ProtQueSt: Query-Conditioned Retrieval-Augmented Generation for Protein Function Annotation

Linrui Ma [*1] **Yiwei Liang** [*1] **Yishu Yu** [**1] **Chuhan Joyce Qi** [**1]

Abstract

Protein function prediction from sequence is inherently query-dependent: the same protein may be characterized by its catalytic activity, domain architecture, or cellular localization depending on the biological question. Prior work has shown that large language models consistently underperform retrieval-based methods on this task, yet simple retrieval transfers annotations from embedding-space neighbors without adapting to the query. We introduce **ProtQueSt**, a retrieval-augmented framework that pairs a structure-aware retriever with a query-conditioned contrastive retriever that aligns protein and annotation representations via Feature-wise Linear Modulation (FiLM). FiLM conditioning and query-pooled negative sampling prove jointly essential, as neither alone improves over the structural baseline. ProtQueSt achieves the highest Entity-BLEU (48.79, +37% over RAPM) and LLM-as-a-judge score reported on Prot-Inst-OOD. This result supports reframing text-based protein understanding as a query-conditioned retrieval problem rather than a single fixed sequence-to-text mapping.

1. Introduction

Accurate protein functional annotation is fundamental to biomedicine and drug discovery. Recent protein-language models such as Evolla (Zhou et al., 2025), ProteinGPT (Xiao et al., 2024), and BioReason-Pro (Fallahpour et al., 2026) have demonstrated strong capabilities, but they typically require predicted 3D structures, interaction networks, or Gene Ontology terms beyond the raw sequence. For novel or orphan proteins, amino acid sequence is often the only reliable input, making sequence-based prediction a practically critical setting. Meanwhile, LLMs consistently underperform retrieval-based methods on sequence-only benchmarks (Wu et al., 2025), a finding our own results confirm (Table 1).

While this sequence-only constraint poses a challenge for current models, it is theoretically sufficient for functional annotation. As Anfinsen's dogma (Anfinsen, 1973) states, protein functionality arises from its amino acid sequence. However, high-quality annotation relies on *text-based protein sequence understanding*: the ability to translate a raw amino acid sequence directly into the nuanced natural language concepts that define its actual function. While prior work like Prot2Text-V2 (Fei et al., 2025) advances this by reformulating function prediction as free-text generation, it assumes a single fixed description. In practice, a protein's function is rarely context-independent; the same sequence may be queried for catalytic activity, domain architecture, or cellular localization depending on the specific context, making the ability to yield distinct descriptions essential.

Therefore, we rethink text-based protein sequence understanding as a query-driven ability: dynamically mapping sequences to biological entities conditioned on the query.

While retrieval-based methods outperform LLMs on protein function prediction, they do not achieve true sequence understanding. Systems based on sequence alignment or embedding similarity transfer annotations from similar proteins without adapting to query. To move beyond this nearest-neighbor transfer, the retriever must learn query-conditioned alignment between protein sequences and annotations. This draws on the broader idea behind conditional representation learning (Liu et al., 2025) and feature modulation (Perez et al., 2018), where the same input amplifies different features depending on the conditioning signal. However, any conditioning module can only emphasize signals already present in the base embedding.

We introduce ProtQueSt (Protein Query-conditioned Structure-aware Retrieval-augmented Framework) with the following contributions:

- We reapproach protein function annotation as text-based protein sequence understanding and reframe the latter as a query-driven ability, where the functional meaning of a sequence is conditioned on the specific

[*]Equal contribution (first authorship) [**]Equal contribution (second authorship) [1]Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Correspondence to: Linrui Ma <linrui@mit.edu>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

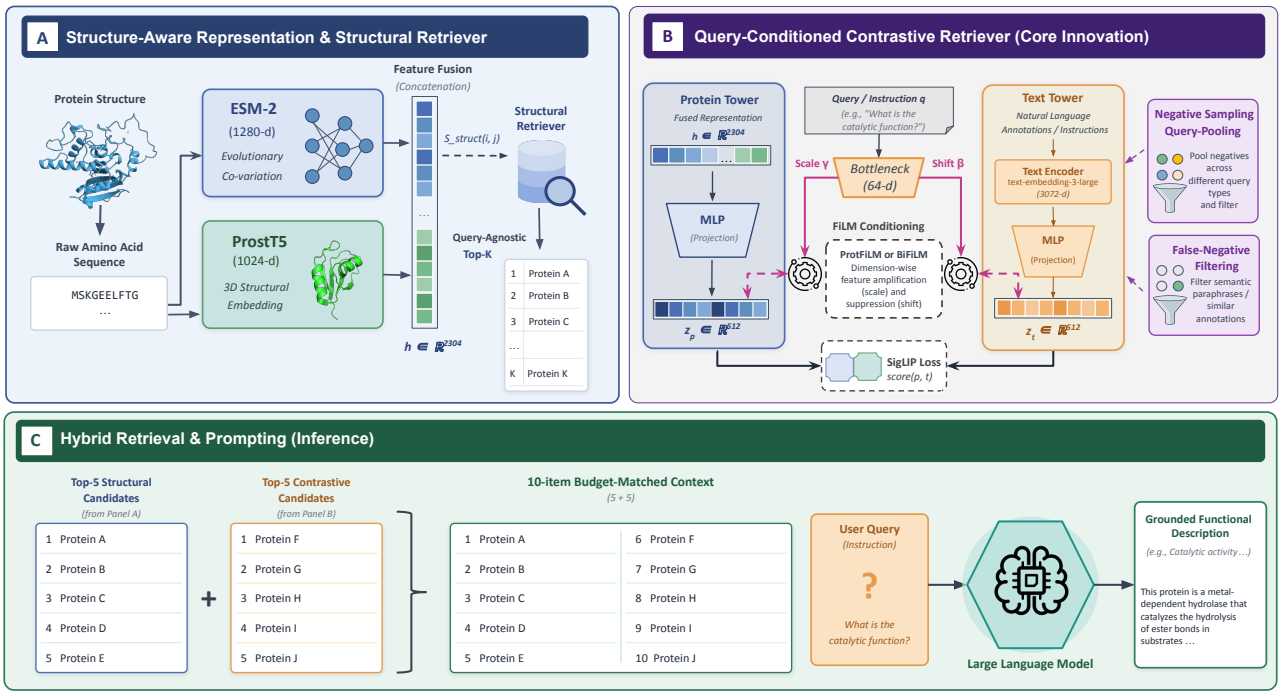

Figure 1. ProtQueSt Architecture. (A) A structure-aware retriever fuses ESM-2 and ProstT5 embeddings for query-independent top-K candidate retrieval. (B) A query-conditioned contrastive retriever uses FiLM conditioning and query-pooled negative sampling to align protein-text representations via SigLIP loss. (C) Hybrid retrieval is combined for grounded functional description generation by an LLM.

biological question asked.

- We realize this through a contrastive retriever that applies Feature-wise Linear Modulation (FiLM) to select query-relevant features from frozen protein embeddings, trained with query-pooled negative sampling that provides truly query-specific contrastive signal.
- We present **ProtQueSt**, a hybrid retrieval system that combines structure-aware and query-conditioned channels to achieve state-of-the-art results on the Prot-Inst-OOD benchmark.

2. Background and Related Work

LLMs for protein function prediction. An extensive body of work adapts LLMs for protein function prediction via pretraining or instruction tuning. BioT5+ (Pei et al., 2024) integrates International Union of Pure and Applied Chemistry (IUPAC) nomenclature and multi-task tuning to align representations of small molecules and proteins with natural language. Similarly, InstructProtein (Wang et al., 2024) aligns protein sequences and text through instruction tuning grounded in knowledge graphs, supporting bidirectional sequence-to-text generation. BioReason-Pro (Fallahpour et al., 2026) extends this line with reinforcement learning (RL)-based training over curated biological knowledge to improve protein function reasoning.

Retrieval-augmented protein annotation. RAPM (Wu et al., 2025) pioneered retrieval-augmented protein function description, using a dual-key system that combines MMseqs2 sequence alignment and ESM-2 embedding similarity to retrieve neighbor proteins and prepend their annotations to the LLM prompt. Evolla-10B (Zhou et al., 2025) takes an alternative route, internalizing protein-text alignment using a 10B-parameter protein-language fusion model trained end-to-end on large-scale joint corpora, removing the explicit retrieval step at the cost of a heavy parameter footprint. Beyond functional annotation, PoET-2 (Truong Jr & Bepler, 2025) and Protriever (Weitzman et al., 2025) use homologous retrieval for fitness and variant-effect prediction.

Contrastive protein-text alignment. A separate approach casts protein-text grounding as a contrastive learning problem. ProteinCLIP (Wu et al., 2024) and ProtST (Xu et al., 2023) both align protein sequence representations with textual descriptions via contrastive objectives over *(sequence, text)* pairs, producing a shared embedding space suited for retrieval and zero-shot transfer. Prot2Text-V2 (Fei et al., 2025) further extends this recipe with a stronger encoder stack and a contrastive auxiliary objective for protein-to-text generation.

3. Methods

Our system retrieves protein functional annotations through two complementary channels, a *structural retriever* and a *contrastive metadata retriever*, whose outputs are merged and passed to an LLM for retrieval-augmented generation.

3.1. Structure-Aware Protein Representation and Structural Retriever

Protein function is largely determined by 3D structure, so a representation encoding structural signals should capture additional functional information beyond sequence data. A combination of both structural and sequence information would therefore convey a richer representation. At the same time, any downstream conditioning module can only expose signals already present in the base embedding (Section 1). To raise this representational ceiling, we fuse frozen ESM-2 (Lin et al., 2023) (1,280-d, evolutionary co-variation) with ProstT5 (Heinzinger et al., 2023) (1,024-d, 3D fold information predicted from sequence) to yield $\mathbf{h} \in \mathbb{R}^{2304}$, providing both the structural retriever and the contrastive module with a structure-aware input.

The structural retriever scores each candidate protein j against query protein i with a weighted cosine fusion:

$$s_{\text{struct}}(i,j) = \alpha \cdot \cos(\mathbf{p}_i, \mathbf{p}_j) + (1-\alpha) \cdot \cos(\mathbf{e}_i, \mathbf{e}_j), \quad (1)$$

where $\mathbf{p}$ and $\mathbf{e}$ are L2-normalized ProstT5 and ESM-2 embeddings, and α=0.7 (sweep in Appendix K.1). This replaces RAPM's dual-key pipeline with a unified dense score. The top-K candidates are retrieved from the full training corpus, and their curated annotations are retained.

3.2. Query-Conditioned Contrastive Retriever

The structural retriever is query-independent by design: the same protein always yields the same neighbors regardless of whether the user asks about catalytic activity or cellular localization. The contrastive channel closes this gap by learning a query-conditioned compatibility function. On the protein side, we reuse the same fused ESM-2 and ProstT5 representation ($\mathbf{h} \in \mathbb{R}^{2304}$). On the text side, following the ProteinCLIP (Wu et al., 2024) paradigm of aligning protein representations with natural-language descriptions, we use the OpenAI TEXT-EMBEDDING-3-LARGE model to generate 3072-d embeddings as the text representation for both annotation descriptions and instruction queries.

Architecture. We adopt a two-tower contrastive model with lightweight two-layer MLP adapters that project both protein and text embeddings to a shared 512-d space (full specification in Appendix B). The compatibility score is cosine similarity scaled by a learnable temperature τ.

FiLM query conditioning. By design, the base protein adapter yields a fixed embedding $\mathbf{z}_p$, forcing it to serve all query types during direct contrastive training. Yet, the fused embedding encodes a rich set of functional signals across its dimensions. We apply Feature-wise Linear Modulation (FiLM) (Perez et al., 2018) as a lightweight conditioning mechanism that selectively emphasizes the dimensions relevant to a given query type.

