CALIMEM: SEARCH-EFFICIENT DATA AGENT VIA CALIBRATED MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Data curation is a first-order design choice in machine learning, yet tailoring training data to a downstream model and task remains costly in labor, expertise, and compute. Datasets are therefore often processed through fixed pipelines and reused across applications. We show that this assumption is fragile: across 55 dataset–model tasks spanning biological domains, the same curation recipe can help one model while hurting another. Effective curation must adapt to the downstream setting, motivating automated search over data recipes. We first build an agentic system that searches for curation policies at deployment, improving data utility via *test-time curation (TTC)*. Although achieving performance gains, its search is driven largely by recent feedback and remains local. Adding persistent memory helps the agent retain evidence across iterations and reveals a deeper problem: it forms claims that extend beyond the conditions in which they were observed, sometimes abandoning an entire strategy after a single unsuccessful trial.

To address this problem, we introduce CALIMEM, a calibrated-memory agentic loop that requires the agent to predict the outcome of each proposed policy before executing it. Comparing this prediction with the observed result gives the agent a *reality check* on the claims behind its proposal. It uses the prediction error to retain supported claims, revise mistaken ones, and specify the conditions under which a conclusion holds. This makes the agent's beliefs testable, reduces overgeneralization, and supports more effective long-horizon exploration. Across biological and LLM post-training tasks, CALIMEM consistently outperforms fixed curation baselines and published methods, and exceeds full-data training by up to 19.4%. Code is open-sourced [1].

"However beautiful the strategy, you should occasionally look at the results" – Winston Churchill

1 INTRODUCTION

How a training set is constructed determines what a model learns and how efficiently it learns it. Data curation has accordingly grown beyond cleaning into a broad design space: removing redundant or low-value examples, selecting data for targeted capabilities, and setting the mixture of sources (Lee et al., 2022; Xie et al., 2023a; Xia et al., 2024; Wu et al., 2024; Chen et al., 2026). Tailoring these choices to a downstream model and task, however, demands expertise and repeated training runs. In practice, datasets are often processed through fixed pipelines and reused across applications. Such pipelines support consistency and reproducibility, but *a curation choice that benefits one downstream setting need not benefit another*.

This limitation is particularly evident in biological data, where data preparation commonly follows established workflows for single-cell transcriptomics (Luecken & Theis, 2019; Heumos et al., 2023; Luecken et al., 2022), protein variant-effect prediction (Notin et al., 2023), and non-coding variant-effect prediction (Marin et al., 2024). These workflows establish useful standards, but they do not establish which curation choices best serve a particular downstream application or model. We examine 55 dataset–model tasks across these 3 biological domains, comparing 12 standard curation recipes at a fixed data budget. The same recipe can help one model while hurting another. On one batch-integration dataset, for example, a recipe ranks 1st for cVAE (a variational autoencoder

[1] Code repository: https://anonymous.4open.science/r/CaliMem-450E

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

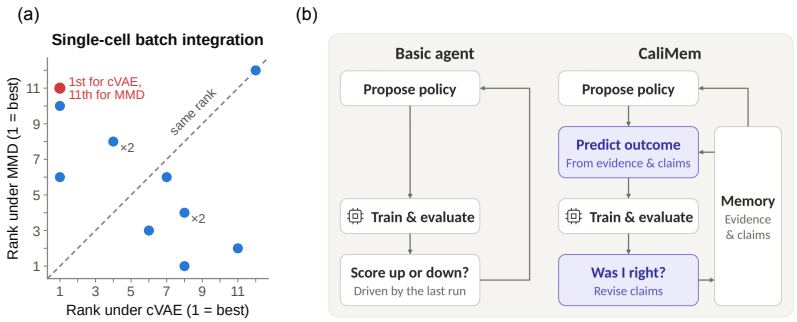


Figure 1: **Curation depends on the downstream model, and calibrated memory guides the search for a suitable policy. (a)** 12 standard recipes on a single-cell batch-integration dataset, ranked by the performance of two models trained on their half-size subsets. The recipe that ranks 1st for cVAE ranks 11th for MMD. **(b)** A basic agent proposes a policy, trains and evaluates the model, and responds to the latest score change. CALIMEM records evidence and claims in memory, predicts the outcome of each proposed policy, and compares its prediction with the result. The comparison guides revisions to the claims used in subsequent iterations.

that conditions its decoder on the batch label) but 11th for MMD-AE (an autoencoder trained with a distribution discrepancy penalty between batches (Figure 1a). Averaged across recipes, standard curation performs no better than selecting a random subset of the same size. *Effective curation therefore calls for searching for a policy suited to the downstream setting.* The potential reach of such search extends beyond biology: data selection and mixing also shape language-model pretraining, instruction tuning, and vision-language training (Xie et al., 2023a; Xia et al., 2024; Wu et al., 2024).

Generalist coding agents can carry out machine-learning experiments end to end (Chan et al., 2024; Karpathy, 2026; Rank et al., 2026), making data curation a natural target for automated search. Following the propose–evaluate–revise setting of Curation-Bench (Kang et al., 2026), we build an agent that searches for curation policies at deployment, improving data utility via *test-time curation (TTC)*. Given a candidate data pool, a selection budget, and a fixed training and evaluation pipeline, it proposes a policy, trains on the selected subset, observes the evaluation score, and decides what to try next. The agent improves data utility, but its trajectories are strongly driven by recent feedback. It often changes one policy setting at a time and uses the latest score change to choose its next move (e.g., Table 1). This local search can miss interactions among choices and underuse evidence accumulated earlier in the trajectory.

To help the agent learn from its full trajectory, we add persistent memory (Shinn et al., 2023). The memory records executed policies and their outcomes as evidence, alongside the agent's claims about what that evidence means. *Retaining these claims makes a deeper failure mode visible: the agent frequently states conclusions substantially beyond what its experiments warrant.* For example, adding class rebalancing to a confidence-filtered policy lowers performance, and a memory-only agent concludes that rebalancing does not help (Figure 2). The experiment establishes only that this particular combination failed under the tested conditions. It does not explicitly rule out rebalancing with a different filter, setting, or policy; though, the broad claim persists in the agent's belief essentially prevents the agent from revisiting the strategy.

We introduce CALIMEM, a calibrated-memory loop that subjects such claims to an empirical *reality check*. Before executing each proposed policy, the agent predicts its outcome from the evidence and claims in memory. After evaluation, it compares the prediction with the observed score. This gap reveals when the agent's understanding of a policy was mistaken, even if the policy itself improved performance. The agent then retains supported claims, revises mistaken ones, or limits them to the conditions under which they hold. In the rebalancing example, an unexpected loss directs attention to its interaction with the confidence filter rather than ruling out rebalancing altogether. The resulting claims are more carefully qualified and provide a stronger basis for long-horizon exploration.

We evaluate CALIMEM on the 55 biological dataset–model tasks and on LLM post-training tasks. It improves agentic search, outperforms fixed curation baselines and published methods, and, in the strongest setting, exceeds full-data training by up to 19.4%. **Our contributions are threefold:** (i) we show that standard curation recipes have sharply different effects across downstream models and tasks; (ii) we identify local feedback-driven search and overgeneralized memory claims as limitations of agentic curation; and (iii) we develop CALIMEM, which uses predictions and their errors to keep the agent's claims grounded in observed evidence.

2 Related Work

Data curation. Filtering and deduplication (Lee et al., 2022), target-aware sample selection (Wang et al., 2025a; Xia et al., 2024; Wu et al., 2024; Ivison et al., 2025), and optimization of data-source mixtures (Xie et al., 2023a; Ye et al., 2025; Liu et al., 2025a; Chen et al., 2025; 2026) all improve training data; see Albalak et al. (2024) for a survey. A recurring lesson is that there is no context-free "best" data: selection rankings change with pool, budget (Ivison et al., 2025), and compute (Muennighoff et al., 2023; Goyal et al., 2024). The same dependence appears in scientific workflows. Preprocessing choices in single-cell pipelines (e.g., normalization, feature selection) interact and shift downstream results (Vieth et al., 2019; Germain et al., 2020; Luecken et al., 2022). This dependence motivates *test-time curation (TTC)*: the agent searches the policy space against the downstream pipeline at deployment rather than committing to one recipe in advance.

