Diversity by Design: Addressing Mode Collapse Improves scRNA-seq Perturbation Modeling on Well-Calibrated Metrics

Gabriel M. Mejia [*1] **Henry E. Miller** [*1] **Francis J. A. Leblanc** [1] **Bo Wang** [2] **Brendan Swain** [1]
Lucas Paulo de Lima Camillo [1]

Abstract

Recent benchmarks reveal that models for single-cell perturbation response are often outperformed by simply predicting the dataset mean. We trace this anomaly to a metric artifact: control-referenced deltas and unweighted error metrics reward mode collapse whenever the control is biased or the biological signal is sparse. Large-scale *in silico* simulations and analysis of two real-world perturbation datasets confirm that shared reference shifts, not genuine biological change, drives high performance in these evaluations. We introduce differentially expressed gene (DEG)–aware metrics, weighted mean-squared error (WMSE) and weighted delta R^2 ($R^2_w(\Delta)$) with respect to all perturbations, that measure error in niche signals with high sensitivity. We further introduce negative and positive performance baselines to calibrate these metrics. With these improvements, the mean baseline sinks to null performance while genuine predictors are correctly rewarded. Finally, we show that using WMSE as a loss function reduces mode collapse and improves model performance.

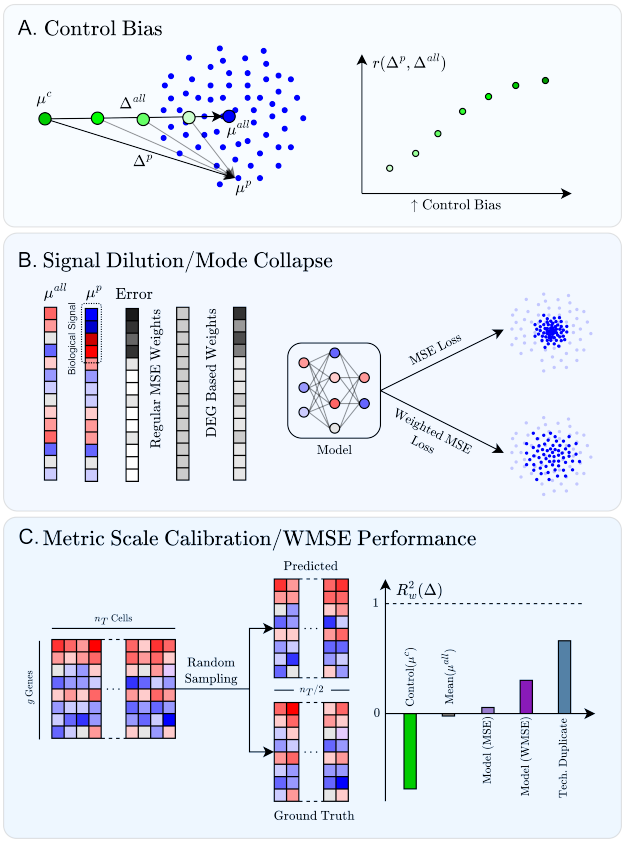

Figure 1. (A) Under increasing amounts of systematic control bias, the Pearson correlation between Δ^p and Δ^{all} increases artificially. (B) True biological signal is diluted in scRNA-seq data causing mode collapse in model predictions. Introducing biologically aware weights (WMSE) ameliorates this problem. (C) We introduce metrics with a performance scale calibrated by negative and positive baselines. We also find that WMSE as a training objective improves model performance compared to MSE.

1. Introduction

In recent years, technological breakthroughs in experimental methodologies have catalyzed the emergence of large-scale, publicly available, single-cell RNA sequencing (scRNA-seq) perturbation datasets (Peidli et al., 2024), which capture phenotypic changes of individual cells under specific perturbations. Models trained on these datasets to predict perturbation responses may unlock virtual molecule and genetic perturbation screening capabilities, which could yield novel therapeutics that reverse disease and restore cell function. In recent years, researchers have proposed a diverse array of perturbation-response models leveraging various learning paradigms, including optimal transport (CellOracle (Kamimoto et al., 2023)), prior knowledge graph learning (GEARS (Roohani et al., 2024)), and transformer-based foundation models (scGPT (Cui et al., 2024), scFoundation

[*]Equal contribution [1]Shift Bioscience, Cambridge, UK [2]University of Toronto, Vector Institute, Toronto, Canada. Correspondence to: Lucas Paulo de Lima Camillo <lucas@shiftbioscience.com>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

(Hao et al., 2024)).

