rbio1 - training scientific reasoning LLMs with biological world models as soft verifiers

Ana-Maria Istrate, Fausto Milletari, Fabrizio Castrotorres,
Jakub M. Tomczak, **Michaela Torkar**, **Donghui Li**, **Theofanis Karaletsos**
Chan Zuckerberg Initiative, Redwood City, CA, USA
{aistrate, fmilletari, fcastrotorres}@chanzuckerberg.com
{jtomczak,michaela.torkar, dli, tkaraletsos}@chanzuckerberg.com

Abstract

Reasoning models are typically trained against verification mechanisms in formally specified systems such as code or symbolic math. In open domains like biology, however, we lack exact rules for large-scale formal verification and instead rely on lab experiments to test predictions. Such experiments are slow, costly, and cannot scale with computation. In this work, we show that biological world models and other prior knowledge can serve as approximate oracles for *soft verification*, allowing reasoning systems to be trained without additional experimental data. We introduce two paradigms for this process: **RLEMF** (reinforcement learning with experimental model feedback) and **RLPK** (reinforcement learning from prior knowledge). Using these paradigms, we develop **rbio1**, a reasoning model for biology post-trained from a pretrained LLM with reinforcement learning. Soft verification distills biological world models into **rbio1**, which achieves state-of-the-art performance on the PERTURBQA benchmark. We present rbio1 as a proof of concept that predictions from biological models can train powerful reasoning systems using simulations rather than experimental data, offering a new paradigm for model training.

1 Introduction

Foundation models [1–3] have recently shown promise in biology by learning high-dimensional manifolds and producing embeddings useful across tasks [2, 4, 5]. Yet, their utility depends on large annotated datasets and substantial compute. A central challenge is how to transfer the knowledge of large biological foundation models into more flexible and interactive systems, such as conversational reasoning LLMs, that can aid scientific exploration.

Reasoning models such as DeepSeek-R1 [6] and Qwen2 [7] extend LLMs toward multi-step, structured inference, showing remarkable generalization across domains, including medicine [8, 9] and chemistry [10]. These systems are often trained through reinforcement learning with verifiable feedback—such as RLHF [11, 12] or RLVR [13]—where supervision derives from human preference or exact oracles. However, in domains like biology, such exact reward signals are unavailable: experiments are costly, slow, and cannot scale with computation. This motivates exploring *soft verification*, where feedback comes from approximate oracles such as learned models or structured prior knowledge.

The concept of *Virtual Cell Modeling* (VCM) [14–16] envisions predictive systems capable of simulating biological processes, such as transitions from diseased to healthy states. Advances in large-scale modeling have enabled foundation models across biological modalities—transcriptomics [17–

The NeurIPS 2025 Workshop on AI Virtual Cells and Instruments: A New Era in Drug Discovery and Development (AI4D3 2025), San Diego, California, USA, 2025.

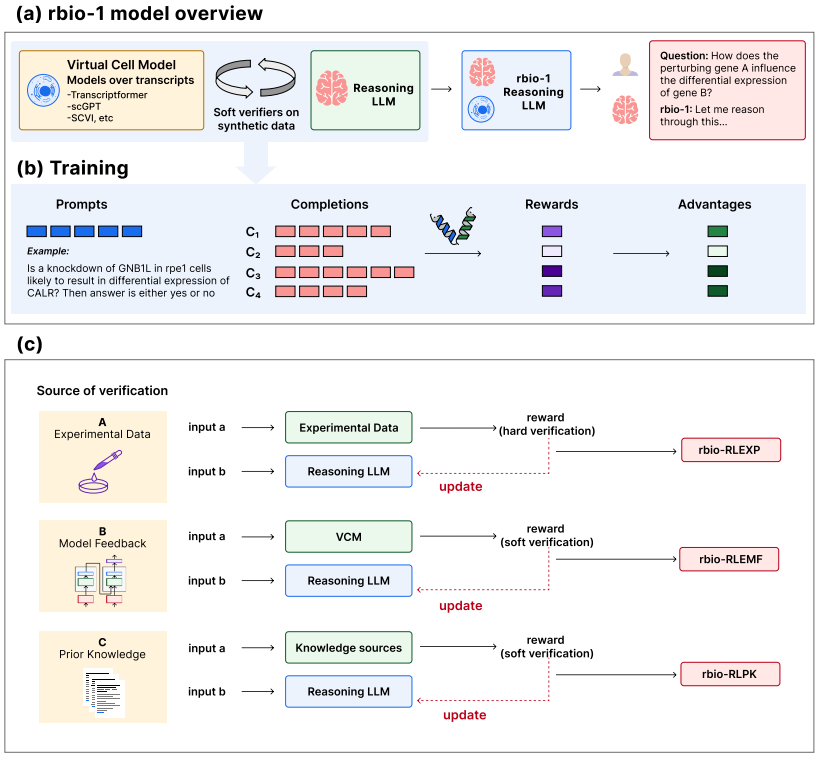


Figure 1: rbio1 overview. (a) Distilling Virtual Cell Models into reasoning LLMs via soft verification. (b) GRPO loop with VCM-based rewards. (c) rbio training paradigms: **rbio-RLEXP**: RL with experimental data; **rbio-RLEMF**: RL with experimental model feedback; **rbio-RLPK**: RL from prior knowledge

21], imaging [22], proteomics [2, 5], genomics [23], and multimodal integration [24–29]. However, integrating these siloed models into a shared, interpretable space remains difficult.

We propose aligning such world models of biology into a common representation using language as the bridge. This alignment aggregates multimodal biological knowledge while making it accessible through natural language—allowing experimentalists to interact conversationally with biological models. By distilling the outputs of world models into LLMs, we convert experimental insights into human-readable reasoning models. Our goal is to reduce dependence on experimental data by learning from *virtual experiments* generated by biological models and structured knowledge sources.

To realize this, we introduce two paradigms for reinforcement learning with approximate biological verifiers: (1) **RLEMF** (Reinforcement Learning with Experimental Model Feedback), where predictive biological models serve as soft verifiers; and (2) **RLPK** (Reinforcement Learning from Prior Knowledge), where knowledge bases such as Gene Ontology guide rewards. Both fit within a broader family of reinforcement learning from AI feedback (RLAIF) [30], adapted to the scientific domain where formal or human supervision is limited. In this setting, verifiers return probabilistic rewards quantifying the coherence of a reasoning trace with biological evidence or prior knowledge. We use these verifiers to train reasoning models through *soft verification*—a process in which simulations and knowledge-based predictions replace hard experimental supervision.

We evaluate this approach on the PERTURBQA benchmark [31], which tests whether models can predict the outcome of gene perturbations—*e.g.*, "Does knocking down gene_A in a given cell line cause differential expression of gene_B?"—without access to experimental data. Traditional models either rely on expensive experimental datasets or finetuning of foundation models. In contrast, our model, **rbio1**, learns through soft verification with AI verifiers and generalizes across cell lines and biological tasks, achieving competitive or state-of-the-art performance.

Our contributions are the following:

1. **Training paradigms for reasoning from simulation.** We introduce two complementary forms of training with approximate verifiers: **RLEMF** (reinforcement learning with experimental model feedback), which uses predictive models of experimental data as probabilistic oracles; and **RLPK** (reinforcement learning from prior knowledge), which uses structured resources such as Gene Ontology as knowledge-based verifiers. Together, these define a general framework for training reasoning models from simulations and structured priors.
2. **rbio1: reasoning through virtual experiments.** We present **rbio1**, a reasoning model for biology trained under the paradigms above. **rbio1** learns to reason about genetic perturbations—e.g., whether knocking down gene_A affects gene_B's expression—achieving competitive or state-of-the-art performance on the PERTURBQA benchmark without requiring experimental data.
3. **Composing verifiers for stronger reasoning.** We show that mixtures of verifiers combining experimental model feedback (VCMs) and prior knowledge compose more robust reward structures and improve generalization across cell lines and biological contexts.
4. **Test-time reasoning and interpretability.** We find that chain-of-thought prompting at inference further enhances reasoning performance, producing interpretable reasoning traces that link predictions to underlying biological mechanisms.

2 Related Work

Recent reasoning-oriented LLMs—such as OpenAI's o-series, Claude 3.7/4, Gemini 2.5, and DeepSeek-R1—exhibit strong multi-step inference across domains. Their development spans four paradigms: (i) inference-time scaling (e.g., chain-of-thought, self-consistency); (ii) pure RL approaches like DeepSeek-R1-Zero, where traces emerge from accuracy- and format-based rewards; (iii) hybrid supervised finetuning plus RL, as in DeepSeek-R1; and (iv) distillation into smaller backbones such as Qwen [7, 32] or Llama [1, 6]. Persistent challenges remain in hallucination, logical consistency, verbosity, and interpretability—all linked to supervision quality.

Domain-specific reasoning has gained traction in scientific fields. BioReason [8] couples a genomic encoder with an LLM for disease–pathway inference, while CellReasoner [9] frames cell-type annotation as explicit reasoning. Both achieve strong results but rely on curated datasets, limiting scalability and motivating richer, model-based reasoning signals.

Our approach differs by using machine learning models of biology directly as reward-generating verifiers. Prior works integrate model embeddings into reasoning traces but still depend on annotated datasets; we instead shape rewards via predictions from Virtual Cell Models (VCMs), showing that biological world models can directly supervise reasoning LLMs.

Wu et al. [31] propose SUMMER, an inference-time pipeline combining knowledge graphs, retrieval, and chain-of-thought prompting for perturbation prediction. Although it improves on PERTURBQA, causal directionality remains error-prone and preprocessing requires large models. In contrast, our models reach comparable or better performance using only simulated predictions and benefit further from chain-of-thought prompting.

Our work also connects to concurrent research on *soft* and *AI-based* verification. In RLAIF [30] and related efforts, LLMs serve as reward models. Our **RLEMF** approach 3.3 differs by not requiring a language-based verifier, instead leveraging predictive biological models defined over experimental data and linked through prompting and embeddings. This bridges models of experimental data—providing AI feedback—and reasoning LLMs that learn to generate accurate scientific explanations. We similarly use probabilistic, verifiable rewards but extend beyond text models. Saad et al. [33] also employ LLMs as soft verifiers and combine them; in contrast, we generalize to arbitrary biological models and treat multiple verifiers as distinct reward functions. In a framework most related to our **RLPK** paradigm 3.4, Yu et al. [34] use the reasoning LLM itself to score answers, whereas we rely on structured scientific knowledge bases such as Gene Ontology.

