Chan
Zuckerberg
Initiative

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, and Theofanis Karaletsos*

Chan Zuckerberg Initative, Redwood City, CA

*Corresponding authors: A.I. (aistrate@chanzuckerberg.com), T.K. (tkaraletsos@chanzuckerberg.com)

Abstract

Reasoning Models are typically trained against verification mechanisms in formally specified systems such as code or symbolic math. However, in open domains like biology, we do not generally have access to exact rules facilitating formal verification at scale, and oftentimes resolve to testing hypotheses in the lab to assess the validity of a prediction. Verification by performing real experiments is slow, expensive, and inherently does not scale with computation. In this work, we show that one can use world models of biology or other prior knowledge as approximate oracles over biological knowledge to utilize as soft verification to train reasoning systems without the need for additional experimental data. We introduce rbio1, a reasoning model for biology that is post-trained from a pretrained LLM using reinforcement learning and uses learned models of biology to obtain biological knowledge for verification during training. We show that soft verification successfully distills biology world models into rbio, at the example of achieving leading performance on perturbation prediction against the PerturbQA benchmark compared to state-of-the-art models; we demonstrate the benefits of compositions of verifiers to learn more general rbio models. We believe rbio provides a proof of concept that demonstrates that predictions from bio-models can be used to train powerful reasoning models using simulations, rather than experimental data, as a new training paradigm.

1 Introduction

Foundation models [1–3] have recently emerged as a successful approach to effectively capture biological data distribution by learning high-dimensional manifolds and generating embeddings that can be used at the core of a multitude of downstream applications. The utility of foundation models has been demonstrated on various problems [2, 4, 5], mostly making use of supervision signals such as labels. However, these applications rely on the availability of large amounts of annotated training data as well as the presence of an adequate computational infrastructure that can support the demanding requirements of foundation models. It would be beneficial then to mitigate these issues by "splicing" the knowledge captured by large foundation models trained on biology data into more flexible, interactive, and potentially smaller models, such as conversational LLMs. Moreover, the idea of virtual cell modeling has been put forward in the scientific literature as early as [6, 7], its promises reignited in [8] and explored in recent works [9–11]. The promise of virtual cell model systems is that of building powerful predictive models of biology that bypass the need for experimental data and are able to generate predictions for any cell state transitions, such as from diseased to healthy state and vice versa. Due to availability in compute, large-scale data generation, and technological machine learning advancements, there have been significant advances in the realm of foundation models for biology. Most of these models are built on a specific experimental modality, such as transcriptomics [10, 12–15], imaging [16], proteomics [2, 5], genomics [17] or even language [18] - with some work exploring models trained on multiple modalities [11, 19–23].

© 2025 The authors

This work is licensed under Creative Commons Attribution 4.0 International license (CC-BY).

*Corresponding authors: Ana-Maria Istrate, aistrate@chanzuckerberg.com; Theofanis Karaletsos, tkaraletsos@chanzuckerberg.com

Figuring out how to integrate separate models trained on diverse datasets and potentially independent modalities is the main challenge in achieving a true virtual cell model paradigm, where a shared representation space integrates knowledge across all these world models of biology [8]. To mitigate these issues and build towards a truly universal virtual cell system, we formulate the research goal as follows: develop a method that allows for integration of world models of biology into a common space, using language as the connecting modality. The benefit of this approach is that it also allows the transformation of complex biological models into conversational models that can engage users through natural language - so that experimentalists and computational users can easily interact with and access knowledge from world models of biology. By distilling a biological model into an LLM, we are able to distill knowledge derived from experimental data encoded in its structure and convert it into a model of natural language that enables interactive, human-readable dialogue. This can also be seen as an alignment of a reasoning LLM to a biology world model.

Hence, there are a number of main motivations behind this approach: first is to aggregate information from world models of biology into a universal representation space; second is to make the underlying science more accessible, allowing users to explore hypotheses, ask questions, and receive informed responses as if conversing with the model itself. And last but not least, if we can use powerful models of biology, instead of experimental data, to train models just as performant, we can bypass the need for large-scale data generation, or the requirement for full sharing or dissemination of often-times sensitive datasets, which can be slow, expensive and often unscalable.

In an environment where virtual cell models and other models of scientific datasets abound, finding means to integrate their semantics gracefully is paramount to fulfilling the vision of a general purpose model that can be used by experts across different domains and spans knowledge from both language based sources like papers and experimental domains like foundation models of raw biological data.

Unlike traditional biological models, which often require specialized knowledge to interpret, conversational models provide an accessible interface for querying and understanding system behavior. LLMs are also typically more generalizable, capable of adapting to a broader range of questions and scenarios than their biological counterparts, especially if they were pretrained on large language corpora containing facts from different domains. Additionally, their flexibility makes them powerful tools for hypothesis generation, explanation, and discovery, expanding how we engage with and extract insights from biological knowledge.

To achieve our goals, we leverage reasoning LLMs [24, 25], a powerful class of language models showing great promise in, e.g., medicine [26, 27] and chemistry [28] that can simulate complex biological interactions through dialogue. Inspired by this recent progress, we propose to employ reinforcement learning (RL), specifically GRPO [29], to encourage behaviors aligned with biological insights learned by world models of biology; this enables more accurate scientific reasoning and effective communication. Additionally, we make use of a new learning paradigm - reward formulation by a domain-specific biological model - to shape the model's learning process without relying solely on large amounts of labeled data, but instead use model predictions. The idea of using surrogate model predictions - rather than labeled data - has been applied to other domains in the general reinforcement learning literature [30] and black-box optimization [31, 32]. However, it remains a vibrant frontier in the domain of reasoning LLMs[33, 34], and we are, as far as we know, the first ones to apply it in the space of reasoning models for biology, which are largely trained on experimental data or rich domain knowledge. We refer to this concept as soft verification throughout the paper.

Distilling information from world models of biology into LLMs allows us to develop conversational models that are both scientifically grounded and adaptable to real-world inquiries. The intuition behind this is that foundation models do indeed sufficiently capture the manifold underlying biological data in a way that allows us to "walk" it in order to obtain more datapoints and their meaningful representations. This allows us to make appropriate comparisons and predictions that can be ultimately used as "soft" scoring signals for the textual reasoning traces produced by LLMs. This is also fundamentally different from optimizing LLMs for next token prediction on biology-specific text, as shown in [35–39]. We present the schematic representation of the proposed approach in Figure1. In terms of testing the capabilities of our models, we focus on genetic perturbation prediction - predicting the outcome of knocking down a gene on other genes in the cell - as the downstream task, given its centrality in understanding cell state transitions.

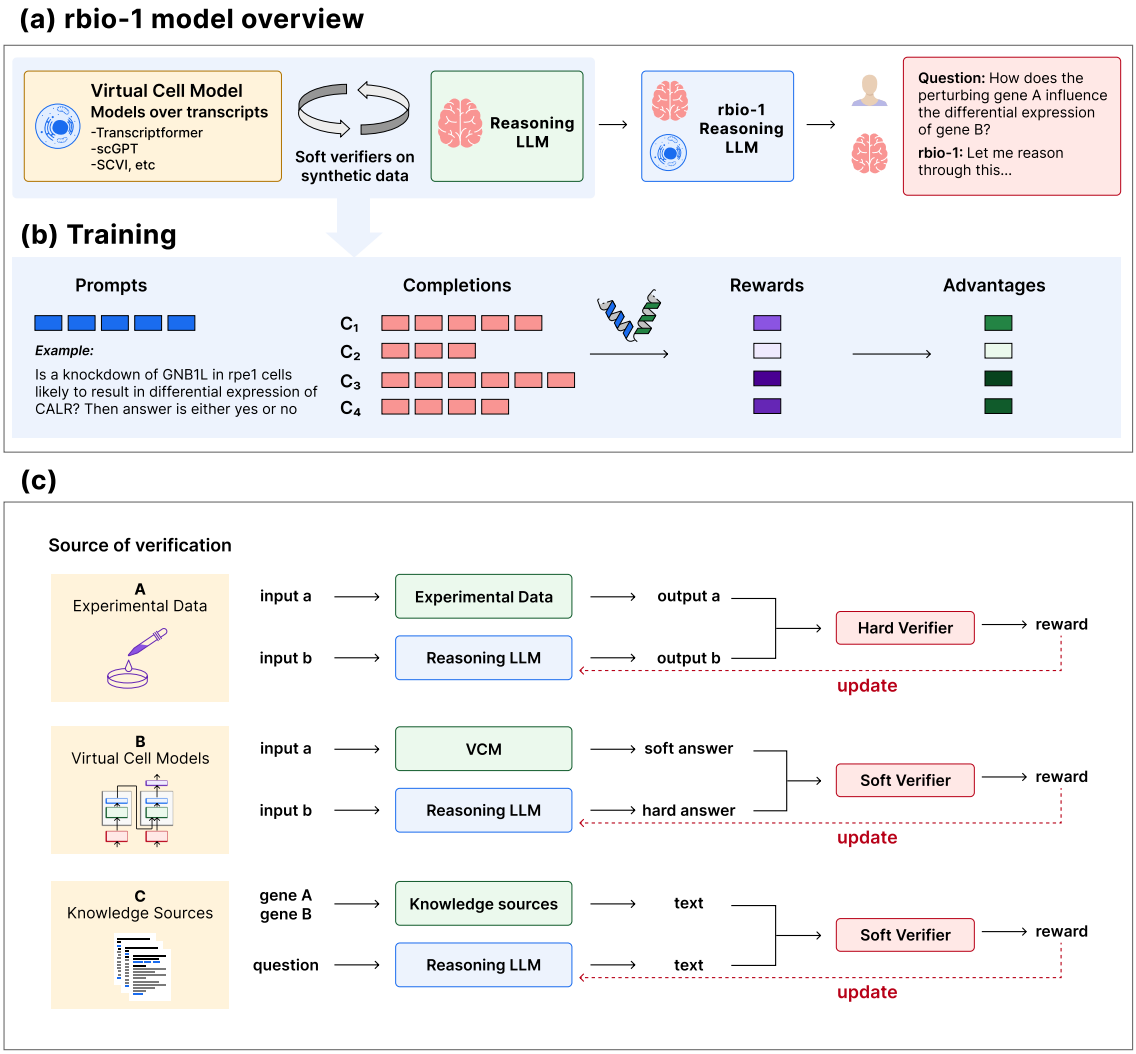


FIGURE 1. | **rbio1 Overview. (a)** Overview of rbio1. rbio1 distills biological information from Virtual Cell Models into Reasoning LLMs through soft verification - using simulations, rather than experimental data, as training signals. The output is a biologically-informed reasoning model that users can converse with in natural language. **(b)** Schematic of training process using GRPO. For each prompt, the LLM generates N completions, and each gets a reward based on a VCM response. Rewards get converted into advantages that guide the training signal **(c)** A general scheme for our soft supervision learning of reasoning LLMs. The red dashed arrow indicates the update flow for training with reinforcement learning (here: GRPO). **A**. An instance of the general hard supervision using experimental data to generate rewards as binary outputs (hard answer) about $gene_A$ being perturbed by $gene_B$. **B**. An instance of the general soft supervision with a biological model in the form of a neural network providing a probability (soft answer) about $gene_A$ being perturbed by $gene_B$, and a soft verifier being a set of rules determining the reward for the given probability and the answer (yes/no) provided by the reasoning LLM. **C**. An instance of the general soft supervision with Knowledge Sources - Gene Ontology (GO), turning the information about gene perturbation into facts expressed in natural language. The soft verifiers (keywords-based, ROUGE scores, likelihood estimation) take the thinking part of the answer given by the reasoning LLM and together with the description provided by GO, returns a reward.

