GOAgent: Tool-Orchestrating Language Agents for Protein Function Annotation

Manu Ponnapati [1] **Brian Lynch** [2] **Joseph Jacobson** [1]

Abstract

Proteins constitute a significant portion of the functional apparatus of biological systems, yet accurate protein function annotation remains a fundamental challenge in computational biology. Conventional bioinformatics pipelines yield reliable annotations for well-characterized proteins but fail to generalize to novel sequences lacking close homologs in reference databases. Recent deep learning methods leveraging learned representations of protein sequence and structure could provide improvements over traditional homology-based approaches, though they are constrained by the completeness of the training datasets. Here we introduce GOAgent, a large language model agent that invokes a curated suite of sequence- and structure-derived bioinformatics tools and reasons over their outputs to predict Gene Ontology (GO) terms. We further evaluate a variant in which, rather than calling pre-defined tools, the agent is given a sandboxed execution environment together with the relevant input files and must orchestrate the analysis itself. We show that tool-augmented reasoning improves GO annotation quality over a zero-shot LLM baseline. We further train GOAgent end-to-end on multi-turn tool-calling rollouts via a policy optimization objective, yielding improved performance under both standard GO term prediction and a harder per-domain annotation task in which the agent must associate each predicted term with a specific functional region in the protein. While GOAgent does not match the raw accuracy of recent dedicated multimodal models for protein function prediction, it offers complementary strengths: extensibility and auditable tool-grounded reasoning, enabling new bioinformatics tools to be seamlessly incorporated.

[1]MIT Media Lab [2]Independent. Correspondence to: Manu Ponnapati <manvitha@mit.edu>.

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

1. Introduction

Proteins constitute a significant portion of the functional apparatus of biological systems, yet accurate protein function annotation remains a fundamental challenge in computational biology (Zhou et al., 2019). Our ability to sequence proteins has far outpaced our ability to functionally characterize them, leaving many sequenced proteins unannotated (Consortium, 2023). Conventional bioinformatics pipelines grounded in sequence similarity yield reliable annotations for well-characterized proteins but fail to generalize to novel sequences lacking close homologs in reference databases (Kulmanov et al., 2018; Kulmanov & Hoehndorf, 2020). Deep learning models that leverage learned representations of protein sequence and structure have emerged as a powerful alternative, demonstrating strong predictive performance even across evolutionarily distant sequences (Wang et al., 2025b). However, these models are constrained by the limited diversity of annotated training data and provide little interpretability into the features that drive their predictions (Sanderson et al., 2023; Novakovsky et al., 2023).

In parallel, there has been an explosion of LLM agents developed for autonomous scientific discovery (Lála et al., 2023; Mitchener et al., 2025; Stevens, 2025; Lu et al., 2026; Ghareeb et al., 2026). These models are compelling not only for their reasoning capabilities but also because they can integrate multimodal information and contextualize biological knowledge beyond sequence or structure features alone. Building on this progress, we hypothesise that an LLM agent equipped with established protein analysis tools can address the dual challenge of accuracy and interpretability in protein function prediction.

Here we introduce **GOAgent**, a large language model agent that invokes a curated suite of sequence- and structure-derived bioinformatics tools and reasons over their outputs to predict Gene Ontology (GO) terms. Rather than learning implicit feature representations, GOAgent explicitly invokes bioinformatics tools — including physicochemical sequence characterisation, transmembrane topology prediction (Hofmann et al., 1993), PROSITE motif scanning (Sigrist et al., 2026), binding-site likelihood estimation across protein, nucleic acid and small-molecules (Krapp et al., 2023), Rosetta-based structural metrics and solvent accessibility profiling (Leaver-Fay et al., 2011), inter-residue

bond analysis, and signal peptide classification via SignalP6 (Teufel et al., 2022) — and reasons over their outputs in natural language. This tool-based design yields an extensible framework that supports the addition of further modalities, for example molecular dynamics simulations and produces human-readable reasoning chains alongside its GO term predictions. We further evaluate a variant in which, rather than calling pre-defined tool functions, the agent is given a sandboxed execution environment together with access to relevant data files.

To evaluate the contribution of tools, we first establish baselines using the agent without tools and a larger zero-shot model (Qwen3-14B), then demonstrate that equipping the agent with tools yields measurable improvements in GO term prediction (Yang et al., 2025). We extend this framework in two ways: (i) by prompting the agent to predict GO terms at the domain level in addition to the full sequence; and (ii) by training the agent with GRPO on approximately 1,500 proteins with 8 rollout replicates each, showing that this yields additional gains on both tasks. While GOAgent does not match the raw accuracy of dedicated multimodal models for protein function prediction like BioReason-Pro (Fallahpour et al., 2026), it offers complementary strength.

