How Post-Training Shapes Biological Reasoning Models

Lukas Fesser*
Harvard University
lukas_fesser@g.harvard.edu

Hanlin Zhang*
Harvard University
hanlinzhang@g.harvard.edu

Michelle M. Li
Harvard University
michelleli@g.harvard.edu

Eric Wang
Google DeepMind
ericzwang@google.com

Bryan Perozzi
Google Research
bperozzi@google.com

Shekoofeh Azizi
Google DeepMind
shekazizi@google.com

Sham M. Kakade
Harvard University
sham@seas.harvard.edu

Marinka Zitnik
Harvard University
marinka@hms.harvard.edu

Abstract

Scientific reasoning models for biology combine language models with foundation models trained on multimodal biological data, including DNA, RNA, and proteins. These models are built through post-training, yet how each stage shapes reasoning and generalization remains poorly understood. We study when post-training improves performance and when it induces over-specialization. Across genomics, transcriptomics, and proteins, we train and evaluate more than 100 biological reasoning models under controlled variation in backbone, continued pre-training (CPT), supervised fine-tuning (SFT), and reinforcement learning (RL), measuring both in-domain (ID) and out-of-domain (OOD) performance. We find that each post-training stage reshapes generalization in a distinct way rather than contributing uniform gains. CPT improves downstream performance by aligning models with biological language. SFT consistently increases ID performance but can cause OOD performance to peak early and decline as models fit the training distribution. RL, when applied to strong SFT checkpoints with aligned rewards, improves OOD performance and partially recovers generalization. These results show that biological reasoning does not improve monotonically with additional supervision or compute. Instead, performance depends on how training stages are composed. Under fixed post-training budgets, the strongest ID-OOD trade-off comes from sufficient SFT, larger RL allocations, and asymmetric adaptation capacity across stages. Code is available at https://github.com/mims-harvard/bio-posttrain and selected model checkpoints can be found here https://huggingface.co/collections/mims-harvard/bio-posttrain.

1 Introduction

Biology is becoming a central testbed for scientific reasoning models. Recent systems combine language models with biological foundation models trained on DNA, RNA, proteins, and other molecular data [1–3]. Their predictions require mapping natural-language task descriptions to molecular representations, integrating modality-specific evidence, and carrying intermediate biological state

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

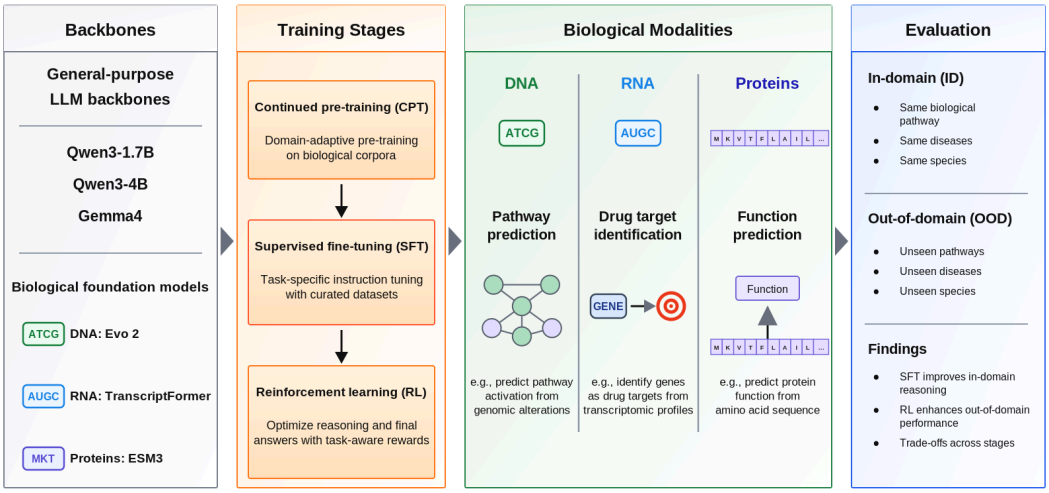

Figure 1: **Training dynamics define distinct generalization regimes in biological reasoning models.** We compare backbone choice, continued pre-training (CPT), supervised fine-tuning (SFT), and reinforcement learning (RL) across genomics, transcriptomics, and protein tasks, and evaluate each stage on biologically meaningful in-domain (ID) and out-of-domain (OOD) splits.

across multiple inference steps. Post-training is widely used to build such models, but its effects remain poorly understood across training stages [4]. Despite strong empirical gains, it remains unclear how different stages of post-training shape reasoning and generalization.

Recent work has explored new forms of supervision, scaling strategies, and training objectives, including reinforcement learning for reasoning [5–7], large-scale post-training datasets [8, 9], and domain-specific adaptation pipelines [10, 11]. Other studies examine how reward design [12, 13], self-improving and world-model-based approaches [14–16], and training dynamics [17–19] influence model behavior. While these approaches improve task performance, they provide limited insight into how individual post-training stages affect generalization.

Biology provides a particularly stringent test of generalization. In mathematics and code, many out-of-domain problems retain the same underlying structure as the training examples, even when surface details change. In biology, unseen pathways, diseases, species, and perturbations often involve different mechanisms and biological processes [20, 21]. As a result, high benchmark performance does not necessarily indicate robust biological reasoning [22–24]. Models that perform well on familiar benchmarks may fail when transferred to new biological systems [25, 26]. Additional post-training or larger models can therefore increase in-domain performance without improving biological generalization.

When additional post-training improves biological generalization, rather than primarily increasing fit to the training distribution, remains unclear. Existing studies typically examine one modality, one benchmark family, or one post-training stage at a time. Post-training itself is a sequence of stages, including continued pre-training, supervised fine-tuning, and reinforcement learning [27–33]. These stages may interact in non-obvious ways, and gains from one stage may depend on the stages that precede or follow it. Yet existing models often differ simultaneously in backbone, data, scale, and supervision, making controlled comparisons difficult. Most studies also focus on final performance rather than training dynamics, and out-of-domain evaluation in biology remains limited and inconsistently defined.

Present work. We present a controlled study of post-training in biological reasoning models. Across genomics, transcriptomics, and proteins, we train and evaluate more than 100 biological reasoning models to examine when post-training improves biological generalization and when it primarily increases fit to the training distribution. Using matched model families, tasks, and data settings, we isolate the effects of backbone choice, continued pre-training (CPT), supervised fine-tuning (SFT), and reinforcement learning (RL) on both in-domain (ID) and out-of-domain (OOD) performance. We find that the same post-training budget can produce different generalization regimes depending on how it is allocated across stages: SFT increases ID performance but can narrow OOD robustness, RL strengthens OOD performance when initialized from strong SFT checkpoints, and CPT improves downstream adaptation.

Our contributions are conceptual, empirical, and practical. Conceptually, we show that biological reasoning does not improve monotonically with additional post-training. Empirically, we present a controlled study across genomics, transcriptomics, and proteins that reveals consistent training dynamics across biological modalities. Practically, we derive design principles for post-training under limited compute. The strongest ID-OOD trade-offs generally come from combining sufficient SFT with larger RL allocations and allocating adaptation capacity asymmetrically across stages.

2 Background and Related Work

Biological Foundation Models and Multimodal Reasoning. Recent work has extended foundation modeling to a wide range of biological data [34], including DNA [35–42], RNA [43, 44], gene expression [45, 10, 46–50], and proteins [51–54]. Much of this literature focuses on learning representations or solving predictive tasks within a single modality, for example by modeling sequences or cellular profiles directly. More recent systems combine language models with biological inputs and structured context [2, 1, 23, 55], enabling tasks that more closely resemble reasoning over pathways, cell states, or protein function. Despite this progress, the literature remains fragmented. Models differ substantially in architecture, modality, training data, supervision format, and evaluation protocol, making it difficult to compare results or isolate which components drive performance. While recent biological foundation models provide a basis for multimodal reasoning, the field still lacks a unified view of how training choices affect downstream behavior, particularly under distribution shift.

Training Dynamics in LLMs: Training Stages Are Not Uniformly Additive. Modern language models are developed through a sequence of training stages rather than optimized in a single pass. A base model is first selected or pretrained, then adapted through domain-specific continued pre-training, task-specific supervised fine-tuning, and, in some cases, reinforcement learning [56]. These stages serve distinct roles. Continued pre-training expands coverage of domain vocabulary and structure while preserving general capabilities [57]. Supervised fine-tuning specializes the model to task formats and target behaviors, often yielding large gains on benchmark tasks. Reinforcement learning further aligns outputs with task-dependent reward signals, improving performance but increasing sensitivity to reward design and sampling [58].

Recent work suggests that these stages are not uniformly additive [59–61]. Performance gains can saturate or reverse as additional training is applied, and improvements on targeted evaluations need not translate to better generalization. PYTHIA [62] enables fine-grained analysis of training dynamics across checkpoints and scales, while EVOLM [4] characterizes non-monotonic behavior across training stages. Fine-tuning scaling laws further show that the interaction between model size, pretraining data, and supervision depends on the adaptation method [63]. A growing body of work has studied how to conduct continued pre-training effectively: Gupta et al. [64] investigate learning rate re-warming schedules, Ibrahim et al. [65] propose scalable data mixing and replay strategies, Parmar et al. [66] characterize when mid-training transfers general capabilities versus degrading them, and Ke et al. [67] introduce soft-masking to mitigate catastrophic forgetting. Cheng et al. [68] further show that reformatting domain corpora as reading comprehension during continued pre-training yields stronger downstream performance than raw text exposure. Together with recent large-scale systems such as COMPOSER 2 [69], these studies indicate that continued pre-training often serves as a critical transition stage enabling effective downstream adaptation [70], while supervised fine-tuning and reinforcement learning can produce strong task gains at the cost of reduced robustness if scaled or applied incautiously [71, 72].

3 Experimental Setup: Tasks, Data, and Training Stages

We design the experimental setup to isolate post-training effects while keeping the biological tasks, model families, and evaluation splits comparable across modalities. We first define the reasoning tasks and ID/OOD splits, then describe the model architecture and post-training pipeline.

3.1 Biological Reasoning Tasks and Evaluation Splits

We evaluate post-training across three domains: DNA, RNA, and proteins. Each task combines natural-language context with modality-specific biological inputs and uses ID and OOD splits. Dataset details and example prompts are provided in the appendix A.

Pathway Prediction. Pathway prediction asks the model to infer how a genetic variant propagates through a molecular pathway to produce a disease phenotype. We use the KEGG-derived benchmark introduced in BIOREASON [1], which evaluates mechanistic reasoning over pathway structure rather than variant classification alone. Each example combines a reference DNA sequence, a variant DNA sequence, and textual pathway and gene context; the model generates a natural-language answer grounded in both sequence and pathway information. We define ID and OOD splits by pathway network, so OOD examples come from previously unseen molecular networks.

Drug Target Identification. Target identification asks the model to choose the most promising therapeutic target for a disease and cell type. We adapt the cell-type-specific target nomination benchmark from MEDEA [73], simplifying it from an agentic tool-use setting to a fixed-input reasoning task. The model receives a natural-language description of the disease, cell type, and candidate genes, together with TranscriptFormer embeddings [45] for five candidates in normal and disease states, and identifies the best-supported target. We use four diseases for training and ID evaluation, and reserve hepatoblastoma as the OOD disease, except in Figure 6, where we used Sjogren's disease instead.

Protein Function Prediction. Protein function prediction asks the model to infer the function of an uncharacterized protein from multimodal evidence. We build on the curated UniProt-based dataset introduced in BIOREASON-PRO [3], which pairs experimentally supported GO annotations with protein-level context. Each example provides protein embeddings and text context, including organism, InterPro domain annotations [74], and protein–protein interactions. The model predicts protein function from this combined representation. We split data by ontology graph region (in the main text) or by species (in Appendix D.3), with two held-out species forming the OOD test set. We always remove the ontology graph inputs used in the original BIOREASON-PRO setup to align the task format with our study.

3.2 Language Models, Biological Foundation Models, and Post-Training Pipeline

We study biological reasoning through a common post-training pipeline built on general-purpose LLM backbones. Our main experiments use Qwen3-1.7B and Qwen3-4B, two dense models from the Qwen3 family, which support both reasoning-oriented and standard inference modes [75]. Figure 6 also includes an ablation using the larger Qwen3-8B model. We further include Gemma 4 E2B as a backbone ablation to test whether the observed training dynamics persist under a different lightweight open model family [76].

To represent biological modalities, we couple the LLM backbone to frozen biological foundation models through trainable projection layers. For DNA tasks, we use Evo2-1B, a genome foundation model trained for genome-scale sequence modeling and design across domains of life [38]. For RNA and transcriptomic tasks, we use TranscriptFormer, a cross-species single-cell model trained over evolutionary-scale transcriptomic data [45]. For protein tasks, we use ESM-3, a protein language model trained to model sequence, structure, and function across evolutionary scales [52].

Post-training pipeline consists of three stages: (1) **Continued pre-training:** adaptation on general biological text using the standard next-token prediction loss. (2) **Supervised fine-tuning:** training on task-specific reasoning examples with the autoregressive language-model loss on target responses. (3) **Reinforcement learning:** optimization from a supervised checkpoint with a task-aligned reward objective, encouraging outputs that maximize task success rather than imitate reference traces. To enable controlled comparisons, we vary one factor at a time. In the main experiments, we scale SFT and RL compute, study the effect of CPT, and include ablations on backbone choice and LoRA rank. Full hyperparameters and implementation details are provided in the appendix.

We use a model signature to denote each configuration across training stages and biological domains. For example, Q1-P-C-$S_{8,20}$-$R_{16,20}$ represents a model with the following setup:

- Q1-P: Qwen3-1.7B backbone evaluated in the protein setting. We use `D`, `R`, and `P` to denote DNA, RNA, and protein tasks, respectively. In our notation, blue denotes Qwen3-1.7B models, orange denotes Qwen3-4B models, purple denotes Qwen3-8B models, and green denotes Gemma 4 E2B models.
- C: Continued pre-training on general biological text, specifically, the biology domain from FineFineWeb [77].
- $S_{8,20}$: Eight epochs of supervised fine-tuning on 20,000 task-specific reasoning traces.

- $R_{16,20}$: Reinforcement learning for 16 epochs on 20,000 data points.

When a stage is omitted, the model has not undergone that part of the pipeline. For example, Q4-R-$S_{4,1}$ denotes a model fine-tuned directly from the Qwen3-4B + TranscriptFormer backbone for 4 epochs on 1,000 reasoning traces in the RNA setting without CPT or RL, while G-R-C-$S_{8,1}$ denotes a Gemma + TranscriptFormer model adapted with CPT and then fine-tuned for 8 epochs.

4 Results: How Training Stages Shape Biological Reasoning in LLMs

We now present our main results related to scaling post-training for biology in compute- or data-bound settings. Concrete model settings for all our experiments in this section, including context windows, input sequence lengths, and other hyperparameters can be found in the appendix.

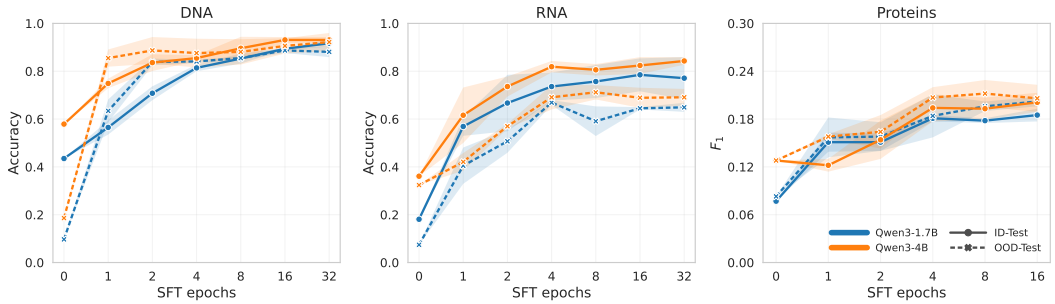

Figure 2: **Supervised fine-tuning improves in-domain performance but can reduce out-of-domain robustness.** As SFT compute increases, ID performance continues to improve, while OOD performance peaks earlier and declines, indicating over-specialization to the training data. We report mean and std. over 3 random seeds.

4.1 Supervised Fine-Tuning Increases Accuracy but Narrows Generalization

We begin by studying how supervised fine-tuning scales in biological reasoning models using pretrained Qwen3-1.7B and Qwen3-4B backbones [75].

