SwissProt-Evidence & Pannot-FG: Fine-Grained Evidence-Grounded Protein Function Annotation with Homology Shift

Anonymous ACL submission

Abstract

Protein large language models can annotate protein function, yet instruction tuning can shift reliance from localized biochemical evidence to global sequence correlations, reducing robustness under homology shift. To push the protein understanding to a broader perspective, we assemble SwissProt-Evidence, a benchmark pairing protein-level function targets with residue-indexed site records and supporting text. It includes both protein-level tasks and residue-level fine-grained annotations, with C90/C30 splits enabling ID and OOD protein homology shift evaluation. To enable evidence-grounded protein understanding, we propose Pannot-FG, a training framework that jointly learns from functional labels and residue-level annotations. Pannot-FG employs cross-level objectives that explicitly align global predictions with localized biochemical evidence. Across both in-distribution and out-of-distribution settings, Pannot-FG improves site typing, evidence localization, and cross-level consistency. The results highlight the importance of integrating fine-grained supervision for robust and interpretable protein reasoning.

1 Introduction

Proteins execute biological function through *precise, localized biochemical interactions*, where a small number of residues—such as catalytic sites, ligand-binding pockets, or regulatory motifs—act as the primary determinants of activity (Blum et al., 2025; Sigrist et al., 2026). Accurately identifying these causal elements is critical not only for function prediction, but for mechanistic understanding, enzyme engineering, and therapeutic development. However, despite the rapid growth of protein sequence databases enabled by high-throughput sequencing, most proteins remain annotated at a coarse, label-only level, lacking explicit links between sequence and biochemical mechanism.

Protein Language Models (PLMs) (Brandes et al., 2022; Elnaggar et al., 2021), have demonstrated strong performance in sequence-to-function prediction by leveraging large-scale pretraining on evolutionary data. By modeling amino acid sequences as a biological language, these models capture rich statistical dependencies and structural priors. However, their success largely stems from exploiting *sequence conservations*, rather than identifying the *localized causal features* that govern function. As a result, PLMs can achieve high predictive accuracy while relying on evolutionary shortcuts, without recovering the underlying biochemical evidence.

This limitation becomes particularly evident in tasks requiring reasoning or explanation, such as protein question answering and functional captioning. In these settings, models often produce *hallucinated rationales* (Yang et al., 2025), where predicted functions are accompanied by explanations that are inconsistent with known catalytic residues or reaction mechanisms. This exposes a fundamental gap: while protein function is inherently fine-grained and causal, existing learning paradigms remain predominantly global and correlational.

To bridge this gap, we advocate for *evidence-grounded protein reasoning*, where models are required not only to predict function, but also to identify the responsible residues or regions and provide mechanistically consistent explanations. Achieving this requires both fine-grained supervision and training strategies that explicitly discourage shortcut learning.

In this work, we first introduce SwissProt-Evidence, a Swiss-Prot–derived benchmark that aligns functional annotations with multi-granular biochemical evidence, including residue-, span-, and region-level features. Building on this dataset, we propose two lightweight yet effective training objectives—*local contrast* and *counterfactual masking*—that explicitly disrupt global correlations

and encourage models to learn causal dependencies between localized sequence features and protein function.

Furthermore, we establish a principled generalization protocol based on hierarchical sequence clustering (C90/C30), which disentangles in-family (ID) and cross-family (OOD) evaluation. This enables systematic assessment of robustness under evolutionary shift and provides a controlled testbed for evaluating evidence grounding beyond label prediction.

Together, our framework moves beyond correlation-based function prediction toward *mechanistically faithful protein understanding*, where models are evaluated not only by what they predict, but by whether they can justify their predictions with biologically consistent evidence. In this paper, we make the following contributions:

- **Principled generalization protocol.** We propose a hierarchical data split suite that disentangles in-family (ID) and cross-family (OOD) generalization via two-level sequence clustering (C90/C30), providing a controlled and reproducible testbed for robustness under evolutionary shift.
- **Fine-grained benchmark.** We introduce SWISSPROT-EVIDENCE, a Swiss-Prot–derived benchmark that aligns functional labels with multi-granular biochemical evidence (residue-, span-, and region-level), enabling systematic evaluation of evidence grounding rather than label-only prediction.
- **Lightweight fine-grained supervision.** We develop two simple yet effective objectives—*local contrast* and *counterfactual masking*—that explicitly couple predictions to localized biochemical evidence, improving localization accuracy and evidence faithfulness without architectural changes.

2 Related Work

2.1 Protein Language Models (PLMs)

Protein language models (PLMs) learn general-purpose representations of protein sequences through large-scale self-supervised training. They have become a standard backbone for diverse downstream protein tasks, enabling efficient transfer while preserving general biochemical and evolutionary priors. Representative PLM families include encoder-only models like ESM-2 (Lin et al., 2022) , ProtBERT (Brandes et al., 2022) for generation task, decoder-only models like ProGen (Madani et al., 2023), ProtGPT2 (Ferruz et al., 2022) for prediction tasks and encoder-decoder models like ProstT5 (Heinzinger et al., 2024), Ankh (Elnaggar et al., 2023) for translating tasks.

2.2 Fine-Grained Protein Understanding

Unlike global function understanding (e.g., Gene Ontology annotation (Consortium, 2023) and CAFA-style (Friedberg et al., 2025) benchmarks), fine-grained protein understanding focuses on the mechanistic and site-specific “why" behind model outputs. There are a few datasets and benchmark (e.g., VenusX (Tan et al., 2025), InterPro (Blum et al., 2025), PROSITE (Sigrist et al., 2026)) contributing to research on such granularity. Recently, this area is heavily supported by advances in mechanistic interpretability, such as InterPLM (Simon and Zou, 2025), which utilizes sparse autoencoders to decompose dense PLM embeddings into thousands of human-interpretable features representing specific binding sites and structural motifs. Other works like ProtSAE (Liu et al., 2026) and MotifAE (Liu et al., 2025) focus on disentangling these latent spaces to isolate functional sequence patterns. Based on the same attention to fine-grained understanding, Our work explores the model’s ability to identify the important sites, motifs and domains by curated datasets and local-context strategy while training cross modalities.

2.3 Multi-modal LMs and Science reasoning

There still exists challenges for LMs to understand cross modalities in diverse fields. For protein understanding, it has increasingly moved toward multimodal architectures that align protein sequences with other scientific modalities, like 3D structure or natural language. This shift mirros the evolution of Vision-Language Modles (VLMs) (e.g. CLIP (Radford et al., 2021), BLIP (Li et al., 2022), BLIP2 (Li et al., 2023), Flamingo (Alayrac et al., 2022)). Our work also draws on a similar architecture with BLIP2, yielding pre-training model for future reasoning exploration. In parallel, a number of multimodal frameworks designed to support scientific reasoning have emerged in broader domains, including MoleculeSTM (Liu et al., 2024a) and MoICA (Liu et al., 2024b) for molecular-text alignment, as well as GenMS (Yang et al., 2024), MatterChat (Tang et al., 2025) for materials science.

2.4 Dataset/Benchmarks

A growing number of datasets and benchmarks have been proposed to enhance and evaluate the protein understanding capabilities of PLMs and LMs. A prominent line of work focuses on constructing large-scale instruction-tuning datasets to adapt general-purpose LLMs to protein-related tasks (Fang et al., 2024). In parallel, several multimodal protein–language modeling efforts introduce curated datasets to support model training and evaluation; for example, ProtLLM (Zhuo et al., 2024) constructs InterPT, a large-scale interleaved protein–text corpus for pre-training on mixed protein–text sequences. Beyond dataset construction, a variety of benchmarks have been proposed to assess protein reasoning across task-specific challenges. For instance, BioMaze (Zhao et al., 2025) evaluates LLMs' ability in biological pathway reasoning. However, most existing benchmarks primarily assess protein-level functional understanding, leaving fine-grained residue-level evaluation largely underexplored. VenusX first systematically unified diverse residue-level functional annotations under a standardized evaluation protocol (Tan et al., 2025). Building on these efforts, our model further emphasizes fine-grained, evidence-grounded reasoning under challenging generalization settings.

3 Benchmark: SwissProt-Evidence

To support fine-grained, evidence-grounded supervision, we construct SwissProt-Evidence from UniProtKB/Swiss-Prot (UniProt Consortium, 2025), a manually curated subset of UniProtKB comprising 573,661 reviewed protein entries. Swiss-Prot provides high-quality annotations spanning multiple biological semantics and levels of granularity, including both global protein-level descriptions and residue-level sequence annotations aligned to specific amino acid positions.

While UniProtKB contains a broad spectrum of annotation types, not all are equally suitable for learning sequence-grounded functional representations. We therefore select a subset of annotations guided by the following criteria to ensure that supervision signals are biologically meaningful, structurally coherent, and amenable to learning:

- **Biological relevance and complementarity.** We retain annotation types that capture essential and complementary aspects of protein function, forming a structured and hierarchical representation of protein biology.
- **Sequence grounding.** We prioritize annotations that admit a direct or learnable mapping to the underlying amino acid sequence, excluding those primarily determined by external experimental conditions or contextual factors.
- **Controlled expressiveness.** We select annotations with relatively standardized and constrained textual descriptions, facilitating consistent and interpretable supervision.
- **Task separability.** We avoid overly broad or abstract annotations that may introduce information leakage across tasks, thereby reducing the risk of shortcut learning.

To convert heterogeneous structured annotations into unified supervision signals, we develop a template-based natural language generation pipeline (Figure 1). For each selected annotation type, we design a set of seed templates that specify how structured fields are mapped into natural language descriptions. These templates are subsequently expanded into diverse paraphrased variants through a human–LLM collaborative process, improving linguistic diversity while preserving semantic fidelity.

During dataset construction, annotation fields are normalized into template-compatible representations, and one template is randomly sampled to instantiate each example. This procedure yields diverse yet semantically consistent instruction-style samples grounded in curated biological evidence.

Task	Evidence Type	Eval Level
GO	Functional labels	Protein
EC	Enzymatic function	Protein
Catalytic Activity	Reaction description	Protein
Cofactor	Ligand / cofactor usage	Protein
Subcellular Loc	Cellular compartment	Protein
Pathway	Biological pathway	Protein
Regions	Structural / sequence spans	Residue
Sites	Catalytic / binding residues	Residue

Table 1: Overview of protein annotation tasks and their evaluation granularity in SwissProt-Evidence. Protein-level tasks assess global functional understanding, while residue-level tasks evaluate localization and evidence grounding.

We defines a unified benchmark spanning both global functional prediction and fine-grained evidence grounding across protein families. As summarized in Table 1, we organize the benchmark into two evaluation levels. The protein-level tasks test whether a model can infer a protein's overall biochemical role and cellular context from its se-

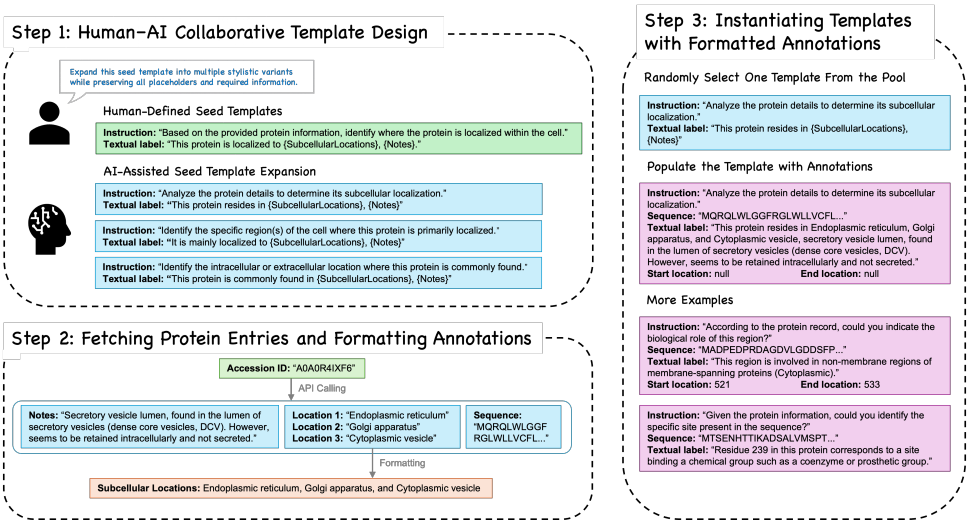


Figure 1: Overview of the dataset construction pipeline.

quence and associated evidence. On the other hand, the residue-level tasks further require localization ability. Together, these two levels explicitly separate what a protein does from where in the sequence the evidence lies, enabling systematic evaluation of evidence-grounded reasoning.

Each sample is represented using a unified schema:

- **Instruction**: a natural-language prompt describing the biological task.
- **Sequence**: the full-length amino acid sequence of the protein.
- **Textual label**: a grounded natural-language description derived from curated annotations.
- **Location**: residue indices specifying the sequence region or site associated with the annotation (when applicable).

4 Methods

In this section, we introduce the model formulation and our methods including fine-grained pretraining and evidence grounded post-training.

4.1 Formulation and Base Model

We consider protein–text alignment and reasoning. Each pre-training instance consists of a protein sequence x and a natural-language description y (e.g., UniProt functional text). For a subset of examples, we additionally observe residue-level evidence annotations as an index set $R \subseteq \{1, \ldots, L\}$, indicating functionally relevant residues (e.g., active sites, binding residues). While for post-training, we provide the set of ground truth annotations gt corresponding to the protein x.

Protein encoder. A pretrained protein language model (PLM) f_P maps x to contextual residue embeddings

$$H = f_P(x) \in \mathbb{R}^{L \times d_P}. \tag{1}$$

Q-Former. We employ a BLIP2-style Q-Former module to interface protein embeddings with text and generation objectives. Concretely, the Q-Former provides (i) a set of soft query outputs Z obtained by attending to H,

$$Z = F_{Q,\text{prot}}(Q, H) \in \mathbb{R}^{N_q \times d_Q}. \tag{2}$$

and (ii) a text representation h_T for the paired description y (via its internal text-encoding branch).