Contributions.

1. We introduce GOAgent, a tool-orchestrating LLM agent for GO annotation that operates without learned protein embeddings and instead reasons over outputs from established bioinformatics tools
2. We show that tool-augmented Qwen3-8B outperforms zero-shot Qwen3-8B and matches or exceeds zero-shot Qwen3-14B in some cases.
3. We train the agent end-to-end on multi-turn tool-calling rollouts and demonstrate further gains over the untrained tool-using agent on both standard and per-domain annotation tasks.
4. We perform tool ablations and find that performance gains are distributed across the toolset rather than dominated by any single tool.
5. We open-source our codebase for agent rollouts, training, and evaluation, deployable on Modal Labs to enable reproduction without dedicated in house GPU infrastructure. We additionally release precomputed tool outputs for all tools across the full CAFA5 dataset and a limited subset of the temporal evaluation dataset from (Fallahpour et al., 2026).

2. Related Work

Early approaches to protein function prediction relied primarily on sequence homology, inferring function by transferring annotations from characterized proteins to novel sequences (Altschul et al., 1990; 1997; Kulmanov et al., 2018; Kulmanov & Hoehndorf, 2020; Lv et al., 2019; Torres et al., 2021; Ibtehaz et al., 2023). Subsequent work introduced machine learning and deep learning classifiers that learn latent sequence-to-function mappings, surpassing homology-based methods by capturing higher order relationships between sequence and functional terms. These models have progressively incorporated hierarchical modelling of the GO structure, semantic similarity objectives, and protein language model embeddings to exploit the structured relationships between functional terms. CAFA (Radivojac et al., 2013) is the community-wide benchmark for protein function prediction to provide standarized assessment of various protein function methods.

LLM-based approaches have increasingly been applied to protein function prediction, typically by combining pretrained sequence representations with language-driven reasoning. ProtNLM (Gane et al., 2022) applies pretrained language representations to downstream annotation tasks. Huo et al. (Huo et al., 2024) and STELLA (Xiao et al., 2025a) propose multimodal LLMs integrating sequence and structural encoders with general language model reasoning, demonstrating that bridging protein representations with contextual biological knowledge improves functional description prediction. ProtChatGPT (Wang et al., 2025a), ProteinGPT (Xiao et al., 2025b), and ProteinChat (Guo et al., 2023) explore conversational and multimodal interfaces that jointly incorporate structure and sequence encoders for natural language-based property prediction. More recently, dedicated multimodal models have pushed further by combining rich protein language model embeddings with supervised reasoning traces, achieving strong performance on CAFA-style benchmarks(Friedberg et al., 2023; Radivojac et al., 2013). Concurrently, PFUA (Fan et al., 2026) demonstrates that that agent-style reasoning with domain-specific tools outperforms text-only reasoning models on protein function understanding, providing empirical evidence that protein function prediction is a knowledge-intensive task that benefits from grounding in external tools rather than internal reasoning alone. Related structure-informed prediction systems such as AF2Bind (Gazizov et al., 2026) and Boltz-2 (Passaro et al., 2025) provide high-resolution insights into binding and interaction properties, but remain specialised to particular prediction subtasks and do not produce multi-evidence reasoning chains over heterogeneous tool outputs.

GOAgent takes a different approach, operating as a multi-turn tool-calling agent that explicitly invokes bioinformatics tools at inference time and reasons over their outputs in natural language. GOAgent does not approach the raw accuracy of dedicated multimodal protein function models like BioReason-Pro which relies on an underlying GO-GPT model trained using protein language embeddings to pre-

dict GO terms (Fallahpour et al., 2026). Rather, it offers a complementary paradigm in which any instruction-tuned model with tool-calling support can serve as the backbone, and new bioinformatics tools can be incorporated.

3. Methods

Code is available at https://github.com/ManvithaPonnapati/phage.

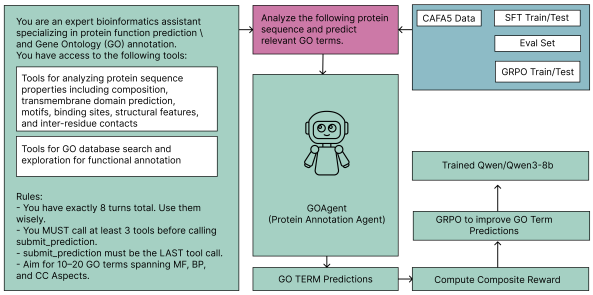

Figure 1. Schematic overview of GOAgent. Diagram shows a summary of the tools available to GOAgent and the interaction between the environment and the agent. Example task prompts for sequence-level and domain-level GO term annotations.