Fixed data, variable compute. We first consider a data-constrained regime based on the DNA and RNA tasks. For each backbone, we train model families of the form Q1-D-$S_{\{1,2,4,8,16,32\},1}$, Q4-D-$S_{\{1,2,4,8,16,32\},1}$, Q1-R-$S_{\{1,2,4,8,16,32\},1}$, and Q4-R-$S_{\{1,2,4,8,16,32\},1}$, where the subscript indicates the number of SFT epochs and the use of the full available training set in each domain. We then evaluate both in-domain and out-of-domain performance. Figure 2 reveals a generalization trade-off induced by supervised fine-tuning. The amount of training that maximizes ID performance is in most settings larger than the amount that maximizes OOD performance, indicating that continued fine-tuning improves fit to the training distribution after OOD generalization has already peaked. In DNA, for example, Q1-D-S improves its ID accuracy from roughly 0.57 at 1 epoch to about 0.92 by 32 epochs, while its OOD accuracy peaks earlier, around 0.89 at 16 epochs, and then declines to about 0.87 by 32 epochs.

The RNA setting shows this trend most clearly. Q1-R-S gains nearly 0.22 ID accuracy points from 1 to 16 epochs, but while its OOD performance also improves, it comes within at most 0.07 of ID accuracy. For Q4-R-S, ID accuracy rises from roughly 0.62 at 1 epoch to about 0.84 by 32 epochs, whereas OOD accuracy peaks much earlier, around 8 epochs, and then drifts downward by the end of training. This same pattern persists on the proteins task: Q1-P-$S_{\{1,2,4,8,16\},1}$ and Q4-P-$S_{\{1,2,4,8,16\},1}$ both continue to improve ID F1 through all 16 epochs, but OOD F1 peaks earlier and then declines. More SFT compute therefore improves in-domain accuracy more reliably than it improves biological generalization.

Fixed compute, variable data. We next study a compute-constrained regime in proteins by fixing training to a single SFT epoch and varying the amount of supervision. Concretely, we train Q1-P-$S_{1,\{1,2.5,5,10,20\}}$ and Q4-P-$S_{1,\{1,2.5,5,10,20\}}$. Figure 3 shows that increasing data results in better ID and OOD F1 than increasing epochs on a fixed dataset. For Q1-P-S, scaling from 1.25K to 20K training examples increases ID F1 from 0.2 to 0.33 and OOD F1 from 0.22 to 0.33. For Q4-P-S, the corresponding gains are from about 0.25 to 0.37 on ID and from about 0.27 to just above 0.40 on

OOD. While these gains are significant, they flatten quickly: after roughly 5K samples or 25% of the data, both curves improve only marginally, and the larger 4B model even drops in ID and OOD performance.

These results suggest that SFT is a strong driver of in-domain biological reasoning, but that scaling it naively, either through more epochs or more data, does not reliably translate into better (OOD) performance. Instead, the dominant pattern is over-specialization: the model becomes better at the benchmark distribution while becoming less robust to biological shift. At the same time, the protein results indicate that these two ways of scaling compute are not equivalent. Increasing epochs on a fixed dataset produces the sharper trade-off, with OOD performance peaking early and then declining as the model repeatedly fits the same supervision. Increasing data while holding epochs fixed produces a similar pattern, but with better overall performance. Both ID and OOD performance still show diminishing returns. Under a fixed compute budget, data scaling is therefore the more robust strategy when additional supervision is available.

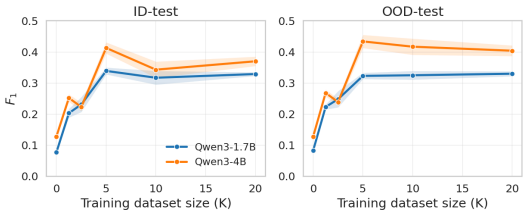

Figure 3: **Increasing data improves performance more than increasing SFT epochs.** Scaling dataset size yields gains in both ID and OOD performance, but with diminishing returns, in contrast to the overfitting behavior observed when scaling epochs.

Finding 1. Supervised fine-tuning is a strong driver of in-domain biological reasoning, but scaling it further tends to over-specialize models and can hurt out-of-domain robustness.

4.2 Reinforcement Learning Recovers Generalization After Fine-Tuning

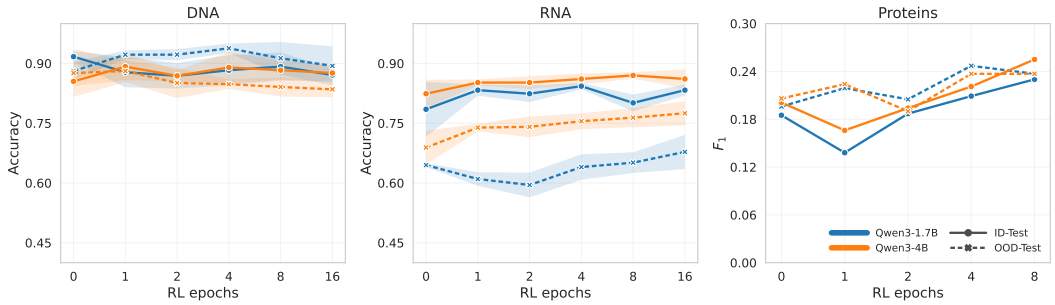

Figure 4: **Reinforcement learning consistently improves out-of-domain robustness.** Starting from strong SFT checkpoints, RL increases both ID and OOD performance, with the largest gains in OOD and diminishing returns after the first few epochs in most settings. DNA/ RNA mean and std. over 3 random seeds.

Scaling RL epochs. We now ask whether reinforcement learning can recover some of the robustness lost under SFT. Starting from the strongest SFT checkpoints selected on validation loss, we train model families of the form Q1-D-$S_{8,1}$-$R_{\{1,2,4,8,16\},1}$, Q4-D-$S_{4,1}$-$R_{\{1,2,4,8,16\},1}$, Q1-R-$S_{4,1}$-$R_{\{1,2,4,8,16\},1}$, and Q4-R-$S_{8,1}$-$R_{\{1,2,4,8,16\},1}$ in the DNA and RNA settings, together with Q1-P-$S_{4,1}$-$R_{\{1,2,4,8\},1}$ and Q4-P-$S_{4,1}$-$R_{\{1,2,4,8\},1}$ in the protein setting. Figure 4 shows that, unlike SFT, RL generally improves both ID and OOD performance quite consistently over the range we study, especially on the RNA and proteins tasks. The gains are not only directional but substantial: in DNA, OOD accuracy rises by about 0.06 over the RL sweep, and in proteins the OOD improvement is similarly large, especially for Q1-P-$S_{4,1}$-$R_{\{1,2,4,8\},1}$, which gains roughly 0.05 absolute F_1 from the SFT initialization to the best RL checkpoint. Across tasks, the largest gains typically appear in the first few RL epochs, with later epochs yielding smaller additional improvements, especially OOD. Q4-D-$S_{4,1}$-$R_{\{1,2,4,8,16\},1}$ is the main exception to this trend: here, peak OOD performance is attained after a single RL epoch, after which accuracy falls monotonically, while ID performance remains stable throughout.

Finding 2. Scaling RL from a strong SFT checkpoint yields steady improvements in both ID and OOD performance. The largest gains occur out-of-domain, and most of the improvement often appears within the first few RL epochs.

4.3 Continued Pre-Training Establishes the Foundation for Biological Reasoning

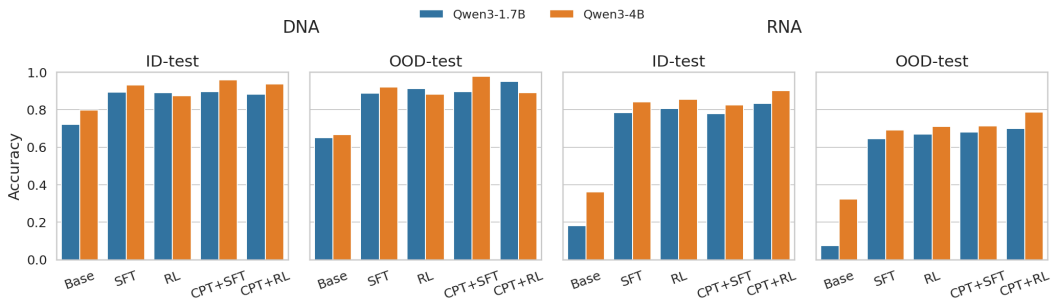

Figure 5: **Continued pre-training improves the effectiveness of downstream post-training.** CPT improves both SFT and RL performance, with the largest gains often appearing after RL and in out-of-domain settings.

We next study whether continued pre-training changes how much downstream post-training can help. In the DNA and RNA settings, we first adapt the base backbones with continued pre-training on biological texts, yielding model families of the form Q1-D/R-C and Q4-D/R-C. We then evaluate these models under the strongest post-training configurations identified above, namely Q1-D-C-$S_{32,1}$, Q4-D-C-$S_{16,1}$, Q1-D-C-$S_{8,1}$-$R_{4,1}$, and Q4-D-C-$S_{8,1}$-$R_{16,1}$ for DNA and Q1-R-C-$S_{16,1}$, Q1-R-C-$S_{16,1}$-$R_{4,1}$, Q4-R-C-$S_{32,1}$, and Q4-R-C-$S_{32,1}$-$R_{8,1}$ for RNA. Figure 5 shows that CPT improves downstream performance at most stages we test, but that the size of the gain depends strongly on the stage. The gains are generally modest at the SFT stage in domain and markedly larger after RL, especially out-of-domain. Here, CPT lifts ID and OOD performance by visibly larger margins than SFT alone.

This effect is especially pronounced for the smaller Q1-D model out-of-domain, where CPT improves the effectiveness of SFT followed by RL by 0.06. For Q4-D, the same pattern holds, but from a stronger starting point and with gains more visible with CPT and SFT only. These findings qualitatively also hold in the RNA setting and suggest that especially for the relatively small models we consider, CPT can act as a bridge between a general-purpose backbone and the reasoning demands of biology. Without CPT, downstream training must learn biological language, task structure, and reasoning behavior at the same time.

Finding 3. Continued pre-training, even on general biological texts, often improves the effectiveness of both SFT and RL, with the strongest gains appearing OOD when post-training is applied after the model has first adapted to biological language.

4.4 Backbone Strength Shifts Performance Ceiling but Not Training Dynamics

To further test whether our main findings depend on the choice of base model, we repeat the RNA experiments with an off-the-shelf backbone Gemma model [76]. In addition to Q1-R and Q4-R, we evaluate the more recent Gemma4-E2B backbone, denoted G-R, and the larger Qwen3-8B model, denoted Q8-R. Concretely, for SFT scaling we train model families of the form Q1-R-$S_{\{1,2,4,8,16\},1}$, Q4-R-$S_{\{1,2,4,8,16\},1}$, G-R-$S_{\{1,2,4,8,16\},1}$, and Q8-R-$S_{\{1,2,4,8,16\},1}$. For RL scaling, we then start from the strongest SFT checkpoint (as measured by validation loss) for each backbone and train Q1-R-$S_{4,1}$-$R_{\{1,2,4,8\},1}$, Q4-R-$S_{8,1}$-$R_{\{1,2,4,8\},1}$, G-R-$S_{4,1}$-$R_{\{1,2,4,8\},1}$, and Q8-R-$S_{16,1}$-$R_{\{1,2,4,8\},1}$. With SFT only, G-R trails the smaller Q1-R models OOD, but keeps up in-domain, as Figure 6 shows.

After one epoch, G-R is still comparable to Q4-R and outperforms Q1-R by around 0.08 in-domain. After that, more SFT helps the Qwen models more, at least out-of-domain, but G-R accuracy still increases in-domain until epoch 8 and out-of-domain until epoch 4. The qualitative trend is therefore unchanged: supervised fine-tuning is most effective for in-domain performance: its out-of-domain

gains are generally smaller and performance peaks earlier and then plateaus or even declines with additional epochs. Backbone quality seems to increase early SFT gains especially, but smaller models can eventually catch up and the core ID-OOD gap induced by SFT persists across models.

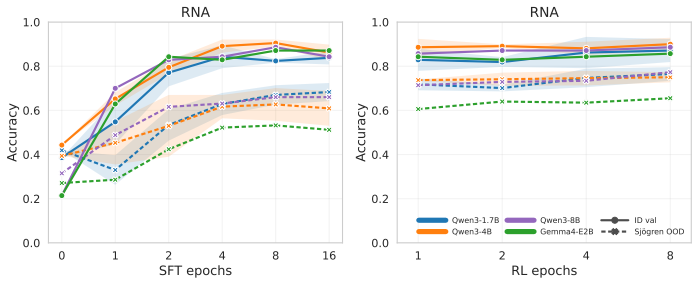

Figure 6: **Stronger backbones improve performance achievable with post-training but preserve training dynamics.** Q-R displays an initial jump in performance when starting RL, unlike Q1-R and generally performs slightly better ID and OOD. Mean and std. over 3 random seeds for the 1.7B and 4B models, single runs for Gemma and the 8B model.

Reinforcement learning exhibits a similar pattern. Starting from the strongest SFT checkpoint for each backbone, Figure 6 shows that RL improves OOD performance more reliably than additional SFT, and that the larger backbones benefit somewhat more visibly from this stage. In particular, G-R-$S_{4,1}$-$R_{\{1,2,4,8\},1}$ qualitatively follows the larger Q4-R-$S_{8,1}$-$R_{\{1,2,4,8\},1}$ trajectory more closely than the smaller Q1-R-$S_{4,1}$-$R_{\{1,2,4,8\},1}$ model. Both G-R and Q4-R improve more clearly out-of-domain after the first RL epoch and continue to gain almost monotonically. Gemma improves by 0.05 OOD between RL epochs 1 and 8, and by an impressive 0.14 compared to the SFT starting point. It remains below all three Qwen3 models throughout training, at least OOD. The larger Qwen3-8B model improves by 0.06 between RL epochs 1 and 8, and by 0.11 compared to its last SFT. This seems to indicate that backbone choice does not qualitatively alter the role of the training stage itself. The backbone matters primarily for the level of performance achievable early on with post-training, whereas the structure of the training dynamics appears to be stable across model families.

Finding 4. Larger models/ stronger backbones can raise performance, especially early on in training, but they do not change the basic training dynamics, including the SFT-induced ID-OOD performance gap.

4.5 Adaptation Capacity Should Be Allocated Stage-specifically

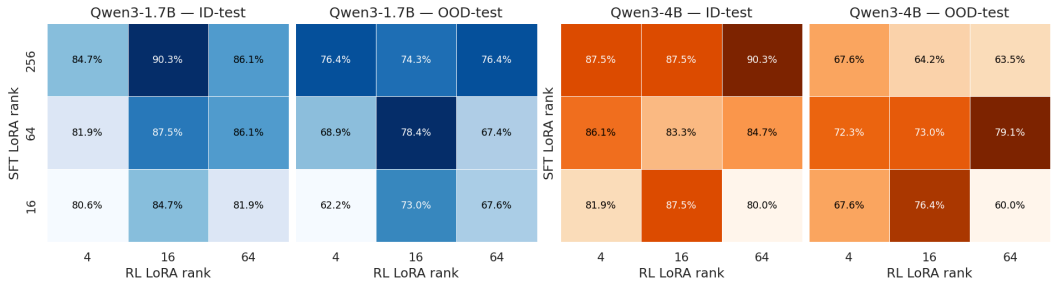

Figure 7: **Optimal adaptation often requires asymmetric capacity across training stages.** Higher LoRA rank benefits SFT, while lower rank is sufficient for RL, indicating that different stages require different adaptation capacity. Shown are results for drug target identification (RNA) tasks.

We further study how adaptation capacity should be allocated across post-training stages by running a joint SFT–RL LoRA ablation in the RNA setting. Using Q1-R-$S_{4,1}$-$R_{8,1}$ and Q4-R-$S_{8,1}$-$R_{8,1}$ as our reference model families, we vary the SFT LoRA rank over $r_{\mathrm{SFT}} \in \{16, 64, 256\}$ and the RL LoRA rank over $r_{\mathrm{RL}} \in \{4, 16, 64\}$, with the corresponding scaling factors set proportionally to rank. For each backbone, every model is first fine-tuned with SFT using the same training data, optimizer, and epoch budget as in our main RNA experiments, and is then further optimized with RL using the same reward and training schedule. We evaluate the final checkpoint from each configuration on both ID-test and OOD-test splits, and summarize the results as heatmaps over $(r_{\mathrm{SFT}}, r_{\mathrm{RL}})$. This setup

isolates whether the best end-to-end pipeline prefers symmetric adapter budgets across stages or an asymmetric allocation in which SFT and RL use different amounts of trainable capacity.