In our implementation, the instruction text embedding $\mathbf{q} \in \mathbb{R}^{3072}$ is first compressed to a 64-d bottleneck $\tilde{\mathbf{q}} = \text{GELU}(\text{LayerNorm}(\mathbf{q})\mathbf{W}_q)$. A 64-d bottleneck suffices because protein function queries span only a handful of distinct question types, so this compression discards noise while preserving coarse query identity. From $\tilde{\mathbf{q}}$, two linear maps produce a scale vector and a shift vector, each in $\mathbb{R}^{512}$, which modulate the adapter embedding $\mathbf{z} \in \{\mathbf{z}_p, \mathbf{z}_t\}$ dimension-by-dimension:

$$\mathbf{z}_{\text{out}}[d] = \mathbf{z}[d] \cdot \big(1 + \text{scale}(\tilde{\mathbf{q}})[d]\big) + \text{shift}(\tilde{\mathbf{q}})[d]. \quad (2)$$

Initialization and training details are in Appendix C.

We explore two variants: **ProtFiLM**, which query conditions only the protein encoder, and **BiFiLM**, which extends the query condition to both encoders (the query additionally modulates which text features are matched). ProtFiLM adds ∼6.8M parameters and BiFiLM ∼7.1M; all base encoder weights remain frozen.

Negative sources and query pooling. Each protein's training negatives come from two sources: *in-batch negatives* (gold annotations of other proteins in the same mini-batch) and *pool negatives* (annotations of the protein's structural retrieval candidates, providing harder comparisons; details in Appendix C). In a multi-task batch, negatives from different query types are trivially distinguishable. Query pooling addresses this: for each anchor with instruction $\mathbf{q}_i$, we mask out any candidate whose instruction satisfies $\cos(\mathbf{q}_i, \mathbf{q}_j) < \tau_{\text{query}}$. This ensures that only candidates instructively similar to the anchor provide gradient signals.

False-negative filtering. A separate problem arises on the *annotation* side: in-batch annotations may be paraphrases of the anchor's gold that are equally correct. Penalizing the model for scoring these highly corrupts the gradient. We mask any in-batch candidate j for which $\cos(\mathbf{t}_j, \mathbf{t}_{\text{gold}}) \geq \tau_{\text{fn}}$, where $\mathbf{t}$ denotes the annotation embedding. Pool negatives are separately pre-filtered at a fixed threshold during data preparation (Appendix C).

Training objective. Because query pooling and false-negative filtering mask different subsets per anchor, the effective negative set varies across proteins in the batch. This is incompatible with InfoNCE, which requires a shared softmax denominator. We therefore optimize a pairwise sigmoid loss (SigLIP) (Zhai et al., 2023):

$$\mathcal{L}_{ij} = -\log \sigma\big(y_{ij}(\text{score}(p_i, t_j) + b)\big) \quad (3)$$

Details and a Soft-SigLIP variant are in Appendix C.

3.3. Hybrid Retrieval and Prompting

At inference, we retrieve the top-5 candidates from each channel (structural and contrastive) and concatenate them into a 10-item context, budget-matched to the structural-only baseline for fair comparison. The contrastive channel applies the same query mask used during training (τ_{query}=0.65) to filter the annotation pool. Both source lists are presented with source labels in a structured prompt to Claude Sonnet 4.6, which generates the final functional description (prompt template in Figure 3; generation configuration in Appendix D).

4. Experiments and Results

4.1. Experimental Setup

Dataset. We evaluate all models on Prot-Inst-OOD (Wu et al., 2025), the out-of-distribution evaluation split of a UniProt-derived protein-instruction corpus (Appendix A).

Metrics. Our primary lexical metric is Entity-BLEU (Wu et al., 2025), which restricts BLEU to tokens present in the metadata so the score reflects biologically-relevant content. We also report ROUGE-L (Lin, 2004) and METEOR (Banerjee & Lavie, 2005), and an LLM-as-a-judge metric (full rubric in Appendix E). We also report Recall@10 for retrieval (Appendix F). ROUGE-L is reported in the appendix.

Our model. The ProtQueSt entry in Table 1 uses ProtFiLM with SigLIP loss, query pooling at τ_{query}=0.65, false-negative filtering at τ_{fn}=0.70, and hybrid 5+5 merging. This configuration is selected by deterministic Recall@10 on the n=256 ablation subset (Section 4.3; Appendix I), then evaluated on the full 14,503-sample benchmark. Only-Structure uses the same structural retriever and LLM generator but retrieves all 10 candidates from the structural channel. Both models use Claude Sonnet 4.6 as the generator.

Baselines. We compare against 4 families in the protein-text prediction regime (Table 1; implementation details and fine-tuning hyperparameters in Appendix L): *LLM-only (3-shot)* prompts frontier general-purpose models without protein-specific input; *domain-specific pretrained* uses protein-text foundation models with end-to-end alignment; *fine-tuned on the OOD train split* provides direct supervision on the target task; and *retrieval-based* transfers annotations from neighbor proteins, the family that ProtQueSt belongs to.

4.2. Results

ProtQueSt achieves the highest Entity-BLEU (48.79) and Judge-F (6.40) across all models (Table 1), improving over the strongest prior retrieval baseline, RAPM with Sonnet 4.6, by 13.23 Entity-BLEU points and 0.34 Judge-F points. The gain over Only-Structure (44.26 Entity-BLEU, 6.19 Judge-F) isolates the value of the query-conditioned contrastive channel: the structural retriever already provides strong neighbors, but the contrastive retriever adds candidates that are better matched to the requested biological aspect.

Table 1. Evaluation for ProtQueSt and baselines on Prot-Inst-OOD. Lexical metrics: Entity-BLEU (E-BL) and METEOR. LLM-based metric: LLM-as-judge Final score (Judge-F, 0–10 scale). Full-dataset results in Appendix H.

Method	E-BL	METEOR	Judge-F
LLM-only (3-shot)			
Llama-3.1-8B	3.21	18.70	3.31
Qwen3-8B	1.09	24.27	3.30
Sonnet 4.6	2.96	28.96	4.24
GPT-5.4	0.30	17.84	3.88
Fine-tuned on OOD training data			
Qwen3-8B-FT	2.20	34.72	4.07
BioT5+	4.37	32.64	4.49
InstructProtein	5.96	31.68	3.87
Prot2Text-V2	19.35	49.81	6.23
Domain-specific pretrained			
Evolla-10B-DPO	10.54	19.05	3.96
Retrieval-based			
MMseqs2	21.91	44.52	5.60
RAPM with GPT-4.1	24.11	37.30	5.89
RAPM with Sonnet 4.6	35.56	42.48	6.06
Only-Structure	44.26	46.42	6.19
ProtQueSt	**48.79**	**50.84**	**6.40**

The lexical and judge metrics are complementary. Entity-BLEU rewards recovering curated bio-entities, while Judge-F also captures plausibility, specificity, and precision under an LLM-as-judge rubric. ProtQueSt leads on both, suggesting that the contrastive channel improves biological coverage without merely adding unsupported details. Bootstrap 95% CIs are tight (Entity-BLEU $[48.44, 50.16]$, Judge-F $[6.31, 6.49]$; full per-task intervals in Appendix Table 27). Per-task analysis further shows that the contrastive channel contributes complementary candidates that the structural retriever misses, with the largest relative Recall@10 gains on catalytic activity and domain/motif queries (Appendix F).

4.3. Ablation Studies

We conduct ablations to isolate the effect of each design choice. Reported metrics include Entity-BLEU (end-to-end) and Recall@10 (deterministic retrieval quality). Full tables with all model variants are in Appendix I.

Query pooling and FiLM are jointly essential (Table 2). Without query pooling, ProtFiLM *degrades* below the structural-only baseline (Recall@10: 0.229 vs. 0.231). With query pooling (τ_{query}=0.65), the same model jumps to 0.324, a 40% improvement. Query pooling alone (no FiLM) yields a smaller gain (0.256). The two mechanisms are complementary: query pooling provides clean training signal by restricting comparisons to the same question type,

Table 2. Ablation study (n=256 per task, masked contrastive pool). R@10: Recall@10; E-BL: Entity-BLEU. All use hybrid 5+5 merging except Struct-only (10 structural).

FiLM	Q-pool	R@10	E-BL
Struct-only baseline		0.231	44.95
None	None	0.238	44.24
ProtFiLM	None	0.229	42.49
BiFiLM	None	0.264	44.19
None	0.65	0.256	45.87
ProtFiLM	0.65	**0.324**	**47.46**
BiFiLM	0.65	0.303	45.96

Table 3. False-negative threshold ablation (Recall@10, masked, hybrid 5+5, query-pooled). Best per row in bold.

FiLM	Loss	None	τ_{fn}=0.65	τ_{fn}=0.70
BiFiLM	SigLIP	0.285	**0.313**	0.303
ProtFiLM	SigLIP	0.310	0.308	**0.324**

and FiLM gives the adapter the capacity to act on that signal by selecting query-relevant feature channels (Appendix Figure 6). Neither alone is sufficient.

False-negative filtering. Removing the filter degrades Recall@10 of both BiFiLM and ProtFiLM methods (Table 3), confirming that unmasked paraphrases corrupt the training signal. ProtFiLM also improves with filtering at τ_{fn}=0.70, though the effect is smaller. The asymmetry aligns with the architectural difference: BiFiLM applies query conditioning to the text embedding, amplifying annotation features through the modulation; ProtFiLM leaves the text embedding unconditional and is less exposed to false-negative paraphrases.

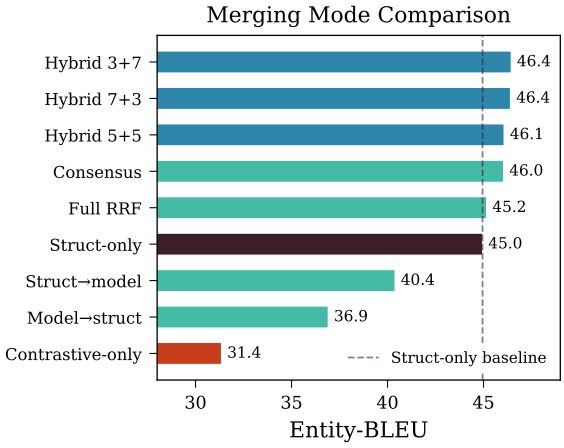

Figure 2. Merging mode sweep with BiFiLM + SigLIP (τ_{query}=0.65, n=256). All modes use budget K=10. Dashed line marks the structural-only baseline. Hybrid modes (blue) cluster at the top; two-stage re-ranking degrades performance.

Retrieval merging. Hybrid merging is robust: 5+5, 3+7, and 7+3 splits perform within 0.4 Entity-BLEU points of each other on the ablation subset, while more complex reciprocal-rank and two-stage reranking variants underperform (Figure 2). We use 5+5 because it is budget-matched, simple, and gives both retrieval sources equal capacity.

The query-conditioned design also remains effective when swapping to a fully open-source encoder and generator (Appendix Table 20; qualitative analysis in Appendix J).

5. Conclusion and Future Directions

Text-based protein sequence understanding should be query-conditioned: the functional meaning of a sequence emerges from the interaction between the sequence and a specific biological question. ProtQueSt demonstrates this by adding a contrastive channel that learns query-dependent protein-text correspondence through FiLM conditioning, complementing the structural retriever. Our ablations confirm that this requires both query-conditioned representation (FiLM) and query-conditioned training signal (query pooling), as neither alone improves over structural retrieval.

Our approach currently relies on a single benchmark and a fixed set of frozen encoders, which limits the scope of our evaluation. Future work could extend the query-conditioned paradigm to richer conditioning signals such as compositional multi-axis queries and multi-turn refinement. Advances in protein encoders would directly raise the representational ceiling that conditioning can expose, and additional benchmarking would test how broadly the paradigm generalizes. Computational cost analysis, broader impact, and additional limitations are discussed in Appendices K and M.

References

Anfinsen, C. B. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973. doi: 10.1126/science.181.4096.223. URL https://www.science.org/doi/abs/10.1126/science.181.4096.223.

Anthropic. Messages API reference, 2025. URL https://docs.anthropic.com/claude/reference/messages. Temperature defaults to 1.0; recommended closer to 1.0 for generative tasks.

Banerjee, S. and Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Goldstein, J., Lavie, A., Lin, C.-Y., and Voss, C. (eds.), *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909/.