3.1. Dataset

We train and evaluate GOAgent on the Critical Assessment of Functional Annotation 5 (CAFA5) dataset (Zhou et al., 2019), the most recent iteration of the community-wide challenge for protein function prediction. CAFA5 provides a large-scale benchmark of protein sequences paired with Gene Ontology annotations spanning all three GO subontologies: Molecular Function (MF), Biological Process (BP), and Cellular Component (CC).

The agent is trained on set of 1,500 proteins with 8 rollout replicates per protein. Generalization is assessed on a held-out validation set of 1,000 proteins. We additionally test on proteins whose UniProt entries were published after the training cutoff for CAFA5 and Qwen3-8b. We also evaluate on subset of test data released by (Fallahpour et al., 2026) via Hugging Face which is a temporal dataset dated after the training date for the base model; full-dataset evaluation on this set was not performed due to the computational cost of multi-step rollouts.

3.2. Tools

The GOAgent toolset consists of domain-specific bioinformatics tools that return sequence and structure-derived features in natural language format. Each tool accepts a sequence hash as its identifier and returns structured JSON, which is rendered as natural language within the agent's context window. We precomputed the outputs of all sequence- and structure-based tools for the full CAFA5 target set. Motivated by the size of these precomputed files, tool outputs are served via a lightweight REST API: a FastAPI server hosting the precomputed data is made accessible via an ngrok tunnel, with each tool call implemented as an HTTP request to this endpoint. This setup decouples tool-serving from the agent, which also facilitates integration of remote or third-party tools and allows evaluation to scale across machines without requiring local copies of the full dataset.

Tools available to GOAgent. Sequence and structure tools are precomputed for the full CAFA5 target set; GO ontology tools are called live against the GO ontology.

Sequence & Structure.

- `get_sequence_information` — length, amino acid frequencies, net charge, hydrophobicity.
- `check_if_transmembrane_protein` — transmembrane topology prediction from TMPred.
- `get_motif_information` — PROSITE motif and domain scanning.
- `get_likelihood_information` — residue-level binding-site likelihood scores.
- `get_rosetta_details` — DSSP secondary structure annotation and per-residue solvent accessibility.
- `get_bond_information` — inter-residue contacts and bonds.
- `get_signalp_prediction` — SignalP 6 signal peptide classification.

GO Ontology.

- `go_search` — keyword search returning matching term IDs, namespaces, and definitions.
- `go_explore` — retrieves child terms or ancestors for a given GO term ID.
- `check_if_a_term_is_valid` — validates that a GO term ID is not obsolete.
- `submit_prediction` — terminal action: emits the final GO term set with a free-text justification.

3.3. Policy Optimization

We train GOAgent using Group Relative Policy Optimization (GRPO). At each iteration, a batch of proteins is sampled and N rollouts are generated per protein under the current policy. Relative advantages are computed within each group. We discard zero-variance groups and apply asymmetric clipping.

The reward is a weighted sum of two components scaled by a term-count penalty:

$$r = \left(w_{\text{GO}} \cdot r_{\text{GO}} + w_{\text{tool}} \cdot r_{\text{tool}}\right) \cdot s(n). \tag{1}$$

r_{GO} is a namespace-weighted combination of per-aspect SimGIC and BMA-Lin scores across MF, BP, and CC. r_{tool} rewards breadth of tool invocations across the rollout. The scale factor $s(n)$ linearly penalises submissions with fewer than $N_{\min}$ predicted GO terms, saturating at 1.0 above $N_{\min}$. SimGIC and BMALin are one of the many semantic similarity metrics for comparing two sets of GO terms. We chose these two to enable partial reward/credit for most trajectories.

3.4. Training Infrastructure

The training pipeline runs on Modal cloud infrastructure, which we chose because it removes the need for a dedicated GPU cluster or institutional H100 access: compute scales on demand and is released between runs, making the full GRPO pipeline practical for researchers without fixed hardware. Rollout generation is parallelised across multiple GPU nodes, with each node running an asynchronous vLLM engine that interacts with a centralised tool server over HTTP. Several tools in the GOAgent toolset themselves require non-trivial GPU inference to produce their outputs so we provide precomputed outputs for all tools; The released code includes the training, rollout, scoring, SFT, and GRPO scripts, along with Modal API definitions for individual tools.