Agents for machine-learning research and data work. Large language model (LLM) agents increasingly carry out iterative machine-learning experimentation, from bounded machine learning tasks to open-ended research engineering (Huang et al., 2025a; Chan et al., 2024; Wijk et al., 2025; Rank et al., 2026). Agents have also been applied to discovering dataset problems, data standardization, web collection, and general data-science workflows (Huang et al., 2025a; Thirukovalluru et al., 2025; Zhang et al., 2026a). Closest to us, Kang et al. (2026) place a generalist agent in a fixed train–evaluate loop for data refinement; Du et al. (2026) broaden autonomous data engineering with tree-structured search and global memory. We take the curation loop itself as given and focus on a complementary question: *what does an agent learn about the response of one downstream pipeline over a long search trajectory, and when should it trust what it has learned?*

Memory, self-improvement, and predictive agent state. Persistent-memory methods allow agents to accumulate reflections, experiences, and reusable strategies without changing model weights (Shinn et al., 2023; Zhao et al., 2024; Suzgun et al., 2026; Zhang et al., 2025b). Harness optimizers instead evolve prompts or workflows from execution feedback (Khattab et al., 2024; Yuksekgonul et al., 2024; Zhang et al., 2025a). Our scaffold is fixed; what evolves is the agent's beliefs about the target task. Predicting before acting connects to model-based agents (Gu et al., 2025; Gkountouras et al., 2025; Richens et al., 2025) and verbalized belief states (Singh et al., 2026). Unlike confidence calibration (Geng et al., 2024), the calibrated object is the set of empirical claims used to choose the agent's next intervention. Appendix A extends this discussion.

3 CaliMem: Calibrated Memory for Agentic Data Curation

We build CaliMem in three steps, each motivated by a failure observed in the previous one: a basic agent that implements the curation loop (Section 3.2), a persistent memory that lets it use its whole trajectory (Section 3.3), and a calibration step that checks the claims in that memory against measured outcomes (Section 3.4).

3.1 Problem formulation: the data curation loop

An automated data-curation agent operates through an iterative propose–evaluate–revise loop (Kang et al., 2026). The agent is given a candidate data pool $\mathcal{D}$, a selection budget B, and a fixed training and evaluation pipeline. At iteration t, it proposes a *data policy* π_t, a program that selects a subset of exactly B examples from $\mathcal{D}$ subject to task-specific validity constraints, such as a minimum number of examples per class or per batch (Appendix B.1). The downstream model is trained on this subset under the fixed pipeline and measured on the evaluation split, which produces the score $y_t = S(\pi_t)$. The agent observes y_t and proposes the next policy π_{t+1}. After t iterations, the trajectory

$$\mathcal{H}_t = \{(\pi_1, y_1), \ldots, (\pi_t, y_t)\} \tag{1}$$

holds all the empirical experience available to guide subsequent search. We write $\pi_t^\star$ for the *incumbent*, the best policy in $\mathcal{H}_t$, and $y_t^\star = \max_{i \le t} y_i$ for its score. In the default setup, the agent builds on the incumbent: a change is kept if it improves on $y^\star$ and reverted otherwise (as shown in Table 1). Every run starts from the same seeded random subset as π_1 and is allotted T iterations, each costing one full training run; the quantity of interest is $y_T^\star$.

Test-time curation. We refer to this search as *test-time curation* (TTC): rather than preparing the data once with a fixed recipe, compute is spent at deployment, by retraining the fixed downstream

model, to find the subset that serves that model best. The trade is particularly favorable in scientific domains such as biology, where new data come from costly experiments while many downstream models are relatively convenient to train (Appendix B.1). In this context, extracting more utility from an existing pool is then far cheaper than enlarging it. Because the pipeline is fixed, a difference in score between two iterations is attributable to the change in the data policy.

3.2 BASIC AGENT: LOCAL, REACTIVE SEARCH

We first implement the loop with a basic agent. It receives the goal, the budget, and the commands to submit a subset and to run training and evaluation, together with a minimal protocol that asks it to read its previous results before each move; which policies to try, and how to revise them, is left to the agent. Applied to test-time curation in both the biological and the LLM post-training domains, the basic agent already improves on the random starting subset and, in the biological domains, matches or exceeds the best human-designed recipe chosen separately for each subtask (Basic agent rows of Tables 12 and 13).

Its trajectories, however, are dominated by recent feedback. A curation policy consists of a selection *strategy*, such as allocating the budget across perturbations in proportion to their size, and the *parameters* that implement it, such as thresholds, counts, and weights. Once the agent has found a plausible strategy, it typically holds it fixed and spends its iterations adjusting one setting at a time.

Table 1: Excerpt from the trajectory of the basic agent on single-cell perturbation-response prediction (Norman19, GEARS); the complete log of all 15 iterations is in Table 10. Scores are evaluation cosine similarity between predicted and observed log fold changes ($\times 100$). A perturbation is the set of cells that received the same genetic perturbation, and control cells are unperturbed. *Keep* means the policy became the incumbent; *discard* means the agent reverted to the incumbent. Each row lists what the agent changed and a paraphrase of the note it wrote after seeing the score.

Iteration	Score	Status	What changed	Agent's note
1	61.30	keep	Random 50% subset (baseline)	The score to beat.
4	63.63	keep	Give each perturbation at least 100 cells (all of its cells if it has fewer); split the rest by perturbation size	New best: small perturbations now keep all their cells, and large ones keep their share.
5	63.83	keep	Control cells: ∼4,900 → 2,000	New best: cutting controls frees cells for perturbations at little cost.
6	**67.53**	keep	Control cells: 2,000 → 1,000	Large gain: more perturbation cells matter more than more controls.
7	66.60	discard	Control cells: 1,000 → 500	Worse: 500 controls are too few, so 1,000 is the lower limit. Next, keep 1,000 controls and try a higher per-perturbation minimum.
8	65.32	discard	Control cells back to 1,000; per-perturbation minimum: 100 → 150	Worse: a higher minimum takes cells from large perturbations, so 100 is the right minimum.

Local refinement. Table 1 illustrates a sequence of local changes within a curation strategy. The agent adjusts the number of control cells while retaining the allocation rule for perturbed cells, then changes the per-perturbation minimum after further reductions in controls stop improving the score.

To describe this type of move, write a policy as $\pi = \pi_{m,\theta}$, where m specifies the strategy and θ its numerical settings. A local proposal changes one coordinate of the incumbent: $\widetilde{\theta}_t = \theta^\star_{t-1} + \alpha_t s_t e_{j_t}$, $\widetilde{\pi}_t = \pi_{m^\star_{t-1},\widetilde{\theta}_t}$, where e_{j_t} is the basis vector for the selected *coordinate*, j_t is the selected setting, $s_t \in \{-1, +1\}$ its direction, and $\alpha_t > 0$ is a feasible step size. The resulting score is compared with the incumbent to determine whether the proposal is retained. This resembles directional derivative-free direct search ("zeroth-order coordinate optimization"), which makes decisions solely based on evaluations rather than an explicitly or implicitly modeled derivative (e.g., "gradient") (Conn et al., 2009; Jordana et al., 2025).

Local refinement is not itself a failure. The concern is that the agent may inaccurately interpret a result obtained at one configuration as a *conclusion* about settings or strategy variants that have not been tested. Such a conclusion can restrict later search even when the underlying mechanism remains useful under different conditions. (For example, after reducing the number of control cells from 1,000 to 500 hurts performance in Table 1, the agent concludes that 1,000 is a lower limit, even though the result only establishes this under the current perturbation-allocation rule.) This has two consequences. First, the search refines the current strategy rather than questioning it, and *settles into a local optimum of that strategy.* Second, a sign describes a move at one point of the policy space,

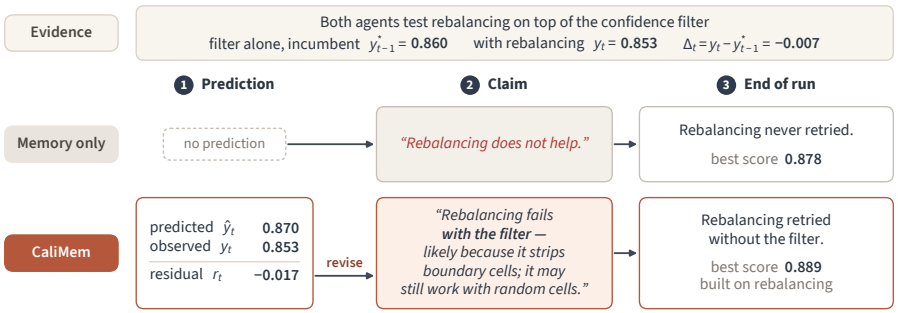


Figure 2: **Without a prediction, one failure closes a whole direction; with a prediction, the failure is tied to its condition.** Cell-type annotation (`pancreas_crosstech`, SVM; evaluation macro-F1). Both agents see rebalancing fail on top of a confidence filter (0.860 → 0.853). Top: the failure becomes a claim about rebalancing as a whole. Bottom: the gap between the predicted $+0.01$ and the observed -0.007 ties the failure to the filter, and rebalancing stays open. The top row follows the memory-only run; the bottom row adapts the memory + prediction run on the same task to the same move, and its best score is from that run.

yet the agent records what it learns as a general conclusion; yet whether an effect depends on the other settings, or reverses outside the range tested, cannot be read from a sign.