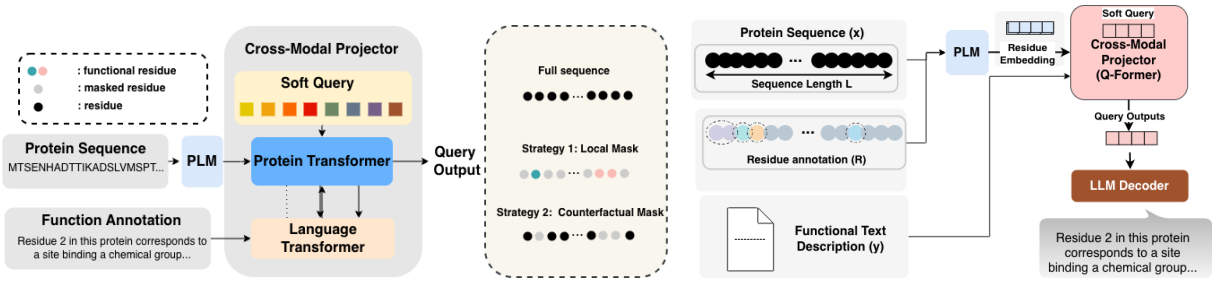
$$h_T = F_{Q,\text{text}}(y)_{\texttt{[CLS]}} \in \mathbb{R}^{d_T}. \tag{3}$$

We follow standard formulations for protein–text contrast, matching, and conditional generation and provide exact definitions in Appendix C. The pretraining part (stage 1) is mainly focusing on Q-Former training.

LLM decoder. The LLM decoder after Q-Former module would perceive the soft query embeddings Z and prompt to generate natural language output.

4.2 Fine-grained Pretraining

Our set of fine-grained objectives that encourage protein–text alignment to be *evidence-grounded* at the residue level.



(a) Cross-Modal Projector Module. (b) Global Architecture for training.

Figure 2: Overview of the proposed Pannot-FG framework.

Baseline global objectives. We build on standard BLIP2 Stage–1 objectives for multimodal alignment, consisting of protein–text contrastive (PTC), protein–text matching (PTM), and conditional language modeling (LM):

$$\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{PTC}} + \lambda_{\text{PTM}}\,\mathcal{L}_{\text{PTM}} + \lambda_{\text{LM}}\,\mathcal{L}_{\text{LM}}. \quad (4)$$

We use the max-query similarity for PTC as in Appendix C.

Local contrast (Strategy 1: evidence sufficiency). Our first strategy enforces that the model can align protein and text *using only evidence residues*. We compute auxiliary local objectives by replacing H with the evidence-isolated representation $H_R = \mathcal{P}(H; R)$ inside the same matching and generation pathways:

$$, \quad (5)$$

where we omit repeated module arguments for readability (full instantiation in Appendix C). Intuitively, $\mathcal{L}_{\text{loc}}$ encourages *sufficiency*: the evidence residues alone should support the paired description.

Counterfactual masking (Strategy 2: evidence necessity). Our second strategy enforces that evidence residues are *necessary* for alignment. We construct a counterfactual protein representation $\widetilde{H} = \mathcal{M}(H; R)$ and penalize cases where removing evidence does not reduce protein–text compatibility. Concretely, letting $s(\cdot,\cdot)$ denote the protein–text similarity used by PTC, we apply a margin-based ranking loss

$$\mathcal{L}_{\text{cf}} = \max\Big(0,\, m + s(\widetilde{H}, y) - s(H, y)\Big), \quad (6)$$

averaged over the batch. This objective encourages *necessity*: masking evidence residues should measurably degrade compatibility with the paired text.

Overall fine-grained pretraining objective. Our final pretraining objective adds the proposed fine-grained terms to the baseline:

$$\boxed{\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{base}} + \lambda_{\text{loc}}\,\mathcal{L}_{\text{loc}} + \lambda_{\text{cf}}\,\mathcal{L}_{\text{cf}}.} \quad (7)$$

When residue evidence is unavailable, we drop $\mathcal{L}_{\text{loc}}$ and $\mathcal{L}_{\text{cf}}$ for that example and optimize only $\mathcal{L}_{\text{base}}$. This builds the foundation for multi-granular pretraining

4.3 Evidence preference post-training

After fine-grained pretraining, we would align the soft embeddings from the q-former with the LLM decoder. The SFT training with LLM decoder is the basic alignment tool. The SFT training samples includes instructions and the location information are textualized into text labels following our templates. We further align generation behavior with residue-level evidence using preference optimization. For each input, we construct pairs of candidate responses that differ in evidence consistency (e.g., grounded vs. ungrounded with respect to R) and optimize a GRPO-style objective that increases the likelihood of evidence-consistent outputs. If the GRPO training set has This post-training stage improves faithfulness to residue evidence without requiring explicit chain-of-thought supervision. Full details of preference construction and optimization are given in Appendix F.

4.4 Evaluation Protocol

4.4.1 Residue localization metrics

Evaluation Metrics. To precisely quantify intra-sequence localization, we formulate a Multiple-Choice (MC) ranking task. For each protein-text pair, we construct a set of N candidate masks consisting of one ground-truth functional mask and $N-1$ decoy masks. Decoys are generated by randomly sampling amino acid positions within the

same sequence, constrained to the same cardinality as the ground truth. Following the scoring protocol in our implementation, the compatibility score s for a mask k is defined as:

$$s_k = \max_q \langle \text{proj}(\mathbf{Z}_{q,k}), \text{proj}(\mathbf{h_T}) \rangle \quad (8)$$

where $\mathbf{z}_{q,k}$ represents the q-th query output from the Q-former conditioned on mask k, and $\mathbf{t}$ denotes the text features. We report:

- **MC_Acc**: Top-1 accuracy, indicating how often the ground-truth mask achieves the highest compatibility score.
- **MC_Mrr**: Mean Reciprocal Rank, calculating the average of the reciprocal ranks of the true functional mask among the N candidates.
- **MC_Rec5**: Recall@5, the proportion of samples where the ground-truth mask is ranked within the top 5.

4.4.2 Faithfulness (sufficiency/necessity)

We evaluate whether model predictions are grounded in localized biochemical evidence using masking-based necessity and sufficiency tests. Based on our previous notation of H, R, H_R and $\widetilde{H}$, necessity measures whether the evidence is required for the prediction: we compare the model's confidence for the correct label when conditioned on H versus $\widetilde{H}$, where a substantial degradation indicates that the masked regions are essential for the decision. In contrast, sufficiency measures whether the evidence alone is adequate to support the prediction: we evaluate whether the model preserves its prediction when conditioned only on H_R.

4.4.3 Dialogue mechanistic evaluation

For mechanistic evaluation, we analyze systematic generation behaviors under a fixed prompt, align free-form predictions to curated ground truth by mapping functional, molecular, and spatial evidence expressed in text to their corresponding annotated entities, and apply task-specific, rule-based criteria to assess whether predictions are mechanistically consistent with the ground truth rather than relying on surface-level string matching.

5 Experiments

5.1 Setup

We evaluate fine-grained protein function annotation under a BLIP2-style protein–text architecture. Our experiments are designed to (i) assess the benefit of SwissProt-Evidence, which provides multi-granular biochemical evidence and supports both ID and OOD-family evaluation, and (ii) quantify the effectiveness of our **fine-grained pretraining objectives**. We used A100 GPUs on computing clusters for the experiments. More details about the experiment are attached in Appendix C.

5.2 Pilot Study: Validating Fine-Grained Objectives

Prior to large-scale pre-training on our proprietary datasets, we conduct a pilot study on the **VenusX** benchmark (Tan et al., 2025) to validate the efficacy of our proposed objective functions. This preliminary investigation serves to isolate the performance gains attributable to the *Local Contrastive* (LC) and *Counterfactual* (CF) strategies in a controlled setting, ensuring the architectural choices are optimized for evidence-grounded reasoning before scaling.

	Strategy	Acc ↑	Mrr ↑	Rec5 ↑
X	BL	41.48	0.544	68.41
	BL + LC	42.16	0.546	65.00
	BL + CF	43.30	0.558	67.05
	BL+LC+CF	**44.09**	**0.588**	**76.25**
MF50	BL	46.74	0.620	80.98
	BL + LC	52.72	0.676	90.22
	BL + CF	52.17	0.658	85.87
	BL+LC+CF	**58.70**	**0.722**	**91.85**
MP50	BL	49.16	0.633	82.87
	BL + LC	46.91	0.629	84.27
	BL + CF	56.18	0.696	89.04
	BL+LC+CF	**59.83**	**0.723**	**91.29**
MP90	BL	47.89	0.622	81.26
	BL + LC	45.90	0.613	84.37
	BL + CF	55.88	0.676	83.48
	BL+LC+CF	**56.98**	**0.691**	**84.59**

Table 2: Ablation results of pre-training strategies across four VenusX_Res_Act benchmarks. We evaluate the impact of Local Contrastive (LC) and Counterfactual (CF) strategies compared to baseline (BL). The metrics (Acc, Mrr, and Rec5) are for localization multiple choice as defined in sec 4.4.1

Quantitative Analysis. As summarized in Table 2, the integration of both strategies in the **Full (LC+CF)** framework consistently yields superior performance across all benchmarks. In the cross-family (`Res_Act_X`) split, our full model achieves a **7.84%** improvement in `Loc_mc_Rec5` over the baseline, demonstrating robust generalization to novel protein families.

The synergy between LC and CF becomes particularly evident in the mixed-family splits. In the

high-similarity MP90 setting, the **Full** model outperforms the baseline by **9.09%** in Loc_mc_Acc. These results indicate that the **LC** objective enhances localized precision, while the **CF** objective effectively mitigates rote sequence memorization. These findings provide a rigorous empirical foundation for our subsequent large-scale pre-training phase.

Faithfulness The ablation results in Table 2 provide evidence for the faithfulness of our pretraining strategies. While local contrastive learning (LC) improves discriminability of annotated regions, its effect is inconsistent across splits, particularly under more challenging distribution shifts. In contrast, counterfactual masking (CF) yields consistent improvements, indicating that explicitly enforcing dependence on annotated regions is critical for reliable localization. The full strategy combining LC and CF achieves the strongest performance across all benchmarks, suggesting that faithful reasoning requires both recognizing relevant evidence and being causally dependent on it. Notably, the largest gains are observed on MF and MP splits, where global sequence correlations are weaker, further supporting the role of evidence grounding in robust generalization.

5.3 Large Scale Cross-Modal Pretraining

To extend fine-grained training to a large-scale setting, we apply SwissProt-Evidence to cross-modal pretraining using a Q-Former–based Stage-1 framework that aligns protein representations with textual annotations. We adopt a curriculum that first emphasizes global protein–text pairs to establish coarse functional semantics, followed by fine-grained annotations to refine localized and attribute-level alignment. All annotation types are unified under a single pretraining objective, with coefficient values selected based on the ablation study in Table 7.

Task	Split	Acc ↑	Mrr ↑	Rec5 ↑
FG Stage1	ID	31.72	0.5235	84.76
	OOD	30.91	0.5092	83.48
Baseline	mix	8.63	0.285	51.02

Table 3: Multi-choice localization performance of Stage-1 models compared with baseline models. Here we use sites tasks for the evaluation.

We first evaluate the effect of Stage-1 pretraining on fine-grained localization using the multi-choice localization (Loc_mc) task. As shown in Table 3, fine-grained pretraining leads to a substantial improvement in localization sensitivity compared to the baseline model. With eight candidate locations per query, the baseline performs close to random guessing, whereas the Stage-1 model achieves approximately 30% accuracy, indicating that cross-modal pretraining successfully induces non-trivial site-level grounding between textual descriptions and protein regions.

Beyond localization, Stage-1 pretraining consistently improves cross-modal retrieval performance. As shown in Table 4, in-batch Recall@20 exceeds 90% across most tasks and approaches saturation for enzyme-centric annotations such as EC numbers. While in-batch top-1 accuracy remains moderate, this behavior indicates that fine-grained pretraining effectively organizes protein–text representations into semantically coherent neighborhoods, even when precise ranking among highly similar candidates remains challenging.

Full-set retrieval remains challenging, with low absolute accuracy but non-trivial Recall@20, reflecting the scale and ambiguity of the candidate space. Notably, EC number prediction exhibits the strongest performance, consistent with conserved catalytic motifs and standardized functional descriptions, whereas attributes such as cofactors show lower accuracy due to their heterogeneous and weakly localized nature. Together, these results indicate that Stage-1 cross-modal pretraining effectively structures the global embedding space but is insufficient on its own for precise decision-making or evidence-faithful prediction, motivating subsequent post-training stages.

5.4 Training with LLM Decoder

In Stage-2, we further fine-tune the pretrained model using task-specific instruction–response pairs derived from SwissProt-Evidence. The training aims to align the Q-former's soft embeddings with LLM decoder. It substantially improves fluency and task recognition across all annotation types: the model reliably adopts task-appropriate language (e.g., enzymatic descriptions for EC and catalytic tasks, biological process narratives for GO). Mechanistic evaluation reveals that the model predominantly produces abstract functional narratives.

We analyze four Stage-2 configurations that differ in decoder adaptation and Stage-1 initializa-

Task	Baseline Stage-1				Fine-grained Stage-1 (ID)				Fine-grained Stage-1 (OOD)			
	InAcc	InR20	FullAcc	FullR20	InAcc	InR20	FullAcc	FullR20	InAcc	InR20	FullAcc	FullR20
Cofactor	6.09	52.03	0.27	2.54	34.74	94.37	1.70	23.64	33.43	94.21	1.70	22.85
EC	12.63	59.40	0.29	3.90	69.70	99.57	3.30	45.57	68.36	99.57	3.27	45.01
GO	28.91	68.92	3.17	16.81	68.43	99.36	7.88	48.00	68.24	99.27	7.88	48.36
CA	24.89	76.32	1.32	12.95	58.75	98.89	4.32	47.82	58.12	98.35	4.42	47.45
Sites	1.97	31.65	0.07	0.42	12.32	56.87	0.85	6.41	12.73	56.12	0.85	6.41

(a) Protein → Text retrieval (p2t)

Task	Baseline Stage-1				Fine-grained Stage-1 (ID)				Fine-grained Stage-1 (OOD)			
	InAcc	InR20	FullAcc	FullR20	InAcc	InR20	FullAcc	FullR20	InAcc	InR20	FullAcc	FullR20
Cofactor	4.73	47.21	0.23	1.62	32.98	94.59	1.55	19.58	32.24	93.25	1.55	19.22
EC	8.96	55.32	0.21	2.38	66.78	99.47	2.24	37.59	66.40	99.47	2.46	37.23
GO	19.36	64.43	1.00	7.58	66.27	98.93	5.38	43.89	66.42	98.79	5.41	43.26
CA	15.48	70.50	0.50	5.24	57.97	98.74	3.90	45.90	57.13	98.74	3.85	45.90
Sites	1.74	31.61	0.01	0.19	12.70	54.68	0.47	4.93	11.47	54.02	0.47	4.93

(b) Text → Protein retrieval (t2p)

Table 4: Stage-1 retrieval performance comparing vanilla Q-Former baseline vs SwissProt-Evidence. Top: protein-to-text (p2t). Bottom: text-to-protein (t2p). In-batch metrics measure global semantic alignment; full-set metrics test large-pool retrieval.