However, recent benchmarking studies have reported that naïve predictors such as linear models often outperform these sophisticated architectures (Li et al., 2024a; Wu et al., 2024; Csendes et al., 2025; Li et al., 2024b; Bendidi et al., 2024; Wenteler et al., 2024; Ahlmann-Eltze et al., 2024). Even more concerning, a mean baseline, consisting of predicting the average of all perturbed cells in the training set (disregarding any individual perturbation label), not only achieves high performance in current metrics but actually outperforms most deep learning architectures without any learning taking place (see 2.3 for examples).

Motivated by these troubling results, our goal in this work was to answer a simple question: "*why does the mean baseline perform well on standard perturbation response model metrics?*". To answer this, we used simulated data as a discovery tool to understand the different dataset factors that can influence mean baseline performance and then confirmed our findings on two of the most commonly used benchmarking datasets (Norman et al., 2019; Replogle et al., 2022). We highlight the impact of systematic control bias (Fig. 1A) on inflating mean baseline performance and the impact of signal dilution (Fig. 1B) on contributing to mode collapse in model predictions. To address these issues, we introduce Weighted MSE (WMSE) and $R^2_w(\Delta)$, alternative metrics that are sensitive to differentially-expressed gene (DEG) signals against all perturbations (not control), and give null performance under mode collapse to the mean. We calibrate these metrics by introducing negative and positive baselines, including a novel technical duplicate baseline (Fig. 1C) which gives a realistic estimate of optimal performance given the intrinsic variance of the dataset. Moreover, we show that WMSE can be used as a training loss to prevent mode collapse and improve model performance (Fig. 1C).

Together, these advances allow more transparent assessment of perturbation-response models and provide a general strategy for improving model performance. The code to replicate our results is available at `https://github.com/shiftbioscience/ICML2025_pertmodel_metrics`.

2. Background

2.1. Single-cell Perturbation Data

Single-cell RNA sequencing (scRNA-seq) measures the abundance of RNA transcripts in thousands-to-millions of individual cells, producing a gene-by-cell count matrix $\mathbf{X} \in \mathbb{N}^{g \times n_T}$. Owing to low capture efficiency and drop-out, $\mathbf{X}$ is sparse (typically $> 90\%$ zeros), and is modeled well by a negative-binomial distribution. Standard pre-processing includes library-size normalization, log1p transformation, and feature selection of $\sim$ 2–5k most highly variable genes.

Large-scale perturbation screens combine CRISPR or chemical interventions with scRNA-seq, yielding paired control–perturbation observations suitable for supervised learning. The screening technology Perturb-seq knocks out, represses (CRISPRi) or activates (CRISPRa) target genes prior to measurement of gene expression (Dixit et al., 2016). Perturb-seq datasets are often used to train perturbation-response models that aim to generalize to unseen perturbations or tissue types, with the hope of enabling large-scale *in silico* screens where millions of perturbations can be tested without the need for costly and time-consuming wet lab experiments.

2.2. Perturbation-Response Models

Predictive models fall into four archetypes. (i) Simple linear baselines: ridge or principal-component regression that extrapolate additively from control and single-perturbation means. (ii) Autoencoder-based models: scGen, CPA and scVI fine-tune autoencoders to encode a cell and an intervention; counterfactuals are obtained by vector arithmetic in latent space (Lotfollahi et al., 2019; 2023; Lopez et al., 2018); (iii) Prior knowledge graph learning: GEARS learns gene embeddings on a co-expression graph and perturbation embeddings on a gene-ontology graph, then decodes their interaction to predict expression shifts (Roohani et al., 2024). (iv) Transformer-based foundation models: scGPT (Cui et al., 2024) and scFoundation (Hao et al., 2024) pretrain on millions of cells and adapt to perturbation tasks via conditioning tokens. Although every paradigm reflects key assumptions and inductive biases, they all strive to learn a conditional generation function $\hat{\mathbf{X}}^p = f(p, \mathbf{X}^\mathbf{c})$ that predicts how the control cell population $\mathbf{X}^\mathbf{c}$ would respond to perturbation p.