Essentially, we are interested in differential expression - change of expression (up or down) of genes affected by gene knockdowns. We translate this into the following natural language query:

Is a knockdown of gene A in this particular cell line likely to result in differential expression of gene B?

This type of task-specific question in the field of transcriptomics is usually tackled in one of two ways: (1) - through bespoke models that are trained on task-specific data, or (2) - through foundation models pre-trained on large-scale corpora, adapted to downstream tasks either through finetuning or zero-shot capabilities. For the task of perturbation prediction, recent literature shows that bespoke models tend to outperform models that fall under the second category [40]. Creating bespoke models usually requires large amounts of specialized training data that has to be obtained experimentally in a lab, limiting training to data availability and resources. With the rise of foundation models for transcriptomics, there is tremendous potential for using the biological knowledge that models have learned during large-scale pre-training runs and apply it for task-specific questions - such as perturbation prediction - to inform prediction. However, prompting models of biology trained exclusively on biological data - and not language - for out-of-distribution tasks, is not a straightforward task. We show that on the PerturbQA dataset [41] - which contains single-gene perturbation knockdown datasets on four cancer cell lines (RPE1, K562, HEPG2, JURKAT), reasoning models trained using soft verifiers as reward signals learn to generalize on out-of-distribution datasets and tasks, bypassing the need to train on experimental data.

Our contributions can be summarized as the following:

(1) Proposing a mechanism for distilling knowledge from world models of biology into reasoning models by using their predictions during RL training. We refer to this mechanism as 'soft verification'. We coin the resulting suite of reasoning models rbio1
(2) We demonstrate that rbio1 can learn to reason about tasks like perturbation, for example predicting the effect of gene knockdowns on differential expression of other genes in the cell, by interacting with models simulating on-task with competitive performance to an rbio model trained against hard experimental data
(3) Further, we demonstrate that more general Virtual Cell Models (VCMs) can be used to tutor rbio1 models with soft verification to learn transferable general biological information, such as gene co-expression information or gene regulatory networks, leading to competitive performance with bespoke models or models trained on experimental data off-task in the perturbation setting from (2)
(4) We demonstrate that multiple sources of verification in the form of models can be composed using our method to train more powerful rbio models, thus providing a framework for a universal Virtual Cell Model that distills biological knowledge from diverse models trained on distinct biological tasks and modalities (i.e. evolutionary gene co-expression information from Transcriptformer, rich gene representation from ESM) into a shared representational space through language
(5) We show that probing models' reasoning capabilities at test-time through chain-of-thought increases model performance to state of the art on PerturbQA without tool-use or access to experimental data at test time.

2 Methods

In this section we describe our methodology, starting with reviewing reinforcement learning for reasoning models as the basis for hard verification using experimental data. We then describe two novel forms of reward mechanisms for biological reasoning models, based on utilizing soft verification stemming from models of biology and prior knowledge, rather than hard experimental data. We contextualize both against concrete instantiations for the relevant application on perturbation; however the same approach can be readily be applied with different models and mechanisms to other tasks and domains.

2.1 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 - which comes from the supervised finetuned LLM; r_ϕ a reward model emitting rewards r_i; $L_{GRPO}(\theta)$ the surrogate objective and β the coefficient used for the KL penalty.

Given these variables, Group Relative Policy Optimization (GRPO)[29] training maximizes the following objective function, with the goal of increasing the accumulated collective rewards $\{r_{i,\geq t}\}$:

$$J_{GRPO} = E_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(O|q)}\left[L_{GRPO}(\theta)\right] \tag{1}$$

We use the clipped surrogate objective version of L_{GRPO}, namely:

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

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

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

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

For a task with a binary outcome and a given verifier V that can be used to emit rewards, we can change the reward r_i in Equation 4 to the following under $r_\phi = r_V$:

$$r_i(q,o_i) = r_i\phi(q,o_i) = r_iV(q,o_i) = p(q,o_i|V) \tag{6}$$

where $p(q|V)$ is the probability of q, o_i under V.

2.2 Rbio-Hard Verification

For example, for the task of perturbation prediction, we have multiple types of verifiers, as showcased in Figure 1c - each yielding a different type of reward that can be plugged into Equation 6. Since the query q is dependent on interaction of two genes: $gene_A$ and $gene_B$, we make this explicit in the notation. If we have access to a dataset of experimental data D_{EXP}, then $V = \{D_{EXP}\}$ and the reward function r_i becomes:

$$r_i(q,o_i) = r_i(q_{gene_A,\ gene_B}, o_i) \tag{7}$$

$$r_i(q,o_i) = p(q_{gene_A,\ gene_B}, o_i|D_{EXP}) \tag{8}$$

$$r_i(q,o_i) = \begin{cases} 1 \text{ if } o_i = \text{True and } D_{EXP}(q_{gene_A,\ gene_B}) = \text{True} \\ 1 \text{ if } o_i = \text{False and } D_{EXP}(q_{fene_A,\ gene_B}) = \text{False} \\ 0 \text{ else} \end{cases} \tag{9}$$

Essentially - for a query related to a pair of genes $gene_A, gene_B$ - if we predict an outcome that is the same as that which we can verify experimentally - whether the outcome is positive, or negative - the model gets a positive reward. We refer to this as *hard verification* - as outlined in Figure 1c - A.

2.3 Rbio-VCM: using models as a reward mechanism with soft verification

Besides $V = \{D_{EXP}\}$, we also have access to bio-models as sources of verification, which we can use in the absence of D_{EXP}. In this case, the emitted rewards will not be binary $\in \{0, 1\}$, but will take the form of a probability of success emitted by a verifier V based on a bio model $M_{bio} \in \{MLP, VCM\}$. This idea of generating *soft verification* scores using biomodels M_{bio} to generate predictions instead is outlined in Figure 1c - B. To summarize:

$$r_i(q, o_i) = p(q, o_i|V) = \begin{cases} r_i^{hard}(q, o_i) \text{ if } V = D_{EXP} \\ r_i^{soft}(q, o_i) \text{ if } V \neq D_{EXP} \end{cases} \tag{10}$$

with

$$p(q, o_i|V) = \begin{cases} \in \{0, 1\} \text{ if } V = D_{EXP} \\ 0 \leq p \leq 1 \text{ if } V \neq D_{EXP} \end{cases} \tag{11}$$

Hence, if we have a learned model of biology $M_{bio} = \{MLP, VCM\}$ that can answer questions about $q(gene_A,\ gene_B)$ then Equation 6 becomes:

$$r_i^{soft}(q_{gene_A,\ gene_B}, o_i) = p_(q_{gene_A,\ gene_B}, o_i|M_{bio}) \tag{12}$$

For q as a perturbation prompt and M_{bio} as a simple model of perturbation prediction, such as an MLP, Equation 12 becomes:

$$r_i^{soft}(q^{perturbation}_{gene_A,\ gene_B}, o_i) = p(q^{perturbation}_{gene_A,\ gene_B}|M_{bio} = \text{MLP}) \tag{13}$$

$$r_i^{soft}(q^{perturbation}_{gene_A,\ gene_B}, o_i) = p(q_{gene_A,\ gene_B}|\text{MLP}) \tag{14}$$

$$0 \leq r_i^{soft}(q^{perturbation}_{gene_A, gene_B}, o_i) \leq 1 \tag{15}$$

For q as prompts related to co-expression interactions between genes, we use pointwise mutual information scores (PMIs) between pairs of genes retrieved from Transcriptformer. Equation 12 becomes:

$$r_i^{soft}(q^{co-expression}_{gene_A,\ gene_B}, o_i) = p(q^{co-expression}_{gene_A,\ gene_B}|M_{bio} = \text{Transcriptformer}) \tag{16}$$

$$r_i^{soft}(q^{co-expression}_{gene_A,\ gene_B}, o_i) = PMI(gene_A,\ gene_B|\text{Transcriptformer}) \tag{17}$$

$$r_i^{soft}(q^{co-expression}_{gene_A,\ gene_B}, o_i) = \log\left(\frac{P(gene_A,\ gene_B)}{P(gene_A)P(gene(B)}\right) = \log\left(\frac{P(gene_A|gene_B)}{P(gene_A)}\right). \tag{18}$$

We retrieve PMIs for all gene pairs present in the Transcriptformer pre-training datasets and include the top 0.05 interactions with highest scores in rbio experiments that include Transcriptformer. We also include a small number of negative interactions from the bottom 0.05 pairs with lowest PMIs from Transcriptformer.

2.4 Rbio-Prior: using prior knowledge as a reward mechanism with soft verification

In the previous case, our queries q are in-distribution for both the experimental datasets we have available D_{EXP} - as showcased in Equation 6 and for the models of biology M_{bio} as showcased in Equation 12. We can also use Knowledge Sources - denoted as - KS, such as the $KS = GO\ Database$ to answer the same types of questions - as outlined in Figure 1c - C.

$$r_i^{soft}(q_{gene_A,\ gene_B}, o_i) = p(q_{gene_A\ gene_B}|\text{Knowledge Source KS}) \tag{19}$$

$$0 \leq r_i^{soft}(q_{gene_A,\ gene_B}, o_i) \leq 1 \tag{20}$$

We experiment with three types of reward calculation based on knowledge sources: ROUGE-based scores, keywords based-scores and likelihood estimations, all of which require querying the GO ontology for annotations for $gene_A$ and $gene_B$

$$gene_A^{annot|KS} = query_KS(gene_A) \tag{21}$$

$$gene_B^{annot|KS} = query_KS(gene_b) \tag{22}$$

$$r_i^{soft}(q_{gene_A,\, gene_B}, o_i) = r_i^{soft|KS}(gene_A^{annot}, o_i) + r_i^{soft|KS}(gene_B^{annot}, o_i) \tag{23}$$

where

$$r_i^{soft|KS}(s_1, s_2) = \begin{cases} ROUGE_1(s_1, s2), ROUGE_2(s_1, s2), ROUGE_{LCS}(s_1, s_2) \text{ for ROUGE-based verifiers} \\ KWS(s_1, s2) \text{ for keywords-based verifiers} \\ LL(s_1, s2|\pi) \text{ for likelihood-based verifiers} \end{cases} \tag{24}$$

For instance, for ROUGE-based verifiers, Equation 23 becomes:

$$r_i^{soft|KS,ROUGE}(q_{gene_A\, gene_B}, o_i) = \sum_X ROUGE_X(gene_A^{annot}, o_i) + ROUGE_X(gene_B^{annot}, o_i) \tag{25}$$

where

$$\text{ROUGE-1}(s_1, s_2) = \frac{2 \cdot \#\text{unigrams}(s_1 \cap s_2)}{\#\text{unigrams}(s_1) + \#\text{unigrams}(s_2)} \tag{26}$$

$$\text{ROUGE-2}(s_1, s_2) = \frac{2 \cdot \#\text{bigrams}(s_1 \cap s_2)}{\#\text{bigrams}(s_1) + \#\text{bigrams}(s_2)} \tag{27}$$

$$\text{ROUGE-L}(s_1, s_2) = \frac{2 \cdot LCS(s_1, s_2)}{|s_1| + |s_2|} \tag{28}$$

where s_2 is the reference, LCS represents the longest common subsequence and $\cap$ = multiset intersection (counts included up to the minimum frequency in both texts).

For keywords-based verifiers, Equation 23 becomes:

$$r_i^{soft|KS,keywords}(q_{gene_A\, gene_B}, o_i) = KWS_X(gene_A^{annot}, o_i) + KWS_X(gene_B^{annot}, o_i) \tag{29}$$

$$KWS(s_1, s_2) = \frac{|s1 \cap s2|}{|s_1|} \tag{30}$$

where s_1 corresponds to a gene annotation retrieved from the GO Ontology - essentially counting the number of keywords in gene annotations used in the reasoning trace, aligning the model with existing sources of scientific knowledge.