Finally, replacing labeled data with surrogate model predictions has precedent in reinforcement learning [35] and black-box optimization [36, 37]. We extend this idea to reasoning models for biology, shifting supervision from experiments to simulations and expanding the design space of verifiers for scientific reasoning.

3 Methods

Training reasoning models with reinforcement learning (RL) involves three components: a dataset of prompts, an LLM that generates completions, and a verifier that assigns rewards. In standard domains, verifiers return exact signals—e.g., whether code executes or a math answer is correct. In biology, some queries can be validated experimentally (hard verification), but large-scale testing is infeasible. Virtual Cell Models (VCMs) provide an alternative by generating predictive feedback, enabling *soft verification*. Consider a perturbation query such as: *Is a knockdown of AARS in hepg2 cells likely to result in differential expression of ATAD2B?* During training, the LLM produces completions o_i for query q, which are evaluated by one of three verifier types (Fig. 1; Table 2):

1. **Hard verification:** when an experimental outcome exists in D_{EXP}, the reward is binary, $r_i^{\text{hard}}(q; o_i) \in \{0, 1\}$.
2. **Soft verification with models:** if experiments are unavailable, predictions from a biological model M provide probabilistic rewards $r_i^{\text{soft}}(q; o_i) = f_n(M(q_i; c_j))$, where $0 \leq r_i^{\text{soft}} \leq 1$.
3. **Soft verification with knowledge sources:** curated resources such as the Gene Ontology (GO) [38, 39] return rewards based on term overlap, ROUGE similarity, or the likelihood of GO annotations under the reasoning model.

Tables 1 and 2 summarize the prompting setup and corresponding reward formulations.

Table 1: Example of system and training prompts used for verifier training.

System Prompt	Training Prompts
A conversation between User and Biologist. The user asks a question, and the Biologist solves it. The biologist first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within ***<think> </think>*** *and* ***<answer> </answer>*** *tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>.*	*Is a knockdown of* ***gene_A*** *in* ***cell_line*** *cells likely to result in differential expression of* ***gene_B****?* **Example:** *Is a knockdown of AARS in hepg2 cells likely to result in differential expression of ATAD2B?*

Table 2: Verifiers used during RL training. EXP = exp data; MLP = multilayer perceptron; GO = Gene Ontology.

Verifier	Type	Reward Signal	Source
EXP	Hard	Binary $r_i^{\text{hard}} \in \{0, 1\}$	Experimental data
MLP	Soft	Probability $r_i^{\text{soft}} = p,\ 0 \leq p \leq 1$	VCM
GO	Soft	ROUGE, keyword, likelihood	Knowledge base

3.1 Reinforcement Learning for reasoning

Let $P(Q)$ denote a dataset used for training; q a query sampled from $P(Q)$, G a set of outputs generated during training by the reasoning LLM π_θ; o_i a generated sequence of tokens with tokens $o_{i,t}$ in response to q; π_{ref} a reference base model from the supervised finetuned LLM; r_ϕ a reward model emitting rewards r_i; $L_{GRPO}(\theta)$ the surrogate objective and β the coefficient for the KL penalty. Given these variables, Group Relative Policy Optimization (GRPO) [40] training maximizes the following objective function, with the goal of increasing the accumulated collective rewards $\{r_{i,\geq t}\}$:

$$J_{GRPO} = \mathbb{E}_{q \sim P(Q),\, \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}} \big[L_{GRPO}(\theta) \big]. \tag{1}$$

We use the clipped surrogate objective:

$$L_{GRPO}(\theta) = \frac{1}{|G|} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min\Big(\frac{\pi_\theta(o_{i,t}|q,o_{i<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i<t})} \hat{A}_{i,t},\ g(\epsilon, \hat{A}_{i,t}) \Big) - \beta D_{KL}[\pi_\theta || \pi_{\text{ref}}] \tag{2}$$

$$g(\epsilon, \hat{A}_{i,t}) = \text{clip}\Big(\frac{\pi_\theta(o_{i,t}|q,o_{i<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i<t})},\ 1-\epsilon,\ 1+\epsilon \Big)\, \hat{A}_{i,t} \tag{3}$$

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_1,\ldots,r_G\})}{\text{std}(\{r_1,\ldots,r_G\})} \tag{4}$$

$$D_{KL}[\pi_\theta||\pi_{\text{ref}}] = \frac{\pi_{\text{ref}}(o_{i,t}|q,o_{i<t})}{\pi_\theta(o_{i,t}|q,o_{i<t})} - \log\frac{\pi_{\text{ref}}(o_{i,t}|q,o_{i<t})}{\pi_\theta(o_{i,t}|q,o_{i<t})} - 1 \tag{5}$$

3.2 Rbio-RLEXP: reinforcement learning with Hard Verification

In this framework we are most similar to the RLHF scenario, where direct observations of experimental data are translated into language tokens and used to train a reasoning model directly using GRPO. For a task with a binary outcome and a verifier V that emits binary rewards, r_i in Eq. 4 becomes: $\mathrm{r}_i^{hard}(q, o_i) = r_i(q, o_i \mid V) \in \{0, 1\}$

We consider the existence of a broad experimental dataset D_{EXP} that can be used as a source of this reward feedback given a query, where the reward takes the shape of a label about a scientific fact we can ask the model to reason about. If the outcome of q has been validated experimentally and is available in D_{EXP}, we can directly verify o_i and emit a binary reward using $V = \{D_{EXP}\}$:

$$r_i^{hard}(q, o_i) = r_i(q, o_i \mid D_{EXP}) \tag{6}$$

$$r_i^{hard}(q, o_i) = \begin{cases} 1, & o_i = \text{True},\ D_{\text{EXP}}(q) = \text{True}, \\ 1, & o_i = \text{False},\ D_{\text{EXP}}(q) = \text{False}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The RLEXP algorithm is presented in Alg. 1.

3.3 Rbio-RLEMF: reinforcement learning with experimental model feedback

Similar to the framework of RLAIF, we here propose a related process which utilizes arbitrary other (non-LLM) models as feedback mechanisms for a query, in our example world models of biology defined on experimental data that can be queried appropriately. In the absence of experimental data D_{EXP} for RL-training as explained in Sec. 3.2, predictive models of such data M can act as surrogate verifiers ($V = \{M\}$). If q has not been validated experimentally, we can verify o_i using predictions from a biological model M. The reward is $r_i^{soft}(q; o_i) = fn(M(q_i; c_j))$, where c_j denotes the context (e.g., cell line or covariates). The emitted rewards are probabilistic:

$$r_i^{soft}(q, o_i) = r_i(q, o_i \mid V) = p(q, o_i \mid M), \quad 0 \le p \le 1 \tag{8}$$

In the example of the biological application of evaluating perturbation prompts with a model, we consider $M :=$ MLP, and $p(q, o_i \mid M)$ becomes the model's predicted probability for q being true. We show an overview of RLEMF in Alg. 2.

3.4 Rbio-RLPK: reinforcement learning from prior knowledge

Another avenue we propose for injecting knowledge into reasoning models for science is via prior knowledge. Here, given a structured database of prior knowledge, we can query a reasoning model against knowledge in that database and score the model itself against it. Given that knowledge sources-denoted KS are able to act as verifiers $V = \{KS\}$ and emit rewards, we can generate rewards on o_i based on KS using some metric m. This setting is outlined in Fig. 1c - C for the case of perturbation prediction. Concretely, we use curated resources such as the Gene Ontology (GO) [38, 39], which provides gene annotations across axes like *molecular processes*, *cellular components*, and *biological processes*. We show an overview of RLPK under Alg. 3. In this paradigm, Eq. 4 then becomes:

$$r_i^{soft}(q, o_i) \ = \ r_i(q, o_i \mid V) \ = \ r_m(q, o_i \mid KS), \tag{9}$$

where rewards are not necessarily confined to $[0, 1]$, but depend on the chosen metric.

We experiment with three types of metrics: ROUGE-based scores, keywords-based scores, and likelihood estimations, all of which require querying KS (GO ontology) for prior knowledge on q. Assuming we have access to $\{q_j\}$ pieces of prior knowledge on q, we accumulate rewards as:

$$q_j^{prior} \mid KS = \texttt{query\_KS}(q_j), \qquad r_i^{soft}(q, o_i) \ = \ \sum_j r_m(q_j^{prior}, o_i \mid KS). \tag{10}$$

ROUGE-based verifiers. We request the model to expose the relevant gene facts inside <gene_info> tags - which we refer to as $o_i^{relevant}$ - and compute standard ROUGE-1/2/L F-scores between q_j^{prior} and the extracted $o_i^{relevant}$:

$$r_m \;=\; \sum_j \sum_{X\in\{1,2,L\}} \mathrm{ROUGE}{-}X\big(q_j^{prior},\, o_i^{relevant}\big).$$

Keywords-based verifiers. We count normalize overlap of GO keywords present in the reasoning trace:

$$r_m \;=\; \sum_j \mathrm{KWS}\big(q_j^{prior},\, o_i^{relevant}\big), \qquad \mathrm{KWS}(s_1,s_2) \;=\; \frac{|s_1 \cap s_2|}{|s_1|}. \tag{11}$$

Likelihood-based verifiers. For likelihood-based verifiers, we use the likelihood of the prior knowledge $\{q_j^{prior}\}$ under our learned policy π_θ to generate rewards under the reasoning model, encouraging higher likelihood for scientifically accurate facts to reduce hallucinations. To account for variability in sequence length, we average over the sequence tokens y_k in q_j^{prior}:

$$r_m(q, o_i \mid LL) = \sum_j LL_{\pi_\theta}(q_j^{prior}); \quad LL_{\pi_\theta}\Big(q_j^{prior} \mid \pi_\theta\Big) = \frac{1}{T}\sum_{k=1}^{T} \log p_{\,\pi_\theta}(y_k \mid y_{<k}) \tag{12}$$

Normalization. GRPO uses normalized advantages (Eq. 4), mapping rewards to mean 0 and std 1. When composing multiple verifiers, imbalanced scales can skew updates, with GO-based rewards most affected due to skewed metric distributions. We therefore normalize GO-based rewards to $[0, 1]$ using an Exponential Moving Average (EMA) before policy updates:

$$\tilde{r} \leftarrow (1-\alpha)\tilde{r} + \alpha r_m, \quad \tilde{v} \leftarrow (1-\alpha)\tilde{v} + \alpha(r_m - \tilde{r}_{\text{prev}})(r_m - \tilde{r}), \tag{13}$$

$$\bar{r} = 0.5 + \frac{1}{2z_{\max}} \,\mathrm{clip}\left(\frac{r_m - \tilde{r}}{\max\big(\sqrt{\tilde{v}+\varepsilon},\, s_{\min}\big)}, -z_{\max},\, z_{\max}\right). \tag{14}$$

This $\bar{r}$ provides a normalized reward signal, used in Eq. 4 to compute the token-level advantages.