2.3. Pitfalls of Common Performance Metrics

Performance is generally quantified at the "pseudobulk" (aggregation of single cells) level after averaging all ground truth and predicted cell profiles per perturbation. This aggregation helps with the sparsity of scRNA-seq data and turns the task into an average effect prediction problem. Although numerous metrics have been introduced, the two most reported ones are MSE (or MAE) and Pearson(Δ). While MSE is defined as the average L_2 error, Pearson(Δ) aspires to capture perturbation effects with respect to a control mean profile μ^c. For a single perturbation with average profile μ^p and predicted profile $\hat{\mu}^p$, Pearson(Δ) is defined as $r(\mu^p - \mu^c, \hat{\mu}^p - \mu^c) = r(\Delta^p, \hat{\Delta}^p)$, which aims to assess whether predicted changes from control are in the same direction as in real data.

Although intuitive, Pearson(Δ) faces two main limitations: (i) it does not consider the scale of change and hence predictions that do not capture the dynamic range of the true

Table 1. Parameter space used for simulation and real data experiments.

Parameter	Effect	Range Simulation	Range *Norman19*	Range *Replogle22*
g	Number of genes in the dataset	Linear: $1000 - 8192$	$\log_2$: $2 - 8192$	$\log_2$: $2 - 8192$
n_0	Number of control cells	log: $10 - 8192$	$\log_2$: $1 - 8192$	$\log_2$: $1 - 8192$
n_p	Number of cells per perturbation	log: $10 - 256$	$\log_2$: $2 - 256$	$\log_2$: $2 - 64$
k	Number of perturbations per dataset	log: $10 - 2000$	$\log_2$: $1 - 175$	$\log_2$: $1 - 1334$
β	Amount of systemic bias in the control	Linear: $0 - 2$	Linear: $0 - 2$	Linear: $0 - 2$
δ	Probability of perturbing a gene	Linear: $0.001 - 0.1$	Quantiles: $0 - 1$	Quantiles: $0 - 1$
ϵ	Multiplicative effect of perturbations	log: $1.2 - 5.0$	Quantiles: $0 - 1$	Quantiles: $0 - 1$
μ_l	Library size scaling (Data quality)	log: $0.2 - 5.0$	Deciles: $0 - 1$	Deciles: $0 - 1$

Δ^p signal can still perform well on this metric, and (ii) it heavily relies on the definition of a control population which might be biased depending on the dataset or the perturbation type. If that is the case, a mean baseline will perform quasi-optimally because the effect of having any perturbation dominates over the unique effect of a specific perturbation (Fig. 1A). On the other hand, the MSE suffers from signal dilution. As illustrated in Fig. 1B, MSE treats every error equally. Thus, it provides low error estimates for predictions that fail to capture important, low-dimensional biological changes but accurately track the general data distribution (e.g., the mean baseline). To address these problems, prior work has computed these metrics only on a subset of perturbation-specific DEGs, genes which change expression significantly as a result of a perturbation when compared to a control population. However, if used as an optimization approach, this requires sparse supervision which causes issues with genes not commonly recognized as DEGs (e.g., a gene only detected once as upregulated in a single training perturbation will systematically be learned as highly expressed). And second, this filtering makes performance metrics blind to true negatives (perturbations with no DEGs).

Multiple recent benchmarks that employ these and other metrics have independently found that predicting the mean baseline (the mean of all perturbed cells; μ^{all}) often matches or surpasses state-of-the-art models. Li et al. (2024a) assessed ten methods across multiple modeling tasks where the mean baseline achieved the lowest MAE(Δ) and nearly the highest Pearson(Δ). Analyzing four Perturb-seq datasets, Csendes et al. (2025) found that the mean baseline exceeds scGPT and scFoundation on Pearson(Δ). Wenteler et al. (2024) showed that the mean baseline tracks top-20 DEG effects as closely as scGPT, Geneformer, or UCE (Cui et al., 2024; Theodoris et al., 2023; Rosen et al., 2023). Finally, Ahlmann-Eltze et al. (2024) reported that four foundation and two deep-learning models fail to beat a mean baseline in evaluations across multiple datasets.