In both in-domain settings and OOD with the 1.7B model, We find a clear asymmetry between the two stages. Increasing the SFT rank from 16 to 64 or even 256 produces a clear upward shift in ID performance and usually also improves OOD performance, whereas increasing the RL rank beyond 16 yields much smaller gains and can even reduce performance. The highest ID regions in Figure 7 cluster at $r_{\mathrm{SFT}} = 256$, showing that SFT benefits from having enough capacity to absorb task format, domain structure, and multimodal reasoning patterns. By contrast, for RL, the strongest ID and OOD regions are concentrated at $r_{\mathrm{RL}} = 16$ for the smaller 1.7B model and at $r_{\mathrm{RL}} = 64$ for the larger 4B model. The best overall configurations are therefore those where the SFT rank is at least as large as the RL rank, and are often those with **high-capacity SFT and low-capacity RL**. This pattern holds for both Q1-R and Q4-R, suggesting that post-training in biological reasoning benefits from being stage-specific not only in compute and data allocation, but also in adaptation capacity.

Finding 5. In most cases, the best post-training pipelines use asymmetric adaptation capacity, with LoRA ranks for SFT at least as large as those for RL.

4.6 Optimal Post-Training Requires Balancing SFT and RL

Finally, we study how to allocate post-training across supervised fine-tuning and reinforcement learning. In this experiment, we evaluate model families of the form Q1-R-S$_{s,1.5}$-R$_{8-s,1.5}$ and G-R-S$_{s,1.5}$-R$_{8-s,1.5}$, where the total post-training schedule is fixed at eight epoch-level passes, and only the split between SFT and RL is varied. The right panel repeats this allocation after eight initial SFT epochs, giving sixteen total epochs. This is not a strictly FLOP-matched comparison: an RL epoch is more expensive than an SFT epoch because GRPO uses multiple autoregressive rollouts, reward computation, and KL anchoring. We therefore interpret this setup as an epoch-budget allocation study that compares SFT–RL allocations under a common pass-count constraint within each panel, rather than as an exact compute-normalized optimum.

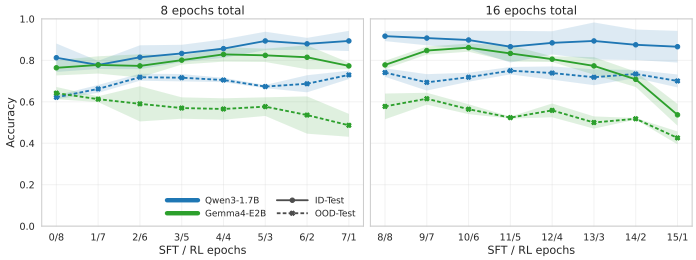

Figure 8: **Under a fixed post-training budget, enough SFT for task competence followed by RL gives a strong ID-OOD trade-off.** On RNA, the preferred allocation varies by model when starting from the base model, while after an eight-epoch SFT warm start, 1–3 additional SFT epochs followed by larger RL budgets give the strongest OOD accuracy.

Figure 8 shows that the best allocation is not at either extreme after a supervised warm start. In the right panel, G-R and Q1-R achieve their highest OOD accuracy with one and three additional SFT epochs, respectively, reaching approximately 0.62 and 0.75. Allocating all eight additional epochs to RL gives slightly lower OOD accuracy than these mixed schedules. SFT-heavy allocations preserve relatively high ID accuracy for Q1-R, but substantially reduce both ID and OOD performance for G-R. The Q1-R curves also show stronger ID retention than G-R across several larger-SFT allocations, but this does not remove the OOD decline as SFT dominates the budget. The left panel shows more model-dependent behavior: Q1-R maintains higher ID accuracy across most allocations, whereas its highest OOD accuracy occurs at $7/1$, around 0.73, and G-R achieves its highest OOD accuracy with pure RL, around 0.64. Overall, after the supervised warm start, the strongest OOD results concentrate in the one-to-three additional SFT epoch range, followed by five to seven RL epochs. The DNA and Proteins results in the appendix further support the overall benefit of balancing SFT and RL. This suggests that later post-training passes are better spent on RL once SFT has established task competence, although a compute-normalized study using estimated FLOPs would be needed to identify the exact optimal SFT–RL trade-off.

Finding 6. Under a fixed epoch-level post-training schedule, larger SFT allocations can preserve strong ID performance, but after a supervised warm start, the best ID-OOD trade-off comes from a small amount of additional SFT followed by a larger RL allocation.

5 Discussion

Our results show that post-training stages play distinct roles in biological reasoning. CPT adapts models to biological language, SFT establishes task competence, and RL improves transfer beyond the training distribution. These stages should therefore not be treated as interchangeable sources of compute. In practice, this suggests a simple recipe: use CPT to align models with biological language, use enough SFT to establish task competence, and allocate later post-training to RL, especially when OOD robustness matters.

The SFT-induced trade-off also highlights why biology provides a demanding setting for studying post-training dynamics. In our RNA experiments, OOD accuracy drops by roughly 4 percent from its peak as SFT continues, following an approximately monotonic decline rather than a plateau. Biology exposes generalization failures that are often less apparent in conventional reasoning benchmarks. In mathematics and code, many OOD problems preserve the same underlying structure as the training examples, even when surface details change. In biology, unseen pathways, diseases, species, and perturbations often involve different mechanisms and biological processes. As a result, optimization that improves performance on the training distribution can simultaneously reduce the ability to transfer across biological systems.

Limitations and future work. Our study has several limitations. First, although we evaluate post-training across DNA, RNA, and protein reasoning tasks with biologically meaningful OOD splits, our conclusions rest on a limited set of tasks, benchmarks, and model families. It remains unclear how broadly these trends extend to other scientific reasoning settings and richer biological workflows. Second, our results suggest that RL depends on reward design, task structure, and the quality of the supervised starting point, but we have only begun to characterize these dependencies. Our fixed-schedule SFT-RL study is also not fully compute-normalized. Finally, our evaluation measures final-answer correctness rather than the validity of intermediate reasoning steps. We therefore cannot fully distinguish genuine biological reasoning from shortcut strategies that produce correct outputs. Future work should test these trade-offs on broader benchmarks and examine how reward design, compute-normalized stage allocation, and adaptation capacity shape ID-OOD robustness in scientific reasoning models.

More broadly, our results suggest that progress in scientific reasoning will depend not only on larger models or more post-training compute, but on understanding how different stages shape generalization. In biology, the strongest models are not those that optimize longest on a fixed distribution, but those that preserve the ability to transfer across biological systems. Understanding and controlling these training dynamics may therefore be as important as scaling model size itself.

Acknowledgements

L.F. is supported by the Kempner Graduate Fellowship at Harvard University. H.Z. and S.K. acknowledge the Chan Zuckerberg Initiative Foundation for establishing the Kempner Institute for the Study of Natural and Artificial Intelligence. M.M.L. and M.Z. gratefully acknowledge the support, in part, by NSF CAREER Award 2339524, ARPA-H Biomedical Data Fabric (BDF) Toolbox Program, Amazon Faculty Research, Google Research Scholar Program, AstraZeneca Research, GlaxoSmithKline Award, Roche Alliance with Distinguished Scientists (ROADS) Program, Sanofi iDEA-iTECH Award, Boehringer Ingelheim Award, Merck Award, Optum AI Research Collaboration Award, Pfizer Research, Gates Foundation (INV-079038), Chan Zuckerberg Initiative, Collaborative Center for XDP at Massachusetts General Hospital, John and Virginia Kaneb Fellowship at Harvard Medical School, Biswas Computational Biology Initiative in partnership with the Milken Institute, Harvard Medical School Dean's Innovation Fund for the Use of Artificial Intelligence, and the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders.

Authors affiliated with Google DeepMind and Google Research (Eric Wang, Shekoofeh Azizi, Bryan Perozzi) participated in this work in an advisory capacity only.

References

[1] Adibvafa Fallahpour, Andrew Magnuson, Purav Gupta, Shihao Ma, Jack Naimer, Arnav Shah, Haonan Duan, Omar Ibrahim, Hani Goodarzi, Chris J Maddison, and Bo Wang. Bioreason: Incentivizing multimodal biological reasoning within a dna-llm model. *arXiv preprint arXiv:2505.23579*, 2025. doi: 10.48550/arXiv.2505.23579.

[2] Ana-Maria Istrate, Fausto Milletari, Fabrizio Castrotorres, Jakub Tomczak, Michaela Torkar, Donghui Li, and Theofanis Karaletsos. rbio1-training scientific reasoning llms with biological world models as soft verifiers. *bioRxiv*, pages 2025–08, 2025. doi: 10.1101/2025.08.18.670981.

[3] Adibvafa Fallahpour, Arman Seyed-Ahmadi, Parsa Idehpour, Omar Ibrahim, Purav Gupta, Jack Naimer, Kevin Zhu, Arnav Shah, Shihao Ma, Abhinav Adduri, Talu Güloglu, Nuo Liu, Haotian Cui, Arihant Jain, Max de Castro, Amirfaham Fallahpour, Antonio Cembellin-Prieto, John S. Stiles, Filip Nemčko, Alexander A. Nevue, Hyungseok C. Moon, Lucas Sosnick, Olivia Markham, Haonan Duan, Michelle Y. Y. Lee, Andrea F. M. Salvador, Chris J. Maddison, Christoph A. Thaiss, Chiara Ricci-Tam, Brian S. Plosky, Dave P. Burke, Patrick D. Hsu, Hani Goodarzi, and Bo Wang. Bioreason-pro: Advancing protein function prediction with multimodal biological reasoning. *bioRxiv*, 2026. doi: 10.64898/2026.03.19.712954. URL https://www.biorxiv.org/content/early/2026/03/20/2026.03.19.712954.

[4] Zhenting Qi, Fan Nie, Alexandre Alahi, James Zou, Himabindu Lakkaraju, Yilun Du, Eric Xing, Sham Kakade, and Hanlin Zhang. EvoLM: In search of lost training dynamics for language model reasoning. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, pages 92385–92416, 2025.

[5] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z F Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J L Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R J Chen, R L Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S S Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W L Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X Q Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y K Li, Y Q Wang, Y X Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y X Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z Z Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

[6] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, pages 136065–136108, 2025.

[7] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, pages 64304–64339, 2025.

[8] Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. Megascience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv:2507.16812*, 2025. doi: 10.48550/arXiv.2507.16812.

[9] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G Dimakis, and Ludwig Schmidt. OpenThoughts: data recipes for reasoning models. *arXiv [cs. LG]*, 4 June 2025. doi: 10.48550/arXiv.2506.04178.

[10] Syed Asad Rizvi, Daniel Levine, Aakash Patel, Shiyang Zhang, Eric Wang, Curtis Jamison Perry, Ivan Vrkic, Nicole Mayerli Constante, Zirui Fu, Sizhuang He, David Zhang, Cerise Tang, Zhuoyang Lyu, Rayyan Darji, Chang Li, Emily Sun, David Jeong, Lawrence Zhao, Jennifer Kwan, David Braun, Brian Hafler, Hattie Chung, Rahul M Dhodapkar, Paul Jaeger, Bryan Perozzi, Jeffrey Ishizuka, Shekoofeh Azizi, and David van Dijk. Scaling large language models for next-generation single-cell analysis. *bioRxivorg*, pages 2025–04, 2026. doi: 10.1101/2025.04.14.648850.

[11] Chang Yu, Siyuan Li, Zicheng Liu, Jingbo Zhou, Xianglong Guo, Kai Yu, Yuqing Zhou, Ken Li, Zelin Zang, Zhen Lei, and Stan Z Li. CDBridge: A cross-omics post-training bridge strategy for context-aware biological modeling. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=Hk4Fb6kaYF.

[12] Haoran Sun, Yankai Jiang, Zhenyu Tang, Yaning Pan, Shuang Gu, Zekai Lin, Lilong Wang, Wenjie Lou, Lei Liu, Lei Bai, and Xiaosong Wang. Unleashing scientific reasoning for bio-experimental protocol generation via structured component-based reward mechanism. *arXiv [cs. AI]*, 17 October 2025. doi: 10.48550/arXiv.2510.15600.

[13] Shenghe Zheng, Chenyu Huang, Fangchen Yu, Junchi Yao, Jingqi Ye, Tao Chen, Yun Luo, Ning Ding, Lei Bai, Ganqu Cui, and Peng Ye. Sci-verifier: Scientific verifier with thinking. *arXiv [cs. AI]*, 29 September 2025. doi: 10.48550/arXiv.2509.24285.

[14] Yuhang Chen, Zhen Tan, Ruichen Zhang, Mufan Qiu, and Tianlong Chen. CellDuality: Unlocking biological reasoning in LLMs with self-supervised RLVR. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=I4meJN28Ol.

[15] Zhijian Wei, Runze Ma, Zichen Wang, Zhongmin Li, Shuotong Song, and Shuangjia Zheng. VCWorld: a biological world model for virtual cell simulation. *arXiv [q-bio. CB]*, 29 November 2025. doi: 10.48550/arXiv.2512.00306.

[16] Chang Su, Zhongkai Hao, Zhizhou Zhang, Zeyu Xia, Youjia Wu, Hang Su, and Jun Zhu. Helix: Evolutionary reinforcement learning for open-ended scientific problem solving. *arXiv [cs. LG]*, 2026. doi: 10.48550/arXiv.2603.07642.

[17] Xingwu Chen, Tianle Li, and Difan Zou. Reshaping reasoning in llms: A theoretical analysis of rl training dynamics through pattern selection. *arXiv [cs. LG]*, 5 June 2025. doi: 10.48550/arXiv.2506.04695.

[18] Albert Catalan-Tatjer, Niccolò Ajroldi, and Jonas Geiping. Training dynamics impact post-training quantization robustness. *arXiv [cs. LG]*, 7 October 2025. doi: 10.48550/arXiv.2510.06213.

[19] Fan Chen, Audrey Huang, Noah Golowich, Sadhika Malladi, Adam Block, Jordan T Ash, Akshay Krishnamurthy, and Dylan J Foster. The coverage principle: How pre-training enables post-training. *arXiv [stat. ML]*, 16 October 2025. doi: 10.48550/arXiv.2510.15020.

[20] Zhiting Wei, Yiheng Wang, Yicheng Gao, Shuguang Wang, Ping Li, Duanmiao Si, Yuli Gao, Siqi Wu, Danlu Li, Kejing Dong, Xingbo Yang, Chen Tang, Shaliu Fu, Xiaohan Chen, Wannian Li, Yuzhou You, Chen Zhang, Aibin Liang, Guohui Chuai, and Qi Liu. Benchmarking algorithms for generalizable single-cell perturbation response prediction. *Nature Methods*, 23(2):451–464, 2026. doi: 10.1038/s41592-025-02980-0.

[21] Gabriel Studer, Xavier Robin, Stefan Bienert, Janani Durairaj, Peter Škrinjar, Gerardo Tauriello, Andrew Mark Waterhouse, and Torsten Schwede. A fully automated benchmarking suite to compare macromolecular complexes. *Nature Methods*, 23(2):387–394, 2026. doi: 10.1038/s41592-025-02973-z.

[22] Janani Durairaj, Yusuf Adeshina, Zhonglin Cao, Xuejin Zhang, Vladas Oleinikovas, Thomas Duignan, Zachary McClure, Xavier Robin, Gabriel Studer, Daniel Kovtun, Emanuele Rossi, Guoqing Zhou, Srimukh Veccham, Clemens Isert, Yuxing Peng, Prabindh Sundareson, Mehmet Akdel, Gabriele Corso, Hannes Stärk, Gerardo Tauriello, Zachary Carpenter, Michael Bronstein, Emine Kucukbenli, Torsten Schwede, and Luca Naef. PLINDER: the protein-ligand interactions dataset and evaluation resource. *BioRxiv*, pages 2024–07, 2024. doi: 10.1101/2024.07.17.603955.

[23] Owen Queen, Yepeng Huang, Robert Calef, Valentina Giunchiglia, Tianlong Chen, George Dasoulas, Leann Tai, Gianmarco Abbadessa, Owain Howell, Michelle M Li, Yasha Ektefaie, Ayush Noori, Ildiko Farkas, Joseph Brown, Tom Cobley, Karin Hrovatin, Tom Hartvigsen, Fabian J Theis, Bradley L Pentelute, James Zou, Vikram Khurana, David Owen, Richard Nicholas, Manolis Kellis, and Marinka Zitnik. ProCyon: a multimodal foundation model for protein phenotypes. *bioRxiv*, 2024. doi: 10.1101/2024.12.10.627665.

[24] Yasha Ektefaie, Andrew Shen, Daria Bykova, Maximillian G Marin, Marinka Zitnik, and Maha Farhat. Evaluating generalizability of artificial intelligence models for molecular datasets. *Nature Machine Intelligence*, 6(12):1512–1524, 2024. doi: 10.1038/s42256-024-00931-6.

[25] Kasia Z Kedzierska, Lorin Crawford, Ava P Amini, and Alex X Lu. Zero-shot evaluation reveals limitations of single-cell foundation models. *Genome Biology*, 26(1):101, 2025. doi: 10.1186/s13059-025-03574-x.

[26] Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22(8):1657–1661, 2025. doi: 10.1038/s41592-025-02772-6.