For likelihood-based verifiers, Equation 23 becomes:

$$r_i^{soft|KS,likelihood}(q_{gene_A,\, gene_B}, o_i) = LL(gene_A^{annot}|\pi_\theta) + LL(gene_B^{annot}|\pi_\theta) \tag{31}$$

$$LL\big(gene_i^{\text{annot}} \mid \pi_\theta\big) = \sum_{j=1}^{m_i} \log p_{\pi_\theta}\big(t_j \mid t_{<j}\big) \tag{32}$$

2.5 Composable Verification for model integration

For cases where we have multiple types of verifiers V_j emitting rewards $r_{i,j}$, Equation 6 becomes:

$$r_i(q, o_i) = \sum_{r_i, j} r_{i,j}(q, o_i|V_j) \tag{33}$$

where $V_j = \{D_{EXP}, Model(D_{EXP}\}$ is a verifier that can be based either on available experimental data or models of experimental data.

2.6 Normalization For Soft Scores Verifiers

Most of the soft verifiers mentioned in previous sections will not emit a soft score that is within the [0, 1] range, which means that it will have to be normalized in order to be used as a reward. Hence - for a pair of genes $(gene_A, gene_B)$, if we have a soft verifier that emits scores between $[x_{min}, x_{max}]$ representing how confident the soft verifier is that perturbing $gene_A$ would lead to an increase of $gene_B$, or, in the case of Transcriptformer, how confident the model is that $gene_A$ and $gene_B$ will be co-expressed together, we will have a significance *threshold* such as:

(1) $\forall x, x \in [x_{min}, threshold]$ - we believe the verifier says the effect is not significant

(2) $\forall x, x \in [threshold, x_{max}]$ - we believe the verifier says the effect is significant

For example, an MLP emits probabilities in the range [0, 1] with $threshold_{MLP} = 0.5$.

Then, we will have two distinct situations:

1. We need to normalize the scores from (1) into the [0, 0.5] range

For this, we use min-max normalization to essentially normalize $x_{below} \in [x_{min}, threshold]$ to $x_{below-norm} \in [0, 0.5]$ First, we normalize $x_{below} \in [x_{min}, threshold]$ to $x_{below-norm} \in [0, 1]$ by

$$x_{below-norm} = \frac{x - x_{min}}{x_{max} - x_{min}} = \frac{x - x_{min}}{threshold - x_{min}} \tag{34}$$

Then, we shift from the [0, 1] range to [0, 0.5] through:

$$x_{below-norm} = 0.5\frac{x - x_{min}}{threshold - x_{min}} \tag{35}$$

2. We need to normalize the scores from (2) into the [0.5, 1] range

We can also use min-max normalization to normalize $x_{above} \in [threshold, x_{max}]$ to $x_{above-norm} \in [0.5, 1]$ First, we normalize $x_{above} \in [threshold, x_{max}]$ to $x_{above-norm} \in [0, 1]$ by

$$x_{above-norm} = \frac{x - x_{min}}{x_{max} - x_{min}} = \frac{x - threshold}{x_{max} - threshold} \tag{36}$$

Then, we shift from the [0, 1] range to [0.5, 1] through:

$$x_{above-norm} = 0.5 + 0.5\frac{x - threshold}{x_{max} - threshold} \tag{37}$$

We use these normalization techniques for all the soft verifiers we use.

3 Results

We structure the results section as follows: we start with a discussion on the metrics we used, and then go into details into the five main results - (1) Our proposed methodology and introducing the idea of *soft verification* for the task of perturbation prediction, (2) Training rbio models with soft verification on task simulations, and showing that they perform competitively with models trained on hard experimental data, (3) Using Virtual Cell Models (VCMs) to teach rbio models off-task biology that can exhibit transfer to perturbation, (4) Exploring the effect of combinations of biological verifiers and showing they improve generalization and (5) Showing that Chain-Of-Thought prompting at test-time elevates rbio models to state-of-the-art performance on the PerturbQA benchmark.

3.1 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 [41]. We compute the following metrics:

$$Recall\ (TPR) = \frac{TP}{TP + FN} \tag{38}$$

$$TNR = \frac{TN}{TN + FP} \tag{39}$$

$$Precision = \frac{TP}{TP + FP} \tag{40}$$

$$F1\ Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{41}$$

$$Balanced\ Accuracy = \frac{TPR + TNR}{2} \tag{42}$$

$$MCC\ (Matthews\ Correlation\ Coefficient) = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{43}$$

Within the perturbation prediction setting, we are interested in whether perturbing a gene will have a significant effect on other genes in the cells, so TPR (True Positive Rate) is a metric we value higher than TNR (True Negative Rate) - as positives are fewer and more significant, whereas negatives will be overhwhelmingly present in cells. The MCC (Matthews Correlation Coefficient) is a measure of correlation between the ground truth and predictions, taking all of TP, TN, FP and FN into consideration. Given that our datasets are heavily imbalanced, as seen in Figure 5d, and the significant of true positives, we believe that among the metrics considered, TPR, coupled with MCC are more telling of model performance on this task, although Balanced Accuracy and F1-Score are also good signals, including the effect of TNs. We make all metrics available for transparency and completion.

3.2 rbio1 introduces a mechanism to distill the knowledge from world models of biology into reasoning models through soft verification, a training paradigm that uses simulations, rather than hard ground-truth data during RL training

The basic training paradigm of reasoning models trained through reinforcement learning consists of the following elements: a dataset of prompts that are used to prompt an LLM and generate completions, a verifier capable of verifying the quality of the generations against a given set of rules, and a reward mechanism that assigns rewards based on the output from the verifier. Most verifiers used in the general literature output precise verifications - such as - code runs/or not (for coding generation), the math equation is valid or not (for math solving capabilities). In the case of biological datasets, when experimental data is available, certain outcomes are verifiable. For example, we can verify an answer to the following question: ***Is a knockdown of AARS in hepg2 cells likely to result in differential expression of ATAD2B?*** by testing its outcome in a lab. We can then use the outcome of the experimental data to assign rewards during reinforcement learning training and guide the model towards outcomes aligned with experimental data (*hard verification*). However, not all biological statements, or even outcomes of pairs of gene interactions, are verifiable in a lab, simply due to the sheer volume of possible interactions. This is in fact the promise of virtual cell models of biology [8] - bypassing the need to verify outcomes experimentally, and instead using virtual systems to generate predictions. Here, we propose using these predictions as verifications during RL training. In this way, we use virtual cell models as oracles capable of emitting answers to biological questions under certain uncertainties given their capabilities. We refer to this mechanism as *soft verification*.

Assume we are interested in training a reasoning model to get really good at answering the types of genetic perturbation queries such as the one above - ***Is a knockdown of AARS in hepg2 cells likely to result in differential expression of ATAD2B?***. During training, for a given a biological query q_i, the reasoning LLM will generate a number of completions. For a given completion c_j, we can reward the model in multiple ways:

(1) If the outcome of qi has been validated experimentally and we have access to it in a dataset D_{EXP}, we can verify c_j directly and emit a reward. Since the answer is either yes/no, the reward generated will be binary: $r_{hard}(q_i; c_j) = 1$ for yes, 0 for no; this is *hard verification*.
(2) If the outcome of q_i has not been validated experimentally, then we can verify c_j by using predictions from learned models of biology (assuming we have access to a model M_{bio} that is able to emit such predictions). In this case, the reward generated will be a function of M_{bio}: $r_{soft}(q_i; c_j) = M_{bio}(q_i; c_i)$, $0 \leq r_{soft}(q_i; c_j) \leq 1$; this is soft verification.
(3) Lastly, we can verify q_i using Established Knowledge Sources which have scientifically validated information on biological entities - such as the GO Ontology [42, 43] which gives us gene-level annotations across multiple axes: *molecular processes*, *cellular components* and *biological processes*. For this approach, we look at different types of reward estimations: *keywords-based*: where we retrieve gene annotations for both the knocked out gene and the monitored gene and we reward the model for using as many of the keywords mentioned in the gene annotations as possible; *scores based on ROUGE* [44] - a semantic similarity measure - where we compute the ROUGE-1 (based on unigrams overlap), ROUGE-2 (based on bigrams overlap) and ROUGE-L (based on Longest Common Subsequence) metrics between the gene annotations and the gene information the model uses in the reasoning trace. For the keywords-based and ROUGE-based metrics, we prompt the model to tell us what gene information it is using during reasoning by asking it to provide it inside the `<gene_info></gene_info>` tags. Lastly, we experimented with scoring the *likelihood* of the gene annotations retrieved from GO Ontology under our own reasoning model - the idea being that we want to stir the model towards higher likelihood of scientifically accurate facts, minimizing the potential hallucinations.

We showcase all the different types of verifiers in Table 1 and offer detailed descriptions under the Methods section 2.

TABLE 1. Verifiers used during RL training and their descriptions, as well as example prompts. Verifiers: "Exp" is experimental; "MLP" is multi-layer perceptron; "TF" is Transcriptformer; "GO" is Gene Ontology.

Type	**Verifier**	**Description**	**Example Prompts**	**Rewards**	**Source**
HARD	EXP	PerturbQA training data	*Is a knockdown of AARS in hepg2 cells likely to result in differential expression of ATAD2B?*	Binary: 1 for correct, 0 for incorrect; $r_{binary} \in \{0, 1\}$	Experimental data
SOFT	MLP	MLPs trained on perturbation data using gene embeddings	*Is a knockdown of AARS in hepg2 cells likely to result in differential expression of ATAD2B?*	Prediction score from MLP; $r_{soft} = p$ where $0 \leq p \leq 1$ and $p = \text{MLP}(\text{gene}_A, \text{gene}_B)$	VCM
SOFT	TF	Transcriptformer Foundation Model	*If transcription factor ADNP is activated, is expression of gene RABGAP1 going to be high? The answer is either yes or no.*	Pointwise Mutual Information (PMI) scores: $r_{soft} = p$ where $0 \leq p \leq 1$ and $p = \text{TF}(\text{gene}_A, \text{gene}_B)$	VCM
SOFT	GO	Knowledge Database on Genes	*Is a knockdown of AARS in hepg2 cells likely to result in differential expression of ATAD2B?*	ROUGE scores, keywords mentions of GO annotations, likelihood estimation $p = \text{GO}(\text{gene}_A, \text{gene}_B)$	Knowledge Base

3.3 rbio models trained with soft verification on tasks like perturbation generalize on out-of-distribution perturbation datasets, showing that task simulation trains competitive models with hard experimental data

We show that on the PerturbQA dataset [41] - which contains CRISPRi single-gene perturbation knockdown datasets on four cancer cell lines (RPE1, K562, HEPG2, JURKAT) - models trained using soft verifiers learn to generalize on out-of-distribution cell lines, potentially bypassing the need to train on cell-line specific experimental data. We start by evaluating a simple, yet powerful predictive model for single-gene knockdown perturbation prediction: an MLP trained to predict differential expression of gene knockdown perturbation. This choice is justified given that recent literature has shown that simple models still surpass more complicated model architectures for this task [40]. We trained a 2 layers-deep MLP with 64 hidden neurons on differential expression responses from three cell lines and then used it to generate binary predictions on the fourth cell line. We used the predictions on the fourth cell line as soft verification scores during RL training. This procedure is shown in Figure 2c. We experimented with using one-hot-encoders, gene2vec [45] and ESM [5] embeddings for the gene representations for the MLP. We show that models trained in this way, *rbio-MLP-leave-one-out-one-hot* and *rbio-MLP-leave-one-out-gene2vec*, achieve performance comparable to models trained solely on experimental data.