3. Methods

3.1. *In silico* Simulations

We model synthetic datasets containing n_0 control cells and k perturbations with a constant number of n_p observed cells per perturbation. Each cell is represented by a random raw count vector $X \in \mathbb{R}^g$ where each component represents the observed expression of a single gene in that cell under a unique perturbation. Mathematically, we model the expression value $X^p_{i,j}$ of the i^{th} gene of perturbation p in the specific cell j as a negative binomial with a fixed per-gene dispersion and variable mean:

$$X^p_{i,j} \sim \mathrm{NB}\left(\mu^p_{i,j}, \theta_i\right) \tag{1}$$

Where θ_i is the fixed dispersion of gene i and $\mu^p_{i,j}$ captures the simulated perturbation effects as follows:

$$\text{Perturbations:} \quad \mu^p_{i,j} = l_j \alpha^p_i (\mu^c_i + \beta \lambda_i) \tag{2}$$

$$\text{Control:} \quad \mu^c_{i,j} = l_j \mu^c_i \tag{3}$$

μ^c_i being the average of control expression, λ_i a scalar symbolizing a realistic systematic bias between the control population and all other perturbations (only depends on the gene), β a global dataset parameter controlling the severity to which λ_i is applied (zero for a perfectly centered control), α^p_i a multiplicative effect on gene i associated with perturbation p, and l_j the library size component and affects every gene of the cell j equally. Both α^p_i and l_j are random variables by themselves distributed as shown:

$$\alpha^p_i \sim \begin{cases} 1, & P = 1 - \delta \\ 1/\epsilon, & P = \delta/2 \\ \epsilon, & P = \delta/2 \end{cases} \tag{4}$$

$$l_j \sim \mathrm{LogNormal}(\mu_l, \sigma_l^2) \tag{5}$$

Here, δ represents the average probability of perturbing a gene, $\epsilon > 1$ the strength of the effect, and μ_l, σ_l^2 the mean

and variance of the library size scaling factor. All $\delta, \epsilon, \mu_l, \sigma_l^2$ are constant for the entire dataset. Following reasonable priors, we define $\lambda_i = \mu_i^{all} - \mu_i^c$ as the difference between the average perturbed expression and the control expression in the *Norman19* dataset which is also used to estimate θ_i, μ_i^c, and σ_l^2. Given this setup, we perform random sampling to generate an array of synthetic datasets from the parameter space in Table 1. Following standard processing, every synthetic dataset is library-size normalized to 10^4 counts per cell and log1p transformed. We generated 10^4 synthetic datasets and evaluated Pearson(Δ) and MSE on 4 gene sets: (1) all genes, (2) affected genes, which reflects the true simulated perturbed genes ($\alpha_i^p \neq 1$), (3) observed DEGs vs control and (4) observed DEGs vs the rest of the perturbations.

3.2. Real Data Experiments

To evaluate the realism of our simulated results, we created analogous experiments in real-world data. We processed and analyzed two datasets commonly used in benchmarks: (1) *Norman19*, a CRISPRa Perturb-seq dataset with genes activated alone or in combos of two (Norman et al., 2019) and (2) *Replogle22*, a genome-wide CRISPRi Perturb-seq dataset (Replogle et al., 2022). Datasets were randomly downsampled such that each perturbation label had the same number of cells (256 for *Norman19* and 64 for *Replogle22*). For both datasets, we selected the top 8192 highly-variable genes using the `highly_variable_genes` function from the scanpy package (Wolf et al., 2018). We then used the `rank_genes_groups` function from scanpy with the `t-test_overestim_var` method to calculate DEGs with respect to the control cells (DEGs vs Control) and with respect to all other perturbations (DEGs vs Rest). A detailed description of the experiments performed on real data is provided in Appendix A and high-level parameter ranges are available in Table 1.