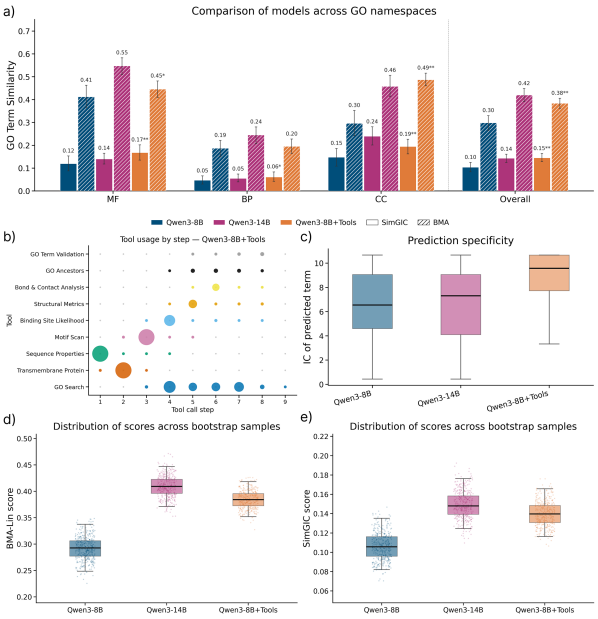

Figure 2. (a) GO semantic similarity (SimGIC and BMA-Lin) across MF, BP, CC, and Overall for zero-shot Qwen3-8B, zero-shot Qwen3-14B, and the Qwen3-8B tool-using base agent. Bars show bootstrapped means with 95% CIs; stars denote significance of the tool-using agent vs. zero-shot Qwen3-8B. (b)Tool usage by step across all rollouts for the Qwen3-8B base agent. Bubble area is proportional to call frequency at that step. (c) prediction specificity (IC of predicted GO terms) per condition. The tool-using agent predicts higher-IC, more specific terms than either zero-shot baseline. (d) BMA-Lin score comparision across zero-shot and tool calling agent (e) SimGIC scores across zero-shot and tool calling agent.

4. Results

4.1. Tool Access Improves GO Annotation Over Zero-Shot LLM Baselines

We compare three conditions on the evaluation set with 2 replicates for every protein: (i) Qwen3-8B zero-shot; (ii) Qwen3-14B zero-shot and (iii) Qwen3-8B with full tool access but without any training. Figure 2 reports SimGIC and BMA-Lin across MF, BP, CC, and Overall.

The tool-using 8B agent outperforms the zero-shot 8B baseline across all four groups. Performance relative to the zero-shot 14B model is mixed; the larger model performs better in some aspects, but the tool-using 8B agent matches or exceeds it in others, particularly CC category.

4.2. GRPO Training Improves GO Term Prediction

Figure 4 evaluates the effect of GRPO training under the standard task, in which the agent is prompted to predict GO terms for a given protein sequence. We compare three conditions, all using Qwen3-8B: (i) zero-shot, (ii) base agent with tools but no training, and (iii) the GRPO-trained agent. The GRPO-trained agent consistently outperforms both the zero-shot model and the untrained tool-using agent on SimGIC and BMA-Lin across all three GO aspects, confirming that the reward signal drives genuine improvement in annotation quality beyond what tool access alone provides.

4.3. GRPO Training Generalises to Domain-Annotated GO Prediction

Figure 5 evaluates the same three conditions under a harder task: rather than predicting a flat set of GO terms, the agent is prompted to predict GO terms *and* associate each predicted term with a specific protein domain or functional region. This forces the agent to localize function to a specific region.

The ordering of conditions is preserved: the GRPO-trained agent outperforms the base agent, which outperforms the zero-shot model, on both SimGIC and BMA-Lin. The magnitude of improvement from GRPO training is comparable to that observed on the standard task, indicating that the policy learned during training generalises to the domain annotation setting without task-specific fine-tuning.

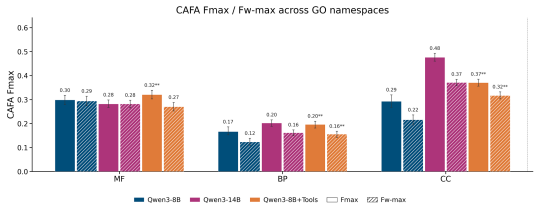

Figure 3. Evaluation under CAFA5 metrics. During training, we used SimGIC and BMA-Lin semantic similarity scores as reward signals because they assign partial credit for semantically related but non-identical predictions (e.g., "zinc finger DNA binding domain" versus "DNA binding domain"), which provides a denser learning signal than exact-match metrics. However, the standard benchmark for protein function prediction is the CAFA F-score. To enable direct comparison with prior work, we additionally evaluate our models on the CAFA metrics.