We compare with models trained on experimental data in two settings: *rbio-EXP-one-cell-line* - model trained on experimental data coming from one cell line that is in-distribution - test and train splits are from the same cell line - showcased in Figure 2a - and *rbio-EXP-leave-one-out* - model trained on experimental data coming from three cell lines that are not used at test time, emulating testing on out of-distribution data - showcased in Figure 2b. We benchmark against SUMMER[41], a state-of-the-art model trained on experimental data, as well as carefully crafted prompts that include rich domain knowledge.

As shown in Figure 2d (aggregated metrics) and Figure 2e (metrics split by cell line), we notice that *rbio-MLP-leave-one-out-one-hot* and *rbio-MLP-leave-one-out-gene2vec* both achieve performance comparable with *rbio-EXP-leave-one-out* across F1-score (0.65/0.66 compared vs 0.67) and MCC (0.60 for both rbio-MLP models vs 0.61 for *rbio-EXP-leave-one-out*). Notably, the *rbio-MLP* models surpass both *rbio-EXP-leave-one-out* and SUMMER in Balanced Accuracy by increasing the TPR (0.77/0.76 for rbio-MLP, compared to 0.68 for *rbio-EXP-leave-one-out* and 0.63 for SUMMER) while maintaining similar TNR. Within the perturbation prediction setting, we're interested in whether perturbing a gene will have a significant effect on other genes in the cells, so TPR is a metric we value higher than TNR - as positives are fewer and more significant. Our results show that using predictions from models of experimental data, rather than experimental data itself can lead to similar results comparable in performance, potentially bypassing the need to train on the latter.

We noticed an interesting effect when comparing the *rbio-MLP* models with *rbio-EXP-one-cell-line* and *rbio-EXP-leave-one-out*. Firstly, rbio models trained on in-distribution cell lines - *rbio-EXP-one-cell-line* - outperform models trained on out-of-distribution cell lines - *rbio-EXP-leave-one-out* - suggesting that there might not be a beneficial transfer learning by training on different cell lines. Secondly, rbio models trained with MLPs in the loop get close to these models in balanced accuracy and TPR - we are catching just as many true positives; however, this also means that more false positives are generated, and models trained with experimental data in the loop still lead in F1-Score. Notably, all the reasoning models we evaluated outperform GEARS [46], a specialized model for perturbation prediction that has bespoke model architecture choices embedded into its design and is trained solely on experimental data, and with rich knowledge information, across all metrics. All models also surpass the base Qwen2.5-3b model, showing that the post-training with RL using predictive models of biology in the loop improves baseline reasoning models and is a valid strategy for embedding biological signals into reasoning LLMs.

Overall, the fact that rbio models trained with predictions from MLPs built on out-of-distribution data get close in Balanced Accuracy and TPRs to models trained on in-distribution experimental data speaks to a potentially significant paradigm: we can use predictive models of biology to generate simulations on synthetic data and use them as noisy signals during training and achieve performance comparative with models trained solely on experimental data obtained in a lab. Simply put, we can bypass the need for niche domain knowledge or large-scale experimental data by training on simulations instead.

FIGURE 2. | **Model performance for models trained on experimental and soft verification using MLP signals: (a) rbio-EXP-one-cell-line**. Training using experimental data from cell line A, testing on cell line A (in-distribution). **(b) rbio-EXP-leave-one-out**. Training using experimental data from all the cell lines not used during testing - i.e. training on cell lines B, C, D, testing on cell line A - (out-of-distribution) **(c) rbio-MLP-leave-one-out**. Training using predictions on cell line A coming from an MLP trained on out-of-distribution cell lines (i.e. B, C, D). **(d) Aggregate metrics for models trained with soft verifiers vs experimental data.** Metrics are computed for each cell line and aggregated over 5 different generation runs. Each bar corresponds to a mean performance aggregated across all four cell-lines: K562, RPE1, JURKAT and HEPG2. *rbio-MLP-leave-one-out-one-hot* uses one-hot representations for gene embeddings and *rbio-MLP-leave-one-out-gene2vec* uses gene2vec as gene representations. *SUMMER* is a model trained with experimental data and detailed biological knowledge in the loop; *GEARS* is a state-of-the-art specialized perturbation prediction model trained solely on experimental data. *Qwen2.5-3b* is the base reasoning model. **(e) Metrics Split by cell line**

3.4 Virtual Cell Models teach rbio off-task biology exhibiting transfer to perturbation

We trained an rbio model with prompts on gene co-expression data - such as *If transcription factor ADNP is activated, is expression of gene RABGAP1 going to be high? The answer is either yes or no.* and rewarded the model with scores emitted by the Transcriptformer [10] model. Specifically, we've used pointwise mutual information scores between pairs of genes from prompting Transcriptformer as a measure of gene co-expression. We then tested the model on the task of perturbation prediction, and noticed a significant improvement compared to baseline across F1-Score (*rbio-TF*: 0.29 vs base: 0.23), Balanced Accuracy (*rbio-TF*: 0.59 vs 0.52). While the TPR decreases (*rbio-TF*: 0.24 vs 0.49), the TNR improves significantly (*rbio-TF*: 0.94 vs base 0.55). Notably, the MCC increases significantly (*rbio-TF*: 0.21 vs base: 0.03) suggesting that outputs from models trained with Transcriptformer in the loop are more correlated with the true test data, while the base model offers no significant correlation. This is significant because *rbio-TF* has been trained with prompts completely unrelated to perturbation prediction, and has learned to transfer information about genes from its large-scale pretraining using cross-evolutionary transcriptomics information onto the base reasoning model, improving on the out-of-distribution task of gene knockdown prediction. Notably, the performance of the *rbio-TF* is on par with GEARS, a specialized perturbation prediction model across F1-score (0.30), Balanced Accuracy (0.59), with *rbio-TF* leading in TPR (0.24 vs 0.18), but lagging in MCC (0.21 vs 0.37). See Figure 3 for the results.

Moreover, whereas in the previous section we used MLP models trained on one-hot and gene2vec as gene representations, we have also experimented with training an rbio model on predictions emitted by MLPs using ESM - a protein language model - as a gene representation - *rbio-MLP* (ESM). This model is trained on simulated predictions, rather than experimental data, across all four cell lines. As shown in Figure 3, this model is very close in performance with *rbio-EXP-all-cell-lines*: a model trained on the training splits of all four cell lines, across all metrics: F1-score (*rbio-MLP-ESM*: 0.67 vs rbio-EXP-all-cell-lines: 0.75), Balanced Accuracy (*rbio-MLP-ESM*: 0.85 vs *rbio-EXP-all-cell-lines*: 0.88), MCC (*rbio-MLP-ESM*: 0.62 vs *rbio-EXP-all-cell-lines*: 0.71), TPR (*rbio-MLP-ESM*: 0.81 vs *rbio-EXP-all-cell-lines*: 0.83). Hence, we can use the rich information provided by the ESM embeddings to build relatively simple predictive models based on available data and use simulations that are close in performance to models of experimental data, transferring the biology learned by ESM embeddings into out-of-distribution tasks, such as genetic knockdown perturbation prediction.

3.5 Models trained on combinations of biological verifiers improve generalization

Whereas training on individual soft verifiers can get us close to training on experimental data, training models on multiple sources of verification generally improves with each added source, showcasing the additive and potentially complementary value of each verifier. In Figure 4, we observe that *rbio-TF+GO+MLP-ESM* - a model trained on combinations of simulations from MLP, Transcriptformer gene co-expression data, and Gene Ontology (GO) verifiers significantly outperforms *rbio-TF* that is trained solely on Transcriptformer gene co-expression data. Performance of this additive model, which is trained on combinations of

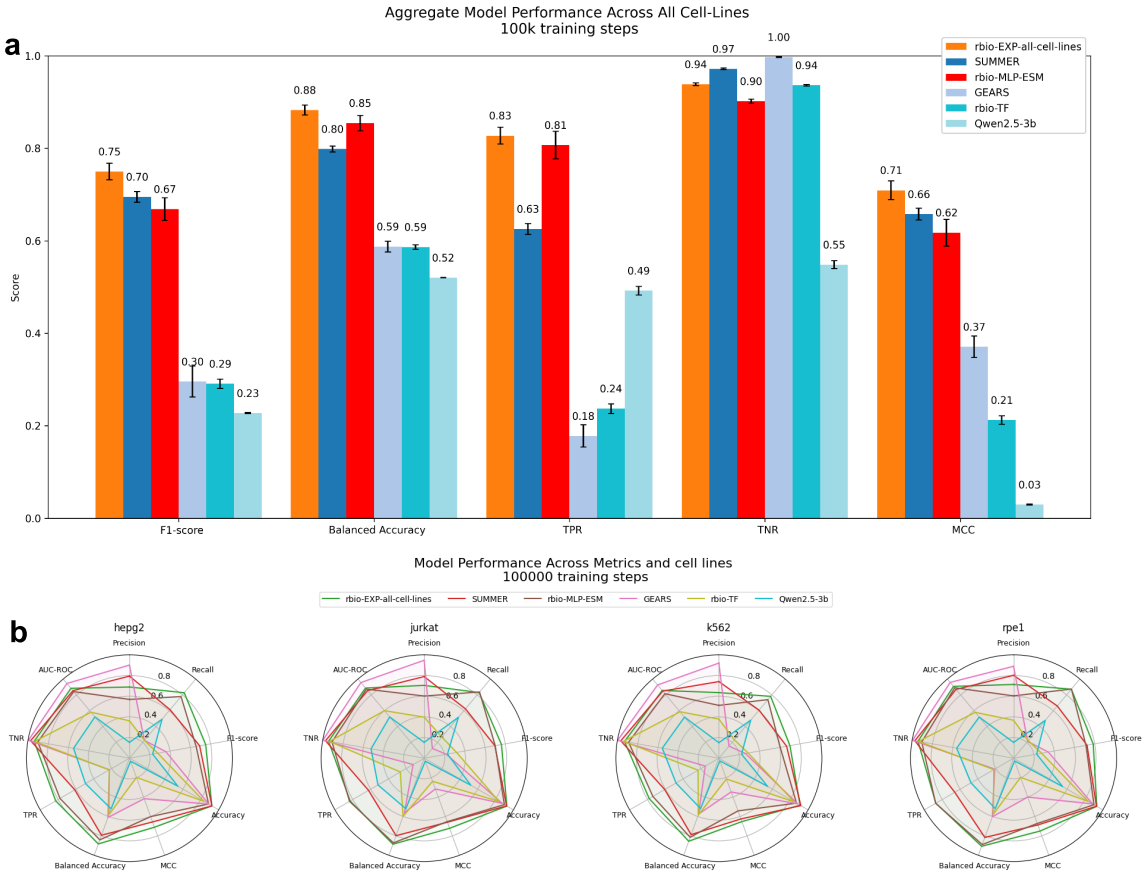


FIGURE 3. | **Model performance for models trained on experimental and soft verification using complex VCMs**: **(a) Aggregate model performance for models trained with soft verifiers and experimental data.** Metrics are computed for each cell line and aggregated over 5 different generation runs. Each bar corresponds to a mean performance aggregated across all four cell-lines: K562, RPE1, JURKAT and HEPG2. *rbio-EXP-all-cell-lines* corresponds to rbio trained on training splits of all four cell lines, *rbio-MLP-ESM* corresponds to rbio trained on predictions on all four cell lines generated by MLPs using ESM embeddings as gene representations trained on out-of-distributions cell lines (i.e.: trained on K562, RPE1, JURKAT and used to predict on the HEPG2 cell-line). Evaluation is on the test splits of the four cell lines. **(c) All metrics model performance split by cell-lines: K562, RPE1, JURKAT and HEPG2.**

simulations, and no experimental data, gets almost as high as models trained solely on experimental data: *rbio-EXP-all-cell-lines* and SUMMER, as shown in Figure 4b and Figure 4c.