3.5 Composable Verification for model integration

For all rbio models, we also use formatting rewards r_{format} - as described in Alg. 4 - and mention rewards $r_{mention}$ as described in Alg. 5 (e.g., for gene mentions). When we train on combinations of verifiers as described in Sec. 4.2, each prompt q can be verified with a different verification source V_s. With multiple verifiers V_k emitting rewards $r_{i,k}$, we then have:

$$r_i(q, o_i) = r_{\text{format}} + r_{\text{mention}} + \sum_k \delta_{ks}\, \lambda_k\, r_{i,k}(q, o_i \mid V_k), \qquad \lambda_k \geq 0. \tag{15}$$

Unless otherwise stated, $\lambda_k = 2$. This gives more weight to the variance of the soft verifiers compared to r_{format} and $r_{mention}$ during GRPO updates.

4 Experiments

4.1 rbio with AI-verification generalizes OOD on perturbation tasks

On PERTURBQA [31] (CRISPRi knockdowns in RPE1, K562, HEPG2, JURKAT), models trained with *soft verifiers* generalize to held-out cell lines, reducing reliance on cell-line–specific experimental data. We first evaluate a 2-layer MLP (64 hidden units) trained on three cell lines and use it to generate predictions on the fourth, which serve as rewards during RL. Gene representations include one-hot, Gene2Vec [41], and ESM [5]. The resulting models, *rbio-MLP-leave-one-out-one-hot* and *rbio-MLP-leave-one-out-gene2vec*, perform comparably to experimental-data–trained rbio models.

We compare to two experimental-data baselines: *rbio-EXP-one-cell-line* (train/test within a cell line; Fig. 2a) and *rbio-EXP-leave-one-out* (train on three cell lines; test on the fourth; Fig. 2b). We also

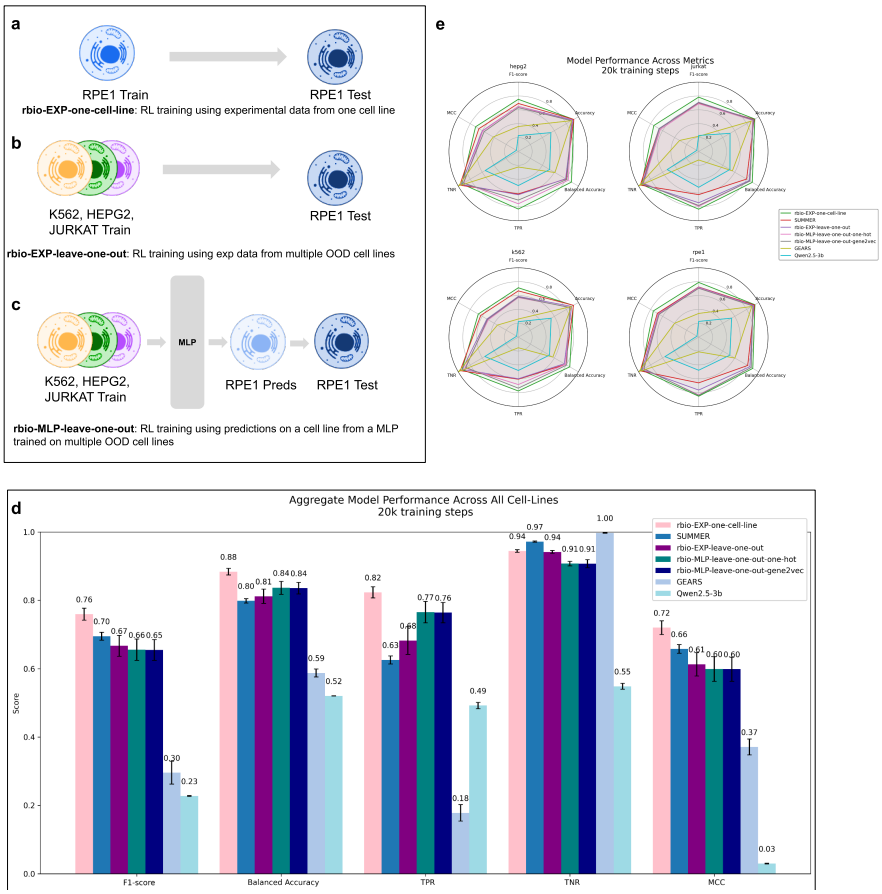


Figure 2: Model performance for experimental vs. simulation-based soft verification. (a) rbio-EXP-one-cell-line: trained and tested on the same cell line (in-distribution). **(b) rbio-EXP-leave-one-out:** trained on three cell lines, tested on the held-out one (out-of-distribution). **(c) rbio-MLP-leave-one-out:** trained using MLP predictions on the held-out line (MLP fit on the others). **(d) Aggregate metrics:** computed over four cell lines (K562, RPE1, JURKAT, HEPG2), averaged across 5 runs. **(e) Metrics split by cell line.** Baselines: *SUMMER* (experimental + domain knowledge), *GEARS* (specialized perturbation model),*Qwen2.5-3b* (base reasoning model).

benchmark against SUMMER [31]. As shown in Fig. 2d–e, the soft-verifier models closely match experimental-data models on F1 and MCC, and exceed them in Balanced Accuracy via higher TPR while maintaining similar TNR. Identifying true effects is paramount in perturbation, so higher TPR is valuable even with some F1 trade-off. All rbio variants also outperform GEARS [42] and the base Qwen2.5-3B.

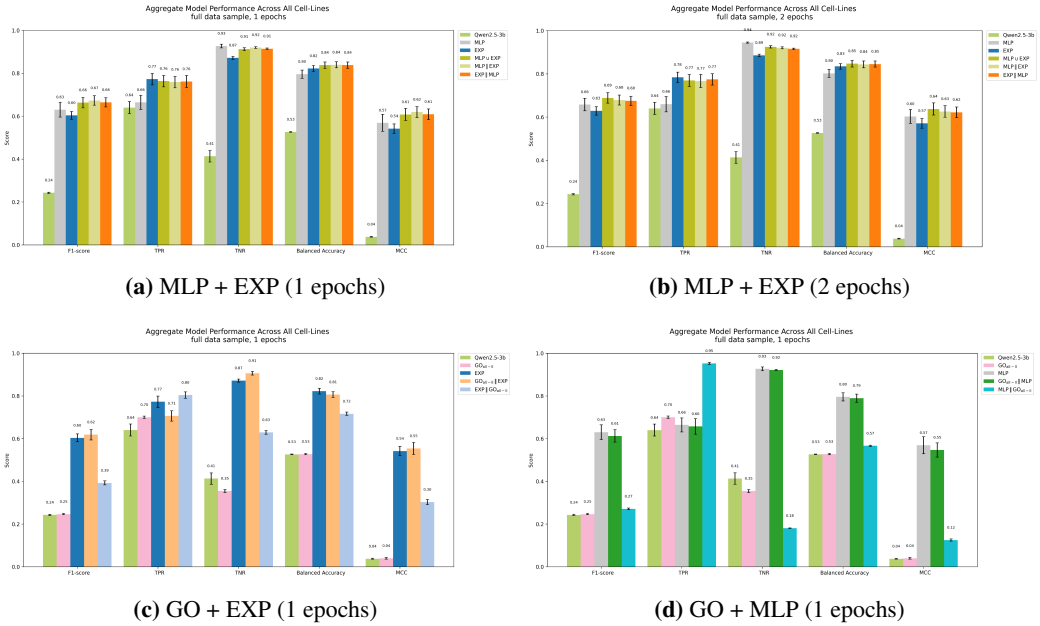
4.2 Training rbio on mixtures of AI-verifiers leads to performance gains

We find that combining verifiers can improve performance over using them individually. Notably, the order in which models see the verifiers matters, reflecting differences in the knowledge provided. For a pair of verifiers V_i, V_j, we evaluate:

1. V_i: trained only with V_i
2. $V_i \| V_j$: trained sequentially, V_i, then V_j
3. $V_i \cup V_j$: trained on a random mixture of V_i and V_j

We experiment with the following combinations of verifiers:

1. V_1 = EXP (hard verifier; experimental data); V_2 = MLP (soft verifier; MLP predictions)



(a) MLP + EXP (1 epochs)

(b) MLP + EXP (2 epochs)

(c) GO + EXP (1 epochs)

(d) GO + MLP (1 epochs)

Figure 4: Composing verifiers. $V_i \| V_j$: sequential (first V_i, then V_j). $V_i \cup V_j$: random mixture.

2. V_1 = EXP (hard verifier; experimental data); $V_2 = GO_{all-ll}$ (soft verifier; GO Ontology Knowledge Source, likelihood-based reward)
3. V_1 = MLP (soft verifier; MLP predictions); $V_2 = GO_{all-ll}$ (soft verifier; GO Ontology Knowledge Source, likelihood-based reward)

Note that the training data for each of V_1, V_2 is independent of each other - i.e. if V_1 is a verifier of experimental data from a dataset D_1, emissions from V_2 will be on an independent dataset D_2 where $D_1 \cap D_2 = \{\}$. In the case of the GO_{all-ll} the soft verification is the likelihood of the prior knowledge $\{q_j^{prior}\}$ we have under our learned policy π_θ as described in Eq. 12. As shown in Fig. 4, adding verifiers can improve performance over using them individually. For V_1 = EXP and V_2 = MLP (Fig. 4a,b, Fig. 7), all three composition strategies (Sec. 4.2) perform similarly in a variety of data sampling regimes, yet each surpasses the single verifiers, underscoring the complementary value of strong verification sources such as experimental data and models of experimental data.

