measurements.

Table 16: Prototype-direction error of the four controller passes and BCE gain of J over 0.

Checkpoint	0	V	H	J	BCE gain $\times 10^6$
C100 Intermediate	1.93271	1.91257	1.92269	1.90244	-0.467
C100 SoCoBio	1.89234	1.87382	1.88107	1.86244	8.922
C50 SoCoBio	1.88761	1.88768	1.89682	1.89689	21.059

D.4 Positive Support and Projection Estimation

The directional result of Theorem 3.2 depends on the projection of an update onto the positives of a class and on how accurately that projection is estimated. Table 17 counts the classes whose empirical projection is positive for each checkpoint and comparator.

Table 17: Number of CLEAN-100 classes with positive empirical update projection.

Checkpoint	Comparator	Positive projection
SoCoBio	0	2445
SoCoBio	V	2372
SoCoBio	H	2446
Intermediate	0	2381
Intermediate	V	2355
Intermediate	H	2345

To examine the estimation accuracy, we vary the positive support while keeping the class set fixed. The training-frequency groups in Figure 1 describe how many annotations a class has during training, whereas the support sizes here determine how accurately a fixed update can be assessed. For each class, the test positives are split into a subsampling pool and a disjoint reference pool, and the projection is estimated 50 times from random subsamples of each size and compared with the projection on the reference pool. The same classes, those with at least 32 positives in the subsampling pool, are used at every support size, namely 135 classes for the comparisons with 0 and H and 128 for the comparison with V. For the final $J - 0$ comparison, the mean squared projection error falls from 0.0020014 at one example to 0.0000762 at 32 examples (Table 18), so the root mean squared error falls to 0.195 of its one-example value, close to the $32^{-1/2} \approx 0.177$ reference in Figure 7(b). This empirical trend is consistent with the $n_i^{-1/2}$ dependence of the sampling allowance in Theorem 3.2.

Table 18: Projection-estimation MSE ($\times 10^4$) as positive support increases on CLEAN-100.

Checkpoint	Comparator	Classes	1	2	4	8	16	32
Intermediate	0	135	17.178	8.040	4.091	2.250	1.145	0.660
Intermediate	V	128	16.330	7.587	4.177	2.051	1.094	0.628
Intermediate	H	135	14.971	8.364	4.301	2.342	1.336	0.860
SoCoBio	0	135	20.014	9.807	4.860	2.623	1.337	0.762
SoCoBio	V	128	17.237	8.506	4.621	2.174	1.211	0.696
SoCoBio	H	135	16.898	8.792	4.812	2.641	1.499	0.988

D.5 VALIDATION-SPLIT DIAGNOSTICS

This subsection evaluates the empirical counterparts of Theorems 3.1 and 3.2 on the CLEAN-100 validation split using the same fixed checkpoints and controller passes as above. All model parameters, fused sequence representations, normalization statistics, and peer-selection states are held fixed. The reconstructed four-pass readout matches the original forward computation.

We use the ordinary BCE analyzed in Theorem 3.1, weighting the four hierarchy levels equally and classes uniformly within each level, $\omega_i = 1/(4K_{\ell(i)})$. This diagnostic loss is distinct from the complete training objective. Variance is computed across sequences, preserving dependence among labels within the same sequence.

Table 19: Paired BCE differences and nominal lower endpoints on the CLEAN-100 validation split.

Checkpoint	Comparator	Mean $\times 10^6$	B_A	Nominal lower
Intermediate	0	1.2131	2.12259	-2.710×10^{-3}
Intermediate	V	-0.3105	1.09596	-1.400×10^{-3}
Intermediate	H	0.5044	1.37082	-1.751×10^{-3}
SoCoBio	0	8.9827	3.14029	-4.003×10^{-3}
SoCoBio	V	3.2140	1.65238	-2.108×10^{-3}
SoCoBio	H	5.1174	1.99258	-2.541×10^{-3}

The primary family contains six predefined comparisons, namely two checkpoints and three comparator passes, with nominal $\delta = 0.05$. Half the error budget is assigned to each of the Hoeffding and empirical Bernstein bounds before the smaller radius is taken. All six lower endpoints are negative. At the intermediate checkpoint, the joint pass lowers empirical BCE relative to the disabled and sibling-only passes but raises it relative to the parent-only pass, whereas at the final checkpoint it lowers empirical BCE against all three comparators. On the test split, the intermediate joint pass instead raises BCE against all three comparators (Table 14), so the sign of these small differences depends on the sample. The direction error of the intermediate joint pass is lower than that of every comparator on both splits, so a geometrically favorable update need not reduce BCE relative to every comparator in the actual readout.

Applying the directional diagnostic to the predefined class–pass family yields nominal positive endpoints for only a small subset of comparisons. For $(0, V, H)$, the counts are $(18, 11, 11)$ at the intermediate checkpoint and $(15, 8, 13)$ at the final checkpoint. These are descriptive validation-split measurements rather than independent calibration guarantees.

