How Should We Describe a Gene? Learning Biological Representations for Perturbation Prediction

Anonymous authors
Paper under double-blind review

Abstract

Predicting responses to unseen genetic perturbations requires representing perturbed genes using prior knowledge. Text provides a natural interface to this knowledge, but existing representations rely on fixed, hand-designed descriptions, leaving unclear what to include or how to organise it. We learn a shared description protocol for unseen genes, separating the benefits of richer knowledge from those of its representation. An LLM iteratively revises the protocol using held-out prediction and trial history while the prediction pipeline remains fixed. We predict transcriptome-wide log-fold changes and introduce *Continuous Directed Overlap*, a rank- and direction-aware metric focused on recovering the strongest effects. Across four Perturb-seq datasets, richer knowledge consistently improves prediction, structured descriptions add further gains, and search generally improves validation performance. In our evaluations, the resulting representations outperform the tested gene embeddings across multiple predictors, retain utility across cellular contexts, and enhance predictions from established single-cell generative models.

1 Introduction

Genetic perturbation experiments connect biological interventions to cellular consequences, providing a direct way to probe how genes shape cellular state and function. At single-cell resolution, screens such as Perturb-seq have mapped gene function and regulatory relationships at scale (Dixit et al., 2016; Norman et al., 2019; Replogle et al., 2022; Nadig et al., 2025), with growing relevance to early-stage drug discovery. However, even genome-scale screens cover only a fraction of possible perturbations, cell types, and biological contexts. Models that generalise to unseen perturbations could extend these screens *in silico* and help prioritise experiments and candidate targets (see Related Work). Exactly how to achieve this generalisation remains an open question.

A growing body of work shows that generalising to unseen perturbations is harder than it appears. Many models, including large foundation models, fail to outperform even simple mean-response baselines (Ahlmann-Eltze et al., 2025; Viñas Torné et al., 2025; Wei et al., 2026). Representations built from prior biological knowledge are a notable exception, and text-based gene embeddings in particular, despite their simplicity, are among the few approaches repeatedly shown to improve perturbation prediction, even when paired with relatively simple downstream models (Märtens et al., 2024; Chen & Zou, 2024b; Ramakrishnan et al., 2026; Cole et al., 2026).

Natural language provides a human-interpretable interface to prior biological knowledge: a gene description can combine molecular mechanism, interaction partners, regulatory roles, pathways, cellular context, and known phenotypes in a single representation, drawing on decades of accumulated scientific research. GenePT provided an early demonstration that embedding such descriptions can capture biologically meaningful structure across a range of biological tasks (Chen & Zou, 2024a). More recent methods have more directly used language models to retrieve related perturbations, infer regulatory relationships, or generate biological reasoning (Märtens et al., 2025; Wu et al., 2025b; Martell et al., 2026; Phillips et al., 2026). Across these approaches, however, the biological information used to represent each gene is either fixed in advance or assembled only as part of prediction-time reasoning. This opens a more fundamental question: *what information should the description itself contain, and how should it be expressed?*

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.

Large language models (LLMs) make this question tractable by allowing the description strategy itself to be systematically optimised: they can turn heterogeneous biological knowledge into uniform descriptions, and propose, test, and refine how that knowledge is represented against held-out prediction performance. To generalise to unseen perturbations, the description strategy must itself generalise across genes; we therefore learn a single shared strategy rather than optimising descriptions gene by gene. We formalise this description strategy as a reusable, human-readable protocol and improve it through iterative LLM-guided search, with previous trials guiding subsequent revisions. The rest of the pipeline remains fixed, isolating improvements to the representation itself. LLM-driven search against empirical feedback has proven effective for discovering algorithms and mathematical constructions (Romera-Paredes et al., 2024; Novikov et al., 2025), optimising prompts and text (Yang et al., 2024; Yuksekgonul et al., 2025) and automating research loops (Karpathy, 2026); here, the searched object is instead a biological representation strategy, organised around defined, interpretable changes to the gene-description protocol.

Evaluating these strategies also requires a target that reflects the biological question: which genes respond, by how much, and in which direction. Rather than post-perturbation log-normalised expression, a common target in the field (Roohani et al., 2024), we predict expression log-fold changes (LFCs) relative to control. Existing metrics capture only parts of this objective: whole-transcriptome errors can be dominated by weak responses, while correlation can remain high despite misranking the strongest effects. DEG-recovery and perturbation-discrimination measures (Roohani et al., 2025; Adduri et al., 2026) either discard effect magnitude and ordering or are sensitive to choices such as prediction scale and distance function (Liu et al., 2026). We therefore also introduce *Continuous Directed Overlap* (CDO), which rewards recovery of the strongest effects in the correct rank and direction while continuously penalising rank error.

Contributions. We make three main contributions: (i) We show that the amount and organisation of biological knowledge both shape generalisation to unseen perturbations, yielding representations that outperform existing gene embeddings and improve generative models; (ii) we introduce structured LLM-guided search over shared, human-readable gene-description protocols; and (iii) we formulate direct LFC prediction and introduce CDO, a rank- and direction-aware metric.

2 Related Work

Perturbation prediction and evaluation. Earlier perturbation models learned transferable effects across cellular contexts (Lotfollahi et al., 2019; 2023). For unseen genetic interventions, later methods incorporated gene-interaction priors, functional annotations, language-derived features, and generative models (Roohani et al., 2024; Yu et al., 2025; Wang et al., 2024; Klein et al., 2025; Adduri et al., 2026). Gene representations have also been learned from large single-cell datasets (Theodoris et al., 2023; Cui et al., 2024), protein sequence (Elnaggar et al., 2022; Lin et al., 2023), and external biological knowledge. Systematic studies show that model complexity alone does not guarantee better generalisation, and that both representation and evaluation choices can strongly affect conclusions (Ahlmann-Eltze et al., 2025; Cole et al., 2026; Viñas Torné et al., 2025; Schäfer et al., 2026; Wei et al., 2026; Nicol et al., 2026). Recent benchmarks have therefore introduced DEG-recovery and perturbation-discrimination measures alongside conventional expression-space metrics (Roohani et al., 2025; Adduri et al., 2026; Liu et al., 2026).

Biological language representations. GenePT introduced natural-language gene descriptions as reusable biological representations (Chen & Zou, 2024a), and later work applied similar representations to perturbation prediction (Märtens et al., 2024; Chen & Zou, 2024b; Wang et al., 2024; Zhu & Li, 2026; Ramakrishnan et al., 2026). Other approaches use LLMs directly to reason over related perturbations or regulatory relationships (Märtens et al., 2025; Martell et al., 2026; Phillips et al., 2026). The choice of text encoder can also substantially affect performance (Hedley et al., 2026).

Automated discovery and protocol optimisation. AI agents are increasingly being used to generate and test scientific ideas (M. Bran et al., 2024; Swanson et al., 2025; Gottweis et al., 2026; Ghareeb et al., 2026; Aygün et al., 2026). Autoresearch repeatedly proposes and tests changes against a fixed objective (Karpathy, 2026), but typically leaves the edits themselves open-ended. Work on automatic prompt optimisation instead uses previous candidates, scores, mutation, selection, or reflection to guide later revisions (Yang et al., 2024; Fernando et al., 2024; Agrawal et al., 2026). We combine these ideas by informing the search over description protocols with empirical feedback.

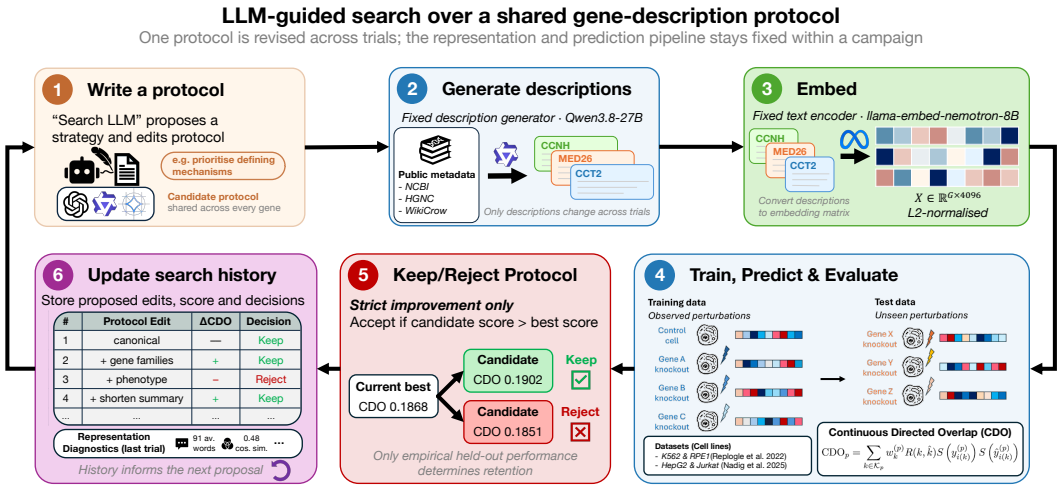


Figure 1: **LLM-guided search over a shared gene-description protocol.** Candidate protocols generate gene descriptions that are embedded and evaluated in a fixed perturbation-prediction pipeline; only improvements in held-out CDO are kept, and the search history informs subsequent proposals.

3 Learning Gene Description Protocols

Overview. We study generalisation to unseen single-gene perturbations by varying how each perturbed gene is represented and asking whether this improves prediction of its transcriptional response. For a perturbation p, the model receives a gene representation $\mathbf{x}_p \in \mathbb{R}^D$, where D is the representation dimension, and predicts the complete transcriptional response $\mathbf{y}_p \in \mathbb{R}^G$, where G is the number of responding genes, giving $f : \mathbf{x}_p \to \mathbf{y}_p$. Training and evaluation are split by perturbed gene, while all response genes are retained in both. Sections 3.1 & 3.2 describe the representations and search procedure; Section 4 describes the targets and evaluation.

3.1 Biological gene representations

To study how biological knowledge should be represented, rather than optimise thousands of gene descriptions independently, we search over one shared *description protocol*: a rule for selecting, prioritising, and organising information about a gene, applied uniformly across genes. This constrains the search space, and requires any improvement to generalise across held-out perturbations. It also separates two otherwise conflated questions: how much biological information is available, and how that information is organised (Figure 1).

We consider two knowledge settings. The first combines gene-centred functional information from NCBI (Brown et al., 2015) with HGNC-approved names, symbols, aliases and gene groups (Seal et al., 2026). The second adds WikiCrow, which provides cited, literature-derived summaries of molecular functions, interactions, pathways, and experimental context (Skarlinski et al., 2024). Comparing these settings separates gains from richer biological knowledge from gains due to how that knowledge is selected and organised.

Description generation. For each gene, the description model receives the shared protocol and available metadata and produces one natural-language description. We use the open-weight Qwen3.8-27B model (Qwen Team, 2026) with temperature 0.0, allowing description generation from a fixed local checkpoint without reliance on a proprietary API.

Search is initiated from a hand-designed *canonical protocol*, requesting a 60–100 word description emphasising specific molecular functions, pathways, interactions, cellular context and supported consequences of gene disruption (Appendix A). This provides a broad, mechanistically grounded baseline from which the search model can propose changes in content, emphasis, and organisation, applied uniformly across genes. The description model receives only the protocol and biological

metadata; search outcomes, response data and fold information affect generation only indirectly through revisions to the shared protocol (Section 3.2).

Descriptions are embedded using `nvidia/llama-embed-nemotron-8b` (Babakhin et al., 2025), yielding one 4,096-dimensional L2-normalised vector per gene. Unlike GenePT's proprietary OpenAI embeddings (Chen & Zou, 2024a), this open-weight encoder allows the representation pipeline to run locally without a closed embedding API and was selected for its strong performance across biological representation tasks (Hedley et al., 2026). Within each campaign, the knowledge sources, description model and decoding configuration, embedding model, predictor, cross-validation folds, and evaluator are fixed; only the shared description protocol changes.

3.2 Structured LLM-guided search

The space of possible description protocols is large and only weakly parameterised, with useful changes involving content, ordering, specificity, compression, mechanistic emphasis, or combinations of these. We therefore optimise the shared protocol iteratively, using previous trials to guide new proposals. At trial t, the search model A receives the incumbent protocol π_t, a compact history $\mathcal{H}_t$ of previous proposals and aggregate validation outcomes, and outcome-blind diagnostics $\mathcal{R}_t$ of the generated descriptions and embedding space:

$$\tilde{\pi}_t = A(\pi_t, \mathcal{H}_t, \mathcal{R}_t)\,. \tag{1}$$

The diagnostics include description length, fallback frequency, embedding similarity and effective rank, together with changes relative to the parent representation. Each proposal states a representation hypothesis and a corresponding intervention. We allow (i) **local** changes to one aspect of the representation, (ii) **multi-axis** changes to two or three related aspects under one coherent hypothesis, and (iii) **structural** changes to the broader strategy for information selection, ordering, level of detail, or mechanistic emphasis.

Search models and campaign controls. We compare three LLMs as search models: GPT-5.6 Sol (OpenAI, 2026), Qwen3.8-27B (Qwen Team, 2026) and Gemma-4-12B (Gemma Team, 2026). GPT-5.6 Sol was run with high reasoning effort. Qwen3.8-27B and Gemma-4-12B use an initial proposal temperature of 0.2; duplicate proposals are retried with new random seeds at temperatures 0.5, then 0.8, and unresolved exact duplicates are rejected before generation. Each campaign begins from the same canonical protocol and evaluates 40 candidates. Candidate protocols are instantiated using the fixed Qwen3.8-27B description-generation pipeline (Section 3.1); the description model, encoder, predictor, evaluator, folds and knowledge setting remain fixed within each campaign.

Each candidate is evaluated through the fixed prediction pipeline, yielding the primary validation score S defined below in Section 4. The incumbent changes only under strict improvement:

$$\pi_{t+1} = \begin{cases} \tilde{\pi}_t, & S(\tilde{\pi}_t) > S(\pi_t), \\ \pi_t, & \text{otherwise.} \end{cases} \tag{2}$$

Secondary metrics do not affect selection. After five consecutive discarded candidates, the next trial must test a substantially different representation strategy; a retained candidate resets the failure count. Appendix B gives the complete search instructions.

4 Experimental Setup

Transcriptome-wide LFC targets. Most perturbation-prediction benchmarks, including the processed setting popularised by GEARS (Roohani et al., 2024), predict post-perturbation log-normalised expression or its difference from control. These targets depend on the preceding normalisation and transformation and are often evaluated over a restricted set of highly variable genes. We instead predict differential-expression LFCs directly. LFC has a consistent interpretation across genes: zero indicates no estimated change, while +1 and -1 correspond to a doubling and halving of expression. Improvements must therefore capture the size and direction of the perturbation response rather than agreement on a dataset-specific processed expression scale.

We construct transcriptome-wide targets from raw UMI counts in four CRISPR Perturb-seq datasets: K562 and RPE1 (Replogle et al., 2022), and HepG2 and Jurkat (Nadig et al., 2025). For each perturbation, counts are aggregated into pseudobulk profiles and compared with matched non-targeting

controls using DESeq2 (Love et al., 2014; Muzellec et al., 2023). This produces a response matrix $\mathbf{Y} \in \mathbb{R}^{P \times G}$, where P is the number of perturbations and $Y_{pg} = \widehat{\mathrm{LFC}}_{pg}$ is the estimated $\log_2$ fold change of response gene g under perturbation p. We predict the complete row of this matrix without significance or effect-size filtering.

For K562 and RPE1, we retain the strongly responding perturbation subsets used by GEARS; HepG2 and Jurkat retain all successfully processed perturbations. The resulting matrix dimensions, together with the numbers of perturbations before and after subset selection, are reported in Table 4. Appendix C details the pseudobulk construction, dataset dimensions, subset selection, and comparison with the faster but substantially cruder SCANPY approximation (Wolf et al., 2018).