[27] Siyuan Wang, Gaokai Zhang, Li Lyna Zhang, Ning Shang, Fan Yang, Dongyao Chen, and Mao Yang. LoongRL: reinforcement learning for advanced reasoning over long contexts. *arXiv [cs. CL]*, 22 October 2025. doi: 10.48550/arXiv.2510.19363.

[28] Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for llms. *arXiv [cs. LG]*, 15 October 2025. doi: 10.48550/arXiv.2510.13786.

[29] Cong Jiang, Xiaofeng Zhang, Fangzhi Zhu, XiaoWei Chen, Junxiong Zhu, and Zheng Zhang. Rethinking LLM reasoning: From explicit trajectories to latent representations. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=CbK7lYbmv8.

[30] Kehua Feng, Keyan Ding, Zhihui Zhu, Lei Liang, Qiang Zhang, and Huajun Chen. CoT-Evo: evolutionary distillation of chain-of-thought for scientific reasoning. *arXiv [cs. CL]*, 15 October 2025. doi: 10.48550/arXiv.2510.13166.

[31] Yiming Gao, Zhen Wang, Jefferson Chen, Mark Antkowiak, Mengzhou Hu, Jungho Kong, Dexter Pratt, Jieyuan Liu, Enze Ma, Zhiting Hu, and Eric P Xing. scPilot: Large language model reasoning toward automated single-cell analysis and discovery. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, pages 1307–1344, 2025.

[32] Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. AI-researcher: autonomous scientific innovation. *arXiv [cs. AI]*, 2025. doi: 10.48550/arXiv.2505.18705.

[33] 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. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, pages 175068–175107, 2025.

[34] Elana Simon, Kyle Swanson, and James Zou. Language models for biological research: a primer. *Nature Methods*, 21(8):1422–1429, 2024. doi: 10.1038/s41592-024-02354-y.

[35] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021. doi: 10.1093/bioinformatics/btab083.

[36] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023. doi: 10.48550/arXiv.2306.15006.

[37] 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, Jeremy Sullivan, Madelena Y Ng, Ashley Lewis, Aaron Lou, Stefano Ermon, Stephen A Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D Hsu, and Brian L Hie. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723):eado9336, 2024. doi: 10.1126/science.ado9336.

[38] Garyk Brixi, Matthew G Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, Eric Nguyen, Chiara Ricci-Tam, David W Romero, Jonathan C Schmok, Ali Taghibakhshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K Wang, Michael T Pearce, Elana Simon, Etowah Adams, Zachary J Amador, Euan A Ashley, Stephen A Baccus, Haoyu Dai, Steven Dillmann, Stefano Ermon, Daniel Guo, Michael H Herschl, Rajesh Ilango, Ken Janik, Amy X Lu, Reshma Mehta, Mohammad R K Mofrad, Madelena Y Ng, Jaspreet Pannu, Christopher Ré, John St John, Jeremy Sullivan, Joseph Tey, Ben Viggiano, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Thomas McGrath, Kimberly Powell, Sudarshan Pinglay, Dave P Burke, Hani Goodarzi, Patrick D Hsu, and Brian L Hie. Genome modelling and design across all domains of life with evo 2. *Nature*, 652(8112):1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

[39] 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, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025. doi: 10.1038/s41592-024-02523-z.

[40] Žiga Avsec, Natasha Latysheva, Jun Cheng, Guido Novati, Kyle Taylor, Tom Ward, Clare Bycroft, Lauren Nicolaisen, Eirini Arvaniti, Joshua Pan, Raina Thomas, Vincent Dutordoir, Matteo Perino, Soham De, Alexander Karollus, Adam Gayoso, Toby Sargeant, Anne Mottram, Lai Hong Wong, Pavol Drotár, Adam Kosiorek, Andrew Senior, Richard Tanburn, Taylor Applebaum, Souradeep Basu, Demis Hassabis, and Pushmeet Kohli. Alphagenome: advancing regulatory variant effect prediction with a unified dna sequence model. *BioRxiv*, pages 2025–06, 2025. doi: 10.1101/2025.06.25.661532.

[41] Andre Cornman, Jacob West-Roberts, Antonio Pedro Camargo, Simon Roux, Martin Beracochea, Milot Mirdita, Sergey Ovchinnikov, and Yunha Hwang. The omg dataset: An open metagenomic corpus for mixed-modality genomic language modeling. *bioRxiv*, pages 2024–08, 2024. doi: 10.1101/2024.08.14.607850.

[42] Yusen Hou, Weicai Long, Haitao Hu, Houcheng Su, Junning Feng, and Yanlin Zhang. PhageBench: Can LLMs understand raw bacteriophage genomes? In *Findings of the Association for Computational Linguistics (ACL)*, pages 28674–28698, 2026. doi: 10.18653/v1/2026.findings-acl.1432.

[43] Philip Fradkin, Ruian "Ian" Shi, Taykhoom Dalal, Keren Isaev, Brendan J Frey, Leo J Lee, Quaid Morris, and Bo Wang. Orthrus: toward evolutionary and functional rna foundation models. *Nature Methods*, 23(5):935–945, 2026. doi: 10.1038/s41592-026-03064-3.

[44] Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, Irwin King, and Yu Li. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. *bioRxiv*, 2022. doi: 10.1101/2022.08.06.503062. Preprint, arXiv:2204.00300.

[45] James Pearce, Sara Simmonds, Gita Mahmoudabadi, Lakshmi Krishnan, Giovanni Palla, Ana-Maria Istrate, Alexander Tarashansky, Benjamin Nelson, Omar Valenzuela, Donghui Li, Stephen R Quake, and Theofanis Karaletsos. TranscriptFormer: A generative cell atlas across 1.5 billion years of evolution. *Science*, 393(6806), 2026. doi: 10.1126/science.aec8514.

[46] 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. doi: 10.1038/s41592-024-02201-0.

[47] 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, and Patrick T Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023. doi: 10.1038/s41586-023-06139-9.

[48] Abhinav Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, Chiara Ricci-Tam, Christopher Carpenter, Vishvak Subramanyam, Aidan Winters, Sravya Tirukkovular, Jeremy Sullivan, Brian Plosky, Basak Eraslan, Nicholas Youngblut, Jure Leskovec, Luke Gilbert, Silvana Konermann, Patrick Hsu, Alexander Dobin, Dave Burke, Hani Goodarzi, and Yusuf Roohani. Predicting cellular responses to perturbation across diverse contexts with state. *BioRxiv*, pages 2025–06, 2025. doi: 10.1101/2025.06.26.661135.

[49] Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024. doi: 10.1038/s41592-024-02305-7.

[50] Ana-Maria Istrate, Donghui Li, and Theofanis Karaletsos. scgenept: Is language all you need for modeling single-cell perturbations? *bioRxiv*, pages 2024–10, 2024. doi: 10.1101/2024.10.23.619972.

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

[52] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S Molina, Neil Thomas, Yousuf A Khan, Chetan Mishra, Carolyn Kim, Liam J Bartie, Matthew Nemeth, Patrick D Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018.

[53] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978.e3, 2023. doi: 10.1016/j.cels.2023.10.002.

[54] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019. doi: 10.1038/s41592-019-0598-1.

[55] Moritz Schaefer, Peter Peneder, Daniel Malzl, Salvo Danilo Lombardo, Mihaela Peycheva, Jake Burton, Anna Hakobyan, Varun Sharma, Thomas Krausgruber, Celine Sin, Jörg Menche, Eleni M Tomazou, and Christoph Bock. Multimodal learning enables chat-based exploration of

single-cell data. *Nature Biotechnology*, 2025. doi: 10.1038/s41587-025-02857-9. Advance online publication.

[56] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[57] Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, Xu Tan, Jie Fu, Wenbo Su, Jiamang Wang, Lin Qu, and Bo Zheng. D-CPT law: Domain-specific continual pre-training scaling law for large language models. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pages 90318–90354, 2024.

[58] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations (ICLR)*, 2024.

[59] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8342–8360, 2020. doi: 10.18653/v1/2020.acl-main.740.

[60] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. doi: 10.48550/arXiv.2001.08361.

[61] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[62] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430, 2023.

[63] Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets LLM finetuning: The effect of data, model and finetuning method. *arXiv [cs.CL]*, 2024. doi: 10.48550/arXiv.2402.17193.

[64] Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to (re)warm your model? *arXiv preprint arXiv:2308.04014*, 2023. doi: 10.48550/arXiv.2308.04014.

[65] Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024. doi: 10.48550/arXiv.2403.08763.

[66] Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models. *arXiv preprint arXiv:2407.07263*, 2024. doi: 10.48550/arXiv.2407.07263.

[67] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. *arXiv [cs.CL]*, 2023. doi: 10.48550/arXiv.2302.03241.

[68] Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models to domains via reading comprehension. In *International Conference on Learning Representations (ICLR)*, 2024.

[69] Cursor Research. Composer 2 technical report. *arXiv preprint arXiv:2603.24477*, 2026. doi: 10.48550/arXiv.2603.24477.

[70] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 10818–10838, 2025.

[71] Zhizheng Wang, Yifan Yang, Qiao Jin, and Zhiyong Lu. Gene-R1: Reasoning with data-augmented lightweight LLMs for gene set analysis. In *Biocomputing 2026: Proceedings of the Pacific Symposium*, pages 494–507, 2026. doi: 10.1142/9789819824755_0035.

[72] Ming Yin, Yuanhao Qu, Ling Yang, Le Cong, and Mengdi Wang. Toward scientific reasoning in llms: Training from expert discussions via reinforcement learning. *arXiv preprint arXiv:2505.19501*, 2025. doi: 10.48550/arXiv.2505.19501.

[73] Pengwei Sui, Michelle M Li, Shanghua Gao, Wanxiang Shen, Valentina Giunchiglia, Andrew Shen, Yepeng Huang, Zhenglun Kong, and Marinka Zitnik. Medea: An omics ai agent for therapeutic discovery. *bioRxivorg*, pages 2026–01, 2026. doi: 10.64898/2026.01.16.696667.

[74] Typhaine Paysan-Lafosse, Matthias Blum, Sara Chuguransky, Tiago Grego, Beatriz Lázaro Pinto, Gustavo A Salazar, Maxwell L Bileschi, Peer Bork, Alan Bridge, Lucy Colwell, Julian Gough, Daniel H Haft, Ivica Letunić, Aron Marchler-Bauer, Huaiyu Mi, Darren A Natale, Christine A Orengo, Arun P Pandurangan, Catherine Rivoire, Christian J A Sigrist, Ian Sillitoe, Narmada Thanki, Paul D Thomas, Silvio C E Tosatto, Cathy H Wu, and Alex Bateman. Interpro in 2022. *Nucleic acids research*, 51(D1):D418–D427, 2023. doi: 10.1093/nar/gkac993.

[75] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388.

[76] Google DeepMind. Gemma 4. https://deepmind.google/models/gemma/gemma-4/, 2026.

[77] m-a-p. FineFineWeb: A comprehensive study on fine-grained domain web corpus. https://huggingface.co/datasets/m-a-p/FineFineWeb, 2024.

[78] M Kanehisa and S Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000. doi: 10.1093/nar/28.1.27.

[79] Minoru Kanehisa, Yoko Sato, Miho Furumichi, Kanae Morishima, and Mao Tanabe. New approach for understanding genome variations in KEGG. *Nucleic Acids Research*, 47(D1): D590–D595, 2019. doi: 10.1093/nar/gky962.

[80] Melissa J Landrum, Shanmuga Chitipiralla, Garth R Brown, Chao Chen, Baoshan Gu, Jennifer Hart, Douglas Hoffman, Wonhee Jang, Kuljeet Kaur, Chunlei Liu, Vitaly Lyoshin, Zenith Maddipatla, Rama Maiti, Joseph Mitchell, Nuala O'Leary, George R Riley, Wenyao Shi, George Zhou, Valerie Schneider, Donna Maglott, J Bradley Holmes, and Brandi L Kattman. ClinVar: improvements to accessing data. *Nucleic Acids Research*, 48(D1):D835–D844, 2020. doi: 10.1093/nar/gkz972.

[81] S T Sherry, M H Ward, M Kholodov, J Baker, L Phan, E M Smigielski, and K Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1):308–311, 2001. doi: 10.1093/nar/29.1.308.

[82] Joanna S Amberger, Carol A Bocchini, François Schiettecatte, Alan F Scott, and Ada Hamosh. OMIM.org: Online mendelian inheritance in man (OMIM®), an online catalog of human genes and genetic disorders. *Nucleic Acids Research*, 43(Database issue):D789–98, 2015. doi: 10.1093/nar/gku1205.

[83] John G Tate, Sally Bamford, Harry C Jubb, Zbyslaw Sondka, David M Beare, Nidhi Bindal, Harry Boutselakis, Charlotte G Cole, Celestino Creatore, Elisabeth Dawson, Peter Fish, Bhavana Harsha, Charlie Hathaway, Steve C Jupe, Chai Yin Kok, Kate Noble, Laura Ponting, Christopher C Ramshaw, Claire E Rye, Helen E Speedy, Ray Stefancsik, Sam L Thompson, Shicai Wang, Sari Ward, Peter J Campbell, and Simon A Forbes. COSMIC: the catalogue of somatic mutations in cancer. *Nucleic Acids Research*, 47(D1):D941–D947, 2019. doi: 10.1093/nar/gky1015.

[84] CZI Cell Science Program, Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M Bell, Emanuele Bezzi, Batuhan Cakir, Jim Chaffer, Signe Chambers, J Michael Cherry, Tiffany Chi, Jennifer Chien, Leah Dorman, Pablo Garcia-Nieto, Nayib Gloria, Mim Hastie, Daniel Hegeman, Jason Hilton, Timmy Huang, Amanda Infeld, Ana-Maria Istrate, Ivana Jelic, Kuni Katsuya, Yang Joon Kim, Karen Liang, Mike Lin, Maximilian Lombardo, Bailey Marshall, Bruce Martin, Fran McDade, Colin Megill, Nikhil Patel, Alexander Predeus, Brian Raymor, Behnam Robatmili, Dave Rogers, Erica Rutherford, Dana Sadgat, Andrew Shin, Corinn Small, Trent Smith, Prathap Sridharan, Alexander Tarashansky, Norbert Tavares, Harley Thomas, Andrew Tolopko, Meghan Urisko, Joyce Yan, Garabet Yeretssian, Jennifer Zamanian, Arathi Mani, Jonah Cool, and Ambrose Carr. CZ CELLxGENE Discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic Acids Research*, 53(D1):D886–D900, 2025. doi: 10.1093/nar/gkae1142.

[85] David Ochoa, Andrew Hercules, Miguel Carmona, Daniel Suveges, Jarrod Baker, Cinzia Malangone, Irene Lopez, Alfredo Miranda, Carlos Cruz-Castillo, Luca Fumis, Manuel Bernal-Llinares, Kirill Tsukanov, Helena Cornu, Konstantinos Tsirigos, Olesya Razuvayevskaya, Annalisa Buniello, Jeremy Schwartzentruber, Mohd Karim, Bruno Ariano, Ricardo Esteban Martinez Osorio, Javier Ferrer, Xiangyu Ge, Sandra Machlitt-Northen, Asier Gonzalez-Uriarte, Shyamasree Saha, Santosh Tirunagari, Chintan Mehta, Juan María Roldán-Romero, Stuart Horswell, Sarah Young, Maya Ghoussaini, David G Hulcoop, Ian Dunham, and Ellen M McDonagh. The next-generation Open Targets Platform: reimagined, redesigned, rebuilt. *Nucleic Acids Research*, 51(D1):D1353–D1359, 2023. doi: 10.1093/nar/gkac1046.

[86] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P Overington. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(Database issue):D1100–7, 2012. doi: 10.1093/nar/gkr777.

[87] UniProt Consortium. UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023. doi: 10.1093/nar/gkac1052.

[88] Rachael P Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J Martin, and Claire O'Donovan. The GOA database: Gene ontology annotation updates for 2015. *Nucleic Acids Research*, 43(Database issue):D1057–63, 2015. doi: 10.1093/nar/gku1113.