D.6 STORED CANDIDATES, ATTENTION, AND GATES

We inspect the learned peer-selection states of the final CLEAN-100 checkpoint together with the Vertical and Horizontal gates. The resulting directed peer graph contains $78{,}160$ selected interactions between distinct classes at the same hierarchy level. A reverse interaction is present for 63.5184% of directed edges, corresponding to 46.5399% reciprocal connected pairs; the two percentages use different denominators.

For target class 3299, the peer-selection distribution before hard discretization assigns probabilities of approximately 0.999642, 0.000336, and 0.0000219 to classes 3301, 3302, and 3300, respectively. Figure 12 visualizes the dominant incoming peer relations. These scores are computed from class prototypes and learned relation parameters and are shared across input sequences.

Across classes with available peer interactions, the Horizontal Controller gate has mean 0.841537, while the entropy of the soft peer-selection distribution is 0.52024. Across non-root classes, the Vertical Controller gate has mean 0.9999972 and minimum 0.9998767. Thus, parent information is almost fully admitted at this checkpoint, whereas most selectivity arises from the learned peer-selection mechanism rather than the vertical gate.

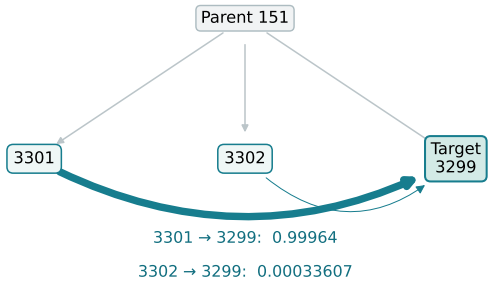

Figure 12: Learned peer-selection scores for target class 3299 at the final CLEAN-100 checkpoint.

D.7 Prototype Geometry Across Hierarchy Branches

We compare prototype geometry between matched joint and disabled-interaction training runs. Cosine similarities are computed within each run before aggregation, avoiding comparisons between the coordinate systems of separately trained models. We distinguish the persistent prototype parameters P from the controller-transformed prototypes used by the readout.

Table 20: Mean cosine-similarity difference between joint and disabled-interaction training for prototype pairs in different hierarchy groups.

Class-pair group	Pairs	Stored	Updated
Leaves, same L3	288,851	-0.010302	$+0.002197$
Leaves, different L3	13,447,810	-0.010291	$+0.002696$
L3, same L2	726	-0.014067	-0.001880
L3, different L2	31,152	-0.012244	-0.000118

The geometry changes are visible both within and across hierarchy branches. Because Horizontal interaction is defined over same-level peers, these aggregate pairwise changes should not be interpreted as evidence that a particular pair did or did not directly exchange a message.

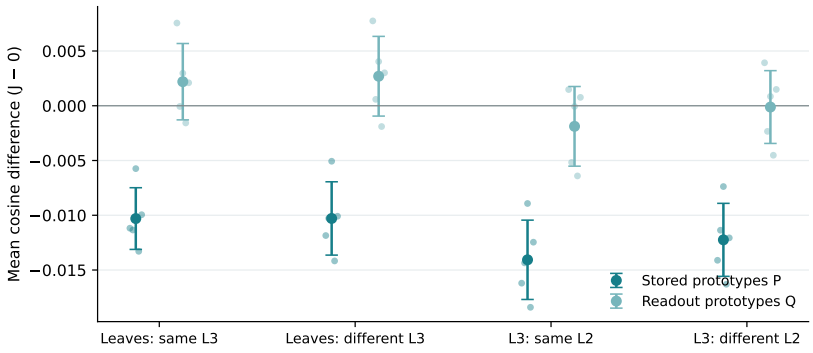

Figure 13: Prototype-geometry differences between joint and disabled-interaction training across hierarchy groups on CLEAN-100.

The recovered EC mapping agrees with all 5,242 leaf paths in the model taxonomy. For pair-level analysis, we restrict attention to the 2,214 leaf classes with positives in the deduplicated CLEAN-100 test set, yielding 2,406,051 cross-branch pairs. Pairs are ranked by the joint-minus-disabled change in cosine similarity, while both the largest increases and decreases are retained. A positive difference indicates greater similarity under joint training, but does not by itself imply movement toward the shared initialization, a direct peer interaction, or a biologically meaningful relation. Sequence-based

Table 21: Illustrative cross-branch leaf pairs among the largest increases in cosine similarity under joint versus disabled training.