Fallahpour, A., Seyed-Ahmadi, A., Idehpour, P., Ibrahim, O., Gupta, P., Naimer, J., Zhu, K., Shah, A., Ma, S., Adduri, A., Gülogluu, T., Liu, N., Cui, H., Jain, A., de Castro, M., Fallahpour, A., Cembellin-Prieto, A., Stiles, J. S., Nemčko, F., Nevue, A. A., Moon, H. C., Sosnick, L., Markham, O., Duan, H., Lee, M. Y. Y., Salvador, A. F. M., Maddison, C. J., Thaiss, C. A., Ricci-Tam, C., Plosky, B. S., Burke, D. P., Hsu, P. D., Goodarzi, H., and Wang, B. Bioreason-pro: Advancing protein function prediction with multimodal biological reasoning. *bioRxiv*, 2026. doi: 10.64898/2026.03.19.712954. URL https://www.biorxiv.org/content/early/2026/03/20/2026.03.19.712954.

Fei, X., Chatzianastasis, M., Carneiro, S. A., Abdine, H., Petalidis, L. P., and Vazirgiannis, M. Prot2text-v2: Protein function prediction with multimodal contrastive alignment, 2025. URL https://arxiv.org/abs/2505.11194.

Heinzinger, M., Weissenow, K., Stärk, H., and Rost, B. ProstT5: Bilingual language model for protein sequence and structure. *bioRxiv*, 2023.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *Science*, 379(6637), 2023.

Liu, H., Sun, C., Hu, P., Li, Y., and Peng, X. Conditional representation learning for customized tasks, 2025. URL https://arxiv.org/abs/2510.04564.

Pei, Q., Wu, L., Gao, K., Liang, X., Fang, Y., Zhu, J., Xie, S., Qin, T., and Yan, R. BioT5+: Towards generalized biological understanding with IUPAC integration and multi-task tuning. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1216–1240, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.71. URL https://aclanthology.org/2024.findings-acl.71/.

Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. FiLM: Visual reasoning with a general conditioning layer. *AAAI Conference on Artificial Intelligence*, 2018.

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017. doi: 10.1038/nbt.3988. URL https://www.nature.com/articles/nbt.3988.

Truong Jr, T. F. and Bepler, T. Understanding protein function with a multimodal retrieval-augmented foundation model. *arXiv preprint arXiv:2508.04724*, 2025.

Wang, Z. et al. InstructProtein: Aligning human and protein language via knowledge instruction. *ACL*, 2024.

Weitzman, R., Groth, P. M., Van Niekerk, L., Otani, A., Gal, Y., Marks, D. S., and Notin, P. Protriever: End-to-end differentiable protein homology search for fitness prediction. *arXiv preprint arXiv:2506.08954*, 2025.

Wu, J., Liu, Z., Cao, H., Hao, L., Feng, B., Shu, Z., Yu, K., Yuan, L., and Li, Y. Rethinking text-based protein understanding: Retrieval or LLM? In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23726–23746, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1211. URL https://aclanthology.org/2025.emnlp-main.1211/.

Wu, K. et al. ProteinCLIP: Contrastive learning of protein representations with text. *bioRxiv*, 2024. doi: 10.1101/2024.05.14.594226.

Xiao, Y., Sun, E., Jin, Y., Wang, Q., and Wang, W. Proteingpt: Multimodal llm for protein property prediction and structure understanding. *arXiv preprint arXiv:2408.11363*, 2024.

Xu, M., Yuan, X., Miret, S., and Tang, J. Protst: Multimodality learning of protein sequences and biomedical texts. In *International Conference on Machine Learning (ICML)*. PMLR, 2023.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *International Conference on Computer Vision (ICCV)*, 2023.

Zhou, X., Han, C., Zhang, Y., Su, J., Zhuang, K., Jiang, S., Yuan, Z., Zheng, W., Dai, F., Zhou, Y., Tao, Y., Wu, D., and Yuan, F. Decoding the molecular language of proteins with evolla. *bioRxiv*, 2025. doi: 10.1101/2025.01.05.630192. URL https://www.biorxiv.org/content/early/2025/01/09/2025.01.05.630192.

A. Dataset Statistics and Characterization

A.1. Per-Task Sample Counts

Table 4 summarizes the Prot-Inst-OOD benchmark composition. The full dataset contains 299,029 protein–annotation pairs: 284,526 in the retrieval pool (train split) and 14,503 in the OOD test set. Of the pool, 270,300 serve as contrastive training anchors; the remaining 14,226 (5% per task, *dev_frac*=0.05, *split_seed*=42) are held out for early-stopping validation. Dev proteins remain in the retrieval pool but are not used as contrastive training anchors.

Table 4. Prot-Inst-OOD dataset split statistics.

Task	Pool	Train	Dev	Test
Catalytic activity	51,187	48,628	2,559	1,987
Domain/motif	42,368	40,250	2,118	2,732
General function	82,275	78,161	4,114	4,297
Protein function	108,696	103,261	5,435	5,487
Total	**284,526**	**270,300**	**14,226**	**14,503**

Prot-Inst-OOD is derived from a UniProt-based protein–description corpus following the out-of-distribution split protocol introduced by Wu et al. (2025). The benchmark is constructed in three steps: (1) all sequences are clustered using MMseqs2 (Steinegger & Söding, 2017) with no minimum sequence identity threshold, (2) clusters are randomly partitioned into training and test sets at an 80/20 ratio, and (3) any test sample whose annotation can be directly retrieved from the training set is reassigned to training, repeated twice to minimize label leakage. Crucially, OODness is defined operationally by this leakage-elimination procedure rather than by an explicit sequence identity cutoff or novel GO term constraint, meaning that test proteins are guaranteed to lack directly retrievable training annotations but may still share sequence similarity with training examples.

A.2. Sequence Length Distribution

Table 5. Protein sequence length statistics per task (test set).

Task	Min length	Max length
Catalytic activity	12	768
Domain/motif	37	768
General function	11	768
Protein function	11	768

No proteins in the test set exceed the ESM-2 or ProstT5 maximum input length of 1,024 tokens (one amino acid corresponds to one token in ESM-2's vocabulary). The maximum observed sequence length across all tasks is 768 residues (Table 5), so truncation is not an issue for this dataset.

A.3. Query Templates and Instruction Embedding Analysis

Each task draws from a pool of paraphrased natural-language instruction templates: 5 for catalytic activity, 10 for domain/motif, 7 for general function, and 38 for protein function (60 total). Table 6 shows one representative template per task. Instructions within a task are semantically coherent (intra-task mean cosine similarity 0.74–0.87), while covering diverse phrasing that helps the model generalize across surface-level variation.

Table 6. Representative instruction templates for the four Prot-Inst-OOD tasks. Each task has multiple paraphrased variants (5, 10, 7, and 38 respectively).

Task	#	Example instruction template
Catalytic activity	5	"Please evaluate the following protein sequence and provide an explanation of the enzyme's catalytic activity, including the chemical reaction it facilitates: "
Domain/motif	10	"Please conduct a domain/motif search on the given protein sequence and provide your findings. The sequence is: "
General function	7	"Assess the following protein sequence and provide a brief report on its primary characteristics: "
Protein function	38	"Considering the protein sequence below, predict its biological function by examining its structural features and comparing it to functionally characterized proteins: "

To validate that the query-pooling threshold τ_{query}=0.65 cleanly separates task types, we report the mean pairwise cosine similarity between all instruction embeddings of each task pair, computed with TEXT-EMBEDDING-3-LARGE (Table 7). All six cross-task means fall below 0.66, with catalytic activity vs. domain/motif the most distinct (0.551) and general function vs. protein function the closest (0.656). Individual cross-task pairs can exceed 0.65 (e.g., 58.6% of general–protein pairs), reflecting genuine semantic overlap between these broader functional queries. This means query pooling treats some instruction pairs from these task categories as interchangeable during negative sampling, which is acceptable given their overlapping annotation space.

A.4. Query-Filtered Pool Sizes

Query filtering at τ_{query}=0.65 restricts which pool candidates provide gradient for each anchor during training. It does not remove proteins from the training set. Because each task has multiple instruction templates, the effective pool size varies per anchor depending on which instruction it carries. Table 8 reports the mean number of eligible pool

Table 7. Mean pairwise cosine similarity of instruction embeddings across task pairs (TEXT-EMBEDDING-3-LARGE). Off-diagonal: averaged over all template pairs between the two tasks. Diagonal: mean intra-task similarity (excluding self-pairs).

	Cat. act.	Dom./mot.	Gen. func.	Prot. func.
Cat. act.	0.867	0.551	0.600	0.595
Dom./mot.	0.551	0.793	0.654	0.639
Gen. func.	0.600	0.654	0.780	0.656
Prot. func.	0.595	0.639	0.656	0.741

candidates per task, averaged over all anchors (and thus all instruction variants).

Table 8. Effective contrastive pool sizes after query filtering (τ_{query}=0.65). Mean ± std computed over all anchors in each task; min/max reflect variation across the different instruction templates.

Task	Mean	Std	Min	Max	% of pool
Cat. act.	69,194	20,442	51,187	109,005	24.3
Dom./mot.	128,333	45,365	43,930	196,590	45.1
Gen. func.	177,747	38,507	126,626	241,421	62.5
Prot. func.	170,654	35,482	33,373	258,202	60.0

Catalytic activity retains only 24.3% of the pool on average, compared to 62.5% for general function, a 2.6× disparity. The wide min–max ranges (driven by the diversity of instruction templates within each task) show that some instruction variants are much more selective than others; for example, certain protein-function instructions retain as few as 33 K candidates while others retain 258 K. Despite this variability, our per-task results (Section F.3) suggest that intrinsic task difficulty (precise reaction chemistry vs. broad functional descriptions) is likely the dominant factor in performance differences.

A.5. Gold Reference Structure

Each sample has two gold components: (1) a compact *metadata* string containing GO-style annotations (e.g., "*lipid binding*" or "*plastid, ribonucleoprotein complex, ribosome — rRNA binding*"), and (2) a longer natural-language *description*. The contrastive retriever is trained to match protein embeddings with annotation metadata embeddings. The LLM generator produces free-text descriptions. Entity-BLEU bridges the two by evaluating whether biologically relevant entities from the metadata appear in the generated text.

B. Model Architecture Details

B.1. Frozen Encoder Details

ESM-2. We use FACEBOOK/ESM2_T33_650M_UR50D (650M parameters). The protein representation is extracted via the CLS token, yielding a 1,280-d embedding that captures evolutionary co-variation patterns.

ProstT5. A 1,024-d embedding is obtained via masked mean pooling over the encoder's *last_hidden_state*, weighted by the attention mask. ProstT5 provides fold-aware representations derived from sequence via structure-conditioned language modeling; "structural" in our context refers to these learned fold representations, not actual 3D coordinate comparison.

Text encoder. We use OpenAI TEXT-EMBEDDING-3-LARGE (3,072-d) for all text embeddings (instruction queries and annotation metadata). All embeddings are pre-cached offline as .npy files keyed by SHA1 hashes; no text encoder runs during training. The cached files are archived for reproducibility.

Fusion. The protein representation is the simple concatenation of ESM-2 and ProstT5 embeddings: $\mathbf{h} \in \mathbb{R}^{2304}$. The adapter MLP learns to fuse them in the shared space.

B.2. Adapter MLP Specification

Each tower uses the same adapter architecture (Table 9). Both outputs are L2-normalized to a shared 512-d embedding space.

Table 9. Adapter MLP specification for protein and text towers.

Layer	Protein tower	Text tower
LayerNorm	2,304	3,072
Linear	2,304 → 1,024	3,072 → 1,024
Activation	GELU	GELU
Dropout	p=0.1	p=0.1
Linear	1,024 → 512	1,024 → 512
LayerNorm	512	512
L2 normalize	unit sphere	

B.3. FiLM Conditioning Details

The instruction embedding $\mathbf{q} \in \mathbb{R}^{3072}$ is first compressed to a 64-d bottleneck: $\tilde{\mathbf{q}} = \text{GELU}(\text{LayerNorm}(\mathbf{q})\mathbf{W}_q)$, where $\mathbf{W}_q \in \mathbb{R}^{3072\times 64}$. The bottleneck is deliberately low-dimensional because protein function queries occupy a narrow subspace of the text embedding space (only a handful of distinct question types exist). From $\tilde{\mathbf{q}}$, two linear maps (bias=False, zero-initialized) produce scale and shift vectors in $\mathbb{R}^{512}$.