4.4. Tool Ablations

To characterise the marginal contribution of each tool we run leave-one-out ablations, removing a single tool from the agent's environment at evaluation time and measuring Δ BMA-Lin relative to the full-tool baseline. No single tool dominates: ablating any one tool produces small drops that are largely within the bootstrap confidence intervals. The relative importance of tools varies by GO aspect — motif and binding-likelihood tools have the largest average effect on MF; transmembrane topology tools matter most for CC; bond and contact analysis contributes most to BP — but none of these differences is statistically conclusive at the sample sizes available.

One notable pattern emerges when GO ontology navigation tools (GO search, GO explore) are ablated: overall SimGIC and BMA-Lin scores decrease, but prediction specificity increases. We interpret this as a precision–recall trade-off: without GO navigation the agent submits fewer, higher-IC terms, reducing recall while improving specificity.

4.5. Comparison to BioReason-Pro

We evaluated our agent on a subset of the test dataset provided by the BioReason-Pro Data Center benchmark (Fallahpour et al., 2026), using the same evaluation metrics reported in that work. To enable direct comparison, we reformulated our agent around sandbox execution tools, allowing it to be extended with new tools while still operating within a controlled environment. Unlike in the previous sections, the agent in this setting has access only to the data files that determine the properties of each instance; the remainder of the analysis code is generated and executed by the agent itself across eight independent replicates, since no reference inference code is provided. To aggregate predictions across replicates, we use a consensus-style voting system to determine the weights of the CAFA evaluation.

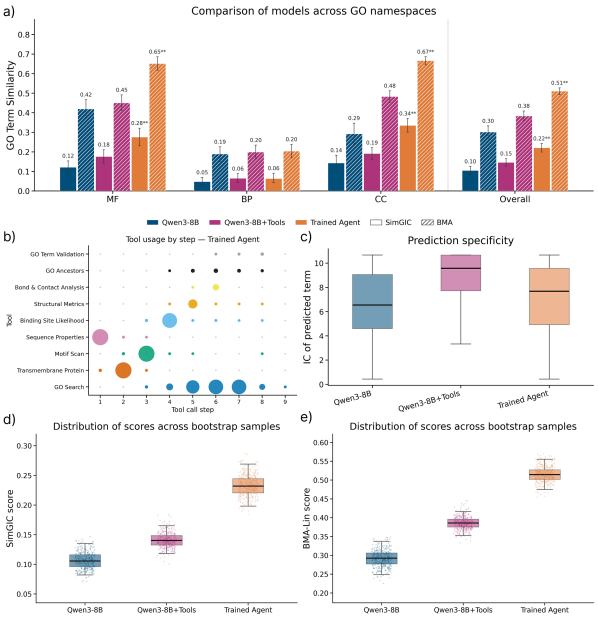

Figure 4. GO semantic similarity for zero-shot Qwen3-8B, base agent (tools, no training), and GRPO-trained agent on the standard GO term prediction task. Bars show SimGIC and BMA-Lin per aspect (MF, BP, CC) and Overall, with 95% bootstrap CIs. (b) Tool call frequency at each step across all trajectories (c) Prediction specificity of the output GO Terms. (d) SimGIC score distribution showing significant improvement in performance for the GRPO agent (e) BMALin score distribution showing significant improvement in performance for the GRPO agent

Figure 8 summarizes tool usage and performance across the eight replicates. On the Molecular Function (MF) namespace, BioReason-Pro reports a score of approximately 0.79 after RL fine-tuning, while our agent achieves 0.50. On BP and CC namespaces, BioReason-Pro reports 0.59 and 0.82, compared to 0.25 and 0.59 from our agent.

5. Discussion

Models such as BioReason-Pro (Fallahpour et al., 2026) represent the current state of the art on CAFA style GO annotation. They are multimodal predictors: a protein language model backbone provides rich pretrained sequence representations, reasoning traces from a larger model supply teacher distilled signal, and the full feature set you could get from tools is presented to the model in a single forward pass. This is a powerful design and the reported Fmax values reflect it, GOAgent does not approach these numbers. However, GOAgent does not use any underlying protein language model embeddings and reasons purely using the available tool information and the sequence. BioReason-Pro benefits from a domain-specialized GO-GPT module trained on top of the base protein language model, as well as RL fine-tuning targeted at related metrics. In contrast, our agent relies purely on general-purpose tool calling without any