Rank	EC pair	Cosine (init / 0 / J)	Train	Test	F1 ↑/↓ first	F1 ↑/↓ second
1	1.9.6.1 – 1.12.2.1	0.330 / 0.245 / 0.288	106 / 11	10 / 1	0 / 0	1 / 1
5	1.7.2.4 – 1.7.99.1	0.382 / 0.310 / 0.348	9 / 131	2 / 10	0 / 0	0 / 0
8	2.7.2.3 – 3.3.1.1	0.359 / 0.345 / 0.382	575 / 202	67 / 19	0 / 0	1 / 0
11	1.9.6.1 – 1.18.6.1	0.392 / 0.348 / 0.384	106 / 152	10 / 22	0 / 0	0 / 0
17	1.1.1.381 – 2.4.1.237	0.319 / 0.324 / 0.358	4 / 3	3 / 2	2 / 1	0 / 0
34	2.4.1.83 – 7.3.2.1	0.313 / 0.324 / 0.356	10 / 303	3 / 29	2 / 0	1 / 0
82	1.9.6.1 – 2.3.1.43	0.330 / 0.228 / 0.257	106 / 8	10 / 2	0 / 0	3 / 0
87	1.2.1.70 – 1.14.13.114	0.338 / 0.284 / 0.313	476 / 2	40 / 1	0 / 0	3 / 0

consistency checks exclude the 34 source records with label-set discrepancies from named biological examples.

Among the 100 largest similarity increases, only 22 pairs become more similar than at the shared initialization. Only two pairs show repeated endpoint F1 improvements across matched runs, and both remain farther apart than at initialization. These examples therefore demonstrate geometric change rather than a validated biological relationship. Establishing the latter would require independent biological evidence and a targeted intervention.

E Formal Analysis and Proofs

This appendix proves the two results of Section 3.6. Theorem 3.1 bounds the BCE gain of the joint pass, Theorem 3.2 bounds its improvement in distance to the positive-class mean direction, and the last subsection derives the local curvature bound used in Section 4.4. Both results concern four passes of one frozen predictor, namely the disabled pass 0, the parent-only pass V, the sibling-only pass H, and the joint pass J. The proofs connect the measured residual correction and positive-class projection to sufficient conditions for a BCE gain and a directional gain. Appendix E.1 defines the statistics, Appendix E.2 states the shared assumptions, and Appendices E.3 and E.4 restate and prove each theorem. Appendix E.5 explains how the simultaneous bounds may be used, and Appendix E.6 derives the local curvature bound and its refinement. Appendix F compares the two results with prior theory.

This appendix proves the two theoretical results of Section 3.6. Theorem 3.1 characterizes the BCE gain of the joint controller pass, while Theorem 3.2 characterizes its improvement toward the positive-class feature direction. Both results compare four passes of one frozen predictor: disabled interaction (0), Vertical-only (V), Horizontal-only (H), and joint interaction (J). Appendix E.1 defines the statistics, Appendix E.2 states the shared assumptions, Appendices E.3 and E.4 prove the two theorems, Appendix E.5 discusses their simultaneous use, and Appendix E.6 derives the localized-curvature refinement used in Section F.

E.1 Setup and Statistics for the Two Theorems

The benchmark and ablation experiments evaluate trained predictors. Here we explain an individual predictor by measuring how its controller updates change loss and prototype direction. Population statements require calibration observations independent of training and model selection; development observations support empirical diagnostics. We compare the same checkpoint with controllers disabled (0), vertical-only (V, parent inheritance), horizontal-only (H, same-level peer interaction), or joint (J, both). These frozen passes share trained parameters, fused sequence representations, and the linear head; they differ only in the applied controller transformations and the resulting unit prototypes $u_i^A = P_i^A/\|P_i^A\|$. For a nonzero vector v, write $\bar{v} = v/\|v\|$, let $\ell_i^A(x)$ denote the class logit, and let $y_{ki} = 1$ when calibration protein k has label i. The number of labeled calibration proteins is n. We take $0 < \delta \leq 1/2$ throughout.

BCE statistics for Theorem 3.1. We first assess prediction loss through the logit changes produced by a prototype update. For $A \in \{0, V, H\}$, let $d_i^A = u_i^J - u_i^A$ and $c_i = (1 - \lambda_{\ell(i)})/\tau$, using the

mixed readout in Eq. (12). Here the level of class i selects the mixing weight, and the temperature scales the prototype branch. The shared linear branch cancels when the two logits are subtracted. With $\bar{z}(x) = z(x)/\|z(x)\|$, define the logit change $q_{ki}^A = \ell_i^J(x_k) - \ell_i^A(x_k) = c_i\langle \bar{z}(x_k), d_i^A\rangle$ and the comparator's residual $r_{ki}^A = y_{ki} - \sigma(\ell_i^A(x_k))$. The residual–change product is positive when the update moves the predicted probability toward the true label. Let κ_{ki}^A be the maximum logistic curvature $\sigma(t)(1-\sigma(t))$ between the two logits. Curvature accounts for the fact that BCE changes nonlinearly with the logit. With fixed class weights $\omega_i \geq 0$ satisfying $\sum_i \omega_i = 1$, we define residual correction, curvature cost, and the bound on the absolute paired contrast:

$$\widehat{\mathcal{A}}_A = \frac{1}{n}\sum_{k,i} \omega_i r_{ki}^A q_{ki}^A, \quad \widehat{\mathcal{C}}_A = \frac{1}{2n}\sum_{k,i} \omega_i (q_{ki}^A)^2 \kappa_{ki}^A, \quad B_A = \sum_i \omega_i c_i \|d_i^A\|.$$

Let R_A be the population weighted BCE, and let $\widehat{\Delta}_A$ average the comparator-minus-joint loss difference over the n proteins. Thus, a positive gain favors J. Residual correction captures the first-order effect, while curvature bounds the cost of taking a finite step. Theorem 3.1 combines them into a lower bound on the gain.

