EVIREAD: AI AGENTS LEARN TO READ BIOLOGICAL EVIDENCE THROUGH RECURSIVE IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

AI agents have made substantial progress in software engineering and mathematics, building on language models strengthened by data and model scaling. The same agent wave has also reached biology, with the hope of combining agents with bioinformatics tools to accomplish downstream tasks more intelligently and automatically. A gap remains between them, however: the outputs of biological tools, such as biological language models (BioLMs), do not automatically translate into better evidence an agent can read and reason over. We introduce EviRead, a two-level framework that turns this heterogeneous biological evidence into agent-readable evidence through recursive improvement (RI). We instantiate EviRead on TempoGO-HQ, a retrospective temporal benchmark of 1,389 proteins for Gene Ontology term prediction. The BioLM-only workflow EviRead$_{base}$ runs five RI rounds and improves on both the training and validation splits, showing that the loop can diagnose and iteratively refine the workflow. The second, tool-augmented workflow EviRead$_{+tools}$ integrates sequence alignment, structural search, and GO predictors, and its final workflow outperforms all twelve compared methods in ontology-macro $F_{\max}$. By making evidence construction and tool coordination learnable at the workflow level, EviRead offers a concrete route to connect specialized biological models and tools with agent-based inference.

1 INTRODUCTION

AI agents have made striking progress across tasks, especially in software engineering and mathematics: they modify repositories, execute programs, and search for formally verifiable proofs (Yang et al., 2024; Hubert et al., 2026). Recent advances in recursive self-improvement (RSI) show that agents can improve their own task-solving programs (Zhang et al., 2025a), while automated workflow search has produced workflows that outperform the evaluated human-designed alternatives (Zhang et al., 2025b). Scaling training data and model capacity further strengthens the language models underlying these agents (Hoffmann et al., 2022).

This trend has also permeated biology, which has a wealth of bioinformatics tools and biological models. Large biological models like the ESM and Evo families learn representations from large collections of protein and nucleotide sequences, capturing relationships within and across biological sequences (Rives et al., 2021; Lin et al., 2023; Brixi et al., 2026), and these models support downstream tasks such as mutation-effect prediction (Meier et al., 2021), protein design (Madani et al., 2023), and protein function prediction (Xu et al., 2023; Su et al., 2025); sequence aligners, structural search, and GO predictors complement them, each producing a specialized output. However, these tools were built for scientists and downstream pipelines rather than for agents: an embedding, an alignment score, or a ranked list of GO terms is not yet a claim an agent can evaluate. An interface is therefore needed that turns such heterogeneous outputs into evidence an agent can read and reason with.

Protein function prediction is a particularly suitable setting for building this interface. Given a protein, the task is to infer its Gene Ontology (GO) labels from available sequence information and functional evidence. Biological databases contain diverse annotations for related proteins, but these records vary in completeness, specificity, and evidence support (The UniProt Consortium, 2025; Gene Ontology Consortium, 2026). The same protein may have a broad family description in one source, a specific activity annotation in another, and functions that remain unannotated (Dessimoz

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

et al., 2013; Zhou et al., 2019). These textual records are a rich but uneven signal, ranging from curated literature-based annotations to noisy, computationally inferred ones; such text becomes useful to an agent only when it is linked with the structured outputs of biological models and tools (embeddings, alignment scores, structural matches), turning scattered database records into evidence an agent can read and weigh. These demands make protein function prediction a natural testbed for agents that combine information and reason under uncertainty.

We introduce the **Evi**dence **Read**ing framework (EviRead), a paradigm in which agents use RI to discover how to read and use heterogeneous biological evidence. Instead of fixing the translation by hand, EviRead lets the agent revise programs that extract signals from biological models and tools, express them as textual evidence, and use that evidence for GO prediction. The framework combines an inner function-prediction agent with an outer recursive improvement loop. The outer agent diagnoses development errors, revises the evidence-construction and decision programs, and evaluates the resulting workflow. Biological models and reference resources remain frozen, and final-test feedback is excluded from recursive improvement. The resulting program is reused across proteins: the agent learns how to read biological information, rather than only revising its answer to one query.

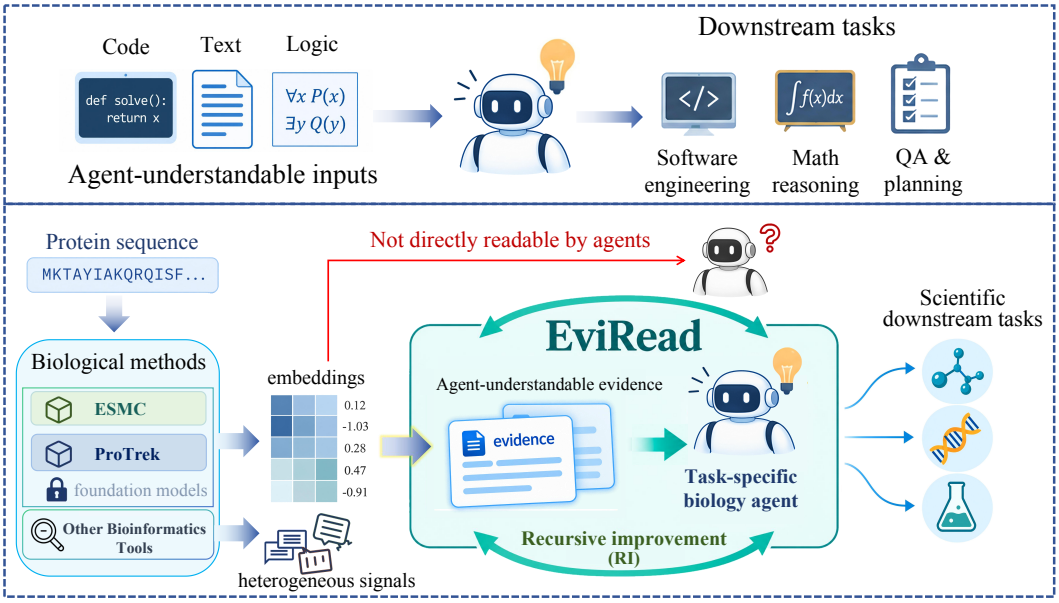

Figure 1: Motivation for EviRead. Biological models and tools are specialized and heterogeneous: they encode protein sequences as embeddings, similarity scores, and functional predictions. Unlike text, logic, and code, which AI agents understand natively, these outputs are not directly readable by agents. EviRead forms the interface and uses recursive improvement to learn to interpret this heterogeneous biological evidence, structuring it into an agent-readable form for downstream tasks.

Following LAFA's benchmark-construction strategy and CAFA-style temporal evaluation principles (Radivojac et al., 2013; Phan et al., 2026), we build an evaluation set and apply a quality-control screening, yielding TempoGO-HQ, a retrospective temporal benchmark of 1,389 proteins. We study two workflows to test whether more sources of evidence help the agent reason and produce better function predictions. EviRead$_{base}$ keeps only the three biological models (ESM2, ESMC, and ProTrek) and asks whether the RI loop can diagnose and iteratively improve the evidence-reading workflow over five rounds. EviRead$_{+tools}$ mounts a tool layer (sequence alignment, structural search, and GO predictors) and asks whether the repaired workflow can coordinate heterogeneous evidence to outperform the individual methods. A donor-transfer analysis further separates high-quality from all GO annotations.

Our contributions are threefold:

- Framework. We propose EviRead, which turns heterogeneous biological evidence (frozen model and tool outputs) into agent-readable evidence and uses an RI loop (capable of repairing its own mechanism) to improve evidence construction and use.
- Benchmark and evaluation. We construct TempoGO-HQ, a quality-controlled 1,389-protein retrospective benchmark with a two-setting donor-transfer protocol that separates high-quality from all GO annotations.
- Application and validation. We apply EviRead to protein function prediction. A model-only workflow improves over five RI rounds, and a tool-augmented workflow, repaired across a multi-round campaign, outperforms the twelve compared methods.