[89] Gene Ontology Consortium, Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, David P Hill, Raymond Lee, Huaiyu Mi, Sierra Moxon, Christopher J Mungall, Anushya Muruganugan, Tremayne Mushayahama, Paul W Sternberg, Paul D Thomas, Kimberly Van Auken, Jolene Ramsey, Deborah A Siegele, Rex L Chisholm, Petra Fey, Maria Cristina Aspromonte, Maria Victoria Nugnes, Federica Quaglia, Silvio Tosatto, Michelle Giglio, Suvarna Nadendla, Giulia Antonazzo, Helen Attrill, Gil Dos Santos, Steven Marygold, Victor Strelets, Christopher J Tabone, Jim Thurmond, Pinglei Zhou, Saadullah H Ahmed, Praoparn Asanitthong, Diana Luna Buitrago, Meltem N Erdol, Matthew C Gage, Mohamed Ali Kadhum, Kan Yan Chloe Li, Miao Long, Aleksandra Michalak, Angeline Pesala, Armalya Pritazahra, Shirin C C Saverimuttu, Renzhi Su, Kate E Thurlow, Ruth C Lovering, Colin Logie, Snezhana Oliferenko, Judith Blake, Karen Christie, Lori Corbani, Mary E Dolan, Harold J Drabkin, David P Hill, Li Ni, Dmitry Sitnikov,

Cynthia Smith, Alayne Cuzick, James Seager, Laurel Cooper, Justin Elser, Pankaj Jaiswal, Parul Gupta, Pankaj Jaiswal, Sushma Naithani, Manuel Lera-Ramirez, Kim Rutherford, Valerie Wood, Jeffrey L De Pons, Melinda R Dwinell, G Thomas Hayman, Mary L Kaldunski, Anne E Kwitek, Stanley J F Laulederkind, Marek A Tutaj, Mahima Vedi, Shur-Jen Wang, Peter D'Eustachio, Lucila Aimo, Kristian Axelsen, Alan Bridge, Nevila Hyka-Nouspikel, Anne Morgat, Suzi A Aleksander, J Michael Cherry, Stacia R Engel, Kalpana Karra, Stuart R Miyasato, Robert S Nash, Marek S Skrzypek, Shuai Weng, Edith D Wong, Erika Bakker, Tanya Z Berardini, Leonore Reiser, Andrea Auchincloss, Kristian Axelsen, Ghislaine Argoud-Puy, Marie-Claude Blatter, Emmanuel Boutet, Lionel Breuza, Alan Bridge, Cristina Casals-Casas, Elisabeth Coudert, Anne Estreicher, Maria Livia Famiglietti, Marc Feuermann, Arnaud Gos, Nadine Gruaz-Gumowski, Chantal Hulo, Nevila Hyka-Nouspikel, Florence Jungo, Philippe Le Mercier, Damien Lieberherr, Patrick Masson, Anne Morgat, Ivo Pedruzzi, Lucille Pourcel, Sylvain Poux, Catherine Rivoire, Shyamala Sundaram, Alex Bateman, Emily Bowler-Barnett, Hema Bye-A-Jee, Paul Denny, Alexandr Ignatchenko, Rizwan Ishtiaq, Antonia Lock, Yvonne Lussi, Michele Magrane, Maria J Martin, Sandra Orchard, Pedro Raposo, Elena Speretta, Nidhi Tyagi, Kate Warner, Rossana Zaru, Alexander D Diehl, Raymond Lee, Juancarlos Chan, Stavros Diamantakis, Daniela Raciti, Magdalena Zarowiecki, Malcolm Fisher, Christina James-Zorn, Virgilio Ponferrada, Aaron Zorn, Sridhar Ramachandran, Leyla Ruzicka, and Monte Westerfield. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 2023. doi: 10.1093/genetics/iyad031.

[90] Damian Szklarczyk, Katerina Nastou, Mikaela Koutrouli, Rebecca Kirsch, Farrokh Mehryary, Radja Hachilif, Dewei Hu, Matteo E Peluso, Qingyao Huang, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, Peer Bork, Lars J Jensen, and Christian von Mering. The STRING database in 2025: protein networks with directionality of regulation. *Nucleic Acids Research*, 53(D1): D730–D737, 2025. doi: 10.1093/nar/gkae1113.

[91] Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, Gaurav Pandey, Jeffrey M Yunes, Ameet S Talwalkar, Susanna Repo, Michael L Souza, Damiano Piovesan, Rita Casadio, Zheng Wang, Jianlin Cheng, Hai Fang, Julian Gough, Patrik Koskinen, Petri Törönen, Jussi Nokso-Koivisto, Liisa Holm, Domenico Cozzetto, Daniel W A Buchan, Kevin Bryson, David T Jones, Bhakti Limaye, Harshal Inamdar, Avik Datta, Sunitha K Manjari, Rajendra Joshi, Meghana Chitale, Daisuke Kihara, Andreas M Lisewski, Serkan Erdin, Eric Venner, Olivier Lichtarge, Robert Rentzsch, Haixuan Yang, Alfonso E Romero, Prajwal Bhat, Alberto Paccanaro, Tobias Hamp, Rebecca Kaßner, Stefan Seemayer, Esmeralda Vicedo, Christian Schaefer, Dominik Achten, Florian Auer, Ariane Boehm, Tatjana Braun, Maximilian Hecht, Mark Heron, Peter Hönigschmid, Thomas A Hopf, Stefanie Kaufmann, Michael Kiening, Denis Krompass, Cedric Landerer, Yannick Mahlich, Manfred Roos, Jari Björne, Tapio Salakoski, Andrew Wong, Hagit Shatkay, Fanny Gatzmann, Ingolf Sommer, Mark N Wass, Michael J E Sternberg, Nives Škunca, Fran Supek, Matko Bošnjak, Panče Panov, Sašo Džeroski, Tomislav Šmuc, Yiannis A I Kourmpetis, Aalt D J van Dijk, Cajo J F ter Braak, Yuanpeng Zhou, Qingtian Gong, Xinran Dong, Weidong Tian, Marco Falda, Paolo Fontana, Enrico Lavezzo, Barbara Di Camillo, Stefano Toppo, Liang Lan, Nemanja Djuric, Yuhong Guo, Slobodan Vucetic, Amos Bairoch, Michal Linial, Patricia C Babbitt, Steven E Brenner, Christine Orengo, Burkhard Rost, Sean D Mooney, and Iddo Friedberg. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, 2013. doi: 10.1038/nmeth.2340.

[92] Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsoh, Alex W Crocker, Kimberley A Lewis, George Georghiou, Huy N Nguyen, Md Nafiz Hamid, Larry Davis, Tunca Dogan, Volkan Atalay, Ahmet S Rifaioglu, Alperen Dalkıran, Rengul Cetin Atalay, Chengxin Zhang, Rebecca L Hurto, Peter L Freddolino, Yang Zhang, Prajwal Bhat, Fran Supek, José M Fernández, Branislava Gemovic, Vladimir R Perovic, Radoslav S Davidović, Neven Sumonja, Nevena Veljkovic, Ehsaneddin Asgari, Mohammad R K Mofrad, Giuseppe Profiti, Castrense Savojardo, Pier Luigi Martelli, Rita Casadio, Florian Boecker, Heiko Schoof, Indika Kahanda, Natalie Thurlby, Alice C McHardy, Alexandre Renaux, Rabie Saidi, Julian Gough, Alex A Freitas, Magdalena Antczak, Fabio Fabris, Mark N Wass, Jie Hou, Jianlin Cheng, Zheng Wang, Alfonso E Romero, Alberto Paccanaro, Haixuan Yang, Tatyana Goldberg, Chenguang Zhao, Liisa Holm, Petri Törönen, Alan J Medlar, Elaine Zosa, Itamar Borukhov, Ilya Novikov, Angela Wilkins, Olivier Lichtarge, Po-Han Chi, Wei-Cheng Tseng, Michal Linial, Peter W Rose, Christophe Dessimoz, Vedrana Vidulin, Saso Dzeroski, Ian Sillitoe, Sayoni Das,

Jonathan Gill Lees, David T Jones, Cen Wan, Domenico Cozzetto, Rui Fa, Mateo Torres, Alex Warwick Vesztrocy, Jose Manuel Rodriguez, Michael L Tress, Marco Frasca, Marco Notaro, Giuliano Grossi, Alessandro Petrini, Matteo Re, Giorgio Valentini, Marco Mesiti, Daniel B Roche, Jonas Reeb, David W Ritchie, Sabeur Aridhi, Seyed Ziaeddin Alborzi, Marie-Dominique Devignes, Da Chen Emily Koo, Richard Bonneau, Vladimir Gligorijević, Meet Barot, Hai Fang, Stefano Toppo, Enrico Lavezzo, Marco Falda, Michele Berselli, Silvio C E Tosatto, Marco Carraro, Damiano Piovesan, Hafeez Ur Rehman, Qizhong Mao, Shanshan Zhang, Slobodan Vucetic, Gage S Black, Dane Jo, Erica Suh, Jonathan B Dayton, Dallas J Larsen, Ashton R Omdahl, Liam J McGuffin, Danielle A Brackenridge, Patricia C Babbitt, Jeffrey M Yunes, Paolo Fontana, Feng Zhang, Shanfeng Zhu, Ronghui You, Zihan Zhang, Suyang Dai, Shuwei Yao, Weidong Tian, Renzhi Cao, Caleb Chandler, Miguel Amezola, Devon Johnson, Jia-Ming Chang, Wen-Hung Liao, Yi-Wei Liu, Stefano Pascarelli, Yotam Frank, Robert Hoehndorf, Maxat Kulmanov, Imane Boudellioua, Gianfranco Politano, Stefano Di Carlo, Alfredo Benso, Kai Hakala, Filip Ginter, Farrokh Mehryary, Suwisa Kaewphan, Jari Björne, Hans Moen, Martti E E Tolvanen, Tapio Salakoski, Daisuke Kihara, Aashish Jain, Tomislav Šmuc, Adrian Altenhoff, Asa Ben-Hur, Burkhard Rost, Steven E Brenner, Christine A Orengo, Constance J Jeffery, Giovanni Bosco, Deborah A Hogan, Maria J Martin, Claire O'Donovan, Sean D Mooney, Casey S Greene, Predrag Radivojac, and Iddo Friedberg. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, 20(1):244, 2019. doi: 10.1186/s13059-019-1835-8.

[93] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[94] Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems 37 (NeurIPS) Datasets and Benchmarks Track*, pages 30811–30849, 2024.

[95] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[96] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2020. ISBN 9781728199986. doi: 10.1109/SC41405.2020.00024.

Appendices

A Additional Details on Biological Reasoning Tasks **22**

A.1 Pathway Prediction Dataset and Construction 22

A.2 Target Identification Dataset and Simplifications Relative to MEDEA 22

A.3 Protein Function Prediction Dataset and Simplifications Relative to BioReason-Pro 23

A.4 Example Prompts and Inputs for Each Task 23

B Experimental Setup **25**

B.1 Base Models, Tokenization, and Input Representations 25

B.2 Continued Pre-training (Mid-training) Setup 26

B.3 Supervised Fine-Tuning Setup 26

B.4 Reinforcement-Learning (GRPO) Setup 27

B.5 LoRA Configurations and Trainable-Parameter Choices 28

B.6 Hyperparameters, Context Windows, Sequence Lengths, and Optimization 28

B.7 Compute Resources and Training Budget 29

C Main-Results Tables **29**

C.1 Results for Figure 2 29

C.2 Results for Figure 3 30

C.3 Results for Figure 4 30

C.4 Results for Figure 5 30

C.5 Results for Figure 6 31

C.6 Results for Figure 8 31

D Additional Experiments **31**

D.1 Scaling Post-Training for Biological Non-Reasoning Tasks 31

D.2 DNA Epoch Allocation 32

D.3 Fixed Epoch, Variable Data during RL 32

D.4 Fixed Budget, Variable Data Allocation between SFT and RL 33

D.5 Reward Model Ablations 33

D.6 ID/ OOD Split Ablations 33

E Asset Licenses and Redistribution Status **35**

A Additional Details on Biological Reasoning Tasks

A.1 Pathway Prediction Dataset and Construction

Our pathway prediction benchmark follows the KEGG-derived reasoning dataset introduced in BIOREASON [1]. The source data begins from KEGG Network Variants and associated disease-pathway annotations [78, 79], which BIOREASON then augments with linked variant metadata from ClinVar, dbSNP, OMIM, and COSMIC [80–83]. The resulting benchmark contains 1,449 examples spanning 298 pathway networks and 37 unique diseases. In the original curation, the key design goal was not simply to label variants, but to preserve the mechanistic chain from mutation to pathway perturbation to phenotype, so that each example could support explicit multi-step biological reasoning rather than endpoint classification alone.

Technically, each KEGG pathway is represented as a structured molecular interaction network using a standardized symbolic notation that encodes activation, inhibition, complex formation, and transcriptional regulation [78, 79]. These pathway graphs are then linked to specific variants through a semi-automated mapping procedure designed to preserve the relationship between genomic loci and pathway entities. For each mapped example, the dataset stores paired reference and variant DNA sequences with precise alignment coordinates; in BIOREASON, these sequences average roughly 4,000 base pairs, with most mutations differing from the reference by only 1–3 nucleotides. The final supervised example consists of variant details, a network definition, and gene-level context on the input side, together with a concise mechanism-to-disease answer and a full reasoning trace on the output side.

A distinctive part of the construction is the generation of causal reasoning paths. In BIOREASON, these traces were produced using Claude 3.7 Sonnet and grounded with contextual disease information from the KEGG disease database [78, 79], then packaged into standardized question-answer pairs for training and evaluation [1]. The reasoning traces have mean length 303.8 words and are intended to make the latent biological mechanism explicit: they verbalize how the mutation perturbs the affected gene, how that perturbation propagates through intermediate pathway interactions, and why the resulting network state is associated with the target disease. In our work, we inherit this benchmark structure but evaluate generalization by splitting at the level of pathway networks, so that out-of-domain examples require transfer to previously unseen molecular systems rather than new variants within familiar ones.

A.2 Target Identification Dataset and Simplifications Relative to MEDEA

Our target identification benchmark is adapted from the cell type specific target nomination task introduced in MEDEA [73], but we convert it from an agentic workflow into a fixed-input reasoning problem. In the original MEDEA setup, the task is defined over five diseases—rheumatoid arthritis, type 1 diabetes mellitus, Sjogren's syndrome, hepatoblastoma, and follicular lymphoma—and 29 cell types, with each analysis asking the model to select the best therapeutic target from a set of five candidate genes for a specified disease–cell-type context. The full benchmark contains 2,400 analyses in total, generated from disease atlases and target-disease resources to test whether models can identify therapeutically plausible targets at cell-type resolution rather than from bulk tissue averages.

The dataset construction in MEDEA proceeds in three stages. First, each disease atlas is processed from CELLxGENE [84] using a standard single-cell pipeline, followed by one-vs-all differential expression analysis to identify disease-specific marker genes for each cell type and disease combination. Second, disease-associated genes are collected from Open Targets [85], keeping genes with nonzero genetic evidence or ChEMBL evidence [86]. Third, ground-truth cell type specific disease targets are defined as genes satisfying both criteria: they are differentially expressed in the relevant disease–cell-type context and supported by disease-target evidence. For each context, MEDEA then forms five-gene candidate sets by sampling one positive target and four negatives, and uses prompt paraphrasing plus multiple random seeds to generate the final benchmark.

Our version keeps the same core supervision signal but simplifies the task substantially relative to MEDEA [73]. Rather than asking an agent to construct a research plan, invoke tools, retrieve literature, and reconcile evidence across multiple modules, we provide the model with the disease, cell type, candidate genes, and transcriptomic evidence directly. Concretely, instead of tool-based

retrieval over single-cell atlases and other external resources, we supply aligned TranscriptFormer embeddings [45] for the five candidate genes in both normal and disease states for the relevant context. This removes the planning, execution, and literature-reasoning burden while preserving the central inferential challenge: the model must still identify which candidate is most supported in the specified disease and cell type, now using a fixed multimodal input rather than an open-ended agentic pipeline. Consistent with the main text, we further use four of the five diseases as the in-domain pool for train/ID-test splitting and reserve hepatoblastoma as the OOD test disease.

A.3 Protein Function Prediction Dataset and Simplifications Relative to BioReason-Pro

Our protein function prediction benchmark is adapted from the curated UniProt-based dataset introduced in BioReason-Pro [3]. The original corpus is designed around experimentally supported protein function annotation rather than generic sequence-level pretraining, and integrates multiple biological modalities into a single example. Starting from UniProt and the GOA database [87, 88], BioReason-Pro retains only proteins with experimental or curated GO evidence codes, standardizes annotations to the January 2023 Gene Ontology [89], and propagates terms upward through the ontology hierarchy to preserve hierarchical completeness. The resulting dataset contains 133,492 proteins spanning 3,135 organisms, with each protein linked not only to its amino acid sequence, but also to organism metadata, subcellular localization, InterPro domain annotations [74], structural information, and protein–protein interaction context [90].

At the instance level, BioReason-Pro constructs a compact multimodal context for each protein by combining InterPro domains with residue ranges [74], the UniProt protein description [87], organism, subcellular localization, STRING interaction partners [90], and GO leaf terms across molecular function, biological process, and cellular component [89]. These contexts are then used to generate synthetic step-by-step reasoning traces, which end in a structured final answer containing a concise function summary, the relevant InterPro domains, GO terms, and an interaction hypothesis. Evaluation follows the CAFA temporal holdout protocol [91, 92]: proteins annotated before November 2022 are used for training and validation, while test proteins are selected from those that gained new experimental annotations between March 2023 and February 2024 and lacked annotations in the target aspect beforehand. The final temporal holdout test set contains 8,630 proteins and 230,824 propagated GO annotations.