Directional statistics for Theorem 3.2. Alongside this loss-based assessment, we ask whether the prototype update points toward the positive-class feature direction. Fix the full family of K nodes before calibration, including unsupported nodes. For $n_i = \sum_k y_{ki} > 0$ and $d_i^A \neq 0$, the empirical projection and its uncertainty allowance are given by

$$\bar{T}_i^A = \frac{1}{n_i}\sum_{k:y_{ki}=1} \left\langle \bar{z}(x_k), \frac{d_i^A}{\|d_i^A\|}\right\rangle, \qquad \epsilon_i = \sqrt{\frac{2\log(6K/\delta)}{n_i}}, \qquad L_i^A = \bar{T}_i^A - \epsilon_i.$$

The positive-class mean is $m_i = \mathbb{E}[\bar{z}(X) \mid Y_i = 1]$, with direction $\mu_i^{\mathrm{dir}} = m_i/\|m_i\|$ when nonzero. The statistic $\bar{T}_i^A$ estimates how strongly this mean points along the update. Subtracting the sampling uncertainty gives L_i^A, a lower estimate of that projection. To express how this uncertainty varies along the tail, order the counts of the $K_+ > 0$ supported leaves decreasingly (rank one has the largest positive support) and set $C_\alpha = \min_{r \leq K_+} n_{(r)} r^\alpha$ for any $\alpha > 0$. This measured count envelope, which requires no fitted power law, enters Theorem 3.2.

E.2 Shared Assumptions

Shared sampling and readout assumptions. Both the BCE and direction comparisons concern passes of one frozen model. Condition on the complete training and model-selection history. The encoder and fusion network, four controller passes, unit prototype vectors, readout parameters, fixed analyzed node family of size K, and nonnegative weights ω_i summing to one are frozen before observing the calibration sample. Any randomness used to instantiate the frozen controller passes is also fixed beforehand. The $n \geq 1$ proteins are independent and identically distributed from the target population, and independent of that history; coordinates of the multilabel vector within a protein may be dependent. Assume $\|\bar{z}(X)\| \leq 1$ almost surely and the shared-branch readout

$$\ell_i^A(X) = a_i(X) + c_i\langle \bar{z}(X), u_i^A\rangle, \qquad 0 \leq c_i < \infty, \tag{12}$$

where $a_i(X)$ is the contribution shared by all passes and c_i is the fixed scale of the prototype branch. We require finite logits almost surely and finite expected BCE risks. The common $a_i(X)$ need not be bounded. Every direction-comparison prototype has norm exactly one; otherwise the squared-distance difference has an additional difference-of-norms term. Take $0 < \delta \leq 1/2$ fixed in advance. Classes of zero population positive probability have no positive calibration observations almost surely and yield no directional certificate. No lower bound on population mean norm, coherence, oracle radius, or tail frequency is assumed. The two theorem events use the same calibration set and need not be independent.

Composition of the controller passes. The measured displacement connects the controller implementation to both proofs. A positive BCE certificate requires residual correction to exceed curvature cost and sampling uncertainty. The joint displacement includes composition effects: with

$$v_i = u_i^V - u_i^0, \qquad h_i = u_i^H - u_i^0, \qquad \eta_i^{\mathrm{comp}} = u_i^J - u_i^V - u_i^H + u_i^0,$$

one has

$$d_i^0 = v_i + h_i + \eta_i^{\text{comp}}, \qquad d_i^V = h_i + \eta_i^{\text{comp}}, \qquad d_i^H = v_i + \eta_i^{\text{comp}}.$$

These are measurable displacements; no additive-controller approximation is used.

E.3 Proof of Theorem 3.1 (BCE Gain)

The proof first bounds the observed BCE contrast by residual correction and local curvature, then controls its sampling error.

Step 1. bound the empirical loss contrast. Write $\alpha_{ki}^A = \ell_i^A(x_k)$, $\beta_{ki} = \ell_i^J(x_k)$, and $h(t) = \sigma(t)(1-\sigma(t))$. For $\phi(y,t) = \log(1+e^t) - yt$, Taylor's integral identity gives

$$\phi(y,a) - \phi(y,a+q) = (y - \sigma(a))q - q^2 \int_0^1 (1-t)h(a+tq)\,dt. \tag{13}$$

Here a and q denote the comparator logit and its joint-pass increment, and κ denotes the maximum of h on that interval. The function h is even and decreases with $|t|$. Its maximum on the closed update interval is attained at the point nearest zero, so the integral lies between zero and $\kappa/2$. Summing with nonnegative weights and averaging proves the empirical inequality in Theorem 3.1.