Predictors and held-out perturbations. During protocol optimisation, we keep the downstream predictor fixed and simple: an unregularised multi-output linear regression with no representation-specific hyperparameter tuning. All representations use the same 9-fold perturbation splits, with the predictor fitted on eight folds and evaluated on unseen perturbations in the ninth. These held-out predictions are repeatedly used for protocol selection, so we treat this as validation.

We additionally evaluate the same representations *post hoc* using k-nearest neighbours ($k = 5$), Gaussian process regression, Ridge regression, and PCA followed by linear regression. These predictors do not participate in protocol optimisation and use the same frozen perturbation splits. Further predictor specifications are given in Appendix E and details on the uncertainty quantification, paired comparisons and significance testing are given in Appendix F.

Standard response metrics. We report full-vector MAE and Pearson correlation, together with MAE@20/50/100 and Correlation@20/50/100. For the top-K metrics, response genes are ranked by the absolute DESeq2 Wald statistic, which incorporates both estimated effect size and uncertainty. Full-vector metrics measure global agreement but can be dominated by the many weakly responding genes, while top-K metrics focus on the strongest observed effects. Correlation captures agreement in relative magnitude but does not explicitly reward recovering their rank or direction.

Continuous Directed Overlap. For many downstream uses of perturbation prediction, the key question is whether a model recovers the strongest transcriptional responses, their ranking, and their direction. A natural metric is top-K overlap, which captures whether strongly responding genes are recovered. However, it conditions evaluation on a hard observed subset. This introduces an arbitrary cutoff, discards rank information beyond that set, and treats genes immediately above and below the cutoff discontinuously. Binary DEG recovery is similarly limited once significance is thresholded.

We therefore introduce *Continuous Directed Overlap* (CDO), which directly compares observed and predicted response rankings. Genes are ranked independently by absolute LFC magnitude, so strong responses in either direction appear near the top of each ranking. CDO rewards genes that occur at similar observed and predicted ranks with the correct direction of effect. The weighting of observed responses and the penalty for rank displacement both vary smoothly with rank, rather than imposing a single top-K boundary.

For perturbation p, let $y_i^{(p)}$ and $\hat{y}_i^{(p)}$ denote the observed and predicted LFCs. Let $i(k)$ be the gene at observed rank k, and $\hat{k}$ its rank in the predicted ordering, with ranks starting at zero. We define the set of observed responses exceeding the minimum effect size y_0 as $\mathcal{K}_p = \{k : |y_{i(k)}^{(p)}| \geq y_0\}$. Then,

$$\mathrm{CDO}_p = \sum_{k \in \mathcal{K}_p} w_k^{(p)} R(k, \hat{k}) S\left(y_{i(k)}^{(p)}\right) S\left(\hat{y}_{i(k)}^{(p)}\right), \tag{3}$$

where

$$w_k^{(p)} = \frac{(1 + k/k_0)^{-\nu}}{\sum_{j \in \mathcal{K}_p} (1 + j/k_0)^{-\nu}}, \quad R(k, \hat{k}) = \left(1 + \frac{|k - \hat{k}|}{k_0}\right)^{-\nu}, \quad S(y) = \begin{cases} +1, & y \geq y_0, \\ 0, & |y| < y_0, \\ -1, & y \leq -y_0. \end{cases} \tag{4}$$

Here $w_k^{(p)}$ prioritises the strongest observed responses, $R(k, \hat{k})$ penalises rank displacement, and $S(y)$ encodes response direction above y_0. The weights are normalised within each perturbation, so a perfect prediction has $\mathrm{CDO}_p = 1$. The parameter k_0 sets the characteristic rank scale for both

response weighting and displacement, while ν controls how rapidly both decay with rank. We use $k_0 = 100$, $\nu = 2$, and $y_0 = \log_2(1.5)$, corresponding to a 1.5-fold expression change. Predicted ranks are computed over all G genes, so large false-positive responses can displace true responders and reduce the score. Appendix D motivates these choices and evaluates their sensitivity across controlled prediction scenarios. Mean CDO across held-out perturbations is the primary validation score S used for protocol selection; all other metrics are secondary.

Baselines. We use a hierarchy of baselines to separate generic perturbation structure, the amount of biological information available, and how that information is represented:

(i) **Mean response.** Assign each held-out perturbation the mean LFC vector of its training fold, requiring no gene representation.

(ii) **Gene name.** Embed only the gene symbol, providing a minimal representation without additional biological information.

(iii) **Raw metadata.** Embed the available source information directly.

(iv) **Canonical description.** Apply the canonical protocol; this is the initial search incumbent.

All representation-based baselines use the same encoder, predictor, folds and evaluator, so differences reflect the information available and how it is represented.

We additionally compare against established biological representations from scGPT (Cui et al., 2024), ESM2 (Lin et al., 2023), Geneformer (Theodoris et al., 2023), GenePT (Chen & Zou, 2024a) and UniProt (Elnaggar et al., 2022). These representations are taken from the benchmark study of Cole et al. (2026) and evaluated within our prediction and evaluation framework. We also compare against scLAMBDA (Wang et al., 2024) and CellFlow (Klein et al., 2025) as established perturbation-prediction models. Because these methods define their own prediction pipelines rather than supplying representations to our fixed predictor, they are evaluated separately on matched perturbation subsets where required, see Appendix H.6 for further details. Further implementation and compute-environment details are provided in Appendix G.

5 RESULTS

Our experiments separate three factors: how much biological information is available about a gene, how it is represented, and whether empirical feedback can improve that representation. We study these first with the fixed linear predictor used for protocol optimisation, then test transfer across cell contexts, downstream predictors and established perturbation-prediction models. CDO is the primary metric; correlation and MAE assess whether gains extend beyond the optimised criterion. Unless stated otherwise, values are means ± bootstrap standard errors; paired 95% bootstrap intervals and extended metrics are reported in Appendix H.

Richer biological information improves perturbation prediction. Prediction improves as progressively richer information is supplied about the perturbed gene. Moving from the gene name to NCBI/HGNC metadata and then to WikiCrow-enriched metadata increases CDO across all four datasets (Table 1, Panel A), with Corr@100 and MAE@100 following the same progression. Paired bootstrap intervals support both information gains in every dataset (Appendix H).

How that information is expressed provides a further benefit. Applying the canonical protocol to the enriched metadata further improves every displayed point estimate over embedding the metadata directly, although these smaller differences are less uniformly resolved (Appendix H). The dominant benefit therefore comes from supplying richer biological knowledge, with an additional benefit from selecting and organising that knowledge into a structured description.

LLM-guided search further improves the shared description protocol. Starting from the same canonical protocol, GPT-5.6 Sol, Qwen3.8-27B and Gemma-4-12B each search over 40 candidates while the generator, encoder, predictor, perturbation splits and evaluator remain fixed. Under matched description-generation seeds, all three selected protocols improve mean linear CDO over canonical on K562 (Table 1, Panel B), with paired CIs excluding zero. Their RPE1 CDO point estimates are also positive, although smaller and not resolved by the paired intervals (Appendix H). No search model is best across all metrics, showing that useful protocols can be found with different LLMs without establishing a ranking between them.

A. Information and representation hierarchy

	K562			RPE1		
Representation	CDO↑	Corr@100↑	MAE@100↓	CDO↑	Corr@100↑	MAE@100↓
Gene name	0.1011 ± 0.0033	0.4440 ± 0.0117	0.5778 ± 0.0091	0.1448 ± 0.0038	0.5345 ± 0.0111	0.9888 ± 0.0152
NCBI/HGNC	0.1171 ± 0.0029	0.5338 ± 0.0098	0.4870 ± 0.0073	0.1710 ± 0.0035	0.6301 ± 0.0092	0.7799 ± 0.0100
+ WikiCrow	0.1283 ± 0.0029	0.5851 ± 0.0089	0.4564 ± 0.0068	0.1831 ± 0.0034	0.6680 ± 0.0085	0.7362 ± 0.0088
Canonical	**0.1290** ± 0.0027	**0.5973** ± 0.0086	**0.4519** ± 0.0072	**0.1882** ± 0.0033	**0.6766** ± 0.0082	**0.7189** ± 0.0085
	HepG2			Jurkat		
Representation	CDO↑	Corr@100↑	MAE@100↓	CDO↑	Corr@100↑	MAE@100↓
Gene name	0.1062 ± 0.0019	0.5306 ± 0.0073	1.1597 ± 0.0133	0.0611 ± 0.0014	0.3814 ± 0.0078	0.9359 ± 0.0111
NCBI/HGNC	0.1269 ± 0.0017	0.6281 ± 0.0058	0.8804 ± 0.0076	0.0730 ± 0.0013	0.4598 ± 0.0069	0.7081 ± 0.0068
+ WikiCrow	0.1328 ± 0.0016	0.6612 ± 0.0050	0.8329 ± 0.0069	0.0753 ± 0.0013	0.4799 ± 0.0065	0.6861 ± 0.0068
Canonical	**0.1422** ± 0.0016	**0.6850** ± 0.0046	**0.8026** ± 0.0067	**0.0804** ± 0.0013	**0.4981** ± 0.0064	**0.6639** ± 0.0066

B. Protocols selected by different search models

	K562			RPE1		
Search model	CDO↑	Corr@100↑	MAE@100↓	CDO↑	Corr@100↑	MAE@100↓
Canonical	0.1290 ± 0.0027	0.5973 ± 0.0086	0.4519 ± 0.0072	0.1882 ± 0.0033	0.6766 ± 0.0082	0.7189 ± 0.0085
GPT-5.6 Sol	0.1330 ± 0.0027	0.6060 ± 0.0084	0.4461 ± 0.0070	0.1908 ± 0.0033	**0.6855** ± 0.0079	**0.7077** ± 0.0081
Qwen3.8-27B	0.1351 ± 0.0027	0.6119 ± 0.0084	**0.4398** ± 0.0070	**0.1912** ± 0.0034	0.6782 ± 0.0084	0.7148 ± 0.0087
Gemma-4-12B	**0.1355** ± 0.0026	**0.6174** ± 0.0081	0.4415 ± 0.0068	0.1892 ± 0.0033	0.6832 ± 0.0078	0.7145 ± 0.0084

C. Cross-context performance of selected protocols

	CDO↑			MAE@100↓		
Dataset	Canonical	K562-selected	RPE1-selected	Canonical	K562-selected	RPE1-selected
K562	0.1290 ± 0.0027	**0.1330** ± 0.0027	0.1305 ± 0.0027	0.4519 ± 0.0072	**0.4461** ± 0.0071	0.4481 ± 0.0070
RPE1	0.1882 ± 0.0033	0.1867 ± 0.0034	**0.1908** ± 0.0033	0.7189 ± 0.0085	0.7160 ± 0.0085	**0.7077** ± 0.0081
HepG2	0.1422 ± 0.0016	**0.1444** ± 0.0015	0.1429 ± 0.0015	0.8026 ± 0.0067	**0.7930** ± 0.0063	0.7991 ± 0.0066
Jurkat	0.0804 ± 0.0013	**0.0810** ± 0.0012	0.0808 ± 0.0012	0.6639 ± 0.0066	0.6589 ± 0.0065	**0.6570** ± 0.0064

Table 1: **Biological information hierarchy, protocol search and cross-context transfer.** A. Prediction performance as biological information is added and organised using the canonical protocol. B. Protocols selected by three search models after 40 candidate evaluations. C. Performance of the GPT-5.6 Sol protocols selected on K562 and RPE1 across target datasets. Values are means with bootstrap standard errors, jointly resampling generation seeds and perturbations for generated descriptions. In C, evaluation on the source dataset is validation; all other applications are frozen transfers, with the predictor refitted on the target dataset. Bold marks the best displayed point estimate, not a statistically resolved difference. Paired 95% CIs are shown in Appendices. H.2, H.3 & H.5.

Figure 2 shows the optimisation trajectory for a single GPT-5.6 Sol run on RPE1. In this run, the selected protocol improves validation CDO by 3.8%, with corresponding gains in correlation and prediction error. Its accepted revisions progressively emphasise mechanistic detail, gene-family specialisation, phenotype, cellular context and physiological role. These changes are cumulative rather than independent ablations, but illustrate how the shared strategy evolves with validation performance. Appendix H.4 shows how they alter an individual gene description.

Selected protocols retain utility across cell contexts. We freeze the GPT-5.6 Sol protocols selected on K562 and RPE1 and apply each unchanged to the other datasets, regenerating descriptions under five matched seeds on every target (Table 1, panel C). Both remain competitive with the canonical protocol outside their source context, including on HepG2 and Jurkat. Absolute CDO remains close to canonical across transfers: five of six point differences are positive, but all paired CIs cross zero. MAE@100 point estimates improve in all six transfers, although these intervals also cross zero; full-profile MAE intervals exclude zero in three of six comparisons (Appendix H.5). These results provide modest evidence of transferable representation structure, particularly for reducing prediction error. Only the description protocol is transferred; the response predictor is fitted separately on each target dataset.

Optimised text representations remain useful across downstream predictors. We compare the K562- and RPE1-selected text representations with GenePT, ESM2, scGPT, Geneformer and UniProt under k-nearest neighbours, Gaussian process regression and Ridge regression, using iden-

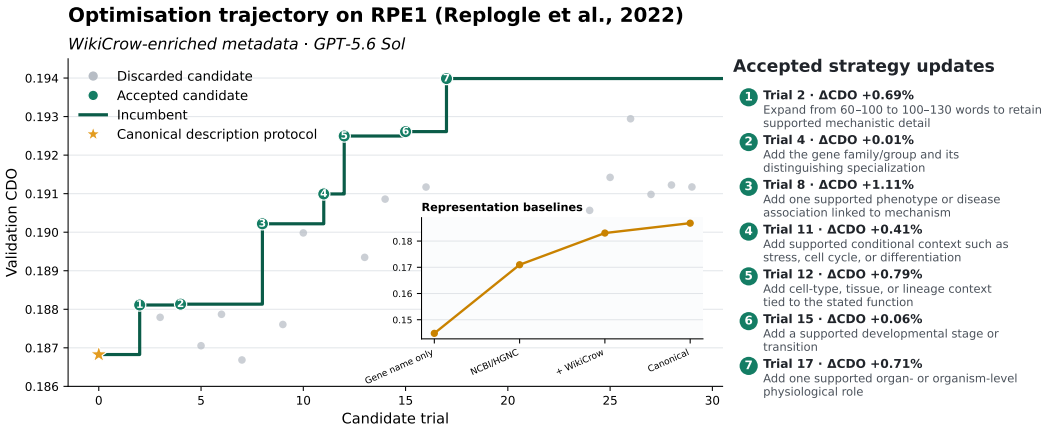


Figure 2: **LLM-guided optimisation of the shared gene-description protocol on RPE1.** Starting from the canonical protocol with WikiCrow-enriched metadata, GPT-5.6 Sol proposes revisions evaluated by 9-fold CV. Grey points denote discarded candidates, green points accepted candidates, and the step line the incumbent validation CDO. Numbered points show cumulative accepted strategy updates. The selected protocol increases within-run validation CDO from 0.1868 to 0.1940, a 3.8% relative improvement. This trajectory illustrates 30 trials of one search run and does not quantify variability across independent searches. **Inset:** the campaign-specific representation hierarchy before search, with gains from NCBI/HGNC metadata, WikiCrow, and the canonical protocol.