When mixing knowledge and experimental sources, order becomes critical. In Fig. 4c,d, and Fig. 9 training first on GO_{all-ll} then on MLP, EXP ($GO_{all-ll} \| MLP$, $GO_{all-ll} \| EXP$) outperforms the reverse. GO-based verification increases TPR (by capturing more positives) - as shown in its performance when evaluated individually compared to baseline - but reduces TNR; subsequent training on experimental data rebalances TNR, improving Balanced Accuracy and MCC. Conversely, starting from EXP or MLP then adding GO_{all-ll} lowers performance, suggesting knowledge sources can dilute experimental signals if applied late. Thus, knowledge is most effective early to guide the model, while stronger experimental signals should refine performance later. We offer more analyses on compositions of verifiers under Fig. 7 9, 11 in Supplementary Material, reinforcing these findings. This aligns with general training paradigms: start with broader, noisier data (e.g. ontologies) to shape representations, then refine with higher-quality data (e.g. experiments) to maximize performance. The strategy is extendable to multiple verifiers $V_1, V_2, \ldots, V_k$, which could capture different sources of knowledge.

4.3 Rbio with chain-of-thought yields state of the art on PERTURBQA

Adding chain-of-thought (CoT) reasoning at inference improves all rbio variants we tested (Table. 3), surpassing SUMMER as state-of-art performance on the PerturbQA benchmark. The CoT prompt that performed the best was: *'The Biologist will evaluate each step of this problem, using logical reasoning and evidence from the prompt."* Examples of performance increase: *rbio-EXP-all-cell-lines*

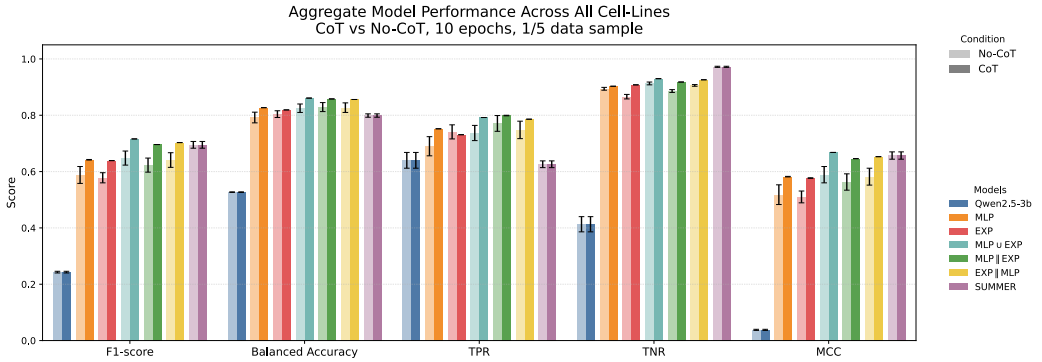


Figure 5: Effect of chain-of-thought prompting. Models using CoT achieve state-of-the-art performance on the PerturbQA benchmark.

Model	F1-score	Balanced Acc.	TPR	TNR	MCC
Models trained on full data size					
rbio-EXP	0.750 ± 0.018	0.883 ± 0.011	0.827 ± 0.018	0.939 ± 0.003	0.709 ± 0.020
rbio-EXP-CoT	**0.786 ± 0.000**	**0.907 ± 0.000**	**0.872 ± 0.000**	0.943 ± 0.000	**0.752 ± 0.000**
rbio-MLP	0.669 ± 0.025	0.855 ± 0.017	0.807 ± 0.030	0.902 ± 0.004	0.618 ± 0.029
rbio-MLP-CoT	0.714 ± 0.000	0.889 ± 0.000	0.873 ± 0.000	0.906 ± 0.000	0.672 ± 0.000
SUMMER	0.695 ± 0.012	0.799 ± 0.006	0.626 ± 0.012	0.972 ± 0.002	0.657 ± 0.013
Qwen2.5-3b	0.231 ± 0.002	0.522 ± 0.001	0.529 ± 0.014	0.515 ± 0.013	0.032 ± 0.001
GEARS	0.296 ± 0.033	0.588 ± 0.012	0.178 ± 0.024	**0.997 ± 0.001**	0.371 ± 0.023
Models trained on 1/5 of full data size					
rbio-MLP	0.588 ± 0.030	0.792 ± 0.019	0.690 ± 0.034	0.894 ± 0.005	0.518 ± 0.035
rbio-MLP-CoT	0.642 ± 0.001	0.827 ± 0.000	0.752 ± 0.000	0.903 ± 0.000	0.582 ± 0.001
rbio-EXP	0.578 ± 0.018	0.804 ± 0.012	0.741 ± 0.025	0.866 ± 0.008	0.510 ± 0.021
rbio-EXP-CoT	0.639 ± 0.000	0.819 ± 0.000	0.731 ± 0.000	0.908 ± 0.000	0.577 ± 0.001
rbio-MLP ∪ EXP	0.648 ± 0.025	0.825 ± 0.015	0.737 ± 0.027	0.913 ± 0.005	0.589 ± 0.029
rbio-MLP ∪ EXP-CoT	**0.716 ± 0.000**	**0.861 ± 0.000**	0.792 ± 0.000	**0.930 ± 0.000**	**0.668 ± 0.000**
rbio-MLP ‖ EXP	0.623 ± 0.025	0.829 ± 0.016	0.771 ± 0.028	0.886 ± 0.005	0.563 ± 0.029
rbio-MLP ‖ EXP-CoT	0.696 ± 0.000	0.858 ± 0.000	**0.799 ± 0.001**	0.918 ± 0.000	0.646 ± 0.000
rbio-EXP ‖ MLP	0.641 ± 0.026	0.827 ± 0.017	0.748 ± 0.031	0.906 ± 0.003	0.582 ± 0.030
rbio-EXP ‖ MLP-CoT	0.703 ± 0.000	0.856 ± 0.000	0.786 ± 0.000	0.926 ± 0.000	0.653 ± 0.000

Table 3: Aggregate performance on the PERTURBQA benchmark (mean ± SE over 5 completions). *rbio-EXP* corresponds to rbio-EXP-all-cell-lines. SUMMER [31] is current SOTA; best models bolded.

F1 0.75→0.79, Balanced Accuracy 0.88→0.91, TPR 0.83→0.87; *rbio-MLP-ESM* F1 0.67→0.71, Balanced Accuracy 0.85→0.89, TPR 0.81→0.87. Examples of answers and reasoning traces can be seen in Figure 15. Shown in Figure 5 are rbio models trained on only one-fifth of the data and tested with and without CoT. Remarkably, adding CoT at inference lets them reach state-of-the-art performance on PerturbQA - with *rbio-MLP ∪ EXP-CoT* surpassing SUMMER despite being trained on a fraction of training data - demonstrating the power of inference-time capabilities and verifier composition in reasoning models.

5 Conclusion

We introduced **rbio1**, a suite of reasoning models trained via *soft verification*, where simulations from biological world models provide rewards in reinforcement learning. This approach achieves performance comparable to experimental-data–trained models, particularly when combined with chain-of-thought prompting. By leveraging predictive bio-models, rbio1 demonstrates that simulations can substitute for costly experimental supervision. Looking ahead, we aim to extend rbio1 across diverse biological models and modalities, moving toward a universal virtual cell system that integrates knowledge from multiple sources into a shared reasoning framework.

6 Acknowledgment

We thank Wyatt Robarts and Jae Lee for legal and security counsel; CZI AI-Infra team for supporting the computing needs; Yun Liu and the CZI Creative team for assistance with figure preparation and graphic design; Maximilian Lombardo and Omar Valenzuela for assisting with open sourcing the code repository and Notebook tutorials; Amanda Mahoney for helpful guidance on communications and outreach related to this work.

References

[1] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[2] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480, 2024.

[5] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[6] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[7] Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[8] Adibvafa Fallahpour, Andrew Magnuson, Purav Gupta, Shihao Ma, Jack Naimer, Arnav Shah, Haonan Duan, Omar Ibrahim, Hani Goodarzi, Chris J Maddison, et al. Bioreason: Incentivizing multimodal biological reasoning within a dna-llm model. *arXiv preprint arXiv:2505.23579*, 2025.

[9] Guangshuo Cao, Yi Shen, Jianghong Wu, Haoyu Chao, Ming Chen, and Dijun Chen. Cellreasoner: A reasoning-enhanced large language model for cell type annotation. *bioRxiv*, pages 2025–05, 2025.

[10] Siddharth M Narayanan, James D Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G Rodriques, and Andrew D White. Training a scientific reasoning model for chemistry. *arXiv preprint arXiv:2506.17238*, 2025.

[11] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[12] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Caleb Voss, Alec Radford, Dario Amodei, Paul Christiano, and Jan Leike. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[13] William Pan, Xuechen Song, Yuchen Zhang, Percy Liang, and Tatsunori B Hashimoto. Reinforcement learning with verifiable rewards from correctness feedback. *arXiv preprint arXiv:2303.17491*, 2023.

[14] Boris M Slepchenko, James C Schaff, Ian Macara, and Leslie M Loew. Quantitative cell biology with the virtual cell. *Trends in cell biology*, 13(11):570–576, 2003.

[15] Leslie M Loew and James C Schaff. The virtual cell: a software environment for computational cell biology. *TRENDS in Biotechnology*, 19(10):401–406, 2001.

[16] Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25): 7045–7063, 2024.

[17] Yanay Rosen, Yusuf Roohani, Ayush Agarwal, Leon Samotorčan, Tabula Sapiens Consortium, Stephen R Quake, and Jure Leskovec. Universal cell embeddings: A foundation model for cell biology. *bioRxiv*, pages 2023–11, 2023.

[18] James D Pearce, Sara E Simmonds, Gita Mahmoudabadi, Lakshmi Krishnan, Giovanni Palla, Ana-Maria Istrate, Alexander Tarashansky, Benjamin Nelson, Omar Valenzuela, Donghui Li, et al. A cross-species generative cell atlas across 1.5 billion years of evolution: The transcriptformer single-cell model. *bioRxiv*, pages 2025–04, 2025.