Step 2. control the paired sampling error. Since $|\partial_t \phi(y,t)| \le 1$ for $y \in \{0,1\}$, the per-protein paired contrast

$$D_k^A = \sum_i \omega_i \left[\phi(y_{ki}, \alpha_{ki}^A) - \phi(y_{ki}, \beta_{ki})\right]$$

satisfies $|D_k^A| \le B_A$. This follows from the shared linear–prototype readout and the normalized-feature bound: the change in each logit is at most its prototype-branch scale times the prototype displacement. The bound concerns a complete protein, so dependence among its labels does not require a separate independence assumption. For $B_A > 0$, the one-sided Hoeffding bound is

$$\Pr\{\mathbb{E}D^A < \widehat{\Delta}_A - t\} \le \exp\left(-\frac{nt^2}{2B_A^2}\right). \tag{14}$$

Use $t = B_A\sqrt{2\log(6/\delta)/n}$ and union-bound the three failure probabilities $\delta/6$. Combining this concentration event with the deterministic empirical inequality yields the claimed result. The probability bound is on the paired loss itself, so no separate concentration of the measured residual correction or curvature is required. If $B_A = 0$, $D_k^A = 0$ almost surely. Finite expected risks identify $\mathbb{E}D^A = R_A - R_J$.

E.4 Proof of Theorem 3.2 (Directional Gain)

The direction comparison needs only the projection along the actual prototype displacement. We bound this scalar projection, translate its sign into a distance comparison, and then express the support requirement by rank.

Step 1. bound the positive-class projection. Fix a node i of positive population probability and a comparator with $d_i^A \neq 0$. Conditional on this node's vector of positive indicators, its positive features are independent draws from $\bar{z}(X) \mid Y_i = 1$. Recall

$$m_i = \mathbb{E}[\bar{z}(X) \mid Y_i = 1], \qquad \theta_i^A = \left\langle m_i, \frac{d_i^A}{\|d_i^A\|} \right\rangle.$$

Because each projection lies in $[-1,1]$, for each realized $n_i = r > 0$,

$$\Pr\left\{\theta_i^A < \bar{T}_i^A - \sqrt{\frac{2\log(6K/\delta)}{r}} \,\middle|\, n_i = r\right\} \le \frac{\delta}{6K}. \tag{15}$$

Average over $r > 0$, treating zero-support cases as having no claim, then union-bound at most $3K$ fixed comparisons. This gives an event of probability at least $1 - \delta/2$ on which $\theta_i^A \ge L_i^A$ for all supported nonzero comparisons.

Step 2. convert a positive projection to directional gain. If $L_i^A > 0$, then $m_i \neq 0$ and $0 < \|m_i\| \leq 1$. Equal prototype norms imply the exact identity

$$\|u_i^A - \mu_i^{\mathrm{dir}}\|^2 - \|u_i^J - \mu_i^{\mathrm{dir}}\|^2 = 2\langle \mu_i^{\mathrm{dir}}, d_i^A \rangle = \frac{2\|d_i^A\|\theta_i^A}{\|m_i\|} \geq 2\|d_i^A\| L_i^A. \tag{16}$$

Step 3. express the support and rank conditions. Under the observed margin condition of Theorem 3.2, the stated support threshold makes $\epsilon_i \leq \gamma/2$, hence $L_i^A \geq \gamma/2 > 0$. Zero displacements yield equal prototypes directly. Finally $n_{(r)} \geq C_\alpha r^{-\alpha}$ follows from the definition of C_α; substitution gives the rank bound and solving the support inequality gives the supported-rank prefix

$$r \leq \min\left\{K_+, \left\lfloor \left(\frac{C_\alpha \gamma^2}{8\log(6K/\delta)}\right)^{1/\alpha} \right\rfloor\right\}$$

for nodes meeting the measured margin condition. A union bound between the two theorem events gives their joint probability $1 - \delta$.

E.5 Using the Simultaneous Bounds

Because the projection bounds hold simultaneously, they also support summaries organized by observed positive support. The family size K must remain the fixed total analyzed node count, not the random number of supported labels. Positive-support filtering, ranking, α, C_α, and γ may be selected after calibration: their claims are deterministic consequences of the same fixed-family event for every positive α and γ. The lower envelope is not an inferred population power law; a fitted log–log line cannot replace its minimum definition without verifying the inequalities at every rank. The sample budget is a sufficient condition on realized positive support and observed margin, not a promise that collecting that many positives produces the margin, and not a bound on the total number of proteins required.

Prototype directions, controller passes, class weights, and δ may not be selected using this calibration sample under the stated guarantee. Previously used validation data support the deterministic empirical identities only.