	K562			RPE1		
Representation	kNN	GP	Ridge	kNN	GP	Ridge
Mean response	0.1080 ± 0.0013	0.1080 ± 0.0013	0.1080 ± 0.0013	0.2897 ± 0.0033	0.2897 ± 0.0033	0.2897 ± 0.0033
GenePT	0.1770 ± 0.0036	0.1654 ± 0.0035	0.1474 ± 0.0025	0.2891 ± 0.0037	0.3042 ± 0.0035	0.3007 ± 0.0034
ESM2	0.1221 ± 0.0026	0.1378 ± 0.0023	0.1293 ± 0.0020	0.2667 ± 0.0034	0.2953 ± 0.0034	0.2952 ± 0.0034
scGPT	0.1338 ± 0.0029	0.1340 ± 0.0024	0.1338 ± 0.0024	0.2677 ± 0.0034	0.2972 ± 0.0034	0.2972 ± 0.0034
Geneformer	0.1136 ± 0.0023	0.1134 ± 0.0016	0.1110 ± 0.0014	0.2565 ± 0.0032	0.2910 ± 0.0033	0.2908 ± 0.0033
UniProt	0.1500 ± 0.0030	0.1414 ± 0.0025	0.1372 ± 0.0024	0.2743 ± 0.0034	0.2965 ± 0.0034	0.2966 ± 0.0034
K562-selected	**0.1929** ± 0.0036	**0.1834** ± 0.0035	**0.1620** ± 0.0027	**0.2965** ± 0.0037	**0.3109** ± 0.0036	**0.3062** ± 0.0035
RPE1-selected	0.1912 ± 0.0037	0.1823 ± 0.0035	0.1599 ± 0.0026	0.2948 ± 0.0037	0.3103 ± 0.0035	**0.3062** ± 0.0035

Table 2: **Comparison with existing gene representations across downstream predictors.** Entries are mean CDO with bootstrap standard errors on strictly matched perturbation panels. Representations use identical perturbations and folds within each dataset. The mean-response baseline is predictor-independent. Bold marks the highest displayed point estimate; paired intervals and linear results are reported in Appendix H.5.

tical perturbations and folds (Table 2). On both datasets, the selected representations outperform every existing embedding under all three predictors, with all paired 95% CIs excluding zero. They also outperform the train-fold mean response, showing that they capture perturbation-specific information beyond shared response structure. The gains therefore extend beyond the linear predictor used during protocol search. Linear and PCA-linear results are reported in Appendix H.5.

The learned representations transfer to established single-cell models. We test whether the optimised text representations remain useful in substantially different perturbation-prediction architectures. scLAMBDA and CellFlow use GenePT and ESM2 perturbation embeddings, respectively; we replace these with our gene-description embeddings while keeping the architectures, training procedures and unseen-perturbation folds fixed (Table 3). Strikingly, this substitution improves every displayed metric across both architectures and datasets. Representations selected using a linear predictor of count-derived LFCs therefore remain effective within two distinct generative models of single-cell expression. This broad transfer suggests that the learned descriptions capture reusable biological structure rather than exploiting a particular prediction pipeline. Canonical, transferred-protocol and non-control-mean comparisons are reported in Appendix H.6.

Model	Dataset	Representation	ΔPearson↑	ΔPearson@100↑	ΔMAE↓	ΔMAE@100↓
scLAMBDA	K562	Native GenePT	0.5353 ± 0.0088	0.6920 ± 0.0094	0.0375 ± 0.0004	0.1730 ± 0.0031
		K562-optimised text	**0.5958** ± 0.0076	**0.7568** ± 0.0077	**0.0357** ± 0.0004	**0.1533** ± 0.0027
	RPE1	Native GenePT	0.6593 ± 0.0072	0.7611 ± 0.0076	0.0533 ± 0.0004	0.2483 ± 0.0034
		RPE1-optimised text	**0.6778** ± 0.0070	**0.7818** ± 0.0073	**0.0517** ± 0.0004	**0.2315** ± 0.0032
CellFlow	K562	Native ESM2	0.4067 ± 0.0066	0.5766 ± 0.0080	0.0418 ± 0.0004	0.2096 ± 0.0030
		K562-optimised text	**0.4622** ± 0.0068	**0.6337** ± 0.0079	**0.0403** ± 0.0004	**0.1958** ± 0.0030
	RPE1	Native ESM2	0.6183 ± 0.0069	0.7241 ± 0.0076	0.0579 ± 0.0005	0.2895 ± 0.0039
		RPE1-optimised text	**0.6197** ± 0.0069	**0.7266** ± 0.0076	**0.0571** ± 0.0005	**0.2814** ± 0.0038

Table 3: **Optimised gene-description embeddings transfer to established single-cell architectures.** We replace the native GenePT embedding in scLAMBDA and ESM2 embedding in CellFlow while keeping architectures, training procedures and unseen-perturbation folds fixed. Here Δ denotes expression change relative to control. Further results are reported in Appendix H.6.

6 Discussion

Our results show that both the biological information available to a gene representation and its organisation affect prediction of unseen perturbations. Richer metadata provides the largest gains, while structured descriptions constructed from the same information improve performance further. Protocol optimisation yields additional validation gains, with accepted revisions emphasising gene-specific mechanisms, phenotypes, and cellular context. Selection by CDO establishes improvement on the prediction objective, but not greater factual or mechanistic accuracy.

The representations remain useful beyond the linear pipeline used during search. They outperform the tested existing gene embeddings under multiple predictors on K562 and RPE1, and improve every reported metric when substituted for the native embeddings in scLAMBDA and CellFlow. Cross-dataset protocol transfer is less conclusive: none of the six CDO differences is resolved at the 95% level, whereas full-profile MAE intervals exclude zero for three of the six transfers, including both transfers to HepG2 and the K562-selected protocol on Jurkat. One direction is therefore to retain a shared gene-level protocol while conditioning part of the description on the target cell type or state, allowing general biological knowledge to be combined with context-specific function.

The present knowledge setting remains limited. Incorporating pathways from KEGG and Reactome (Kanehisa et al., 2025; Milacic et al., 2024), protein complexes from CORUM (Steinkamp et al., 2025), and interactions from BioGRID and STRING (Oughtred et al., 2021; Szklarczyk et al., 2023), potentially organised as knowledge graphs, would provide substantially richer biological context. As the evidence becomes broader, more redundant, and sometimes contradictory, its selection and organisation become more important, potentially creating greater scope for LLM-guided search. Here we demonstrate task-specific optimisation using perturbation prediction; optimising jointly across datasets and tasks could yield representations intended for broader biological use. Our results provide strong impetus to explore text-based biological representations more systematically.

The experiments remain limited to single-gene CRISPR perturbations in four cell lines, with a fixed description model and encoder. The biological text sources may also contain literature related to the benchmark screens, so some gains from richer metadata may reflect benchmark-adjacent prior knowledge; we do not quantify this effect. Repeated use of the same validation folds may yield optimistic estimates, so independent evaluation and repeated search runs are needed. Description generation also incurs computational cost, and higher CDO has not yet been linked to improved experimental prioritisation or biological discovery. Prospective evaluation could test whether the learned representations identify informative perturbations or targets that would otherwise be missed.

7 Conclusion

We present a framework for learning shared, human-readable gene-description protocols for unseen perturbation prediction. Richer knowledge provides the largest gains, while structured descriptions and empirical search improve performance further. The resulting representations outperform the tested existing gene embeddings across downstream predictors and improve established single-cell models when substituted for their native embeddings. Extending this framework to richer knowledge sources, cellular-context-aware descriptions, and multi-task objectives could support both task-specific and broadly reusable biological representations.

AI USE STATEMENT

Generative AI tools were used both as part of the methodology studied in this work and to assist with research execution. In the experiments, language models generated gene descriptions from supplied public biological metadata and proposed revisions to the shared gene-description protocol during the search procedure; the models, prompts, search budgets, and evaluation procedure are described in the Methods and Appendix. Generative AI tools were also used to assist with code generation and debugging, and with editing and improving the presentation of the manuscript. Scientific interpretation of the results and formulation of the conclusions were carried out by the authors. The authors take responsibility for all final scientific claims, code, analyses, and manuscript.

ETHICS STATEMENT

This is a computational study using previously published Perturb-seq datasets and public biological annotations. It involves no new human, animal, or clinical experiments. Predictions are intended for research use and should not be interpreted as experimentally validated biological mechanisms or clinical guidance.

REPRODUCIBILITY STATEMENT

The main text and appendix describe the datasets, preprocessing, fixed perturbation splits, representations, search procedure, predictors, and evaluation metrics. Upon acceptance, we will release the code, optimised description protocols, and precomputed optimised gene embeddings used here.

REFERENCES

Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Mohsen Naghipourfar, Alishba Imran, Rohan Shah, Noam Teyssier, Rishi Verma, Christopher Carpenter, Basak Eraslan, Francis Chalissery, Rajesh Ilango, Vishvak Subramanyam, Chiara Ricci-Tam, Sanjay Nagaraj, Aidan Winters, Mingze Dong, Stefanie Fellinger, Adam Krejci, Tilmann Burckstummer, Sravya Tirukkovular, Jeremy Sullivan, Brian S. Plosky, Nicholas D. Youngblut, Jure Leskovec, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Alexander Dobin, Dave P. Burke, Hani Goodarzi, and Yusuf H. Roohani. Predicting cellular responses to perturbation across diverse contexts with state. *Cell*, 2026. ISSN 0092-8674. doi: 10.1016/j.cell.2026.07.052. URL https://doi.org/10.1016/j.cell.2026.07.052.

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*, 2026.

C. Ahlmann-Eltze, W. Huber, and S. Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22:1657–1661, 2025.

Eser Aygün, Anastasiya Belyaeva, Gheorghe Comanici, Marc Coram, Hao Cui, Jake Garrison, Renee Johnston, Anton Kast, Cory Y. McLean, Peter Norgaard, Zahra Shamsi, David Smalling, James Thompson, Subhashini Venugopalan, Brian P. Williams, Chujun He, Sarah Martinson, Martyna Plomecka, Lai Wei, Yuchen Zhou, Qian-Ze Zhu, Matthew Abraham, Erica Brand, Anna Bulanova, Jeffrey A. Cardille, Chris Co, Scott Ellsworth, Grace Joseph, Malcolm Kane, Ryan Krueger, Johan Kartiwa, Dan Liebling, Jan-Matthis Lueckmann, Paul Raccuglia, Xuefei Julie Wang, Katherine Chou, James Manyika, Yossi Matias, John C. Platt, Lizzie Dorfman, Shibl Mourad, and Michael P. Brenner. An ai system to help scientists write expert-level empirical software. *Nature*, 654(8120):909–916, 2026. doi: 10.1038/s41586-026-10658-6. URL https://doi.org/10.1038/s41586-026-10658-6.

Yauhen Babakhin, Radek Osmulski, Ronay Ak, Gabriel Moreira, Mengyao Xu, Benedikt Schifferer, Bo Liu, and Even Oldridge. Llama-embed-nemotron-8b: A universal text embedding model for multilingual and cross-lingual tasks, 2025.

G. R. Brown, V. Hem, K. S. Katz, M. Ovetsky, C. Wallin, O. Ermolaeva, I. Tolstoy, T. Tatusova, K. D. Pruitt, D. R. Maglott, and T. D. Murphy. Gene: a gene-centred information resource at NCBI. *Nucleic Acids Research*, 43:D36–D42, 2015.

Y. Chen and J. Zou. GenePT: A simple but effective foundation model for genes and cells built from ChatGPT. *bioRxiv*, 2023.10.16.562533, 2024a.

Y. Chen and J. Zou. Genepert: Leveraging genept embeddings for gene perturbation prediction. *bioRxiv*, 2024.10.27.620513, 2024b.

Yunshun Chen, Aaron T. L. Lun, and Gordon K. Smyth. From reads to genes to pathways: differential expression analysis of rna-seq experiments using rsubread and the edger quasi-likelihood pipeline. *F1000Research*, 5:1438, 2016. doi: 10.12688/f1000research.8987.2.

Elijah Cole, Geert-Jan Huizing, Sohan Addagudi, Nicholas Ho, Euxhen Hasanaj, Merel Kuijs, Toby Johnstone, Maria Carilli, Alec Davi, Caleb Ellington, Christoph Feinauer, Pan Li, Romain Menegaux, Shahin Mohammadi, Yanjun Shao, Josiah Zhang, Emma Lundberg, Le Song, Ziv Bar-Joseph, and Eric P. Xing. Foundation models improve perturbation response prediction. *bioRxiv*, 2026.02.18.706454, 2026.

H. Cui, C. Wang, H. Maan, K. Pang, F. Luo, N. Duan, and B. Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21:1470–1480, 2024.

A. Dixit, O. Parnas, B. Li, J. Chen, C. P. Fulco, L. Jerby-Arnon, N. D. Marjanovic, D. Dionne, T. Burks, R. Raychowdhury, B. Adamson, T. M. Norman, E. S. Lander, J. S. Weissman, N. Friedman, and A. Regev. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, 167:1853–1866, 2016.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:7112–7127, 2022.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *International Conference on Machine Learning*, 2024.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Dmytro Shved, Gavin J. Gyimesi, Jon M. Laurent, Samantha M. Wright, Muhammed T. Razzak, Andrew D. White, Silvia C. Finnemann, Michaela M. Hinks, and Samuel G. Rodriques. A multi-agent system for automating scientific discovery. *Nature*, May 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10652-y. URL https://doi.org/10.1038/s41586-026-10652-y.

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*, May 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10644-y. URL https://doi.org/10.1038/s41586-026-10644-y.

Jonathan G. Hedley, Philip Torr, and Kaspar Märtens. GenePT revisited: Do better text embeddings make better gene embeddings? In *ICLR 2026 Workshop on Machine Learning for Genomics Explorations*, 2026. URL https://openreview.net/forum?id=fDLe47paJ1.

M. Kanehisa, M. Furumichi, Y. Sato, Y. Matsuura, and M. Ishiguro-Watanabe. KEGG: biological systems database as a model of the real world. *Nucleic Acids Research*, 53:D672–D677, 2025.

Andrej Karpathy. Autoresearch: Ai agents running research on single-gpu nanochat training automatically. `https://github.com/karpathy/autoresearch`, 2026. GitHub repository.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, Nadezhda Azbukina, Fátima Sanchís-Calleja, Theo Uscidda, Artur Szalata, Manuel Gander, Aviv Regev, Barbara Treutlein, J. Gray Camp, and Fabian J. Theis. Cellflow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, 2025.04.11.648220, 2025.

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.

Qiyuan Liu, Qirui Zhang, Jinhong Du, Siming Zhao, and Jingshu Wang. Effects of distance metrics and scaling on the perturbation discrimination score. In *ICLR 2026 Workshop on Generative AI in Genomics (Gen^2): Barriers and Frontiers*, 2026. URL `https://openreview.net/forum?id=p3Y9B90R5x`.

Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 16(8):715–721, 2019.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular systems biology*, 19(6):MSB202211517, 2023.

M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, 15:550, 2014.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, May 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00832-8. URL `https://doi.org/10.1038/s42256-024-00832-8`.

Marc Boubnovski Martell, Josefa Lia Stoisser, Lawrence Phillips, Aditya Misra, Robert Kitchen, Jesper Ferkinghoff-Borg, Jialin Yu, Philip Torr, and Kaspar Märtens. CausalPert: Grounding LLM Hypotheses in Regulatory Networks for Gene Perturbation Prediction. In *ICLR 2026 Workshop on Machine Learning for Genomics Explorations*, 2026.

K. Märtens, R. Donovan-Maiye, and J. Ferkinghoff-Borg. Enhancing generative perturbation models with LLM-informed gene embeddings. In *ICLR 2024 Workshop on Machine Learning for Genomics Explorations*, 2024.

Kaspar Märtens, Marc Boubnovski Martell, Cesar A. Prada-Medina, and Rory Donovan-Maiye. LangPert: LLM-Driven Contextual Synthesis for Unseen Perturbation Prediction. In *ICLR 2025 Workshop on Machine Learning for Genomics Explorations*, 2025.

Davis J. McCarthy and Gordon K. Smyth. Testing significance relative to a fold-change threshold is a treat. *Bioinformatics*, 25(6):765–771, 2009. doi: 10.1093/bioinformatics/btp053.

Marija Milacic, Deidre Beavers, Patrick Conley, Chuqiao Gong, Marc Gillespie, Johannes Griss, Robin Haw, Bijay Jassal, Lisa Matthews, Bruce May, Robert Petryszak, Eliot Ragueneau, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Ralf Stephan, Krishna Tiwari, Thawfeek Varusai, Joel Weiser, Adam Wright, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The Reactome Pathway Knowledgebase 2024. *Nucleic Acids Research*, 52(D1): D672–D678, 2024.