Our version keeps the same protein function prediction task, but simplifies both the model inputs and the supervised target so the setup matches the multimodal format used throughout this paper [3]. In BioReason-Pro, the model consumes residue-level ESM3 embeddings [52], organism and textual biological context, initial GO-term speculations, and an additional GO graph encoder that injects explicit ontology structure into the language model [89]. Its supervised traces walk the ontology by depth and emit separate molecular-function, biological-process, and cellular-component blocks, followed by a GO summary and a function description. In our benchmark we remove the GO graph encoder. The model receives ESM3 residue embeddings together with text context consisting of the organism, InterPro domain annotations, and those initial GO-term speculations, and must infer function without ontology-graph embeddings. Protein–protein interactions are mentioned in the instruction, but interaction partners are not provided. We also shorten the target. The answer contains only the gold leaf GO identifiers, written between `|GO_SUMMARY_START|` and `|GO_SUMMARY_END|`. The reasoning is not rewritten: sentences are copied from the original trace, with priority given to those that cite a gold leaf term, renumbered as steps, and cut so the full assistant span matches the length distribution of the BioReason DNA supervised targets, with a hard ceiling of 688 tokens. This preserves the core challenge of integrating sequence-derived and symbolic evidence, while making the protein task comparable in both architecture and target length to the DNA and RNA settings.

A.4 Example Prompts and Inputs for Each Task

Pathway prediction. A representative input consists of two versions of the same DNA sequence region, with and without the mutation, including DNA-specific start and padding tokens, followed by a pathway network definition and gene annotations. The model is then asked to infer the biological or disease effect associated with the allele. For example:

```
<|dna_pad|>...<|dna_pad|>
```

Question: Network Definition of the pathway: SOD1* -| BIP -| ERN1 -> XBP1; Genes in the pathway: *SOD1*; superoxide dismutase 1 | *HSPA5*; heat shock protein family A (Hsp70) member 5 | *ERN1*; endoplasmic reticulum to nucleus signaling 1 | *XBP1*; X-box binding protein 1. Given this context, what is the biological effect of this *SOD1* allele, specifically what disease does this contribute to?

Answer: amyotrophic lateral sclerosis

Drug target identification. A representative input consists of prepended RNA padding tokens followed by a disease- and cell-type-specific target selection question over candidate genes. For example:

<|rna_pad|>...<|rna_pad|>

Question: Among the genes *PDRG1*, *PIK3CG*, *TRIM23*, *SUCO*, and *GCKR*, which one exhibits the highest T cell-specific expression relevant for targeted intervention in follicular lymphoma?

Answer: *PIK3CG*

The five projected TranscriptFormer representations are inserted before the text prompt in the same order as the five candidate gene names listed in the question; this fixed ordering defines the alignment between continuous RNA embeddings and discrete gene symbols.

Protein function prediction. A representative input is a single user turn. It begins with one protein placeholder, which is expanded to one <|protein_pad|> per residue plus two ESM special tokens. The turn then contains the instruction, the organism, InterPro annotations, and initial GO term speculations. For example:

<|im_start|>user

Protein: <|protein_pad|><|protein_pad|>...<|protein_pad|>

You are a scientific assistant specialized in protein function prediction. Given a protein sequence, organism information, and additional context (InterPro domain annotations and/or initial GO term speculations), step-by-step reason about the InterPro terms, Gene Ontology (GO) terms regarding molecular function, biological process, and cellular component, protein-protein interactions (PPI), and overall function. Use the provided information as a starting point and improve upon it with deeper analysis. Provide a summary of your findings in your final answer. When you summarize your conclusions, list every Gene Ontology identifier you endorse between <|GO_SUMMARY_START|> and <|GO_SUMMARY_END|> (for example: <|GO_SUMMARY_START|> GO:0005515 GO:0006915 <|GO_SUMMARY_END|>).

Given the protein above from organism Homo sapiens (Human) with the following InterPro annotations:

- IPR013106: Immunoglobulin V-set domain (domain) [26-115]
- IPR036179: Immunoglobulin-like domain superfamily (homologous_superfamily) [22-116]
- IPR007110: Immunoglobulin-like domain (domain) [20-120]
- IPR013783: Immunoglobulin-like fold (homologous_superfamily) [11-119]
- IPR050150: Immunoglobulin Variable Light Chain (family) [19-115]

And the following initial GO term speculations:

Molecular Function (MF): GO:0003674 (molecular function), GO:0005488 (binding), GO:0005515 (protein binding)

Biological Process (BP): GO:0008150 (biological process), GO:0032501 (multicellular organismal process), GO:0003008 (system process), GO:0003014 (renal system process), GO:0097205 (renal filtration), GO:0003094 (glomerular filtration)

Cellular Component (CC): GO:0005575 (cellular component), GO:0110165 (cellular anatomical entity), GO:0071944 (cell periphery), GO:0016020 (membrane), GO:0005576 (extracellular region), GO:0005886 (plasma membrane)

Reason about the function of the protein and focus more on its Molecular Function, Biological Process, and Cellular Component.<|im_end|>

<|im_start|>assistant

<think>

Step 1: The domain architecture begins with IPR013783 (Immunoglobulin-like fold, residues 11–119) and IPR036179 (Immunoglobulin-like domain superfamily, residues 22–116), which define a beta-sandwich Ig-like scaffold.

Step 2: Nested within are IPR007110 (Immunoglobulin-like domain, residues 20–120) and, more specifically, IPR013106 (Immunoglobulin V-set domain, residues 26–115).

...

Step 11: From this mechanistic base, interaction partners can be hypothesized.

</think>

<|GO_SUMMARY_START|> GO:0005886 GO:0070062 GO:0072562
<|GO_SUMMARY_END|><|im_end|>

B Experimental Setup

B.1 Base Models, Tokenization, and Input Representations

DNA experiments. The text backbones for DNA continued pre-training (CPT) and the downstream SFT/RL stages are Qwen3-1.7B and Qwen3-4B, loaded with their native tokenizers [75]. For CPT we train the language model alone on tokenized biological free-text (no DNA encoder is attached). For the post-CPT SFT and RL stages we couple the text backbone to a frozen Evo2-1B encoder via a trainable linear projection [38]; the DNA hidden state is prepended to the text embeddings. DNA sequences are clipped to a maximum length of 2048 nucleotides, with 1024 nucleotides retained on each flank around the variant locus.

RNA experiments. The RNA experiments follow the same overall setup as the DNA experiments, replacing the sequence encoder and biological input modality while keeping the same text backbones and tokenizer choices. The text backbones for RNA CPT, SFT, and RL are Qwen3-1.7B and Qwen3-4B, loaded with their native tokenizers [75]. As in the DNA setting, CPT is performed on tokenized biological free-text using the language model alone, without attaching the RNA encoder.

For the downstream SFT and RL stages, we couple the text backbone to a frozen TranscriptFormer encoder through a trainable linear projection [45]. Each target-identification example contains a natural-language disease and cell-type context, a five-gene candidate set, and TranscriptFormer representations for the candidate genes in normal and disease states [45, 73]. The projected RNA hidden states are prepended to the text-token embeddings before the prompt tokens, so that the language model conditions jointly on transcriptomic representations and the textual task description.

Protein experiments. The text backbones are Qwen3-1.7B and Qwen3-4B, each loaded with its native BPE tokenizer [75]. The padding token is aliased to the end-of-sequence token. SFT and GRPO share one user turn. It opens with a protein placeholder, followed by the task instruction and the biological context: the organism, InterPro domain annotations, and initial GO-term speculations. The instruction requests step-by-step reasoning with emphasis on molecular function, biological process, and cellular component, and asks the model to list every endorsed GO identifier between <|GO_SUMMARY_START|> and <|GO_SUMMARY_END|> [89]. The amino-acid sequence is supplied only through the placeholder. During SFT the assistant turn is left open, and the supervised continuation is a shortened reasoning trace inside <think>, followed by that GO summary. During GRPO the same user turn is followed by an empty thinking block, so the sampled completion is the answer after </think>.

For SFT and GRPO we use the BioReason-Pro protein-conditioned interface without the GO-graph encoder [3]. A frozen ESM3 small open encoder (esm3_sm_open_v1) embeds the amino-acid sequence [52]. We read the hidden states at transformer layer 37, one vector per residue together with the encoder's beginning- and end-of-sequence positions. A trainable projection—a linear layer, a GELU, and a second linear layer—maps these states into the text-embedding space, and the projected vectors replace the protein-placeholder embeddings. The protein encoder remains frozen. SFT

updates the projection and a LoRA adapter on the text model [93]. GRPO merges that adapter into the backbone and trains a new LoRA adapter together with the same projection.

B.2 Continued Pre-training (Mid-training) Setup

We mid-train the two Qwen3 backbones on the biology subset of FineFineWeb [77]. We use the first 200,000 documents for training and hold out the next 5,000 as a fixed evaluation set, yielding a 200K/5K train/eval split.

We do not perform additional benchmark-specific deduplication or decontamination of the FineWeb biology subset beyond the filtering already implicit in the source corpus [94, 77]. Our goal in CPT is to model a realistic domain-adaptation setting starting from publicly available pretrained LLMs, whose original pretraining corpora are not fully auditable and may already contain task-relevant biological text [59, 57, 64]. We therefore treat CPT as exposure to broad biological language rather than as a controlled from-scratch pretraining intervention. Importantly, the CPT corpus does not include our supervised reasoning traces, RL prompts, or any constructed train/test examples from the downstream benchmarks. The main leakage-sensitive comparisons in the paper are stage-wise and relative: all CPT and non-CPT models use the same downstream splits, and OOD evaluation is defined by held-out pathways, diseases, or species. We therefore interpret CPT results as measuring the effect of additional broad biological language adaptation under realistic pretrained-model conditions, not as evidence of strict benchmark decontamination.

Training uses the standard causal-LM next-token prediction loss. Inputs are tokenized with a maximum length of 1024 tokens. We optimize with AdamW under a cosine learning-rate schedule with 3% linear warm-up, weight decay 0.01, gradient clipping 1.0, and bf16 mixed precision [95]. Each run trains for one epoch with a per-device batch size of 1. The CPT hyperparameter sweep varies the learning rate over $1\times10^{-5}, 3\times10^{-4}$ and the gradient-accumulation steps over $64, 128$ for both backbones. We select the best checkpoint by validation loss.

B.3 Supervised Fine-Tuning Setup

DNA SFT. DNA SFT couples a frozen Evo2-1B encoder with either Qwen3-1.7B or Qwen3-4B (including their post-CPT variants; see §B.2) [38, 75]. We use DeepSpeed Stage 2 on a single GPU with bf16 precision, batch size 1, gradient accumulation 8, AdamW with $\eta = 5 \times 10^{-5}$ and weight decay 0.01, and the same warm-up-to-cosine schedule as the protein SFT (5% warm-up, decay floor $0.1\eta_{\max}$) [96, 95]. The maximum DNA sequence length is 2048 nucleotides and the maximum text length is 1024 tokens. The epoch sweep ranges over $\{1, 2, 4, 8, 16, 32\}$ for each of $\{$Qwen3-1.7B, Qwen3-4B, CPT-Qwen3-1.7B, CPT-Qwen3-4B$\}$, with the best CPT learning rate selected per backbone from §B.2. Random seed is 23.

RNA SFT. RNA SFT follows the same training recipe as DNA SFT, replacing the Evo2 DNA encoder with the frozen TranscriptFormer encoder and using the target-identification examples described in §B.1 [45, 73]. Each example provides a disease, cell type, five candidate genes, and TranscriptFormer embeddings for the corresponding normal and disease states [45]. The projected TranscriptFormer hidden states are prepended to the text-token embeddings, and the model is trained to generate the correct target gene.

We use DeepSpeed Stage 2 on a single GPU with bf16 precision, batch size 1, gradient accumulation 8, AdamW with $\eta = 5 \times 10^{-5}$ and weight decay 0.01, and 5% linear warm-up followed by cosine decay to $0.1\eta_{\max}$ [96, 95]. The maximum text length is 1024 tokens. The main epoch sweep ranges over $\{1, 2, 4, 8, 16, 32\}$ for each of Qwen3-1.7B and Qwen3-4B, using the full RNA training set [75]. The same sweep is repeated for the Gemma 4 E2B RNA backbone in the backbone ablation [76]. Random seed is 23.

Protein SFT. SFT pairs the frozen ESM3 small open encoder with the Qwen3-1.7B or Qwen3-4B text backbone (see §B.1) [52, 75]. LoRA adapts the query, key, value, output, and feed-forward projections of the text model ($r = 128$, $\alpha = 256$, dropout 0). The protein projection is trained in full, and the ESM3 encoder stays frozen.

Training uses one GPU, bf16 mixed precision, batch size 1, and gradient accumulation 10 (effective batch size 10). Optimization uses AdamW with $\eta_{\max} = 1\times10^{-4}$, weight decay 0.01, gradient clipping at 1.0, a 5% linear warm-up, and cosine decay to $0.1\eta_{\max}$[95]. The maximum text length is

$4{,}096$ tokens and the maximum protein length is $2{,}000$ residues. Attention uses SDPA, and gradient checkpointing is enabled. Scheduled milestone checkpoints are retained.

Both backbones are trained in two settings. The first uses $1{,}250$ prompts for 16 epochs; the seed sets the initialization and the shuffle, and checkpoints are saved after epochs $\{1, 2, 4, 8, 16\}$. The second uses one fixed $20{,}000$-example subset for a single epoch, in that subset's original order; the seed sets only the initialization, and checkpoints are saved after $\{1{,}250, 2{,}500, 5{,}000, 10{,}000, 20{,}000\}$ examples.

B.4 Reinforcement-Learning (GRPO) Setup

DNA RL. The DNA GRPO runs use a multimodal architecture wrapping Qwen3-1.7B/4B (or their CPT variants) with a frozen Evo2-1B encoder [5, 75, 38]. The text model receives a fresh LoRA adapter attached on top of the SFT-merged checkpoint (see §B.5); the DNA encoder is frozen and the DNA projection is trainable [93]. We use DeepSpeed Stage 2 with bf16 precision on a single GPU [96]. Generation uses 8 rollouts per prompt with max completion length 800, $T = 1$, top-$p = 0.95$, and top-$k = 20$. Optimization runs at $\eta = 1 \times 10^{-5}$ with a cosine schedule, 3% warm-up, per-device batch size 4, gradient accumulation 8, gradient checkpointing, and $\beta = 10^{-4}$ for the KL anchor [56]. The reward combines format-adherence and correctness components following [1]. The number of GRPO epochs is swept over $\{1, 2, 4, 8, 16\}$; each RL run selects the best SFT checkpoint by validation accuracy, merges the SFT LoRA into the base weights, and then attaches a fresh RL adapter.

RNA RL. RNA GRPO follows the same multimodal RL setup as DNA GRPO, replacing the Evo2-1B encoder with the frozen TranscriptFormer encoder and using the target-identification reward [5, 45, 73]. The policy wraps Qwen3-1.7B, Qwen3-4B, or the Gemma 4 E2B RNA backbone with the frozen TranscriptFormer encoder and a trainable projection layer [75, 76, 45]. As in the DNA setting, the SFT LoRA is first merged into the text backbone, after which a fresh RL LoRA adapter is attached for GRPO [93]. The TranscriptFormer encoder remains frozen throughout RL, while the projection layer and RL adapter are trainable.

For each prompt, the model receives the disease, cell type, five candidate genes, and projected TranscriptFormer representations for the corresponding normal and disease states [45, 73]. The reward contains a format-adherence term and a correctness term based on whether the final answer matches the held-out target gene. For the main Qwen RNA sweeps, GRPO starts from the strongest SFT checkpoints identified in the SFT sweep: Qwen3-1.7B-R-SFT4,1 and Qwen3-4B-R-SFT8,1. The number of GRPO epochs is swept over $\{1, 2, 4, 8, 16\}$, yielding model families of the form Qwen3-1.7B-R-SFT4,1-RL$\{1, 2, 4, 8, 16\}$,1 and Qwen3-4B-R-SFT8,1-RL$\{1, 2, 4, 8, 16\}$,1. For the backbone ablation, the same procedure is applied to Gemma 4 E2B-R-SFT4,1-RL$\{1, 2, 4, 8, 16\}$,1.

Unless otherwise stated, RNA GRPO uses the same optimization and generation hyperparameters as DNA GRPO: DeepSpeed Stage 2, bf16 precision, 8 rollouts per prompt, max completion length 800, $T = 1$, top-$p = 0.95$, top-$k = 20$, AdamW with $\eta = 1 \times 10^{-5}$, cosine decay with 3% warm-up, per-device batch size 4, gradient accumulation 8, gradient checkpointing, and KL coefficient $\beta = 10^{-4}$ [96, 95, 5]. The random seed is fixed to 23.