2 Related Work

2.1 Agents and Recursive Self-Improvement

Modern agents interact with external tools, inspect intermediate states, and use execution feedback in multi-step tasks. SWE-agent makes this pattern explicit for software engineering: the agent edits a repository, runs commands, observes failures, and revises its actions (Yang et al., 2024); formal reasoning systems similarly connect model proposals to verifiable environments such as proof checkers (Hubert et al., 2026). Recent work extends this loop to the agent's own program: AFlow searches agentic workflows using task feedback, while the Darwin Gödel Machine revises its own programs and retains changes that survive evaluation (Zhang et al., 2025b;a). Unlike extra test-time inference, RSI changes a reusable procedure inherited by later predictions (Snell et al., 2025).

The same composition principle is entering scientific workflows. Agents now coordinate planning with domain tools for genetic perturbation experiments (Roohani et al., 2024), execute general biomedical analyses (Huang et al., 2025), decompose discovery across collaborating roles (Ghareeb et al., 2026), support hypothesis generation and evaluation (Gottweis et al., 2026), and write expert-level empirical software (Aygün et al., 2026). These systems orchestrate tools and roles, but they do not address how an agent should read the heterogeneous outputs of biological models and tools. EviRead targets this interface and makes its construction an object of RI.

2.2 Protein Function Prediction

Protein function prediction has developed along several technical routes, with Gene Ontology (GO) term prediction as one of the most common instantiations. Classical pipelines transfer functions from homologous or structurally similar proteins, using sequence searches to retrieve related donors and Foldseek to extend retrieval to structural similarity (van Kempen et al., 2024); transfer quality depends on donor selection and annotation specificity.

Deep learning adds complementary routes. DeepGOPlus combines sequence convolution with similarity evidence, and DeepFRI predicts function from structure-aware graph representations (Kulmanov & Hoehndorf, 2020; Gligorijević et al., 2021). Protein language models learn sequence representations at scale and transfer them to function, mutation-effect, and design tasks (Rives et al., 2021; Lin et al., 2023; Meier et al., 2021; Madani et al., 2023), and strong systems often combine these sources. Evaluation has evolved in parallel: CAFA compares predictions with later functional annotations and stresses the open-world nature of protein functions (Radivojac et al., 2013; Dessimoz et al., 2013; Zhou et al., 2019); CAFA-evaluator provides ontology-aware scoring, and LAFA emphasizes reproducible longitudinal assessment (Piovesan et al., 2024; Phan et al., 2026). Our model-only and tool-augmented tracks separate gains from model signals, conventional evidence, and RI.

2.3 Biological Foundation Models and Agent-Readable Evidence

Alongside the rapid progress of language models, biological foundation models have developed in parallel, learning representations from large sequence and structure collections. ESM models and ESM Cambrian establish scaled protein encoders, and ProTrek extends them toward multimodal protein search (Rives et al., 2021; Lin et al., 2023; EvolutionaryScale, 2024; Su et al., 2025). To bridge biology and language, ProtST and BioMedGPT connect biological inputs with text, Cell2Sentence adapts language modeling to cellular states (Xu et al., 2023; Luo et al., 2023; Levine et al., 2024),

and Prot2Text and Protein2Text generate functional descriptions directly from protein representations (Abdine et al., 2024; Jararweh et al., 2025).

These generation methods make biological information easier for language models to consume, but they do not define an evidence interface for an agent: a generated description may omit provenance, conflate correlated observations, or fail to distinguish a candidate function from supporting evidence. EviRead differs in that it does not train a new decoder; it relies on the agent's own reasoning to establish the connection between heterogeneous biological evidence and decision-making. It converts frozen model and tool outputs into agent-readable evidence with provenance and limitations, and uses RI to improve how the agent selects, organizes, and applies that evidence to GO prediction.

3 METHOD

3.1 TASK, DATASET, AND METRICS

Task. We adopt Gene Ontology (GO) term prediction as our protein function prediction task. GO organizes functions into a hierarchy rooted at three broad aspects (molecular function (MF), biological process (BP), and cellular component (CC)) and moving from an ancestor toward a descendant, the functional description becomes progressively more specific: for example, "catalytic activity" generalizes "kinase activity". As a machine-learning problem, the task is multi-label classification over this ontology: the input is a protein sequence (and optionally a structure), and the output is a set of GO terms, each with a confidence score in $[0, 1]$. We study a temporal formulation of this task, following LAFA (Phan et al., 2026): given only the information available at a cutoff time T_0, a system predicts the GO labels that become available at a later time T_1 and were not known at T_0. Because biological databases annotate proteins incompletely and with varying specificity, a prediction and a Gold label often differ in granularity. A prediction is therefore judged after both sets are expanded by upward ontology closure, so that a specific prediction still matches a more general Gold ancestor (Figure 2).

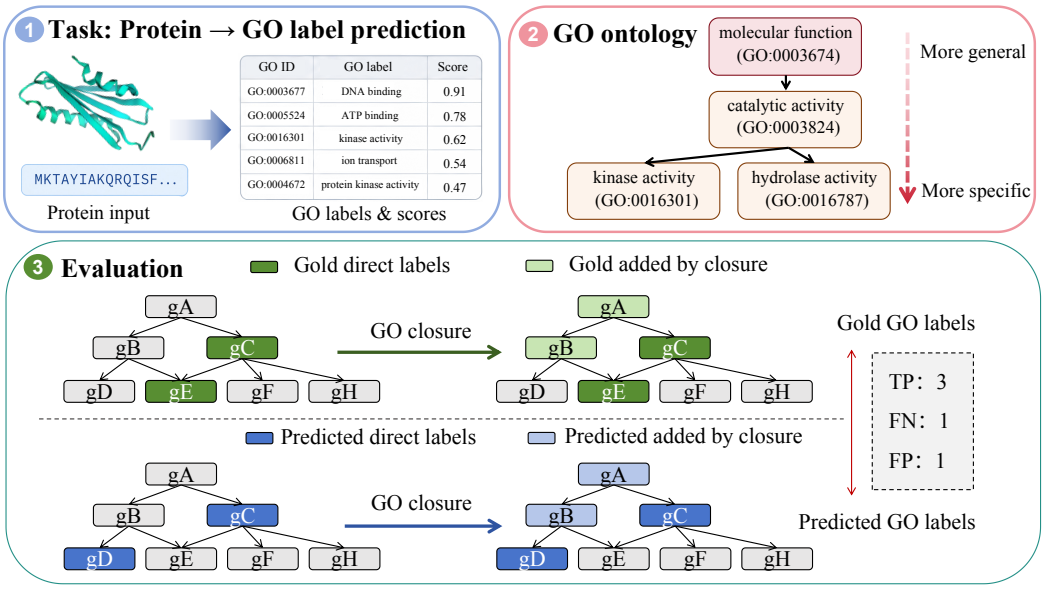

Figure 2: The GO prediction task and its evaluation. (1) The task takes a protein and predicts a set of GO terms, each with a confidence score. (2) GO organizes functions as a directed acyclic graph (DAG), in which general ancestors contain specific descendants and functions become more specific toward the leaves. (3) Evaluation expands both Gold and predicted terms by upward closure, so a specific prediction still matches a general Gold ancestor; precision and recall are computed over the overlap of the two closed sets, and the per-aspect protein-centric F_1 is scanned over thresholds and averaged into the ontology-macro $F_{\max}$.