The zero initialization ensures that FiLM begins as an identity transform ($\mathbf{z}_{\text{out}} = \mathbf{z}$), so any deviation from identity is learned. The evidence that FiLM learns meaningful query-specific modulation (rather than generic capacity) is the comparison: FiLM *degrades* without query pooling (R@10: 0.229 vs. 0.231 structural-only), but with query pooling the same model jumps to 0.324. This pattern (FiLM is harmful without clean signal, beneficial with it) strongly suggests

query-dependent modulation rather than mere parameter overhead. Direct inspection of the learned vectors confirms this: scale vectors are weakly or negatively correlated across tasks (cosines -0.28 to $+0.35$), showing that FiLM amplifies different dimensions for different instructions, while shift vectors act as a largely shared offset (Figure 6).

Parameter counts. Table 10 summarizes trainable parameter counts for each FiLM configuration.

Table 10. Trainable parameter counts by configuration.

Configuration	Parameters
No FiLM (unconditional)	6.57M
ProtFiLM (protein tower only)	6.84M
BiFiLM (both towers)	7.11M

B.4. Notation Summary

Table 11 consolidates the mathematical notation used throughout the paper.

Table 11. Summary of notation.

Symbol	Dim	Description
$\mathbf{h}$	$\mathbb{R}^{2304}$	Fused ESM-2 + ProstT5 protein repr.
$\mathbf{z}_p$	$\mathbb{R}^{512}$	Adapted protein embedding
$\mathbf{z}_t$	$\mathbb{R}^{512}$	Adapted text embedding
$\mathbf{q}$	$\mathbb{R}^{3072}$	Instruction query embedding
$\tilde{\mathbf{q}}$	$\mathbb{R}^{64}$	Compressed query bottleneck
$\mathbf{p}$	$\mathbb{R}^{1024}$	L2-normalized ProstT5 embedding
$\mathbf{e}$	$\mathbb{R}^{1280}$	L2-normalized ESM-2 embedding
$\mathbf{t}$	$\mathbb{R}^{3072}$	Text (annotation) embedding
τ	scalar	Learnable temperature
α	scalar	Structural retriever mixing weight
τ_{query}	scalar	Query-pooling cosine threshold
τ_{fn}	scalar	False-negative cosine threshold
b	scalar	SigLIP bias term

C. Training Details

C.1. Optimization Hyperparameters

Table 12 consolidates all training hyperparameters.

Hardware and compute. All models are trained on a single NVIDIA A6000 GPU. Training takes approximately 5 minutes per epoch; typical runs early-stop at 15–30 epochs, yielding $\sim$1.5–2.5 hours per model. With 21 contrastive model variants, total compute for the full ablation is approximately 40–50 GPU-hours.

C.2. FiLM Initialization and Temperature

The scale and shift maps in the FiLM conditioning layer are initialized to zero, so training begins from the unconditional baseline ($\mathbf{z}_{\text{out}} = \mathbf{z}$) and gradually introduces query-dependent modulation. The bottleneck projection $\mathbf{W}_q$ and all scale/shift parameters are learned end-to-end via the contrastive loss. The learnable temperature τ is parameterized as $\tau = \exp(\log \tau)$ and clamped to $[0.01, 0.20]$, initialized at $\tau_0 = 0.07$.

Table 12. Consolidated training hyperparameters.

Category	Parameter	Value
Optimization	Optimizer	AdamW
	Learning rate	10^{-4}
	Weight decay	2×10^{-2}
	Scheduler	OneCycleLR (cosine)
	Warmup	3% of total steps
	Gradient clipping	1.0 (global norm)
	Seed	42
Batching	Micro batch size	64
	Gradient accumulation	4 steps
	Effective batch size	256
	DataLoader workers	4
Early Stopping	Max epochs	50
	Patience	6 (dev MRR@10)
Architecture	d_{hidden}	1,024
	d_{model}	512
	$d_{\text{query_proj}}$	64
	Dropout	0.1
Temperature	τ_0 (initial)	0.07
	$\tau_{\min}$	0.01
	$\tau_{\max}$	0.20
Negatives	Pool neg. per anchor	15
	SigLIP bias b	-10.0

C.3. Negative Sampling Strategy

Each protein's training negatives come from two sources:

1. **In-batch negatives:** Gold annotations of other proteins in the same mini-batch. Proteins sharing the same gold metadata SHA1 hash are treated as co-positives rather than negatives.
2. **Pool negatives:** Up to 15 annotations sampled from the protein's structural retrieval candidates (top-100 by weighted cosine $\alpha \cdot \cos_{\text{ProstT5}} + (1-\alpha) \cdot \cos_{\text{ESM-2}}$ over the cross-task training corpus). These yield harder comparisons than random sampling because they come from the protein's structural neighborhood. Pool candidates are pre-filtered to exclude (a) the anchor's own gold annotation (exact SHA1 match) and (b) annotations with $\cos(\mathbf{t}_{\text{gold}}, \mathbf{t}_j) \geq \tau_{\text{fn}}$, where $\mathbf{t}$ denotes the metadata embedding.

Query pooling. For each anchor with instruction $\mathbf{q}_i$, any batch or pool candidate j is masked (excluded from gradient) if $\cos(\mathbf{q}_i, \mathbf{q}_j) < \tau_{\text{query}}$. This ensures only candidates with instructively similar queries provide gradient signals,

preventing trivial cross-task shortcuts.

C.4. False-Negative Filtering

False-negative protection operates at three layers:

1. **Candidate building** (offline, train split only): Drop candidates matching the anchor's gold SHA1 hash or exceeding the semantic cosine threshold τ_{fn} against the gold metadata embedding.
2. **Dataset pool sampling** (per-sample): Defense-in-depth cosine filter on subsampled pool negatives, handling cases where manifests were built with a different threshold than training.
3. **Collate-time valid mask** (per-batch): Same cosine threshold applied to the full batch matrix, including in-batch negatives. This layer is not redundant with layers 1–2 because in-batch negatives are arbitrary gold annotations from other proteins that never went through per-anchor candidate construction.

C.5. Data Balancing

A `TaskBalancedSampler` groups dataset indices by task, pads each task's shuffled index list to the length of the largest task (by cycling), then interleaves one index per task per round, flattens, and shuffles. Smaller tasks are oversampled so that all tasks have equal representation per epoch.

C.6. Loss Functions

SigLIP (primary). A pairwise sigmoid loss with bias $b = -10.0$, adopted from the SigLIP paper's recommendation for settings with low positive-pair rates ($\sim$0.4% per row in a batch of $256 + 15$ candidates):

$$\mathcal{L}_{ij} = -\log \sigma\big(y_{ij}(\text{score}(p_i, t_j) + b)\big) \tag{4}$$

Row reweighting ensures that positives and negatives each contribute exactly 50% of each row's loss: positives receive weight $0.5/n_{\text{pos}}$ and negatives $0.5/n_{\text{neg}}$. The loss is symmetrized (protein→text + text→protein) for in-batch negatives and protein→text only for pool negatives.

Soft-SigLIP (variant). Negatives are weighted by inverse metadata similarity to the gold: $w_j = \text{clip}(1 - \cos(\mathbf{t}_{\text{gold}}, \mathbf{t}_j), 0, 1)$, where $\mathbf{t}$ denotes the raw metadata embedding. This *down-weights* negatives that are semantically close to the gold (potential false negatives), providing a softer alternative to the hard threshold.

C.7. Stage 2: Fine-Tuning

An optional second stage loads a Stage 1 checkpoint and switches from the contrastive SigLIP objective to a pointwise ranking loss. For each anchor protein, we form a candidate list consisting of the gold annotation plus up to 49 negatives drawn from the same per-protein candidate pool used in Stage 1 (not re-mined). The loss is softmax cross-entropy that maximizes the probability of the gold annotation scoring highest:

$$\mathcal{L}_{\text{rank}} = -\log \frac{\exp(s_{\text{gold}})}{\sum_j \exp(s_j)},$$

where $s_j = \mathbf{z}_p \cdot \mathbf{z}_{t_j}/\tau$ is the cosine similarity score. The entire protein adapter (including any protein-side FiLM layers) is frozen; only the text adapter (with any text-side FiLM layers) and the learnable temperature $\log \tau$ receive gradients.

Table 13. Stage 2 hyperparameters.

Parameter	Value
Learning rate	2×10^{-5}
Weight decay	10^{-2}
LR schedule	OneCycleLR (cosine, 5% warmup)
Micro batch size	8
Effective batch size	64 (8× grad. accum.)
Max candidates / protein	1 gold + 49 negatives
FN threshold	τ_{fn}=0.7
Max epochs	6
Early stopping patience	3 (dev MRR@10)
Frozen components	Protein adapter (incl. FiLM)

Our ablations (Table 23) show that Stage 2 provides marginal and inconsistent gains over Stage 1 alone.

C.8. Dev Evaluation During Training

During training, validation uses *dev_max_per_task*=512 samples per task with *dev_n_neg*=9 negatives per anchor, computing MRR@10 over 10 candidates (1 positive + 9 negatives). We assume that ranking ability in a 10-candidate setting transfers to larger pools.

D. LLM Generation Prompt and Setup

D.1. Generation Prompt Template

The full generation prompt is shown in Figure 3. The *contexts_json* payload labels each item with `"source": "structural"` or `"source": "contrastive"`, so the LLM sees accurate source metadata.

The *contexts_json* field is structured as {`"structural": [...], "contrastive": [...]`}, where each item contains `db_label` (the annotation text), `confidence level` (the raw numeric similarity score), and `source`.

System prompt:

You are a protein function prediction expert with deep knowledge of biological terminology.

User prompt template *(placeholders {instruction}, {sequence}, {contexts_json}, and {few_shot_json} are filled per sample):*

Instruction: {`instruction`}

Protein sequence: {`sequence`}

Retrieved protein annotations: {`contexts_json`}

Example input–output pairs for this task: {`few_shot_json`}

Based on the instruction, the protein sequence, the retrieved information, and the examples:

- **Use PRECISE biological terminology** from the retrieved annotations
- **Prioritize high-confidence matches**—they are most similar
- **Avoid generic descriptions**—be specific

Output ONLY the functional description of this protein in the following JSON format:

```
{"description": "..."}
```

Do not output any other text or explanation. Only output the JSON answer.

Figure 3. Generation prompt template.

Table 14. LLM generation configuration.

Parameter	Value
Primary model	Claude Sonnet 4.6
Temperature	1.0
Max tokens	2,048
Parallelism	30 threads
Retries	3 (exponential backoff)
Few-shot examples per task	3

D.2. Generation Configuration

Temperature 1.0 is the Anthropic API default and the recommended setting for generative (as opposed to analytical) tasks (Anthropic, 2025). Since our task produces free-text protein descriptions, we use the default without further tuning.

D.3. Few-Shot Example Selection

For each task, 3 few-shot examples are sampled from the training set using a fixed random seed (`np.random.default_rng(42)`). The same 3 examples are reused for every test sample within that task. Protein sequences are truncated to 80 characters followed by "..." in the prompt.

E. LLM-as-Judge Evaluation

E.1. Judge Configuration

We evaluated model predictions using Claude Sonnet 4.6 as an LLM-as-judge evaluator. The judge was run with extended thinking enabled (*budget_tokens*=4,000), temperature 1.0, and effective *max_tokens*=5,024 (combining a 1,024-token output budget on top of the 4,000-token thinking budget). For each model, we scored 500 randomly sampled predictions per task using *seed*=42, across four tasks, for a total of 2,000 nominal judged samples per model. All baselines and our system were evaluated under the same judge configuration to ensure internally consistent cross-model comparisons.

E.2. Scoring Rubric

The judge assigns scores along four discrete axes using the fixed 5-point scale $\{0, 3, 5, 7, 10\}$:

- **Recall:** fraction of reference bio-entities recovered in the prediction.
- **Precision:** whether the prediction remains focused on reference-supported or biologically reasonable claims.
- **Specificity:** granularity of biological content, distinguishing concrete entities from vague functional language.
- **Plausibility:** biological coherence and freedom from contradictions or fabrications.

The final per-sample score is the mean of all active axis scores. Model-level results are macro-averaged across tasks.

E.3. Full Judge Prompt

The full judge prompt is shown in Figure 7.