For evaluation, the generated final answer is normalized by case, whitespace, punctuation, and common gene-symbol formatting variants, and is counted correct only if it exactly matches the held-out target gene symbol. Following [1], the model gets additional rewards for following the expected output structure, i.e. a reasoning trace followed by a one-gene answer, and for staying below the token limit of 1024 tokens (conciseness reward).

Protein RL. We apply Group Relative Policy Optimization (GRPO) to the same protein-conditioned Qwen3-1.7B and Qwen3-4B policies used for SFT [5, 75]. Each run merges an SFT LoRA into the base model, loads that checkpoint's protein projection, and trains a fresh LoRA adapter ($r = 16$, $\alpha = 32$, dropout 0.05) together with the projection [93]. The ESM3 encoder stays frozen, and the prompt contains no GO-graph embeddings [52, 3]. Qwen3-1.7B is initialized from its epoch-16 SFT checkpoint and Qwen3-4B from its epoch-4 SFT checkpoint. Three RL seeds share that checkpoint for each model. A KL penalty with coefficient 10^{-4} anchors the policy to this initialization. Gradient checkpointing is enabled for the policy update and disabled during rollout generation.

During training, each step samples $g = 2$ completions for one prompt at temperature 1.0, top-$p = 0.95$, and top-$k = 20$, with at most 800 new tokens. The reward is the unweighted F1 between the `is_a` and `part_of` closures of the GO identifiers extracted from the completion and of the gold leaf identifiers [89]. The optimizer is AdamW with $\eta = 1 \times 10^{-5}$, weight decay 0.01, a 3% warm-up, cosine decay, gradient clipping at 1.0, and bf16 precision [95]. The prompt includes the organism, InterPro annotations, and initial GO-term speculations, and it does not include protein–protein interaction partners [74].

RL evaluation. Every saved epoch is evaluated on 128 in-distribution and 128 out-of-distribution examples. Generation is greedy and limited to 800 new tokens. We report leaf-entity F1, the exact overlap between extracted and gold leaf GO identifiers.

B.5 LoRA Configurations and Trainable-Parameter Choices

All LoRA adapters target the seven attention and MLP projection matrices per transformer block: q, k, v, o, gate, up, down projections, with Gaussian initialization and no bias [93].

DNA LoRA. Both the DNA SFT and RL stages use $r = 32$, $\alpha = 64$, dropout 0.05 on the same seven target modules. The Evo2 encoder is frozen; the DNA projection is trainable [38]. For RL, when the SFT and RL adapter ranks differ, we merge the SFT LoRA into the base weights and attach a fresh RL adapter at the requested rank [93].

RNA LoRA. The RNA SFT stage uses $r = 64$, $\alpha = 64$, dropout 0.05, while the RL stage uses $r = 16$, $\alpha = 64$, dropout 0.05 on the same seven target modules. The transcriptformer encoder is frozen; the RNA projection is trainable [45]. For RL, when the SFT and RL adapter ranks differ, we merge the SFT LoRA into the base weights and attach a fresh RL adapter at the requested rank [93].

Protein SFT LoRA. Rank $r = 128$, scaling $\alpha = 256$, dropout 0, on the attention and feed-forward projections. The ESM3 encoder is frozen and the GO-graph encoder is omitted. Only the LoRA adapter and the protein projection receive gradients [52, 93].

Protein RL warm-start. The SFT LoRA is merged into the Qwen3-1.7B or Qwen3-4B weights, and the SFT protein projection is loaded from the same checkpoint [93, 75]. GRPO then trains a fresh LoRA adapter ($r = 16$, $\alpha = 32$, dropout 0.05) together with that projection. The merged base model and the ESM3 encoder stay frozen [52, 5].

B.6 Hyperparameters, Context Windows, Sequence Lengths, and Optimization

Table 1 summarizes the configuration used to produce every reported result. All training uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$. All runs use bf16 mixed precision with flash attention where supported.

Table 1: Optimization and context-window settings per training stage. "warm→cos" denotes linear warm-up followed by cosine decay to $\eta_{\min}$.

Stage	Backbone(s)	$\eta_{\max}$	Schedule	Warm-up	Batch (eff.)	Seq. len.
DNA/RNA CPT	Qwen3-1.7B/4B	$\{10^{-5}, 3\times10^{-4}\}$	cosine	3%	$1\times\{64, 128\}$	1024
Protein SFT	Qwen3-1.7B/4B-Thinking	10^{-4}	warm→cos	5%	1×16	10K / 2K[†]
DNA SFT	Qwen3-1.7B/4B (+CPT)	5×10^{-5}	warm→cos	5%	1×8	1024 / 2048[‡]
RNA SFT	Qwen3-1.7B/4B, Gemma 4 E2B	5×10^{-5}	warm→cos	5%	1×8	1024[§]
Protein GRPO	Qwen3-1.7B/4B-Thinking	3×10^{-5}	constant	—	1×1, $g=2$	gen. 256
DNA GRPO	Qwen3-1.7B/4B (+CPT)	10^{-5}	cosine	3%	4×8, $g=8$	gen. 800
RNA GRPO	Qwen3-1.7B/4B, Gemma 4 E2B	10^{-5}	cosine	3%	4×8, $g=8$	gen. 1024

[†] Text / protein-residue length caps. [‡] Text / DNA length caps (truncated symmetrically around the variant locus). [§] Text length cap; TranscriptFormer embeddings are prepended separately.

Weight decay is 0.01 on every stage; gradient clipping is 1.0. The protein GRPO step uses a constant learning rate; we rely on KL regularization and early-stopping checkpoint selection to control late-stage drift.

Sweep grids.

- **DNA/ RNA CPT**: learning rate $\in \{10^{-5}, 3\times10^{-4}\}$, gradient accumulation $\in \{64, 128\}$, 1 epoch (8 runs total).

- **Protein SFT data sweep**: data fraction $\in \{1250, 2500, 5000, 10000, 20000\}$ data points at 1 epoch.
- **Protein RL epoch sweep**: epochs $\in \{1, 2, 4, 8\}$ at 1250 data points.
- **DNA/ RNA SFT/RL epoch sweep**: epochs $\in \{1, 2, 4, 8, 16, 32\}$ for SFT (capped at 8 for RL), across Qwen3-1.7B, Qwen3-4B, and their post-CPT variants.

B.7 Compute Resources and Training Budget

All training and evaluation are run on a GPU cluster using NVIDIA H100 (80 GB) and H200 (141 GB) GPUs. We use one GPU per run; multi-node training was *not* required for any reported result.

- **DNA/RNA CPT**: 1× H100, walltime up to 2 days per run. Eight runs $\approx$ 24 GPU-days.
- **Protein SFT**: 1× H100, walltime up to 2 days per run. Eight runs across the reported data and epoch sweeps.
- **Protein RL**: 1× H200, walltime up to 2 days per run with automatic checkpointing and requeueing. Each training run chains two GPU evaluation jobs, one for the ID split and one for the OOD split, with sampling at $T = 0.7$.
- **DNA/RNA SFT/RL**: 1× H100, walltime up to 2 days per run. SFT and RL are submitted as chained pairs per backbone, with RL automatically selecting the best SFT checkpoint by validation loss.

C Main-Results Tables

C.1 Results for Figure 2

Task	Model	Split	Base	SFT1	SFT2	SFT4	SFT8	SFT16	SFT32
DNA	Qwen3-1.7B	ID	0.435	0.565 ± 0.030	0.708 ± 0.029	0.814 ± 0.014	0.853 ± 0.021	0.894 ± 0.004	0.917 ± 0.014
DNA	Qwen3-1.7B	OOD	0.097	0.634 ± 0.048	0.839 ± 0.014	0.841 ± 0.007	0.855 ± 0.007	0.887 ± 0.011	0.881 ± 0.022
DNA	Qwen3-4B	ID	0.579	0.749 ± 0.020	0.836 ± 0.035	0.855 ± 0.012	0.896 ± 0.048	0.931 ± 0.012	0.931 ± 0.012
DNA	Qwen3-4B	OOD	0.186	0.855 ± 0.036	0.887 ± 0.056	0.876 ± 0.060	0.881 ± 0.053	0.906 ± 0.032	0.922 ± 0.038
RNA	Qwen3-1.7B	ID	0.181	0.569 ± 0.039	0.667 ± 0.118	0.736 ± 0.059	0.757 ± 0.069	0.785 ± 0.069	0.771 ± 0.088
RNA	Qwen3-1.7B	OOD	0.074	0.405 ± 0.076	0.507 ± 0.048	0.669 ± 0.010	0.591 ± 0.062	0.645 ± 0.005	0.649 ± 0.019
RNA	Qwen3-4B	ID	0.361	0.616 ± 0.116	0.736 ± 0.042	0.819 ± 0.024	0.806 ± 0.024	0.824 ± 0.035	0.843 ± 0.016
RNA	Qwen3-4B	OOD	0.324	0.421 ± 0.037	0.570 ± 0.010	0.691 ± 0.051	0.712 ± 0.030	0.689 ± 0.042	0.691 ± 0.035
Protein	Qwen3-1.7B	ID	0.077	0.151 ± 0.012	0.151 ± 0.011	0.181 ± 0.004	0.178 ± 0.003	0.185 ± 0.008	–
Protein	Qwen3-1.7B	OOD	0.083	0.157 ± 0.025	0.158 ± 0.018	0.184 ± 0.027	0.196 ± 0.006	0.202 ± 0.006	–
Protein	Qwen3-4B	ID	0.128	0.122 ± 0.008	0.154 ± 0.024	0.194 ± 0.017	0.193 ± 0.014	0.201 ± 0.012	–
Protein	Qwen3-4B	OOD	0.128	0.158 ± 0.003	0.164 ± 0.021	0.207 ± 0.013	0.212 ± 0.017	0.206 ± 0.017	–

Table 2: Numerical results corresponding to Figure 2. DNA and RNA are evaluated with accuracy; proteins are evaluated with F_1. Values are reported as mean $\pm$ standard deviation after supervised fine-tuning (SFT) for the indicated number of epochs. Protein experiments are evaluated up to 16 SFT epochs. Base model values at epoch 0 are based on a single evaluation run.

C.2 Results for Figure 3

Task	Model	Split	Base	1.25K	2.5K	5K	10K	20K
Protein	Qwen3-1.7B	ID	0.077	0.203 ± 0.014	0.231 ± 0.024	0.339 ± 0.011	0.317 ± 0.022	0.329 ± 0.006
Protein	Qwen3-1.7B	OOD	0.083	0.223 ± 0.008	0.248 ± 0.026	0.323 ± 0.010	0.325 ± 0.014	0.330 ± 0.008
Protein	Qwen3-4B	ID	0.128	0.252 ± 0.012	0.223 ± 0.013	0.413 ± 0.018	0.343 ± 0.026	0.370 ± 0.015
Protein	Qwen3-4B	OOD	0.128	0.268 ± 0.011	0.239 ± 0.010	0.434 ± 0.021	0.417 ± 0.026	0.404 ± 0.017

Table 3: Full numerical results corresponding to Figure 3 for protein function prediction under fixed-compute, variable-data supervised fine-tuning. All post-training runs use one SFT epoch while varying the number of training examples. Performance is evaluated using F_1 on the ID and OOD test splits. Base-model results are reported at 0K as individual values, while SFT results are reported as mean $\pm$ standard deviation for the indicated training dataset sizes.

C.3 Results for Figure 4

Task	Model	Split	RL1	RL2	RL4	RL8	RL16
DNA	Qwen3-1.7B	ID	0.878 ± 0.038	0.869 ± 0.032	0.883 ± 0.038	0.892 ± 0.035	0.871 ± 0.017
DNA	Qwen3-1.7B	OOD	0.922 ± 0.011	0.922 ± 0.014	0.938 ± 0.012	0.913 ± 0.041	0.894 ± 0.049
DNA	Qwen3-4B	ID	0.892 ± 0.034	0.869 ± 0.014	0.890 ± 0.038	0.883 ± 0.024	0.876 ± 0.020
DNA	Qwen3-4B	OOD	0.881 ± 0.028	0.851 ± 0.038	0.848 ± 0.012	0.841 ± 0.024	0.835 ± 0.020
RNA	Qwen3-1.7B	ID	0.833 ± 0.014	0.824 ± 0.021	0.843 ± 0.008	0.801 ± 0.021	0.833 ± 0.014
RNA	Qwen3-1.7B	OOD	0.610 ± 0.017	0.595 ± 0.031	0.640 ± 0.032	0.651 ± 0.026	0.678 ± 0.043
RNA	Qwen3-4B	ID	0.852 ± 0.008	0.852 ± 0.016	0.861 ± 0.014	0.870 ± 0.008	0.861 ± 0.024
RNA	Qwen3-4B	OOD	0.739 ± 0.010	0.741 ± 0.026	0.755 ± 0.020	0.764 ± 0.024	0.775 ± 0.030
Protein	Qwen3-1.7B	ID	0.138	0.187	0.209	0.230	–
Protein	Qwen3-1.7B	OOD	0.219	0.205	0.247	0.237	–
Protein	Qwen3-4B	ID	0.166	0.194	0.221	0.255	–
Protein	Qwen3-4B	OOD	0.224	0.190	0.237	0.237	–

Table 4: Full numerical results corresponding to Figure 4. DNA and RNA are evaluated with accuracy; proteins are evaluated with F_1. Columns report performance after the indicated number of reinforcement-learning (RL) epochs on the ID and OOD test splits. DNA and RNA results are reported as mean $\pm$ standard deviation. Protein RL was evaluated through 8 epochs.

C.4 Results for Figure 5

Task	Model	Split	Base	SFT	SFT+RL	CPT+SFT	CPT+SFT+RL
DNA	Qwen3-1.7B	ID	0.721	0.894	0.892	0.897	0.883
DNA	Qwen3-1.7B	OOD	0.651	0.887	0.913	0.897	0.952
DNA	Qwen3-4B	ID	0.798	0.931	0.876	0.959	0.938
DNA	Qwen3-4B	OOD	0.668	0.922	0.883	0.979	0.890
RNA	Qwen3-1.7B	ID	0.181	0.785	0.806	0.778	0.833
RNA	Qwen3-1.7B	OOD	0.074	0.645	0.669	0.682	0.699
RNA	Qwen3-4B	ID	0.361	0.843	0.856	0.826	0.903
RNA	Qwen3-4B	OOD	0.324	0.691	0.712	0.713	0.787

Table 5: Full numerical results corresponding to Figure 5 for the continued-pretraining (CPT) ablation. DNA and RNA are evaluated with accuracy on the ID and OOD test splits. The Base column reports the non-CPT backbone before task-specific post-training. SFT and SFT+RL report post-training configurations without CPT, while CPT+SFT and CPT+SFT+RL report the corresponding configurations initialized from the CPT-adapted backbone. All values are reported as means.

C.5 Results for Figure 6

Model	Stage	Split	Base	Epoch 1	Epoch 2	Epoch 4	Epoch 8	Epoch 16
Qwen3-8B	SFT	ID	0.214	0.700	0.829	0.843	0.886	0.843
Qwen3-8B	SFT	OOD	0.315	0.488	0.616	0.631	0.660	0.660
Qwen3-8B	RL	ID	–	0.857	0.871	0.871	0.886	–
Qwen3-8B	RL	OOD	–	0.714	0.729	0.734	0.773	–
Gemma4-E2B	SFT	ID	0.214	0.629	0.843	0.829	0.871	0.871
Gemma4-E2B	SFT	OOD	0.271	0.286	0.424	0.522	0.532	0.512
Gemma4-E2B	RL	ID	–	0.843	0.829	0.843	0.857	–
Gemma4-E2B	RL	OOD	–	0.606	0.640	0.635	0.655	–

Table 6: Full numerical results corresponding to Figure 6 for the Qwen3-8B and Gemma4-E2B RNA backbone ablation. Metrics are accuracy on the RNA ID and OOD evaluation splits. The SFT rows report the supervised fine-tuning epoch sweep, including the base model at epoch 0. The RL rows report the reinforcement-learning epoch sweep with RL evaluated through 8 epochs.