3.3 Holistic search via persistent memory

The basic agent's exploration is driven by its most recent result and overlooks patterns that its earlier iterations already reveal. Following work on persistent memory for LLM agents (Shinn et al., 2023; Zhang et al., 2025b), we give the agent an external memory to explicitly record what it has tried, what it has measured, what it has concluded, and which it consults before choosing each new policy.

Evidence and claims. The agent keeps a persistent memory $\mathcal{M}_t = (\mathcal{E}_t, \mathcal{C}_t)$, updated after every iteration and read in full before each decision. The *evidence* $\mathcal{E}_t$ records facts: the policies executed so far with their scores, $\{(\pi_i, y_i)\}_{i<t}$, and facts measured about the pool and the pipeline, such as class counts or batch sizes. The *claims* $\mathcal{C}_t$ record the conclusions the agent draws from this evidence, such as which settings interact, where an effect reverses, and which directions are spent, and are rewritten after each result, $\mathcal{C}_t \xrightarrow{(\pi_t, y_t)} \mathcal{C}_{t+1}$. The next policy is thus informed by the whole memory rather than only local feedback. Compared to the basic agent's log that records what was tried, the claims record what the agent currently believes it means. For example, in one cell type annotation task, the agent revisited the number of input genes across non-adjacent iterations, reducing it from the default 2,000 to 1,684 at iteration 5 and later increasing it to 2,191 at iteration 10. Because both deviations lowered the score, the memory consolidated these observations into the claim that 2,000 genes was already the best (Appendix C.3). Writing the claims down also makes the agent's beliefs inspectable, and reading them reveals a second problem. In memory, a conclusion becomes a claim that is read before every later decision, so an overbroad claim steers the rest of the run.

Figure 2 (left) illustrates how an **overgeneralized conclusion** can restrict subsequent exploration: the memory accurately records the performance decrease, but the agent interprets this result as evidence that rebalancing is generally ineffective. This conclusion extends beyond the tested configuration. Once retained in memory, it guides subsequent decisions and prevents the agent from revisiting rebalancing, potentially excluding configurations that could improve performance.

3.4 Calibrated memory: predict, then calibrate

To reduce overgeneralization, CaliMem requires the agent to predict each proposed policy's score from the evidence and claims in memory before evaluation. After evaluation, it compares the predicted and observed scores and uses the discrepancy to reassess the supporting claims. The intuition is that a performance change indicates whether a policy helped, while a prediction error indicates whether its effect matched the agent's expectations. The agent revises the claims accordingly and uses the updated memory to guide subsequent search.

1. Predict before evaluation. Before executing policy π_t, the agent records a point prediction $\widehat{y}_t$, the evidence and claims supporting it, and an outcome that would challenge those claims. This record is fixed before evaluation: $\widehat{y}_t = \widehat{S}(\pi_t \mid E_t, C_t)$.

2. Two diagnostics from one evaluation: After observing y_t, the agent records *(i) the signed prediction residual (i.e., prediction error)*: $r_t = y_t - \widehat{y}_t$. A positive residual indicates that the outcome exceeded the prediction; a negative residual indicates that the prediction was too optimistic. We also record *(ii) the observed changes* relative to the incumbent: $\Delta_t = y_t - y^\star_{t-1}$. The diagnostics answer different questions. Δ_t determines the keep-or-revert decision, subject to the validity checks. The residual r_t measures the discrepancy between the expected and observed score. Their incumbent-relative directions show whether the proposal crossed the improvement boundary as expected. A policy can therefore become the new incumbent while still producing a substantial prediction miss.

3. Revising the claims. The agent uses the recorded prediction, outcome, diagnostics, and supporting evidence to rewrite its claims in natural language. The interpretation of residual magnitude is left to the agent: Agreement with a prediction is evidence about the tested configuration, not independent validation of every claim used to justify it. Conversely, a discrepancy prompts reassessment but does not by itself identify the cause: the agent must consider the executed policy, the available observations, and possible variability in the measured outcome.

As a result, with CALIMEM, the claims in memory carry the conditions under which they were observed, as in Figure 2 (right), and a failed variant narrows a mechanism instead of closing it.

4 EMPIRICAL STUDIES

How does CALIMEM *compare with fixed curation methods and basic agents in real tasks?* In this section, we investigate the empirical effectiveness of agentic test-time curation with CALIMEM, covering diverse scenarios from computational biology tasks (science domains) to prevalent machine learning tasks, such as post-training for large language models (LLMs).

Experiment Setup. The **science experiments** cover 3 biological domains, each with its own data type: protein sequence and functional data, genomics, and single-cell transcriptomics. We tested 5 tasks common in these domains, including protein variant effect prediction, non-coding variant effect prediction, cell type annotation, batch integration and perturbation response prediction, giving **55 dataset–model tasks** in total. In each task, the agent selects 50% of the pool. **LLM post-training** experiments include *general instruction tuning*, selecting 10,000 rows from the Tülu 3 dataset (Lambert et al., 2024) or LESS mixture (Xia et al., 2024) to fine-tune OLMo-2-1B or Llama-2-7B, and *code instruction tuning*, selecting 10,000 rows from OpenCodeInstruct (Ahmad et al., 2025) to fine-tune Qwen2.5-Coder-1.5B. Every task is run under two agentic pipelines, *Basic agent* and CALIMEM. We test 3 agents with both agentic pipelines: Claude Code driven by Opus 4.8, Sonnet 5, or Haiku 4.5; Codex driven by GPT-5.3-Codex or GPT-5.6 Sol; and OpenHands driven by Qwen3.5-397B-A17B, an open-weight model. The agents are initiated from the same-seeded random subset, counted as the first iteration, and run 15 iterations in biology and 10 in LLM post-training track. Each task is scored by several metrics following major benchmarks. Performance is measured by a fixed scorer on a **held-out** test split *disjoint* from training/validation splits; the agent observes only the resulting metrics. Additional details are provided in Appendix B.

4.1 SCIENCE EXPERIMENTS: BIOLOGY TASKS

Figure 3 reports results across the 5 biological tasks. **Fixed curation baselines:** For each biological domain, we instantiate 12 literature-derived curation recipes. The recipes span the major families of interventions commonly used for that data type—including quality filtering, balancing and stratification, coverage-based selection, and difficulty-based selection, etc. Appendix B.2 gives its source and exact implementation. These baselines represent a diverse set of reasonable human-designed choices available a priori, before observing how a particular downstream model responds. Figure 8 shows the detailed comparison task by task.

Fixed curation is not reliably beneficial. Averaged over the 12 recipes, fixed curation underperforms full-data training in five biological tasks with a full-data reference, by 0.8–8.3%, and even an oracle choosing the best recipe per dataset–model task improves on full data only modestly. The dependence is one of sign, not just magnitude: class balancing is redundant or harmful when the classifier already uses inverse-frequency weights, and coverage-based selection over-samples sparse, noisy regions unless unreliable examples are filtered first. A heuristic that corrects one model's weakness can remove useful signal for another.

Test-time curation exploits this dependence. The Basic agent searches directly against the target pipeline and exceeds both Random and the fixed-recipe alternatives across agent families. In all 24 reported Basic-agent results with a full-data reference, the selected 50% subset also outperforms training on the entire pool.

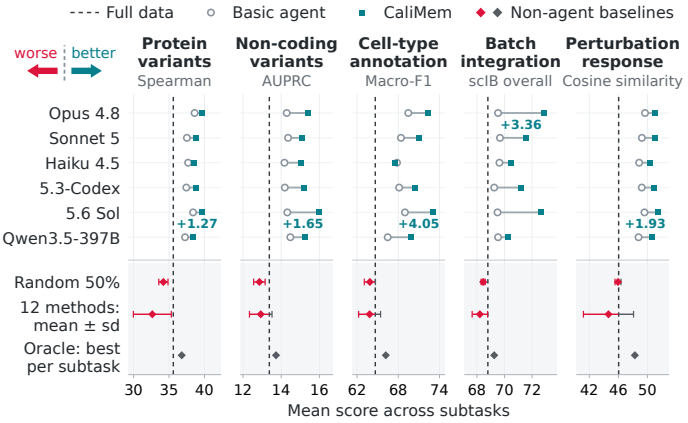


Figure 3: **Main results on the biology domains.** Each entry is the mean, over the dataset–model pairs of the domain, of the best headline score reached within the 15-iteration budget, scaled to 0–100. Random is the seeded subset every run starts from, Full data trains on the entire pool without curation, and the fixed-pipeline rows are standard curation recipes of each domain: the mean ± sd over the 12 recipes, and the best recipe chosen separately for every subtask. Detailed results can be found in Appendix D.