F. Recall@10 Definition and Retrieval Pool Analysis

F.1. Recall@10 Definition

For each test sample, the system produces a merged retrieval list of 10 candidates (5 from the structural channel and 5 from the contrastive channel). We compute Recall@10 by checking whether the gold annotation's exact text (identified by its SHA1 hash) appears anywhere in this merged list:

$$\text{Recall@10} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\text{gold}_i \in \text{merged}_{10}(i)\right] \quad (5)$$

where N is the number of test samples. This is a strict exact-match criterion: semantically equivalent paraphrases of the gold do not count as hits. We report the macro-average across the four tasks. Because Recall@10 depends only on the retriever's ranked output and not on LLM generation, it is fully deterministic and complements the LLM-dependent Entity-BLEU metric in our ablation comparisons.

F.2. Hybrid Merge Mechanics

In the hybrid 5+5 scheme, structural candidates always occupy positions 1–5 and contrastive candidates positions 6–10 in the merged list. Consequently, R@1 is always determined by the structural channel alone and is identical across all contrastive model variants. The contrastive channel contributes exclusively to R@10 improvement (positions 6–10).

At inference, contrastive scores are computed as score $= 0.5 \cdot ((\mathbf{z}_{\text{corpus}} \cdot \mathbf{z}_p) + 1.0)$, normalizing cosine similarity from $[-1, 1]$ to $[0, 1]$. When $\tau_{\text{query}} > 0$, scores are masked $(-\infty)$ for corpus entries whose training queries have cosine similarity below τ_{query} with the current instruction.

F.3. Per-Task Retrieval Pool Coverage

Figure 4 reports per-task Recall@10 for the structural-only baseline and ProtQueSt hybrid on the full dataset (14,503 samples).

The contrastive channel's relative contribution is *largest* on catalytic activity (+107%) despite its smaller absolute Entity-BLEU, suggesting the contrastive retriever finds relevant catalytic annotations but the LLM struggles to synthesize them into correct descriptions.

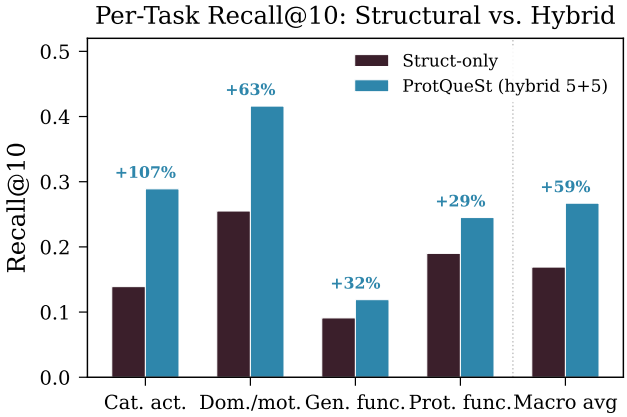

Figure 4. Per-task Recall@10 on the full dataset (14,503 samples). Percentages indicate the relative gain from the contrastive channel. Catalytic activity benefits most (+107%).

G. Full Baseline Comparison

Tables 15, 16, and 17 report the full per-metric breakdown, per-task Entity-BLEU, and per-task Judge-F scores for all baselines and our model on the full Prot-Inst-OOD benchmark.

ProtQueSt achieves the highest Judge-F on catalytic activity (6.67) and protein function (6.82), while Prot2Text-V2 leads on domain/motif (6.92) and general function (6.41). This suggests ProtQueSt's advantage is strongest on tasks requiring precise functional descriptions (catalytic reactions, protein function) rather than structural motif annotation where fine-tuned sequence-to-text models excel at reproducing training-set vocabulary.

Table 15. Full evaluation of all baselines and our model on Prot-Inst-OOD (14,503 samples per model for lexical metrics; 500 predictions per task, 2,000 nominal per model for the judge). **Lexical metrics (0–100 scale):** Entity-BLEU (E-BL), METEOR, and ROUGE-L (RG-L). **LLM-as-judge axes (0–10 scale):** Recall (J-Rec), Precision (J-Prec), Specificity (J-Spec), Plausibility (J-Plaus), and Final (Judge-F). Bold marks the top score in each column.

Method	E-BL	METEOR	RG-L	J-Rec	J-Prec	J-Spec	J-Plaus	Judge-F
LLM-only (3-shot)								
Llama-3.1-8B	3.21	18.70	11.27	0.14	3.01	6.69	3.41	3.31
Qwen3-8B	1.09	24.27	23.52	0.27	2.13	7.90	2.91	3.30
Sonnet 4.6	2.96	28.96	28.57	1.36	2.67	8.13	4.79	4.24
GPT-5.4	0.30	17.84	16.83	0.83	3.16	7.68	3.85	3.88
Domain-specific pretrained								
Evolla-10B-DPO	10.54	19.05	10.18	1.58	3.31	6.37	4.59	3.96
Fine-tuned on OOD training data								
Qwen3-8B-FT	2.20	34.72	37.20	0.74	3.28	8.95	3.31	4.07
BioT5+	4.37	32.64	37.68	1.24	3.99	8.45	4.27	4.49
InstructProtein	5.96	31.68	34.15	0.69	3.57	7.39	3.85	3.87
Prot2Text-V2	19.35	49.81	**50.00**	3.34	**6.39**	8.82	**6.38**	6.23
Retrieval-based								
MMseqs2	21.91	44.52	43.65	2.79	4.92	9.30	5.37	5.60
RAPM (GPT-4.1)	24.11	37.30	28.83	4.33	4.74	8.77	5.73	5.89
RAPM (Sonnet 4.6)	35.56	42.48	32.75	4.57	4.86	9.00	5.79	6.06
Only-Structure	44.26	46.42	27.72	5.34	3.70	9.59	6.13	6.19
ProtQueSt	**48.79**	**50.84**	31.43	**5.59**	4.04	**9.61**	6.33	**6.40**

Table 16. Per-task Entity-BLEU on the full Prot-Inst-OOD benchmark (14,503 samples). Bold marks the top score per task.

Method	Cat. act.	Dom./mot.	Gen. func.	Prot. func.	Avg
LLM-only (3-shot)					
Llama-3.1-8B	2.80	0.01	8.93	1.08	3.21
Qwen3-8B	0.01	0.00	3.95	0.41	1.09
Sonnet 4.6	0.17	0.49	8.43	2.75	2.96
GPT-5.4	0.01	0.02	0.20	0.96	0.30
Domain-specific pretrained					
Evolla-10B-DPO	7.97	4.32	16.69	13.20	10.54
Fine-tuned on OOD training data					
Qwen3-8B-FT	0.75	0.25	1.87	5.92	2.20
BioT5+	0.10	2.34	0.24	14.80	4.37
InstructProtein	1.02	0.00	0.31	22.51	5.96
Prot2Text-V2	18.51	13.94	22.54	22.39	19.35
Retrieval-based					
Only-Structure	45.47	43.84	30.64	57.09	44.26
RAPM (GPT-4.1)	31.29	23.28	8.87	33.02	24.12
RAPM (Sonnet-4.6)	31.53	35.13	24.15	51.43	35.56
MMseqs2	12.79	12.87	16.76	45.22	21.91
ProtQueSt	**50.84**	**52.17**	**32.63**	**59.51**	**48.79**

Table 17. Per-task Judge-F (0–10 scale, 500 samples/task) on Prot-Inst-OOD. Bold marks the top score per task.

Method	Cat. act.	Dom./mot.	Gen. func.	Prot. func.	Avg
LLM-only (3-shot)					
Llama-3.1-8B	2.74	3.62	4.02	2.87	3.31
Qwen3-8B	2.94	4.58	2.51	3.18	3.30
Sonnet 4.6	4.05	5.30	2.96	4.66	4.24
GPT-5.4	3.19	4.76	3.40	4.19	3.88
Domain-specific pretrained					
Evolla-10B-DPO	3.53	4.29	3.61	4.41	3.96
Fine-tuned on OOD training data					
Qwen3-8B-FT	3.35	5.57	2.90	4.45	4.07
BioT5+	3.58	6.02	3.49	4.87	4.49
InstructProtein	2.99	5.06	2.42	5.02	3.87
Prot2Text-V2	5.15	**6.92**	**6.41**	6.45	6.23
Retrieval-based					
MMseqs2	4.37	6.70	4.96	6.36	5.60
RAPM (GPT-4.1)	6.17	6.05	5.14	6.22	5.89
RAPM (Sonnet 4.6)	5.84	6.68	5.31	6.40	6.06
Only-Structure	6.39	6.12	5.46	6.79	6.19
ProtQueSt	**6.67**	6.46	5.63	**6.82**	**6.40**

H. Full-Dataset Per-Task Results

H.1. Per-Task Lexical Results

Table 18 shows per-task results for ProtQueSt on the full Prot-Inst-OOD benchmark (14,503 samples), using the Prot-FiLM + SigLIP model (τ_{query}=0.65) with Claude Sonnet 4.6 generation.

Table 18. Per-task full-dataset results for ProtQueSt (ProtFiLM + SigLIP, τ_{query}=0.65).

Task	n	E-BL	METEOR	RG-L
Catalytic activity	1,987	50.84	53.81	48.25
Domain/motif	2,732	52.17	49.22	21.25
General function	4,297	32.63	41.09	27.54
Protein function	5,487	59.51	59.24	28.70
Macro avg	**14,503**	**48.79**	**50.84**	**31.43**

H.2. Per-Task Retrieval Metrics

Table 19 reports full-dataset per-task retrieval metrics for ProtQueSt and the structural-only baseline. MRR (Mean Reciprocal Rank) is $\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i}$, where rank_i is the position of the gold annotation in the ranked list (0 if absent). R@k is the fraction of samples whose gold annotation appears in the top k.

R@1 is identical per task across all models because position 1 is always the top structural candidate. The contrastive channel contributes exclusively to R@10 improvement.

H.3. Encoder × Generator Ablation

Table 20 crosses two text encoders with two generators, holding all upstream retrieval logic byte-identical.

Table 19. Full-dataset per-task retrieval metrics (14,503 samples). ProtQueSt uses hybrid 5+5 merging.

Retriever	Task	MRR	R@1	R@10
Struct-only	Cat. act.	0.050	0.017	0.139
	Dom./mot.	0.096	0.038	0.255
	Gen. func.	0.037	0.016	0.091
	Prot. func.	0.088	0.050	0.190
	Macro	**0.068**	**0.030**	**0.169**
ProtQueSt	Cat. act.	0.072	0.017	0.289
	Dom./mot.	0.120	0.038	0.416
	Gen. func.	0.041	0.016	0.119
	Prot. func.	0.096	0.050	0.245
	Macro	**0.082**	**0.030**	**0.267**

The fully open-source pipeline (BGE-M3 + Llama-3.1-8B-Instruct) achieves Judge-F 5.93, preserving 85% of the 3.09-point retrieval gain over the no-retrieval Llama-3.1-8B 3-shot baseline (3.31). The BGE-M3 variant was *retrained from scratch* with BGE-M3 text embeddings and rebuilt query/metadata matrices, not used as a drop-in replacement at inference.

Table 20. Ablation of ProtQueSt's text encoder and generation model. We cross OpenAI's TEXT-EMBEDDING-3-LARGE (closed-source, 3072-d) with BAAI's BGE-M3 (open-source, 1024-d) on the encoder side, and Llama-3.1-8B-Instruct (open-weight) with Claude Sonnet 4.6 (proprietary) on the generator side.

Encoder	Generator	E-BL	METEOR	RG-L	J-Rec	J-Prec	J-Spec	J-Plaus	Judge-F
BGE-M3	Llama-3.1-8B-Instruct	19.04	32.31	26.91	3.97	4.62	9.22	5.90	5.93
	Claude Sonnet 4.6	43.68	46.89	30.35	5.27	4.03	9.54	6.22	6.27
TEXT-EMBEDDING-3-LARGE	Llama-3.1-8B-Instruct	23.24	35.61	28.15	4.32	**4.82**	9.22	6.15	6.13
	Claude Sonnet 4.6	**48.79**	**50.84**	**31.43**	**5.59**	4.04	**9.61**	**6.33**	**6.40**

I. Comprehensive Ablation Tables

I.1. Ablation Subset Characterization

All ablation experiments use n=256 samples per task (1,024 total). This subset consists of the *first* 256 rows in JSONL file order per task (a deterministic prefix, not a random sample). Empirically, absolute metric values on this prefix differ from the full 14,503-sample benchmark, and the direction of the gap varies by metric and model. For example, structural-only R@10 is higher on the prefix (0.231 vs. 0.169), while ProtQueSt Entity-BLEU is higher on the full set. This means the subset is not uniformly easier or harder, but simply not representative of the full distribution. **Relative model rankings are preserved** (all models use the same prefix), so the ablations remain valid for comparing design choices.