B. Muzellec, M. Teleńczuk, V. Cabeli, and M. Andreux. Pydeseq2: a python package for bulk rna-seq differential expression analysis. *Bioinformatics*, 39:btad547, 2023.

A. Nadig, J. M. Replogle, and A. N. Pogson et al. Transcriptome-wide analysis of differential expression in perturbation atlases. *Nature Genetics*, 57:1228–1237, 2025.

Phillip B Nicol, Shriya Shivakumar, and Rafael A Irizarry. Spurious correlation inflates performance in single-cell perturbation prediction. *Biorxiv: the Preprint Server for Biology*, 2026.

T. M. Norman, M. A. Horlbeck, J. M. Replogle, A. Y. Ge, A. Xu, M. Jost, L. A. Gilbert, and J. S. Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365:786–793, 2019.

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, 2025. URL https://arxiv.org/abs/2506.13131.

OpenAI. GPT-5.6 System Card. https://deploymentsafety.openai.com/gpt-5-6, July 2026. Accessed 25 September 2026.

R. Oughtred, J. Rust, C. Chang, B.-J. Breitkreutz, C. Stark, A. Willems, L. Boucher, G. Leung, N. Kolas, F. Zhang, S. Dolma, J. Coulombe-Huntington, A. Chatr-Aryamontri, K. Dolinski, and M. Tyers. The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions. *Protein Science*, 30(1):187–200, 2021.

Lawrence Phillips, Marc Boubnovski Martell, Aditya Misra, Josefa Lia Stoisser, Cesar Prada-Medina, Rory Donovan-Maiye, and Kaspar Märtens. Synthpert: Enhancing llm biological reasoning via synthetic reasoning traces for cellular perturbation prediction. In *ICLR 2026 Workshop on Machine Learning for Genomics Explorations*, 2026.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Kalyan Ramakrishnan, Jonathan G. Hedley, Sisi Qu, Puneet K. Dokania, Philip H.S. Torr, Cesar A. Prada-Medina, Julien Fauqueur, and Kaspar Märtens. Beyond Mean Shifts: Predicting Distributional Responses to Unseen Genetic Perturbations. In *ICLR 2026 Workshop on Machine Learning for Genomics Explorations*, 2026.

J. M. Replogle, R. A. Saunders, and A. N. Pogson et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185:2559–2575, 2022.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6. URL https://doi.org/10.1038/s41586-023-06924-6.

Y. Roohani, K. Huang, and J. Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42:927–935, 2024.

Yusuf H. Roohani, Tony J. Hua, Po-Yuan Tung, Lexi R. Bounds, Feiqiao B. Yu, Alexander Dobin, Noam Teyssier, Abhinav Adduri, Alden Woodrow, Brian S. Plosky, Reshma Mehta, Benjamin Hsu, Jeremy Sullivan, Chiara Ricci-Tam, Nianzhen Li, Julia Kazaks, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Hani Goodarzi, and Dave P. Burke. Virtual cell challenge: Toward a turing test for the virtual cell. *Cell*, 188(13):3370–3374, 2025. doi: 10.1016/j.cell.2025.06.008.

Philipp Sven Lars Schäfer, Kendall Reid, Zach Boldyga, Ekin Deniz Aksu, Hugo Hakem, and Julio Saez-Rodriguez. Towards principled evaluation of single-cell perturbation prediction models. *bioRxiv*, pp. 2026–07, 2026.

R. L. Seal, B. Braschi, K. Gray, J. McClay, S. Tweedie, and E. A. Bruford. Genenames.org: the HGNC and PGNC resources in 2026. *Nucleic Acids Research*, 54:D1098–D1107, 2026.

M. D. Skarlinski, S. Cox, J. M. Laurent, J. D. Braza, M. Hinks, M. J. Hammerling, M. Ponnapati, S. G. Rodriques, and A. D. White. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv*, pp. 2409.13740, 2024.

Ralph Steinkamp, George Tsitsiridis, Barbara Brauner, Corinna Montrone, Gisela Fobo, Goar Frishman, Sorin Avram, Tudor I Oprea, and Andreas Ruepp. Corum in 2024: protein complexes as drug targets. *Nucleic Acids Research*, 53(D1):D651–D657, 2025.

Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. The Virtual Lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646(8085):716–723, October 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09442-9.

D. Szklarczyk, R. Kirsch, M. Koutrouli, K. Nastou, F. Mehryary, R. Hachilif, A. L. Gable, T. Fang, N. T. Doncheva, S. Pyysalo, P. Bork, L. J. Jensen, and C. von Mering. The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research*, 51(D1):D638–D646, 2023.

C. V. Theodoris, L. Xiao, A. Chopra, M. D. Chaffin, Z. R. Al Sayed, M. C. Hill, H. Mantineo, E. M. Brydon, Z. Zeng, X. S. Liu, and P. T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618:616–624, 2023.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A Teichmann, Mor Nitzan, and Maria Brbić. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, pp. 1–10, 2025.

Gefei Wang, Tianyu Liu, Jia Zhao, Youshu Cheng, and Hongyu Zhao. Modeling and predicting single-cell multi-gene perturbation responses with scLAMBDA. *bioRxiv*, 2024. doi: 10.1101/2024.12.04.626878. URL https://www.biorxiv.org/content/early/2024/12/08/2024.12.04.626878.

Zhiting Wei, Yiheng Wang, Yicheng Gao, Shuguang Wang, Ping Li, Duanmiao Si, Yuli Gao, Siqi Wu, Danlu Li, Kejing Dong, et al. Benchmarking algorithms for generalizable single-cell perturbation response prediction. *Nature Methods*, 23(2):451–464, 2026.

F. A. Wolf, P. Angerer, and F. J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:15, 2018.

Chih-Hsuan Wu, Xiang Zhou, and Mengjie Chen. Exploring and mitigating shortcomings in single-cell differential expression analysis with a new statistical paradigm. *Genome Biology*, 26:58, 2025a. doi: 10.1186/s13059-025-03525-6.

Menghua Wu, Russell Littman, Jacob Levine, Lin Qiu, Tommaso Biancalani, David Richmond, and Jan-Christian Huetter. Contextualizing biological perturbation experiments through language. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=5WEpbilssv.

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*, 2024.

Hengshi Yu, Weizhou Qian, Yuxuan Song, and Joshua D Welch. Perturbnet predicts single-cell responses to unseen chemical and genetic perturbations. *Molecular systems biology*, 21(8):960–982, 2025.

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. URL https://doi.org/10.1038/s41586-025-08661-4.

Ouyang Zhu and Jun Li. Scouter predicts transcriptional responses to genetic perturbations with large language model embeddings. *Nature Computational Science*, 6(1):21–28, 2026.

A CANONICAL DESCRIPTION PROTOCOL

Gene Description Protocol

Generate one compact, factual description for each gene using only the supplied public gene metadata.

Write a single paragraph of 60-100 words that:

1. Identifies the gene's primary molecular function and product class.
2. Names its most specific supported biological processes, pathways, complexes, substrates, or interaction partners.
3. States the relevant cellular location or cell-state context when supported.
4. Describes the plausible direct consequence of reducing or disrupting the gene's function, inferred only from the stated mechanism.

Prioritise details that distinguish this gene from genes with related functions. Use concrete biological terms instead of generic claims such as "important for cellular function." Omit unsupported fields rather than guessing. Do not mention the prompt, metadata source, prediction task, embeddings, dataset membership, or train/validation/test status.

Apply this protocol uniformly to the fixed gene set. Never use observed perturbation signatures, differential-expression measurements, test labels, or other outcome data when generating a description.

B SYSTEM PROMPT

System Prompt

You are an expert computational and molecular biologist with experience in functional genomics, gene-perturbation experiments, molecular and cellular mechanisms, biological knowledge representation, and representation learning.

You are the proposal agent in a sequential scientific autoresearch campaign.

ROLE AND GOAL

Improve a reusable gene-description protocol so that embeddings of uniformly generated gene descriptions better support prediction of held-out gene-perturbation responses.

Each candidate is one scientific experiment. Propose one coherent biological representation hypothesis and a complete candidate protocol that tests it.

EVIDENCE

Use only the information supplied in the trial prompt:

- the frozen experimental context and available metadata-source capabilities;
- the current accepted protocol;
- aggregate baseline and candidate metrics;
- prior hypotheses, move types, families, decisions and score deltas; and
- any supplied outcome-blind aggregate representation diagnostics.

Do not use, request or infer from individual perturbation signatures, differential-expression values or rankings, per-gene validation performance, validation or test labels, final-test results, or other dataset-output information.

Do not optimize individual gene descriptions. The proposed protocol must apply uniformly to every gene and may use only the public metadata supplied to the description generator. Require no unsupported information. Omit unsupported or conflicting claims rather than inventing them.

FROZEN EXPERIMENT

Only the description protocol may change. The dataset, target, folds, description-generation model and wrapper, metadata sources, embedding model, predictor, evaluator and primary metric are frozen.

The controller keeps a candidate if and only if its validation primary score strictly exceeds the incumbent score. Treat that decision as authoritative. Use score magnitude and component metrics as evidence, but do not treat small differences as proof of a general biological principle and do not design wording to exploit evaluator details.

SEARCH POLICY

Start every candidate from the complete current incumbent. Never silently merge a discarded candidate into it.

Every candidate has exactly one move type:

- local: change exactly one representation dimension while preserving the incumbent's overall organization;
- multi_axis: change two or three related dimensions that jointly implement one representation hypothesis; do not combine unrelated additions;
- structural: write a complete coherent policy from the frozen scientific and safety constraints. A structural move may change length, ordering, selection, consequence requirements, mechanistic emphasis and prose organization together. It need not preserve incumbent wording or format.

The normal controller policy is free_choice. Select local, multi_axis or structural according to which move best tests the most informative eligible hypothesis. The move_type is a descriptive scientific and provenance field, not a preference ordering. If the trial prompt activates a one-shot exploration requirement, test a substantially different representation strategy from the recent failures while choosing the move type that best implements it.

Adding more biological information is not inherently preferable. Removing, compressing, reorganizing, or reprioritizing supported information are equally valid representation hypotheses. Use outcome-blind representation diagnostics to distinguish what changed in the corpus, including moderate shifts, homogenization, or collapse. Representation diagnostics are explanatory evidence only. Do not optimize a diagnostic directly or assume that increasing or decreasing any diagnostic is intrinsically beneficial. Validation primary_score remains the sole keep/discard objective.

Preserve accepted principles semantically when they remain relevant, but do not accumulate clauses, exceptions or mandatory fields merely because they appeared in an earlier policy. Keep the resulting protocol concise, coherent and generally applicable.

Use the supplied history to avoid redundant experiments:

- use hypothesis_family_id only as a concise provenance label; do not assume that changing a free-form identifier makes a hypothesis scientifically new;
- do not rephrase a failed hypothesis without changing its biological representation mechanism;
- second formulations and near-miss retries are disabled unless the trial prompt explicitly states that a prospective rule has been configured;
- after repeated failures, prefer a genuinely different representation family rather than another nearby clause-level edit.

Before responding, select the eligible untested hypothesis that provides the most informative next experiment. Consider biological support, distinctness from previous trials, expected description-level effect, and the risk of removing useful information. Do not output this deliberation.

OUTPUT

Return exactly one JSON object and no surrounding text:

{
”move_type”: ”local|multi_axis|structural”,
”hypothesis_family_id”: ”short-stable-family-id”,
”formulation_id”: ”a”,
”hypothesis”: ”One concise, testable biological-representation hypothesis.”,
”representation_dimensions”: [”dimension”],
”intended_description_effect”: ”Concrete change expected in generated descriptions.”,
”main_risk”: ”The main way this change could reduce representation quality.”,
”proposed_protocol”: ”The complete revised protocol.”,
”rationale”: ”At most two concise sentences explaining why this experiment is informative.”
}

The proposed_protocol must be complete and self-contained. It must differ from the incumbent only as required by the selected move type and hypothesis.

C CONSTRUCTION AND VALIDATION OF THE LFC TARGETS

C.1 FROM PROCESSED EXPRESSION TO LOG-FOLD CHANGE

The construction of the response target is part of the learning problem: it determines the signal that every representation is trained and evaluated to predict. Previous perturbation-prediction work has generally used processed single-cell expression. In the GEARS pipeline, counts are normalised within each cell and log-transformed before the perturbed and control cells are averaged (Roohani et al., 2024). The resulting target is therefore a difference on a processed expression scale, rather than an effect estimated directly from the original counts.

This target is convenient, but its scale depends on the preceding normalisation and transformation. A difference between two mean log-normalised values is not the same as a fold change, and the same numerical difference need not have the same meaning for genes with different baseline expression. We instead use $\log_2$ fold change (LFC), for which zero indicates no estimated response, while $+1$ and -1 correspond to a doubling and halving of expression. This makes the prediction objective easier to interpret: the model must recover the size and direction of each response.

The processed GEARS datasets also restrict the output to highly variable genes. This reduces the size of the task, but variation between individual cells is not the same as a response to a perturbation. A gene can vary little at baseline and still change strongly after an intervention. We therefore return to the raw UMI count matrices for 4 CRISPR Perturb-seq screens, K562 & RPE1 (Replogle et al., 2022), and HepG2 & Jurkat (Nadig et al., 2025), and keep every measured response gene.

C.2 COUNT-BASED TARGET CONSTRUCTION

For each perturbation, we sum the raw counts of cells within each GEM group to form pseudobulk profiles. This gives one count profile per experimental group, rather than treating every cell as an independent replicate. Non-targeting cells from the same GEM group are summed separately and used as matched controls. We then fit an independent DESeq2 model for each perturbation using PyDESeq2, with the design `~ gem_group + condition` (Love et al., 2014; Muzellec et al., 2023). The first term accounts for systematic differences between experimental groups, while the second estimates the effect of the perturbation. DESeq2 performs count-depth normalisation and estimates the variability of each gene before fitting this effect.

For every perturbation–gene pair, we save the unshrunk $\log_2$ fold change, standard error, Wald statistic, and Benjamini–Hochberg-adjusted p-value. Only the LFC is used as the prediction target; the other values are retained for evaluation and quality checks. We do not apply a significance, effect-size, or top-K filter when constructing the matrices, so weak and strong responses remain in the same output space. For each cell line, this produces a matrix $\mathbf{Y} \in \mathbb{R}^{P \times G}$, where $Y_{pg} = \widehat{\mathrm{LFC}}_{pg}$ is the estimated LFC of response gene g under perturbation p. The target for one perturbation is the corresponding complete row of $\mathbf{Y}$.

For direct comparison with earlier results, K562 and RPE1 are restricted to the strongly responding perturbations selected by the GEARS preprocessing procedure (Roohani et al., 2024). This procedure requires at least 50 differentially expressed genes, at least 25 retained cells and, when it can be measured, at least a 30% reduction in the perturbed gene. The perturbation set is therefore held fixed while the target changes from processed expression to count-derived LFC. This restriction should be kept in mind: K562 and RPE1 measure generalisation among perturbations already known to produce a clear response, rather than over the full screen. HepG2 and Jurkat retain every successfully processed perturbation. Table 4 gives the resulting dimensions.

C.3 COMPARISON WITH A FASTER TARGET ESTIMATOR

Fitting a separate DESeq2 model for several thousand perturbations is computationally expensive. We therefore tested whether a faster Scanpy-based calculation could produce sufficiently similar targets (Wolf et al., 2018). This approximation normalises the total count in each cell, applies a log transformation, and estimates LFC from the mean transformed expression of the perturbed and control cells. It avoids fitting a separate count model and is therefore attractive when many perturbations must be processed.

Dataset	Perturbations processed	Perturbations used	Response genes
HepG2	2,393	2,393	9,624
Jurkat	2,392	2,392	8,882
K562	2,057	1,092	8,563
RPE1	2,393	1,543	8,749

Table 4: **Dimensions of the LFC datasets.** "Processed" gives the number of non-control perturbations for which an LFC profile was produced. K562 and RPE1 are subsequently restricted to the strongly responding GEARS subsets for the prediction experiments.