CaliMem further improves agentic search. Across the 30 Basic-agent/CaliMem pairs in Table 12, CaliMem improves 29, spanning protein, genomic, and single-cell tasks and both proprietary and open-weight agents. All 30 CaliMem results with full-data references remain above full-data training despite using only half of the pool. By keeping conclusions tied to the conditions that produced them, CaliMem spends the fixed budget exploring alternative policies rather than committing early to a local one.

4.2 LLM Post-Training Experiments: General and Code Instruction Tuning

We next evaluate substantially more aggressive curation regimes in LLM post-training, retaining only 1.1%, 3.7%, or 0.2% of the respective pools. Results are shown in Figure 4.

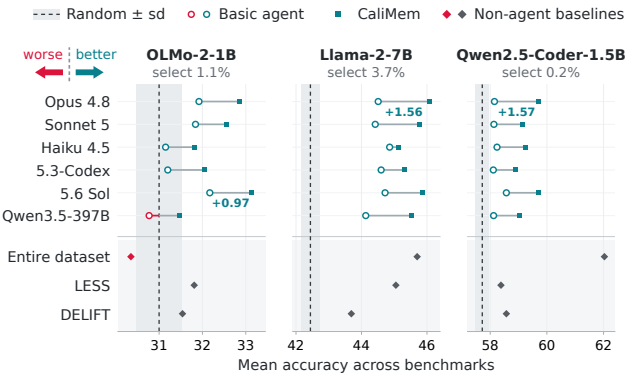


Figure 4: **Main results on LLM post-training tasks.** Each entry is the best mean accuracy (%) over the setup's benchmark suite reached within the 10-iteration budget. Random, LESS (Xia et al., 2024), DELIFT (Agarwal et al., 2025), and the agents all select a subset of the same size from the same pool (the selection ratio is given under each backbone): the dashed line and band mark Random as the mean ± sd over seeds, LESS and DELIFT are human-designed selection methods, and Entire dataset trains on the entire pool without curation. Points and segments left of the dashed line fall below Random. Detailed results can be found in Appendix D.

Agentic search remains effective at very small data budgets. The Basic agent improves over Random in 17 of 18 agent–task combinations, and CaliMem improves over the corresponding Basic agent in all 18. On the two general instruction-tuning setups, the strongest curated subsets even exceed full-data training while using only 1.1% or 3.7% of the data.

CaliMem is also competitive with the task-specific selection method such as LESS, matching or outperforming it in 17 of 18 comparisons. Notably, LESS needs warm-up training and per-example gradients over the pool while CaliMem uses only the downstream train-and-evaluate interface. The discovered policies go beyond filtering heuristics: on Llama-2-7B one agent adapted RDS+ so that source allocation and within-source similarity ranking are decided separately (Appendix F.2).

Figure 5 shows the best score reached so far as the agent spends its iteration budget. Across the various settings shown, CaliMem consistently establishes stronger incumbents earlier and finishes with a higher score. Calibrated memory improves the efficiency of the search itself, allowing evidence from earlier interventions to shape subsequent experiments more effectively.

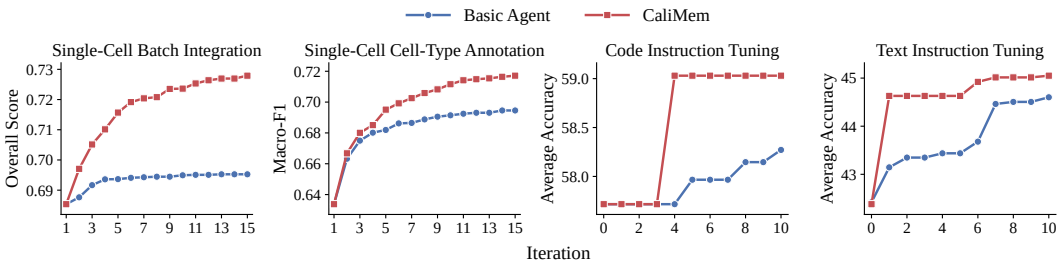


Figure 5: **Best score so far per iteration under the basic agent and CALIMEM** (Claude Opus 4.8). Left two panels: single-cell batch integration and cell-type annotation, averaged over the 15 dataset–model tasks of each domain, 15 iterations. Right two panels: code instruction tuning (Qwen2.5-Coder-1.5B) and instruction tuning (Llama-2-7B on the LESS mixture), 10 iterations. In every panel CALIMEM passes the basic agent's final score within the first 5 iterations.

Takeaway: CALIMEM is more search-efficient as well as more accurate: it reaches the basic agent's final score in a third or fewer of the iterations, and on the biology tasks it keeps improving after the basic agent has plateaued.

5 UNDERSTANDING CALIMEM'S SEARCH BEHAVIOR

The main results show that CALIMEM finds stronger policies with the same experimental budget. We next investigate why. First, we test whether its calibrated memory generalizes beyond summarizing past experiments by asking it to predict the outcomes of curation interventions it has not yet seen (Section 5.1). Second, we ablate persistent memory and prospective prediction to isolate their contributions (Section 5.2). Finally, we compare the adaptive policies found by the agent with fixed biological curation recipes to understand where adaptation provides its advantage (Section 5.3).

5.1 DOES THE MEMORY PREDICT UNSEEN CURATION INTERVENTIONS?

We test whether CALIMEM's memory captures reusable structure rather than merely summarizing past experiments. After iteration k, we freeze the memory and give it to a fresh agent, which predicts the scores of policies executed later in the original trajectory without seeing their outcomes. For each later policy π_j, $\hat{y}_{j|k} = \hat{S}(\pi_j \mid M_k)$, $e_{j|k} = |y_j - \hat{y}_{j|k}|$. We vary k across the trajectory and average forecasting error across dataset–model tasks (prompt in Appendix E.2). Figure 9 shows the results for cell-type annotation. Along a row, error grows with extrapolation distance; down a column, the same target iteration becomes easier to predict as the memory is fitted on a longer prefix. The other domains show the same two patterns (Appendix E).

Finding: CALIMEM remembers beyond which policies succeeded. As the trajectory grows, its memory becomes increasingly predictive of unseen interventions, suggesting that the calibrated claims encode reusable structure about the task's response to data curation.

5.2 ARE MEMORY AND PREDICTION BOTH NECESSARY?

We separately ablate persistent memory and prospective prediction on protein variant-effect prediction and single-cell batch integration. Memory alone improves the Basic agent by 0.44 and 1.71 points, while prediction alone changes performance by -0.09 and $+0.54$. Combining them performs best in both settings, improving a further 0.54 and 1.65 points over memory alone. We hypothesize that comparing predicted and observed scores helps the agent reassess the claims used to justify a policy, while persistent memory preserves the resulting revisions for subsequent decisions. Without explicit memory updates, the agent may recognize an unexpected outcome without consistently revising the assumptions used in later iterations.

Finding: Persistent memory provides a compact, reusable summary of evidence and claims across iterations; prediction provides a prospective check on those claims. Memory alone improves over the Basic agent in both ablation settings, whereas prediction alone has a smaller and inconsistent effect. Combining the two performs best in both settings, consistent with prediction being most useful when its outcome can revise a persistent state that guides subsequent decisions.

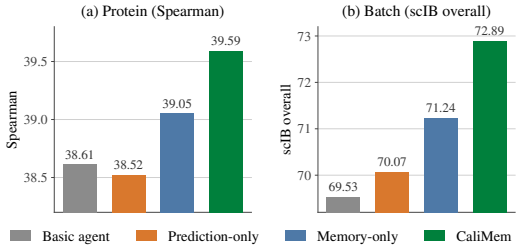


Figure 6: **Ablation of calibrated memory.** Results for Claude Opus 4.8 on protein variant-effect prediction and single-cell batch integration. Prediction-only adds a prospective prediction and comparison without persistent claims; Memory-only maintains persistent evidence and claims but does not require a committed prediction before each experiment. The complete CALIMEM combines both.

5.3 COMPARISON WITH FIXED BIOLOGICAL CURATION PIPELINES

Figure 8 breaks the fixed-recipe comparison down to individual recipes. The same classes of intervention (quality filtering, balancing, stratification, coverage, difficulty-based selection) help in one setting and hurt in another, so no recipe in our set transfers uniformly across datasets and models. In aggregate (Table 12), the mean recipe is below full data in most domain, and even a per-task oracle closes only a small part of the gap. Test-time curation searches beyond this fixed menu: the agent alters thresholds and allocations, combines mechanisms, changes representations, or writes new policies in response to the measured pipeline. Notably, many of its strongest policies build on the same heuristics (Appendix B.2). Their value is conditional on the downstream model and on interactions with the rest of the policy–the search supplies the adaptation a fixed recipe cannot.