[19] Haiyang Bian, Yixin Chen, Xiaomin Dong, Chen Li, Minsheng Hao, Sijie Chen, Jinyi Hu, Maosong Sun, Lei Wei, and Xuegong Zhang. scmulan: a multitask generative pre-trained language model for single-cell analysis. In *International Conference on Research in Computational Molecular Biology*, pages 479–482. Springer, 2024.

[20] Nicholas Ho, Caleb N Ellington, Jinyu Hou, Sohan Addagudi, Shentong Mo, Tianhua Tao, Dian Li, Yonghao Zhuang, Hongyi Wang, Xingyi Cheng, et al. Scaling dense representations for single cell with transcriptome-scale context. *bioRxiv*, pages 2024–11, 2024.

[21] Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

[22] Ankit Gupta, Zoe Wefers, Konstantin Kahnert, Jan N Hansen, Will Leineweber, Anthony Cesnik, Dan Lu, Ulrika Axelsson, Frederic Ballllosera Navarro, Theofanis Karaletsos, et al. Subcell: Vision foundation models for microscopy capture single-cell biology. *bioRxiv*, pages 2024–12, 2024.

[23] Eric Nguyen, Michael Poli, Matthew G Durrant, Brian Kang, Dhruva Katrekar, David B Li, Liam J Bartie, Armin W Thomas, Samuel H King, Garyk Brixi, et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723):eado9336, 2024.

[24] Syed Asad Rizvi, Daniel Levine, Aakash Patel, Shiyang Zhang, Eric Wang, Sizhuang He, David Zhang, Cerise Tang, Zhuoyang Lyu, Rayyan Darji, et al. Scaling large language models for next-generation single-cell analysis. *bioRxiv*, pages 2025–04, 2025.

[25] Guillaume Richard, Bernardo P de Almeida, Hugo Dalla-Torre, Christopher Blum, Lorenz Hexemer, Priyanka Pandey, Stefan Laurent, Marie Lopez, Alexandre Laterre, Maren Lang, et al. Chatnt: A multimodal conversational agent for dna, rna and protein tasks. *bioRxiv*, pages 2024–04, 2024.

[26] Daniel Levine, Syed Asad Rizvi, Sacha Lévy, Nazreen Pallikkavaliyaveetil, David Zhang, Xingyu Chen, Sina Ghadermarzi, Ruiming Wu, Zihe Zheng, Ivan Vrkic, et al. Cell2sentence: teaching large language models the language of biology. *BioRxiv*, pages 2023–09, 2024.

[27] Moritz Schaefer, Peter Peneder, Daniel Malzl, Mihaela Peycheva, Jake Burton, Anna Hakobyan, Varun Sharma, Thomas Krausgruber, Joerg Menche, Eleni M Tomazou, et al. Multimodal learning of transcriptomes and text enables interactive single-cell rna-seq data exploration with natural-language chats. *bioRxiv*, pages 2024–10, 2024.

[28] Hongyoon Choi, Jeongbin Park, Sumin Kim, Jiwon Kim, Dongjoo Lee, Sungwoo Bae, Haenara Shin, and Daeseung Lee. Cellama: foundation model for single cell and spatial transcriptomics by cell embedding leveraging language model abilities. *bioRxiv*, pages 2024–05, 2024.

[29] Ana-Maria Istrate, Donghui Li, and Theofanis Karaletsos. scgenept: Is language all you need for modeling single-cell perturbations? *bioRxiv*, pages 2024–10, 2024.

[30] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Victor Cărune, Abhinav Rastogi, and Sushant Prakash. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

[31] Menghua Wu, Russell Littman, Jacob Levine, Lin Qiu, Tommaso Biancalani, David Richmond, and Jan-Christian Huetter. Contextualizing biological perturbation experiments through language. *arXiv preprint arXiv:2502.21290*, 2025.

[32] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[33] Jon Saad-Falcon, E Kelly Buchanan, Mayee F Chen, Tzu-Heng Huang, Brendan McLaughlin, Tanvir Bhathal, Shang Zhu, Ben Athiwaratkun, Frederic Sala, Scott Linderman, et al. Shrinking the generation-verification gap with weak verifiers. *arXiv preprint arXiv:2506.18203*, 2025.

[34] Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025.

[35] Olivier Francon, Santiago Gonzalez, Babak Hodjat, Elliot Meyerson, Risto Miikkulainen, Xin Qiu, and Hormoz Shahrzad. Effective reinforcement learning through evolutionary surrogate-assisted prescription. In *Proceedings of the 2020 Genetic and evolutionary computation conference*, pages 814–822, 2020.

[36] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

[37] Ioannis Gatopoulos, Romain Lepert, Auke Wiggers, Giovanni Mariani, and Jakub Tomczak. Evolutionary algorithm with non-parametric surrogate model for tensor program optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[38] Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, et al. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 2023.

[39] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[40] Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

[41] Jingcheng Du, Peilin Jia, Yulin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. Gene2vec: distributed representation of genes based on co-expression. *BMC genomics*, 20(Suppl 1):82, 2019.

[42] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Gears: Predicting transcriptional outcomes of novel multi-gene perturbations. *BioRxiv*, pages 2022–07, 2022.

[43] TRL - transformer reinforcement learning. `https://huggingface.co/docs/trl/en/index`. Accessed: 2025-8-15.

Supplementary Material

A Metrics

We formulate the genetic perturbation prediction task as a question in natural language with a binary answer. Given a pair of genes $gene_A$ and $gene_B$, the model is asked to emit a binary answer - **yes** or **no**. We use four CRISPRi single-gene perturbation knockdown datasets on four cancer cell lines (RPE1, K562, HEPG2, JURKAT), post-processed into natural language queries by PerturbQA [31]. We compute the following metrics:

$$Recall\ (TPR)\ = \frac{TP}{TP + FN} \tag{16}$$

$$TNR = \frac{TN}{TN + FP} \tag{17}$$

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{19}$$

$$Balanced\ Accuracy = \frac{TPR + TNR}{2} \tag{20}$$

$$MCC\ (Matthews\ Correlation\ Coefficient) = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{21}$$

B Data and Code Availability

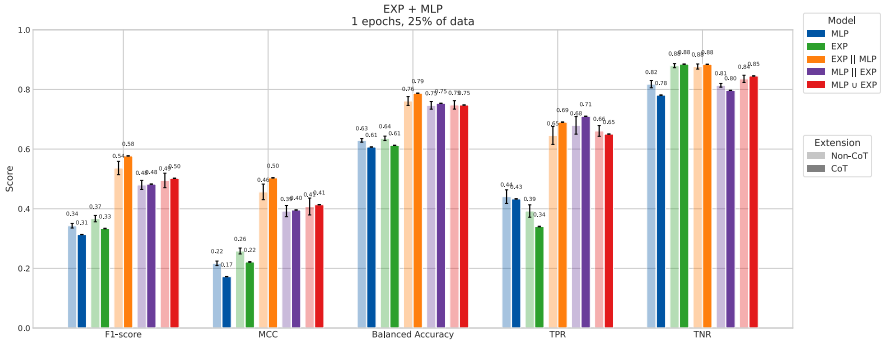
We have used the pre-processed versions, as well as the training and testing splits of the perturbation datasets on the four cell lines (K562, RPE1, HEPG2, JURKAT) from https://github.com/genentech/PerturbQA.

The code is publicly available at https://github.com/czi-ai/rbio, together with a training and an inference script. Model weights are publicly available at https://github.com/czi-ai/rbio. Tutorials, as well as model cards are available on the https://virtualcellmodels.cziscience.com.

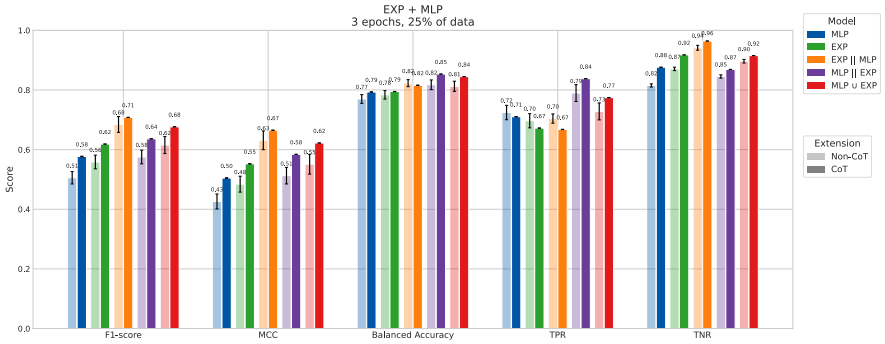
C Training and Evaluation

Models have been trained using the GRPO framework and HuggingFace interface[43]. We use a Qwen2.5-3B-Instruct model as a base model, the model weights having been accessed through HuggingFace. Each model trained on individual verifiers has been trained for 100k steps with max_completion_len=256, taking 10 days for completion on 8 H100 GPUs, with some variation between models. Models containing compositions of verifiers were trained up until 10 epochs and max_completion_len=4096, although subsequent experiment revelead no major difference by training with a smaller max_completion_len. All models used batch_size = 4, n_generation = 4 and a default learning rate of 5e-6. During inference, each model has been prompted for N=5 generations, with the following parameters: max_new_tokens=1024, do_sample=True, temperature=0.7, top_p=0.9, top_k=50. Metrics are reported over 5 different generations. Each model also includes formatting rewards, similarly to how it was done in [6].