Dataset. We construct the benchmark TempoGO-HQ, following the LAFA benchmark construction strategy. The T_0 boundary follows LAFA and uses the September 4, 2025 snapshot (UniProtKB/Swiss-Prot 2025_03 with the GO 2025-07-22 ontology), and the T_1 boundary uses the July 26, 2026 snapshot (UniProtKB/Swiss-Prot 2026_02 with the GO 2026-07-26 ontology). We select proteins from UniProt and apply a quality-control screening procedure (Appendix A), yielding 1,389 anonymous protein targets across 1,417 protein-aspect scoring cases and 10,894 Gold GO pairs. We further hold out 50 proteins as a training set for method development and 60 proteins as a validation set, balancing the number of Gold GO terms across MF, BP, and CC as much as possible.

Metrics. We report four metrics (Table 4 in Appendix D) and use the ontology-macro $F_{\max}$ as the primary metric. For each aspect $a \in \{\mathrm{MF}, \mathrm{BP}, \mathrm{CC}\}$, protein-centric precision $P_a(\tau)$ and recall $R_a(\tau)$ are computed at each threshold $\tau \in \{1.00, 0.99, \ldots, 0.01\}$ after upward closure, and $F_{\max}$ averages the aspect-wise maximal F_1:

$$F_{\max} = \frac{1}{3} \sum_{a \in \{\mathrm{MF},\mathrm{BP},\mathrm{CC}\}} \max_{\tau} \frac{2P_a(\tau)R_a(\tau)}{P_a(\tau) + R_a(\tau)}. \tag{1}$$

3.2 EviRead: A two-level framework

EviRead has two levels (Figure 3). The inner level is a function-prediction agent that converts heterogeneous biological evidence (frozen model and tool outputs) into evidence an agent can read and then predicts GO labels. The outer level is a recursive improvement loop that treats the whole inner workflow as a versioned program and revises it; the loop also repairs its own mechanism when it encounters a bug, which makes EviRead a controlled form of recursive improvement. We describe the inner workflow first, then the improvement loop.

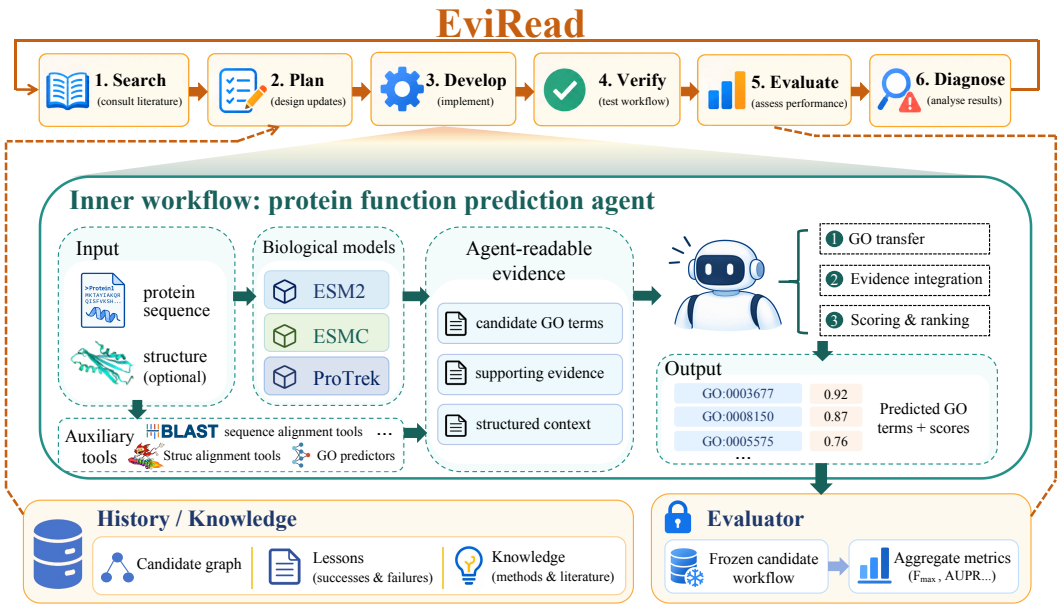

Figure 3: Overview of EviRead. EviRead is a two-level framework. The inner workflow is a protein-function-prediction agent: an anonymous protein sequence (and optional structure) is read by frozen biological models (ESM2, ESMC, ProTrek), whose outputs are converted into an agent-readable evidence layer. The agent integrates this evidence to predict GO labels, with auxiliary tools such as sequence/structure alignment and GO predictors supplying supplementary evidence. The outer loop performs recursive improvement over the evidence-construction and decision programs through Research, Plan, Develop, Verify, Evaluate, and Diagnose stages. It is guided by an append-only history and a frozen evaluator that reports aggregate metrics such as $F_{\max}$.

3.2.1 THE FUNCTION-PREDICTION AGENT

Inner workflow. Given an anonymous protein (its sequence, and optionally a predicted structure), three frozen biological models (ESM2, ESMC, and ProTrek) read it and emit typed observations: each reports the most similar annotated T0 proteins together with their similarity and retrieved GO annotations. These raw observations are not yet something an agent can reason over. EviRead therefore compiles them into a two-sided evidence object. The machine-readable side keeps the hashes, donor identities, and candidate GO terms; the readable side is what the agent sees: a compact list of candidate functions, supporting donors, conflicts, and stated limitations. The agent reasons over the readable side, while every claim stays traceable to the machine record.

The host then turns this evidence into GO candidates and keeps the final decision under its own control. For each aspect (MF, BP, CC) it groups correlated support, prevents the same donor from being counted repeatedly, applies the ontology closure, and selects, rejects, or abstains on each candidate. Only after this locked GO view is built does the workflow produce the explanation, through two language-model stages: the first states the most likely function with qualitative confidence and citations, and the second flags over-specificity, double counting, missing citations, and unsupported claims, within a bounded revision budget. The output is a standard GO prediction surface plus an auditable explanation. In the augmented setting, registered tools such as sequence and structure alignment or GO predictors contribute additional evidence through the same interface.

Two designs. We evaluate two workflows. EviRead$_{base}$ keeps the inner agent connected to ESM2, ESMC, and ProTrek only, and the RI loop runs over this model-only workflow; the quantity of interest is the five-round trajectory of training and validation scores. EviRead$_{+tools}$ mounts the declared tool layer (sequence alignment, structural search, and GO predictors) on top of EviRead$_{base}$, and the RI loop additionally repairs the framework itself across a multi-round campaign; the quantity of interest is the final workflow's score against the twelve methods. Section 4 reports both workflows, together with a donor-transfer analysis that separates high-quality from all GO annotations.

3.2.2 RECURSIVE IMPROVEMENT OF THE WORKFLOW

Outer framework. The outer loop searches over versions of the inner workflow (Figure 3, top). It runs six stages in a cycle. *Research* asks for a general, process-level improvement (for example, a better way to merge the evidence channels) using only permitted sources. *Plan* turns one idea into a falsifiable change: a hypothesis, the expected effect, a control, and a rollback condition. *Develop* implements the change in an isolated worktree where only the allowed evidence-reading code can be edited. *Verify* runs the software checks (type checking, tests, and resource-binding validation) so a broken edit never reaches evaluation. *Evaluate* measures the new version on the development split under the frozen evaluator, and *Diagnose* classifies the remaining errors to start the next cycle. A version is kept only if it improves the objective without breaking guardrails; otherwise the loop rolls back and the attempt remains in the history.

Two boundaries keep the loop honest. First, the loop may only use T0 information to predict the T1 labels: the biological models, ontology, evaluator, and final Gold are frozen, and the held-out validation and test labels are never used to select a revision. Second, every attempt is versioned and hash-bound, so the lineage of changes, their costs, and their outcomes can be replayed. These boundaries make improvement a controlled search over a reusable program rather than unconstrained experimentation.

RI applies to the framework itself, not only to the inner predictor. The improvement loop is also a versioned program: when it encounters a bug (a broken verification step, an invalid campaign binding, or an unhandled failure), it repairs its own mechanism, records the repair, and starts a fresh campaign from the repaired state. The framework therefore becomes more robust as it runs: the outer loop improves the inner workflow, and the loop's own robustness improves through self-repair. This self-repair, together with the versioning and the frozen boundaries, is the sense in which EviRead performs recursive improvement rather than a one-off optimizer.