Finding: The limitation of fixed curation is not a lack of useful heuristics, but a lack of adaptation. Standard biological pipelines contain valuable prior knowledge, yet their effects vary across downstream models. Test-time curation turns those heuristics into a searchable policy space, and CALIMEM uses calibrated memory to explore that space more reliably.

6 CONCLUSION

Data curation is usually done once, with a fixed recipe, and reused. Across 55 dataset–model tasks in biological domains, we found this practice non-ideal: the same recipe helps one frozen model and hurts another, and averaged over recipes it is no better than a random subset of the same size. We therefore treated curation as agentic search carried out at deployment against the fixed downstream pipeline, test-time curation. A basic agent already improves on every fixed baseline. Adding persistent memory lets it use its whole trajectory and exposes a deeper failure: claims that reach beyond their evidence and steer the rest of the run. CALIMEM makes every claim answerable by requiring a prediction before each experiment and writing the prediction error back into the claims. The resulting agent substantially improves on the basic agent in 47 of 48 models, exceeds full-data training by up to 19.4%, and reaches the basic agent's final score in a third of the iterations or fewer.

Limitations and future work. *1. Noise in single-run feedback.* Each iteration is a single training run, so the score the agent reacts to, and the prediction error it learns from, carry run-to-run noise. *2. Cost of test-time curation.* Test-time curation exchanges computation for data utility. Our budgets are practical in the settings studied here, but they would be expensive for substantially larger models or training recipes. Cheaper proxy models, early stopping, learned surrogates, and adaptive allocation of the budget could all reduce the cost. A natural extension is an agent that predicts and decides *which experiments are worth paying for. 3. From curation policies to scientific hypotheses.* Connecting the agent's empirical claims to explicit hypothesis testing, uncertainty-aware experimental design, and review by domain experts is future work, and it matters most where curation decisions guide the acquisition of new experimental data.

AI USE STATEMENT

Generative AI is both the object of this study and a tool used in producing it, and we separate the two. As the object of study, LLM-driven coding agents (Claude Code, Codex, and OpenHands, driven by the models listed in Section 4) proposed, implemented, and revised every curation policy reported in this paper, under the fixed harness of Appendix B.1. We have not used generative AI to formulate or prove mathematical claims or to propose the research hypotheses. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All datasets used in this work are public research benchmarks used under their released terms: the deep mutational scanning assays of ProteinGym, the non-coding variant sets distributed by BEND (ClinVar, an IGVF reporter assay, and GRASP against 1000 Genomes), the scIB single-cell atlases, the PerturBench screens, and the Tülu 3, LESS, and OpenCodeInstruct instruction-tuning pools. The human-derived biological data are de-identified aggregate or cell-level measurements released for research; we collected no new data and conducted no human-subject studies. Automated data curation is a general-purpose capability that, like any tool for improving training data, could be applied to models built for harmful ends; we see no risk specific to this work beyond that of the underlying models and datasets.

REPRODUCIBILITY STATEMENT

We release the code, the evaluation harness, agent protocol files (Basic and CALIMEM), implementation, and example trajectory logs of the runs reported in the paper. Appendix B.1 specifies each domain's datasets, held-out splits, composition floors, frozen models, metrics, and validity guards; Appendix B.3 the fine-tuning hyperparameters of the three instruction-tuning setups; and Appendix E.2 the forecasting prompt. Every run starts from the same seeded random subset, the frozen models train with fixed seeds, the code setup uses deterministic kernels, and each human-designed recipe is materialised with 5 fill seeds. The agent runs themselves are stochastic and several of the driving models are proprietary and may change; we report the model identifiers.

REFERENCES

Tamim Abdelaal, Lieke Michielsen, Davy Cats, Dylan Hoogduin, Hailiang Mei, Marcel J. T. Reinders, and Ahmed Mahfouz. A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biology*, 20(1):194, 2019. doi: 10.1186/s13059-019-1795-z.

Ishika Agarwal, Krishnateja Killamsetty, Lucian Popa, and Marina Danilevsky. Delift: Data efficient language model instruction fine-tuning. In *International Conference on Learning Representations*, volume 2025, pp. 100488–100509, 2025.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2507.19457.

Wasi Uddin Ahmad, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Vahid Noroozi, Somshubra Majumdar, and Boris Ginsburg. Opencodeinstruct: A large-scale instruction tuning dataset for code llms. *arXiv preprint arXiv:2504.04030*, 2025.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

Matthew Amodio, David van Dijk, Krishnan Srinivasan, William S. Chen, Hussein Mohsen, Kevin R. Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, Anita Desai, V. Ravi, Priti Kumar, Ruth Montgomery, Guy Wolf, and Smita Krishnaswamy. Exploring single-cell data with deep multitasking neural networks. *Nature Methods*, 16(11):1139–1145, 2019. doi: 10.1038/s41592-019-0576-7.

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000. doi: 10.1038/75556.

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling optimal LR across token horizons. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2409.19913.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

Carla E. Brodley and Mark A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999. doi: 10.1613/jair.606.

Ziyang Cai, Amir Saeidi, and Harkirat Behl. ML-AutoResearch: Training machine learning research agents with automatically generated environments. *arXiv preprint arXiv:2603.17216*, 2026.

Hyungjoo Chae, Namyoung Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. MLE-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2024.

Yurui Chang, Yongkang Du, Yuanpu Cao, Jinghui Chen, and Lu Lin. ForecastCompass: Guiding agentic forecasting with adaptive factor memory. *arXiv preprint arXiv:2605.30858*, 2026.

Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, Yaliang Li, Bolin Ding, and Jingren Zhou. Data-Juicer: A one-stop data processing system for large language models. In *Companion of the 2024 International Conference on Management of Data (SIGMOD/PODS '24)*, pp. 120–134. ACM, 2024. doi: 10.1145/3626246.3653385.

Mayee Chen, Michael Hu, Nicholas Lourie, Kyunghyun Cho, and Christopher Ré. Aioli: A unified optimization framework for language model data mixing. In *International Conference on Learning Representations*, volume 2025, pp. 46089–46132, 2025.

Mayee F. Chen, Tyler Murray, David Heineman, Matt Jordan, Hannaneh Hajishirzi, Christopher Ré, Luca Soldaini, and Kyle Lo. Olmix: A framework for data mixing throughout LM development. *arXiv preprint arXiv:2602.12237*, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. FLIP: Benchmark tasks in fitness landscape inference for proteins. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/2b44928ae11fb9384c4cf38708677c48-Abstract-round2.html.

Benjamin DeMeo and Bonnie Berger. Hopper: a mathematically optimal algorithm for sketching biological data. *Bioinformatics*, 36(Supplement_1):i236–i241, 2020. doi: 10.1093/bioinformatics/btaa408.

Jiarui Ding, Xian Adiconis, Sean K. Simmons, Monika S. Kowalczyk, Cynthia C. Hession, Nemanja D. Marjanovic, Travis K. Hughes, Marc H. Wadsworth, Tyler Burks, Lan T. Nguyen, John Y. H. Kwon, Boaz Barak, William Ge, Amanda J. Kedaigle, Shaina Carroll, Shuqiang Li, Nir Hacohen, Orit Rozenblatt-Rosen, Alex K. Shalek, Alexandra-Chloé Villani, Aviv Regev, and Joshua Z. Levin. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nature Biotechnology*, 38(6):737–746, 2020. doi: 10.1038/s41587-020-0465-8.

C. Domínguez Conde, C. Xu, L. B. Jarvis, D. B. Rainbow, S. B. Wells, T. Gomes, S. K. Howlett, O. Suchanek, K. Polanski, H. W. King, L. Mamanova, N. Huang, P. A. Szabo, L. Richardson, L. Bolt, E. S. Fasouli, K. T. Mahbubani, M. Prete, L. Tuck, N. Richoz, Z. K. Tuong, L. Campos, H. S. Mousa, E. J. Needham, S. Pritchard, T. Li, R. Elmentaite, J. Park, E. Rahmani, D. Chen, D. K. Menon, O. A. Bayraktar, L. K. James, K. B. Meyer, N. Yosef, M. R. Clatworthy, P. A. Sims, D. L. Farber, K. Saeb-Parsy, J. L. Jones, and S. A. Teichmann. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376(6594):eabl5197, 2022. doi: 10.1126/science.abl5197.

Yaxin Du, Xiyuan Yang, Zhifan Zhou, Wanxu Liu, Zixing Lei, Zimeng Chen, Fenyi Liu, Haotian Wu, Yuzhu Cai, Zexi Liu, et al. Datamaster: Data-centric autonomous ai research. *arXiv preprint arXiv:2605.10906*, 2026.