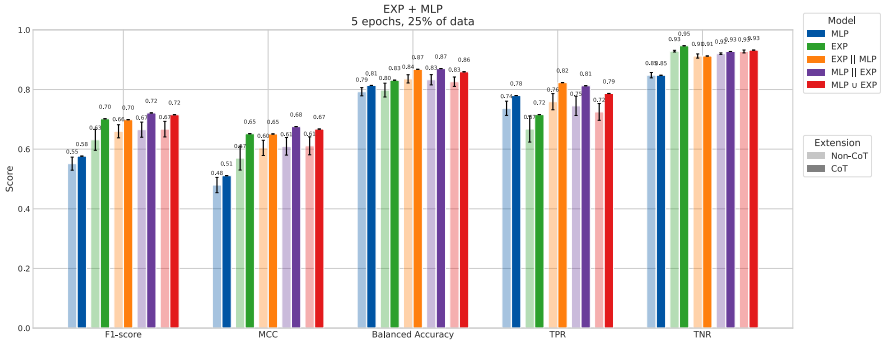
D Composition of Verifiers



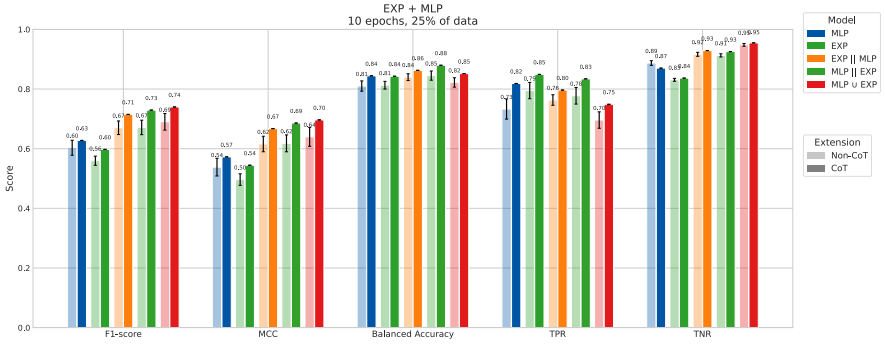
(a) EXP + MLP (1 epochs).



(b) EXP + MLP (3 epochs).

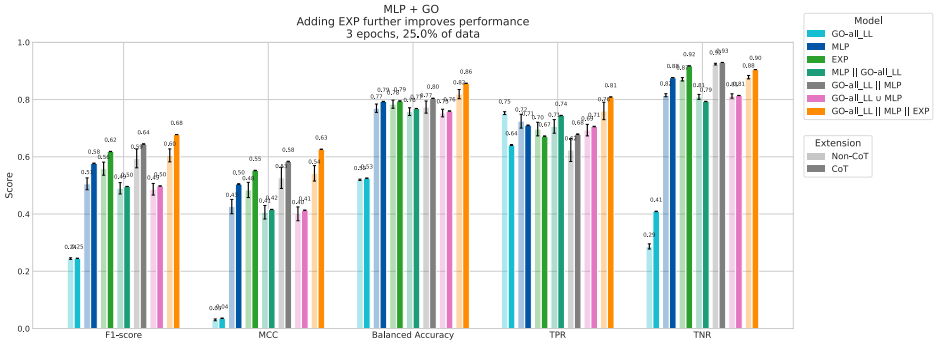


(c) EXP + MLP (5 epochs).

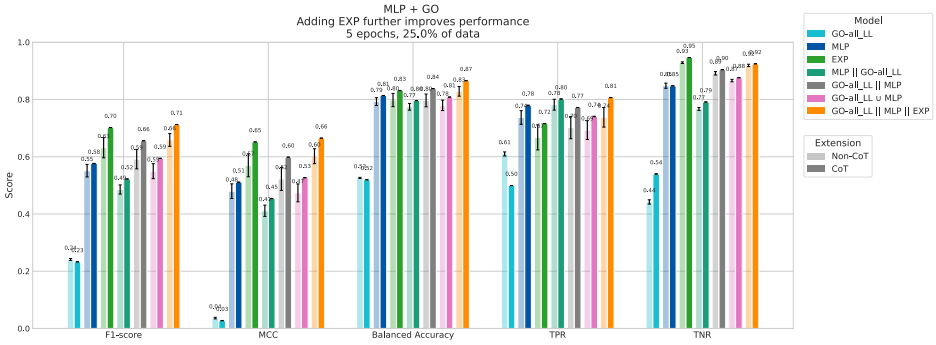


(d) EXP + MLP (10 epochs).

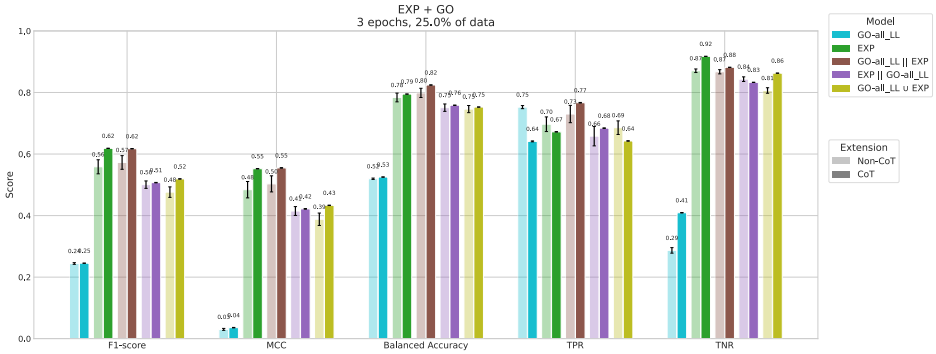
Figure 7: Composition of verifiers metrics. EXP + MLP Performance comparison across combinations of EXP + MLP and training durations (1, 3, 5, 10 epochs) under the 25% data regime.



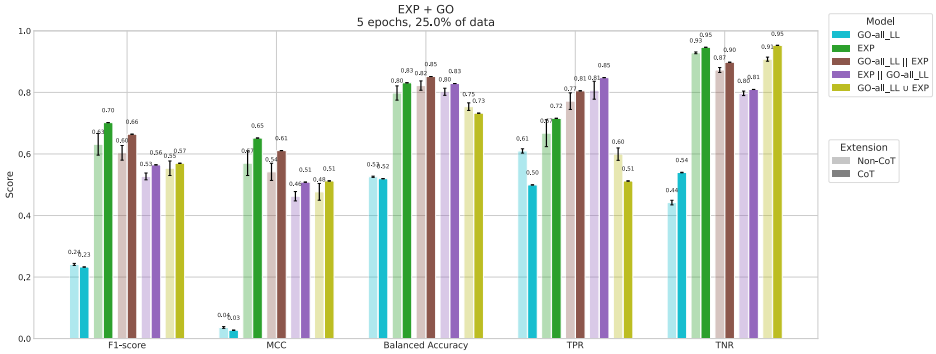
(a) GO + MLP (3 epochs).



(b) GO + MLP (5 epochs).

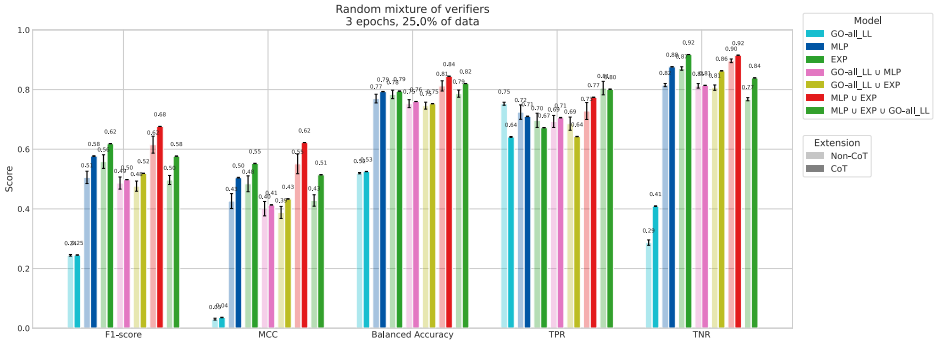


(c) GO + EXP (3 epochs).

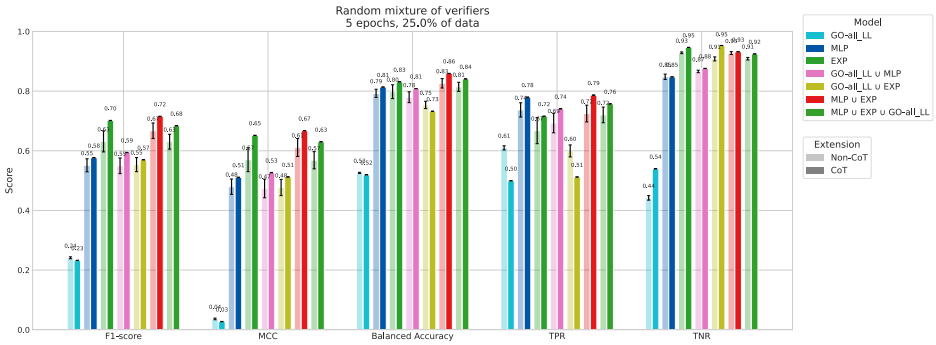


(d) GO + EXP (5 epochs).

Figure 9: Composition of verifiers metrics. GO + EXP and GO + MLP Performance comparison across combinations of GO+EXP and GO+MLP and training durations (3, 5 epochs) under the 25% data regime.



(a) Random mixtures (3 epochs).



(b) Random Mixtures (5 epochs).

Figure 11: Composition of random mixtures of verifiers Performance comparison across combinations of random mixtures of verifiers and training durations (1, 3, 5, 10 epochs) under the 25% data regime.

E Hard-Ground Truth Experiments

E.1 Training on Experimental Data improves performance compared to baselines

The composite radar plots 12 compare a Qwen-based language model that we further optimized with reinforcement learning (GRPO) and a hard verifier against three purely instruction-tuned baselines (Qwen2.5-3 B, Qwen2.5-7 B and Qwen3-32 B). Across four independent test cell-line datasets (K562, RPE1, HEPG2 and Jurkat) our hard-verified RL model encloses the baselines on every axis, namely, Accuracy, Precision, Recall (TPR), F1, Specificity (TNR) and AUC-ROC, indicating superior performance for all six metrics. The largest absolute gains concentrate on Recall, F1 and AUC-ROC, where the new model reaches radii of 0.7–0.9 while the baselines remain below 0.4. These improvements show that coupling reinforcement learning to the hard verifier causes the policy to capture a far larger share of true positives without materially inflating false-positive rates, thereby harmonizing sensitivity with specificity. The benefit is consistent across heterogeneous cell types and does not depend on parameter count: the 32-billion-parameter baseline performs almost identically to its 3-billion counterpart, underscoring that task-aligned objectives, here enforced by hard verification on experimental data, outweigh scale alone. Because missed positives are costlier than occasional false alarms in downstream wet-lab validation, the balanced operating point delivered by hard-verified RL training translates directly into practical utility. Finally, the consistently high AUC-ROC (around 0.9) suggests that the learned policy generalizes its reasoning rather than memorizing characteristics of any single dataset. Taken together, these findings demonstrate that injecting experimentally grounded hard verification into the reinforcement-learning loop is a powerful mechanism for elevating reasoning-centric LLMs beyond what can be achieved with instruction tuning and parameter scaling alone.