When performing individual ablation studies of additions of verifiers, we observe a similar trend: adding verification sources keeps improving performance. For instance, we see in Figure 5a *rbio-TF+EXP* outperforms *rbio-TF*, showcasing that adding experimental data is still beneficial to models trained on Transcriptformer alone. Moreover, *rbio-TF+GO+EXP* outperforms *rbio-TF+EXP*, continuing to showcase that adding information from the GO ontology keeps improving model capabilities. Additionally, *rbio-TF+GO+EXP+MLP* continues to increase in performance. We want to note that during training, at each step, the model sees a prompt with a probability proportional to the density of that verification source in the training data. The distribution of data for each of the verifiers is present in Figure 4a: the majority of the training data will be skewed to gene co-expression data from Transcriptformer. Hence, each of the models trained on aggregations of verifiers that contains Transcriptformer predominantly sees Transcriptformer-related prompts due to the large percentage of the training data that comes from this verification source. However,

by adding small amounts of verification from other sources, model performance increases, showing the power of the diversity of verification sources in improving performance.

For instance in Figure 5b, we find that *rbio-TF+GO+MLP* significantly outperforms *rbio-TF* across metrics: F1-score (*rbio-TF*: 0.29, *rbio-TF+GO+MLP*: 0.68), Balanced Acccuracy (*rbio-TF*: 0.59, *rbio-TF+GO+MLP*: 0.79), TPR (*rbio-TF*: 0.24, *rbio-TF+GO+MLP*: 0.61) and MCC (*rbio-TF*: 0.21, *rbio-TF+GO+MLP*: 0.64). However, as shown in Figure 5b (lower left) adding experimental data leads to a smaller increase for *rbio-TF+GO+MLP*. For example: F1-score (*rbio:TF+GO+MLP*: 0.68, *rbio:TF+GO+MLP+EXP*: 0.69), MCC (*rbio:TF+GO+MLP*: 0.64, *rbio:TF+GO+MLP+EXP*: 0.65), Balanced Accuracy (*rbio:TF+GO+MLP*: 0.79, *rbio:TF+GO+MLP+EXP*: 0.79), TPR (*rbio:TF+GO+MLP*: 0.61, *rbio:TF+GO+MLP+EXP*: 0.60). We notice a counterintuitive trend when comparing *rbio-GO*, *rbio-EXP* and *rbio-GO+EXP*, where adding GO-based soft verifiers seems to hurt performance across metrics, as presented in Figure 5c. This is something that we need to look into further.

Overall, our analyses indicate that we can keep increasing performance by training models on a diverse set of datasets spanning multiple biological tasks and information, such as gene-to-gene interactions and co-expression data from Transcriptformer, task-specific experimental data and knowledge-based verification sources such as the GO ontology.

3.6 Chain-Of-Thought prompting at test-time elevates rbio models to state-of-the art on PerturbQA

While the majority of our efforts have focused on improving reasoning abilities at training time, we also show that through Chain-of-thought [47], we can improve performance of rbio models significantly at test time. As shown in Figure 6, asking the model to offer Chain-of-thought reasoning improves model performance across all metrics and rbio model variations, surpassing other state-of-the-art models, such as SUMMER. We experimented with multiple prompts, but one of the chain-of-thought prompting techniques we found to help the most is adding the following suffix to the system-prompt: **The Biologist will evaluate each step of this problem, using logical reasoning and evidence from the prompt**.

Models performance generally increases across the board. For instance, *rbio-TF+GO+MLP* jumps in F1-score from 0.68 to 0.74, *rbio-EXP-all-cell-lines* from 0.75 to 0.79 and *rbio-MLP-ESM* from 0.67 to 0.71). Other metrics see similar trends: *rbio-TF+GO+MLP* increases in Balanced Accuracy from 0.79 to 0.83, *rbio-EXP* from 0.88 to 0.91 and *rbio-MLP-ESM* from 0.85 to 0.89), with notable improvements in TPR: *rbio-TF+GO+MLP* from 0.61 to 0.68, *rbio-EXP* from 0.83 to 0.87 and *rbio-MLP-ESM* from 0.81 to 0.87). We do notice a somewhat counterintuitive effect on *rbio-TF*, where this type of CoT prompting seems to decrease performance, which we hypothesize can be due to Transcriptformer-based models being trained on different types of prompts.

Overall, our experiments show that we can employ reasoning augmentation techniques to increase reasoning at test time and probe models for the intrinsic knowledge they have accumulated.

3.7 Examples of rbio-prompting in natural language for questions related to diseases like Alzheimer's

We offer examples of answers and reasoning traces generated by the rbio-models on a perturbation question in Figure 7 and more general questions related to disease in Figure 8. We observe that chain-of-thought prompting tends to have models offer more detail in the reasoning trace, which could potentially hint at length of responses being correlated with the improvements in metrics we found as shown in Figure 6. We offer some analysis on potential hallucinations, mentions of correct and incorrect scientific facts in the perturbation-related questions under Figure SI-4, although we want to caution against its interpretation, given that it has been performed on a small sample of N=5 model outputs. We intend to do a more thorough analysis on the qualitative aspects of the reasoning traces as future work.

Moreover, we've looked into answering questions that are completely out-of-distribution, such as questions related to Alzheimer's Disease, aging, autoimmune diseases like rheumatoid arthritis and rare diseases such as lipodystrophy, and offer some examples in Figure 8. In these cases, the qualitative evaluation shows

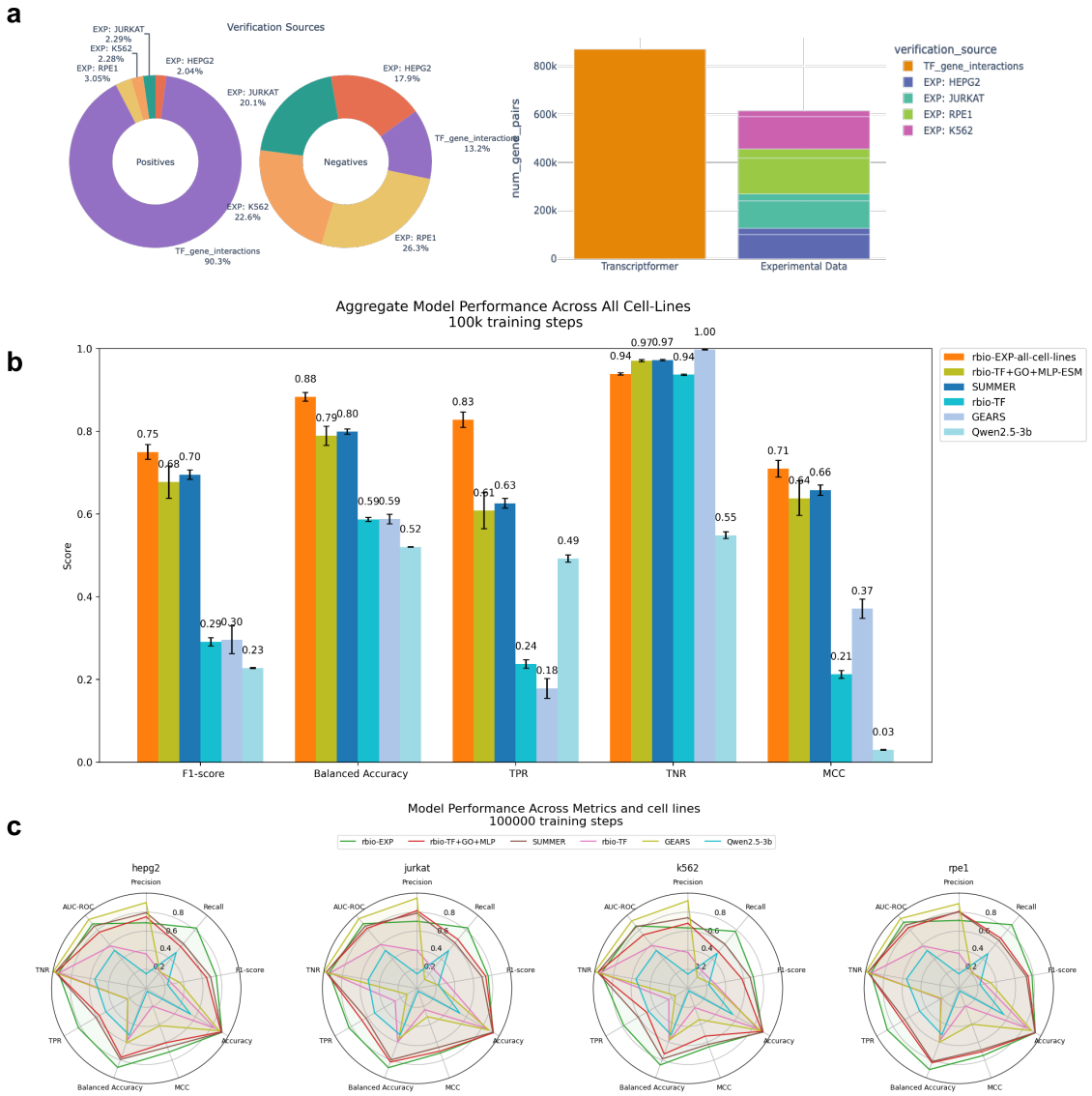


FIGURE 4. | **Analysis of model performance for models trained on composition of verifiers: (a) Data Distribution across verifiers.** Comparison of positive and negative labels from Transcriptformer and Experimental Data. Transcriptformer is able to generate more positive interactions labeled data for training compared to experimental data.**(b) Comparison of model performance for models trained with diverse combinations of soft verifiers against models trained purely on experimental data.** Metrics are computed for each cell line and aggregated over 5 different generation runs. Each bar corresponds to a mean performance aggregated across all four cell-lines: K562, RPE1, JURKAT and HEPG2. rbio-EXP-all-cell-lines corresponds to rbio trained on training splits of all four cell lines, rbio-TF+GO+MLP+ESM corresponds to rbio trained on combinations of soft verifiers: Transcriptformer gene co-expression, the GO Ontology and MLP predictions. **(c) All metrics model performance split by cell-lines**: K562, RPE1, JURKAT and HEPG2.

that models have generally good and consistent reasoning, with only few scientific inaccuracies, and tend to

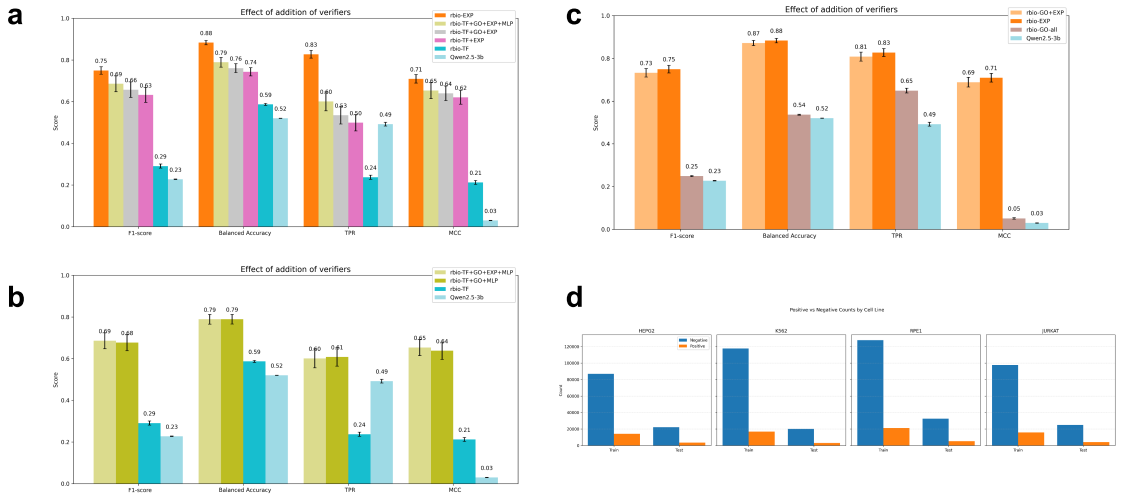


FIGURE 5. | **Analysis on composition of verification sources. a** baseline, rbio-TF, rbio-TF+EXP, rbio-TF+GO+EXP, rbio-TF+GO+EXP+MLP. Each additional verification source increases performance. **(b)** baseline, rbio-TF, rbio-TF+GO+MLP, rbio-TF+GO+MLP+EXP **(c)** baseline, rbio-GO-all, rbio-EXP, rbio-GO+EXP. **(d)**. Distribution of the four cell lines in the training and testing data. Data is imbalanced and skewed towards negative interactions.

offer reasoning that contains gene information and regulatory pathways. The application of rbio models to biological tasks outside of perturbation prediction offers exciting opportunities for future research.