Andre J. Faure, Jörn M. Schmiedel, Pablo Baeza-Centurion, and Ben Lehner. DiMSum: an error model and pipeline for analyzing deep mutational scanning data and diagnosing common experimental pathologies. *Genome Biology*, 21(1):207, 2020. doi: 10.1186/s13059-020-02091-3.

Douglas M. Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nature Methods*, 11(8):801–807, 2014. doi: 10.1038/nmeth.3027.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, et al. DataComp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.

Kanishk Gandhi, Michael Y. Li, Lyle Goodyear, Agam Bhatia, Louise Li, Aditi Bhaskar, Mohammed Zaman, and Noah D. Goodman. BoxingGym: Benchmarking progress in automated experimental design and model discovery. *arXiv preprint arXiv:2501.01540*, 2025. Also presented at the NeurIPS 2025 Workshop on Scaling Environments for Agents.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research*, 2026. arXiv:2507.21046.

Sam Gelman, Sarah A. Fahlberg, Pete Heinzelman, Philip A. Romero, and Anthony Gitter. Neural networks to learn protein sequence–function relationships from deep mutational scanning data. *Proceedings of the National Academy of Sciences*, 118(48):e2104878118, 2021. doi: 10.1073/pnas.2104878118.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 conference of the north American chapter of the association for computational linguistics: human language technologies (volume 1: long papers)*, pp. 6577–6595, 2024.

Pierre-Luc Germain, Anthony Sonrel, and Mark D. Robinson. pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. *Genome Biology*, 21(1):227, 2020. doi: 10.1186/s13059-020-02136-7.

Pierre-Luc Germain, Aaron Lun, Carlos Garcia Meixide, Will Macnair, and Mark D. Robinson. Doublet identification in single-cell sequencing data using scDblFinder. *F1000Research*, 10:979, 2022. doi: 10.12688/f1000research.73600.2.

Mahmoud Ghandi, Dongwon Lee, Morteza Mohammad-Noori, and Michael A. Beer. Enhanced regulatory sequence prediction using gapped k-mer features. *PLOS Computational Biology*, 10 (7):e1003711, 2014. doi: 10.1371/journal.pcbi.1003711.

John Gkountouras, Matthias Lindemann, Phillip Lippe, Efstratios Gavves, and Ivan Titov. Language agents meet causality–bridging llms and causal world models. In *International Conference on Learning Representations*, volume 2025, pp. 37311–37336, 2025.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi: 10.1016/0304-3975(85)90224-5.

M. Grace Gordon, Fumitaka Inoue, Beth Martin, Max Schubach, Vikram Agarwal, Sean Whalen, Shiyun Feng, Jingjing Zhao, Tal Ashuach, Ryan Ziffra, Anat Kreimer, Ilias Georgakopoulos-Soares, Nir Yosef, Chun Jimmie Ye, Katherine S. Pollard, Jay Shendure, Martin Kircher, and Nadav Ahituv. lentiMPRA and MPRAflow for high-throughput functional characterization of gene regulatory elements. *Nature Protocols*, 15(8):2387–2412, 2020. doi: 10.1038/s41596-020-0333-5.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomašev, Dina Zverinski, Ivor Rendulic, Elahe Vedadi, Florian Hasler, Luka Rimanic, Marina Boia, Ivan Budiselic, Ben Feinstein, Mathias Bellaiche, Tom Sheffer, Jan Freyberg, Jeremy Ratcliff, Ottavia Bertolli, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R. D. Costa, José R. Penadés, Gary Peltz, Yossi Matias, James Manyika, Demis Hassabis, Yunhan Xu, Pushmeet Kohli, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Accelerating scientific discovery with Co-Scientist. *Nature*, 655(8122):487–496, 2026. doi: 10.1038/s41586-026-10644-y. Preprint: arXiv:2502.18864 (2025), originally titled "Towards an AI co-scientist".

Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering—data curation cannot be compute agnostic. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22702–22711. IEEE, 2024.

R. Grantham. Amino acid difference formula to help explain protein evolution. *Science*, 185(4154): 862–864, 1974. doi: 10.1126/science.185.4154.862.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your LLM secretly a world model of the internet? model-based planning for web agents. *Transactions on Machine Learning Research*, 2025. arXiv:2411.06559.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. DS-Agent: Automated data science by empowering large language models with case-based reasoning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16813–16848. PMLR, 2024.

Laleh Haghverdi, Aaron T. L. Lun, Michael D. Morgan, and John C. Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature Biotechnology*, 36(5):421–427, 2018. doi: 10.1038/nbt.4091.

Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck, III, Shiwei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P. Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M. Fleming, Bertrand Yeung, Angela J. Rogers, Juliana M. McElrath, Catherine A. Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, 2021. doi: 10.1016/j.cell.2021.04.048.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992. doi: 10.1073/pnas.89.22.10915.

Lukas Heumos, Anna C. Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D. Lücken, Daniel C. Strobl, Juan Henao, Fabiola Curion, Single-cell Best Practices Consortium, Herbert B. Schiller, and Fabian J. Theis. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, 24(8):550–572, 2023. doi: 10.1038/s41576-023-00586-w.

Brian Hie, Hyunghoon Cho, Benjamin DeMeo, Bryan Bryson, and Bonnie Berger. Geometric sketching compactly summarizes the single-cell transcriptomic landscape. *Cell Systems*, 8(6): 483–493.e7, 2019. doi: 10.1016/j.cels.2019.05.003.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7):1114–1122, 2022. doi: 10.1038/s41587-021-01146-5.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations (ICLR)*, 2025.

Benhao Huang, Yingzhuo Yu, Jin Huang, Xingjian Zhang, and Jiaqi W Ma. Dca-bench: a benchmark for dataset curation agents. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5482–5492, 2025a.

Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candès, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25372–25437. PMLR, 2025b.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309. PMLR, 2024.

Wei-Chieh Huang, Weizhi Zhang, Yueqing Liang, Yuanchen Bei, Yankai Chen, Tao Feng, Xinyu Pan, Zhen Tan, Yu Wang, Tianxin Wei, Shanglin Wu, Ruiyao Xu, Liangwei Yang, Rui Yang, Wooseong Yang, Chin-Yuan Yeh, Hanrong Zhang, Haozhen Zhang, Siqi Zhu, Henry Peng Zou, Wanjia Zhao, Song Wang, Wujiang Xu, Zixuan Ke, Zheng Hui, Dawei Li, Yaozu Wu, Langzhou He, Chen Wang, Xiongxiao Xu, Baixiang Huang, Juntao Tan, Shelby Heinecke, Huan Wang, Caiming Xiong, Ahmed A. Metwally, Jun Yan, Chen-Yu Lee, Hanqing Zeng, Yinglong Xia, Xiaokai Wei, Ali Payani, Yu Wang, Haitong Ma, Wenya Wang, Chenguang Wang, Yu Zhang, Xin Eric Wang, Yongfeng Zhang, Jiaxuan You, Hanghang Tong, Xiao Luo, Xue Liu, Yizhou Sun, Wei Wang, Julian McAuley, James Zou, Jiawei Han, Philip S. Yu, and Kai Shu. A survey of agent memory in the second half: Towards self-evolving and long-horizon agents. *Transactions on Machine Learning Research*, 2026. arXiv:2602.06052.

IGVF Consortium. Deciphering the impact of genomic variation on function. *Nature*, 633(8028): 47–57, 2024. doi: 10.1038/s41586-024-07510-0.

Tomislav Ilicic, Jong Kyoung Kim, Aleksandra A. Kolodziejczyk, Frederik Otzen Bagger, Davis James McCarthy, John C. Marioni, and Sarah A. Teichmann. Classification of low quality cells from single-cell RNA-seq data. *Genome Biology*, 17(1):29, 2016. doi: 10.1186/s13059-016-0888-1.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. Large-scale data selection for instruction tuning. *arXiv preprint arXiv:2503.01807*, 2025.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. In *Advances in Neural Information Processing Systems*, volume 37, pp. 10088–10116. Curran Associates, Inc., 2024. doi: 10.52202/079017-0324. Datasets and Benchmarks Track.

Peter Jansen, Oyvind Tafjord, Marissa Radensky, Pao Siangliulue, Tom Hope, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Daniel S. Weld, and Peter Clark. CodeScientist: End-to-end semi-automated scientific discovery with code-based experimentation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13370–13467, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.692.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. doi: 10.1145/582415.582418.

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.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*, 2025.

W. Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127, 2007. doi: 10.1093/biostatistics/kxj037.

Armand Jordana, Jianghan Zhang, Joseph Amigo, and Ludovic Righetti. An introduction to zero-order optimization techniques for robotics. *arXiv preprint arXiv:2506.22087*, 2025.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Feiyang Kang, Hanze Li, Adam Nguyen, Mahavir Dabas, Jiaqi W. Ma, Frederic Sala, Dawn Song, and Ruoxi Jia. Can generalist agents automate data curation? *arXiv preprint arXiv:2606.04261*, 2026.