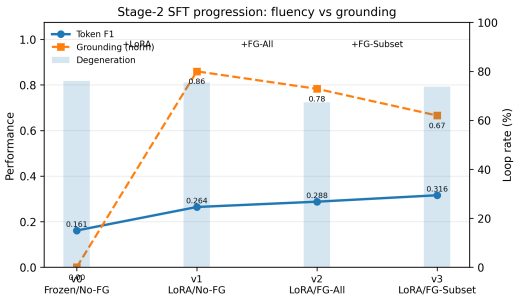


Figure 3: Decoder adaptation improves fluency, fine-grained pretraining improves robustness, but grounding remains the bottleneck.

tion. Comparing v0 (frozen LLM) and v1 (LoRA-enabled) reveals that enabling decoder adaptation yields the largest performance gain, with a substantial increase in token-level F1 across tasks. This indicates that Stage-2 SFT primarily addresses a decoder alignment bottleneck, allowing the model to produce task-appropriate language and structured outputs. Transitioning from v1 to v2, where the model is initialized from a fine-grained Stage-1 checkpoint trained on all tasks, further improves performance consistency and reduces degeneration, suggesting that Stage-1 enhances the semantic quality of Q-Former representations. However, restricting Stage-1 pretraining to a subset of tasks (v3) introduces representation bias, leading to uneven task performance and reduced stability.

Notably, across all configurations, improvements are concentrated in linguistic fidelity rather than mechanistic grounding. While token-level F1 increases substantially from v0 to v3, grounding metrics remain comparatively low and largely flat, indicating that the model learns to produce fluent, task-aligned descriptions without reliably identifying the underlying biochemical evidence. This highlights a fundamental limitation of Stage-2 SFT: it aligns how the model expresses knowledge.

6 Limitations and Future Work

- **KG semantics:** structured semantic validation via knowledge graphs is promisings.
- **Multimodality:** structural tokens (e.g., 3Di) may improve mechanistic reasoning, particularly for functions determined by spatially distal residues.
- **Annotation sparsity:** Swiss-Prot evidence is incomplete; a large portion of proteins are under-explored; methods should handle missing labels.

Impact/Ethics Statement

This paper presents work whose goal is to advance the field of language modeling through the study of simple algorithmic tasks inspired by creativity. There are many potential societal consequences of language modeling — particularly if such methods were applied to sensitive domains such as biology or biotechnology. As such, while misuse of AI in biological contexts is an important concern in general, we do not believe any specific biosecurity or biological misuse risks arise from this focused, abstract, algorithmic study. AI tools were used to assist with writing clarity and code debugging. All scientific content was developed and verified by the authors.

References

Shadab Ahmad, Leonardo Jose da Costa Gonzales, Emily H Bowler-Barnett, Daniel L Rice, Minjoon Kim, Supun Wijerathne, Aurélien Luciani, Swaathi Kandasaamy, Jie Luo, Xavier Watkins, and 1 others. 2025. The uniprot website api: facilitating programmatic access to protein knowledge. *Nucleic Acids Research*, page gkaf394.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. *Preprint*, arXiv:2204.14198.

Matthias Blum, Antonina Andreeva, Laise Cavalcanti Florentino, Sara Rocio Chuguransky, Tiago Grego, Emma Hobbs, Beatriz Lazaro Pinto, Ailsa Orr, Typhaine Paysan-Lafosse, Irina Ponamareva, Gustavo A. Salazar, Nicola Bordin, Peer Bork, Alan Bridge, Lucy Colwell, Julian Gough, Daniel H. Haft, Ivica Letunic, Felipe Llinares-López, and 15 others. 2025. Interpro: the protein sequence classification resource in 2025. *Nucleic Acids Research*, 53(D1):D444–D456.

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. 2022. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110.

The Gene Ontology Consortium. 2023. The gene ontology knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D38–D49.

Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. 2023. Ankh: Optimized protein language model unlocks general-purpose modelling. *Preprint*, arXiv:2301.06568.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. 2021. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *Preprint*, arXiv:2007.06225.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2024. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *Preprint*, arXiv:2306.08018.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. 2022. Protgpt2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348.

Iddo Friedberg, Predrag Radivojac, Paul D Thomas, An Phan, M. Clara De Paolis Kaluza, Damiano Piovesan, Parnal Joshi, Chris Mungall, Martyna Plomecka, Walter Reade, and María Cruz. 2025. Cafa 6 protein function prediction. https://kaggle.com/competitions/cafa-6-protein-function-prediction. Kaggle.

Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Milot Mirdita, Martin Steinegger, and Burkhard Rost. 2024. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6(4):lqae150.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, and 1 others. 2022. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*.

Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Anima Anandkumar. 2024a. Multi-modal molecule structure-text model for text-based retrieval and editing. *Preprint*, arXiv:2212.10789.

Xiangyu Liu, Haodi Lei, Yi Liu, Yang Liu, and Wei Hu. 2026. Protsae: Disentangling and interpreting protein language models via semantically-guided sparse autoencoders. *Preprint*, arXiv:2509.05309.

Zhaoyang Liu and 1 others. 2025. Motifae: Reveals functional motifs from protein language models via unsupervised discovery and interpretability analysis. *bioRxiv*. Preprint.

Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng Chua. 2024b. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *Preprint*, arXiv:2310.12798.

Ali Madani, Ben Krause, Ethan R. Greene, and 1 others. 2023. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41:1099–1106.

Andrew G McDonald, Sinéad Boyce, and Keith F Tipton. 2009. Explorenz: the primary source of the iubmb enzyme list. *Nucleic acids research*, 37(suppl_1):D593–D597.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Christian J. A. Sigrist, Bertrand A. Cuche, Emmanuel de Castro, Etienne Coudert, Nicolas Redaschi, and Alan Bridge. 2026. The prosite database for protein families, domains, and sites. *Nucleic Acids Research*, 54(D1):D451–D458.

Evan Simon and James Zou. 2025. Interplm: discovering interpretable features in protein language models via sparse autoencoders. *Nature Methods*, 22:2107–2117.

Yang Tan, Wenrui Gou, Bozitao Zhong, Liang Hong, Huiqun Yu, and Bingxin Zhou. 2025. Venusx: Unlocking fine-grained functional understanding of proteins. *Preprint*, arXiv:2505.11812.

Yingheng Tang, Wenbin Xu, Jie Cao, Weilu Gao, Steve Farrell, Benjamin Erichson, Michael W. Mahoney, Andy Nonaka, and Zhi Yao. 2025. Matterchat: A multi-modal llm for material science. *Preprint*, arXiv:2502.13107.

UniProt Consortium. 2025. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617.

Sherry Yang, Simon Batzner, Ruiqi Gao, Muratahan Aykol, Alexander L. Gaunt, Brendan McMorrow, Danilo J. Rezende, Dale Schuurmans, Igor Mordatch, and Ekin D. Cubuk. 2024. Generative hierarchical materials search. *Preprint*, arXiv:2409.06762.

Yining Yang, Ruihong Huang, and Yang Shen. 2025. Beyond labels: Explanatory collapse due to instruction tuning in protein LLMs. In *NeurIPS 2025 2nd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences*.

Haiteng Zhao, Chang Ma, Fangzhi Xu, Lingpeng Kong, and Zhi-Hong Deng. 2025. Biomaze: Benchmarking and enhancing large language models for biological pathway reasoning. *Preprint*, arXiv:2502.16660.

Le Zhuo, Zewen Chi, Minghao Xu, Heyan Huang, Heqi Zheng, Conghui He, Xian-Ling Mao, and Wentao Zhang. 2024. Protllm: An interleaved protein-language llm with protein-as-word pre-training. *arXiv preprint arXiv:2403.079205*.

A Appendix: Split Definitions

Our dataset is curated from UniProtKB, a comprehensive repository of protein sequences and annotations. UniProtKB is divided into two components: Swiss-Prot, which contains manually reviewed and curated protein entries, and TrEMBL, which contains automatically annotated entries. In this work, we use Swiss-Prot exclusively as the source of proteins for training(both pre-training and post-training) and evaluation. Swiss-Prot entries are manually curated by domain experts and are associated with higher-quality functional annotations, including experimentally supported evidence for catalytic activity, binding sites, motifs, and domain organization. In contrast, TrEMBL contains a much larger number of sequences, but many annotations are inferred computationally and may be noisy or inconsistent.

We choose Swiss-Prot for three main reasons. First, our objectives involve fine-grained and evidence-grounded reasoning, which requires reliable residue-level supervision. Second, automatically annotated databases often propagate systematic errors, which can confound both training and evaluation. Third, restricting to curated proteins simplifies interpretation and error analysis, allowing model behavior to be analyzed in terms of known biochemical mechanisms.

A.1 Principles for Dataset Splitting

Protein datasets violate standard i.i.d. assumptions in two distinct ways.

First, a single protein may generate multiple training examples (e.g., multiple prompts or residue-level labels). If such examples are split independently, information can leak across splits. Therefore, all splits must be defined *at the protein level*.

Second, protein sequences exhibit strong homology structure. Many proteins are nearly identical, while others share only remote evolutionary relationships. Naïve random splitting leads to near-duplicate leakage, resulting in artificially inflated performance.

We therefore adopt the following principles:

Principle 1 (Protein atomicity). All examples derived from the same protein accession must belong to the same split.

Principle 2 (Hierarchical homology control). We distinguish between two regimes of sequence

similarity: (i) near-duplicate similarity (approximately 90% sequence identity), and (ii) family-level similarity (approximately 30% sequence identity). These represent fundamentally different generalization challenges and must be controlled separately.

Principle 3 (Train once, evaluate many). A single training set is constructed, while multiple evaluation sets are defined with increasing levels of distribution shift.

A.2 Hierarchical Homology-Based Splitting

Let

$$\mathcal{P} = \{p_1, \ldots, p_N\}$$

be the set of curated Swiss-Prot proteins.

We leverage MMseqs2 to implement the following two clustering methods:

- $C_{30}(p)$ assigns protein p to a family-level cluster (proteins with approximately 30% identity in one cluster).
- $C_{90}(p)$ assigns protein p to a near-duplicate cluster (proteins with approximately 90% identity in one cluster).

The goal is to construct five disjoint protein ID sets: $\text{Train}, \text{Val}_{\text{ID}}, \text{Test}_{\text{ID}}, \text{Val}_{\text{OOD}}, \text{Test}_{\text{OOD}}$.

Step 1: Split family-level clusters. Let

$$\mathcal{F}_{30} = \{C_{30}(p) \mid p \in \mathcal{P}\}$$

be the set of unique family clusters. We randomly partition:

$$\mathcal{F}_{30} = \mathcal{F}_{30}^{\text{train}} \cup \mathcal{F}_{30}^{\text{val}} \cup \mathcal{F}_{30}^{\text{test}}.$$

Step 2: Assign OOD-family proteins. For each protein $p \in \mathcal{P}$:

$$\text{split}(p) = \begin{cases} \text{Val}_{\text{OOD}}, & C_{30}(p) \in \mathcal{F}_{30}^{\text{val}}, \\ \text{Test}_{\text{OOD}}, & C_{30}(p) \in \mathcal{F}_{30}^{\text{test}}, \\ \text{unassigned}, & C_{30}(p) \in \mathcal{F}_{30}^{\text{train}}. \end{cases}$$

Define:

$$\mathcal{P}_{\text{train-fam}} = \{p \in \mathcal{P} \mid C_{30}(p) \in \mathcal{F}_{30}^{\text{train}}\}.$$

Step 3: Split near-duplicate clusters inside training families. Let:

$$\mathcal{F}_{90} = \{C_{90}(p) \mid p \in \mathcal{P}_{\text{train-fam}}\}.$$

We randomly partition:

$$\mathcal{F}_{90} = \mathcal{F}_{90}^{\text{train}} \cup \mathcal{F}_{90}^{\text{val}} \cup \mathcal{F}_{90}^{\text{test}}.$$

Step 4: Assign ID and training proteins. For each protein $p \in \mathcal{P}_{\text{train-fam}}$:

$$\text{split}(p) = \begin{cases} \text{Val}_{\text{ID}}, & C_{90}(p) \in \mathcal{F}_{90}^{\text{val}}, \\ \text{Test}_{\text{ID}}, & C_{90}(p) \in \mathcal{F}_{90}^{\text{test}}, \\ \text{Train}, & C_{90}(p) \in \mathcal{F}_{90}^{\text{train}}. \end{cases}$$

This construction ensures: (i) each protein is assigned to exactly one split, (ii) no family-level cluster (C_{30}) is shared between the training split and the OOD validation or test splits, (iii) no near-duplicate cluster (C_{90}) spans the Train, Val_{ID}, and Test_{ID} splits, and (iv) and the resulting ID and OOD splits correspond to distinct generalization regimes—respectively, within-family generalization and cross-family generalization.

A.3 Condensed Training Set via Family-Balanced Sampling

The full training split contains several hundred thousand proteins, which is computationally expensive for large-scale pretraining. Moreover, the distribution of protein families is highly imbalanced, with a small number of families dominating the dataset.

To construct a more compact yet diverse training set, we apply *within-train sampling*, without modifying any evaluation splits.

Let:

$$\text{Train} = \bigcup_{f \in \mathcal{F}_{30}^{\text{train}}} \mathcal{P}_f,$$

where $\mathcal{P}_f$ denotes the set of training proteins in family f.

We define the condensed training set as:

$$\text{Train}_{\text{condensed}} = \bigcup_{f \in \mathcal{F}_{30}^{\text{train}}} \text{Sample}_k(\mathcal{P}_f),$$

where $\text{Sample}_k(\cdot)$ selects at most k proteins uniformly at random from each family.

Optionally, a global target size N can be enforced by further subsampling across families.

This sampling scheme preserves broad family-level diversity while preventing overrepresentation of large families. In contrast, random subsampling tends to concentrate on well-studied protein families and discard rare functional classes.

Importantly, sampling is applied *only within the training split*. All validation and test splits remain unchanged, and therefore all homology guarantees of the original split protocol are preserved.

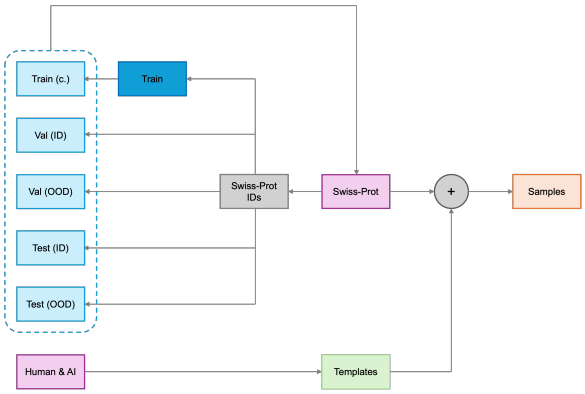


Figure 4: Schematic illustration of the end-to-end dataset construction pipeline, from accession-level splits to template-based sample instantiation.