C.6 Results for Figure 8

8 epochs total (from base model)										
Task	Model	Split	SFT0/RL8	SFT1/RL7	SFT2/RL6	SFT3/RL5	SFT4/RL4	SFT5/RL3	SFT6/RL2	SFT7/RL1
RNA	Qwen3-1.7B	ID	0.813 ± 0.069	0.778 ± 0.020	0.815 ± 0.058	0.833 ± 0.042	0.857 ± 0.045	0.894 ± 0.045	0.880 ± 0.029	0.894 ± 0.049
RNA	Qwen3-1.7B	OOD	0.622 ± 0.010	0.662 ± 0.019	0.719 ± 0.017	0.716 ± 0.018	0.705 ± 0.010	0.673 ± 0.008	0.687 ± 0.041	0.730 ± 0.020
RNA	Gemma4-E2B	ID	0.764 ± 0.037	0.778 ± 0.042	0.773 ± 0.056	0.801 ± 0.021	0.829 ± 0.035	0.824 ± 0.032	0.815 ± 0.058	0.773 ± 0.016
RNA	Gemma4-E2B	OOD	0.642 ± 0.030	0.613 ± 0.014	0.590 ± 0.085	0.570 ± 0.052	0.565 ± 0.052	0.577 ± 0.045	0.536 ± 0.090	0.487 ± 0.055
16 epochs total (8-epoch SFT warm start)										
Task	Model	Split	SFT8/RL8	SFT9/RL7	SFT10/RL6	SFT11/RL5	SFT12/RL4	SFT13/RL3	SFT14/RL2	SFT15/RL1
RNA	Qwen3-1.7B	ID	0.917 ± 0.024	0.907 ± 0.035	0.898 ± 0.045	0.866 ± 0.077	0.884 ± 0.056	0.894 ± 0.089	0.875 ± 0.074	0.866 ± 0.077
RNA	Qwen3-1.7B	OOD	0.741 ± 0.021	0.694 ± 0.040	0.719 ± 0.022	0.750 ± 0.018	0.739 ± 0.032	0.719 ± 0.038	0.734 ± 0.020	0.701 ± 0.031
RNA	Gemma4-E2B	ID	0.778 ± 0.020	0.847 ± 0.020	0.861 ± 0.020	0.833 ± 0.039	0.806 ± 0.014	0.773 ± 0.042	0.708 ± 0.037	0.537 ± 0.053
RNA	Gemma4-E2B	OOD	0.578 ± 0.062	0.615 ± 0.029	0.564 ± 0.024	0.524 ± 0.005	0.559 ± 0.033	0.500 ± 0.030	0.518 ± 0.004	0.426 ± 0.031

Table 7: Full numerical results corresponding to Figure 8 for the fixed-budget SFT–RL allocation experiment on RNA. The total post-training budget is fixed to eight epoch-level passes, and columns indicate the allocation between supervised fine-tuning and reinforcement learning. Values are reported as mean $\pm$ standard deviation across seeds. The top panel starts from the base models; the bottom panel from an 8-epoch SFT warm start.

D Additional Experiments

D.1 Scaling Post-Training for Biological Non-Reasoning Tasks

The main experiments in this paper focus on biological reasoning tasks, where the model must integrate biological inputs with natural-language context and produce mechanistic or structured outputs. To test whether the same post-training trends also appear in a more conventional biological prediction setting, we additionally evaluate supervised fine-tuning on variant effect prediction (VEP), using the coding non-SNV benchmark introduced in [1]. Unlike the KEGG-derived pathway prediction task, which requires multi-step mechanistic inference over molecular networks, VEP-Non-SNV is primarily a classification-style task: given paired reference and variant DNA sequences together with gene and chromosome context, the model predicts whether a coding non-SNV is benign or pathogenic, and, when pathogenic, the associated disease phenotype.

The VEP-Non-SNV dataset is constructed from ClinVar coding non-SNVs, filtered to include nuclear-genome variants affecting at most 64 base pairs, with sufficient clinical review status and transcript matching to GRCh38.p14. The original benchmark in [1] uses stratified train/test partitioning to balance disease representation and augments each example with paraphrased prompts. This makes the task biologically meaningful, but less dependent on explicit chain-of-thought-style mechanistic reasoning than the pathway prediction benchmark. In our setting, this experiment therefore serves as a non-reasoning control for testing whether increasing SFT compute continues to improve performance monotonically. [1] describes this benchmark as containing 36,088 core non-SNV entries and defines the task as predicting benign versus pathogenic status, with conditional disease prediction for pathogenic variants.

SFT Epochs	1	2	4	8	16	32
Accuracy	0.7123	0.7789	0.7965	0.8246	**0.8316**	0.8105

Table 8: SFT epoch scaling on the VEP-Non-SNV task. The model improves with additional supervised fine-tuning up to 16 epochs, after which performance declines, suggesting that even non-reasoning biological prediction tasks can exhibit non-monotonic SFT scaling.

Table 8 reports SFT epoch scaling accuracy for Qwen3-1.7B on VEP-Non-SNV. Performance improves steadily from one to sixteen epochs, increasing from 0.7123 at one epoch to 0.8316 at sixteen epochs, before declining at thirty-two epochs. Thus, even in this less explicitly reasoning-oriented setting, scaling SFT is not strictly monotonic: moderate additional supervision improves performance, but excessive training begins to degrade the final metric. This mirrors the broader pattern observed in the main biological reasoning experiments, where SFT is a strong driver of task performance but can over-specialize when scaled too far.

D.2 DNA Epoch Allocation

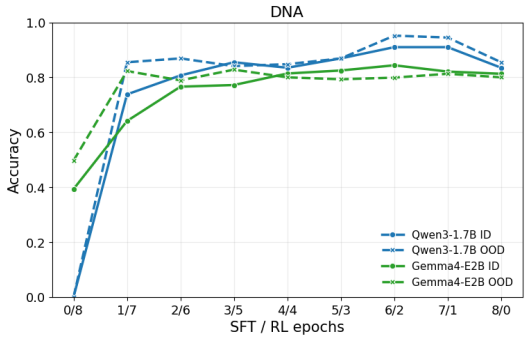

Figure 9: **Under a fixed post-training budget, enough SFT for task competence followed by RL gives a strong ID-OOD trade-off.** Shown are single-run results for pathway prediction (DNA) tasks with 8 epochs split up between SFT and RL for Gemma4 and Qwen3-1.7B. Training started from the base models (no warm start). RL training runs on the Qwen3-1.7B base model without any SFT collapsed.

D.3 Fixed Epoch, Variable Data during RL

Protein task variant. The experiments in this section and the next use a simplified, text-only variant of protein function prediction. Unlike the main protein experiments, which condition on ESM3 embeddings, this variant uses Qwen3-4B-Thinking directly with amino acid sequences, organism information, InterPro annotations, and initial GO-GPT predictions provided as text. GO graph embeddings and protein–protein interaction partners are omitted. GRPO optimizes an unweighted F1 reward computed after propagating predicted and reference Gene Ontology terms through the `is_a` and `part_of` hierarchy. We retain the species-based ID/OOD evaluation splits. Importantly, this propagated GO-F1 differs from the leaf-term F1 used in the main protein experiments, so absolute scores are not directly comparable.

In the main text, we demonstrated non-monotonic behavior when scaling RL *epochs* at fixed data size. A complementary question is whether scaling the *number of RL examples* at a fixed single epoch yields similar saturation. Table 9 reports protein function prediction performance (propagated GO-F1) for Qwen3-4B-Thinking trained with GRPO at $\beta = 10^{-3}$ (strong KL) for one epoch, varying the number of training examples from 4K to 20K.

Both ID and OOD F1 peak at an intermediate data budget (12K for ID, 4K for OOD) and decline as additional examples are added. The best OOD F1 (0.956 at 4K) exceeds the best ID F1 (0.952 at 12K). At the largest budget (20K), both metrics drop substantially below their respective peaks, with OOD F1 falling from 0.956 to 0.884. This seems to indicate that, under a fixed single-epoch schedule, additional RL data does not substitute for the exploration benefits of multi-epoch training and can instead over-constrain the policy. The result complements the epoch-scaling findings and

reinforces our broader conclusion that RL compute allocation, whether measured in epochs or data volume, requires careful calibration to maximize impact on biological reasoning capabilities.

Metric	4K	8K	12K	16K	20K
ID F1	0.935	0.931	**0.952**	0.901	0.877
OOD F1	**0.956**	0.954	0.938	0.921	0.884

Table 9: Protein function prediction F1 (Qwen3-4B-Thinking, GRPO $\beta = 10^{-3}$, 1 epoch) as a function of RL training examples. Increasing data does not monotonically improve performance; both ID and OOD F1 peak at intermediate budgets and decline with further scaling.

D.4 Fixed Budget, Variable Data Allocation between SFT and RL

We next ask whether the fixed-budget SFT–RL trade-off from Section 4.6 also appears when the total number of post-training examples is fixed and only the data allocation between SFT and RL is varied. Table 10 reports protein function prediction performance under a fixed 20K-example post-training budget. Each row allocates a different fraction of the data to SFT and RL while keeping the total number of examples constant.

SFT / RL data	SFT n	RL n	ID F1	OOD F1
0% / 100%	0	20000	0.7711	0.7953
20% / 80%	4000	16000	**0.9470**	**0.9685**
25% / 75%	5000	15000	0.9432	0.9620
40% / 60%	8000	12000	0.9167	0.9360
50% / 50%	10000	10000	0.9285	0.9569
60% / 40%	12000	8000	0.9055	0.9102
80% / 20%	16000	4000	0.8816	0.9065
100% / 0%	20000	0	0.2768	0.3177

Table 10: Fixed-budget data allocation between supervised fine-tuning and reinforcement learning. The total post-training data budget is fixed at 20K examples, and rows vary the fraction assigned to SFT versus RL. Metrics are reported as F1 on the ID and OOD protein function prediction splits.

The best results occur in the mixed SFT–RL regime rather than at either endpoint. Allocating a small fraction of examples to SFT and the majority to RL gives the strongest ID and OOD performance, with the 20%/80% split reaching 0.9470 ID F1 and 0.9685 OOD F1. Pure SFT performs poorly in this setting, while pure RL is substantially better but still far below the mixed allocations, indicating that RL benefits from a modest supervised warm start even when the total data budget is fixed.

D.5 Reward Model Ablations

Domain	Model	SFT init.	RL epoch	OOD Acc.	Format-valid %	Parseable final %	Format-only %	Empty/no-answer %	Mean tokens
RNA	Qwen3-1.7B	S8	0	0.567	100.0	69.6	43.2	0.0	201.1
RNA	Qwen3-1.7B	S4	0	0.547	100.0	79.1	45.3	0.0	210.5
RNA	Qwen3-1.7B	S4	1	0.500	100.0	100.0	50.0	0.0	568.2
RNA	Qwen3-1.7B	S4	2	0.655	100.0	100.0	34.5	0.0	170.9
RNA	Qwen3-1.7B	S4	4	0.669	100.0	100.0	33.1	0.0	419.6
RNA	Qwen3-1.7B	S4	8	0.642	100.0	100.0	35.8	0.0	544.6

Table 11: OOD reward-hacking audit across RNA RL checkpoints. Epoch 0 denotes the SFT initialization before RL. Format-only success is defined as a format-valid output with an incorrect final answer.

D.6 ID/ OOD Split Ablations

To test whether the qualitative training dynamics depend on how we define in-domain and out-of-domain settings, we provide an additional ablation in the RNA setting. Instead of splitting the data by held-out disease, we construct an alternative cell-type split for the target-identification task. We label an example as OOD if its canonical cell type is one of *regulatory T cell*, *exhausted T cell*, or *myeloid cell*, and use the remaining examples for training and validation. This yields 1,418 training examples,

75 validation examples, and 102 OOD test examples. The task format, model architecture, and evaluation protocol are otherwise unchanged from the main RNA experiments: the model receives the disease, cell type, five candidate genes, and aligned TranscriptFormer representations, and is evaluated by greedy generation with exact match against the target gene. This split therefore changes only the biological axis of distribution shift, from held-out disease to held-out cellular context, while preserving the same target-identification setup described earlier.

Stage	Checkpoint	ID val.	OOD test
SFT	1 epoch	54.7	60.8
SFT	2 epochs	54.7	43.1
SFT	4 epochs	58.7	39.2
SFT	8 epochs	62.7	41.2
RL	1 epoch	65.3	88.2
RL	2 epochs	68.0	92.2
RL	4 epochs	66.7	93.1
RL	8 epochs	73.3	95.1

Table 12: RNA target-identification performance under the held-out cell-type split. OOD examples are those whose canonical cell type is *regulatory T cell*, *exhausted T cell*, or *myeloid cell*. All values are exact-match accuracies, reported as percentages, under greedy generation.

Under this cell-type split, supervised fine-tuning again improves in-domain performance while failing to improve OOD generalization. Qwen3-1.7B reaches 54.7% ID accuracy after one epoch and increases to 62.7% by eight epochs, but OOD accuracy drops from 60.8% at one epoch to 39.2–43.1% across later SFT checkpoints. Starting GRPO from the four-epoch SFT checkpoint reverses this behavior: OOD accuracy rises from 39.2% at initialization to 88.2% after one RL epoch and continues improving to 95.1% after eight RL epochs, while ID accuracy also increases from 58.7% to 73.3%. This ablation supports the main conclusion that the SFT–RL contrast is not specific to the held-out hepatoblastoma disease split used in the main RNA experiments. Instead, the same qualitative pattern appears under a distinct biologically meaningful OOD definition: SFT fits the training/validation distribution, whereas RL substantially improves transfer to held-out cellular contexts.

E Asset Licenses and Redistribution Status

Table 13: Existing assets used in this work and their licenses or terms of use.

Asset	Used in this paper	License or terms of use
BIOREASON pathway prediction benchmark (`wanglab/kegg`)	Pathway prediction benchmark for DNA reasoning experiments	Apache-2.0, as listed on the Hugging Face dataset card.
MEDEA / MedeaDB target-identification task	RNA target-identification benchmark and disease/cell-type/candidate-gene setup.	The MEDEA code repository is Apache-2.0. The benchmark data are distributed separately through the mims-harvard/MedeaDB Hugging Face dataset (CC BY-NC-SA 4.0) referenced by the MEDEA README. We use the targetID task structure and TranscriptFormer setting described in MEDEA. We do not redistribute the raw MedeaDB data unless permitted by the MedeaDB dataset license/terms; any derived splits or preprocessing scripts will be released only under terms compatible with the upstream data license.
BioReason-Pro SFT reasoning data (`wanglab/bioreason-pro-sft-reasoning-data`)	Protein function reasoning benchmark/data	Apache-2.0, as listed on the Hugging Face dataset card.
FineFineWeb biology subset	Continued pre-training corpus	Open Data Commons Attribution License, ODC-By v1.0; use is also subject to CommonCrawl Terms of Use.
Qwen3-1.7B / Qwen3-4B	Text LLM backbones for DNA, RNA, and protein experiments	Apache-2.0, as listed for Qwen3 model files on Hugging Face.
Gemma 4 E2B	Backbone ablation for RNA experiments	Apache-2.0, as stated in the official Gemma 4 model card.
Evo2-1B	Frozen DNA encoder	Apache-2.0 for the Evo2 repository and released model artifacts.
TranscriptFormer	Frozen transcriptomic encoder	MIT License, as listed by the TranscriptFormer repository/model page.
ESM-3 / ESM3-1B protein encoder	Frozen protein encoder	Custom non-commercial model-weight terms for the ESM-3 Open Model checkpoint used, including EvolutionaryScale's Cambrian Non-Commercial License Agreement where applicable.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We outline our core claims in the abstract and introduction, specifically that biological reasoning does not improve monotonically with compute. We explicitly state our findings: CPT acts as a bridge to the domain, SFT yields strong in-domain gains but hurts OOD robustness, and RL can recover robustness but depends heavily on setup and the supervised starting point.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a separate paragraph discussing limitations in the discussion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: Our paper focuses entirely on the empirical training dynamics of large language models across genomics, transcriptomics, and proteins. There are no mathematical theorems, proofs, or formal theoretical assumptions to justify.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed information for this in both the main text and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code and launch scripts through an official repository linked in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed information for this in both the main text and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the computational resources required for our experiments, we currently do not perform multiple runs for all model configurations and tasks, although we do include them for SFT runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide detailed information for this in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: Our paper focuses on the training dynamics of reasoning models for the sciences. We do not see any immediate societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: In its present state, the paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the original sources for all datasets, models, and code assets used in this work, and Appendix E provides an asset-level license and redistribution table covering the benchmark datasets, continued-pretraining corpus, model backbones, and biological foundation models.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: In its current form, the paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our data relies strictly on molecular, genomic, and transcriptomic records. We did not use human subjects or crowdsourced annotators.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: Because our experimental loop does not involve human or animal subjects, IRB approval is not applicable to our computational study.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs were used in two ways. First, they were used for coding and writing assistance. Second, LLM-generated reasoning traces are part of the supervised training data used in our experiments: ChatGPT was used to generate reasoning traces for the protein and RNA tasks, and Claude 3.7 Sonnet was used to generate reasoning traces for the DNA pathway-prediction task. These traces were used only as training targets/prompts for post-training experiments and not as evaluators at test time.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.