I.2. Extra Ablation Tables

This appendix expands the body ablation summary (Table 2) with architecture (Table 21), false-negative threshold (Table 22), stage-2 fine-tuning (Table 23), and Recall@10 retrieval metrics (Table 24).

Table 21. Architecture ablation (n=256, masked / unmasked Entity-BLEU). Structural-only baseline: 44.95.

Loss	FiLM	Q-pool	E-BL (m/u)	Δ
SigLIP	None	—	44.24	−0.71
SigLIP	None	0.65	45.87 / 44.04	+0.92
SigLIP	BiFiLM	—	44.19	−0.76
SigLIP	BiFiLM	0.65	45.96 / 45.75	+1.01
SigLIP	ProtFiLM	—	42.49	−2.46
SigLIP	ProtFiLM	0.65	**47.46** / 46.85	**+2.51**
Soft	BiFiLM	0.65	45.97 / 45.60	+1.02
Soft	ProtFiLM	0.65	47.07 / 47.18	+2.12

Table 22. False-negative threshold ablation (n=256, masked Entity-BLEU).

FiLM	Loss	τ_{fn}=0.60	0.65	0.70
BiFiLM	SigLIP	46.35	46.51	45.96
BiFiLM	Soft	46.82	**47.88**	45.97
ProtFiLM	SigLIP	—	47.11	47.46
ProtFiLM	Soft	—	47.35	47.07

Table 23. Softmax ranking stage-2 fine-tuning (n=256, masked Entity-BLEU). Δ: Stage 1+2 minus Stage 1.

Loss	FiLM	fn	Stage 1	Stage 1+2	Δ
SigLIP	BiFiLM	0.70	45.96	46.43	+0.47
SigLIP	BiFiLM	0.65	46.51	46.48	−0.03
Soft	BiFiLM	0.65	47.88	47.44	−0.44
SigLIP	ProtFiLM	0.70	47.46	46.90	−0.56
Soft	ProtFiLM	0.70	47.07	**47.90**	+0.83
SigLIP	ProtFiLM	0.65	47.11	47.22	+0.11
Soft	ProtFiLM	0.65	47.35	46.67	−0.68

Table 24. Recall@10 retrieval metrics (n=256, deterministic). Structural-only Recall@10: 0.231.

FiLM	Config	fn	R@10
With query pooling (τ_{query}=0.65)			
ProtFiLM	SigLIP	0.70	0.324
ProtFiLM	Soft	0.70	0.309
BiFiLM	Soft	0.65	**0.328**
BiFiLM	SigLIP	0.65	0.313
ProtFiLM	SigLIP	0.65	0.308
ProtFiLM	Soft	0.65	0.306
Without query pooling			
BiFiLM	SigLIP	0.70	0.264
None	SigLIP	0.70	0.238
None	SigLIP	0.65	0.256
ProtFiLM	SigLIP	0.70	0.229

I.3. Per-Task Merging Sweep

Figure 5 breaks down Entity-BLEU by task for the main merging strategies (n=256), revealing that catalytic activity and general function are consistently the hardest tasks for the contrastive channel.

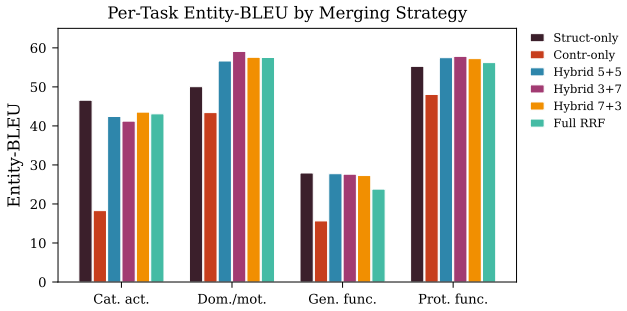

Figure 5. Per-task Entity-BLEU by merging strategy (n=256, BiFiLM + SigLIP, τ_{query}=0.65). The contrastive channel benefits domain/motif and protein function most; catalytic activity and general function remain structurally dominated.

The key pattern across merging strategies is that the contrastive channel helps most on domain/motif and protein function, where it lifts scores well above structural-only (e.g., domain/motif: 50.05 → 56.63 with hybrid 5+5), but offers little on catalytic activity and general function, where structural-only already dominates the contrastive-only channel by a wide margin (46.58 vs. 18.29 and 27.93 vs. 15.65). Consequently, hybrid 5+5 balances the two channels well overall: it matches or exceeds structural-only on every task except catalytic activity (where replacing 5 structural slots with contrastive candidates costs ∼4 points). Shifting the budget toward contrastive (hybrid 3+7) further benefits domain/motif (+2.5) but widens the catalytic activity gap. Full RRF underperforms hybrid merging on general function (23.80 vs. 27.75), suggesting that interleaving by reciprocal rank can dilute the structural signal on tasks where contrastive retrieval is weak.

I.4. SigLIP vs. Soft-SigLIP Comparison

Table 25 consolidates the comparison between SigLIP and Soft-SigLIP across all FiLM configurations and false-negative thresholds (all with query pooling at τ_{query}=0.65).

Table 25. SigLIP vs. Soft-SigLIP comparison (n=256, masked Entity-BLEU, query-pooled).

FiLM	τ_{fn}	SigLIP	Soft	Δ
BiFiLM	0.60	46.35	46.82	+0.47
BiFiLM	0.65	46.51	**47.88**	+1.37
BiFiLM	0.70	45.96	45.97	+0.01
ProtFiLM	0.65	47.11	47.35	+0.24
ProtFiLM	0.70	**47.46**	47.07	−0.39

Soft-SigLIP tends to help BiFiLM more consistently than ProtFiLM. BiFiLM applies query conditioning to the text encoder, making it more sensitive to annotation similarity; the soft weighting of near-gold negatives may be more beneficial in this setting. For ProtFiLM, the pattern is inconsistent (Soft helps at τ_{fn}=0.65 but hurts at 0.70).

I.5. Masked vs. Unmasked Evaluation

For models trained with query pooling (τ_{query}=0.65), Table 21 reports both masked (contrastive pool filtered at τ_{query}=0.65, matching training) and unmasked (full pool) Entity-BLEU. The gap is generally <1 Entity-BLEU point, suggesting the contrastive model learns robust embeddings that transfer beyond the query-filtered training pool.

J. Qualitative Examples

J.1. Success Cases

We present three test samples where ProtQueSt produces correct predictions, each illustrating a different channel-contribution pattern: structural-only success (S1), contrastive-only rescue (S2), and complementary contribution from both channels (S3).

Example S1: Domain/motif (GIY-YIG endonuclease). *Structural channel succeeds; contrastive channel misses the gold annotation.*

- **Gold:** *GIY-YIG*
- **Structural top-3:** (1) ***GIY-YIG*** (cos =0.975); (2) "endodeoxyribonuclease activity" (0.945); (3) "endonuclease activity" (0.933). The structural channel retrieves the exact gold domain at rank 1 and functionally related endonuclease annotations at ranks 2–5.
- **Contrastive top-3:** (1) Integrase catalytic (0.663); (2) SAM-dependent MTase C5-type (0.640); (3) viral DNA recombinase (0.631). All five contrastive candidates are DNA-processing domains but none is the gold GIY-YIG motif.
- **Prediction:** "...GIY-YIG endonuclease domain, consistent with endodeoxyribonuclease activity and DNA binding functionality. This protein is predicted to function as a strand-specific endonuclease..."
- **Analysis:** The GIY-YIG superfamily has a distinctive fold that structural retrieval identifies directly. The contrastive channel retrieves DNA-associated domains (integrases, methyltransferases) that share functional keywords but not the specific motif. The LLM correctly anchors on the structural rank-1 annotation and integrates the endonuclease context from structural ranks 2–3 to produce a precise, detailed prediction.

Example S2: Domain/motif (FHA domain). *Contrastive rescues; structural channel misses the gold annotation.*

- **Gold:** *FHA* (Forkhead-Associated domain)
- **Structural top-3:** (1) "Plays a role in microtubule organization. Required for centriole subdistal appendage assembly" (cos =0.996); (2) "Plays a role in microtubule organization" (0.993); (3) same (0.988). All five structural candidates describe biological processes (microtubule organization, cell growth regulation) but none names the FHA domain.
- **Contrastive top-3:** (1) ***FHA, PH*** (0.734); (2) ***FHA*** (0.725); (3) DWD box (0.671). The contrastive channel retrieves the exact gold domain at ranks 1–2.
- **Prediction:** "...FHA (Forkhead-Associated) domain, PH (Pleckstrin Homology) domain, and SH3-binding motifs. The protein likely plays a role in microtubule organization and centriole subdistal appendage assembly..."
- **Analysis:** The structural channel retrieves structurally homologous proteins but their annotations describe cellular functions, not domain identity. The contrastive channel, conditioned on a domain-prediction instruction, retrieves domain-level labels. The LLM synthesizes both: domain names from contrastive candidates and functional context from structural candidates, producing a comprehensive answer.

Example S3: Catalytic activity (glutamyl-tRNA synthetase). *Both channels contribute; contrastive provides the exact reaction, structural provides enzymatic context.*

- **Gold:** *ATP + L-glutamate + tRNA(Glu) = AMP + diphosphate + L-glutamyl-tRNA(Glu)*
- **Structural top-3:** (1) "Catalyzes the attachment of glutamate to tRNA(Glu) in a two-step reaction" (cos =0.999); (2) same description (0.998); (3) "HIGH region, KMSKS region" (0.998). Gold reaction appears at structural rank 5 (0.994).
- **Contrastive top-3:** (1) ***ATP + L-glutamate + tRNA(Glu) = ... L-glutamyl-tRNA(Glu)*** (0.724); (2) *ATP + L-glutamine + tRNA(Gln) = ...* (0.692); (3) *ATP + L-cysteine + tRNA(Cys) = ...* (0.655). All five contrastive candidates are aminoacyl-tRNA synthetase reactions.
- **Prediction:** "...catalytic activity in the form of this chemical reaction: ATP + L-glutamate + tRNA(Glu) = AMP + diphosphate + L-glutamyl-tRNA(Glu)."
- **Analysis:** The structural channel correctly identifies the enzyme family (glutamyl-tRNA synthetase) and retrieves the gold at rank 5, but its top candidates emphasize the two-step mechanism and conserved sequence motifs. The contrastive channel places the exact reaction equation at rank 1 with high confidence. Together, the channels provide both the mechanistic context and the precise reaction formula, enabling the LLM to produce an exact match.

J.2. Failure Cases

We present two failure cases with different root causes.

Example F1: Catalytic activity (collagenase → phospholipase A2). Failure type: **retrieval failure** (gold not in top-10).

- **Gold:** Cleavage of the triple helix of collagen at 775-Gly|Ile-776 in the α1(I) chain.
- **Structural top-3:** (1) calcium ion binding / protein stabilization (0.922); (2) snake venom PLA2 / phospholipid hydrolysis (0.918); (3) lactose synthase regulatory subunit (0.911).
- **Contrastive top-3:** (1) hydroquinone oxidation (0.653); (2) retinal oxidation (0.643); (3) phosphate monoester hydrolysis (0.629).
- **Prediction:** "...Snake venom phospholipase A2 (PLA2) catalyzes the calcium-dependent hydrolysis of the 2-acyl groups in 3-sn-phosphoglycerides..."
- **Analysis:** The gold annotation (collagenase) does not appear in any of the 10 retrieved candidates. The structural channel returns calcium-binding proteins (collagenases are calcium-dependent metalloproteinases, explaining the structural similarity), while the contrastive channel retrieves generic hydrolysis reactions. The LLM faithfully synthesizes the retrieved PLA2 annotation, producing a confident but incorrect prediction.