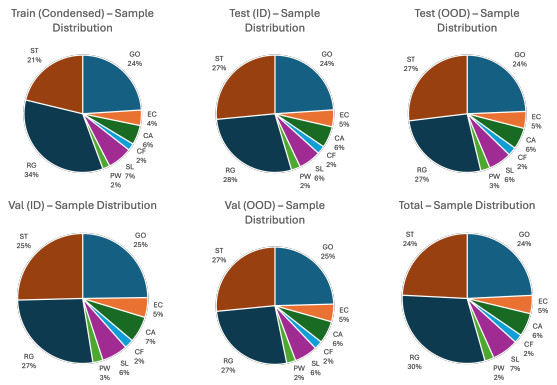


Figure 5: Distribution of supervision samples across tasks for each dataset split.

B Appendix: Dataset Construction

B.1 Task Overview and Biological Grounding

Protein understanding refers to the ability to infer biologically meaningful information encoded in protein molecules across multiple levels of abstraction. In addition to identifying the global functional label of a protein, it is also necessary to model the cofactors required for manifesting specific functions, the cellular locations in which these functions are performed, the higher-level biological processes the protein participates in, as well as the specific residue-level features that encode the corresponding function. Together, these aspects constitute a structured and hierarchical narrative of the inherent biological meaning of proteins.

Based on this formulation, we define eight protein-centric tasks in our dataset, which span *protein-* and *residue-level* annotations, as follows.

Protein-level tasks. Protein-level tasks provide global annotations defined over entire protein sequences and describe the overall biochemical function, cellular context, and functional semantics of proteins. We include the following six protein-level tasks:

- **Gene ontology**: Provides a general description of protein-related biological semantics using controlled ontology terms across molecular function (MF), biological process (BP), and cellular component (CC).
- **EC number**: Enzyme Commission (EC) number, which provides a standardized classification of enzymatic catalytic functions.
- **Catalytic activity**: Specifies the biochemical reactions catalyzed by the enzyme.
- **Cofactor**: Specifies the metal ions or small organic molecules required for enzymatic activity.
- **Subcellular location**: Describes the cellular compartments in which the protein is localized, capturing instance-level localization information, including membrane topology and orientation when available.
- **Pathway**: Describes the biological pathways in which the protein participates, capturing pathway hierarchy and step-level roles when available.

Residue-level tasks. Residue-level tasks provide localized functional annotations associated with specific sequence segments or individual amino acid residues. These tasks capture fine-grained functional evidence underlying global protein behavior and include:

- **Regions**: Annotates contiguous sequence segments that define structural, topological, or functional organization within the protein sequence.
- **Sites**: Annotates short, explicitly defined residue-level sites associated with specific functional roles, such as catalytic activity or molecular binding.

B.2 Dataset Statistics

B.3 UniProtKB Annotation Selection Rationale

For each protein entry, UniProtKB provides a JSON-formatted record whose data structure is shown below.

Note: Throughout this section, we refer exclusively to the JSON-formatted UniProtKB entry as the upstream data structure. Although UniProtKB provides multiple formats for programmatic access

Split	#Prot.	GO	GO-BP	GO-MF	GO-CC	EC
Train (c.)	127,619	291,071	94,992	98,829	97,250	51,731
Test (ID)	22,014	54,709	17,828	19,038	17,843	10,669
Test (OOD)	54,387	134,418	43,621	46,480	44,317	25,829
Val (ID)	22,587	56,290	18,298	19,604	18,388	11,198
Val (OOD)	55,414	137,421	44,071	48,238	45,112	27,167
Total	282,021	673,909	218,810	232,189	222,910	126,594

Split	CA	CF	SL	PW	RG	ST
Train (c.)	65,373	23,861	83,133	24,057	417,436	257,666
Test (ID)	13,467	5,173	14,165	5,714	64,109	60,647
Test (OOD)	32,597	13,188	35,243	14,169	147,973	148,659
Val (ID)	14,801	5,183	14,670	5,696	61,641	57,658
Val (OOD)	34,574	12,898	35,677	14,011	149,289	148,472
Total	160,812	60,303	182,888	63,647	840,448	673,102

Table 5: Number of supervision samples per task across different dataset splits. All splits are performed at the protein level. c. = condensed training split; ID = In-Distribution; OOD = Out-of-Distribution (see Appendix A). #Prot. = number of protein ids; GO = Gene Ontology (BP = Biological Process, MF = Molecular Function, CC = Cellular Component); EC = EC Number; CA = Catalytic Activity; CF = Cofactor; SL = Subcellular Location; PW = Pathway; RG = Regions; ST = Sites.

(e.g., XML, TSV) whose organization exhibit certain difference, we adopt the JSON format as the canonical representation since it is the native format targeted by our data construction pipeline.

```
UniProtKBEntry = {
    # --- Entry-level metadata ---
    "primaryAccession": str,
    "secondaryAccessions": list[str],
    "uniProtkbId": str,
    "entryType": str,
    "active": bool,
    "annotationScore": float,
    "entryAudit": dict,
    "inactiveReason": dict,
    "fragment": bool,
    "internalSection": dict,
    "extraAttributes": dict,

    # --- Biological metadata ---
    "sequence": dict,
    "proteinExistence": str,
    "organism": dict,
    "organismHosts": list[dict],
    "lineages": list[dict],
    "genes": list[dict],
    "geneLocations": list[dict],

   # --- Functional semantic annotations ---
   "uniProtKBCrossReferences": list[dict],
    "proteinDescription": dict,
    "comments": list[dict],
    "features": list[dict],
    "keywords": list[dict],
}
```

The information contained in a UniProtKB JSON record can be grouped into three categories: entry-level metadata, biological metadata, and functional semantic annotations. When constructing our dataset, all protein-centric tasks are derived exclusively from keys belonging to the functional semantic annotation category, while the other two categories are intentionally excluded.

Entry-level metadata primarily describes database bookkeeping information, such as accession management, curation status, and versioning history. These fields do not encode biologically meaningful properties of proteins themselves and therefore do not provide appropriate supervision signals for protein understanding.

Biological metadata, including information related to the source organism and gene context, does contain biologically relevant information. However, given that protein biological functions are often preserved across different species and gene backgrounds, which imposes strong evolutionary constraints and leads to high evolutionary conservation of the functional amino acid regions encoding these functions, defining the host organism or gene as prediction targets would primarily encourage the model to recognize where a protein comes from, rather than to reason about what the protein does.

UniProtKB provides protein information cross-referenced from external databases in the `uniProtKBCrossReferences` field. We extract Gene Ontology (GO) terms and identifiers from this field by selecting entries that are cross-referenced to the Gene Ontology database.

The `proteinDescription` field provides an exhaustive and structured representation of protein nomenclature for unambiguous protein identification. This field itself does not contain annotations that directly describe protein function. We only extract the complete set of EC numbers for each protein and leverage the functional semantics encoded by the EC numbers themselves to construct the corresponding text labels.

Task	UniProtKB Field	Identifier	Task level
Gene Ontology	uniProtKBCrossReferences	GO	Protein-level
EC Number	proteinDescription	–	Protein-level
Catalytic Activity	comments	CATALYTIC ACTIVITY	Protein-level
Cofactor	comments	COFACTOR	Protein-level
Subcellular Location	comments	SUBCELLULAR LOCATION	Protein-level
Pathway	comments	PATHWAY	Protein-level
Regions	features	$\mathcal{R}$	Residue-level
Sites	features	$\mathcal{S}$	Residue-level

Table 6: Mapping between protein-centric tasks and their corresponding UniProtKB source fields and identifiers. $\mathcal{R}$ denotes the set of region-level features, including Topological domain, Transmembrane, Intramembrane, Domain, Repeat, Calcium binding, Zinc finger, DNA binding, Nucleotide binding, Region, Coiled coil, Motif, and Compositional bias. $\mathcal{S}$ denotes the set of site-level features, including Active site, Metal binding, Binding site, and Site.

The comments field provides global, protein-level descriptions of proteins, covering diverse aspects of biological knowledge and spanning a wide range of annotation types. We extract information from this field following three considerations: (a) since annotations in this field are predominantly expressed in free text, a strictly standardized vocabulary is not guaranteed for all annotation types, and we therefore retain only annotation types whose semantics are relatively controlled; (b) to ensure that the extracted supervision signals are learnable from protein sequences, we restrict our selection to annotation types that admit a direct mapping to intrinsic, sequence-encoded biological properties, while excluding those that depend primarily on external biological context such as organism-specific phenotypes, experimental conditions, or application-oriented usage; (c) to avoid information overlap across tasks and reduce the risk of shortcut learning, we further exclude annotation types that provide high-level, cross-task summaries or free-text descriptions that may duplicate information available in other protein- or residue-level annotations. Based on these considerations, we select information from CATALYTIC ACTIVITY, COFACTOR, PATHWAY, and SUBCELLULAR LOCATION for task construction. Correspondingly, we discard comment types that do not satisfy the above criteria. For example, the FUNCTION annotation is typically excluded, as it provides highly abstract and weakly controlled summaries of protein function. Such descriptions may overlap with information used in other tasks (e.g., catalytic activity or pathway), and may also include content that does not admit a learnable mapping from the protein sequence itself.

The features field provides residue-level annotations that describe regions or sites of interest along the protein sequence. We retain only Regions and Sites, and exclude other feature types that depend on external biological context, experimental conditions, or processing states rather than intrinsic sequence-encoded properties.

We intentionally exclude the keywords field from sample construction. This is because the semantic granularity of keywords is coarse, which makes them unsuitable as learning targets for protein understanding. Moreover, keywords often aggregate multiple types of protein information, which can substantially overlap with information provided by Gene Ontology and free-text annotations.

B.4 Sample Construction Pipeline

When constructing samples for a given protein entry across multiple tasks, we first retrieve its complete UniProtKB JSON record via the UniProtKB REST API (Ahmad et al., 2025) (See Algorithm 1).

Note: In practice, to improve data construction efficiency, we issue batched API requests by querying multiple UniProt accessions in a single call, rather than fetching records individually. This strategy mitigates the impact of rate limits and enforced inter-request delays imposed by the UniProtKB REST API during large-scale sample generation. The API allows up to 100 protein accessions to be queried per batch request, which we adopt as the default batch size in our implementation.

Algorithm 1: Fetch UniProtKB JSON record for a protein entry

Require: UniProt primary accession protein_id
Ensure: UniProtKB JSON record for the queried entry
0: url ← "https://rest.uniprot.org/uniprotkb/" || protein_id || ".json"
0: record ← HTTP_GET_JSON(url)
0: **return** record =0

B.4.1 Gene Ontology (UniProtKB Cross-references)

The Gene Ontology (GO) annotations are extracted from the `uniProtKBCrossReferences` field by selecting entries with the `database` key equal to `"GO"`. Each such entry provides a single GO term together with its corresponding GO identifier for the target protein. We first aggregate all extracted GO entries across the protein record. Each entry is then converted into a GO element by combining the term name and its identifier, and grouped according to its ontology namespace into biological process (BP), molecular function (MF), and cellular component (CC). For each non-empty namespace, the corresponding GO elements are formatted into a template-ready natural language representation, from which one sample is instantiated. As a result, up to three GO samples (BP/MF/CC) can be generated for a single protein entry (See Algorithm 2).

B.4.2 EC Number (UniProtKB Protein Description)

We extract the complete set of EC numbers associated with each protein from the `proteinDescription` field. Each EC number consists of four digits. The first three digits encode a hierarchical enzymatic classification with progressively finer granularity, while the fourth digit uniquely specifies a concrete enzymatic reaction. We observe that the semantic space of the first three digits is finite and well defined. Leveraging the official textual definitions provided by the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB) (McDonald et al., 2009), we map each of the first three digits to its corresponding standardized textual description. For each protein record, we generate one sample per EC number, which includes both the EC identifier itself and the mapped textual interpretation derived from its hierarchical classification, when available (See Algorithm 3).

Algorithm 2: Construct Gene Ontology samples from a JSON protein record

Require: UniProtKB JSON record `record` for one protein entry
Require: Template pools `go_template_pools`
Ensure: Samples `bp_sample`, `mf_sample`, `cc_sample`

```
// Part I: Extract raw GO annotations from a full JSON record
raw_go_entries ← empty list
for all entry ∈ record.uniProtKBCrossReferences do
    if entry.database = "GO" then
        raw_entry ← {raw_id : entry.id, raw_term : FindValue(entry.properties, "GoTerm")}
        append raw_entry to raw_go_entries
    end if
end for
// Part II: Split and format raw GO annotations into template-ready style
bp_elements, mf_elements, cc_elements ← empty lists
for all e ∈ raw_go_entries do
    go_element ← FormatGO(e.raw_term, e.raw_id)
    if IsGOType(e, "BP") then
        append go_element to bp_elements
    else if IsGOType(e, "MF") then
        append go_element to mf_elements
    else if IsGOType(e, "CC") then
        append go_element to cc_elements
    end if
end for
bp_str ← JoinAsNL(bp_elements)
mf_str ← JoinAsNL(mf_elements)
cc_str ← JoinAsNL(cc_elements)
// Part III: Fill formatted GO annotations into templates to generate samples
bp_template ← PickTemplate(go_template_pools, "BP")
mf_template ← PickTemplate(go_template_pools, "MF")
cc_template ← PickTemplate(go_template_pools, "CC")
bp_sample ← Instantiate(bp_template, record, bp_str)
mf_sample ← Instantiate(mf_template, record, mf_str)
cc_sample ← Instantiate(cc_template, record, cc_str)
return bp_sample, mf_sample, cc_sample =0
```

B.4.3 Catalytic Activity & Cofactor & Subcellular Location & Pathway (UniProtKB Comments)

The raw protein information for catalytic activity, cofactor, subcellular location, and pathway is uniformly extracted from the `comments` field, by filtering entries according to their corresponding `commentType` values.

For the catalytic activity task, we leverage the annotated catalytic reactions of a protein and generate one sample for each catalytic activity in which the protein is involved.

For the cofactor task, a protein may be associated with one or multiple cofactors, which can act at the same or different sites to support similar or distinct biochemical functions. It is therefore necessary to preserve the boundaries between different cofactor groups, rather than merging all cofactors into a single description. UniProtKB encodes this site-level semantic structure by grouping cofactors that act on the same site within a single `COFACTOR` comment entry, while initiating a new comment block for cofactors associated with a different site. Accordingly, we generate one sample per `COFACTOR` comment entry, corresponding to one group of cofactors acting on the same site. Within each group, all cofactor names are concatenated together with the associated textual note, if available, which provides

additional contextual information such as alternative cofactors, stoichiometry, or other site-specific remarks. The resulting representation is then filled into the corresponding template to construct a supervision sample.