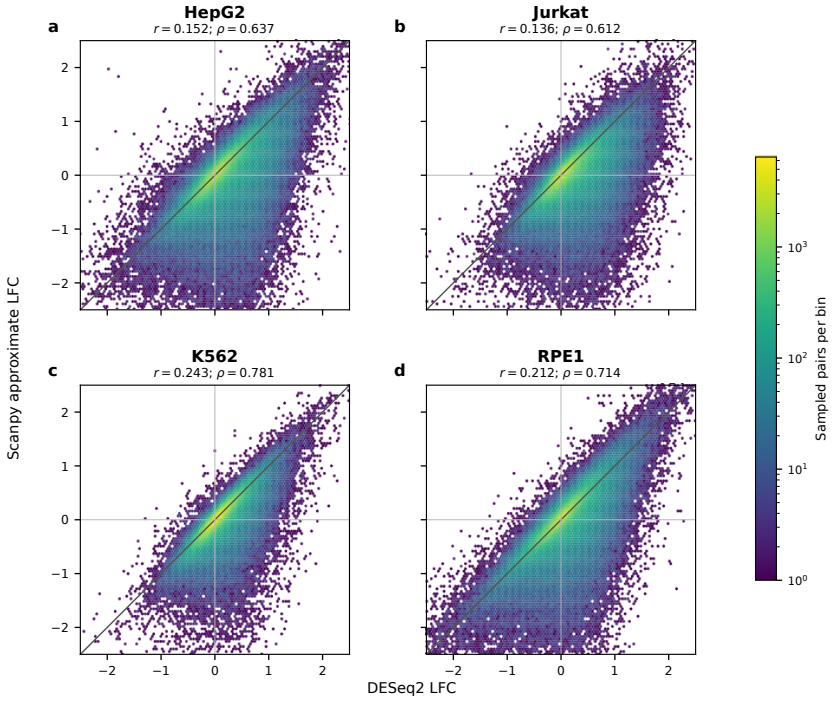


Figure 3: Pairwise comparison of DESeq2 and Scanpy-derived LFCs for **a**, HepG2; **b**, Jurkat; **c**, K562; and **d**, RPE1. Each panel shows 250,000 uniformly sampled perturbation–gene pairs; colour gives the number of pairs in each bin on a logarithmic scale and the diagonal marks exact agreement. Both positive and negative effects are shown on a common axis. The displayed range contains at least 99.5% of the sampled absolute LFCs; more extreme values are omitted only from the visualisation. Pearson r is calculated over all aligned pairs and Spearman ρ over the plotted sample.

The two methods nevertheless make different assumptions. The Scanpy estimate is calculated after cell-level normalisation and does not model variation in the raw counts or match perturbations and controls by GEM group. These differences matter here because the estimated values are not only used for an exploratory analysis: they become the labels used to train and compare every model, and the objective optimised by our LLM-guided search. A faster estimator is only a suitable replacement if it preserves the effect sizes and the identity of the strongest responses, rather than their broad ordering alone.

We compare the two estimators before applying the final GEARS perturbation subsets. The comparison therefore covers every successfully processed perturbation: 23,030,232 perturbation–gene pairs for HepG2, 21,245,744 for Jurkat, 17,614,091 for K562, and 20,936,357 for RPE1.

Figure 3 shows the complete signed relationship, including genes estimated to increase and decrease after perturbation. The methods preserve some broad ordering, with sampled Spearman correlation from 0.612 to 0.781, but the points spread widely from the equality line. Global Pearson correlation

Dataset	Global r	Sampled ρ	Median r_p	Jaccard	Precision	Recall	F1	Sign	Top-100
HepG2	0.152	0.637	0.264	0.137	0.138	0.868	0.239	0.994	0.070
Jurkat	0.136	0.612	0.541	0.101	0.103	0.555	0.174	0.951	0.120
K562	0.243	0.781	0.810	0.163	0.167	0.858	0.279	0.997	0.310
RPE1	0.212	0.714	0.528	0.208	0.213	0.900	0.344	0.999	0.190

Table 5: **Agreement between DESeq2 and the Scanpy approximation.** DEG metrics use $\tau = \log_2(1.5)$, with $q < 0.05$ additionally required for the DESeq2 reference set. Top-100 is the median fraction of genes shared between the two absolute-LFC rankings. Sign is the directional agreement conditional on a gene being selected by both methods.

is only 0.136–0.243, indicating substantial disagreement in the numerical target values. The higher K562 within-perturbation correlation does not remove this scale disagreement across the full matrix.

We report Pearson correlation across all perturbation–gene pairs, Spearman correlation over a random sample of pairs, and the median Pearson correlation calculated separately for each perturbation. We also compare the sets of genes called differentially expressed, meaning that their estimated response is both large enough and, for DESeq2, statistically supported. At threshold τ, the DESeq2 reference requires $q < 0.05$ and $|\widehat{\mathrm{LFC}}| \geq \tau$, whereas the Scanpy approximation can only apply the LFC threshold because it does not produce an equivalent adjusted p-value. Directional precision, recall, and F1 additionally require agreement on whether expression increases or decreases. Our main comparison uses $\tau = \log_2(1.5) \approx 0.585$, corresponding to at least a 1.5-fold change. We also report Jaccard agreement and the median overlap between the 100 genes with the largest absolute LFC for each perturbation.

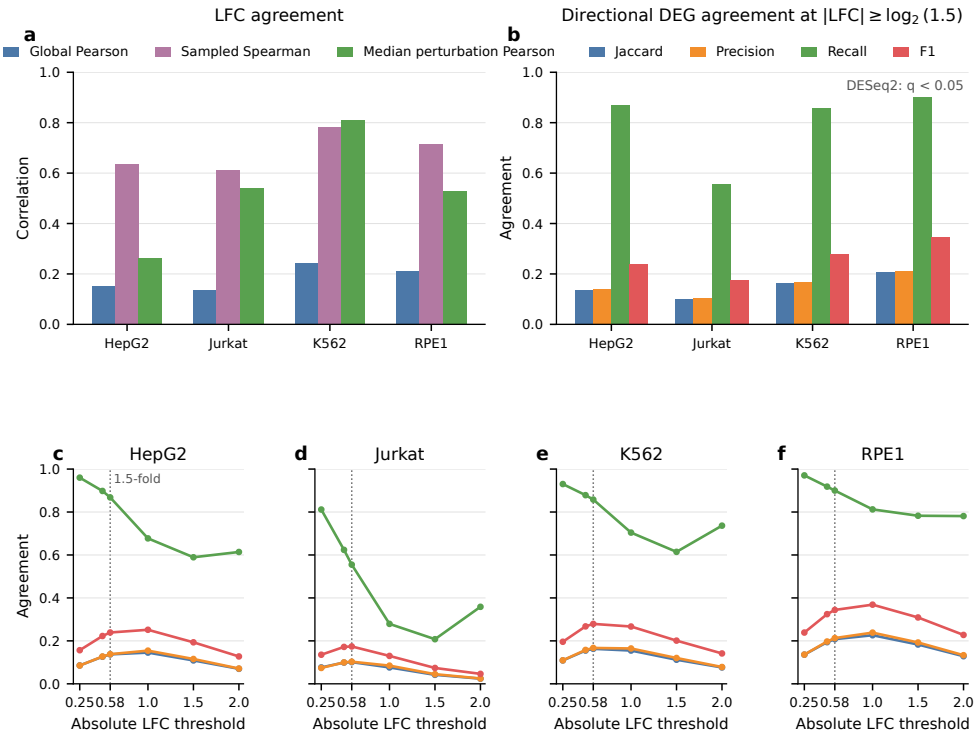


Figure 4: Summary of agreement between the two target estimators. **a**, Correlation across all perturbation–gene pairs, sampled rank correlation, and median within-perturbation correlation. **b**, Agreement between directional DEG sets at $|\mathrm{LFC}| \geq \log_2(1.5)$; the DESeq2 reference additionally requires $q < 0.05$. **c–f**, Sensitivity to the absolute LFC threshold. Scanpy retains relatively high recall but has low precision and Jaccard agreement across datasets and thresholds.

The median top-100 overlap ranges from 7% for HepG2 to 31% for K562, showing that the methods often select different genes as the strongest responses. The thresholded comparison has a consistent high-recall, low-precision pattern. At the 1.5-fold threshold, Scanpy recovers between 55.5% and 90.0% of the directional DESeq2 calls, but only 10.3% to 21.3% of its own directional calls occur in the DESeq2 reference set. Fig. 4 shows that this pattern persists across effect-size thresholds.

The absolute precision and recall values should be interpreted with the asymmetry of the DEG definitions in mind: DESeq2 requires statistical support, whereas Scanpy uses only an effect threshold. This does not explain the weak LFC correlations or the low top-100 overlap, neither of which uses the adjusted p-value. High sign agreement is also conditional on both methods selecting the same gene and should not be read as high agreement over the complete target.

We do not treat DESeq2 as biological ground truth. The relevant question is whether the much faster approximation would define a sufficiently similar supervised learning problem. It does not: Scanpy captures some broad ordering but does not consistently reproduce the effect sizes or the strongest response genes. We therefore use the DESeq2-derived targets throughout. Its count-based model, explicit treatment of gene-level variability, and matching across experimental groups provide a more statistically appropriate LFC estimate for these data, although the resulting targets remain subject to sampling noise and model assumptions.

D CDO PARAMETER SELECTION AND VALIDATION

The main text defines Continuous Directed Overlap (CDO) and the parameters y_0, k_0 and ν. Here we describe how these parameters were determined and evaluate the behaviour of the metric.

D.1 EVALUATION CONTEXT

Evaluation design can change conclusions about perturbation prediction: expression-space metrics may reflect systematic variation or the choice of controls (Viñas Torné et al., 2025; Schäfer et al., 2026; Wei et al., 2026; Nicol et al., 2026). The Virtual Cell Challenge and Cell-Eval introduced DEG-recovery and perturbation-discrimination measures (Roohani et al., 2025; Adduri et al., 2026). Binary DEG scores discard effect magnitude and ordering; perturbation discrimination asks a different question and is sensitive to prediction scale and distance choice (Liu et al., 2026). CDO instead evaluates the ranking and direction of the strongest transcriptional responses.

D.2 FIXED LFC THRESHOLD

We treat y_0 differently from k_0 and ν. The latter control the rank geometry of CDO, whereas y_0 determines which observed expression changes are sufficiently large to be considered biologically relevant. We therefore specify y_0 *a priori* from general practice rather than tuning it on our datasets.

We use $y_0 = \log_2(1.5) \simeq 0.585$, corresponding to a 1.5-fold expression change. Thresholds of this magnitude have precedent in differential-expression analysis: TREAT formalised testing relative to a minimum fold change of biological interest and used a 1.5-fold threshold in its evaluation (McCarthy & Smyth, 2009); an established edgeR RNA-seq workflow similarly uses $\log_2(1.5)$ when testing for reasonably large expression changes (Chen et al., 2016); and the same threshold has recently been used in single-cell differential-expression analysis (Wu et al., 2025a). We therefore fix y_0 *a priori*, independently of the CDO parameter sweep.

At this threshold, all perturbations retain at least one evaluated gene. Across the 1092 K562 perturbations, the median number of genes passing the threshold is $K_p = 593.5$, with a 10–90% range of 118.3–1875.8. Across the 1543 RPE1 perturbations, the median is $K_p = 1792$, with a 10–90% range of 307–3014.8.

Here $\mathcal{G}_p = \{i : |y_i^{(p)}| \geq y_0\}$ and $K_p = |\mathcal{G}_p|$. Two implementation details are worth noting. First, membership of $\mathcal{G}_p$ depends only on the observed response, while predicted ranks are computed over all genes. Thus, genes with negligible observed effects can still reduce CDO if assigned large predicted effects, since they displace genuine responders in the predicted ranking. Second, $S(\hat{y})$ applies the same effect-size threshold to the prediction. A correctly signed but arbitrarily small predicted LFC therefore receives no directional credit.

D.3 CONTROLLED VALIDATION

To characterise CDO independently of any learned predictor, we construct a set of controlled predictions directly from each observed LFC vector. These controls separately probe sensitivity to noise, direction, response magnitude, and false-positive rankings. Table 6 defines each control and reports its CDO at the final parameter setting, $y_0 = \log_2(1.5)$, $k_0 = 100$, $\nu = 2$.

For the Gaussian-noise controls, σ_p denotes the standard deviation of observed LFCs across genes for perturbation p. For the random control, the observed LFC values are permuted across gene identities, preserving their marginal distribution while eliminating gene-specific correspondence. For the 10% sign-flip control, 10% of genes in $\mathcal{G}_p$ are selected at random and their signs reversed without changing their magnitudes. For the false positive control, up to 100 genes with weak observed effects are assigned random signs and predicted magnitudes larger than the largest observed $|y|$, thereby promoting them above the genuine responders in the predicted ranking.

The results provide several direct calibration checks. A perfect prediction gives $\mathrm{CDO}_{\mathrm{perfect}} = 1$, whereas preserving the absolute ranking while reversing every response direction gives $\mathrm{CDO}_{\mathrm{wrong}} = -1$. Random gene-wise permutations produce scores close to zero in both datasets. The noise controls also behave monotonically, with moderate corruption scoring above stronger corruption. The 10% sign-flip control also closely follows its analytical expectation. If flipped genes are sampled independently of rank, $\mathbb{E}[\mathrm{CDO}_{\mathrm{flip10}}] \simeq 0.9(1) + 0.1(-1) = 0.8$. The magnitude controls illustrate the role of the predicted-effect threshold. Doubling every LFC leaves the absolute ranking and direction unchanged and keeps all evaluated predictions above y_0, giving $\mathrm{CDO}_{\mathrm{double}} = 1$ exactly. Halving the response also preserves ranking and direction, but causes some predictions to fall below y_0. Its score therefore depends on the distribution of observed effect sizes near the threshold, producing the different values observed in K562 and RPE1.

D.4 SENSITIVITY TO k_0 AND ν

Having fixed y_0, we vary the two parameters controlling rank geometry. The unnormalised observed-rank weight and rank-agreement terms are

$$\omega(k) = \left(1 + \frac{k}{k_0}\right)^{-\nu}, \qquad r(k, \hat{k}) = \left(1 + \frac{|k - \hat{k}|}{k_0}\right)^{-\nu}. \tag{5}$$

Here k_0 sets the characteristic rank scale and ν controls the steepness of the decay. We evaluate $\nu \in \{1.25, 1.5, 1.75, 2, 2.25, 2.5, 3\}$ and $k_0 \in \{25, 50, 75, 100, 125, 150, 200, 300\}$ across all K562 and RPE1 perturbations, with y_0 fixed.

Figures 5 and 6 show smooth and qualitatively consistent behaviour across datasets. More tolerant rank geometry increases the scores of noisy predictions but also gives greater credit to false-positive displacement, while random predictions remain close to zero throughout. The half-magnitude control is more sensitive to settings that concentrate weight on the strongest responders, whereas the double-magnitude control remains exactly one.

We select $(k_0, \nu) = (100, 2)$ as an interpretable central setting rather than by optimising downstream performance. It gives $r(\Delta k) = (1 + \Delta k/100)^{-2}$, so displacements of 50, 100, and 200 ranks retain approximately 0.44, 0.25, and 0.11 credit, respectively. The selected point lies within a broad, smooth region of both sensitivity maps. These values were fixed before protocol search and held constant throughout all subsequent experiments.

Table 6: Controlled predictions used to validate CDO and their resulting scores for the selected parameters $y_0 = \log_2(1.5)$, $k_0 = 100$, and $\nu = 2$. Reported values are mean $\pm$ standard deviation across perturbations.