4 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 gene-co-expression data on information from Transcriptformer has been obtained from Transcriptformer according to the methodology presented in the paper.

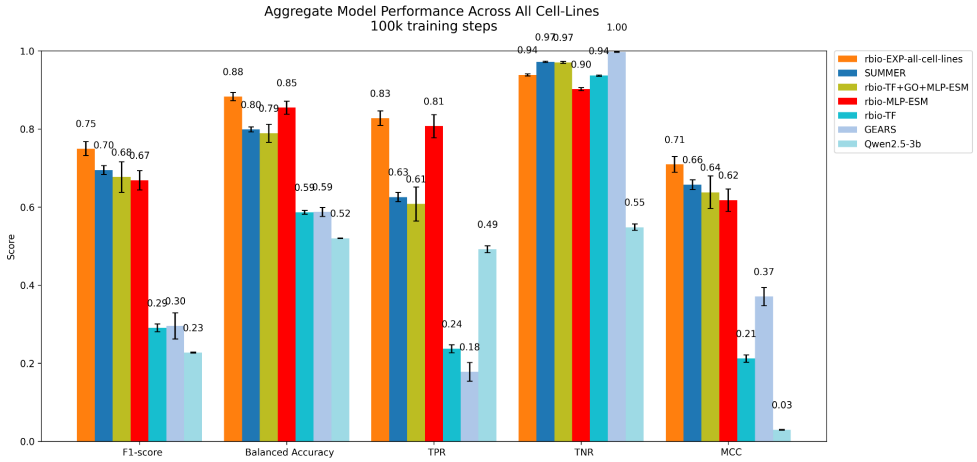
The code is publicly available at https://github.com/czi-ai/rbio, together with a training and an inference script. Model weights are publicly available at https://github.com/czi-ai/rbio. Tutorials, as well as model cards are available on the https://virtualcellmodels.cziscience.com.

5 Training and Evaluation

Models have been trained using the GRPO framework and HuggingFace interface[48]. We use a Qwen2.5-3B-Instruct model as a base model, the model weights having been accessed through HuggingFace. Each model has been trained for 100k steps, taking 10 days for completion on 8 H100 GPUs, with some variation between models. During training, we 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 [25].

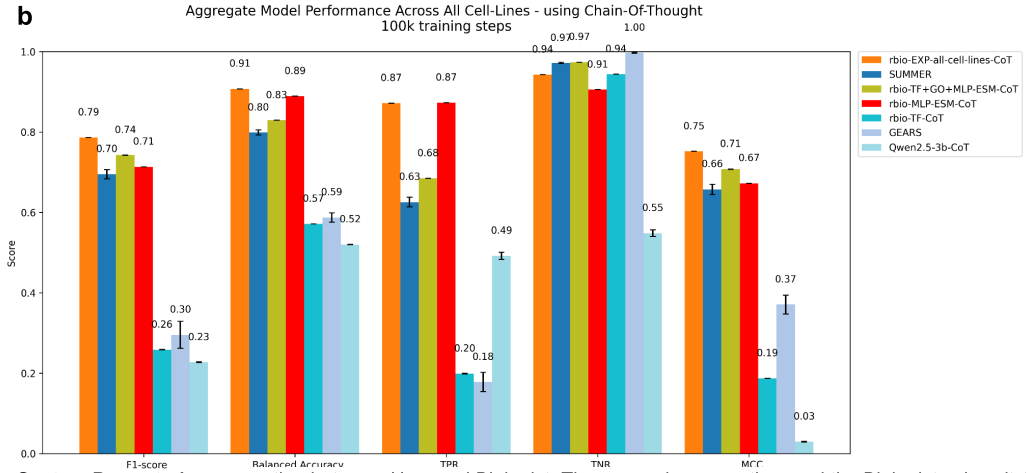
6 Related Work

Contemporary reasoning-oriented large language models (LLMs) - exemplified by OpenAI's o-series (o1, o3), Anthropic Claude 3.7 and Claude 4 Opis, Google Gemini 2.5, and DeepSeek-R1 - display emergent



System Prompt: *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>.*

b



System Prompt: *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>. The Biologist will evaluate each step of this problem, using logical reasoning and evidence from the prompt.*

FIGURE 6. | **Chain-Of-Thought Prompting Effect: (a)** Comparison of model performance for models trained with combinations of soft verifiers and experimental data using a generic system prompt. This prompt has been used during training. Metrics are computed for each cell line and aggregated over 5 different generation runs. Each bar corresponds to a mean performance aggregated across all four cell-lines: K562, RPE1, JURKAT and HEPG2. rbio-EXP-all-cell-lines corresponds to rbio trained on training splits of all four cell lines, rbio-TF+GO+MLP+ESM corresponds to rbio trained on combinations of soft verifiers: Transcriptformer gene co-expression, the GO Ontology and MLP predictions, rbio-MLP-ESM is trained on MLP predictions only, rbio-TF trained on gene co-expression data only. **(b) Same comparison, using Chain-Of-Thought prompts, asking the model to think step by step.**

capacities for multi-step inference, logical deduction, and the manipulation of abstract relationships across diverse domains. We can situate these systems within four complementary development paradigms: (i) inference-time compute scaling (e.g., chain-of-thought prompting or self-consistency voting) that boosts logical depth without retraining but at significant serving-cost; (ii) pure reinforcement learning approaches such as DeepSeek-R1-Zero, where reasoning traces emerge solely from accuracy- and format-based rewards;

Is a knockdown of ISCA2 in rpe1 cells likely to result in differential expression of CEP295? The answer is either yes or no

system_prompt
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>.

Example answer

Knockdown of ISCA2 in RPE1 cells would likely lead to differential expression of CEP295. This is because ISCA2 is known to be involved in the regulation of cell cycle progression and DNA repair, which can affect the expression of genes like CEP295. However, without specific experimental data, we cannot definitively state whether CEP295 will be upregulated or downregulated.

system_prompt
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>.The Biologist will evaluate each step of this problem, using logical reasoning and evidence from the prompt.

Example answer using Chain-of-Thought

ISCA2 is a protein that is involved in the regulation of cell cycle progression. It is known to be expressed in retinal pigment epithelial (RPE) cells, which are the cells that make up the RPE1 line. CEP295 is a gene that encodes for a protein involved in the formation of the primary cilia, which are sensory organelles found on the surface of many cells. Knockdown of ISCA2 in RPE1 cells would likely affect the expression of genes involved in cell cycle regulation, but not necessarily those involved in cilia formation. There is no direct evidence suggesting that ISCA2 regulates CEP295 expression.

system_prompt
You are an AI model trained as a Biologist through reinforcement learning. I will ask you a question, you will come up with a reasoning process based on what you have learned during training and then you will give me 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>. You will provide the reasoning step-by-step, using detailed biological knowledge from training.

Example answer using self-aware Chain-of-Thought

Knockdown of ISCA2 in RPE1 cells would likely lead to differential expression of CEP295. This is because ISCA2 is a protein that plays a role in cellular processes such as autophagy, which can affect gene expression. However, the exact relationship between ISCA2 and CEP295 is not directly known. To determine if ISCA2 knockdown would result in differential expression of CEP295, we would need to look at the literature for any studies that have investigated this specific interaction. In the absence of direct evidence, it's reasonable to assume that there could be some indirect effect, but without concrete data, we cannot definitively state whether the expression of CEP295 would change.

FIGURE 7. | **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: rbio-TF+GO+MLP.

(iii) hybrid supervised fine-tuning followed by RL, adopted by DeepSeek-R1, which first aligns on curated reasoning corpora and then refines with consistency-oriented rewards to achieve state-of-the-art open-model performance; and (iv) distillation and lightweight fine-tuning, which transfer these capabilities into smaller Qwen [24, 49] and Llama-class models [1, 25] trading peak accuracy for efficiency. While these LLMs rely on high-dimensional function approximation rather than explicit symbolic manipulation, prompting techniques such as chain-of-thought, self-consistency sampling, and tool augmentation consistently enhance performance on arithmetic, commonsense, and scientific reasoning benchmarks. Empirical evaluations

What do similarities in expression patterns between evolutionarily distant species tell us about the molecular pathogenesis underlying Alzheimer's Disease?

Similarity in expression patterns between evolutionarily distant species can provide insights into the molecular mechanisms that are conserved across different organisms. If we find that certain genes or proteins are expressed similarly in species that diverged long ago, it suggests that these genes or proteins play crucial roles in fundamental biological processes. Alzheimer's Disease is a complex neurodegenerative disorder characterized by the accumulation of amyloid-beta plaques and tau tangles. The molecular pathogenesis of Alzheimer's involves several genes and proteins, including APP (amyloid precursor protein), PSEN1, PSEN2, APOE, and tau. By studying the expression patterns of these genes and proteins in evolutionarily distant species, we can identify conserved pathways that may be involved in the disease. This could potentially lead to the discovery of new therapeutic targets or biomarkers for Alzheimer's.

Evolutionary conservation of aging mechanisms suggests that certain pathways are conserved across different species and play a role in the aging process. To slow down the aging process, one would need to target these conserved pathways. I will consider some well-known pathways involved in aging, such as the insulin/IGF-1 signaling pathway, the mTOR pathway, and the NAD+ metabolism pathway. These pathways have been shown to be conserved across various organisms and their dysregulation is associated with accelerated aging.

Based on evolutionary conservation of aging mechanisms, the insulin/IGF-1 signaling pathway, the mTOR pathway, and the NAD+ metabolism pathway should be targeted for slowing down the aging process.

Based on evolutionary conservation of aging mechanisms, which pathways should be targeted for slowing down the aging process?

Based on evolutionary conservation of gene expression patterns, which specific cell types are most important in the autoimmune response in rheumatoid arthritis and which molecular pathways are involved in the pathology?

Evolutionary conservation of gene expression patterns can be used to identify conserved genes and pathways that are likely to be important in the autoimmune response. Rheumatoid arthritis is an autoimmune disease where the immune system attacks the joints. To find out which cell types and molecular pathways are involved, I would look at studies that have identified genes and pathways that are conserved across different species and are known to be involved in immune responses. For example, the interleukin-1 receptor (IL-1R) pathway is highly conserved and plays a role in inflammation and immune responses. Studies have shown that IL-1R signaling is upregulated in rheumatoid arthritis patients, indicating its importance in the disease.

Based on evolutionary conservation of gene expression patterns, the specific cell types most important in the autoimmune response in rheumatoid arthritis include B cells, T cells, and macrophages. The molecular pathways involved in the pathology are those that regulate inflammation, such as the interleukin-1 receptor (IL-1R) pathway.

FIGURE 8. | **Example answers for non-perturbation related questions using an rbio-TF model trained on gene co-expression data.**

show that they can rival or sometimes even surpass symbolic and neuro-symbolic systems; nevertheless, open challenges persist in mitigating hallucinations, ensuring logical consistency over long contexts, controlling verbosity, and elucidating the mechanisms that yield apparently structured reasoning behaviour.