For the subcellular location task, each annotated subcellular location of a protein is combined with its associated membrane topology and orientation information, if available, to form a subcellular location description element. UniProtKB further indicates whether a given subcellular location annotation refers to the canonical protein product or to a specific isoform or alternative product, and, in the latter case, explicitly provides the corresponding isoform or alternative product name. We extract this attribution information accordingly and, when applicable, include the specified isoform or alternative product name as part of the supervision annotation. All such elements are then aggregated and, together with the associated free-textual note if present, formatted into a template-ready natural language representation. Finally, a matched template is called to generate a supervision sample, typically resulting in one sample per protein entry.

For the pathway task, UniProtKB represents each pathway in which a protein is involved using a hierarchical annotation scheme with controlled vocabulary. Well-characterized pathways are annotated with three hierarchical levels. When available, the specific step at which the protein functions within the pathway is also indicated. Pathways that are not fully characterized are represented using a two-level structure. Our templates are designed to accommodate this hierarchical structure, thereby preserving the pathway-level organization in the generated samples. Accordingly, one supervision sample is created for each pathway in which the protein is involved (See Algorithm 4 and 5).

B.4.4 Regions & Sites (UniProtKB Features)

The raw protein information for both the regions and sites tasks is uniformly extracted from the UniProtKB `features` field. The construction procedures for these two residue-level tasks follow a highly consistent strategy.

For each residue-level sequence annotation, UniProtKB provides a start residue position and an end residue position, together with a categorical `type` field that specifies the semantic class of the annotated feature. In addition, an optional free-text description may be present. We first filter feature entries based on their `type` values to restrict the scope to region-level and site-level annotations of interest. For each retained feature entry, we extract the feature type, start and end residue locations, and the associated textual description when available. Leveraging the official textual definitions provided in the UniProtKB documentation, we map each feature type to a fixed, standardized textual definition that specifies the semantic meaning of the annotated segment in a template-compatible form. Finally, for each residue-level annotation, we construct one instruction sample using the mapped textual definition, residue span, and optional description (See Algorithm 6).

Algorithm 3: Construct EC number samples from a JSON protein record

Require: UniProtKB JSON record `record` for one protein entry
Require: Template pools `ec_template_pools`
Ensure: Samples `ec_samples`
0: **// Part I: Extract the full set of EC numbers from a full JSON record**
0: `ec_numbers` ← `record.proteinDescription.recommendedName.ecNumb`
0: **// Part II: Parse EC numbers and map hierarchical textual definitions**
0: `ec_records` ← empty list
0: **for all** `e` ∈ `ec_numbers` **do**
0: `class_text` ← LOOKUP(GETFIRSTDIGIT(`e`))
0: `subclass_text` ← LOOKUP(GETFIRSTDIGIT(`e`), GETSECONDDIGIT(`e`))
0: `subsubclass_text` ← LOOKUP(GETFIRSTDIGIT(`e`), GETSECONDDIGIT(`e`), GETTHIRDDIGIT(
0: `ec_record` ← {`ec_number` : `e`, `ec_text` : {`class_text`, `subclass_text`, `subsubclass_text`}}
0: append `ec_record` to `ec_records`
0: **end for**
0: **// Part III: Fill EC information into templates to generate samples**
0: `ec_samples` ← empty list
0: **for all** `r` ∈ `ec_records` **do**
0: `ec_template` ← PICKTEMPLATE(`ec_template_pools`, `r`)
0: `ec_sample` ← INSTANTIATE(`ec_template`, `record`, `r`)
0: append `ec_sample` to `ec_samples`
0: **end for**
0: **return** `ec_samples` =0

Algorithm 4: Construct comment-based task samples from a JSON protein record

Require: UniProtKB JSON record `record` for one protein entry
Require: Template pools `ca_tpl_pools`, `cf_tpl_pools`, `sl_tpl_pools`, `pw_tpl_pools`
Ensure: Samples `ca_samples`, `cf_samples`, `sl_samples`, `pw_samples`
0: **// Part I: Extract raw comment annotations from a full JSON record**
0: `raw_ca_entries`, `raw_cf_entries`, `raw_sl_entries`, `raw_pw_entries` ← empty lists
0: **for all** `c` ∈ `record.comments` **do**
0: **if** `c.commentType` = "CATALYTIC ACTIVITY" **then**
0: `raw_ca_entry` ← `c.reaction.name`
0: append `raw_ca_entry` to `raw_ca_entries`
0: **else if** `c.commentType` = "COFACTOR" **then**
0: `cofactors` ← [`x.name` **for** `x` ∈ `c.cofactors`]

```
        notes ← [t.value for t ∈ c.note.texts]
        raw_cf_entry ← {cofactors : cofactors, notes : notes}
        append raw_cf_entry to raw_cf_entries
    else if c.commentType = "SUBCELLULAR LOCATION" then
        loc_info ← [{location : s.location.value, orientation : s.orientation.value, topology : s.topology.value} for s ∈ c.subcellularLocations]
        notes ← [t.value for t ∈ c.note.texts]
        raw_sl_entry ← {name : c.molecule, loc_info : loc_info, notes : notes}
        append raw_sl_entry to raw_sl_entries
    else if c.commentType = "PATHWAY" then
        raw_pw_entry ← [t.value for t ∈ c.texts]
        append raw_pw_entry to raw_pw_entries
    end if
end for
// Part II: Format raw comment annotations into template-ready style
cf_records ← empty list
for all e ∈ raw_cf_entries do
    cf_str ← JoinAsNL(e.cofactors)
    note_str ← JoinWithSpace(e.notes)
    cf_record ← {Cofactors : cf_str, Notes : note_str}
    append cf_record to cf_records
end for
sl_records ← empty list
for all e ∈ raw_sl_entries do
    loc_elements ← empty list
    for all L ∈ e.loc_info do
        loc_element ← FormatLocation(L.location, L.topology, L.orientation)
        append loc_element to loc_elements
    end for
    sl_str ← JoinAsNL(loc_elements)
    note_str ← JoinWithSpace(e.notes)
    sl_record ← {Name : e.name, SubcellularLocations : sl_str, Notes : note_str}
    append sl_record to sl_records
end for
pw_records ← empty list
for all e ∈ raw_pw_entries do
    for all p ∈ e do
        (super_pw, pw, sub_pw, step) ← ParsePathwayText(p)
        pw_record ← {SuperPathway : super_pw, Pathway : pw, SubPathway : sub_pw, Step : step}
        append pw_record to pw_records
    end for
end for=0
```

Algorithm 5: Construct comment-based task samples from a JSON protein record (continued)

```
// Part III: Fill formatted comment annotations into templates to generate samples
(A) Catalytic activity
ca_samples ← empty list
for all r ∈ raw_ca_entries do
    ca_tpl ← PickTemplate(ca_tpl_pools)
    ca_sample ← Instantiate(ca_tpl, record, r)
    append ca_sample to ca_samples
end for
(B) Cofactor
cf_samples ← empty list
for all r ∈ cf_records do
    cf_tpl ← PickTemplate(cf_tpl_pools, r)
    cf_sample ← Instantiate(cf_tpl, record, r)
    append cf_sample to cf_samples
end for
(C) Subcellular location
sl_samples ← empty list
for all r ∈ sl_records do
    sl_tpl ← PickTemplate(sl_tpl_pools, r)
    sl_sample ← Instantiate(sl_tpl, record, r)
    append sl_sample to sl_samples
end for
(D) Pathway
pw_samples ← empty list
for all r ∈ pw_records do
    pw_tpl ← PickTemplate(pw_tpl_pools, r)
    pw_sample ← Instantiate(pw_tpl, record, r)
    append pw_sample to pw_samples
end for
return ca_samples, cf_samples, sl_samples, pw_samples =0
```

Algorithm 6: Construct region and site samples from a JSON protein record

```
Require: UniProtKB JSON record record for one protein entry
Require: Template pools rg_tpl_pools, st_tpl_pools
Ensure: Samples rg_samples, st_samples
// Part I: Extract raw feature annotations from a full JSON record
raw_rg_entries, raw_st_entries ← empty lists
for all f ∈ record.features do
    (t, start, end, desc) ← (f.type, f.location.start.value, f.location.end.value, f.descript
    raw_entry ← {type : t, start : start, end : end, desc : desc}
    if IsRegionType(t) then
        append raw_entry to raw_rg_entries
    else if IsSiteType(t) then
        append raw_entry to raw_st_entries
    end if
end for
// Part II: Format raw feature annotations into template-ready style
rg_records ← empty list
for all e ∈ raw_rg_entries do
    text ← Lookup(e.type, "Regions")
    rg_record ← {StartLocation : e.start, EndLocation : e.end, Text : text, Description : e.desc}
    append rg_record to rg_records
end for
st_records ← empty list
for all e ∈ raw_st_entries do
    text ← Lookup(e.type, "Sites")
    st_record ← {StartLocation : e.start, EndLocation : e.end, Text : text, Description : e.desc}
    append st_record to st_records
end for
// Part III: Fill formatted feature annotations into templates to generate samples
(A) Regions
rg_samples ← empty list
for all r ∈ rg_records do
    rg_tpl ← PickTemplate(rg_tpl_pools, r)
    rg_sample ← Instantiate(rg_tpl, record, r)
    append rg_sample to rg_samples
```

```
end for
(B) Sites
st_samples ← empty list
for all r ∈ st_records do
    st_tpl ← PICKTEMPLATE(st_tpl_pools, r)
    st_sample ← INSTANTIATE(st_tpl, record, r)
    append st_sample to st_samples
end for
return rg_samples, st_samples =0
```

C Detailed Fine-grained Pre-training algorithm for Protein–Text Alignment

C.1 Common Notation

Let a protein sequence be denoted by x of length L, and its residue-level embeddings produced by a pretrained protein language model (PLM, e.g., ESM) be

$$H = f_P(x) \in \mathbb{R}^{L \times d_P}, \quad H_i \in \mathbb{R}^{d_P}$$

where $\mathbb{H}_{\sqsupset}$ denote the embedding of residue i.

A textual description y (e.g., functional annotation) is encoded as

$$h_T = f_T(y)_{\texttt{[CLS]}} \in \mathbb{R}^{d_T},$$

where the [CLS] token representation is used.

The Q-former(F_Q) contains two parts, they are protein cross-modality module($F_{Q,prot}$) and text encoding module($F_{Q,text}$). In the protein cross-modality module, soft queries(learnable embeddings) are processed through self-attention layer and then apply their attention on protein embeddings through a cross attention layer. After the protein embeddings are gathered in the cross attention module, the embeddings would be processed by the feed forword network. While for the text encoding module, the architecture is simple text encoder, i.e. PubMedBert.

The Q-Former consists of N_q learnable query tokens $Q \in \mathbb{R}^{N_q \times d_Q}$ that cross-attend to the protein encoder outputs H:

$$Z = F_{Q,prot}(Q, H) \in \mathbb{R}^{N_q \times d_Q}.$$

Or if the attention of protein and text is bidirectional we could write Z as:

$$Z = F_Q(Q, H, y) \in \mathbb{R}^{N_q \times d_Q}. \tag{9}$$

If the attention is causal from protein to text, we could write:

$$y = F_{Q,causal}(Q, H, y_{1:T-1}) \tag{10}$$

C.2 Baseline: BLIP2 Stage–1 Objectives

Let a protein sequence be denoted by x of length L, and a textual description by y. The protein encoder (PLM) produces contextual residue embeddings

$$H = f_P(x) \in \mathbb{R}^{L \times d_P}, \tag{11}$$

while the text encoder (e.g., PubMedBERT) outputs a sequence of token representations, from which we take the [CLS] token as

$$h_T = F_{Q,text}(y)_{\texttt{[CLS]}} \in \mathbb{R}^{d_T}. \tag{12}$$

The Q-Former employs N_q learnable query tokens $Q \in \mathbb{R}^{N_q \times d_Q}$ and attends to the protein embeddings:

$$Z = F_{Q,prot}(Q, H) \in \mathbb{R}^{N_q \times d_Q}. \tag{13}$$

Or if the attention on the text is not blocked (bidirectional) we could write Z as:

$$Z = F_Q(Q, H, y) \in \mathbb{R}^{N_q \times d_Q}. \tag{14}$$

Protein–Text Contrastive (PTC) loss. For each protein–text pair (x_i, y_i), we first individually calculate $Z = F_{Q,prot}(Q, H) \in R^{N_q d_Q}$ and h_T, then compute the cosine similarity as the maximum query–text similarity:

$$\mathcal{L}_{\text{PTC}} = \frac{1}{2B} \sum_{i=1}^{B} \Big[-\log \frac{\exp(s_{ii}/\tau)}{\sum_j \exp(s_{ij}/\tau)} - \log \frac{\exp(s_{ii}/\tau)}{\sum_j \exp(s_{ji}/\tau)} \Big] \tag{15}$$

where τ is a temperature parameter.

Protein–Text Matching (PTM). In the matching scenario, the attention is bidirectional between protein and text modalities. Given the query outputs Z and text features, a binary classifier distinguishes matched and mismatched pairs:

$$\mathcal{L}_{\text{PTM}} = \text{CE}(\,\text{MLP}_{\text{clf}}(F_Q(Q, H, y)),\ \text{label} \in \{0, 1\})\,. \tag{16}$$

Language Modeling (LM). In the language modeling, there is causal attention from protein to text. When the Q-Former is used to condition a text decoder, the LM loss is

$$\mathcal{L}_{\text{LM}} = \mathcal{L}_{NTP}\big(F_{Q,causal}(Q, H, y_{1:T-1}),\ y_{2:T}\big). \tag{17}$$

Baseline objective.

$$\boxed{\mathcal{L}_{\text{BLIP2-base}} = \mathcal{L}_{\text{PTC}} + \lambda_{\text{PTM}}\mathcal{L}_{\text{PTM}} + \lambda_{\text{LM}}\mathcal{L}_{\text{LM}}.} \tag{18}$$

C.3 Strategy 1: Local Contrastive Learning (ESM-only Pooling)

Let $R \subseteq \{1, \ldots, L\}$ be annotated residues and $\tilde{R}$ be annotated the rest of annotated residues. We pool the PLM embeddings and normalize:

$$H_R = \mathcal{M}_{\tilde{R}}(H) \tag{19}$$

If you want only the mean pooling of the annotated embeddings, you could set

$$H_R = \text{norm}\Big(\frac{1}{|R|}\sum_{i \in R} H_i\Big) \tag{20}$$

Under this local setup, we could have Q The local matching loss is

$$\boxed{\mathcal{L}_{\text{local-ptm}} = \mathcal{L}_{\text{PTM}}(F_Q(Q, H_R, y)).} \tag{21}$$

$$\boxed{\mathcal{L}_{\text{local-lm}} = \mathcal{L}_{\text{lm}}(F_{Q,causal}(Q, H_R, y)).} \tag{22}$$

Total objective:

$$\mathcal{L}^{(1)} = \mathcal{L}_{\text{BLIP2-base}} + \lambda_{\text{loc}}(\mathcal{L}_{\text{local-ptm}} + \mathcal{L}_{\text{local-lm}}). \tag{23}$$

C.4 Strategy 2: Masked-Residue Counterfactual Negatives

To capture residue necessity, the annotated residues R are masked:

$$\widetilde{H} = \mathcal{M}_R(H), \tag{24}$$

$$z_P^+ = F_{Q,prot}(Q, H), \quad z_P^- = F_{Q,prot}(Q, \widetilde{H}). \tag{25}$$

A margin-based counterfactual loss is defined as

$$\boxed{\mathcal{L}_{\text{cf}} = \mathbb{E}\big[\max\big(0, m + s(z_P^-, h_T) - s(z_P^+, h_T)\big)\big]\,.} \tag{26}$$

The total loss is

$$\mathcal{L}^{(2)} = \mathcal{L}_{\text{BLIP2-base}} + \lambda_{\text{cf}}\mathcal{L}_{\text{cf}}. \tag{27}$$

D Appendix: Pretraining Experiments Details

D.1 Metrics

We evaluate models along two complementary dimensions: global protein–text alignment and fine-grained evidence localization. While global retrieval metrics quantify semantic correspondence at the sequence level, localization metrics explicitly measure whether predictions are grounded in correct biochemical evidence.