E.6 Local Curvature Along the Actual Logit Update

Theorem 3.1 bounds curvature along each actual logit update. We now derive that cost and refine it by partitioning the update interval. This explains why the local calculation can reveal a favorable BCE contrast that a global curvature bound obscures. Condition on fixed model parameters and nonnegative weights ω_i summing to one. Let

$$q_i = \ell_i^J(X) - \ell_i^A(X), \qquad r_i = Y_i - \sigma(\ell_i^A(X)),$$

with finite logits almost surely. Write

$$\mathcal{A} = \mathbb{E}\sum_i \omega_i r_i q_i, \qquad Q = \mathbb{E}\sum_i \omega_i q_i^2 < \infty. \tag{17}$$

For the shared mixed readout, $q_i = c_i\langle \bar{z}(X), u_i^J - u_i^A \rangle$ with $\|\bar{z}(X)\| \leq 1$ and fixed finite c_i, so these moments exist. Let Δ denote the expected paired BCE contrast. It is defined without requiring finite individual risks, since BCE is 1-Lipschitz and $Q < \infty$; to write $\Delta = R_A - R_J$, additionally require both risks finite.

Proposition E.1 (Localized curvature cost)**.** *Define $h(t) = \sigma(t)(1-\sigma(t))$, an even function decreasing on $[0, \infty)$. Let*

$$\begin{aligned} \rho_i &= \mathrm{dist}\left([\min(\ell_i^A(X), \ell_i^J(X)), \max(\ell_i^A(X), \ell_i^J(X))], 0\right), \\ \kappa_i &= h(\rho_i), \qquad C_{\mathrm{loc}} = \frac{1}{2}\mathbb{E}\sum_i \omega_i q_i^2 \kappa_i. \end{aligned} \tag{18}$$

Here ρ_i is the smallest absolute logit along the update interval. Thus $\rho_i = 0$ if the endpoints straddle or touch zero, and otherwise

$$\rho_i = \min\left(|\ell_i^A(X)|, |\ell_i^J(X)|\right).$$

Then

$$\mathcal{A} - C_{\mathrm{loc}} \le \Delta \le \mathcal{A}, \qquad 0 \le C_{\mathrm{loc}} \le Q/8. \tag{19}$$

The cost is strictly smaller than $Q/8$ exactly when

$$\Pr\big(\exists i : \omega_i q_i^2 > 0,\ \rho_i > 0\big) > 0$$

for a finite node set. In particular, $\mathcal{A} > C_{\mathrm{loc}}$ is a sufficient gain condition that is never more conservative than $\mathcal{A} > Q/8$, but neither condition is guaranteed to hold.

Proof. For $\phi(y,t) = \log(1+e^t) - yt$, the exact integral remainder retains the curvature encountered by the update:

$$\phi(Y_i, \ell_i^A) - \phi(Y_i, \ell_i^J) = r_i q_i - q_i^2 \int_0^1 (1-t) h(\ell_i^A + t q_i)\, dt. \tag{20}$$

Since h is even and decreasing with absolute logit, its maximum on the update interval is $h(\rho_i)$. The integral is nonnegative and at most $\kappa_i/2$. Weighting and taking expectations proves the sandwich. Moreover,

$$Q/8 - C_{\mathrm{loc}} = \tfrac{1}{2}\mathbb{E}\sum_i \omega_i q_i^2 [1/4 - h(\rho_i)]. \tag{21}$$

Every summand is nonnegative; $h(\rho_i) < 1/4$ exactly when $\rho_i > 0$, proving the strictness criterion. A zero increment contributes no curvature cost, whether or not the interval touches zero. □

Stable evaluation and saturation. For $\rho \ge 0$, evaluate

$$h(\rho) = \frac{e^{-\rho}}{(1+e^{-\rho})^2}$$

to avoid overflow. Intervals that remain far from zero have a much smaller curvature cost than the global bound. If an update crosses zero, the one-interval bound reverts to $1/4$ even when most of its path is saturated. Neither saturation nor localization alone establishes beneficial residual correction. Numerical underflow can round a strictly positive curvature to zero even though the mathematical cost remains positive, so conservative rounding or higher precision is required when numerically certifying the bound.

Refining the interval bound. A single interval can still be conservative when the update crosses zero. Subdivide the path to obtain upper and lower bounds on its integrated curvature cost. For a partition $\mathcal{P} : 0 = t_0 < t_1 < \cdots < t_N = 1$, let

$$w_j = \int_{t_{j-1}}^{t_j} (1-t)\, dt = (t_j - t_{j-1}) \left(1 - \frac{t_j + t_{j-1}}{2}\right). \tag{22}$$

On the subinterval with logit endpoints $\ell_i^A + t_{j-1} q_i$ and $\ell_i^A + t_j q_i$, let κ_{ij} and $\underline{\kappa}_{ij}$ be the maximum and minimum of h, respectively. The maximum is computed by distance to zero as above; the minimum is h at the endpoint with greater absolute value. Define

$$C_{\mathcal{P}}^{+} = \mathbb{E}\sum_i \omega_i q_i^2 \sum_j w_j \kappa_{ij}, \qquad C_{\mathcal{P}}^{-} = \mathbb{E}\sum_i \omega_i q_i^2 \sum_j w_j \underline{\kappa}_{ij}. \tag{23}$$