Reasoning LLMs unlock many opportunities for carrying out inference in expert domains like chemistry [50] or biology - i.e. GenePT[18]. For instance, BioReason [26] extends the recent wave of multimodal

reasoning models by tightly coupling a genomic encoder with an LLM, thereby overcoming the long-standing divide between black-box DNA foundation models and text-centric reasoning systems. The hybrid of an LLM (e.g., Qwen3 [49] and a foundation model (e.g., StripedHyena like in Evo2 [51] or Nucleotide Transformer [52]) achieves state-of-the-art performance on disease-pathway inference and variant-effect prediction while generating biologically coherent, stepwise explanations that facilitate expert validation – advantages that underscore the model's potential for hypothesis generation and precision-medicine workflows. Nevertheless, the approach inherits several limitations, e.g., the reliance on curated resources such as KEGG may bias conclusions toward well-annotated pathways. This drawback highlights the need for training reasoning LLMs to generalize outside their training distribution. Cao et al.[27] extend LLM-based bioinformatics by framing cell-type annotation as an explicit reasoning task. Their CellReasoner model combines a compact, Qwen2.5 backbone [24] with a three-stage fine-tuning curriculum that requires only 380 expert chain-of-thought exemplars, yet surpasses both heuristic algorithms (e.g., singleR) and larger general LLMs on diverse scRNA-/scATAC-seq datasets. Advantages include (i) open-source availability, (ii) lightweight deployment compatible with consumer GPUs, and (iii) marker-level explanation chains that facilitate biological validation. However, the approach inherits several limitations: performance has been demonstrated on a small set of well-curated datasets, leaving robustness to noisy or rare cell populations untested; reliance on manually selected marker genes and reasoning traces may embed human bias; and the reinforcement-learning stage increases training complexity and can induce verbose outputs that hamper high-throughput workflows. As a result, it remains uncertain whether the biological knowledge was properly transferred to the LLM.

The key difference between our models and the aforementioned ones is that we use VCMs and other biology models directly, to provide a reward signal. Other approaches have used external models as sources of knowledge by incorporating their outputs (embeddings or other types of outputs) into the reasoning traces, yet using hard ground truth from annotated datasets to assign correctness rewards. The key difference between our models and the aforementioned ones is that we use VCMs and other biology models directly, to provide a reward signal. Other approaches have used external models as sources of knowledge by incorporating their outputs (embeddings or other types of outputs) into the reasoning traces, yet using hard ground truth from annotated datasets to assign correctness rewards. This has the consequence that our approach is the first to fully show that biological world models like VCMs can be distilled into a reasoning model, effectively aligning our reasoning models to the biology model.

In other work, Wu et al.[41] propose SUMMER (SUMMarize, retrievE, answeR), an inference-time pipeline that compresses each gene's knowledge-graph neighborhood into textual summaries, retrieves analogous perturbation-gene exemplars via graph proximity, and feeds both evidence streams through chain-of-thought prompts to a compact Llama-3-8B model for prediction. While this hybrid strategy surpasses graph-based, single-cell-foundation, and naïve-LLM baselines in terms of macro-AUROC scores around 0.60-0.61 for differential expression and 0.62-0.69 for direction-of-change, the absolute gains remain modest, and the authors acknowledge that their PERTURBQA benchmark is "largely unsolved". They further document key deficiencies: the model frequently mis-assigns causal directionality, predicting upstream effects where none exist; computational constraints force the use of a 70-billion-parameter Llama-3 variant merely to pre-generate gene summaries, with inference relegated to 8 B and repeated three times to tame stochasticity; and, despite outperforming prior art, SUMMER indicates a potential of the world models of biology while signalling limited robustness and the need for richer, more principled reasoning beyond retrieval-augmented prompting. SUMMER trains on experimental data outcomes and ingests carefully curated biological information in the system prompts. We show that we are able to achieve comparable performance by training models with predictions from virtual cell models, bypassing the need for large-scale experimental data or niche domain knowledge. Moreover, by making use of Chain-of-thought prompting techniques to increase reasoning at test time, we surpass SUMMER - showing that we can train models just as powerful by tapping into rich biological knowledge from world models of biology.

Our method also shares some aspects with recent and concurrent advances in reasoning model training. For instance, in concurrent work Yu et al. [34] propose utilizing the LLM to score reference answers as a reward signal, a mechanism similar to one of our soft verifiers in **Rbio-Prior** (see Sec. 2.4) where we utilize prior knowledge and the model's own judgement of it (using likelihood or ROUGE scores). In other recent

work [33] other LLMs are used to provide soft verification of LLM responses for reasoning mirroring our other soft verification mechanism Rbio-VCM, with a crucial addition of the fact that they can be used in weighted combinations per reward to shape a strong verifier. However, the models used here are still LLMs scoring an LLM output, whereas in **Rbio-VCM** (see Sec. 2.3) we can resort to entirely arbitrary biology world models as soft verifiers so long as we can simulate the reward from the prompt, and also combine different verifiers as separate reward functions. We will leave it open for future work to combine multiple soft verifiers per reward as the authors propose.

Last but not least, the main contribution of our work is introducing simulations, rather than labeled data, as reward signals for training reasoning models. Variations of this paradigm have been introduced before in the general reinforcement learning literature, such as in Francon et al.[30]. We are, as far as we know, the first ones to apply it in the space of reasoning models for biology, which are largely trained on experimental data or rich domain knowledge.

7 Conclusion

In this paper, we propose rbio1, a suite of biological reasoning models trained using simulations coming from world models of biology as rewards- rather than hard ground truth data - with reinforcement learning through a process that we call soft verification. We show that this approach leads to models just as performant as models trained on experimental data, especially when prompted using Chain-of-thought techniques. We offer a powerful alternative to building models used solely on experimental data, by leveraging the wealth of biological knowledge that is present in the myriad of specialized bio-models available in the biomedical ecosystem. We offer a proof of concept of this approach by using simple, yet performant, models of perturbation prediction - such as MLPs trained on gene embeddings, as well as powerful models of biology - such as Transcriptformer for transcriptomics and ESM for proteomics. Moving forward, we envision extending rbio1 to incorporate multiple types of models and modalities, serving as a platform for building a truly universal virtual cell model that aggregates knowledge from powerful models across modalities and tasks.

8 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 preprint.

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] 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.

[7] 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.

[8] 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.

[9] Abhinav K Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, et al. Predicting cellular responses to perturbation across diverse contexts with state. *bioRxiv*, pages 2025–06, 2025.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] Yiqun Chen and James Zou. Genept: a simple but effective foundation model for genes and cells built from chatgpt. *bioRxiv*, pages 2023–10, 2024.

[19] 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.

[20] 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.

[21] 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.

[22] 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.
[23] Ana-Maria Istrate, Donghui Li, and Theofanis Karaletsos. scgenept: Is language all you need for modeling single-cell perturbations? *bioRxiv*, pages 2024–10, 2024.
[24] Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.
[25] 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.
[26] 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.
[27] 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.
[28] 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.
[29] Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.
[30] 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.
[31] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.
[32] 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.
[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] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.
[36] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.
[37] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.
[38] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409, 2022.
[39] Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, et al. Biomedlm: A 2.7 b parameter language model trained on biomedical text. *arXiv preprint arXiv:2403.18421*, 2024.
[40] Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, pages 1–5, 2025.

[41] 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.

[42] 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.

[43] 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.

[44] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[45] 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.

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

[47] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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

[49] 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.

[50] Eric Wang, Samuel Schmidgall, Paul F Jaeger, Fan Zhang, Rory Pilgrim, Yossi Matias, Joelle Barral, David Fleet, and Shekoofeh Azizi. Txgemma: Efficient and agentic llms for therapeutics. *arXiv preprint arXiv:2504.06196*, 2025.

[51] Garyk Brixi, Matthew G Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, et al. Genome modeling and design across all domains of life with evo 2. *BioRxiv*, pages 2025–02, 2025.

[52] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025.

Supplementary Material

A Hard-Ground Truth Experiments

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

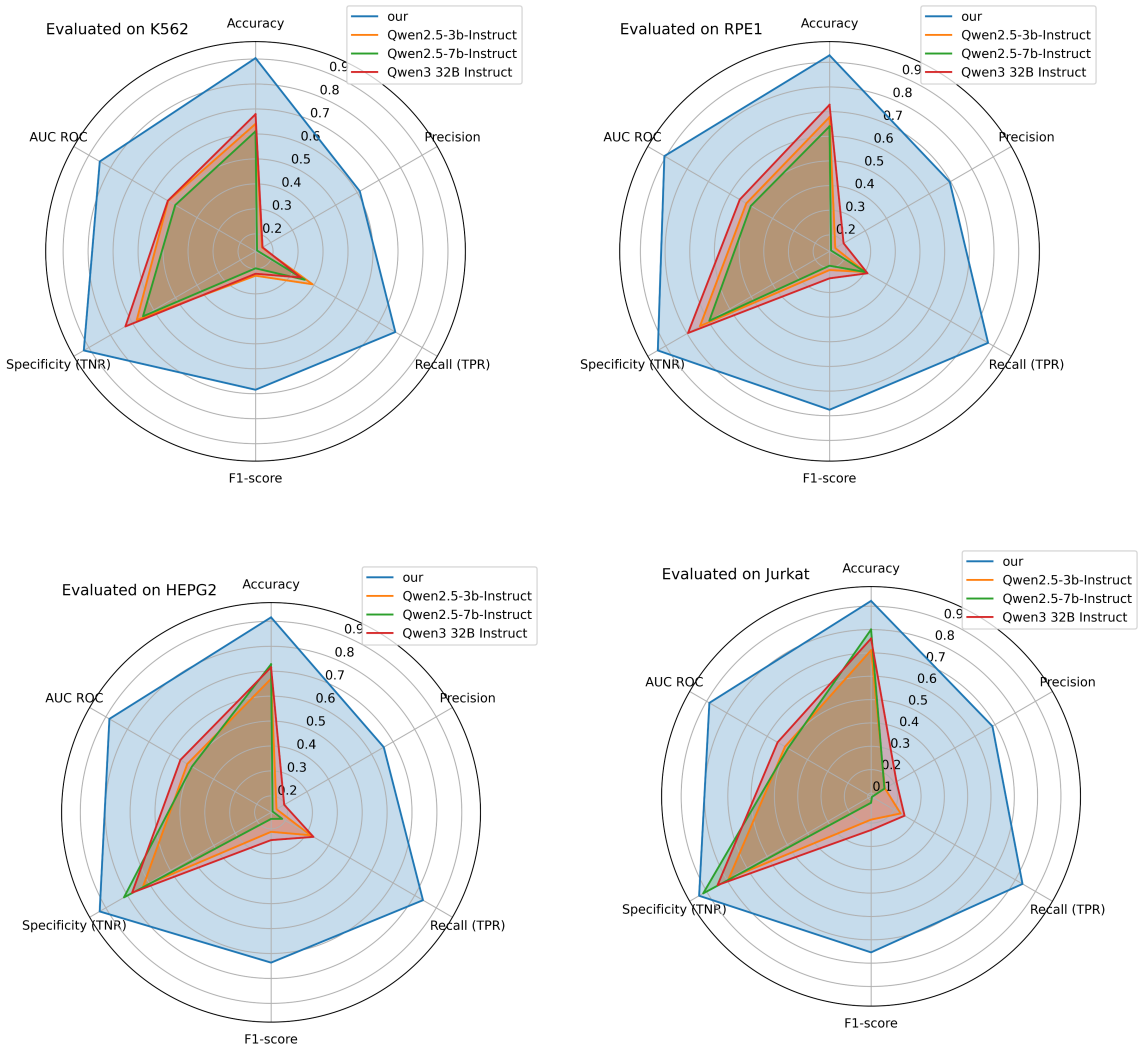


FIGURE SI-1. | **Comparison of rbio-EXP with Qwen baselines**. Our refers to rbio-EXP trained on one cell line at a time.