3.3. Proposed Metrics

3.3.1. Weighted Delta R^2: $R_w^2(\Delta)$

Given a set of positive weights $\{w_i\}$ that add to one, average perturbed expression levels $\{\mu_i^{all}\}$, ground truth expression levels $\{\mu_i^p\}$ and pseudobulked predicted values $\{\hat{\mu}_i^p\}$, $i \in \{1, 2, \ldots, g\}$ with g the number of genes in the dataset, we define $R_w^2(\Delta)$ for a single perturbation as follows:

$$R_w^2(\Delta) = 1 - \frac{\sum_i w_i(\Delta_i - \hat{\Delta}_i)^2}{\sum_i w_i(\Delta_i - \bar{\Delta}_w)^2} \tag{6}$$

$$\bar{\Delta}_w = \sum_{i=1}^{g} w_i \Delta_i \tag{7}$$

Where $\Delta_i = \mu_i^p - \mu_i^{all}$ and $\hat{\Delta}_i = \hat{\mu}_i^p - \mu_i^{all}$ represent the real and predicted changes from the average of all perturbed cells respectively. Note that reference values for delta computation μ_i^{all} are the center of all perturbed cells in the dataset instead of the traditional definition which computes against the control population μ_i^c. We propose $R_w^2(\Delta)$ as a significantly more stringent alternative to Pearson(Δ) with the following four advantages. (i) As a goodness of fit metric, the scale and dynamic range of the predictions does matter. It is not enough to estimate the direction of change as with Pearson(Δ). (ii) Because we set the reference to the mean of all perturbed cells in the dataset, there is, by definition, no systematic bias that can inflate metrics unintentionally. (iii) Because of the properties of R^2, any constant average predictions ($\hat{\mu}^p = \mu^{all}$) will yield a strictly negative result for any specific perturbation (see Appendix B for derivation). (iv) This metric, while still computing in full transcriptomic space, can prioritize more biologically significant genes like DEGs by changing the weights definition.

3.3.2. Weighted Error: WMSE

We propose to evaluate the perturbation prediction task with a modified version of the classical MSE regression metric defined as follows for a single perturbation:

$$\text{WMSE} = \sum_{i=1}^{g} w_i(\mu_i^p - \hat{\mu}_i^p)^2 \tag{8}$$

While simple, this modification of regular MSE addresses the main pitfalls of error metrics currently used in the task (see 2.3). Unlike standard MSE, WMSE allows for gene signal prioritization, such as for perturbation-specific DEGs. In other words, WMSE is more sensitive to perturbation-specific signals, which is particularly important given that only a small proportion of genes change meaningfully in response to each perturbation (Nadig et al., 2025). Moreover, WMSE can directly replace MSE as training loss for many models allowing for biologically meaningful supervision.

3.3.3. Weights Definition

Although the weight set $\{w_1, w_2, \ldots, w_g\}$ may be arbitrarily chosen to highlight any biologically relevant signal on the data, here we choose the weights to prioritize perturbation-specific DEGs. This intuitively assigns higher importance to genes that change meaningfully in a perturbation while still allowing consideration of the whole transcriptome. Our weight computation procedure for a single perturbation is the following: (i) we determine $t-$scores for every gene with respect to the rest of the perturbed cells in the dataset (using scanpy's `sc.tl.rank_genes_groups` (Wolf et al., 2018) function with `method='t-test_overestim_var'` and `reference='rest'`) (ii) we apply an absolute value transformation, (iii) we perform min-max normalization to

Table 2. Pearson correlations between metric performance of the mean baseline ($\hat{\mu}^p = \mu^{all}$) and dataset parameters in simulation experiments. Correlation values lower than -0.2 are highlighted in **blue** and higher than 0.2 are highlighted in **red**.

Metric	Gene Group	ϵ	g	n_p	β	δ	n_0	k	μ_l
Pearson(Δ) ($\uparrow$)	All	**-0.32**	0.00	**0.28**	**0.54**	**-0.28**	**-0.48**	-0.10	0.03
	Affected	**-0.25**	-0.08	0.06	0.12	-0.09	-0.14	**-0.68**	-0.01
	DEGs vs Control	**-0.50**	-0.03	0.00	**0.45**	**-0.28**	**-0.39**	**-0.23**	-0.13
	DEGs vs Rest	**-0.33**	-0.04	**0.20**	**0.35**	0.00	**-0.40**	**-0.43**	-0.03
MSE ($\downarrow$)	All	**0.45**	**-0.44**	**-0.35**	0.04	**0.42**	-0.01	0.02	-0.17
	Affected	**0.83**	**-0.42**	-0.02	0.03	0.01	0.01	0.05	-0.02
	DEGs vs Control	**0.70**	**-0.46**	**-0.25**	0.01	-0.06	0.06	0.05	-0.12
	DEGs vs Rest	**0.67**	**-0.48**	**-0.26**	0.03	-0.08	0.01	0.04	-0.14