The improvement loop can edit exactly the evidence-reading decisions of this workflow: which donors to keep, how many neighbors to retain, how to organize and phrase the evidence, how to weight evidence codes, how to select the candidate frontier, and how the synthesis and review stages

phrase their judgments. The biological models, the ontology, the evaluator, and the Gold labels remain frozen.

4 RESULTS

We construct two function-prediction workflows, EviRead$_{base}$ and EviRead$_{+tools}$, and ask whether the agent can make good use of BioLM embeddings and whether integrating more tools and more information improves function prediction. The two workflows differ in the evidence they receive: EviRead$_{base}$ sees only the transferable GO labels carried by the BioLMs, with none of the database's known annotations for the query protein, while EviRead$_{+tools}$ receives all T0-time database information for the query protein plus the tool layer. We first show that the RI loop works (Section 4.1), then compare EviRead$_{+tools}$ against all twelve methods (Section 4.2), and finally analyze how the evidence source affects the results (Section 4.3).

4.1 THE RECURSIVE IMPROVEMENT OF EVIREAD$_{base}$

EviRead$_{base}$ connects the inner agent to ESM2, ESMC, and ProTrek only, and receives only the transferable GO labels carried by these models, without the database's known annotations for the query protein. We run five RI rounds and record the ontology-macro $F_{\max}$ on the training and validation splits (Figure 4). Starting from a weak baseline, the loop produced four candidate workflows; the validation score rose from 0.0700 to 0.2401 over the five rounds, with some fluctuation as the loop explored different evidence-reading changes. What this section establishes is the mechanism: the loop locates problems, edits the corresponding evidence-reading code, and re-evaluates.

The five-round RI trace. Each round integrates one new piece of information, evaluates the change, and uses the resulting signal to plan the next edit. Green frames mark candidates kept as the best so far; red frames mark candidates rejected after evaluation; gray marks the baseline.

R1 (r09): baseline

integrate three frozen BioLM channels only → train 0.179 / val 0.070. Diagnosis: capability gap ⇒ plan: add a score-fusion/calibration stage.

↓

R2 (i001): rejected

add a score-fusion/calibration stage combining BioLM and homolog scores → train 0.146 ↓ / val 0.093 ↑. Diagnosis: too conservative ⇒ plan: backtrack and retune the global GO threshold.

↓

R3 (i002): kept

adjust the single scalar global GO decision threshold → train 0.201 ↑ / val 0.141 ↑. Diagnosis: both splits improved ⇒ plan: fine-grained threshold search.

↓

R4 (i003): rejected

fine-grained search over the global threshold → train 0.183 ↓ / val 0.072 ↓. Diagnosis: search falsified ⇒ plan: backtrack and robust calibration.

↓

R5 (i004): best

pooled score normalization plus a deterministic threshold → train 0.200 / val 0.240 ↑↑. Diagnosis: large validation gain ⇒ plan: keep as the best workflow.

The trace shows what the loop discovered and how it reasoned. The early rounds tried fusion-based changes (combining BioLM and homolog evidence into a calibrated score), and the later rounds made local parameter adjustments (retuning and searching the global GO decision threshold). In each round the loop read the training and validation scores, diagnosed the gap, and planned a targeted edit; when an edit regressed, it backtracked to the best candidate and tried a different direction. The loop learned from its failures, switching to a robust pooled calibration that lifted the validation score from 0.1414 to 0.2401. Because the model-only setting fixes the three evidence channels, all of these gains come from reading the same evidence better rather than from adding new sources. The conclusion is that the agent can locate its own problems from the training scores and the gap to the compared methods, revise the evidence-reading code accordingly, and improve the metric.

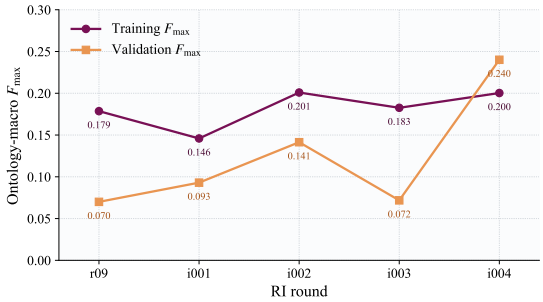

Figure 4: Five RI rounds of the model-only EviRead$_{base}$ workflow on the training and validation splits. The validation $F_{\max}$ rises from 0.0700 to 0.2401 over the five rounds, while the training $F_{\max}$ fluctuates around 0.20.

4.2 EviRead$_{+tools}$ against all twelve methods

EviRead$_{+tools}$ mounts a tool layer (sequence alignment, structural search, and GO predictors) on top of EviRead$_{base}$ and, unlike EviRead$_{base}$, receives all the T0-time database information for the query protein, so the inner agent can request BLASTP, Foldseek, and DeepGOPlus-style evidence in addition to the three model channels. This difference is deliberate: we ask whether more information lets the agent autonomously reason semantically to predict function better and improve itself. Bringing up EviRead$_{+tools}$ required multiple rounds of iteration and bug repair: the workflow reworked how the embedding information is fused and gave more weight to the agent's semantic reasoning, while the loop repaired its own mechanism whenever it hit a bug (Section 3.2.2). The resulting EviRead$_{+tools}$ version achieves a training $F_{\max}$ of 0.6029 and a validation $F_{\max}$ of 0.6493, exceeding every compared method on both splits (Table 1). With the full database evidence and more tools, the workflow becomes robust rather than brittle.

Table 1: The twelve compared methods and EviRead$_{+tools}$ on the TempoGO-HQ training (50) and validation (60) splits. Donor-transfer methods are reported in the all-GO setting (all annotations, including inferred labels).

Method	Train-50 $F_{\max}$	Val-60 $F_{\max}$
GOA nonexp oracle	0.4878	0.5340
MMseqs2-identity	0.5457	0.5646
ProTrek	0.5346	0.5607
BLASTP-identity	0.5261	0.5602
EAT-ProtT5-10NN	0.4878	0.5340
ESMC	0.4971	0.5287
ESM2	0.5109	0.4990
InterLabelGO+	0.4662	0.4603
Foldseek-AFDB-full	0.4710	0.4445
DeepGOPlus	0.4412	0.3727
mDeepFRI GCN	0.3839	0.2808
InterProScan	0.0633	0.1128
EviRead$_{+tools}$	**0.6029**	**0.6493**

4.3 Analysis: the evidence source matters

The donor-transfer methods can either transfer only high-quality GO annotations (the HQ-GO donor setting) or all GO annotations, including computationally inferred labels. We run the same comparison under both settings (Figure 5). When all protein GO labels are considered, every method improves markedly (for example, MMseqs2-identity rises from 0.3610 to 0.5646), showing how strongly the database information shapes the result. EviRead$_{+tools}$ is evaluated under both settings as well: with HQ-GO donors it reaches 0.483, already above every baseline in that setting, and with all GO labels plus the tool layer it reaches 0.6493, again the highest. The agent's gain is therefore consistent across settings: it coordinates the available evidence better in both cases, and more evidence, properly read, leads to better function prediction.