Then

$$\mathcal{A} - C_{\mathcal{P}}^{+} \le \Delta \le \mathcal{A} - C_{\mathcal{P}}^{-}, \qquad 0 \le C_{\mathcal{P}}^{-} \le C_{\mathcal{P}}^{+} \le C_{\mathrm{loc}}. \tag{24}$$

Nested refinement makes $C_{\mathcal{P}}^{+}$ nonincreasing and $C_{\mathcal{P}}^{-}$ nondecreasing. Both converge to the exact expected integrated curvature cost as the mesh tends to zero. If $\mathbb{E}\sum_i \omega_i |q_i|^3 < \infty$, their gap additionally satisfies

$$C_{\mathcal{P}}^{+} - C_{\mathcal{P}}^{-} \le \frac{\mathrm{mesh}(\mathcal{P})}{12\sqrt{3}} \mathbb{E}\sum_i \omega_i |q_i|^3. \tag{25}$$

Proof. Bound h above and below on each subinterval and integrate with its exact nonnegative weight $1-t$. Refinement tightens both bounds because maxima decrease and minima increase within each parent interval. Continuity gives pointwise convergence of the weighted upper and lower sums. They are bounded by $q_i^2/8$; dominated convergence therefore applies under the finite-Q assumption. Finally, $|h'(x)| \le 1/(6\sqrt{3})$: writing $p = \sigma(x)$, maximize $p(1-p)|1-2p|$ over $p \in [0,1]$. The oscillation on one subinterval is at most

$$\frac{|q_i|(t_j - t_{j-1})}{6\sqrt{3}}.$$

Summing the oscillation bounds using $\sum_j w_j = 1/2$ gives Eq. (25). □

Relation to the statistical guarantee. Partition refinement improves the deterministic curvature calculation while retaining the same residuals and logit increments. Its limit recovers the exact paired loss contrast. Population confidence still comes from paired-loss concentration under calibration data independent of training and model selection; reused validation observations support only the empirical calculation.

F Extended Related Work and Comparison with Prior Theory

This appendix extends Section 2 and the theoretical discussion in Section 3.6. We compare Theorems 3.1 and 3.2 with the guarantees of GHG and SC, the structural guarantee of coherent HMC, and the computational object of GloEC. These results address different questions: graph-distribution generalization in GHG, collaboration among learners in SC, hierarchical output coherence in coherent HMC, and the effect of a realized prototype update in SoCoBio. Because their objects and assumptions differ, the corresponding bounds are not directly numerically comparable.

F.1 Graph-Distribution Generalization in GHG

GHG considers an agent a whose overall graph distribution is a mixture $D_a^{\mathrm{all}} = \omega D_a + \omega^{\mathrm{syn}} D_a^{\mathrm{syn}} + \omega^{\mathrm{kl}} D_a^{\mathrm{kl}}$. In the notation of the original paper, its Theorem 1 bounds the target risk as

$$\begin{aligned}\epsilon_{D_A}(F_a) \le{}& \epsilon_{D_a^{\mathrm{all}}}(F_a) + \omega C(D_a, D_A) \\ &+ \omega^{\mathrm{syn}} \epsilon_{D_A}(Y_a^{\mathrm{syn}}) + \omega^{\mathrm{kl}} \epsilon_{D_A}(Y_a^{\mathrm{kl}}) \\ &+ \frac{1-\omega}{2} d_{H_a \Delta H_a}(\psi \circ D_a^{\mathrm{syn}}, \psi \circ D_A).\end{aligned} \tag{26}$$

Following the formal definition in GHG's Appendix B.1,

$$\begin{aligned}C(D_a, D_A) &= \frac{1}{2} d_{F_a \Delta F_a}(D_a, D_A) + \eta(D_a), \\ \eta(D_a) &= \min_{F \in F_a} \left\{ \epsilon_{D_a}(F) + \epsilon_{D_A}(F) \right\}.\end{aligned} \tag{27}$$

Here $\eta(D_a)$ is the minimum joint risk on the two distributions, and the final term of Eq. (26) is an $H\Delta H$ discrepancy between graph distributions (Li et al., 2025, Theorem 1, Eq. (13)). GHG therefore asks whether a learned graph mixture can control risk on a target graph distribution.

SoCoBio instead fixes one predictor and compares four controller passes on the same biomolecular inputs. Equation (9) lower-bounds the paired weighted multi-label BCE contrast through the residual correction $\widehat{\mathcal{A}}_A$, the finite-step curvature cost $\widehat{\mathcal{C}}_A$, and a concentration allowance controlled by the realized prototype displacement B_A. Thus, the SoCoBio result concerns the realized joint pass J relative to 0, V, or H, whereas Eq. (26) concerns transfer across graph distributions.