the $[0, 1]$ range, (iv) we square the weights to accentuate differences, and (v) we normalize the whole weight set to add up to 1. The key differentiator of this method is the use of all other perturbed cells as reference for DEG calculation (DEGs vs Rest) instead of the experiment's control (DEGs vs Control). This selection ensures the prioritized genes are the ones that make that perturbation unique from all the others without introducing control bias.

3.4. Technical Duplicate Baseline

scRNA-seq perturbation data poses multiple challenges for reproducible perturbation modeling, such as low capture, difficult annotation of perturbed cells, low perturbation efficiency, and high dimensionality which all contribute to high variance in average effect estimation for any given perturbation. To address this problem, we propose a technical duplicate baseline which tries to answer a simple question: "*how would a technical duplicate of the dataset perform in predicting a mean perturbation effect?*". Achieving this level of performance for a model would mean that its prediction errors are comparable to the variance of the experiment itself, defining a performance ceiling. We compute this baseline by randomly dividing the population of cells receiving a perturbation in half and using one half of the cells to predict the other half.

3.5. MSE vs WMSE Training

To assess whether WMSE can serve not only as an evaluation metric but also as a useful learning objective, we retrained GEARS from scratch with default hyperparameters under three loss functions: (i) its original loss, (ii) the standard (unweighted) MSE, and (iii) WMSE. For *Norman19* we train GEARS for the combination prediction task using all single perturbations and half of the combination perturbations for training and validation with the remaining half of combinations for testing. For *Replogle22* we perform the unseen gene prediction task where half the data is used for training and validation and the remaining perturbations are used for testing. In both cases, we train two GEARS models to get predictions for the all combinations or all unseen genes.

4. Experimental Results

4.1. *In silico* Screen

We ran an *in silico* screen as a discovery tool to pinpoint the factors that inflate mean-baseline performance (results summarized in Table 2). As expected, both perturbation effect size (ϵ) and gene perturbation probability (δ) generally decrease baseline performance. This is expected as stronger perturbations give more distinct signals and thus μ^{all} (mean of all perturbed cells) is a worse approximation of any individual μ^p (mean of specific perturbation). Similarly, the number of genes g reduced all error metrics due to library size normalization (more genes imply generally lower values post normalizing to 10^4 counts). And finally, the number of cells per perturbation n_p also increased performance, likely due to less sparsity in every pseudobulk μ^p estimation.

Apart from these effects, three parameters strongly modulated Pearson(Δ): the magnitude of control bias (β), the number of control cells (n_0), and the total number of perturbations (k). An in-depth analysis of control bias influence is provided in the following while additional results concerning n_0 and k are provided in Appendices C and D respectively.

4.2. Control Bias β

Systematic bias is readily apparent when evaluating DEGs against control. This is illustrated in the *Norman19* dataset in which a substantial proportion of DEGs are shared across multiple perturbations (Fig. 2A and Supplemental Figs. 5B,5C,5D,5E). Notably, the *Replogle22* dataset has fewer cells per perturbation (64 compared to 256 in *Norman19*), and thus less power to detect DEGs. Yet, we still observe