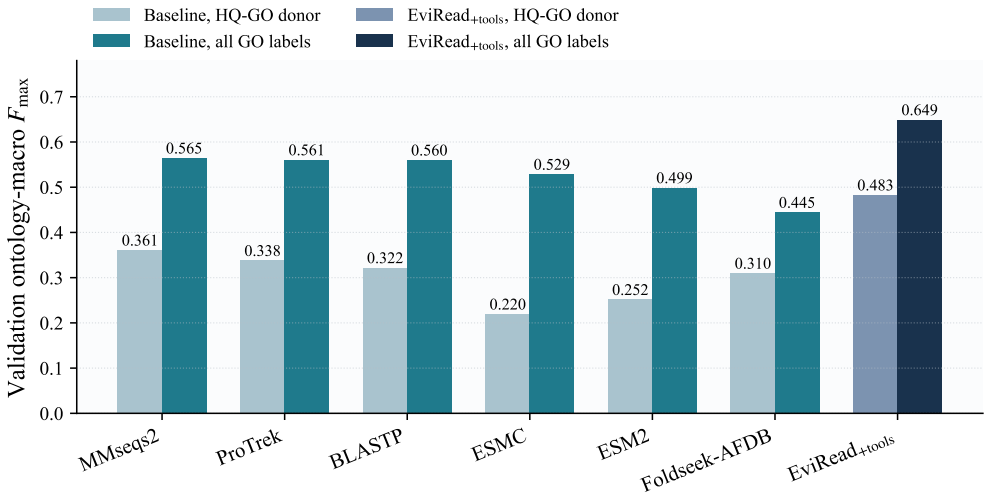

Figure 5: Validation ontology-macro $F_{\max}$ under two evidence settings: HQ-GO donor (light bars) vs. all GO labels (dark bars). Baselines are shown in gray/teal and EviRead$_{+tools}$ in navy. For every method, transferring all GO labels raises the score, showing the influence of the database information. EviRead$_{+tools}$ is the highest under both settings: it reaches 0.483 with HQ-GO donors, already above every baseline, and 0.6493 with all GO labels plus the tool layer.

5 Discussion

We propose EviRead, a framework that connects AI agents with the heterogeneous evidence produced by bioinformatics tools and models: an inner function-prediction agent turns model and tool outputs into readable, traceable evidence, and an outer recursive-improvement (RI) loop revises this evidence-reading procedure itself. We construct TempoGO-HQ, a retrospective temporal benchmark of 1,389 proteins, and apply EviRead to Gene Ontology term prediction. On the model-only EviRead$_{base}$, five RI rounds lifted the validation $F_{\max}$ from 0.0700 to 0.2401, establishing that the RI loop works; on EviRead$_{+tools}$, which adds a tool layer and receives the full database evidence, further iteration produced a workflow that reached 0.6493 and outperformed all twelve compared methods. A comparative analysis across evidence settings then shows how strongly the evidence source shapes prediction accuracy, and why more evidence, properly read, leads to better predictions. Overall, EviRead uses recursive improvement to bridge bioinformatics tools and agents, providing a foundation for more automated biological research, and the same interface can be applied to further downstream biological tasks in future work.

AI USE STATEMENT

We used a generative language-model assistant to draft and edit portions of the manuscript, improve readability, and help organize the methodology and experimental reporting. It also assisted with editing LaTeX and creating or revising figure assets. We did not use generative AI to generate synthetic data, prediction scores, or experimental results, nor to analyze or interpret experimental results. The authors reviewed all AI-assisted content and take responsibility for the final submission.

REPRODUCIBILITY STATEMENT

We will release the TempoGO-HQ dataset and the EviRead codebase, together with the evaluation scripts, the frozen T_0/T_1 resource snapshots, and the prediction outputs used in the paper, upon acceptance. The benchmark construction, the evidence-reading contract, and the evaluation protocol are documented in Appendices A–D, and all primary results are reported on the training (50 proteins) and validation (60 proteins) splits of TempoGO-HQ.

ETHICS STATEMENT

Predicted GO labels are computational hypotheses rather than confirmed biological activities or clinical guidance. Evidence provenance, incomplete annotation, resource differences, and retrospective selection must remain explicit when interpreting our results. Code modifications are limited to authorized modules and cannot alter evaluation resources. The benchmark relies on public sequence and annotation resources (UniProtKB/Swiss-Prot and the Gene Ontology), used under their respective licenses; evaluation labels and target identities remain private.

REFERENCES

Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. Prot2Text: Multimodal protein's function generation with GNNs and transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. URL https://arxiv.org/abs/2307.14367.

Eser Aygün, Anastasiya Belyaeva, Gheorghe Comanici, Marc Coram, Hao Cui, Jake Garrison, Renee Johnston, Anton Kast, et al. An AI system to help scientists write expert-level empirical software. *Nature*, 654(8120):909–916, 2026. doi: 10.1038/s41586-026-10658-6.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, et al. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652(8112):1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

Christophe Dessimoz, Nives Škunca, and Paul D. Thomas. CAFA and the open world of protein function predictions. *Trends in Genetics*, 29(11):609–610, 2013. doi: 10.1016/j.tig.2013.09.005.

EvolutionaryScale. ESM Cambrian: Revealing the mysteries of proteins with unsupervised learning. Official model announcement, December 4, 2024. URL https://www.evolutionaryscale.ai/blog/esm-cambrian.

Gene Ontology Consortium. Guide to GO evidence codes. Official documentation, 2026. URL https://geneontology.org/docs/guide-go-evidence-codes/. Accessed September 14, 2026.

Ali E. Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Dmytro Shved, Gavin J. Gyimesi, Jon M. Laurent, et al. A multi-agent system for automating scientific discovery. *Nature*, 655(8122):497–505, 2026. doi: 10.1038/s41586-026-10652-y.

Vladimir Gligorijević, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12:3168, 2021. doi: 10.1038/s41467-021-23303-9.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 655(8122):487–496, 2026. doi: 10.1038/s41586-026-10644-y.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, et al. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Junze Zhang, Yin Di, et al. Biomni: A general-purpose biomedical AI agent. *bioRxiv*, 2025. doi: 10.1101/2025.05.30.656746. URL https://www.biorxiv.org/content/10.1101/2025.05.30.656746.

Thomas Hubert, Rishi Mehta, Laurent Sartran, et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 651:607–613, 2026. doi: 10.1038/s41586-025-09833-y. First published online November 12, 2025.

Ala Jararweh, Oladimeji Macaulay, David Arredondo, Yue Hu, Luis E. Tafoya, Kushal Virupakshappa, and Avinash Sahu. Protein2Text: Resampling mechanism to translate protein sequences into human-interpretable text. In *Proceedings of NAACL 2025: Human Language Technologies (Volume 3: Industry Track)*, pp. 918–937. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-industry.68. URL https://aclanthology.org/2025.naacl-industry.68/.

Maxat Kulmanov and Robert Hoehndorf. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2020. doi: 10.1093/bioinformatics/btz595.

Daniel Levine, Syed A Rizvi, Sacha Lévy, Nazreen Pallikkavaliyaveetil, David Zhang, Xingyu Chen, Sina Ghadermarzi, Ruiming Wu, Zihe Zheng, Ivan Vrkic, Anna Zhong, Daphne Raskin, Insu Han, Antonio Henrique De Oliveira Fonseca, Josue Ortega Caro, Amin Karbasi, Rahul Madhav Dhodapkar, and David Van Dijk. Cell2Sentence: Teaching large language models the language of biology. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 27299–27325. PMLR, 2024. URL https://proceedings.mlr.press/v235/levine24a.html.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. BioMedGPT: Open multimodal generative pre-trained transformer for BioMedicine, 2023. URL https://arxiv.org/abs/2308.09442.

Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41:1099–1106, 2023. doi: 10.1038/s41587-022-01618-2. URL https://www.nature.com/articles/s41587-022-01618-2.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/f51338d736f95dd42427296047067694-Abstract.html.

An Phan, Yanli Wang, Frimpong Boadu, Jianlin Cheng, Predrag Radivojac, and Iddo Friedberg. LAFA: A framework for reproducible longitudinal assessment of protein function annotation models, 2026. URL https://arxiv.org/abs/2604.20782v2. Version 2, June 10, 2026.

Damiano Piovesan, Davide Zago, Parnal Joshi, M. Clara De Paolis Kaluza, Mahta Mehdiabadi, Rashika Ramola, Alexander Miguel Monzon, Walter Reade, Iddo Friedberg, Predrag Radivojac, and Silvio C. E. Tosatto. CAFA-evaluator: a python tool for benchmarking ontological classification methods. *Bioinformatics Advances*, 4(1):vbae043, 2024. doi: 10.1093/bioadv/vbae043.