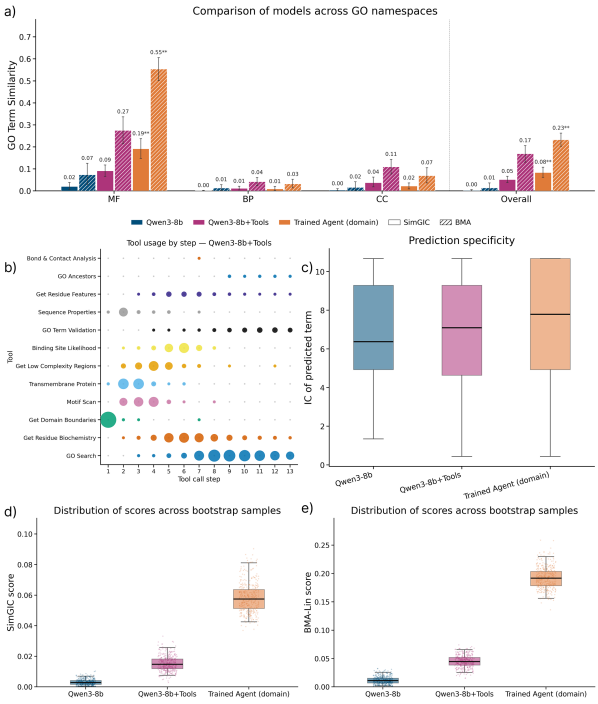

Figure 5. GO semantic similarity on the domain-annotated prediction task, in which the agent must associate each predicted GO term with a specific protein domain. The GRPO-trained agent maintains its advantage under the more demanding annotation setting.

task-specific training or auxiliary GO models. That a tool-using agent can approach these specialized scores suggests that much of the performance gain in BioReason-Pro can be attributed to learned domain priors rather than reasoning capability alone, and that sandboxed tool execution is a viable, extensible alternative when training a domain-specific model is impractical. A further advantage of this design is that any instruction-tuned model with tool-calling support is a valid backbone for GOAgent. As stronger frontier models become available, swapping them in is a configuration change — the tool layer remains the same and continues to provide signal on top of whatever reasoning model sits underneath. Training would further improve the performance of the agent.

GOAgent's ceiling is bounded by the quality and coverage of the available tools rather than the backbone itself, and errors or gaps in motif libraries, binding-site predictors, or structure-derived features propagate directly into predictions. GRPO training was performed on a relatively small set of proteins; larger training sets and longer rollout horizons may close some of the gap to backbone-bound models, though we noticed a degradation in reasoning trace quality on longer horizons. Single-tool ablation effects are also noisy, so we do not make strong claims about which individual tools are essential.

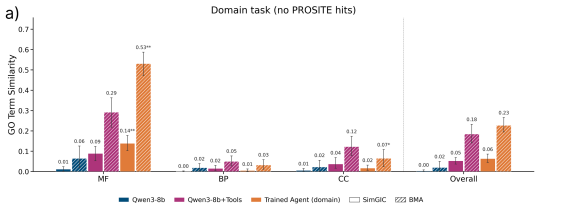

Figure 6. Ablation controlling for PROSITE tool usage. Because the domain annotation agent has access to PROSITE and related domain-boundary recognition tools, a natural concern is that the observed learning gains arise trivially from the agent querying PROSITE and copying its domain assignments. To rule this out, we filtered out all trajectories in which the PROSITE tool was invoked at any point and recomputed the evaluation metrics on the remaining subset. The trained agent continues to outperform the baseline even without PROSITE access, indicating that its improvements reflect genuine domain-recognition capability rather than direct lookup of PROSITE annotations.

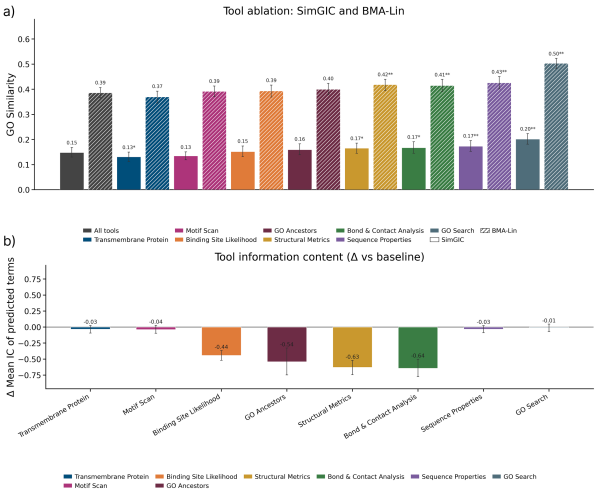

Figure 7. Tool ablation results. Each bar shows Δ BMA-Lin and Δ SimGIC (ablated − baseline) for removing one tool, with 95% bootstrap CIs.