Example F2: Protein function (chloroplast thylakoid → DNA regulation). Failure type: **generation failure** (correct annotation retrieved but LLM ignores it).

- **Gold:** *chloroplast thylakoid*
- **Structural top-3:** (1) regulation of DNA-templated transcription (0.926); (2) regulation of DNA-templated transcription (0.925); (3) H_2O_2 removal / lignin degradation (0.924).
- **Contrastive top-3:** (1) **chloroplast thylakoid** (0.706); (2) extracellular region, mitochondrion (0.706); (3) chloroplast thylakoid lumen (0.704).
- **Prediction:** "...primary function related to regulation of DNA-templated transcription, with potential involvement in oxidative stress responses..."

- **Analysis:** The contrastive channel correctly retrieves "chloroplast thylakoid" at rank 1, but the structural channel (positions 1–5) provides DNA regulation annotations. Because structural candidates occupy the first five positions in the prompt, the LLM exhibits primacy bias and follows the structural channel's narrative, ignoring the correct contrastive retrieval at position 6. This highlights a limitation of the fixed 5+5 ordering.

J.3. Query-Conditioning Demonstration

A central design feature of ProtQueSt is that the contrastive retriever conditions on the instruction, producing different retrievals for the same protein depending on the task. We verify this concretely using a 768-residue protein that appears in both the *domain/motif* and *general function* OOD test splits:

Protein sequence (768 residues)

MEQTFNPLQNKRSVVGQQVILECQLEGQPDPVIKWLKDGQNVTNCPDYQ
IFQEGNKFKLIIPSVQMADAGRFDCQAINAAGSKSSSCILIVAPPPSPGERQ
FMTAPSPRPPPTPVGPAAPYVVKELKHQMLKIGSSARFECRITAFPAAEITW
LKNGKPITNTSKYKIDNDPDSGICSLTIAMMFAEDVGQYSCSARNAHGQAV
TSAEILYKDKYNEWLREEQIKITQEKKRSMMEELDNAVQQPRKQKGTFYT
PHSQRLLEQLYTEEKVETDIKINESEASVENVPFQGVPQPPQVIRPLQPVSII
EGQKAELTCQIKGNPTPKVRWMKNGVPVQNSQRLQTSYNGAVASLIIKITF
AEDAGMYTLVAENQFGRTNQSANIQILTQNSVNGVVHRKNNAVEQQKLSR
DLAVTPDLLGQGRQKPIFQQPLCDLQVAENQNVRFDVRISGRPYPNIQWL
KNGVLLQHGHRYKLLSSQNDLNTLIIYMATVEDSGTYTCVATNESGQAQCS
CELTVKAHSQGSAAHFTEKFNSLIVHPGDSVELKCSAVGQPRPTYHWYKD
DEELIPGQCPYDIVNMPNGTRLRINNVRLDDSGCFQCNAVNMYGTAVHKA
PVKVQMKSSSQQITPFPRDSTSDQALIASATPELATIRHRQRAAGLSNAAWF
VEAASAEPPKIIGLSADHLSLLEGQLAHVEVRFQPENDQNLKVVWFKDDL
PLEQKARLMFNIEAGRASFDIDYVQLSDGGLYKCLVVNQKGKAEATFTISV
TELPEV

Structural channel (query-independent). Because the structural retriever uses only protein embeddings, both queries return identical candidates: (1–2) 5'-flap endonuclease / exonuclease annotations (cos =1.000, 0.989); (3–5) Ig-like / SH3 / Fibronectin type-III domain lists (0.984 each).

Contrastive channel with domain/motif instruction. Instruction: "*Please examine the following protein sequence and predict any domains or motifs you can discern.*"

- **Contrastive top-5:** (1) Ig-like, SH3, Fibronectin type-III (0.684); (2) SH3, IQ, PH, Ig-like, DH, Fibronectin type-III, Protein kinase (0.680); (3) Ig-like, Fibronectin type-III, Protein kinase (0.675); (4) Protein kinase, Fibronectin type-III, Ig-like C2-type (0.668); (5) Fibrinogen C-terminal, Laminin G-like, EGF-like (0.654).
- All five contrastive candidates are **domain-level annotations**, directly matching the instruction type.
- **Gold:** Ig-like, Fibronectin type-III, Protein kinase. **Prediction:** Correct.

Contrastive channel with general function instruction. Instruction: "*Could you evaluate the protein with this amino acid sequence and present a summary of its features?*"

- **Contrastive top-5:** (1) "Controls actin polymerization and depolymerization" (0.685); (2) "Functions as a component of the nuclear pore complex (NPC)" (0.679); (3) "Plays an important role in regulation of muscle cell contractile activity" (0.675); (4) "Necessary for signaling by class 3 semaphorins and subsequent remodeling of the cytoskeleton" (0.675); (5) "Involved in cytoskeleton remodeling" (0.674).
- All five contrastive candidates are **functional descriptions**, matching the general-function instruction type. None overlaps with the domain-query retrievals above.

Analysis. The structural candidates are identical across both queries, confirming that the structural channel is query-independent. In contrast, the contrastive channel produces entirely disjoint candidate sets: domain annotations for the domain query and functional summaries for the general-function query. This demonstrates that the FiLM-conditioned contrastive model learns to route the same protein to different regions of the annotation embedding space depending on the instruction, validating the query-conditioning mechanism.

K. Additional Analyses

K.1. Structural Retriever α Ablation

The structural retriever uses a fixed mixing weight α=0.7 (ProstT5 weight) in the weighted cosine score $s_{\text{struct}} = \alpha \cdot \cos_{\text{ProstT5}} + (1-\alpha) \cdot \cos_{\text{ESM-2}}$. We sweep $\alpha \in \{0.0, 0.2, 0.5, 0.7, 0.9, 1.0\}$ evaluated by Recall@10 on the test set to validate this choice. Table 26 reports per-task and macro-averaged results.

The results confirm that α=0.7 achieves the highest macro Recall@10 (0.1686), though the landscape is flat for $\alpha \geq 0.5$, with all values in $[0.5, 1.0]$ within 0.0011 of each other. ProstT5 dominates the structural signal. Using ProstT5 alone (α=1.0) drops macro recall by only 0.0011 relative to the best mixture. Conversely, ESM-2 alone (α=0.0) loses 0.0235, confirming that structure-derived embeddings (ProstT5) carry the majority of the retrieval-relevant information but sequence embeddings (ESM-2) provide a complementary boost, particularly for domain/motif (+0.0506 from α=0.0 to α=0.7).

Table 26. Structural retriever Recall@10 as a function of the ProstT5 mixing weight α. The pool is the full cross-task train set ($\sim$284K proteins); queries are test-split proteins. Bold indicates the best value per column.

α	Cat. act.	Dom./mot.	Gen. func.	Prot. func.	Macro
0.0	0.1112	0.2042	0.0854	0.1795	0.1451
0.2	0.1389	0.2467	**0.0931**	**0.1919**	0.1677
0.5	0.1389	0.2518	0.0901	0.1914	0.1680
0.7	0.1394	**0.2548**	0.0905	0.1899	**0.1686**
0.9	0.1399	0.2544	0.0901	0.1874	0.1679
1.0	**0.1404**	0.2537	0.0901	0.1859	0.1675

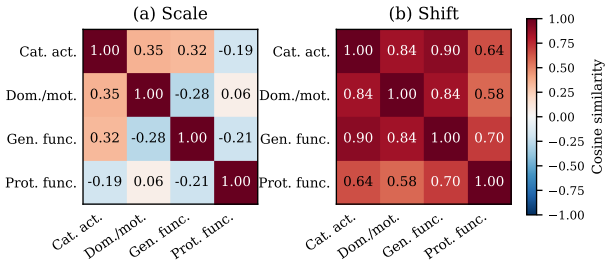

Figure 6. Cross-task cosine similarity of the learned FiLM vectors. (a) Scale vectors are weakly or negatively correlated across tasks, showing that FiLM amplifies different embedding dimensions for different instructions. (b) Shift vectors are highly correlated, acting as a largely shared offset.

K.2. FiLM Conditioning Analysis

To verify that FiLM genuinely adapts to the instruction embedding rather than learning a task-independent correction, we extract the learned scale and shift vectors for a representative instruction from each of the four tasks. Since both vectors are computed solely from the instruction embedding (not from the protein input), a given instruction produces exactly one scale vector and one shift vector in $\mathbb{R}^{512}$. Figure 6 reports the pairwise cosine similarity of these vectors across tasks. Scale cosines range from -0.28 (domain/motif vs. general function) to $+0.35$ (catalytic activity vs. domain/motif), with several pairs near zero or negative, showing that FiLM amplifies different embedding dimensions depending on the instruction. Shift vectors are more correlated across tasks (cosines 0.58–0.90), acting primarily as a shared offset rather than a task-specific routing mechanism.

K.3. Confidence Intervals

We compute 95% bootstrap confidence intervals by resampling predictions 1,000 times (seed 42). Entity-BLEU is a corpus-level metric, so each resample recomputes corpus BLEU; Judge-F and METEOR are per-sample means. Recall@10 is deterministic and requires no CI. Tables 27 and 28 report the results.

Table 27. Bootstrap 95% confidence intervals for ProtQueSt. Entity-BLEU and METEOR: full 14,503-sample benchmark. Judge-F: 500 samples per task.

Task	E-BL 95% CI	METEOR 95% CI	Judge-F 95% CI
Cat. act.	[48.65, 53.28]	[52.44, 55.17]	[6.45, 6.91]
Dom./mot.	[51.71, 56.36]	[47.77, 50.72]	[6.33, 6.68]
Gen. func.	[31.31, 33.95]	[40.39, 41.84]	[5.46, 5.82]
Prot. func.	[58.51, 60.15]	[58.47, 60.00]	[6.62, 6.94]
Macro	[48.44, 50.16]	[50.23, 51.43]	[6.31, 6.49]

Table 28. Recall@10 hit counts for ProtQueSt (deterministic).

Task	Hits	n	Recall@10
Cat. act.	575	1,987	0.2894
Dom./mot.	1,136	2,732	0.4158
Gen. func.	513	4,297	0.1194
Prot. func.	1,343	5,487	0.2448
Total	3,567	14,503	0.2459

Table 29. Computational cost breakdown.

Component	Cost
Training	
GPU hardware	NVIDIA A6000
Time per model	$\sim$1.5–2.5 h
Total (21 variants)	$\sim$40–50 GPU-h
Inference (per sample)	
Protein emb. (ESM-2+ProstT5)	$\sim$0.5–2.0 s
Structural retrieval (Faiss)	$<$10 ms
Contrastive scoring (MLP)	$<$5 ms
LLM generation (Sonnet 4.6)	$\sim$2–5 s
Storage	
Protein emb. cache	$\sim$4 GB
Text emb. cache	$\sim$2 GB
Faiss HNSW index	$\sim$500 MB
Adapter weights	$\sim$27 MB

K.4. Computational Cost Summary

Protein embedding extraction dominates single-sample latency because both ESM-2 (650M parameters) and ProstT5 must process the full sequence. However, embeddings are cached for the entire corpus, so retrieval at inference requires only the query protein's embeddings (computed once) plus fast Faiss lookup and MLP scoring. The LLM generation step is the primary cost driver for batch evaluation due to API rate limits and per-token pricing.

L. Baseline Details

L.1. Baseline Implementation Details

LLM-only (3-shot). Each LLM receives 3 in-context exemplars per task (same seed-42 selection as ProtQueSt) with protein sequences truncated to 80 characters. GPT-5.4 uses the OpenAI Responses API (GPT-5.4-2026-03-05, *reasoning_effort=none*, *max_output_tokens=8192*, concurrency 8).

Llama-3.1-8B-Instruct and Qwen3-8B use greedy decoding (*do_sample=False*, seed 42). Claude Sonnet 4.6 uses temperature 1.0 (matching our generation setup). All LLM-only baselines are evaluated on the full 14,503-sample test set for lexical metrics; judge scores use 500 samples per task.