Predrag Radivojac, Wyatt T. Clark, Tal Ronnen Oron, Alexandra M. Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, et al. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, 2013. doi: 10.1038/nmeth.2340.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118.

Yusuf Roohani, Andrew Lee, Qian Huang, Jian Vora, Zachary Steinhart, Kexin Huang, Alexander Marson, Percy Liang, and Jure Leskovec. BioDiscoveryAgent: An AI agent for designing genetic perturbation experiments, 2024. URL https://arxiv.org/abs/2405.17631.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.03314.

Jin Su, Yan He, Shiyang You, Shiyu Jiang, Xibin Zhou, Xuting Zhang, Yuxuan Wang, Xining Su, Igor Tolstoy, Xing Chang, Hongyuan Lu, and Fajie Yuan. A trimodal protein language model enables advanced protein searches. *Nature Biotechnology*, 44(8):1310–1316, 2025. doi: 10.1038/s41587-025-02836-0.

The UniProt Consortium. UniProt: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 2025. doi: 10.1093/nar/gkae1010.

Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42:243–246, 2024. doi: 10.1038/s41587-023-01773-0.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. ProtST: Multi-modality learning of protein sequences and biomedical texts. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 38749–38767. PMLR, 2023. URL https://proceedings.mlr.press/v202/xu23t.html.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel machine: Open-ended evolution of self-improving agents, 2025a. URL https://arxiv.org/abs/2505.22954.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=z5uVAKwmjf.

Naihui Zhou, Yuxiang Jiang, Timothy R. Bergquist, Alexandra J. Lee, Balint Z. Kacsoh, Alex W. Crocker, Kimberley A. Lewis, George Georghiou, Huy N. Nguyen, Md Nafiz Hamid, et al. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, 20:244, 2019. doi: 10.1186/s13059-019-1835-8.

A BENCHMARK CONSTRUCTION

Population and identity. The benchmark contains 1,389 unique anonymous protein sequences whose public IDs have no reverse UniProt mapping. Each target is accompanied by a structure predicted with SimpleFold-3B; a structure supplies geometric information only, not a GO label.

T0 resources. The T_0 boundary follows LAFA and uses the September 4, 2025 snapshot: the full UniProtKB/Swiss-Prot 2025_03 release, including all reviewed proteins together with their Gene Ontology annotations and the associated UniProt annotation text, as well as the GO 2025-07-22 ontology and the frozen information-accretion (IA) table. Together these provide the sequence, annotation, and textual evidence that T_0-legal methods may consult.

T1 Gold. The July 26, 2026 snapshot supplies a frozen full UniProt-GOA GAF target extract. Positive evidence codes are EXP, IDA, IPI, IMP, IGI, IEP; HTP, HDA, HMP, HGI, HEP; IC and TAS. Computational and phylogenetic categories such as IEA, ISS, ISO, ISA, ISM, RCA, and IBA do not enter positive Gold. An association qualified with NOT states that a function is absent, so the term is excluded from the positive Gold set. Terms are canonicalized, roots and unresolvable invalid or obsolete terms removed, and `is_a`/`part_of` closure applied. The temporal difference from the T0 known-label closure defines `temporal_gold`. For proteins that already carried annotations at T_0, those already-known terms are masked from both Gold and predictions, so evaluation scores only the newly gained functions. The release contains 1,417 protein–aspect cases and 10,894 target–GO Gold pairs.

Quality screening. The quality of an evaluation set strongly affects how meaningful prediction results are, and GO term prediction is inherently an open-world problem: biological databases are heterogeneous and incomplete, so many proteins carry only partial GO annotations (Dessimoz et al., 2013; Radivojac et al., 2013). We therefore screen the evaluation set so that its Gold annotations are comparatively well-established. After upward closure, we require every Gold term to have at least one more specific, leafward Gold descendant, and a protein is retained only if no Gold term has all of its leafward terms outside the Gold set (Figure 6). The criterion is justified as follows: whenever every Gold term has another Gold term below it, every Gold path can be followed down to a leaf term that has no outgoing edge. Along such a path the function is pinned to a specific, well-studied term, so its GO annotation is more trustworthy than a general term unsupported by any specific descendant. All 1,389 proteins in TempoGO-HQ satisfy this criterion.

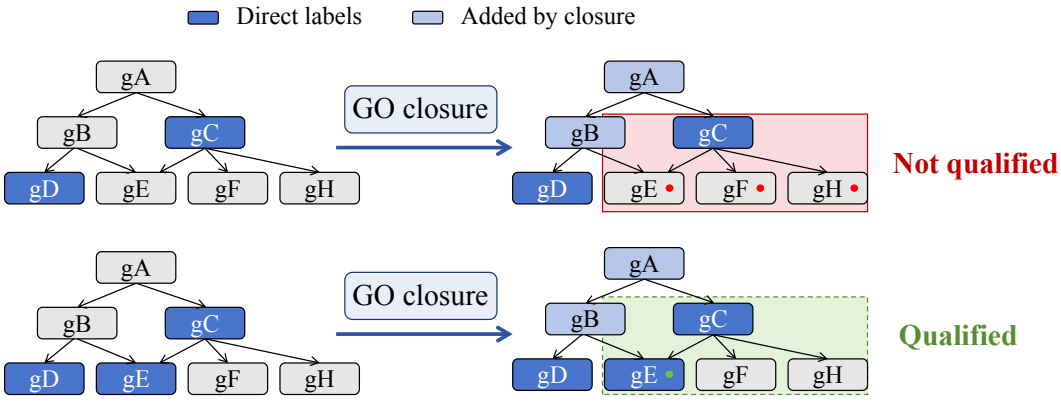

Figure 6: Protein-level quality screening of TempoGO-HQ. For each protein, the Gold GO labels are completed by upward closure, and closure-derived terms are treated as Gold. The protein is qualified only if no Gold label has all of its leafward, more specific terms outside the Gold set; otherwise it is not qualified (top, not qualified; bottom, qualified).

Selection and contamination. Target inclusion depends on T1 Gold availability, so the benchmark is a retrospective, T1-Gold-selected benchmark with this boundary stated explicitly. The documented DeepGOPlus audit finds no evaluation target–GO Gold-pair overlap with its training annotations.

Gold, private target identities, T0/T1 source rows, and selection materials remain outside the agent's accessible resources.

Table 2: Resources and method versions used by TempoGO-HQ. The Released column gives each resource's own version date; methods list their tool version where available and a dash otherwise.

Resource / method	Version	Released
Benchmark resources		
T0 donor annotations (LAFA)	UniProtKB/Swiss-Prot 2025_03	2025-03
T0 GO ontology	2025-07-22	2025-07-22
T0 taxonomy	2025-09-01	2025-09-01
PDB100 structure database	250101	2025-01-01
AFDB-Swiss-Prot structure database	v5	2025-08-24
OMA orthology database	All.Jul2024	2024-07
T1 annotation snapshot	Swiss-Prot 2026_02 + GO 2026-07-26	2026-07-26
Methods		
BLASTP-identity	BLAST+ 2.17.0	—
MMseqs2-identity	MMseqs2 18-8cc5c	—
Foldseek-AFDB-full	—	—
DeepGOPlus	DeepGOPlus 1.0.2	—
mDeepFRI GCN	mDeepFRI v1.0	—
InterLabelGO+	—	—
EAT-ProtT5-10NN	—	—
ESM2, ESMC, ProTrek	—	—
InterProScan	InterProScan 5.75-106.0	—
GOA nonexp oracle	—	—

B Methods, Resources, and Recorded Status

This section supplements the 12 methods with their categories, working principles, and concrete algorithm implementations. Table 3 groups the methods by category, and their evaluation scores are reported in Appendix D.