The composite radar plots SI-1 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.

A.2 Training on experimental data from multiple sources shows increased performance, showing there is beneficial cross-dataset knowledge transfer

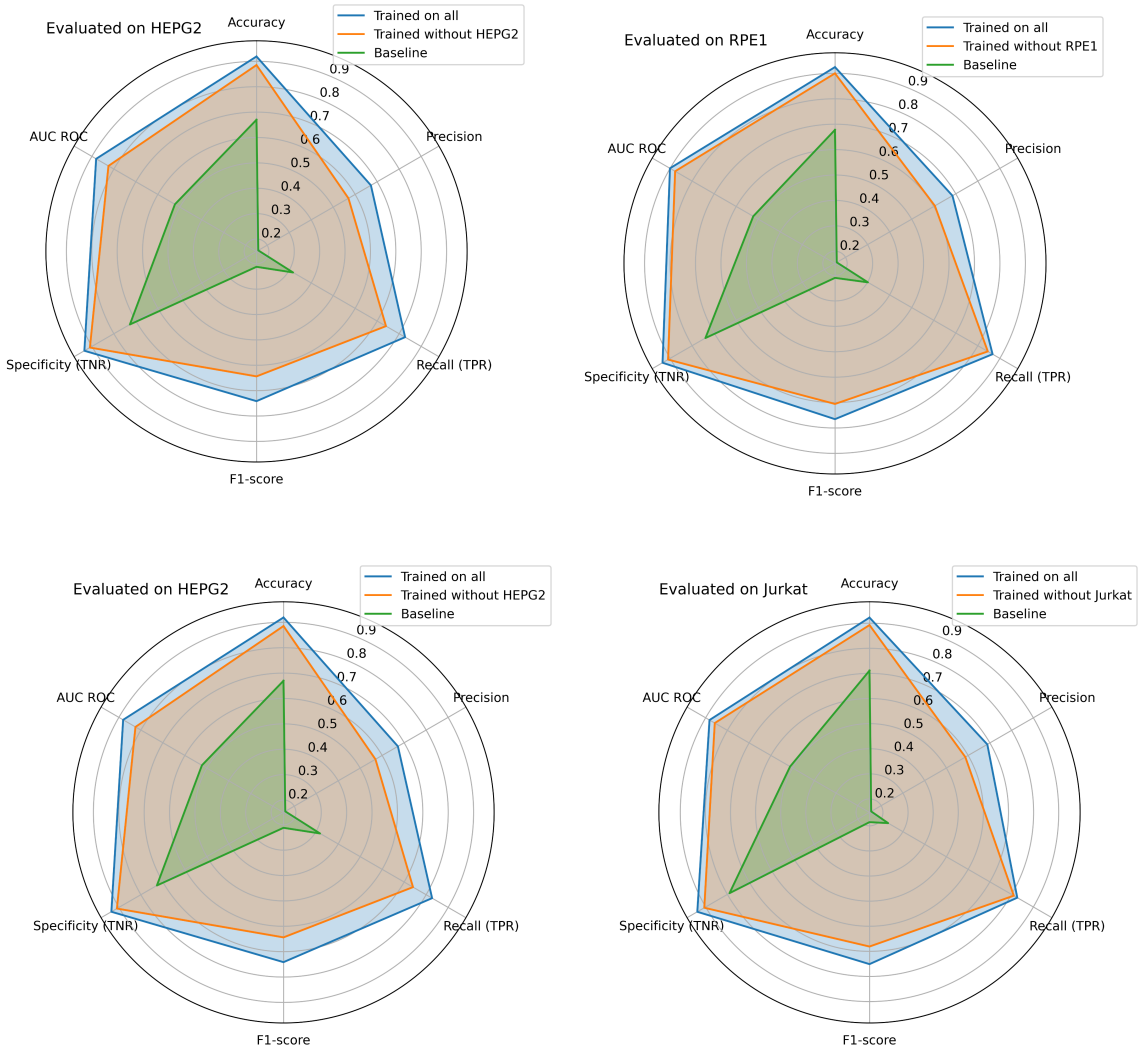


FIGURE SI-2. | **Comparison of rbio-EXP trained on training splits of all cell lines vs trained on training splits of cell lines outside of the test distribution.**. Our refers to rbio-EXP trained on one cell line at a time.

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 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.

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

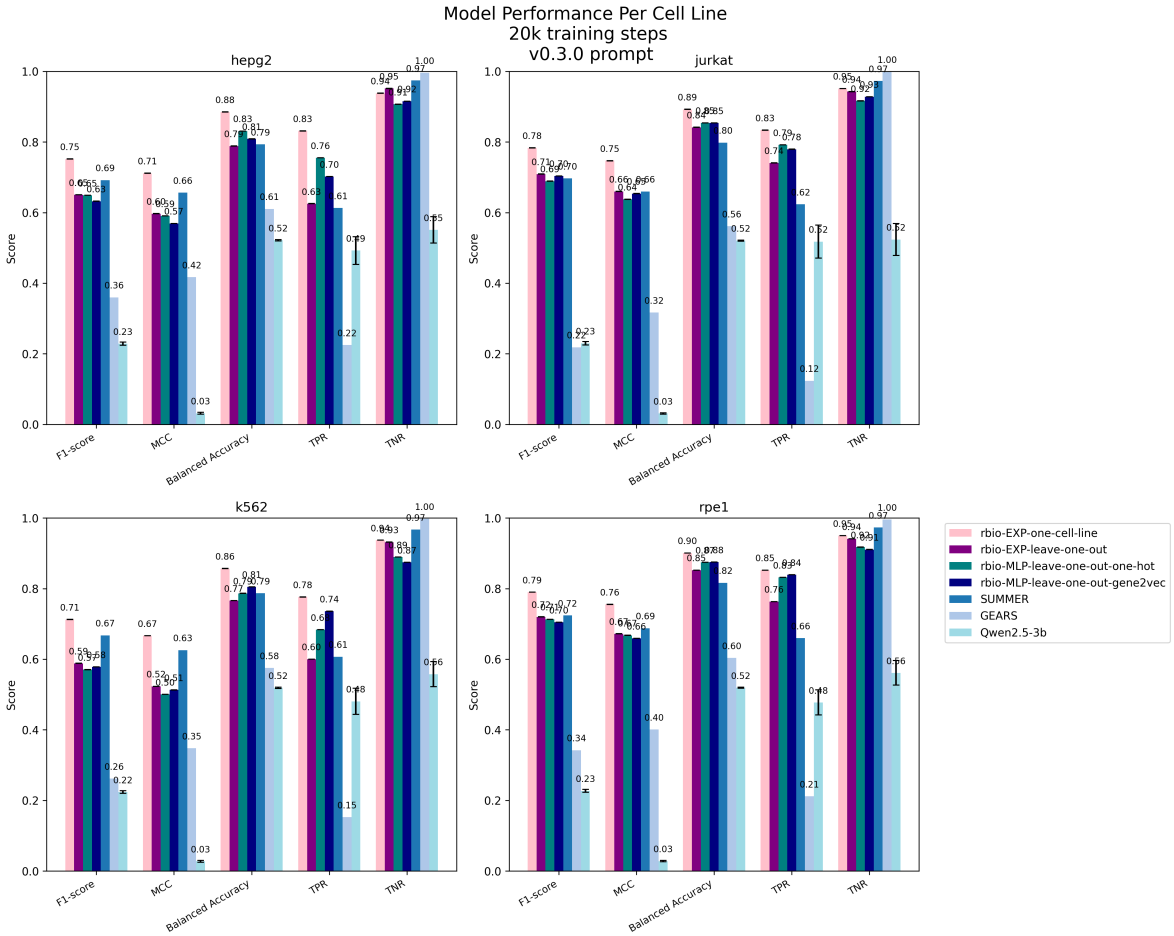


FIGURE SI-3. | **Analysis on performance on rbio models trained on experimental data for in-distribution vs out of distribution datasets**. rbio-EXP-one-cell-line is trained on one the training split from one cell line at a time and tested on the test split of that cell line (in-distribution), rbio-EXP-all-cell-lines 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)

C Qualitative Analysis for N = 5 generated perturbation prediction prompts

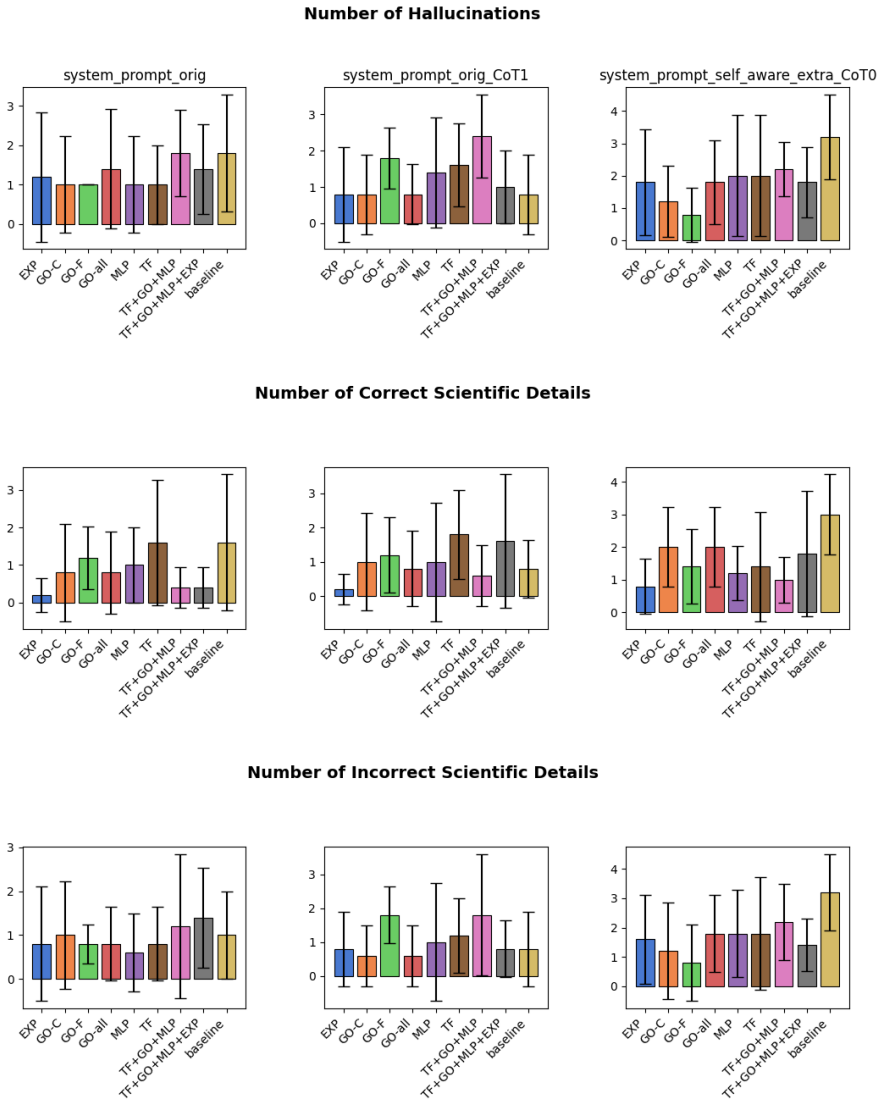


FIGURE SI-4. | **Qualitative analysis of hallucinations/number of correct and incorrect scientific facts on a small sample of N = 5 generated outputs for perturbation questions.**