Control	Symbol	Prediction	Purpose	Expected	K562	RPE1
Perfect	$\mathrm{CDO}_{\mathrm{perfect}}$	$\hat{\mathbf{y}} = \mathbf{y}$	Exact ranking, magnitude, and direction.	$\mathrm{CDO} = 1$	1.000	1.000
Small noise	$\mathrm{CDO}_{\mathrm{small}}$	$\hat{\mathbf{y}} = \mathbf{y} + \boldsymbol{\epsilon}$, $\epsilon_i \sim \mathcal{N}\big(0, (0.5\sigma_p)^2\big)$	Moderate perturbation of magnitudes and ranks.	$\mathrm{CDO}_{\mathrm{small}} > \mathrm{CDO}_{\mathrm{large}}$	0.512 ± 0.090	0.488 ± 0.080
Large noise	$\mathrm{CDO}_{\mathrm{large}}$	$\hat{\mathbf{y}} = \mathbf{y} + \boldsymbol{\epsilon}$, $\epsilon_i \sim \mathcal{N}\big(0, \sigma_p^2\big)$	Stronger perturbation of magnitudes and ranks.	$\mathrm{CDO}_{\mathrm{large}} < \mathrm{CDO}_{\mathrm{small}}$	0.323 ± 0.086	0.308 ± 0.087
Random	$\mathrm{CDO}_{\mathrm{random}}$	$\hat{\mathbf{y}} = P_\pi \mathbf{y}$	Preserve the marginal LFC distribution but destroy gene-specific correspondence.	$\mathrm{CDO} \approx 0$	0.007 ± 0.008	0.004 ± 0.006
Wrong sign	$\mathrm{CDO}_{\mathrm{wrong}}$	$\hat{\mathbf{y}} = -\mathbf{y}$	Correct absolute ranking but reversed directions.	$\mathrm{CDO} = -1$	-1.000	-1.000
Half magnitude	$\mathrm{CDO}_{\mathrm{half}}$	$\hat{\mathbf{y}} = 0.5\mathbf{y}$	Correct rank and direction but underestimated effects; some predictions fall below y_0.	Dataset-dependent	0.313 ± 0.250	0.625 ± 0.279
Double magnitude	$\mathrm{CDO}_{\mathrm{double}}$	$\hat{\mathbf{y}} = 2\mathbf{y}$	Correct rank and direction but overestimated effects.	$\mathrm{CDO} = 1$	1.000	1.000
10% sign flip	$\mathrm{CDO}_{\mathrm{flip10}}$	Reverse the signs of 10% of genes in $\mathcal{G}_p$	Controlled directional errors while preserving absolute ranks.	$\mathrm{CDO} \approx 0.8$	0.801 ± 0.031	0.798 ± 0.036
100 false positives	$\mathrm{CDO}_{\mathrm{100FP}}$	Promote up to 100 weak-response genes above all true responders	Controlled high-ranking false positives that displace genuine responders.	$\mathrm{CDO} = r(100) = 0.25$	0.250	0.250

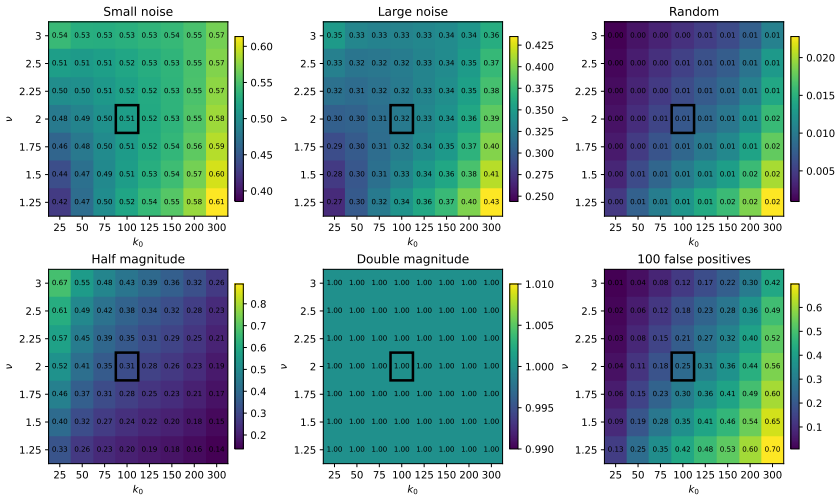


Figure 5: Sensitivity of CDO to k_0 and ν in K562. Mean CDO across perturbations for six controlled predictions over the full 7×8 parameter grid. The black box marks the selected setting $(k_0, \nu) = (100, 2)$.

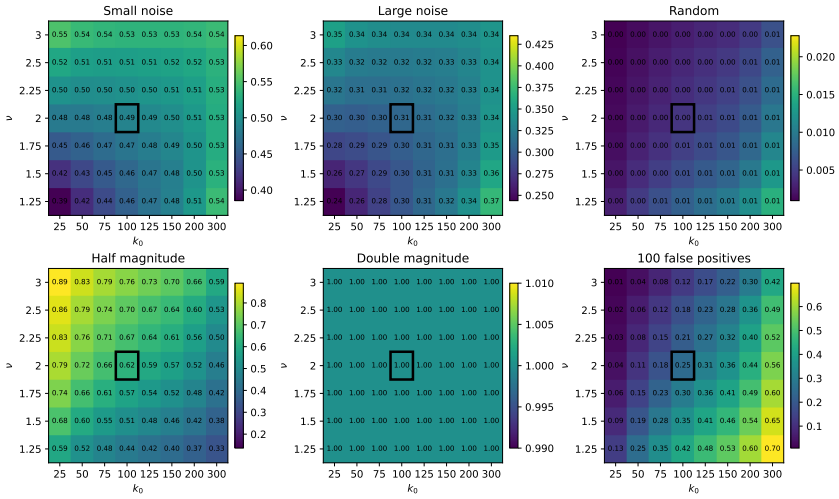


Figure 6: Sensitivity of CDO to k_0 and ν in RPE1. As in Fig. 5, evaluated across RPE1 perturbations. Colour scales for each control are shared between K562 and RPE1 to allow direct comparison.

D.5 Final Parameter Choice

The parameter sweep is intended as a sensitivity analysis rather than a hyperparameter optimisation. There is no single setting that should maximise all controlled scores: increasing rank tolerance raises the score of useful but noisy predictions, but necessarily also raises the score assigned to false-positive ranking errors. Similarly, concentrating the metric more strongly on the highest-ranked responders changes its sensitivity to response-magnitude underestimation. The relevant question is therefore whether the chosen parameters give an interpretable compromise and lie within a broad region of stable behaviour. We select $k_0 = 100$ and $\nu = 2$. These parameters give the simple inverse-square rank penalty

$$r(\Delta k) = \left(1 + \frac{\Delta k}{100}\right)^{-2}, \tag{6}$$

where $\Delta k = |k - \hat{k}|$. Hence, $r(20) = 0.694,\ r(50) = 0.444,\ r(100) = 0.250,\ r(200) = 0.111$ and the displacement receiving half credit is $\Delta k_{1/2} = 100(\sqrt{2} - 1) \simeq 41.4$. Thus, errors of a few tens of ranks retain substantial credit, while moving a strong responder hundreds of positions down the predicted ranking contributes little. The selected setting lies within the smooth central region of the heatmaps for both datasets rather than at an extreme of either parameter axis.

These values are fixed before the search campaigns and held constant throughout all subsequent experiments.

E PREDICTOR SPECIFICATION

To test whether representation quality depends on the downstream readout, we re-evaluate frozen gene representations with several predictors after protocol optimisation. All models use the same archived nine perturbation folds as the primary analysis; neither the representations nor fold assignments are changed. Predictor-specific preprocessing and hyperparameter selection are performed independently within each training fold and then applied to the corresponding held-out perturbations. None of the alternative predictors influences protocol search or selection.

Let $X \in \mathbb{R}^{n \times D}$ denote the training representations in one fold and $Y \in \mathbb{R}^{n \times G}$ the corresponding transcriptome-wide responses.

Linear regression. The primary predictor used during protocol optimisation is unregularised multi-output ordinary least squares,

$$(\widehat{\mathbf{b}}, \widehat{B}) = \arg\min_{\mathbf{b}, B} \left\| Y - \mathbf{1}\mathbf{b}^\top - XB \right\|_F^2, \tag{7}$$

with predictions

$$\widehat{Y}_* = \mathbf{1}\widehat{\mathbf{b}}^\top + X_* \widehat{B}. \tag{8}$$

An intercept is fitted, but the representation features are not otherwise scaled, reduced or regularised.

k-nearest neighbours. For kNN we fix $k = 5$ and identify neighbours by Euclidean distance in the stored representation space. For a held-out representation x_*,

$$\widehat{\mathbf{y}}(x_*) = \frac{1}{|\mathcal{N}_5(x_*)|} \sum_{i \in \mathcal{N}_5(x_*)} \mathbf{y}_i, \tag{9}$$

where $\mathcal{N}_5(x_*)$ contains the five nearest training representations. Neighbours are weighted uniformly, and their complete response vectors are averaged coordinate-wise. No additional scaling or dimensionality reduction is applied.

Gaussian process regression. We use a shared isotropic radial-basis-function kernel

$$k_\theta(x, x') = a \exp\left(-\frac{\|x - x'\|_2^2}{2\ell^2} \right), \tag{10}$$

with training covariance

$$K_\theta = [k_\theta(x_i, x_j)]_{ij} + (\eta + 10^{-9})I. \tag{11}$$

The response dimensions are modelled as conditionally independent zero-mean Gaussian processes sharing the same kernel hyperparameters. The length scale ℓ, amplitude a and diagonal noise η are initialised to 50, 0.05 and 0.05, respectively, and optimised independently within each training fold for 300 Adam steps with learning rate 0.1. Predictions use the posterior mean,

$$\widehat{Y}_* = K_{X,X_*}^\top K_\theta^{-1} Y. \tag{12}$$

No feature or response scaling is applied.

Ridge regression. Ridge regression tests whether regularising the linear readout changes the relative quality of the representations. Within each training fold, features are centred and divided by a single global RMS scale,

$$\widetilde{x}_i = \frac{x_i - \bar{x}}{s}, \qquad s = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|x_i - \bar{x}\|_2^2}. \tag{13}$$

We then fit

$$(\widehat{\mathbf{b}}_\alpha, \widehat{B}_\alpha) = \arg\min_{\mathbf{b},B} \left\| Y - \mathbf{1}\mathbf{b}^\top - \widetilde{X}B \right\|_F^2 + \alpha \|B\|_F^2, \tag{14}$$

with an unpenalised intercept. The regularisation strength is selected independently within each training fold from

$$\alpha \in \{0.01, 0.1, 1, 10, 100\} \tag{15}$$

using leave-one-out prediction error on the training data. A single selected α is shared across all response genes. The training-fold centring, scaling and α selection are then fixed before evaluation on the corresponding held-out perturbations.

PCA followed by linear regression. To test whether performance depends on the high dimensionality of the representation, we also project each training fold onto

$$r = \min(128, n_{\text{train}} - 1, d) \tag{16}$$

principal components. Writing

$$X_c = X - \mathbf{1}\bar{x}^\top \approx U_r S_r V_r^\top, \qquad Z = X_c V_r, \tag{17}$$

we fit unregularised multi-output linear regression to Z and predict held-out perturbations as

$$\widehat{Y}_* = \mathbf{1}\widehat{\mathbf{b}}^\top + \left[(X_* - \mathbf{1}\bar{x}^\top)V_r\right]\widehat{C}. \tag{18}$$

PCA is fitted only on the training fold, without whitening or variance scaling. The number of components is determined solely by the fixed maximum above and is not selected using prediction performance.

Across all post hoc predictors, the representation artifacts remain unchanged and the same held-out perturbations are used as in the primary analysis. Differences in performance therefore test whether gains from the learned representation persist across alternative readout assumptions rather than under a new protocol search.

F BOOTSTRAP UNCERTAINTY AND PAIRED COMPARISONS

Performance estimates are calculated from out-of-fold predictions for held-out perturbations. For each metric, the central estimate is the arithmetic mean of the perturbation-level values over the eligible evaluation panel. Cross-validation folds are used only to construct out-of-fold predictions and are not treated as independent replicates.

Unless stated otherwise, uncertainty is estimated using 10,000 nonparametric bootstrap replicates. Each replicate samples held-out perturbations with replacement from the pooled eligible panel and recomputes the mean. Reported $\pm$ values are bootstrap standard errors, defined as the standard deviation of these bootstrap means. Response genes are not resampled.

Comparisons between representations are paired. Both representations are evaluated on the same perturbations using shared bootstrap draws, producing a paired difference in each replicate. We report the observed mean difference and the 2.5th and 97.5th percentiles of the bootstrap distribution as the paired 95% confidence interval (CI). Matched-subset analyses restrict resampling to the common perturbation panel used by every method in the comparison.

For protocols evaluated using five independently generated description sets, each replicate resamples both generation seeds and held-out perturbations with replacement, then averages over the sampled seed–perturbation combinations. The same sampled seeds and perturbations are shared between representations in paired comparisons. These intervals therefore capture variability across perturbations and description generations for a fixed protocol.

All estimates condition on the saved out-of-fold predictions and frozen perturbation splits. They do not include uncertainty from refitting the predictor, changing the cross-validation split, or repeating the protocol search. In particular, intervals for protocols selected on the same validation folds do not account for selection among the 40 evaluated candidates and should not be interpreted as independent post-selection inference.

G IMPLEMENTATION AND COMPUTE ENVIRONMENT

The pipeline was implemented in Python 3.10 using NumPy/SciPy, pandas, scikit-learn, PyTorch, Transformers, SentenceTransformers and vLLM. Predictor definitions, cross-validation and metric parameters follow Section 4; all predictors use the same nine perturbation folds (split seed 0), and each perturbation receives exactly one out-of-fold prediction. Final matched descriptions were generated locally with `Qwen/Qwen3.8-27B` through a vLLM OpenAI-compatible server at temperature 0, using generation seeds 1001–1005. Descriptions were embedded in batches of 64 with `nvidia/llama-embed-nemotron-8b`, producing 4,096-dimensional L2-normalised float32 vectors. Generation, embedding and evaluation artefacts were cached separately; resumable jobs validated hashes, dimensions and fold coverage before reusing completed stages. The search orchestration used a common proposal interface: GPT-5.6 Sol was run through Codex, whereas Qwen3.8-27B and Gemma-4-12B were served locally through vLLM's OpenAI-compatible API. All local GPU workloads—including search-model inference, description generation, embedding, GPU-based predictors, and the scLAMBDA and CellFlow experiments—ran on NVIDIA H200 NVL GPUs.

H EXTENDED EXPERIMENTAL RESULTS

This appendix reports the extended metrics and paired uncertainty analyses underlying the main-text results. Unless stated otherwise, all evaluations use the frozen nine-fold perturbation splits and CDO configuration described in Section 4. Evaluations on the dataset used to select a protocol are validation results; applications of a frozen protocol to another dataset are cross-dataset tests.

H.1 UNCERTAINTY AND REPORTING CONVENTION

All error bars report the bootstrap standard error of the displayed mean. For a deterministic or single-realisation representation, we draw the eligible held-out perturbations with replacement, recompute the mean, and repeat this 10,000 times. For a protocol evaluated under five description generations, each replicate independently resamples both the five generation seeds and the held-out perturbations, then averages the sampled seed–perturbation combinations. This incorporates perturbation and description-generation variability for a fixed protocol, but not variability from repeating protocol search.

Pairwise intervals use shared bootstrap draws for both representations, so predictions remain paired by perturbation and, where applicable, by generation seed. We report the 2.5th and 97.5th percentiles of the resulting paired differences. The nine cross-validation folds construct out-of-fold predictions but are not treated as replicates or bootstrap strata, and response genes are not resampled. These estimates are conditional on the saved fitted models and frozen splits. For an in-domain protocol chosen using the same validation data, they also do not account for selecting the best of 40 candidates and should not be interpreted as independent post-selection inference.

H.2 INFORMATION AND REPRESENTATION HIERARCHY

Table 7 gives all linear-predictor metrics for the representations used to establish the information hierarchy. The dominant improvement comes from supplying additional biological information, with a smaller further gain from organising the enriched metadata using the canonical description protocol.