Table 3: The 12 methods grouped by category and their required input.

Category	Method	Input
Sequence alignment	BLASTP-identity, MMseqs2-identity	Sequence
Structure alignment	Foldseek-AFDB-full	Structure
Deep learning	DeepGOPlus, mDeepFRI GCN, InterLabelGO+	Sequence / Structure / Sequence
Protein language models	ESM2, ESMC, ProTrek, EAT-ProtT5-10NN	Sequence
Signature-based	InterProScan	Sequence
Non-experimental retrieval	GOA nonexp oracle	Protein ID

B.1 Sequence and Structure Transfer

Sequence-transfer baseline tools search an anonymous query against the frozen T0 donor sequences and transfer the direct GO terms of matched donors, scoring each transferred term by local identity or a normalized bitscore. BLASTP-identity and MMseqs2-identity use percent identity divided by 100. The structure-transfer baseline tool Foldseek-AFDB-full searches the query structure against AFDB Swiss-Prot v5 and bridges each hit to a T0 donor through a deterministic MMseqs2 sequence bridge (at least 90% identity and 90% coverage), scoring transferred terms by alignment TM-score times query coverage. No method looks up the target's own accession. Each transfer method is evaluated under two annotation-transfer settings: the primary one follows LAFA and transfers only high-quality GO annotations, while the second transfers all GO annotations, including computationally inferred labels; the comparison isolates the contribution of inferred labels. Algorithms 1 and 2 spell out the two representative transfer protocols.

Algorithm 1: BLASTP-identity transfer.

Require: anonymous query sequence q; frozen T0 donor set $\mathcal{D}$ with direct GO terms; ontology O.
Ensure: prediction scores $s(q, g)$ for GO terms g.
1: **for** each donor $d \in \mathcal{D}$ **do**
2: align q to d with BLASTP; record identity $\mathrm{id}(q, d)$.
3: **for** each direct GO term g of d **do**
4: $s(q, g) \leftarrow \max\big(s(q, g),\, \mathrm{id}(q, d)/100\big)$.
5: **end for**
6: **end for**
7: apply upward closure on O: propagate each term's score to its ancestors by maximum.
8: **return** $\{(q, g, s(q, g))\}$.

Algorithm 2: Foldseek-AFDB-full structure transfer.

Require: anonymous query structure u_q; T0 structure database $\mathcal{S}$; sequence bridge to T0 donors; ontology O.
Ensure: prediction scores $s(q, g)$ for GO terms g.
1: **for** each structure hit $h \in \mathcal{S}$ **do**
2: record TM-score $\mathrm{tm}(u_q, h)$ and query coverage $\mathrm{cov}(u_q, h)$.
3: bridge h to a T0 donor d with sequence identity $\geq 90\%$ and coverage $\geq 90\%$.
4: **for** each direct GO term g of d **do**
5: $s(q, g) \leftarrow \max\big(s(q, g),\, \mathrm{tm}(u_q, h) \cdot \mathrm{cov}(u_q, h)\big)$.
6: **end for**
7: **end for**
8: apply upward closure on O.
9: **return** $\{(q, g, s(q, g))\}$.

B.2 Learned Predictors, Signature Methods, and the Diagnostic Oracle

DeepGOPlus runs its native sequence-CNN plus DIAMOND hybrid predictor and retains the output scores. mDeepFRI GCN builds a single-chain C-alpha contact graph from the target structure and runs the structure-GCN checkpoints for MF, BP, and CC. ESM2, ESMC, and ProTrek are evaluated as standalone retrieval-and-transfer predictors over the frozen T0 donors. InterLabelGO+ combines ESM2-based neural predictions with DIAMOND-driven AlignmentKNN using its published dynamic weighting. EAT-ProtT5-10NN mean-pools ProtT5 embeddings and transfers the direct terms of the ten nearest T0 donors (Algorithm 3). InterProScan supplies canonical GO terms from domain signatures, scored 1.0 without expanding obsolete `consider` suggestions.

The `goa_nonexp_t0_same_accession_oracle` is a diagnostic upper reference: it uses evaluator-private target identities to read the target's own T0 non-experimental annotations (ISS, ISO, ISA, ISM, IGC, RCA, IBA, IBD, IKR, IRD, IEA, ND, NAS), excluding experimental codes, IC/TAS, and NOT, and assigns the retained terms score 1.0. It measures how much future Gold is foreseeable from prior database annotations of the same protein, not anonymous de novo prediction, and stays outside the formal ranking.

Algorithm 3: EAT-ProtT5-10NN transfer.

Require: anonymous query sequence q; frozen T0 donor set $\mathcal{D}$ with direct GO terms; ProtT5 encoder; ontology O.
Ensure: prediction scores $s(q, g)$ for GO terms g.
1: mean-pool ProtT5 residue embeddings for q and every donor $d \in \mathcal{D}$.
2: select the 10 nearest donors $\mathcal{N}_{10}$ by Euclidean distance in embedding space.
3: **for** each GO term g carried by some donor in $\mathcal{N}_{10}$ **do**
4: $s(q, g) \leftarrow \frac{1}{10}\,\big|\{\, d \in \mathcal{N}_{10} : g \text{ is a direct term of } d \,\}\big|$.

5: **end for**
6: apply upward closure on O.
7: **return** $\{(q, g, s(q, g))\}$.

C EviRead Implementation Contract

This appendix supplies the operational details behind Section 3.2 and Section 3.3: the inner prediction episode, the fairness and data-separation guarantees, the language agents, how a campaign is launched and versioned, how broken edits are repaired, and an example prompt. Together these details make the framework's robustness concrete and auditable.

C.1 The inner prediction episode

A single protein passes through seven stages. (1) *Collect fixed evidence*: prepare the anonymous input, run the frozen evidence backend, load the BioLM observations, and collect only registered tool outputs. (2) *Infer GO*: normalize candidates, apply donor and evidence policies, and build the host-controlled GO object with budgets, closure, and score mapping. (3) *Construct bounded views*: keep the full bundle for audit but expose only the selected candidates, leading abstentions, citation-ready evidence, and deterministic disclosures. (4) *Synthesize*: state the narrowest supported function with qualitative confidence, rationale, alternatives, conflicts, limitations, and exact citations. (5) *Critique and revise*: an independent critic checks support, citations, specificity, double counting, and identity leakage, within a fixed revision budget. (6) *Assemble*: inject the locked GO view into the final prediction and write the standard prediction file and reports. (7) *Validate*: recompute hashes and check schemas, evidence IDs, GO decisions, citations, and privacy. A failed validation is an invalid artifact, not a low score.

C.2 Fairness: predicting T1 from T0 only

The benchmark measures a system's ability to predict the T1 labels from T0 information alone. Three mechanisms enforce this. First, the frozen T0 resources (the Swiss-Prot 2025_03 annotations, the GO 2025-07-22 ontology, and the biological-model checkpoints) are the only inputs the predictor may use; the T1 Gold, target accessions, and selection materials live in the evaluator-private area and are never exposed to the predictor. Second, the data is split so that the training set may inform method development, the validation set returns only aggregate metrics, and the test labels remain closed until the workflow is frozen. Third, the RI research stage is audited for leakage: a research question that mentions a target, a GO answer, or Gold terminology is rejected, and only training feedback (never a benchmark protein's answer) may drive a change. These mechanisms make the reported improvement a genuine temporal prediction.

C.3 The three prediction stages

The inner workflow completes each prediction in three stages. *GO decision.* A bounded language-model call reviews candidate cards (parent concept names, donor context, evidence codes, and similarity, but not the GO IDs themselves) and returns accepted, rejected, or abstained for each candidate; the host reconstructs the mapping only after validating the result. *Synthesis.* A language model writes the function narrative from the bounded evidence and GO views. *Review.* A separate call checks the narrative for missing citations, over-specific claims, donor double counting, and promotion of an abstained term. In the current configuration, the GO decision uses a high-reasoning language model, and the synthesis and review stages use structured calls to a general-purpose model; the exact model binding is fixed per campaign. These language-model stages cannot create GO IDs, read the evaluator-private files, or install tools; the host owns input redaction, tool execution, GO inference, Git commits, stage transitions, and final validation.