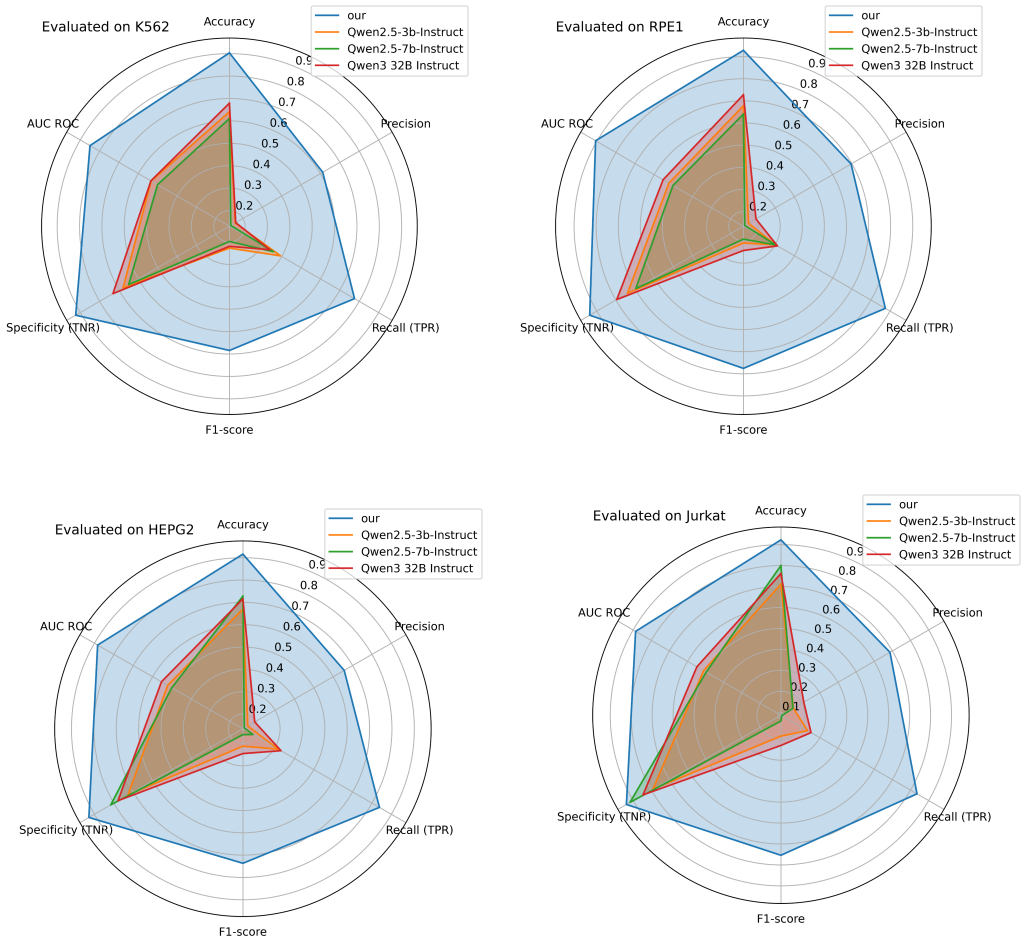


Figure 12: | Comparison of rbio-EXP with Qwen baselines. Our refers to rbio-EXP trained on one cell line at a time.

F Training on experimental data from multiple sources shows increased performance, showing there is beneficial cross-dataset knowledge transfer

Across all four evaluation datasets, the model trained jointly on the complete, heterogeneous collection ("Trained on all", blue polygons) delivers the most consistent and highest-scoring performance envelope. Accuracy, recall (TPR), specificity (TNR) and AUC-ROC cluster tightly around 0.88-0.92, while F1-scores remain around 0.80 despite the precision penalty introduced by class imbalance in these assays.

Withholding the target dataset during training ("Trained without X", orange) causes a uniform yet modest degradation. Accuracy, AUC-ROC and TNR typically fall by 2-4 percentage points, and F1-score by about 5 points. Importantly, however, this leave-one-out model still outperforms the best single-source baseline (green) by large margins ($\geq$ 15 points in accuracy and $\geq$ 0.20 in AUC-ROC on every dataset). The result demonstrates that information learned from the remaining three datasets generalizes meaningfully to the held-out system, providing clear evidence of positive transfer rather than over-fitting to individual batches or cell-line specific artifacts.

The shape of the radar plots is also instructive. Metrics that depend on negative class recognition (specificity, AUC-ROC) decline only slightly when a dataset is omitted, whereas precision and F1, which require well-calibrated decision boundaries for the minority class, drop more noticeably. This

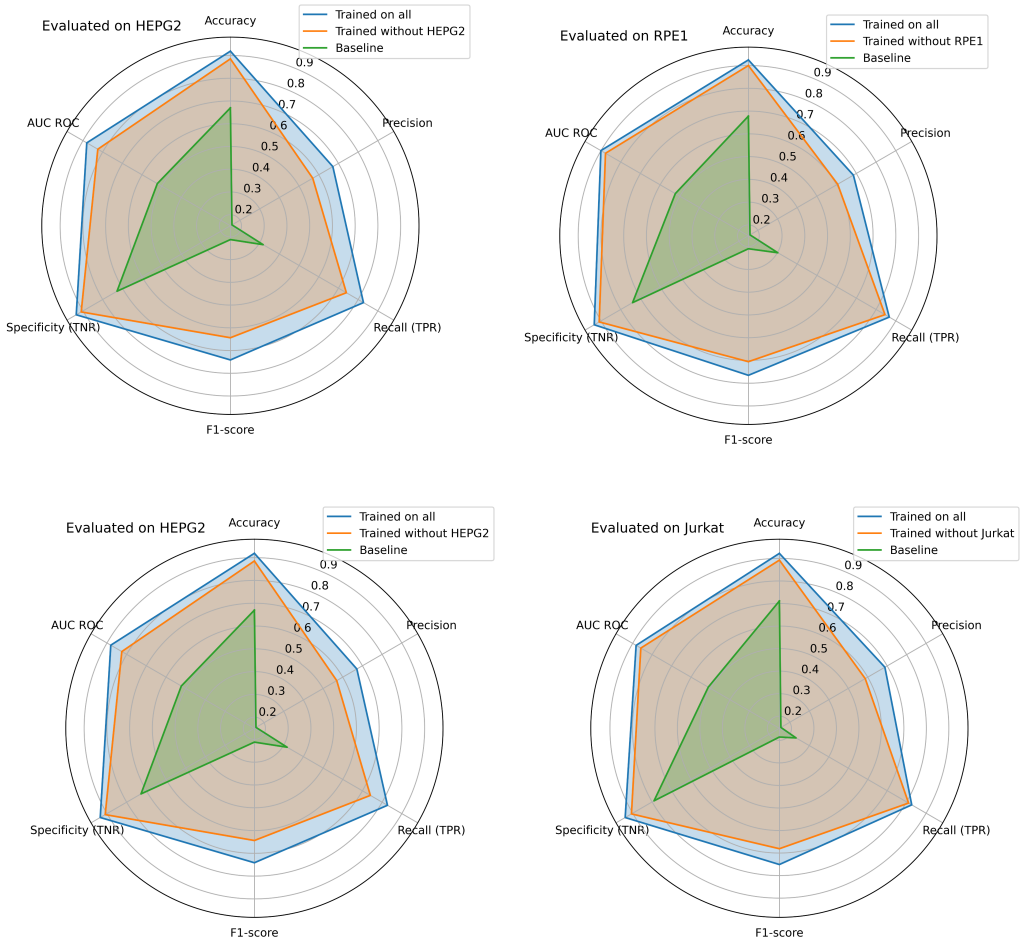


Figure 13: | Comparison of rbio-EXP trained on training splits of all cell lines vs trained on training splits of cell lines outside of the test distribution.. Our refers to rbio-EXP trained on one cell line at a time.

pattern implies that shared features across experiments chiefly support broad discrimination between classes; fine-grained calibration still benefits from direct exposure to the target distribution.

Taken together, these observations indicate that aggregating experimental data from disparate sources yields a richer representation space and endows the model with a transferable prior that boosts generalization to unseen datasets. Consequently, the incremental cost of acquiring and harmonizing additional training datasets is rewarded with measurable gains in robustness and predictive power, an encouraging result for scaling this approach to new cell types or assay modalities.

G Analysis of training on in and out-of-distribution experimental data

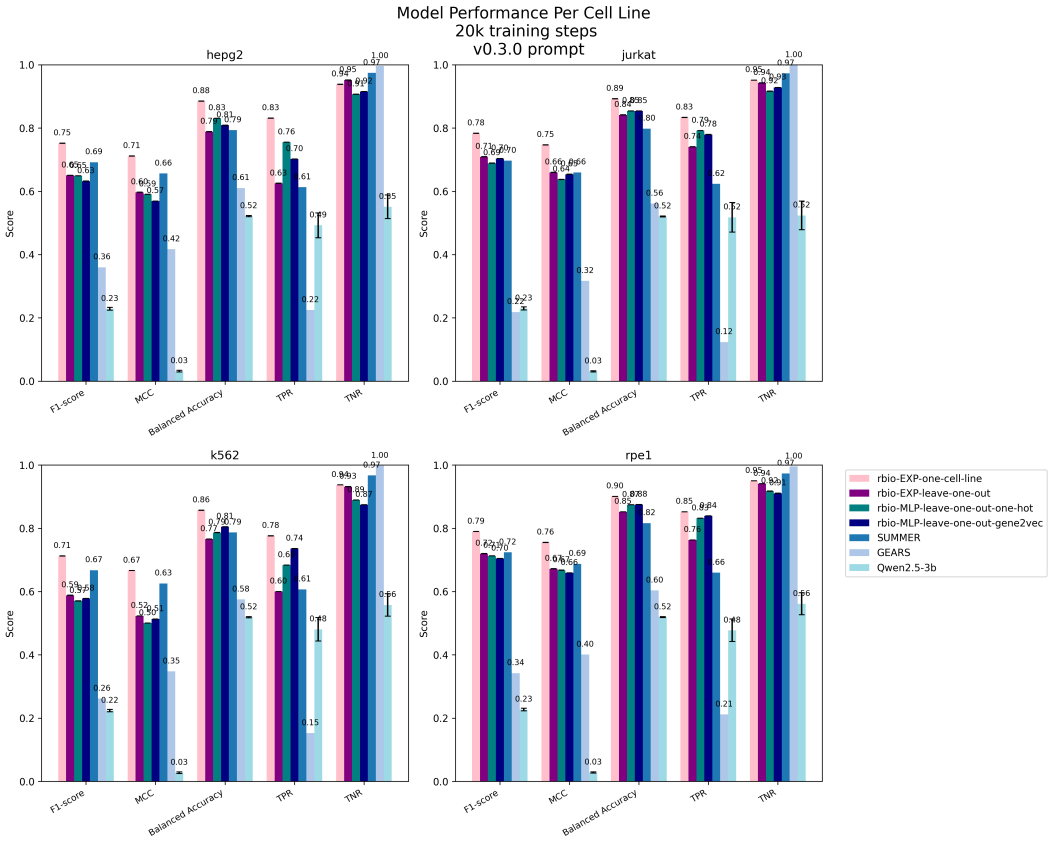


Figure 14: | Analysis on performance on rbio models trained on experimental data for in-distribution vs out of distribution datasets. rbio-EXP-one-cell-line is trained on one the training split from one cell line at a time and tested on the test split of that cell line (in-distribution), rbio-EXP-all-cell-lines is trained on the training splits on all four cell lines and tested on separate test splits, and rbio-EXP-leave-one-out is trained on the three cell lines not in test (out of distribution)