The CDO intervals for the NCBI/HGNC versus name and WikiCrow versus NCBI comparisons are positive in all four datasets. Canonical descriptions also increase the CDO point estimate over directly embedded WikiCrow metadata in every dataset; the paired interval excludes zero for RPE1, HepG2 and Jurkat, but not K562. Secondary metrics show the same overall pattern. For Corr@100, the canonical-minus-WikiCrow interval excludes zero in K562 [0.0006, 0.0240], HepG2 [0.0141, 0.0335] and Jurkat [0.0058, 0.0307], but not RPE1 [-0.0054, 0.0227]. For MAE@100, where negative differences favour canonical, it excludes zero in RPE1 [-0.0340, -0.0007], HepG2 [-0.0419, -0.0186] and Jurkat [-0.0330, -0.0120], but not K562 [-0.0109, 0.0019]. Thus, richer information provides the largest and most uniform gain, with additional evidence that structuring the same information improves representation quality.

H.3 SEARCH-MODEL ROBUSTNESS

Each search model receives the same canonical starting protocol and a budget of 40 evaluated candidates. Because the original campaigns use independently generated descriptions, selected protocols are compared only after regenerating them with the same description model, decoding configuration and five generation seeds.

The within-run improvements and the controlled matched comparison answer different questions. Table 9 records the candidate selected within each original campaign relative to that campaign's own canonical generation. It demonstrates optimisation within a run, but should not be used to rank search models. The subsequent tables regenerate all selected protocols under common conditions.

All three selected protocols improve linear CDO over canonical on K562, with paired intervals excluding zero. Their RPE1 linear point estimates are also positive, but the intervals cross zero. In the single GPT-5.6 Sol trajectory displayed in Figure 2, the within-run RPE1 gain is +0.0072 [0.0028, 0.0114]. In that run, Corr@20 changes by +0.0211 [0.0082, 0.0344], Corr@50 by +0.0241

Dataset	Representation	CDO↑	Corr@20↑	Corr@50↑	Corr@100↑	MAE@20↓	MAE@50↓	MAE@100↓	Full MAE↓
K562	Gene name	0.1011 ± 0.0033	0.4057 ± 0.0131	0.4226 ± 0.0123	0.4440 ± 0.0117	0.6963 ± 0.0131	0.6148 ± 0.0106	0.5778 ± 0.0091	0.4537 ± 0.0061
	NCBI/HGNC	0.1171 ± 0.0029	0.4927 ± 0.0116	0.5146 ± 0.0101	0.5338 ± 0.0098	0.6044 ± 0.0111	0.5230 ± 0.0087	0.4870 ± 0.0073	0.3453 ± 0.0028
	+ WikiCrow	0.1283 ± 0.0029	0.5293 ± 0.0109	0.5623 ± 0.0095	0.5851 ± 0.0089	0.5675 ± 0.0105	0.4900 ± 0.0082	0.4564 ± 0.0068	0.3160 ± 0.0021
	Canonical	0.1290 ± 0.0027	0.5535 ± 0.0105	0.5759 ± 0.0092	0.5973 ± 0.0086	0.5599 ± 0.0106	0.4846 ± 0.0085	0.4519 ± 0.0072	0.3125 ± 0.0021
RPE1	Gene name	0.1448 ± 0.0038	0.4983 ± 0.0120	0.5231 ± 0.0113	0.5345 ± 0.0111	1.1406 ± 0.0185	1.0470 ± 0.0163	0.9888 ± 0.0152	0.6262 ± 0.0076
	NCBI/HGNC	0.1710 ± 0.0035	0.5875 ± 0.0103	0.6204 ± 0.0096	0.6301 ± 0.0092	0.9302 ± 0.0130	0.8352 ± 0.0109	0.7799 ± 0.0100	0.4548 ± 0.0031
	+ WikiCrow	0.1831 ± 0.0034	0.6260 ± 0.0097	0.6563 ± 0.0089	0.6680 ± 0.0085	0.8813 ± 0.0116	0.7877 ± 0.0097	0.7362 ± 0.0088	0.4224 ± 0.0022
	Canonical	0.1882 ± 0.0033	0.6351 ± 0.0096	0.6652 ± 0.0086	0.6766 ± 0.0082	0.8624 ± 0.0117	0.7714 ± 0.0094	0.7189 ± 0.0085	0.4126 ± 0.0021
HepG2	Gene name	0.1062 ± 0.0019	0.5462 ± 0.0080	0.5402 ± 0.0076	0.5306 ± 0.0073	1.3573 ± 0.0162	1.2382 ± 0.0145	1.1597 ± 0.0133	0.8068 ± 0.0092
	NCBI/HGNC	0.1269 ± 0.0017	0.6545 ± 0.0061	0.6433 ± 0.0058	0.6281 ± 0.0058	1.0528 ± 0.0095	0.9476 ± 0.0084	0.8804 ± 0.0076	0.5508 ± 0.0032
	+ WikiCrow	0.1328 ± 0.0016	0.6819 ± 0.0056	0.6759 ± 0.0051	0.6612 ± 0.0050	1.0059 ± 0.0089	0.9000 ± 0.0076	0.8329 ± 0.0069	0.5107 ± 0.0022
	Canonical	0.1422 ± 0.0016	0.7015 ± 0.0053	0.6972 ± 0.0048	0.6850 ± 0.0046	0.9673 ± 0.0086	0.8667 ± 0.0073	0.8026 ± 0.0067	0.4934 ± 0.0022
Jurkat	Gene name	0.0611 ± 0.0014	0.3663 ± 0.0086	0.3769 ± 0.0080	0.3814 ± 0.0078	1.0533 ± 0.0123	0.9722 ± 0.0113	0.9359 ± 0.0111	0.7483 ± 0.0089
	NCBI/HGNC	0.0730 ± 0.0013	0.4433 ± 0.0079	0.4556 ± 0.0071	0.4598 ± 0.0069	0.8401 ± 0.0086	0.7513 ± 0.0074	0.7081 ± 0.0068	0.4934 ± 0.0032
	+ WikiCrow	0.0753 ± 0.0013	0.4590 ± 0.0077	0.4752 ± 0.0069	0.4799 ± 0.0065	0.8191 ± 0.0085	0.7295 ± 0.0074	0.6861 ± 0.0068	0.4686 ± 0.0025
	Canonical	0.0804 ± 0.0013	0.4765 ± 0.0074	0.4915 ± 0.0068	0.4981 ± 0.0064	0.7931 ± 0.0080	0.7063 ± 0.0071	0.6639 ± 0.0066	0.4475 ± 0.0024

Table 7: **Complete linear-predictor information hierarchy.** Values are means with bootstrap standard errors. Gene-name and raw-metadata rows resample held-out perturbations; generated-description rows jointly resample five matched generation seeds and held-out perturbations.

Dataset	NCBI − name	WikiCrow − NCBI	Canonical − WikiCrow
K562	+0.0160 [0.0121, 0.0200]	+0.0112 [0.0079, 0.0146]	+0.0007 [-0.0027, 0.0042]
RPE1	+0.0262 [0.0202, 0.0322]	+0.0121 [0.0069, 0.0173]	+0.0052 [0.0002, 0.0103]
HepG2	+0.0207 [0.0178, 0.0236]	+0.0059 [0.0032, 0.0085]	+0.0094 [0.0068, 0.0120]
Jurkat	+0.0118 [0.0095, 0.0142]	+0.0023 [0.0001, 0.0045]	+0.0051 [0.0029, 0.0073]

Table 8: **CDO changes along the information hierarchy.** Each cell reports the candidate-minus-reference difference with its paired 95% bootstrap interval. Comparisons involving canonical descriptions jointly resample five matched generation seeds and held-out perturbations, while the raw WikiCrow representation remains fixed across seed draws. Folds are not resampling strata. These fixed-representation comparisons involve no search-selected protocols.

Dataset	Search model	Canonical CDO	Selected trial	Selected CDO	Difference
K562	GPT-5.6 Sol	0.131617	24	0.134771	+0.003154
RPE1	GPT-5.6 Sol	0.186826	17	0.193987	+0.007161
K562	Qwen3.8-27B	0.129350	31	0.134624	+0.005274
RPE1	Qwen3.8-27B	0.188440	22	0.193271	+0.004831
K562	Gemma-4-12B	0.128641	20	0.135738	+0.007097
RPE1	Gemma-4-12B	0.185909	37	0.190833	+0.004925

Table 9: **Within-run validation improvements for the three search models.** Every campaign completed its fixed 40-candidate budget. Each difference is the selected candidate minus that campaign's independently generated canonical baseline. These rows are not controlled comparisons between search models.

Dataset	Selected protocol	ΔCDO [95% CI]	ΔCorr@100	ΔMAE@100
K562	GPT-5.6 Sol	+0.0040 [0.0009, 0.0071]	+0.0087 ± 0.0054	-0.0058 ± 0.0030
K562	Qwen3.8-27B	+0.0060 [0.0026, 0.0096]	+0.0146 ± 0.0057	-0.0122 ± 0.0035
K562	Gemma-4-12B	+0.0065 [0.0035, 0.0096]	+0.0200 ± 0.0053	-0.0105 ± 0.0030
RPE1	GPT-5.6 Sol	+0.0025 [-0.0018, 0.0068]	+0.0088 ± 0.0056	-0.0112 ± 0.0066
RPE1	Qwen3.8-27B	+0.0030 [-0.0015, 0.0073]	+0.0016 ± 0.0063	-0.0041 ± 0.0072
RPE1	Gemma-4-12B	+0.0009 [-0.0037, 0.0054]	+0.0065 ± 0.0061	-0.0044 ± 0.0074

Table 10: **Matched improvements over canonical under five description-generation seeds.** The CDO column reports mean paired differences with joint seed–perturbation 95% bootstrap intervals; Corr@100 and MAE@100 report mean differences with bootstrap standard errors. Candidate and canonical use the same generator, decoding configuration, seed identities, encoder, folds and predictor. Negative MAE is favourable. The bootstrap conditions on the already-selected protocol and does not represent variation from repeating search.

[0.0123, 0.0364], Corr@100 by +0.0251 [0.0135, 0.0370], and MAE@100 by -0.0380 [-0.0521, -0.0238]. These intervals are conditional on selecting the protocol using the same validation folds.

H.4 Search trajectory and gene-level interpretation

Figure 2 displays one GPT-5.6 Sol search run on RPE1. The accepted revisions progressively add supported mechanistic detail, gene-family specialisation, mechanism-linked phenotype, conditional and cellular context, developmental state and physiological role. Because each accepted protocol becomes the parent of later candidates, these are cumulative steps in one trajectory rather than independent ablations.

CDC42 case study. CDC42 provides an illustrative example of how the description of one gene changes along the displayed search trajectory. The canonical description captures its core identity as a Rho-family GTPase and its roles in actin organisation, polarity and membrane dynamics:

> *CDC42 encodes a small Rho-family GTPase that functions as a molecular switch cycling between active GTP-bound and inactive GDP-bound states. Localized to the plasma membrane via C-terminal prenylation, it regulates actin cytoskeleton organization, specifically promoting filopodia formation, cell polarity, and endocytosis. Key interaction partners include N-WASP, which activates the Arp2/3 complex, and the polarity scaffold Par6. Disrupting CDC42 function impairs actin polymerization and membrane dynamics, leading to defective cell migration, loss of epithelial polarity, and compromised immune cell signaling.*

Across the accepted revisions, the description adds an explicit distinction from related Rho GTPases, mechanism-linked disease phenotypes, neuronal and immune-lineage contexts, and organism-level tissue and haematopoietic roles. The final description is:

> *CDC42 encodes a small GTPase belonging to the Rho family, functioning as a molecular switch that cycles between active GTP-bound and inactive GDP-bound states. It is distinguished by its specific role in regulating actin cytoskeleton organization, particularly filopodia formation, and cell polarity. The protein localizes to the plasma membrane via C-terminal prenylation and interacts with effectors such as N-WASP, Par6, and PAK to coordinate endocytosis, exocytosis, and junction maturation. These activities are critical in epithelial morphogenesis, neuronal development, and immune cell signaling, including T cell homeostasis and B cell differentiation. At the organismal level, CDC42 maintains tissue integrity and hematopoietic function. Mutations are associated with Takenouchi-Kosaki syndrome and NOCARH syndrome, linking defective cytoskeletal remodeling to developmental delay, macrothrombocytopenia, and autoinflammation. Disrupting CDC42 function impairs cell migration, polarity establishment, and immune cell differentiation, leading to compromised tissue architecture and immune dysregulation.*

Description	Words	CDO↑	Corr@100↑	MAE@100↓
Canonical	80	0.1520	0.8317	0.7650
Final selected	140	0.3114	0.9218	0.4647

Table 11: **Gene-level prediction metrics for the CDC42 example.** Values are out-of-fold metrics from the canonical and final descriptions in the single RPE1 search run shown in Figure 2.

The two nearest embedding neighbours remain RAC1 and RAC3, while the remaining top-five neighbours change from GMIP, ARHGAP6 and RAN to GMIP, CDC123 and NCKAP1. Thus, the added contextual information does not erase the core small-GTPase identity. The richer final description coincides with higher CDO and Corr@100 and lower MAE@100 for CDC42 (Table 11). This is a selected validation example from a cumulative search trajectory, not an independent ablation: every accepted stage regenerates all gene descriptions, so the metric changes cannot be attributed causally to any individual addition.

H.5 Transfer across datasets and downstream predictors

H.5.1 Cross-dataset protocol transfer

The source protocol is frozen before application to a target dataset, but the response predictor is fitted using that target dataset's training folds. Consequently, these are tests of protocol transfer

rather than zero-shot transfer of a fitted response model. Main-text Table 1 reports five-seed absolute means. The paired analysis below compares each transferred protocol with canonical descriptions generated under the same five seeds on the target dataset; its intervals should not be reconstructed by subtracting independently generated marginal means.

Five of the six CDO and Corr@100 point differences are positive, while MAE@100 is lower in all six transfers. However, every 95% interval in Table 12 crosses zero. Full-profile MAE provides somewhat stronger evidence: its interval excludes zero for both transfers to HepG2 and for K562 to Jurkat, but not for the other three transfers. The five-seed analysis therefore supports modest reuse of the learned protocol, particularly for reducing prediction error, without establishing uniform transfer across cell contexts.

H.5.2 Comparison across downstream predictors

Strict common-coverage predictor panel. Table 13 extends the main comparison with the linear and PCA–linear readouts and with the canonical and gene-name controls. Every representation within a dataset uses the same perturbations, response genes, folds and predictor configuration.

These comparisons support the main-text claim that both selected text representations outperform every tested established embedding under kNN, GP and Ridge on K562 and RPE1.

H.5.3 Four-dataset Ridge comparison

The strict Ridge panel provides a compact four-dataset check under one regularised readout. It uses 1,010 K562, 1,362 RPE1, 1,977 HepG2 and 1,976 Jurkat perturbations, with identical perturbations and folds across every representation within a dataset.

H.6 Integration with established single-cell architectures

scLAMBDA and CellFlow use GenePT and ESM2 perturbation embeddings, respectively. We replace only these embeddings with K562-selected or RPE1-selected gene-description embeddings while holding the architecture, training procedure, hyperparameters and unseen-perturbation folds fixed.

Tables 16 and 17 report complete Pearson and MAE panels. The four values are calculated over all response genes and the observed top 20, 50 and 100 genes. Top-K genes are selected from the observed response and held fixed across representations. Absolute panels use all 1,085 K562 and 1,529 RPE1 perturbations; the scLAMBDA K562 paired analysis uses the common 965-perturbation subset with finite outputs from every compared representation.

The native-to-text substitution is well supported across both architectures and datasets: every displayed Pearson interval is positive and every MAE interval is negative.