Joyce B. Kang, Aparna Nathan, Kathryn Weinand, Fan Zhang, Nghia Millard, Laurie Rumker, D. Branch Moody, Ilya Korsunsky, and Soumya Raychaudhuri. Efficient and precise single-cell reference atlas mapping with Symphony. *Nature Communications*, 12(1):5890, 2021. doi: 10.1038/s41467-021-25957-x.

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.

Andrej Karpathy. autoresearch: AI agents running research on single-GPU nanochat training automatically. https://github.com/karpathy/autoresearch, 2026. Accessed 2026-09-08.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Saiful Haq, Ashutosh Sharma, Thomas Joshi, Hanna Moazam, Heather Miller, et al. Dspy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations*, volume 2024, pp. 54928–54958, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Martin Kircher, Daniela M. Witten, Preti Jain, Brian J. O'Roak, Gregory M. Cooper, and Jay Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nature Genetics*, 46(3):310–315, 2014. doi: 10.1038/ng.2892.

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods*, 16(12):1289–1296, 2019. doi: 10.1038/s41592-019-0619-0.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Melissa J. Landrum, Jennifer M. Lee, Mark Benson, Garth R. Brown, Chen Chao, Shanmuga Chitipiralla, Baoshan Gu, Jennifer Hart, Douglas Hoffman, Wonhee Jang, Karen Karapetyan, Kenneth Katz, Chunlei Liu, Zenith Maddipatla, Adriana Malheiro, Kurt McDaniel, Michael Ovetsky, George Riley, George Zhou, J. Bradley Holmes, Brandi L. Kattman, and Donna R. Maglott. ClinVar: improving access to variant interpretations and supporting evidence. *Nucleic Acids Research*, 46(D1):D1062–D1067, 2018. doi: 10.1093/nar/gkx1153.

Dongwon Lee, David U. Gorkin, Maggie Baker, Benjamin J. Strober, Alessandro L. Asoni, Andrew S. McCallion, and Michael A. Beer. A method to predict the impact of regulatory variants from DNA sequence. *Nature Genetics*, 47(8):955–961, 2015. doi: 10.1038/ng.3331.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, 2022.

Houyi Li, Wenzhen Zheng, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuantao Fan, Zhenyu Ding, Haoying Wang, Ning Ding, Shuigeng Zhou, Xiangyu Zhang, and Daxin Jiang. Predictable scale: Part I, Step Law – optimal hyperparameter scaling law in large language model pretraining. *arXiv preprint arXiv:2503.04715*, 2025.

Ziming Li, Qianbo Zang, David Ma, Jiawei Guo, Tuney Zheng, Minghao Liu, Xinyao Niu, Yue Wang, Jian Yang, Jiaheng Liu, Wanjun Zhong, Wangchunshu Zhou, Wenhao Huang, and Ge Zhang. AutoKaggle: A multi-agent framework for autonomous data science competitions. *arXiv preprint arXiv:2410.20424*, 2024.

Aly Lidayan, Jakob Bjorner, Satvik Golechha, Kartik Goyal, and Alane Suhr. ABBEL: Learning natural-language belief states for memory-efficient interaction. *arXiv preprint arXiv:2512.20111*, 2025.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

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.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. In *International Conference on Learning Representations*, volume 2025, pp. 38305–38339, 2025a.

Zexi Liu, Yuzhu Cai, Xinyu Zhu, Yujie Zheng, Runkun Chen, Ying Wen, Yanfeng Wang, Weinan E, and Siheng Chen. ML-Master: Towards AI-for-AI via integration of exploration and reasoning. *arXiv preprint arXiv:2506.16499*, 2025b.

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018. doi: 10.1038/s41592-018-0229-2.

Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D. Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, Sergei Rybakov, Alexander V. Misharin, and Fabian J. Theis. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology*, 40(1):121–130, 2022. doi: 10.1038/s41587-021-01001-7.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Malte D. Luecken and Fabian J. Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, 2019. doi: 10.15252/msb.20188746.

Malte D. Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F. Mueller, Daniel C. Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, and Fabian J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, 2022.

Aaron T. L. Lun, Davis J. McCarthy, and John C. Marioni. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Research*, 5:2122, 2016. doi: 10.12688/f1000research.9501.2.

Yuyu Luo, Guoliang Li, Ju Fan, and Nan Tang. Data agents: Levels, state of the art, and open problems. *arXiv preprint arXiv:2602.04261*, 2026. SIGMOD 2026 Tutorial.

Feiyang Ma and Matteo Pellegrini. ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics*, 36(2):533–538, 2020. doi: 10.1093/bioinformatics/btz592.

Tianyi Ma, Yiyue Qian, Zheyuan Zhang, Zehong Wang, Xiaoye Qian, Feifan Bai, Yifan Ding, Xuwei Luo, Shinan Zhang, Keerthiram Murugesan, Chuxu Zhang, and Yanfang Ye. AutoData: A multi-agent system for open web data collection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Frederikke Isa Marin, Felix Teufel, Marc Horlacher, Dennis Madsen, Dennis Pultz, Ole Winther, and Wouter Boomsma. BEND: Benchmarking DNA language models on biologically meaningful tasks. In *International Conference on Learning Representations (ICLR)*, pp. 15246–15281, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/429e7b31625a8b7839f9e4d6e2aa9bb9-Abstract-Conference.html.

B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) – Protein Structure*, 405(2):442–451, 1975. doi: 10.1016/0005-2795(75)90109-9.

Mark Mazumder, Colby Banbury, Xiaozhe Yao, Bojan Karlaš, William Gaviria Rojas, Sudnya Diamos, Greg Diamos, Lynn He, Alicia Parrish, Hannah Rose Kirk, Jessica Quaye, Charvi Rastogi, Douwe Kiela, David Jurado, David Kanter, Rafael Mosquera, Will Cukierski, Juan Ciro, Lora Aroyo, Bilge Acun, Lingjiao Chen, Mehul Raje, Max Bartolo, Evan Sabri Eyuboglu, Amirata Ghorbani, Emmett Goodman, Addison Howard, Oana Inel, Tariq Kane, Christine R. Kirkpatrick, D. Sculley, Tzu-Sheng Kuo, Jonas W. Mueller, Tristan Thrush, Joaquin Vanschoren, Margaret Warren, Adina Williams, Serena Yeung, Newsha Ardalani, Praveen Paritosh, Ce Zhang, James Zou, Carole-Jean Wu, Cody Coleman, Andrew Ng, Peter Mattson, and Vijay Janapa Reddi. DataPerf: Benchmarks for data-centric AI development. In *Advances in Neural Information Processing Systems*, volume 36, pp. 5320–5347. Curran Associates, Inc., 2023. doi: 10.52202/075280-0235.

Ludovico Mitchener, Angela Yiu, Benjamin Chang, Mathieu Bourdenx, Tyler Nadolski, Arvis Sulovari, Eric C. Landsness, Daniel L. Barabasi, Siddharth Narayanan, Nicky Evans, Shriya Reddy, Martha Foiani, Aizad Kamal, Leah P. Shriver, Fang Cao, Asmamaw T. Wassie, Jon M. Laurent, Edwin Melville-Green, Mayk Caldas, Albert Bou, Kaleigh F. Roberts, Sladjana Zagorac, Timothy C. Orr, Miranda E. Orr, Kevin J. Zwezdaryk, Ali E. Ghareeb, Laurie McCoy, Bruna Gomes, Euan A. Ashley, Karen E. Duff, Tonio Buonassisi, Tom Rainforth, Randall J. Bateman, Michael Skarlinski, Samuel G. Rodriques, Michaela M. Hinks, and Andrew D. White. Kosmos: An AI scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.

Takashi Miyata, Sanzo Miyazawa, and Teruo Yasunaga. Two types of amino acid substitutions in protein evolution. *Journal of Molecular Evolution*, 12(3):219–236, 1979. doi: 10.1007/BF01732340.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A. Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379, 2023. doi: 10.52202/075280-2810.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with

reasoning memory. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.25140.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Stefan Peidli, Tessa D. Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J. Schumacher, Jake P. Taylor-King, Debora S. Marks, Augustin Luna, Nils Blüthgen, and Chris Sander. scPerturb: harmonized single-cell perturbation data. *Nature Methods*, 21(3):531–540, 2024. doi: 10.1038/s41592-023-02144-y.

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*, volume 37, pp. 30811–30849. Curran Associates, Inc., 2024. doi: 10.52202/079017-0970.

John C. Platt. Probabilities for SV machines. In Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans (eds.), *Advances in Large-Margin Classifiers*, pp. 61–74. The MIT Press, 2000. ISBN 9780262283977. doi: 10.7551/mitpress/1113.003.0008.

Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel A. R. Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul I. W. de Bakker, Mark J. Daly, and Pak C. Sham. PLINK: A tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007. doi: 10.1086/519795.

Danrui Qi, Zhengjie Miao, and Jiannan Wang. CleanAgent: Automating data standardization with LLM-based agents. *arXiv preprint arXiv:2403.08291*, 2024. Also appeared in the VLDB 2025 Workshop on Data Driven AI (DATAI).

Ben Rank, Hardik Bhatnagar, Ameya Prabhu, Shira Eisenberg, Karina Nguyen, Matthias Bethge, and Maksym Andriushchenko. PostTrainBench: Can LLM agents automate LLM post-training? *arXiv preprint arXiv:2603.08640*, 2026.

Zhe Ren, Yimeng Chen, Dandan Guo, Guowei Rong, Tonghui Li, R. B. Xiong, Qingfeng Lan, Wenyi Wang, Li Nanbo, Yibo Yang, Mingchen Zhuge, and Jürgen Schmidhuber. Self-improvements in modern agentic systems: A survey. *arXiv preprint arXiv:2607.13104*, 2026.

Jonathan Richens, David Abel, Alexis Bellot, and Tom Everitt. General agents contain world models. *arXiv preprint arXiv:2506.01622*, 2025.

Martiño Ríos-García, Nawaf Alampara, Chandan Gupta, Indrajeet Mandal, Sajid Mannan, Ali Asghar Aghajani, N. M. Anoop Krishnan, and Kevin Maik Jablonka. AI scientists produce results without reasoning scientifically. *arXiv preprint arXiv:2604.18805*, 2026.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, 2024. doi: 10.1038/s41587-023-01905-6.

Alan F. Rubin, Hannah Gelman, Nathan Lucas, Sandra M. Bajjalieh, Anthony T. Papenfuss, Terence P. Speed, and Douglas M. Fowler. A statistical framework for analyzing deep mutational scanning data. *Genome Biology*, 18(1):150, 2017. doi: 10.1186/s13059-017-1272-5.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using LLM agents as research assistants. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5977–6043, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.320.

Nabeel Seedat, Jiashuo Liu, and Mihaela van der Schaar. What's the next frontier for data-centric AI? Data savvy agents. *arXiv preprint arXiv:2511.01015*, 2025.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1708.00489.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023. NeurIPS 2023.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 761–769, 2016. doi: 10.1109/CVPR.2016.89.

Joykirat Singh, Zaid Khan, Archiki Prasad, Justin Chih-Yao Chen, Akshay Nambi, Hyunji Lee, Elias Stengel-Eskin, and Mohit Bansal. Agent-brace: Decoupling beliefs from actions in long-horizon tasks via verbalized state uncertainty. *arXiv preprint arXiv:2605.11436*, 2026.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

C. K. Sruthi and Meher Prakash. Deep2Full: Evaluating strategies for selecting the minimal mutational experiments for optimal computational predictions of deep mutational scan outcomes. *PLOS ONE*, 15(1):e0227621, 2020. doi: 10.1371/journal.pone.0227621.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.e21, 2019. doi: 10.1016/j.cell.2019.05.031.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991. doi: 10.1145/122344.122377.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106, 2026.

Izumi Takahara and Teruyasu Mizoguchi. Toward auditable AI scientists: A hypothesis evolution protocol for LLM agents. *arXiv preprint arXiv:2607.09195*, 2026.

Raghuveer Thirukovalluru, Bhuwan Dhingra, David Edwin Carlson, et al. Indata: Towards secure multi-step, tool-based data analysis. *arXiv preprint arXiv:2511.11933*, 2025.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442. Association for Computational Linguistics, 2023.

Hoa Thi Nhu Tran, Kok Siong Ang, Marion Chevrier, Xiaomeng Zhang, Nicole Yee Shin Lee, Michelle Goh, and Jinmiao Chen. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biology*, 21(1):12, 2020. doi: 10.1186/s13059-019-1850-9.

Beate Vieth, Swati Parekh, Christoph Ziegenhain, Wolfgang Enard, and Ines Hellmann. A systematic evaluation of single cell RNA-seq analysis pipelines. *Nature Communications*, 10(1):4667, 2019. doi: 10.1038/s41467-019-12266-7.

Gilles Wainrib, Barbara Bodinier, Haitem Dakhli, Josep Monserrat, Almudena Espin Perez, Sabrina Carpentier, Roberta Codato, and John Klein. Can AI scientist agents learn from lab-in-the-loop feedback? Evidence from iterative perturbation discovery. *arXiv preprint arXiv:2603.26177*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. arXiv:2305.16291.

Pingjie Wang, Hongcheng Liu, Yusheng Liao, Ziqing Fan, Yaxin Du, Shuo Tang, Yanfeng Wang, and Yu Wang. Selecting auxiliary data via neural tangent kernels for low-resource domains. 2025a.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025b.

Joshua D. Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z. Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887.e17, 2019. doi: 10.1016/j.cell.2019.05.006.

Sean Whalen, Jacob Schreiber, William S. Noble, and Katherine S. Pollard. Navigating the pitfalls of applying machine learning in genomics. *Nature Reviews Genetics*, 23(3):169–181, 2022. doi: 10.1038/s41576-021-00434-9.

Hjalmar Wijk, Tao Roa Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Joshua M. Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Jun Koba Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66772–66832. PMLR, 2025.

Bruce J. Wittmann, Yisong Yue, and Frances H. Arnold. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Systems*, 12(11):1026–1045.e7, 2021. doi: 10.1016/j.cels.2021.07.008.

F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018. doi: 10.1186/s13059-017-1382-0.

Samuel L. Wolock, Romain Lopez, and Allon M. Klein. Scrublet: Computational identification of cell doublets in single-cell transcriptomic data. *Cell Systems*, 8(4):281–291.e9, 2019. doi: 10.1016/j.cels.2018.11.005.

Xindi Wu, Mengzhou Xia, Rulin Shao, Zhiwei Deng, Pang Wei Koh, and Olga Russakovsky. ICONS: Influence consensus for vision-language data selection. *arXiv preprint arXiv:2501.00654*, 2024.

Yan Wu, Esther Wershof, Sebastian M. Schmon, Marcel Nassar, Błażej Osiński, Ridvan Eksi, Zichao Yan, Rory Stark, Kun Zhang, and Thore Graepel. PerturBench: Benchmarking machine learning models for cellular perturbation analysis. In *Advances in Neural Information Processing Systems*, 2025.

Nan Miles Xi and Jingyi Jessica Li. Benchmarking computational doublet-detection methods for single-cell RNA sequencing data. *Cell Systems*, 12(2):176–194.e6, 2021. doi: 10.1016/j.cels.2020.11.008.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S. Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023a.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34201–34227. Curran Associates, Inc., 2023b. doi: 10.52202/075280-1482.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024.

Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I. Jordan, and Nir Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular Systems Biology*, 17(1):e9620, 2021. doi: 10.15252/msb.20209620.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. arXiv:2502.12110.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*, 2024.

Xu Yang, Xiao Yang, Shikai Fang, Yifei Zhang, Jian Wang, Bowen Xian, Qizheng Li, Jingyuan Li, Minrui Xu, Yuante Li, Haoran Pan, Yuge Zhang, Weiqing Liu, Yelong Shen, Weizhu Chen, and Jiang Bian. R&D-Agent: An LLM-agent framework towards autonomous data science. *arXiv preprint arXiv:2505.14738*, 2025.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Xiao Yu, Baolin Peng, Ruize Xu, Michel Galley, Hao Cheng, Suman Nath, Jianfeng Gao, and Zhou Yu. Dyna-Think: Synergizing reasoning, acting, and world model simulation in AI agents. *arXiv preprint arXiv:2506.00320*, 2025.

Xiao Yu, Baolin Peng, Michel Galley, Hao Cheng, Qianhui Wu, Janardhan Kulkarni, Suman Nath, Zhou Yu, and Jianfeng Gao. Dyna-Mind: Learning to simulate from experience for better AI agents. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.09577.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025. doi: 10.1038/s41586-025-08661-4.

Dan Zhang, Sining Zhoubian, Min Cai, Fengzu Li, Lekang Yang, Wei Wang, Tianjiao Dong, Ziniu Hu, Jie Tang, and Yisong Yue. Datascibench: An llm agent benchmark for data science. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3685–3728, 2026a.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Tjarko Lange, and Jeff Clune. Darwin Gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations (ICLR)*, 2026b. arXiv:2505.22954.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025a.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. *arXiv preprint arXiv:2510.04618*, 2025b.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Jian Zhou and Olga G. Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12(10):931–934, 2015. doi: 10.1038/nmeth.3547.