6. Limitations

- GOAgent underperforms embedding-backbone models such as BioReason-Pro on raw GO accuracy. The ceiling is bounded by the quality and coverage of the available tools, not the backbone.
- Errors or gaps in motif libraries, binding-site predictors, or structure-derived features propagate directly into predictions.
- GRPO training was performed on approximately 1,500 proteins. Larger training sets and longer rollout horizons may close some of the gap to backbone-bound models but we noticed a degradation in reasoning trace quality on longer horizons.
- **Ablation noise.** Single-tool ablation effects are noisy,

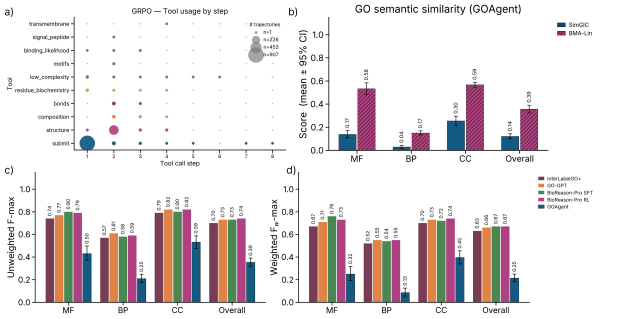

Figure 8. Comparison of our tool-using agent to BioReason-Pro across eight replicates on a subset of the BioReason-Pro benchmark. **(A)** Tool-call usage distribution across replicates, showing which sandbox tools the agent invokes during evaluation. **(B)** GO semantic similarity scores (including SimGIC and related measures) used as reward functions and reported here as evaluation metrics for direct comparison. **(c)** CAFA unweighted F-score under the same evaluation protocol. **(d)** CAFA weighted F-score, computed using the consensus voting scheme across replicates.

so we cannot make strong claims about which individual tools are essential.

Impact Statement

This work presents a tool-augmented framework for protein function understanding that integrates large language models with external bioinformatics tools and databases. By enabling models to reason over explicit biological evidence GOAgent may help improve the interpretability of computational protein analysis workflows. We are not aware of immediate harmful societal consequences specific to this work beyond the broader risks associated with the misuse or overreliance on AI-generated scientific predictions.

LLM Use We used large language models in coding assistants and to refine the formatting and text of this paper.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

Consortium, T. U. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1): D523–D531, 01 2023. ISSN 0305-1048. doi: 10.1093/nar/gkac1052. URL https://doi.org/10.1093/nar/gkac1052.

Fallahpour, A., Seyed-Ahmadi, A., Idehpour, P., Ibrahim, O., Gupta, P., Naimer, J., Zhu, K., Shah, A., Ma, S., Adduri, A., et al. Bioreason-pro: Advancing protein function prediction with multimodal biological reasoning. *bioRxiv*, pp. 2026–03, 2026.

Fan, C., Ma, Z., Meng, H., Zhang, A., Du, W., Zhang, J., Gao, Y. Q., Cao, Z., and Fu, G. Interleaved tool-call reasoning for protein function understanding. *arXiv preprint arXiv:2601.03604*, 2026.

Friedberg, I., Radivojac, P., Paolis, C., Piovesan, D., Joshi, P., Reade, W., and Howard, A. Cafa 5 protein function prediction. 2023.

Gane, A., Bileschi, M. L., Dohan, D., Speretta, E., Héliou, A., Meng-Papaxanthos, L., Zellner, H., Brevdo, E., Parikh, A., Martin, M. J., Orchard, S., and Colwell, L. J. ProtNLM: Model-based natural language protein annotation. https://storage.googleapis.com/brain-genomics-public/research/proteins/protnlm/uniprot_2022_04/protnlm_preprint_draft.pdf, 2022. Preprint.

Gazizov, A., Lian, A., Goverde, C., Mou, J., Ovchinnikov, S., and Polizzi, N. F. Af2bind: predicting small-molecule binding sites using the pair representation of alphafold2. *Nature Methods*, pp. 1–10, 2026.

Ghareeb, A. E., Chang, B., Mitchener, L., Yiu, A., Szostkiewicz, C. J., Shved, D., Gyimesi, G. J., Laurent, J. M., Wright, S. M., Razzak, M. T., et al. A multi-agent system for automating scientific discovery. *Nature*, pp. 1–3, 2026.

Guo, H., Huo, M., Zhang, R., and Xie, P. Proteinchat: Towards achieving chatgpt-like functionalities on protein 3d structures. *Authorea Preprints*, 2023.

Hofmann, K., Stoffel, W., et al. Tmbase-a database of membrane spanning protein segments. 1993.