D.1.1 Global Protein–Text Retrieval

We first evaluate global alignment between protein and text representations using bidirectional retrieval metrics. Let $\mathcal{P}$ denote a set of protein sequences and $\mathcal{T}$ their corresponding textual descriptions. Both modalities are encoded using a shared Q-Former architecture into a joint embedding space.

We consider two retrieval directions:

- **Protein-to-Text (P2T):** given a protein embedding, retrieve its corresponding text description.
- **Text-to-Protein (T2P):** given a text embedding, retrieve its corresponding protein.

For each direction, we report results under two settings:

- **Fullset retrieval**, where each query is ranked against all candidates in the evaluation split.
- **In-batch retrieval**, where candidates are restricted to the current mini-batch.

We report the following metrics:

- **Top-1 Accuracy (Acc):** the fraction of queries for which the correct match is ranked first.
- **Recall@20 (Rec@20):** the fraction of queries for which the correct match appears among the top 20 retrieved candidates.

These metrics primarily measure global semantic alignment between protein and text representations, but do not capture whether predictions rely on correct localized biochemical evidence.

D.1.2 Fine-Grained Evidence Localization (Local MC)

To explicitly evaluate evidence grounding, we introduce a *local multi-choice* (loc_mc) localization task that probes whether the model can identify the correct protein location associated with a given textual evidence description.

Task formulation. For each annotated protein–text pair, we construct a candidate set of protein locations

$$\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_K\},$$

where $K = 8$ by default. Each candidate ℓ_k corresponds to a plausible biochemical location on the

protein, which may represent an individual residue, a contiguous residue span, or a predefined functional region.

Exactly one candidate corresponds to the ground-truth annotated evidence, while the remaining candidates are randomly sampled distractor locations. Under this setup, random guessing yields an expected accuracy of $1/K$.

Scoring and selection. Each candidate location ℓ_k is encoded using the protein-side tower, while the corresponding textual evidence is encoded using the text-side tower. Localization is performed by computing a similarity score between the text embedding and each candidate location embedding. Candidates are ranked according to these similarity scores, and the highest-scoring location is selected.

Localization metrics are computed per annotation and averaged across all annotations associated with a protein.

Metrics. We report three complementary localization metrics:

- **MC Accuracy (Acc_{loc}):** the fraction of instances for which the ground-truth location is ranked first.
- **MC Mean Reciprocal Rank (MRR_{loc}):**

$$\text{MRR}_{\text{loc}} = \mathbb{E}\left[\frac{1}{\text{rank}(\ell^*)}\right],$$

  where ℓ^* denotes the ground-truth location.
- **MC Recall@5 ($\text{Rec@5}_{\text{loc}}$):** the fraction of instances for which the ground-truth location appears among the top 5 ranked candidates.

While global retrieval metrics assess whether protein and text representations are aligned at the sequence level, strong retrieval performance does not guarantee that predictions depend on correct localized evidence. The proposed localization metrics explicitly evaluate *where* the model grounds its reasoning on the protein sequence. Improvements in loc_mc metrics therefore reflect enhanced evidence-dependent reasoning, rather than reliance on global correlations or memorization.

D.2 Training Behavior

We could notice from the figure 6 that the loss is stably decreasing with descending learning rate. The learning rate is largest learning rate is $3e-5$ and it has warm-up period for the first 3000 steps.

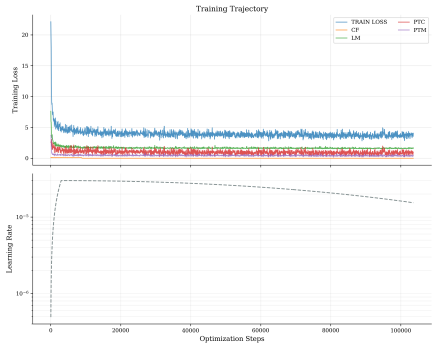


Figure 6: Behavior of training loss during stage 1: Q-Former training

D.3 Strategy Ablation Results

Table 7 studies the effect of weighting the local contrastive objective (λ_{local}) and the counterfactual masking objective (λ_{cf}) during fine-grained pre-training. When both objectives are removed, localization performance degrades substantially, indicating that global supervision alone fails to induce representations that reliably encode causal evidence at the residue or span level. Introducing either objective in isolation improves localization quality, suggesting that each objective imposes a complementary inductive bias. Specifically, increasing λ_{local} primarily improves top-ranked localization accuracy and mean reciprocal rank, reflecting sharper alignment between evidence representations and textual descriptions. In contrast, increasing λ_{cf} consistently improves recall-oriented metrics, indicating enhanced coverage of alternative plausible evidence under perturbations. Combining both objectives yields the strongest overall localization performance, while preserving protein-to-text retrieval accuracy on held-out test data, demonstrating that fine-grained supervision enhances evidence grounding without degrading global semantic alignment.

D.4 Hyperparameter ablation

Table 8 reports an ablation study of fine-grained pretraining strategies across multiple datasets with varying sequence diversity and homology levels. Strategy 1 (local contrast) explicitly aligns localized protein evidence with textual descriptions, while Strategy 2 (counterfactual masking) enforces causal dependence on evidence by contrasting masked and unmasked variants. When neither strategy is applied, localization performance degrades consistently across all datasets, indicating that global supervision alone does not induce robust evidence-aware representations. Enabling ei-

Run	λ_{local}	λ_{cf}	$\text{Acc}_{\text{loc}} \uparrow$	$\text{MRR}_{\text{loc}} \uparrow$	$\text{Rec@5}_{\text{loc}} \uparrow$	P2T Acc_{test}	P2T $\text{Rec@20}_{\text{test}}$
0	0.25	0.25	37.14	0.5015	60.41	15.94	80.08
1	0.50	0.25	37.96	0.5047	65.31	15.54	79.68
2	0.25	0.50	38.78	0.5115	64.49	15.94	80.48
3	0.50	0.50	37.55	0.5113	66.12	15.94	79.28
4	0.75	0.50	**42.45**	**0.5350**	67.76	15.54	78.88
5	0.50	0.75	41.35	0.5240	67.25	15.94	78.88
6	0.75	0.75	40.00	0.5299	**68.16**	15.94	80.08
abl	0.25	0.00	35.51	0.4552	48.98	15.54	80.88
abl	0.00	0.25	31.43	0.4462	58.37	15.94	80.88
abl	0.00	0.00	30.20	0.4190	46.53	**16.33**	80.08

Table 7: Ablation of fine-grained objective weights. Acc_{loc}, MRR_{loc}, and $\text{Rec@5}_{\text{loc}}$ measure evidence localization quality on the validation set. Protein-to-text (P2T) retrieval metrics are reported on the test set.

Strat.	P2T Acc	P2T Rec@20	T2P Acc	T2P Rec@20	InB P2T Acc	InB T2P Acc	Acc_{loc}	MRR_{loc}	$\text{Rec@5}_{\text{loc}}$
act_X									
12	1.36	20.68	1.59	22.73	3.07	3.07	44.09	0.588	76.25
1	1.36	21.25	1.59	22.73	3.07	3.07	42.16	0.546	65.00
2	1.48	20.57	1.59	22.61	3.07	3.07	43.30	0.558	67.05
0	1.25	21.59	1.48	22.84	3.07	3.07	41.48	0.544	68.41
act_mf50									
12	33.87	96.24	36.02	96.77	60.22	61.29	58.70	0.722	91.85
1	34.95	96.77	34.95	96.77	60.75	60.75	52.72	0.676	90.22
2	35.48	96.77	34.95	96.77	61.83	61.29	52.17	0.658	85.87
0	34.95	96.77	36.02	96.77	61.29	60.75	46.74	0.620	80.98
act_mp50									
12	21.04	80.60	21.04	80.87	50.27	51.09	59.83	0.723	91.29
1	21.31	81.15	21.58	81.15	51.09	51.09	49.16	0.633	82.87
2	21.58	80.60	21.31	80.87	50.27	51.09	56.18	0.696	89.04
0	21.31	81.15	21.58	81.15	51.09	51.09	49.16	0.633	82.87
act_mp90									
12	9.67	77.36	9.78	77.80	54.84	55.38	56.98	0.691	84.59
1	21.58	80.60	21.31	80.87	50.27	51.09	56.18	0.696	89.04
2	9.34	77.03	10.11	77.69	54.18	55.27	55.88	0.676	83.48
0	9.34	77.25	10.00	77.91	54.29	55.38	47.89	0.622	81.26

Table 8: Dataset-wise ablation of fine-grained pretraining strategies. Strategy codes indicate which objectives are enabled: 0 = none, 1 = local contrast only, 2 = counterfactual masking only, 12 = both. When enabled, each strategy uses a default weight $\lambda = 0.25$.

ther strategy in isolation improves localization metrics, with Strategy 1 favoring precision-oriented measures such as Accloc and MRRloc, and Strategy 2 improving recall-oriented behavior. Combining both strategies (code 12) yields the strongest and most consistent improvements, particularly on datasets with higher redundancy or structural similarity, while preserving retrieval performance. These results suggest that the two objectives impose complementary inductive biases that generalize across data regimes.

D.5 large scale fine-grained pretraining

The table 9 presents the loss performance for different dataset.

The table10 presents the localization multi-choice performance on the two residue-level tasks.

The table 11 presents the baseline retrieval performance on our SWISSPROT-EVIDENCE's val and test dataset. Here the metrics are not distinguished between id and ood dataset.

The table 12 present the re-rank test retrieval performance of the baseline model.

The table 13 present the different loss for the baseline model. We notice that the diffenrence between ID and OOD is not significant.

Task	Split	Total Loss ↓	LM ↓	PTC ↓	PTM ↓
Cofactor	ID	4.148	1.788	1.636	0.725
	OOD	4.183	1.821	1.612	0.750
EC number	ID	2.843	1.542	0.683	0.618
	OOD	2.869	1.543	0.688	0.638
Gene Ontology (MF)	ID	2.834	1.648	0.761	0.425
	OOD	2.882	1.656	0.777	0.449
Catalytic activity	ID	3.621	1.592	0.936	1.094
	OOD	3.583	1.623	0.926	1.035
Sites (binding/catalytic)	ID	5.881	1.527	2.974	0.628
	OOD	5.964	1.536	3.038	0.633
Regions (motifs/domains)	ID	–	–	–	–
	OOD	–	–	–	–

Table 9: Validation loss decomposition for Stage-1 pretraining. OOD-family splits exhibit systematically higher loss for functionally demanding tasks, indicating reduced robustness under family shift.

The table 14 presents the localization metrics for the baseline model.

Task	Split	MC Acc ↑	MC MRR ↑	MC Rec@5 ↑
Sites	ID	31.72	0.5235	84.76
	OOD	30.91	0.5092	83.48

Table 10: Multi-choice localization performance of Stage-1 models. Only the Sites task shows non-zero localization in the provided logs.

E Appendix: Example Samples

E.1 Catalytic Activity

Sample A
Instruction: Determine the catalytic function of this enzyme using the provided protein information.
Sequence: MRIGLYPGTFDPITLGHQDIIQRALELVDRL VIGVAINRDKSPLFALEDRVAMVREECDRIVAKRGGE IVVHPFENLLIDCARDVGATVIVRGLRAVADFEYEFQ MVGMNRAMDDSIETVFLMADARRQAIASKLVKEIAR LGGDVSKFVSPTVRDALVGRFAR
Textual label: Catalytic Activity: According to the analyzed protein data, the enzyme catalyzes this reaction: (R)-4'-phosphopantetheine + ATP + H(+) = 3'-dephospho-CoA + diphosphate.
Start location: None **End location:** None

Sample B
Instruction: Based on the provided protein information, describe the catalytic reaction that this enzyme facilitates.
Sequence: MIVVRKALPEEHKEILQVAKQSKYTKDFSN QVMFSSEAAYNKGWIHVAEHEGEIRGFYCIREKVRAP ETVLYFIGVAQEAKGLGLGKKLIEHIMATTRHRRLTL NVNKQNEEARAFYDRLGFTVAGESLGGEGLALFKEW
Textual label: Catalytic Activity: Analysis of the protein suggests that this enzyme catalyzes the reaction: 5-aminomethyl-dUMP in DNA + acetyl-CoA = 5-acetylaminomethyl-dUMP in DNA + CoA + H(+).
Start location: None **End location:** None

Sample C
Instruction: Describe the biochemical transformation catalyzed by this enzyme using the given protein information.
Sequence: MIQSQHSQTARHALAETVVLAKARKNLSF AQLTEGTGLSEAFVTAALLGQHALPADAARVVADKL GLDDDAVLLLQMIPLRGSIDDRVPTDPTIYRFYEMLQ VYGTTLKALVHEKFGDGIISAINFRLDVKKVDDPEGG SRAVITLDGKYLPTKPF
Textual label: Catalytic Activity: The protein exhibits catalytic activity for the reaction: cyanate + hydrogencarbonate + 3 H(+) = NH4(+) + 2 CO2.
Start location: None **End location:** None

E.2 Cofactor

Sample A
Instruction: Refer to the protein data to determine which cofactors the enzyme uses and summarize any details regarding binding or metal preference.
Sequence: MSSVTLVAKSKLPTPWGTFTLVGFQETGTG KDHAALVMGDITGDEPVLGRIHSECLTGDALFSLRCD CGFQLQAALENIAKAGRGVLLYVRQEGRGIGLLNKIR AYHLQDQGADTVEANVALGFAADMRDYTICADMLK QLEVKSLKLMTNNPRKMKAMESFGIPVAERVPLQEG RNPHNEFYLSTKANKLDHMFKK
Textual label: Cofactor: The enzyme employs Zn(2+) for its catalytic reactions, and binds 1 zinc ion per subunit.
Start location: None **End location:** None