F.2 Dynamically Weighted Collaboration in SC

SC analyzes a collection of M collaborating learners with random collaboration weights in a binary hypothesis-space setting. Its Theorem 4.1 states that, with probability at least $1-\delta$,

$$GError(f) \le \sum_{m=1}^{M} \mathbb{E}[w_m]\, \widehat{\mathrm{err}}(f^m) + \sum_{m=1}^{M} \mathbb{E}[w_m]\, \mathcal{R}_m(f^m) + \sum_{m=1}^{M} \mathrm{Cov}(w_m, \ell_m) + M\sqrt{\frac{\ln(1/\delta)}{2N}}. \tag{28}$$

The four terms are weighted empirical loss, weighted Rademacher complexity, weight–loss covariance, and a concentration term, respectively (Yao et al., 2025, Theorem 4.1, Eq. (16)). SC contrasts static collaboration weights, whose covariance term vanishes, with dynamic weighting rules intended to make this covariance non-positive.

SC is the closest precedent at the level of collaboration, but it analyzes distinct learners and their dynamic weights, whereas SoCoBio analyzes class prototypes and fixed-pass contrasts within one shared predictor. Equation (9) contains no learner-weight covariance; instead, $r_{ki}^{A} q_{ki}^{A}$ measures whether the realized logit change corrects the comparator residual on sample k. Likewise, SC's complexity term has no direct analogue in the directional result, whose sampling allowance

$$\epsilon_i = \sqrt{\frac{2\log(6K/\delta)}{n_i}}$$

depends on the positive support of class i.

F.3 Structural Guarantees in Coherent HMC

Coherent HMC enforces a score order along each parent–child edge: for a child c and parent p, coherence requires $s_c(x) \leq s_p(x)$ (Giunchiglia & Lukasiewicz, 2020). With a common threshold, this condition implies ancestor closure of every thresholded prediction.

SoCoBio's hierarchy-consistency objective has a related motivation but does not impose the same hard score-order guarantee. Theorems 3.1 and 3.2 address different quantities: Eq. (9) concerns the BCE effect of a realized joint prototype update, while Eq. (10) concerns its movement toward the positive-class feature direction. Hierarchical coherence and these two update gains are therefore separate properties, summarized in Table 22.

Table 22: Comparison of the theoretical objects and guarantees.

Result	Quantity certified	What the result does *not* certify
GHG, Eq. (26)	Target-graph risk through mixture risk and graph discrepancy	Paired BCE change for a frozen prototype-controller update
SC, Eq. (28)	Generalization of dynamically weighted learners	Positive-class prototype direction or hierarchical closure
Coherent HMC	Parent–child score order and threshold closure	Lower BCE or improvement relative to a controller pass
SoCoBio, Eqs. (9)–(10)	Paired BCE and positive-class direction for J versus A	Graph-domain adaptation or a universal coherence guarantee

F.4 The Computational Object in GloEC

GloEC learns label-related projection, graph-convolution, and classifier parameters through backpropagation (Huang et al., 2024). For each input protein, its hierarchy encoder computes label-node states $T(x) = M_l S(x) M_c$ from sequence features $S(x)$ and then propagates these states through three GCN layers. The matrices M_l and M_c, the graph transformations, and the classifier are persistent trainable parameters, whereas $T(x)$ and its propagated node states are input-dependent activations recomputed for each protein.

The EC taxonomy supplies the parent–child relations. The paper describes top-down and bottom-up adjacency matrices with self-loops, direction-specific graph convolutions, and nonlinear residual fusion. The inspected public implementation uses the same hierarchy to propagate directional and self-loop messages before nonlinear residual updates. In either case, the taxonomy specifies the communication structure, while the learned transformations determine the resulting node representations.

SoCoBio instead maintains a persistent prototype matrix P whose class prototypes are optimized directly. Within each forward computation, the Vertical Controller applies gated parent–child calibration, and the Horizontal Controller performs learned same-level peer selection followed by gated aggregation. These controller-transformed prototypes are shared across input sequences and enter the mixed linear–prototype readout, while gradients through the controllers update both P and the

controller parameters. Thus, SoCoBio differs from GloEC in both the computational object being propagated—persistent class prototypes rather than input-dependent label-node states—and the way hierarchical and peer interactions are selected and incorporated.

In a linear special case, prototype mixing and score mixing can coincide. For compatible matrices, a fixed mixing operator G, prototype matrix P, and feature vector z satisfy

$$(GP)z = G(Pz). \tag{29}$$

Mixing prototypes before a linear readout is then equivalent to mixing the corresponding scores. This identity covers only the specified linear computation. The complete architectures additionally differ in their nonlinear transformations, gates, normalization, interaction structure, input dependence, and readouts; parameterization alone therefore establishes neither full equivalence nor strict expressive superiority. Our empirical comparisons assess the complete predictors and the measured consequences of their learned updates.