Huo, M., Guo, H., Cheng, X., Singh, D., Rahmani, H., Li, S., Gerlof, P., Ideker, T., Grotjahn, D. A., Villa, E., et al. Multi-modal large language model enables protein function prediction. *bioRxiv*, pp. 2024–08, 2024.

Ibtehaz, N., Kagaya, Y., and Kihara, D. Domain-pfp allows protein function prediction using function-aware domain embedding representations. *Communications Biology*, 6 (1):1103, 2023.

Krapp, L. F., Abriata, L. A., Cortés Rodriguez, F., and Dal Peraro, M. Pesto: parameter-free geometric deep learning for accurate prediction of protein binding interfaces. *Nature communications*, 14(1):2175, 2023.

Kulmanov, M. and Hoehndorf, R. Deepgoplus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 01 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz595. URL https://doi.org/10.1093/bioinformatics/btz595.

Kulmanov, M., Khan, M. A., and Hoehndorf, R. Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34(4):660–668, 02 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx624. URL https://doi.org/10.1093/bioinformatics/btx624.

Lála, J., O'Donoghue, O., Shtedritski, A., Cox, S., Rodriques, S. G., and White, A. D. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*, 2023.

Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., Kaufman, K. W., Renfrew, P. D., Smith, C. A., Sheffler, W., et al. Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. In *Methods in enzymology*, volume 487, pp. 545–574. Elsevier, 2011.

Lu, C., Lu, C., Lange, R. T., Yamada, Y., Hu, S., Foerster, J., Ha, D., and Clune, J. Towards end-to-end automation of ai research. *Nature*, 651(8107):914–919, 2026.

Lv, Z., Ao, C., and Zou, Q. Protein function prediction: From traditional classifier to deep learning. *PROTEOMICS*, 19(14):1900119, 2019. doi: https://doi.org/10.1002/pmic.201900119. URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/pmic.201900119.

Mitchener, L., Yiu, A., Chang, B., Bourdenx, M., Nadolski, T., Sulovari, A., Landsness, E. C., Barabasi, D. L., Narayanan, S., Evans, N., et al. Kosmos: An ai scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.

Novakovsky, G., Dexter, N., Libbrecht, M. W., Wasserman, W. W., and Mostafavi, S. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics*, 24(2):125–137, 2023.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., Sokolov, A., Graim, K., Funk, C., Verspoor, K., Ben-Hur, A., et al. A large-scale evaluation of computational protein function prediction. *Nature methods*, 10(3):221–227, 2013.

Sanderson, T., Bileschi, M. L., Belanger, D., and Colwell, L. J. Proteinfer, deep neural networks for protein functional inference. *elife*, 12:e80942, 2023.

Sigrist, C. J., Cuche, B. A., de Castro, E., Coudert, E., Redaschi, N., and Bridge, A. The prosite database for protein families, domains, and sites. *Nucleic Acids Research*, 54(D1):D451–D458, 2026.

Stevens, S. Biobench: A blueprint to move beyond imagenet for scientific ml benchmarks. *arXiv preprint arXiv:2511.16315*, 2025.

Teufel, F., Almagro Armenteros, J. J., Johansen, A. R., Gíslason, M. H., Pihl, S. I., Tsirigos, K. D., Winther, O., Brunak, S., von Heijne, G., and Nielsen, H. Signalp 6.0 predicts all five types of signal peptides using protein language models. *Nature biotechnology*, 40(7):1023–1025, 2022.

Torres, M., Yang, H., Romero, A. E., and Paccanaro, A. Protein function prediction for newly sequenced organisms. *Nature Machine Intelligence*, 3(12):1050–1060, 2021.

Wang, C., Fan, H., Quan, R., and Yang, Y. Protchatgpt: Towards understanding proteins with large language models, 2025a. URL https://arxiv.org/abs/2402.09649.

Wang, W., Shuai, Y., Zeng, M., Fan, W., and Li, M. Dpfunc: accurately predicting protein function via deep learning with domain-guided structure information. *Nature communications*, 16(1):70, 2025b.

Xiao, H., Lin, W., Chen, X., Wang, H., Chen, K., Li, J., Sun, Y., Dai, S., Wu, B., and Ye, Q. Stella: Towards protein function prediction with multimodal llms integrating sequence-structure representations. *arXiv preprint arXiv:2506.03800*, 2025a.

Xiao, Y., Sun, E., Jin, Y., Wang, Q., and Wang, W. Proteingpt: Multimodal llm for protein property prediction and structure understanding, 2025b. URL https://arxiv.org/abs/2408.11363.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., Lewis, K. A., Georghiou, G., Nguyen, H. N., Hamid, M. N., et al. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, 20(1):244, 2019.