Sample B
Instruction: Identify the cofactors associated with this protein that are required for its enzymatic activity.
Sequence: MNYSKAKYIMGAAKVSQLPEDTGVEVAFA GRSNAGKSSALNTLTDQKGLARVSKTPGRTQLINLFD LGNNNRLVDLPGYGYAKVSESIKRQWQSEMENYLTS RKCLNGIVLLVDSRHELKEFDSLMIEMAISFDLNLHIL LTKADKLNNKERAQANRMIESFLKTFVSTDKISYQLF SSLTKMGLDKFKEKLDTWYQ
Textual label: Cofactor: The enzyme associates with Mg(2+) to carry out its function.
Start location: None **End location:** None

Sample C
Instruction: From the protein entry, report all required cofactors together with relevant details about their binding as recorded in the notes.
Sequence: MEAILNKNMKILIVDDFSTMRRIVKNLLRD LGFNNTQEADDGLTALPMLKKGDFDFVVTDWNMPG MQGIDLLKNIRADEELKHLPVLMITAEAKREQIIEAAQ AGVNGYIVKPFTAATLKEKLDKIFERL
Textual label: Cofactor: The enzyme requires Mg(2+) for its catalytic process, and binds 1 Mg(2+) ion per subunit.
Start location: None **End location:** None

E.3 Subcellular Location

Sample A
Instruction: Describe where within the cell this protein is situated based on its characteristics.
Sequence: MSLEQLAGRLISGDIGATAVIKMTGEIIYQS PNWSVDGVHAINVYKNREPSIIIQGVKYSVIDVNEDR LIATNVGGQGHIVGAVAGGKALLIGYVSPNGDARTA YIQIDKTARQLSKIL
Textual label: Subcellular Location: This protein is situated in Cytoplasm, cytoskeleton.
Start location: None **End location:** None

Sample B
Instruction: Determine the cellular compartment or region in which this protein is found, using the provided protein data.
Sequence: MMEAHFKYHGNLTGRAHFPTLATEVDTSSD KYSNLYMYVGLFLSLLAILLILLFTMLLRLKHVISPINS DSTESVPQFTDVEMQSRIPTP
Textual label: Subcellular Location: The protein is found in Membrane (Single-pass membrane protein).
Start location: None **End location:** None

Sample C
Instruction: Identify the compartment, membrane, or organelle where this protein is localized.
Sequence: MKLLSITLTSIVISMVFYQTPITTEARSLRKT NDQDHFKAGFTDDFVPTSPGNSPGVGHKKGNVNVEG FQDDFKPTEGRKLLKTNVQDHFKTGSTDDFAPTSPGH SPGVGHKKGNVNVESSEDDFKHKEGRKLQQTNGQN HFKTGSTDDFAPTSPGNSPGIGHKKGHANVKGFKDD FAPTEEIRLQKMNGQDHFKTGSTDDFAPTTPGNSPGM GHKKGDDFKPTTPGHSPGVGHAVKNDEPKA
Textual label: Subcellular Location: C-terminally encoded peptide 9.2: Secreted, extracellular space, apoplast, accumulates in xylem sap under nitrogen (N)-starved conditions
Start location: None **End location:** None

Table 11: Test retrieval performance of the Stage-1 pretrained model. Since ID and OOD splits share identical retrieval metrics, results are reported without split.

	Full-set				In-batch			
Task	P2T Acc	P2T R@20	T2P Acc	T2P R@20	P2T Acc	P2T R@20	T2P Acc	T2P R@20
Cofactor	0.27	2.54	0.23	1.62	6.09	52.03	4.73	47.21
EC	0.29	3.90	0.21	2.38	12.63	59.40	8.96	55.32
Catalytic activity	1.32	12.95	0.50	5.24	24.89	76.32	15.48	70.50
Gene ontology	3.17	16.81	1.00	7.58	28.91	68.92	19.36	64.43
Sites	0.07	0.42	0.01	0.19	1.97	31.65	1.74	31.61
Regions	0.37	2.09	0.12	0.91	5.73	38.60	3.65	37.28

Table 12: Re-ranked test retrieval performance of the Stage-1 pretrained model. Retrieval metrics are identical across ID and OOD splits and therefore reported jointly.

	Full-set				In-batch			
Task	P2T Acc	P2T R@20	T2P Acc	T2P R@20	P2T Acc	P2T R@20	T2P Acc	T2P R@20
Cofactor	0.23	3.16	0.08	1.46	7.30	58.18	4.14	48.75
EC	0.27	3.69	0.15	2.34	13.82	67.62	7.56	55.69
Catalytic activity	1.48	14.37	0.40	4.11	29.00	87.20	12.80	73.25
Gene ontology	3.12	17.22	0.51	4.70	30.94	77.16	12.66	59.01
Sites	0.10	0.50	0.01	0.23	2.17	31.73	1.80	31.69
Regions	0.45	2.38	0.09	0.78	6.90	43.78	3.80	38.75

E.4 Pathway

Sample A
Instruction: Summarize the pathway hierarchy of this protein and the metabolic step it carries out.
Sequence: MNTETKSLPAWLDKVRWDDNGLVPVIAQEASTNDVLMFAWMNREALAKTIETQRAVYYSRSRKRLWFKGEESGHVQHVHEVRLDCDEDVVLLKVEQVSGIACHTGRHSCFFQKFEGTVDGGDWVAVEPVLKDPEHIYK
Textual label: Pathway: Engaged in L-histidine from 5-phospho-alpha-D-ribose 1-diphosphate, this protein functions within the L-histidine biosynthesis pathway of Amino-acid biosynthesis, participating in step 3/9.
Start location: None **End location:** None

Sample B
Instruction: Describe the metabolic role of this protein by identifying its pathway and broader hierarchical association.
Sequence: MDILLLLLVCLLTCGGQMLQKQAVVSWQRRPCNHWQKLRSPWLIASVAALGCGMLLWIYLLQRLPLSMAYPMLSINLVLVLVGSRLFFHEQISYHNWLGVGAIIVGALLLGGLL
Textual label: Pathway: It participates in the lipopolysaccharide biosynthesis pathway, a component of the Bacterial outer membrane biogenesis system.
Start location: None **End location:** None

Sample C
Instruction: Identify how this protein integrates into the pathway hierarchy and specify the step it participates in.
Sequence: MELKSYVREIADFPKKGVSYKDITPLLLDPEAMRLAVKLFMKNLPHGKIDKVVGIESRGFFFATMLAEKLNAGFIPVRKPGKLPHRTHKENYDLEYGSDSLEIHEDAIKKGEKVLLHDDVLATGGTAAATCKLIEKCGGEIVQCNFLVELEFLKGVDKLKGYDVCSLIKYS
Textual label: Pathway: Participating in AMP from adenine under AMP biosynthesis via salvage pathway of Purine metabolism, this protein functions in step 1/1.
Start location: None **End location:** None

E.5 Gene Ontology

Sample A
Instruction: From the protein information below, find out the biological process that this protein is known to participate in.
Sequence: MNSSLDFLILILMFGGTSSNSVKQTGQITVSEGASVTMNCTYTSTGYPTLFWYVEYPSKPLQLLQRETMENSKNFGGGNIKDKNSPIVKYSVQVSDSAVYYCLLG
Textual label: It is known to participate in adaptive immune response (GO:0002250).
Start location: None **End location:** None

Sample B
Instruction: Given the protein details, identify the molecular function it performs.
Sequence: MMEFGLSWVFLVAIFKGVQCEVQLVESGEGLVQPGGSLRLSCAASGFTFSSYAMHWVRQAPGKGLEYVSAISSNGGSTYYADSVKGRFTISRDNSKNTLYLQMGSLRAEDMAVYYCAR
Textual label: It performs antigen binding (GO:0003823).
Start location: None **End location:** None

Sample C
Instruction: Based on the protein information, indicate the cellular component in which this protein is situated.
Sequence: MASQVVLSFALVVVLAVFVGQVDSCPSDCKCDYRSSQCRPANDDVHPNVCIDHYCVVMNLAKREQRPELSPGALDDSSEEKDNEASLA
Textual label: This protein is situated in extracellular region (GO:0005576).
Start location: None **End location:** None

Table 13: Validation loss decomposition for the Stage-1 pretrained model. Each task is reported with two subrows corresponding to ID and OOD-family splits.

Task	Split	Val Loss	CF	LM	Local LM	Local PTM	PTC	PTM
Cofactor	ID	13.77	0.00	6.28	0.00	0.00	3.60	3.88
	OOD	13.79	0.00	6.26	0.00	0.00	3.67	3.86
EC	ID	14.20	0.00	7.02	0.00	0.00	3.35	3.83
	OOD	14.17	0.00	7.03	0.00	0.00	3.31	3.83
Catalytic activity	ID	12.85	0.00	6.47	0.00	0.00	2.79	3.59
	OOD	12.80	0.00	6.47	0.00	0.00	2.77	3.57
Gene ontology	ID	14.10	0.00	7.53	0.00	0.00	2.85	3.72
	OOD	14.07	0.00	7.53	0.00	0.00	2.83	3.71
Sites	ID	25.85	0.10	8.24	8.19	10.75	3.96	4.05
	OOD	25.83	0.10	8.24	8.18	10.72	3.95	4.06
Regions	ID	25.42	0.11	8.26	8.05	10.25	4.10	3.79
	OOD	25.40	0.11	8.27	8.05	10.22	4.09	3.79

Table 14: Localization multiple-choice performance on the validation set. This evaluation is only applicable to fine-grained localization tasks.

Task	MC Acc (%)	MC MRR	MC Rec@5 (%)
Sites	8.63	0.285	51.02
Regions	29.51	0.472	69.50

E.6 EC Number

Sample A
Instruction: From the available protein data, determine its EC number.
Sequence: MAFGTEPTPTGLADPPIDDLMEHADSKYALAIFAAKRARQINSYFTQLNEGLLQNVGPLVEYQNQEKPLSIAFREINSGLLEETLGEDDLTEGN
Textual label: The Enzyme Commission number of this enzyme is 2.7.7.6. It belongs to transferases, which are enzymes transferring phosphorus-containing groups, with membership in nucleotidyltransferases.
Start location: None **End location:** None

Sample B
Instruction: Analyze the provided protein information to identify its EC number.
Sequence: MEIRVFQQSDFEEVILLWEHCDLLRPWNDPEMDIERKLNHDPELFLVAEVSGAIVGSVMGGYDGHRGSAYYLGVHPDFRGRGFANALISRLEKKLIARGCPKLNIMVREDNDAVIGMYEKLDYETQDTIMLGKRLIVDQEY
Textual label: The EC number corresponding to this enzyme is 2.3.1.-. It belongs to transferases, specifically the acyltransferases, which are enzymes transferring groups other than amino-acyl groups.
Start location: None **End location:** None

Sample C
Instruction: Based on the protein record provided, find the enzyme's EC number.
Sequence: MAVLTDDQVDAALPELGGWERADGALRRSVKFPAFLDGIEAVRRVAERAEAKDHHPDIDIRWRTVTFVLVTHSEGGITEKDLQMAREIDEILDR
Textual label: This enzyme is assigned EC number 4.2.1.96. It belongs to lyases, under the carbon-oxygen lyases, and classified as the hydro-lyases.
Start location: None **End location:** None

E.7 Regions

Sample A
Instruction: According to the protein information, could you describe what this annotated region corresponds to?
Sequence: MASSSVEFRCFVGGLAWATDSNSLEKAFSVYGEIVEAKIVSDRETGRSRGFGFVTFLEEEAMRSAIEAMNGHILDGRNITVNEAQQRGGGGGGGYNRGGGYGGRRDGGGFSRGGGGGYGGGGGGGYGGGRDRGYGGGGGYGGGRDSRGSGGGGSEGGWRN
Textual label: This annotated region corresponds to a modular protein domain (RRM).
Start location: 8 **End location:** 86

Sample B
Instruction: Describe the function of the annotated region in the protein record.
Sequence: MKKMMTFLKKAKVKAFTLVEMLVVLLIISVLFLLFVPNLTKQKEAVNDKGKAAVVKVVESQAELYSLEKNEDASLRKLQADGRITEEQAKAYKEYNDKNGGANRKVND
Textual label: This region corresponds to a membrane-spanning region (Helical).
Start location: 16 **End location:** 36

Sample C
Instruction: Can you determine, from the protein data, what functional element this region corresponds to?
Sequence: MLFRSAFVLLIVAFASACLVQHVQAKQIPVRRSLSTDASMSSAAGKLNRRWWFGFTGSLGKEPDNGQVQIKIIPDALIIKNPPANKDDLNKLIENLKRKHPRFKTVVMPTDPNGDVVIWE
Textual label: This region corresponds to a region of interest within the protein sequence (PID14 protease inhibitor domain).
Start location: 46 **End location:** 59

E.8 Regions with Embedded Location Information

Sample A
Instruction: Given the protein information, could you help me specify the functional region along the sequence?
Sequence: MMEFGLSWVFLVAIFKGVQCEVQLVESGEGLVQPGGSLRLSCAASGFTFSSYAMHWVRQAPGKGLEYVSAISSNGGSTYYADSVKGRFTISRDNSKNTLYLQMGSLRAEDMAVYYCAR
Textual label: Residues 21–45 in this protein correspond to a region of interest within the protein sequence (Framework-1).

Sample B
Instruction: Based on the provided protein data, could you tell me what this sequence region represents?
Sequence: MEFGLSWVFLAGILKGVQCEVQLVESGGGL VQPGGSLRLSCAASGFTFSNSDMNWARKAPGKGLEW VSGVSWNGSRTHYVDSVKRRFIISRDNSRNSLYLQKN RRRAEDMAVYYCVR
Textual label: The segment from residue 78 to residue 115 of this protein represents a region of interest within the protein sequence (Framework-3).

Sample C
Instruction: Given the annotated protein information, what does this sequence region denote?
Sequence: MGTRFLLALCLVLLVLGFEVQGAQLPQQD ERPSPALLSQVQESLSSYWESAKAAAQKLYEKTYLPA VDEKLRDLYSKSTAAMSTYTGIFTDQVLSVLKGEE
Textual label: This region (residues 66–74) denotes a region of interest within the protein sequence (Lipid binding).