Fine-tuned models. Prot2Text-V2, BioT5+, InstructProtein, and Qwen3-8B-FT are fine-tuned on the OOD training split. We use published checkpoints where available; Qwen3-8B-FT is fine-tuned in-house. **BioT5+ caveat:** The BioT5+ checkpoint (QIZHIPEI/BIOT5-PLUS-BASE) was pretrained on Mol-Instructions, which overlaps with Prot-Inst-OOD annotations; for fair comparison, see RAPM Table 3 (Wu et al., 2025).

Domain-specific pretrained. Evolla-10B-DPO is evaluated zero-shot (3-shot prompting) without fine-tuning on our training data.

Retrieval-based. MMseqs2 retrieves the top-10 most similar training-set proteins by sequence identity and passes their annotations to the generation LLM (same Sonnet 4.6 prompt). RAPM (Wu et al., 2025) uses MMseqs2 + ESM-2 dual-key retrieval with GPT-4.1 (temperature 0.7, top-p 0.9) or Sonnet 4.6 (temperature 0.7) as generators. Only-Structure uses our ProstT5 + ESM-2 weighted-cosine structural retriever (α=0.7) with 10 structural candidates (no contrastive channel).

L.2. Per-Task Baseline Breakdown

Per-task Entity-BLEU and Judge-F for all baselines are reported in Appendix G, Tables 16 and 17.

L.3. Fine-Tuning Hyperparameters

All four OOD-fine-tuned baselines (Qwen3-8B-FT, BioT5+, InstructProtein, Prot2Text-V2) are fine-tuned in-house on the Prot-Inst-OOD training split (284,526 examples). Common settings across all four runs: AdamW optimizer, learning rate 1×10^{-4}, cosine learning-rate schedule with warmup ratio 0.1, weight decay 0.01, bf16 mixed precision, and seed 42. Table 30 reports the per-baseline hyperparameters that vary across runs. Hardware: 4× NVIDIA A6000 GPUs; training durations were approximately 15.7 h (Qwen3-8B-FT), 2.6 h (BioT5+), 8.9 h (InstructProtein), and 11.5 h (Prot2Text-V2).

Generation configuration for fine-tuned baselines. For all OOD-fine-tuned baselines at evaluation time, we use greedy decoding (*do_sample=False*, seed 42) with each model's published prompt format. Prot2Text-V2 uses its native sequence-only input; BioT5+ and InstructProtein follow their published instruction templates; Qwen3-8B-FT uses our standard 3-shot prompt template (Appendix D).

Table 30. Per-baseline fine-tuning hyper-parameters. Shared settings (AdamW, lr 10^{-4}, cosine schedule, warmup 0.1, weight decay 0.01, bf16, seed 42) are listed in the §L.3 prose.

Baseline	Parameter	Value
Qwen3-8B-FT	Base model	Qwen3-8B
	Method	LoRA: `q,k,v,o`
	LoRA config	r=8, α=32, dropout 0.05
	Per-device batch	2
	Grad accumulation	8
	GPUs	4
	Effective batch	64
	Max seq. length	2,048
	Epochs	2
	Optimizer steps	8,892
	Grad checkpointing	Yes
	Distributed	DeepSpeed ZeRO-2
BioT5+	Base model	BIOT5-PLUS-BASE
	Method	Full fine-tuning
	Per-device batch	4
	Grad accumulation	8
	GPUs	2
	Effective batch	64
	Max src. / tgt. length	2,048 / 512
	Epochs	2
	Optimizer steps	8,890
	Grad checkpointing	No
	Distributed	DeepSpeed ZeRO-2
InstructProtein	Base model	InstructProtein
	Method	LoRA: `q,k,v,out`
	LoRA config	r=8, α=32, dropout 0.05
	Per-device batch	2
	Grad accumulation	8
	GPUs	2
	Effective batch	32
	Max seq. length	2,048
	Epochs	2
	Optimizer steps	17,782
	Grad checkpointing	Yes
	Distributed	DeepSpeed ZeRO-2
Prot2Text-V2	Base model	Prot2Text-V2-11B
	Architecture	ESM-2 3B + Llama-3.1-8B
	Method	LoRA: Llama (attn + MLP)
	LoRA config	r=8, α=32, dropout 0.05
	Per-device batch	1
	Grad accumulation	4
	GPUs	4
	Effective batch	16
	Max prot. / desc. length	1,021 / 512
	Epochs	1
	Optimizer steps	17,471
	Grad checkpointing	No
	Distributed	PyTorch DDP

M. Broader Impact and Limitations

M.1. Limitations

- **Retrieval ceiling equals corpus ceiling.** ProtQueSt can only surface annotations present in the training corpus. Proteins with genuinely novel functions not yet characterized in any database fall outside the system's reach. This constraint is shared by all current approaches, including generative models, since no system can predict functions for which no training signal exists.

- **Frozen encoder bottleneck.** FiLM can only re-weight features already encoded by frozen ESM-2 and ProstT5 embeddings. If a functional signal is absent from these representations, no amount of conditioning can surface it.

- **Single-vector protein representation.** CLS-token and mean-pooled embeddings collapse variable-length sequences into fixed-size vectors, discarding residue-level information. Functions governed by specific local motifs or binding sites may be underrepresented.

- **Benchmark scope.** All evaluations use the Prot-Inst-OOD benchmark; generalization to other protein function corpora and newly deposited proteins remains to be tested.

M.2. Broader Impact

Automated protein function annotation can accelerate characterization of the rapidly growing "dark proteome," the millions of sequenced proteins with no experimentally verified function. By providing retrieval-augmented functional descriptions grounded in curated databases, ProtQueSt could assist biocurators in prioritizing and drafting annotations for poorly characterized proteins.

However, automated annotation at scale carries the risk of error propagation: if the retrieval corpus contains incorrect annotations or the LLM generator introduces hallucinations, these errors could be amplified across downstream analyses. We emphasize that ProtQueSt is designed as a *recommendation engine* for expert biocurators, not as a replacement for experimental validation or expert review.

M.3. Reproducibility Statement

We plan to release all source code for training, evaluation, and the retrieval pipeline, along with trained adapter weights ($\sim$6.8M parameters). We may additionally release precomputed Faiss indices, cached protein embeddings, and archived text embeddings for exact reproducibility. The fully open-source configuration (BGE-M3 encoder + Llama-3.1-8B generator) achieves Judge-F 5.93 without any proprietary API dependencies.

Acknowledgements

We thank MIT Computer Science & Artificial Intelligence Laboratory (CSAIL) for GPU access. We thank Zehua Wang for helpful feedback and insightful discussions regarding this research. We also thank Bowen Yu, Yifei Jin, and Yue Zhuo for their generous support of this research.

System prompt:

You are an expert biocurator evaluating protein function predictions against curated UniProt ground truth. Apply the rubric strictly and consistently. Biological knowledge matters—recognize synonyms, equivalent rephrasings, and EC-class relationships. Identical inputs must yield identical scores. Scores are chosen from a fixed 5-point scale: {0, 3, 5, 7, 10}. Do not output other values for any axis.

User prompt template *(placeholders {instruction}, {description}, {metadata}, {prediction} are filled per sample):*

Evaluate a `PREDICTION` against a curated `GROUND TRUTH` for a protein function prediction task.

INPUT

TASK INSTRUCTION:
{`instruction`}

GROUND TRUTH DESCRIPTION *(curated from UniProt):*
{`description`}

KEY BIO-ENTITIES *(atomic entities pre-extracted from UniProt annotations; top-level separators are "—" and ";" while "+" and "=" within a single entity denote reaction components):*
{`metadata`}

PREDICTION:
{`prediction`}

STEP 1: Parse reference entities.

Split `KEY BIO-ENTITIES` into atomic entities using "—" and ";" as top-level separators. Keep reaction components joined: "(R)-prunasin + H2O = D-glucose + mandelonitrile" is one catalytic-reaction entity. For concept lists like "lipid binding" or "nucleus, cytoplasm", each comma-separated item is one entity unless the comma is inside a reaction.

Let N_{ref} be the count of atomic reference entities.

STEP 2: Match reference entities—5 match types.

For each reference entity, determine whether it is expressed in `PREDICTION` using these match types, in priority order:

Type 1 **Exact:** case-insensitive substring match of the entity or core term.

Type 2 **Lexical variant:** morphological or formatting variant. For example, "mitochondrion" ≡ "mitochondrial"; "DNA-binding" ≡ "binds DNA"; "ATP hydrolysis" ≡ "hydrolyzes ATP".

Type 3 **Biochemical synonym:** different terms for the same biological entity. For example, "peptidase" ≡ "protease" ≡ "proteinase"; "plasma membrane" ≡ "cell membrane"; "endoplasmic reticulum" ≡ "ER". Do not count superclass-only matches.

Type 4 **Reaction equivalence:** catalytic reaction described with different syntax. For example, "(R)-prunasin + H2O = D-glucose + mandelonitrile" ≡ "hydrolyzes prunasin to produce glucose and mandelonitrile".

Type 5 **Not a match:** superclass-only, same family but different member, opposite direction, or related but distinct biology.

Let N_{matched} be the count of reference entities matched via Types 1–4. For each match, record the type as 1, 2, 3, or 4.

STEP 3: Count contradictions.

A contradiction is a prediction claim mutually exclusive with the reference, such as kinase vs. phosphatase, mitochondrion vs. nucleus, catalytic reaction A vs. catalytic reaction B, or DNA-binding vs. RNA-binding only. Additional compatible or unrelated predictions are not contradictions.

Let $N_{\text{contradicted}}$ be the count of distinct contradictions.

Figure 7. Prompt instructions for LLM-as-a-judge evaluation of protein function predictions.

Figure 7, continued.

STEP 4: Score four axes. Each axis score must be one of $\{0, 3, 5, 7, 10\}$.

- **Axis 1—Recall:** fraction of reference entities recovered via Types 1–4. Let $r = N_{\text{matched}}/N_{\text{ref}}$.

$$\begin{aligned} 10 &: r = 1.00; \\ 7 &: r \geq 0.60; \\ 5 &: r \geq 0.40; \\ 3 &: r \geq 0.15; \\ 0 &: r < 0.15. \end{aligned}$$

- **Axis 2 (Precision):** prediction stays on-topic to reference biology. Let U be the count of biological claims unsupported by the reference or reasonable inference.

$$\begin{aligned} 10 &: U = 0; \\ 7 &: U \leq 2; \\ 5 &: U \leq 5; \\ 3 &: U \leq 10; \\ 0 &: U > 10 \text{ or prediction describes a different protein.} \end{aligned}$$

- **Axis 3—Specificity:** granularity of biological content, independent of correctness. A prediction can be specifically wrong or generally right.

$$\begin{aligned} 10 &: \text{concrete entities, substrates, products, EC numbers, domains,} \\ &\quad\ \text{localizations, named reactions, or specific residues;} \\ 7 &: \text{specific biology one step coarser;} \\ 5 &: \text{family or class level only;} \\ 3 &: \text{very general biology;} \\ 0 &: \text{generic platitudes with no specific biological content.} \end{aligned}$$

- **Axis 4—Plausibility:** free of contradictions and fabrications.

$$\begin{aligned} 10 &: N_{\text{contradicted}} = 0 \text{ and all claims are grounded;} \\ 7 &: N_{\text{contradicted}} = 0 \text{ with some speculation;} \\ 5 &: N_{\text{contradicted}} = 1; \\ 3 &: N_{\text{contradicted}} = 2; \\ 0 &: N_{\text{contradicted}} \geq 3 \text{ or describes a different protein.} \end{aligned}$$

STEP 5: Final score.

$$\texttt{FINAL_SCORE} = \text{round}\left(\frac{\text{Axis1} + \text{Axis2} + \text{Axis3} + \text{Axis4}}{N_{\text{active axes}}}\right),$$

where $N_{\text{active axes}}$ excludes any axis scored -1.

OUTPUT FORMAT—return exactly these 8 lines, no preamble, no commentary:

```
N_ref:  <integer>
N_matched:  <integer>
Match_types:  <comma-separated type integers per matched entity>
N_contradicted:  <integer>
N_unsupported:  <integer U from Axis 2>
Axis_scores:  <recall>,<precision>,<specificity>,<plausibility>
Critique:  <one sentence:  key mismatch, synonym match, or strength>
Final_score:  <integer 0--10>
```