C.4 LAUNCHING AN RI CAMPAIGN

A campaign starts from a clean, Git-bound repository. Initialization binds a campaign identifier, the objective (here the ontology-macro $F_{\max}$), the development split, the frozen evaluator, the model and resource hashes, the budget, and the change-path policy. Each stage reads a canonical input file and writes a canonical output file, and the controller appends a hash-linked event only after validating the output, so a campaign can resume from the last completed stage.

Versioning works the same way throughout the loop: each candidate is a Git commit in a detached worktree, recorded with its parent, source commit, plan hash, metric, and promotion decision, and the events form an append-only hash chain. A candidate is promoted only after the objective, guardrails, software verification, and resource binding all pass; the previous best is never mutated. A broken edit (a type error, a failing test, or an unbound resource) is routed to a bounded repair path (a fresh Develop–Verify cycle) and recorded; a repair cannot bypass normal verification. A new tool requires a precommit, a private receipt, and classification as required or unused. Before final scoring, a freeze manifest binds the retained workflow, split, evaluator, tools, models, ontology, and outputs, so the final result can be recomputed from the manifest alone.

A campaign is launched with a short prompt; the following shows a complete example (paths are illustrative).

Example prompt. Launch a new campaign and run the full RI loop, optimizing the function-prediction workflow in `<workflow>/`. The objective is to raise the primary metric $F_{\max}$ above the existing methods in `<results>/`. Use `<data>/train_50` as the training set and `<data>/val_60` as the validation set, and run the complete RI loop. The expected outcome is an iterated function-prediction agent workflow with strong performance.

C.5 RESOURCE PLANES AND WORKFLOW ACCOUNTING

The model-only plane provisions ESM2, ESM-C, and ProTrek together with the registered T0 donor interface and the GO vocabulary; it excludes alignment, structure search, orthology, and external predictors. The augmented plane keeps the same target protocol and evaluator while mounting the declared sequence, structure, domain, orthology, and learned-predictor providers. Each provider returns an explicit status and a hash-bound result; an unavailable provider is recorded as unavailable, never as negative biology. A model-only and an augmented result are comparable only after logging the plane, model and index hashes, ontology, split, evaluator, and output receipt. EviRead$_{base}$ is the model-only workflow, and EviRead$_{+tools}$ is the tool-augmented workflow that mounts the declared tool layer on top of it. The two workflows are reported separately: the model-only trajectory demonstrates the RI loop over five rounds, while the tool-augmented workflow is the final function-prediction system compared against the twelve methods.

D PREDICTION NORMALIZATION AND METRIC DETAILS

Every method submits a prediction file that assigns, to each protein, a score in $[0, 1]$ for each GO term it predicts. Before scoring, all predictions are normalized onto the same GO ontology: each term is mapped to its canonical form, and a predicted term automatically also counts for all of its ancestors through upward closure, so a specific prediction still matches a more general Gold term. For proteins that already carried annotations at T_0, those known terms are masked from both predictions and Gold, so the evaluation scores only the newly gained functions.

Table 4: The four evaluation metrics of TempoGO-HQ. Each aspect uses its own optimal threshold, and the macro summary averages the three aspect values.

Metric	Formula	Description
Ontology-macro $F_{\max}$ (primary)	$\frac{1}{3}\sum_a \max_\tau \frac{2P_a(\tau)R_a(\tau)}{P_a(\tau)+R_a(\tau)}$	Protein-centric F_1 maximized over thresholds and averaged across MF, BP, and CC.
AUPR	$\int P\,dR$	Area under the protein-centric precision–recall curve.
IA-w$F_{\max}$	$\frac{1}{3}\sum_a \max_\tau \frac{2\,wP_a(\tau)\,wR_a(\tau)}{wP_a(\tau)+wR_a(\tau)}$	F_1 with each GO term weighted by its information accretion (IA), averaged across aspects.
n$S_{\min}$	$\sqrt{\mathrm{ru}^2+\mathrm{mi}^2}$	Semantic distance: remaining uncertainty ru (IA of missed Gold terms) plus misinformation mi (IA of wrong predictions), normalized per protein by $\mathrm{IA}(\mathrm{Gold}\cup\mathrm{Pred})$.

Table 5 reports the ontology-macro $F_{\max}$ of the 12 methods on the training (50 proteins) and validation (60 proteins) splits, together with the per-aspect validation $F_{\max}$.

Table 5: The twelve compared methods and EviRead$_{+tools}$ on the TempoGO-HQ training (50) and validation (60) splits. Methods are reported in the HQ-GO donor setting; the agent row uses the full tool-augmented (all-GO) evidence. MF/BP/CC are validation-split per-aspect $F_{\max}$. The GOA oracle is a diagnostic upper reference that reads prior same-accession non-experimental annotations.

Method	Train-50 $F_{\max}$	Val-60 $F_{\max}$	$\mathrm{MF}_{\mathrm{val}}$	$\mathrm{BP}_{\mathrm{val}}$	$\mathrm{CC}_{\mathrm{val}}$
GOA nonexp oracle	0.4878	0.5340	0.7014	0.2736	0.6271
InterLabelGO+	0.4662	0.4603	0.6334	0.2815	0.4658
DeepGOPlus	0.4412	0.3727	0.5190	0.2672	0.3319
ProTrek	0.3922	0.3380	0.4824	0.1344	0.3973
MMseqs2-identity	0.3792	0.3610	0.4347	0.2891	0.3593
mDeepFRI GCN	0.3839	0.2808	0.4657	0.1233	0.2533
ESM2	0.3419	0.2517	0.3998	0.1144	0.2409
ESMC	0.2811	0.2196	0.3241	0.1098	0.2250
BLASTP-identity	0.3439	0.3221	0.3775	0.2179	0.3709
Foldseek-AFDB-full	0.3140	0.3099	0.3958	0.2432	0.2906
EAT-ProtT5-10NN	0.3346	0.2248	0.3718	0.1028	0.1998
InterProScan	0.0633	0.1128	0.1977	0.1406	0.0000
EviRead$_{+tools}$	**0.6029**	**0.6493**	—	—	—

EviRead$_{+tools}$ achieves the highest validation $F_{\max}$ (0.6493), exceeding every method on the macro metric. Across the table, BP is consistently the lowest-scoring aspect (every method scores below 0.30 on BP), while MF and CC are generally stronger. This reflects the open-world difficulty of biological-process prediction, where Gold annotations are sparser and more diverse than for molecular function or cellular component.

All 1,389 targets belong to the evaluation population, regardless of how many predictions a method emits; protein–aspect cases follow the release's eligibility, so the 1,417 scoring cases are not three times 1,389. For each aspect, protein-centric precision and recall are computed at thresholds from 1.00 down to 0.01 in steps of 0.01, the maximal harmonic mean is found independently in each aspect, and the three maxima are averaged into the ontology-macro $F_{\max}$.

Table 4 summarizes how the four metrics are computed. AUPR is the area under the threshold-scanned protein-centric precision–recall curve. IA-w$F_{\max}$ weights each GO term by its information accretion, defined as $\mathrm{IA}(g) = -\log p(g)$, where $p(g)$ is the empirical frequency with which term g or one of its descendants appears among the T0 annotations; rarer, more specific terms therefore receive higher weight. $S_{\min}$ combines the remaining uncertainty (the IA of missed Gold terms) with the misinformation (the IA of incorrect predictions), and n$S_{\min}$ normalizes these per protein by the IA of the Gold–prediction union.