E.9 Sites

Sample A
Instruction: Based on the provided record, what site is annotated in this protein?
Sequence: MKVTTIIAALLSVAVVDAQNNAQCRKLGL PCHATKTIPYVVKKGDTLTHIAHDIYKRKVGICDLAY TNHIGKNPNLIYAGQTLLIPTDCKTIDDGSCL
Textual label: The annotated site is associated with a site binding a chemical group such as a coenzyme or prosthetic group.
Start location: 77 **End location:** 77

Sample B
Instruction: According to the annotation, identify the type of site present in the protein.
Sequence: MSSTILVIHGPNLNLLGKREPEVYGHLTLD NINRQLIAQAEQASITLDTFQSNWEGAIVDRIHQAQTE GVKLIIINPAALTHTSVALRDALLGVAIPFIEVHLSNVH AREAFRHHSYLSDKAIGVICGLGAKGYSFALDYAIEKI QPSNPN
Textual label: This protein includes a site corresponding to amino acid residues directly involved in enzymatic activity (Proton acceptor).
Start location: 24 **End location:** 24

Sample C
Instruction: Based on the protein information, could you describe the annotation of the site?
Sequence: MSEIKLIVGLANPGAEYAQTRHNAGAWYV QELARICGVTLVPDSKYFGLTARAVLQGKDVRLLIPT TYMNLSGKAVGALANFFRITPEEILVAHDELDMPPGV AKFKLGGGHGGHNGLKDIIAKLANDKNFYRLRIGIGH PGDKNQVSGYVLGKAPAKEQELIEAAIDEAVRSTEIL YKENMVKAMTRLHSFKAE
Textual label: This site corresponds to a specific amino acid site of interest within the sequence (Stabilizes the basic form of H active site to accept a proton).
Start location: 95 **End location:** 95

E.10 Sites with Embedded Location Information

Sample A
Instruction: Given the protein information, could you specify the residue range that defines this site?
Sequence: MTTFLIKHKASGKFLHPKGGSSNPANDTNL VLHSDIHERMYFQFDVVDERWGYIKHAASGKIVHPL GGKADPPNETKLVLHQDRHDRALFAMDFFNDNIIHK AGKYVHPKGGSTNPPNETLTVMHGDKHGAMEFIFVS PKNKDKRVLVYV
Textual label: Residues 35–37 form a site associated with a site binding a chemical group such as a coenzyme or prosthetic group.

Sample B
Instruction: From the annotation, what site is identified in the protein?
Sequence: MIRRPPAVVCYICGREYGTKSISIHEPQCLK KWHNENNLLPKELRRPVPKKPEVRTITAKGFYDLDA LNEAAWTSAHSQLVPCNVCGRTFLPDRLIVHQRSCKP KAAK
Textual label: Residue 10 serves as a site binding a chemical group such as a coenzyme or prosthetic group.

Sample C
Instruction: Identify and describe the specific site annotated in the protein.
Sequence: MSHSVKIYDTCIGCTQCVRACPTDVLEMIP WDGCKAKQIASAPRTEDCVGCKRCESACPTDFLSVR VYLWHETTRSMGLAY
Textual label: The protein has a site at residue 14 that represents a site binding a chemical group such as a coenzyme or prosthetic group.

E.11 Additional Analysis of Stage-2 SFT Variants

To better understand the behavior of Stage-2 supervised fine-tuning (SFT), we compare four variants that differ in decoder adaptation and Stage-1 initialization. Specifically, **v0** (*Frozen/No-FG*) fine-tunes the Q-Former while keeping the LLM decoder frozen and does not use a fine-grained Stage-1 checkpoint; **v1** (*LoRA/No-FG*) additionally enables LoRA adaptation on the decoder; **v2** (*LoRA/FG-All*) initializes from a fine-grained Stage-1 checkpoint trained on all tasks; and **v3** (*LoRA/FG-Subset*) initializes from a Stage-1 checkpoint trained on only a subset of tasks.

Figure 7(a) shows that the largest improvement occurs from **v0** to **v1**, where the only major change is enabling LoRA on the LLM decoder. This suggests that, without decoder adaptation, Stage-2 SFT is primarily limited by a *decoder alignment bottleneck*: the Q-Former can be optimized, but a frozen decoder cannot fully translate its representations into task-appropriate outputs. Once LoRA is enabled, the model becomes substantially better at producing fluent, structured, and task-aware responses.

Comparing **v1** and **v2** isolates the effect of fine-grained Stage-1 pretraining under the same LoRA-based Stage-2 setup. Initializing from a fine-grained Stage-1 checkpoint trained on all tasks improves performance consistency and lowers degeneration, indicating that Stage-1 contributes a stronger and more stable representation prior for the Q-Former. In contrast, **v3**, which uses a Stage-1 checkpoint trained on only a subset of tasks, shows a less favorable trade-off: it retains strong fluency but exhibits weaker robustness, consistent with a more biased or less transferable intermediate repre-

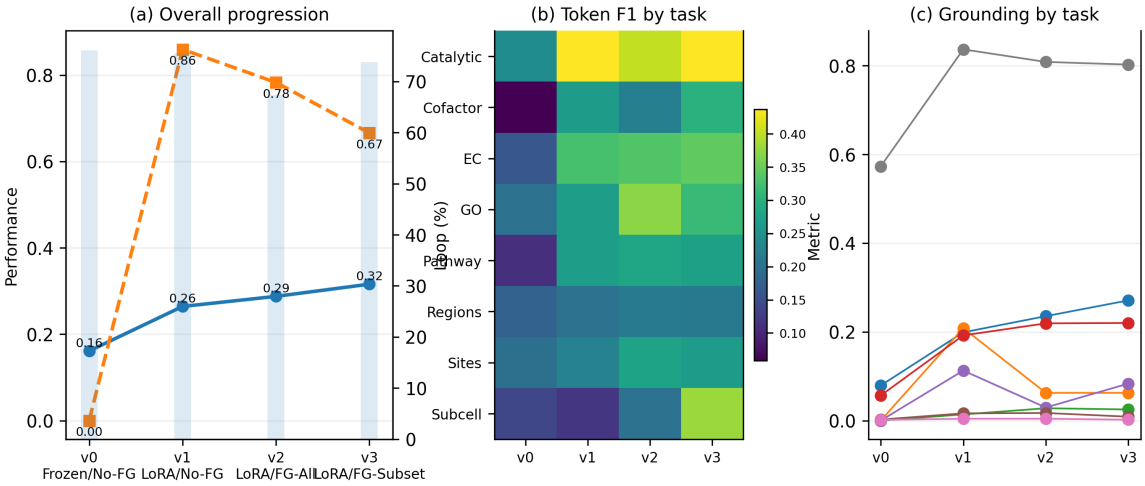


Figure 7: Additional analysis of Stage-2 SFT variants. **(a)** Overall progression across training configurations. Enabling decoder LoRA from v0 to v1 yields the largest gain in token-level performance, indicating that decoder adaptation is the primary bottleneck when the LLM is frozen. Moving from v1 to v2, i.e., introducing a fine-grained Stage-1 checkpoint trained on all tasks, further improves overall performance and reduces degeneration. Replacing this checkpoint with a subset-task Stage-1 initialization (v3) leads to slightly higher fluency but weaker robustness. **(b)** Token-level F1 by task. Improvements are broad but uneven, with stronger gains on semantic tasks such as catalytic activity, EC description, and gene ontology, while structurally grounded tasks remain comparatively difficult. **(c)** Task-specific grounding metrics. Despite clear improvements in fluency, grounding remains weak and highly task-dependent across all variants, revealing a persistent gap between task-conditioned language generation and mechanistic evidence localization.

sentation.

The task-level heatmap in Figure 7(b) further shows that these gains are not uniform. Semantic tasks, such as catalytic activity and gene ontology, benefit most from decoder adaptation and improved initialization, whereas the structurally grounded tasks remain challenging. This observation is reinforced by Figure 7(c), where task-specific grounding metrics remain relatively low and unstable across variants. In particular, improvements in token-level F1 do not translate proportionally into gains in EC tuple recovery or residue-level localization.

Overall, these results suggest a three-stage interpretation of the Stage-2 variants: **v0→v1** mainly reflects the benefit of decoder adaptation; **v1→v2** reflects the value of a stronger fine-grained Stage-1 prior; and **v2→v3** highlights the cost of reducing Stage-1 task coverage. Across all configurations, however, the dominant remaining limitation is unchanged: Stage-2 SFT improves *how the model expresses function* more than *how the model grounds function in biochemical evidence*.

F Post-Training RL Rewards for Evidence-Grounded EC Prediction

F.1 Task definition and notation

We define a post-training task for *enzyme commission (EC) number prediction* with explicit *evidence-grounded reasoning*. Each training instance corresponds to a single protein and is represented as

$$\mathcal{D}_i = \big(\mathrm{acc}_i,\ \mathbf{s}_i,\ \mathcal{T}_i,\ \mathcal{E}_i\big), \tag{28}$$

where:

- acc_i denotes the UniProt accession identifier;
- $\mathbf{s}_i$ denotes the amino-acid sequence;
- $\mathcal{T}_i$ denotes the set of functional targets, including EC numbers and optionally reaction identifiers;
- $\mathcal{E}_i$ denotes a set of curated evidence annotations supporting the functional assignment.

The post-training objective is conditional generation: given $(\mathrm{acc}_i, \mathbf{s}_i)$, the model must generate (i) a structured, evidence-grounded explanation and (ii) a final EC prediction consistent with $\mathcal{T}_i$.

F.2 Raw JSONL dataset format

The dataset is stored in JSON Lines (JSONL) format, with one protein per line. Each line contains a complete annotation record. A representative example is shown below:


```
{
  "accession": "O08679",
  "annotation_score": 5.0,
  "targets": {
   "ec_numbers": ["2.7.11.1", "2.7.11.26"],
   "rhea_ids": ["RHEA:17989", "RHEA:46608"]
  },
  "evidence": [
    {
      "type": "CATALYTIC_ACTIVITY_RHEA",
      "start": null,
      "end": null,
      "value": "RHEA:17989"
    },
    {
      "type": "ACTIVE_SITE",
      "start": 175,
      "end": 175,
      "value": "ACTIVE_SITE"
    },
    {
      "type": "BINDING_SITE",
      "start": 59,
      "end": 67,
      "value": "ChEBI:CHEBI:30616",
      "meta": {"ligand_name": "ATP"}
    }
  ],
  "sequence": "...",
  "sequence_length": 722
}
```


The `targets` field provides label-level supervision, while the `evidence` field provides curated grounding signals, including reaction-level evidence (Rhea), chemical entities (ChEBI), and residue- or span-level annotations (e.g., active or binding sites).

F.3 Demo as structured output and evidence anchoring

To illustrate the required output format and evidence anchoring scheme, we provide a representative example of a valid model completion for a protein annotated with EC 2.7.11.1 (protein kinase activity).

```
<work>
REACTION_HYPOTHESIS:
The protein catalyzes a phosphorylation
reaction transferring a phosphategroup from
ATP to a protein substrate
[E1|CAT:RHEA:17989].

DECISION_PATH:
Step1: EC class = 2
Step2: EC subclass = 2.7
Step3: EC sub-subclass = 2.7.11

COUNTERFACTUAL_CHECK:
If the ATP-binding region were absent, the
phosphorylation reaction would not be supported,
and the enzyme would not fall under EC class 2
[E2|FEAT:BINDING_SITE@59-67].

FINAL_CAUSE:
The presence of a conserved catalytic residue
and ATP-binding site supports classification
as a protein kinase [E3|FEAT:ACTIVE_SITE@175].
</work>

<answer>
EC = 2.7.11.1
</answer>
```

This example demonstrates several key properties enforced during post-training. First, the output strictly follows the required two-block structure (`<work>` and `<answer>`), with all mandatory section headers. Second, intermediate reasoning steps explicitly decompose the EC hierarchy (class, subclass, and sub-subclass) prior to the final prediction. Third, all evidence references are expressed using canonical anchors (e.g., `CAT:RHEA:17989`, `FEAT:BINDING_SITE@59-67`), which can be directly validated against the curated evidence set associated with the protein. Finally, the counterfactual section explicitly states how the prediction would change if a key piece of evidence were removed, discouraging post-hoc rationalization.

For each instance $\mathcal{D}_i$, the model is prompted to produce an output with a strict two-block schema:

1. A `<work>` block containing four mandatory sections: `REACTION_HYPOTHESIS`, `DECISION_PATH`, `COUNTERFACTUAL_CHECK`, and `FINAL_CAUSE`.

2. A `<answer>` block containing the predicted EC number or `NONE`.

Within the `<work>` block, evidence must be cited

using canonical anchors such as:

[CAT:RHEA:17989], [CHEBI:ChEBI:CHEBI:30616],

This explicit anchoring makes evidence usage enumerable and automatically verifiable.

F.4 Reward functions

Post-training is performed using a weighted combination of reward functions, each targeting a specific failure mode in structured, evidence-grounded generation.

Notation. Given a model completion $\hat{y}_i$ for instance $\mathcal{D}_i$, we extract: (i) a work block $\hat{w}_i$, (ii) an answer block $\hat{a}_i$, (iii) a predicted EC $\widehat{\mathrm{EC}}_i$, and (iv) a set of predicted evidence anchors $\widehat{\mathcal{A}}_i$.

Schema compliance. This reward enforces adherence to the required output structure.

$$R_{\text{schema}}(\hat{y}_i) = \begin{cases} 1, & \text{if required headers are present,} \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

Hierarchical EC correctness. We assign partial credit based on agreement with the EC hierarchy:

$$R_{\text{EC}}(\hat{y}_i) = \max_{\text{EC}^\star \in \mathcal{T}_i^{\text{EC}}} \sum_{k=1}^{4} w_k \, \mathbb{I}[c_k = c_k^\star], \tag{30}$$

where $(c_1, \ldots, c_4)$ denote EC levels and $(w_1, w_2, w_3, w_4) = (0.2, 0.2, 0.2, 0.4)$.

Evidence-anchor precision. This reward penalizes hallucinated evidence:

$$R_{\text{ev}}(\hat{y}_i) = \frac{|\widehat{\mathcal{A}}_i \cap \mathcal{A}_i^{\text{allowed}}|}{\max(1, |\widehat{\mathcal{A}}_i|)}. \tag{31}$$

Decision-path consistency. We measure consistency between intermediate reasoning steps and the predicted EC:

$$R_{\text{dp}}(\hat{y}_i) = \frac{1}{3} \sum_{k=1}^{3} \mathbb{I}\big[\text{Step}_k \text{ matches } \widehat{\mathrm{EC}}_i\big]. \tag{32}$$

Overall reward. The final reward is a weighted sum:

$$R(\hat{y}_i) = \lambda_{\text{schema}} R_{\text{schema}} + \lambda_{\text{EC}} R_{\text{EC}} + \lambda_{\text{ev}} R_{\text{ev}} + \lambda_{\text{dp}} R_{\text{dp}}. \tag{33}$$