Frozen source	Target	ΔCDO [95% CI]	ΔCorr@100 [95% CI]	ΔMAE@100 [95% CI]
K562	RPE1	-0.0015 [-0.0063, 0.0032]	-0.0052 [-0.0190, 0.0082]	-0.0029 [-0.0187, 0.0127]
K562	HepG2	+0.0022 [-0.0002, 0.0045]	+0.0071 [-0.0018, 0.0161]	-0.0096 [-0.0202, 0.0007]
K562	Jurkat	+0.0006 [-0.0015, 0.0027]	+0.0052 [-0.0065, 0.0167]	-0.0050 [-0.0143, 0.0044]
RPE1	K562	+0.0015 [-0.0014, 0.0044]	+0.0082 [-0.0012, 0.0177]	-0.0038 [-0.0092, 0.0014]
RPE1	HepG2	+0.0007 [-0.0015, 0.0028]	+0.0017 [-0.0064, 0.0098]	-0.0035 [-0.0127, 0.0055]
RPE1	Jurkat	+0.0004 [-0.0016, 0.0023]	+0.0070 [-0.0040, 0.0176]	-0.0069 [-0.0159, 0.0022]

Table 12: **Five-seed paired tests of frozen protocol transfer.** Each row reports transferred minus target-matched canonical. Protocols are regenerated under five matched seeds; intervals are 95% percentiles from 10,000 joint seed–perturbation bootstrap draws shared across protocols. Positive ΔCDO and ΔCorr and negative ΔMAE favour the transferred protocol. Folds are used only to align out-of-fold predictions and are not resampled.

Dataset	Representation	Linear	kNN	GP	Ridge	PCA–linear
K562	Gene name	0.1041 ± 0.0034	0.1613 ± 0.0037	0.1557 ± 0.0036	0.1416 ± 0.0027	0.1585 ± 0.0036
	GenePT	0.0886 ± 0.0026	0.1770 ± 0.0036	0.1654 ± 0.0035	0.1474 ± 0.0025	0.1677 ± 0.0033
	ESM2	0.1163 ± 0.0025	0.1221 ± 0.0026	0.1378 ± 0.0023	0.1293 ± 0.0020	0.1462 ± 0.0027
	scGPT	0.0851 ± 0.0020	0.1338 ± 0.0029	0.1340 ± 0.0024	0.1338 ± 0.0024	0.1364 ± 0.0026
	Geneformer	0.0783 ± 0.0020	0.1136 ± 0.0023	0.1134 ± 0.0016	0.1110 ± 0.0014	0.1200 ± 0.0023
	UniProt	0.0410 ± 0.0019	0.1500 ± 0.0030	0.1414 ± 0.0025	0.1372 ± 0.0024	0.1482 ± 0.0028
	Canonical	0.1321 ± 0.0030	0.1925 ± 0.0037	0.1824 ± 0.0035	0.1623 ± 0.0027	0.1824 ± 0.0034
	K562-selected	0.1349 ± 0.0029	0.1929 ± 0.0036	0.1834 ± 0.0035	0.1620 ± 0.0027	0.1829 ± 0.0033
	RPE1-selected	0.1319 ± 0.0030	0.1912 ± 0.0037	0.1823 ± 0.0035	0.1599 ± 0.0026	0.1825 ± 0.0033
RPE1	Gene name	0.1523 ± 0.0041	0.2784 ± 0.0036	0.3016 ± 0.0035	0.2994 ± 0.0034	0.2880 ± 0.0035
	GenePT	0.1109 ± 0.0033	0.2891 ± 0.0037	0.3042 ± 0.0035	0.3007 ± 0.0034	0.2902 ± 0.0035
	ESM2	0.2029 ± 0.0034	0.2667 ± 0.0034	0.2953 ± 0.0034	0.2952 ± 0.0034	0.2859 ± 0.0034
	scGPT	0.2086 ± 0.0034	0.2677 ± 0.0034	0.2972 ± 0.0034	0.2972 ± 0.0034	0.2832 ± 0.0035
	Geneformer	0.2005 ± 0.0032	0.2565 ± 0.0032	0.2910 ± 0.0033	0.2908 ± 0.0033	0.2728 ± 0.0033
	UniProt	0.0974 ± 0.0033	0.2743 ± 0.0034	0.2965 ± 0.0034	0.2966 ± 0.0034	0.2834 ± 0.0034
	Canonical	0.1960 ± 0.0037	0.2953 ± 0.0037	0.3111 ± 0.0036	0.3068 ± 0.0035	0.2993 ± 0.0036
	K562-selected	0.1974 ± 0.0036	0.2965 ± 0.0037	0.3109 ± 0.0036	0.3062 ± 0.0035	0.2984 ± 0.0036
	RPE1-selected	0.2022 ± 0.0035	0.2948 ± 0.0037	0.3103 ± 0.0035	0.3062 ± 0.0035	0.2988 ± 0.0036

Table 13: **Complete strictly matched CDO panel for K562 and RPE1.** The common panels contain 1,010 K562 and 1,362 RPE1 perturbations. Entries are mean CDO with perturbation-bootstrap standard errors. PCA is fitted using only the training folds.

Dataset	Predictor	Candidate	GenePT	ESM2	scGPT	Geneformer	UniProt
K562	kNN	K562-selected	+0.0159 [0.0121, 0.0196]	+0.0708 [0.0648, 0.0769]	+0.0591 [0.0528, 0.0653]	+0.0793 [0.0727, 0.0861]	+0.0429 [0.0378, 0.0481]
		RPE1-selected	+0.0141 [0.0107, 0.0176]	+0.0691 [0.0628, 0.0753]	+0.0574 [0.0510, 0.0638]	+0.0776 [0.0708, 0.0845]	+0.0412 [0.0359, 0.0465]
	GP	K562-selected	+0.0180 [0.0150, 0.0209]	+0.0456 [0.0406, 0.0504]	+0.0494 [0.0441, 0.0547]	+0.0700 [0.0639, 0.0761]	+0.0420 [0.0374, 0.0466]
		RPE1-selected	+0.0169 [0.0142, 0.0197]	+0.0445 [0.0396, 0.0495]	+0.0483 [0.0431, 0.0536]	+0.0689 [0.0629, 0.0749]	+0.0409 [0.0362, 0.0456]
	Ridge	K562-selected	+0.0146 [0.0126, 0.0166]	+0.0327 [0.0295, 0.0359]	+0.0282 [0.0249, 0.0316]	+0.0510 [0.0469, 0.0551]	+0.0248 [0.0219, 0.0278]
		RPE1-selected	+0.0124 [0.0105, 0.0143]	+0.0305 [0.0275, 0.0336]	+0.0261 [0.0229, 0.0293]	+0.0488 [0.0449, 0.0529]	+0.0227 [0.0199, 0.0255]
RPE1	kNN	K562-selected	+0.0075 [0.0041, 0.0109]	+0.0299 [0.0251, 0.0346]	+0.0288 [0.0239, 0.0337]	+0.0401 [0.0349, 0.0453]	+0.0222 [0.0178, 0.0266]
		RPE1-selected	+0.0057 [0.0027, 0.0088]	+0.0281 [0.0236, 0.0327]	+0.0270 [0.0223, 0.0317]	+0.0383 [0.0331, 0.0435]	+0.0205 [0.0162, 0.0247]
	GP	K562-selected	+0.0067 [0.0046, 0.0088]	+0.0156 [0.0125, 0.0186]	+0.0137 [0.0107, 0.0167]	+0.0199 [0.0165, 0.0232]	+0.0144 [0.0113, 0.0174]
		RPE1-selected	+0.0061 [0.0042, 0.0081]	+0.0150 [0.0119, 0.0179]	+0.0131 [0.0101, 0.0161]	+0.0193 [0.0159, 0.0226]	+0.0138 [0.0108, 0.0168]
	Ridge	K562-selected	+0.0055 [0.0040, 0.0070]	+0.0110 [0.0091, 0.0129]	+0.0090 [0.0070, 0.0110]	+0.0154 [0.0133, 0.0175]	+0.0096 [0.0077, 0.0115]
		RPE1-selected	+0.0054 [0.0041, 0.0068]	+0.0110 [0.0091, 0.0128]	+0.0090 [0.0071, 0.0109]	+0.0154 [0.0133, 0.0175]	+0.0096 [0.0077, 0.0114]

Table 14: **Paired CDO differences between selected text and established gene embeddings.** Each cell is selected text minus the column representation followed by its paired 95% perturbation-bootstrap interval. All comparisons use the strict common perturbation panel and 10,000 shared draws. Every displayed interval excludes zero.

Reference representation	K562	RPE1	HepG2	Jurkat
Gene name	+0.0207 [0.0182, 0.0232]	+0.0074 [0.0057, 0.0090]	+0.0089 [0.0074, 0.0104]	+0.0050 [0.0043, 0.0057]
GenePT	+0.0148 [0.0127, 0.0170]	+0.0061 [0.0047, 0.0075]	+0.0064 [0.0053, 0.0076]	+0.0042 [0.0036, 0.0048]
ESM2	+0.0329 [0.0296, 0.0364]	+0.0116 [0.0097, 0.0134]	+0.0098 [0.0082, 0.0115]	+0.0061 [0.0052, 0.0069]
scGPT	+0.0285 [0.0250, 0.0321]	+0.0096 [0.0077, 0.0116]	+0.0096 [0.0079, 0.0112]	+0.0069 [0.0060, 0.0079]
Geneformer	+0.0513 [0.0470, 0.0557]	+0.0160 [0.0138, 0.0181]	+0.0218 [0.0192, 0.0245]	+0.0091 [0.0079, 0.0103]
UniProt	+0.0251 [0.0220, 0.0283]	+0.0102 [0.0083, 0.0121]	+0.0067 [0.0052, 0.0082]	+0.0057 [0.0049, 0.0065]

Table 15: **Canonical rich text versus reference representations under Ridge.** Each cell is canonical rich-text CDO minus the named reference, followed by a paired 95% perturbation-bootstrap interval. All intervals use 10,000 shared draws and exclude zero.

Model	Dataset	Representation	All	@20	@50	@100
scLAMBDA	K562	Native GenePT	0.5353 ± 0.0089	0.6821 ± 0.0110	0.6922 ± 0.0097	0.6920 ± 0.0095
		K562-selected text	0.5958 ± 0.0076	0.7499 ± 0.0089	0.7586 ± 0.0081	0.7568 ± 0.0077
		RPE1-selected text	0.5920 ± 0.0077	0.7415 ± 0.0092	0.7523 ± 0.0082	0.7515 ± 0.0078
	RPE1	Native GenePT	0.6593 ± 0.0072	0.7602 ± 0.0081	0.7631 ± 0.0077	0.7611 ± 0.0075
		K562-selected text	0.6791 ± 0.0070	0.7837 ± 0.0077	0.7861 ± 0.0074	0.7837 ± 0.0072
		RPE1-selected text	0.6778 ± 0.0070	0.7803 ± 0.0078	0.7835 ± 0.0074	0.7818 ± 0.0072
CellFlow	K562	Native ESM2	0.4067 ± 0.0066	0.5516 ± 0.0098	0.5738 ± 0.0086	0.5766 ± 0.0080
		K562-selected text	0.4622 ± 0.0068	0.6069 ± 0.0096	0.6310 ± 0.0084	0.6337 ± 0.0079
		RPE1-selected text	0.4557 ± 0.0068	0.5995 ± 0.0096	0.6239 ± 0.0085	0.6268 ± 0.0079
	RPE1	Native ESM2	0.6183 ± 0.0069	0.7172 ± 0.0085	0.7245 ± 0.0079	0.7241 ± 0.0076
		K562-selected text	0.6201 ± 0.0069	0.7212 ± 0.0084	0.7276 ± 0.0078	0.7270 ± 0.0076
		RPE1-selected text	0.6197 ± 0.0069	0.7208 ± 0.0084	0.7273 ± 0.0078	0.7266 ± 0.0076

Table 16: **Complete perturbation-effect Pearson panel for scLAMBDA and CellFlow.** Entries are perturbation-macro means with bootstrap standard errors from 10,000 perturbation resamples. Values are calculated on predicted versus observed mean log-normalised expression changes relative to control; higher is better. Paired comparisons use common finite perturbation subsets and are reported separately.

Model	Dataset	Representation	All	@20	@50	@100
scLAMBDA	K562	Native GenePT	0.0375 ± 0.0004	0.2659 ± 0.0051	0.2083 ± 0.0039	0.1730 ± 0.0031
		K562-selected text	0.0357 ± 0.0004	0.2347 ± 0.0045	0.1840 ± 0.0034	0.1533 ± 0.0027
		RPE1-selected text	0.0359 ± 0.0004	0.2389 ± 0.0045	0.1873 ± 0.0034	0.1560 ± 0.0027
	RPE1	Native GenePT	0.0533 ± 0.0004	0.3722 ± 0.0054	0.2973 ± 0.0041	0.2483 ± 0.0034
		K562-selected text	0.0517 ± 0.0004	0.3419 ± 0.0051	0.2739 ± 0.0039	0.2293 ± 0.0032
		RPE1-selected text	0.0517 ± 0.0004	0.3453 ± 0.0051	0.2765 ± 0.0039	0.2315 ± 0.0032
CellFlow	K562	Native ESM2	0.0418 ± 0.0004	0.3286 ± 0.0049	0.2546 ± 0.0038	0.2096 ± 0.0030
		K562-selected text	0.0403 ± 0.0004	0.3090 ± 0.0048	0.2382 ± 0.0037	0.1958 ± 0.0030
		RPE1-selected text	0.0405 ± 0.0004	0.3104 ± 0.0048	0.2396 ± 0.0037	0.1971 ± 0.0030
	RPE1	Native ESM2	0.0579 ± 0.0005	0.4391 ± 0.0061	0.3485 ± 0.0047	0.2895 ± 0.0039
		K562-selected text	0.0572 ± 0.0005	0.4276 ± 0.0060	0.3395 ± 0.0046	0.2820 ± 0.0038
		RPE1-selected text	0.0571 ± 0.0005	0.4267 ± 0.0060	0.3389 ± 0.0046	0.2814 ± 0.0038

Table 17: **Complete perturbation-effect MAE panel for scLAMBDA and CellFlow.** Entries are perturbation-macro means with bootstrap standard errors from 10,000 perturbation resamples; lower is better. All other conventions follow Table 16.

Model	Dataset	Candidate − native	ΔPearson	ΔPearson@100	ΔMAE	ΔMAE@100
scLAMBDA	K562	Canonical text	+0.0608 [0.0525, 0.0693]	+0.0625 [0.0524, 0.0728]	-0.00159 [-0.00186, -0.00133]	-0.0177 [-0.0201, -0.0153]
		K562-selected text	+0.0641 [0.0553, 0.0733]	+0.0676 [0.0568, 0.0783]	-0.00164 [-0.00192, -0.00136]	-0.0185 [-0.0211, -0.0160]
	RPE1	Canonical text	+0.0199 [0.0166, 0.0233]	+0.0217 [0.0175, 0.0264]	-0.00167 [-0.00200, -0.00134]	-0.0176 [-0.0208, -0.0146]
		RPE1-selected text	+0.0184 [0.0153, 0.0217]	+0.0206 [0.0164, 0.0251]	-0.00159 [-0.00192, -0.00127]	-0.0169 [-0.0200, -0.0139]
CellFlow	K562	Canonical text	+0.0535 [0.0463, 0.0610]	+0.0533 [0.0437, 0.0629]	-0.00142 [-0.00164, -0.00120]	-0.0135 [-0.0156, -0.0114]
		K562-selected text	+0.0555 [0.0484, 0.0630]	+0.0572 [0.0474, 0.0669]	-0.00146 [-0.00169, -0.00124]	-0.0139 [-0.0159, -0.0118]
	RPE1	Canonical text	+0.0015 [0.0007, 0.0023]	+0.0024 [0.0013, 0.0034]	-0.00072 [-0.00097, -0.00047]	-0.0068 [-0.0094, -0.0043]
		RPE1-selected text	+0.0014 [0.0005, 0.0023]	+0.0025 [0.0014, 0.0036]	-0.00084 [-0.00109, -0.00058]	-0.0082 [-0.0108, -0.0055]

Table 18: **Paired changes relative to each architecture's native embedding.** Each cell is text minus native followed by a paired 95% perturbation-bootstrap interval. Positive Pearson and negative MAE differences favour text. Every displayed interval excludes zero.