H Examples of reasoning traces

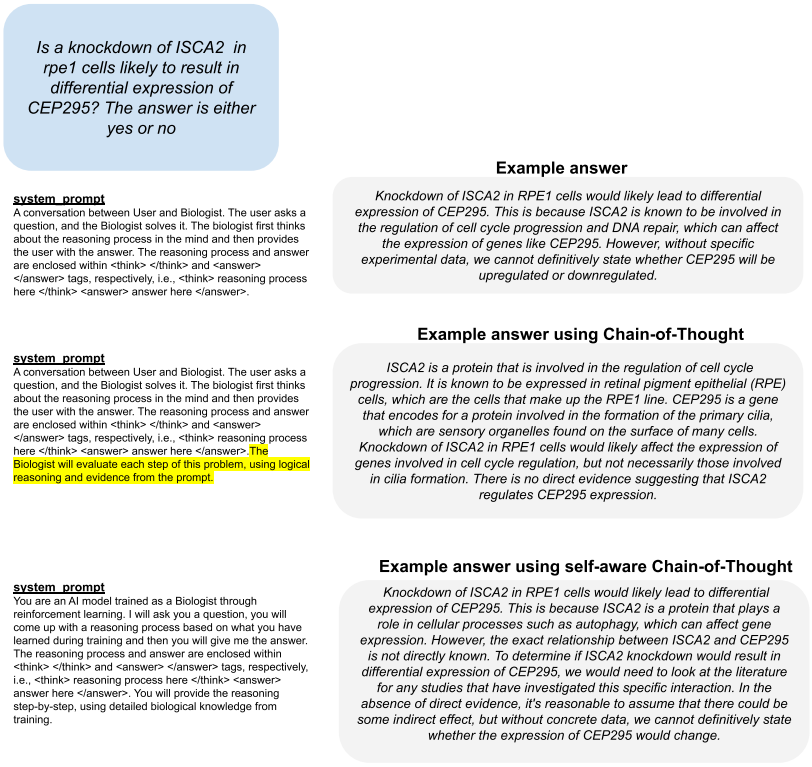


Figure 15: | Examples of Reasoning Traces for perturbation prediction answering a question sampled randomly from the test set. Different example answers for different chain-of-thought techniques.Outputs from a model trained on combinations of soft-verifiers.

Algorithm 1 Rbio-RLEXP: Reinforcement Learning with Hard Verification

Require: Dataset of prompts/experimental outcome labels $\{X_i, Y_i\}_{i=1}^{N}$
Require: Model parameters θ implementing policy π_θ
Require: Hyperparameters: τ (temperature), G (generations per prompt), β (KL penalty), ϵ (clipping)
Ensure: Trained model with parameters $\tilde{\theta}$ implementing policy $\pi_{\tilde{\theta}}$

Initialize θ from supervised fine-tuned LLM
for each step $t = 1$ to T **do**
  Sample batch indices $b \subset \{1, \ldots, N\}$ uniformly at random
  Retrieve batch $\{X_b, Y_b\}$ from dataset
  for each prompt X_b in batch **do**
    for $i = 1$ to G **do**
      Generate sequence $o_i \sim \pi_\theta(\cdot \mid X_b)$ using model θ and policy π_θ
      Extract binary answer $\hat{a}_i$ from o_i (if existing)
      if answer $\hat{a}_i$ exists **then**
        Score against ground truth Y_b:
        if $\hat{a}_i = Y_b$ **then**
          $r_i^{\text{hard}} = 1$
        else
          $r_i^{\text{hard}} = 0$
        end if
      else
        $r_i^{\text{hard}} = 0$ {Penalize missing answer}
      end if
      Add auxiliary rewards: $r_i = r_i^{\text{hard}} + r_{\text{format}} + r_{\text{mention}}$
    end for
  end for
  Compute normalized advantages $\hat{A}_{i,t}$ using Eq. (4)
  Update θ via GRPO objective (Eq. 2) with KL divergence penalty (Eq. 5)
end for
return $\tilde{\theta}$

Algorithm 2 Rbio-RLEMF: Reinforcement Learning with Experimental Model Feedback

Require: Dataset of prompts $\{X_i\}_{i=1}^{N}$ (without experimental labels)
Require: Pre-trained frozen model Φ (e.g., MLP, VCM)
Require: Model parameters θ implementing policy π_θ
Require: Reward transformation function η: maps model predictions to rewards in $[0, 1]$
Require: Hyperparameters: τ (temperature), G (generations per prompt), β (KL penalty), ϵ (clipping)
Ensure: Trained model with parameters $\tilde{\theta}$ implementing policy $\pi_{\tilde{\theta}}$

Initialize θ from supervised fine-tuned LLM
for each step $t = 1$ to T **do**
  Sample batch indices $b \subset \{1, \ldots, N\}$ uniformly at random
  Retrieve batch $\{X_b\}$ from dataset
  for each prompt X_b in batch **do**
    for $i = 1$ to G **do**
      Generate sequence $o_i \sim \pi_\theta(\cdot \mid X_b)$ using model θ and policy π_θ
      Extract binary answer $\hat{a}_i$ from o_i (if existing)
      if answer $\hat{a}_i$ exists **then**
        Query frozen model: $\hat{p} = \Phi(X_b)$ {Model prediction}
        Transform prediction to reward: $r_i^{\text{soft}} = \eta(\hat{p}, \hat{a}_i) \in [0, 1]$
      else
        $r_i^{\text{soft}} = 0$ {Penalize missing answer}
      end if
      Add auxiliary rewards: $r_i = r_i^{\text{soft}} + r_{\text{format}} + r_{\text{mention}}$
    end for
  end for
  Compute normalized advantages $\hat{A}_{i,t}$ using Eq. (4)
  Update θ via GRPO objective (Eq. 2) with KL divergence penalty (Eq. 5)
end for
return $\tilde{\theta}$

Algorithm 3 Rbio-RLPK: Reinforcement Learning from Prior Knowledge

Require: Dataset of prompts $\{X_i\}_{i=1}^{N}$ (without labels)
Require: Knowledge source KS (e.g., Gene Ontology)
Require: Model parameters θ implementing policy π_θ
Require: Knowledge scoring function ν: scores reasoning traces against prior knowledge
Require: Hyperparameters: τ (temperature), G (generations per prompt), β (KL penalty), ϵ (clipping)
Ensure: Trained model with parameters $\tilde{\theta}$ implementing policy $\pi_{\tilde{\theta}}$

Initialize θ from supervised fine-tuned LLM
for each step $t = 1$ to T **do**
  Sample batch indices $b \subset \{1, \ldots, N\}$ uniformly at random
  Retrieve batch $\{X_b\}$ from dataset
  for each prompt X_b in batch **do**
    Query knowledge source: $\{q_j^{\text{prior}}\} = \texttt{query\_KS}(X_b)$ {Retrieve relevant prior knowledge}
    for $i = 1$ to G **do**
      Generate sequence $o_i \sim \pi_\theta(\cdot \mid X_b)$ using model θ and policy π_θ
      Extract gene_information o_i^{relevant} from o_i (from `<gene>` tags)
      if gene_information o_i^{trace} exists **then**
        Score gene_information against prior knowledge: $r_i^{\text{soft}} = \nu(o_i^{\text{trace}}, \{q_j^{\text{prior}}\})$
      else
        $r_i^{\text{soft}} = 0$ {Penalize missing gene information}
      end if
      Add auxiliary rewards: $r_i = r_i^{\text{soft}} + r_{\text{format}} + r_{\text{mention}}$
    end for
  end for
  Compute normalized advantages $\hat{A}_{i,t}$ using Eq. (4)
  Update θ via GRPO objective (Eq. 2) with KL divergence penalty (Eq. 5)
end for
return $\tilde{\theta}$

Algorithm 4 Formatting Reward r_{format}

Require: Completion o_i
Require: Set of formatting constraints $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$
Ensure: Formatting reward $r_{\text{format}} \in [0, 1]$

Initialize score vector $s = []$
for each constraint $F_j \in \mathcal{F}$ **do**
  if F_j is satisfied in o_i **then**
    Append 1.0 to s
  else
    Append 0.0 to s
  end if
end for
$r_{\text{format}} = \frac{1}{|\mathcal{F}|} \sum_{j=1}^{|\mathcal{F}|} s_j$
return r_{format}

Algorithm 5 Mention Reward r_{mention}

Require: Completion o_i
Require: Set of desired terms $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$
Ensure: Mention reward $r_{\text{mention}} \in [0, 1]$
Extract reasoning trace o_i^{trace} from o_i (from `<think>` tags)
Initialize score vector $s = []$
for each term $t_j \in \mathcal{T}$ **do**
  if t_j appears in o_i^{trace} **then**
    Append 1.0 to s
  else
    Append 0.0 to s
  end if
end for
$r_{\text{mention}} = \frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} s_j$
return r_{mention}