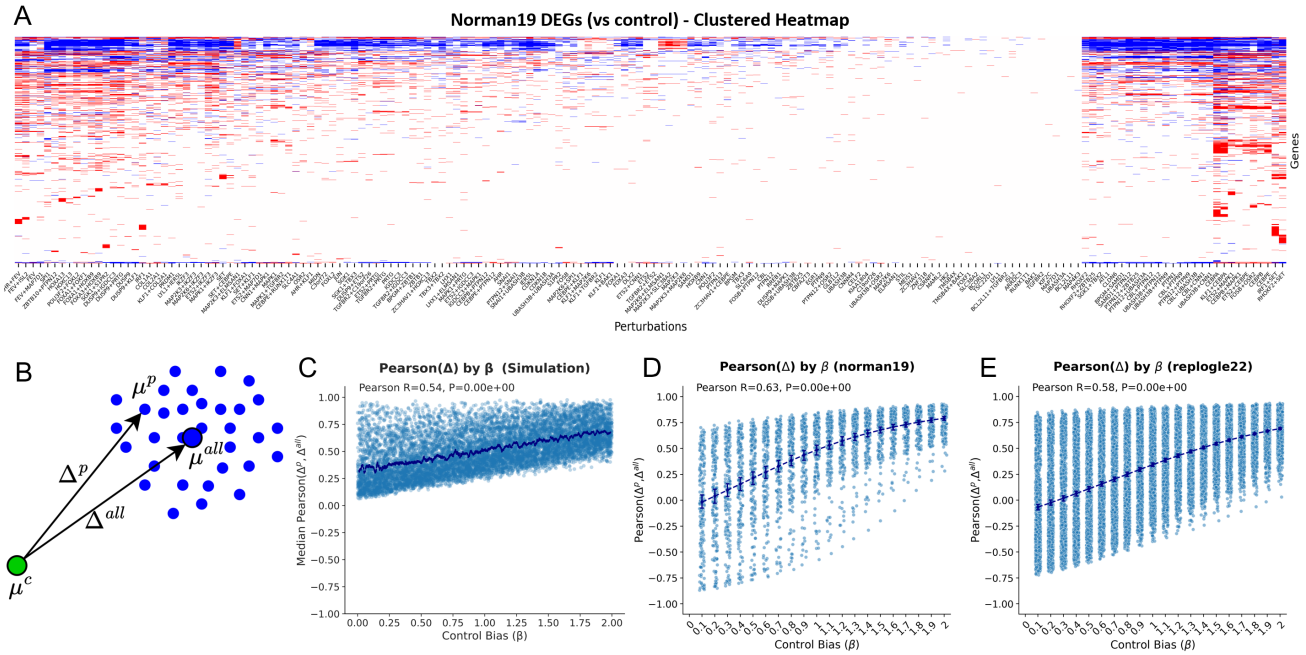

Figure 2. (A) Trinary (up, down, or unchanged) clustermap of significant DEGs for each perturbation compared to control. (B) Schematic showing high correlation between Δ^p and Δ^{all} due to the systematic bias of μ^c. (C) Pearson(Δ) performance of the mean baseline (μ^{all}) under increasing control bias (β) in simulations. Trend line shows moving average. (D) Pearson(Δ) performance of μ^{all} under increasing β in the *Norman19* dataset ($\beta = 0$ no control bias, $\beta = 1$ exact dataset control bias, $\beta = 2$ double the dataset control bias). Trend line shows mean and 95% CI of the mean. (E) Same as (D) in the *Replogle22* dataset.

that many DEGs are conserved across perturbations in this dataset (Supplemental Fig. 5A). As shown in Fig. 2B, we hypothesized that the high performance of the mean baseline on Pearson(Δ) could be explained if the effect of any perturbation (compared to the control mean) is similar to the effect of all perturbations compared to control. Under this hypothesis, the global expression difference ($\Delta^{all} = \mu^{all} - \mu^c$) becomes highly correlated with perturbation-specific differences ($\Delta^p = \mu^p - \mu^c$) as the distinction between perturbed and control states dominates the more subtle differences among perturbations. In our simulations, control bias (β) showed a high correlation with mean baseline (μ^{all}) performance on Pearson(Δ) (Fig. 2C, $r = 0.54$). This behavior was also observed in real data when introducing or removing control bias, with even stronger correlations of 0.63 and 0.58 in the *Norman19* (Fig. 2D) and *Replogle22* (Fig. 2E) datasets, respectively. These findings suggest that even a modest control bias leads to increased mean baseline performance on the Pearson(Δ) metric without any actual learning taking place. Moreover, the *Norman19* and *Replogle22* datasets are not considered poor-quality and, indeed, are used widely for model benchmarking (Wu et al., 2024; Li et al., 2024a; Csendes et al., 2025; Li et al., 2024b; Bendidi et al., 2024; Wenteler et al., 2024; Ahlmann-Eltze et al., 2024). Addressing control bias is likely not a matter of picking better datasets but rather picking a better reference. Instead of using non-targeting (NT) control cells, a more unbiased analysis might leverage all perturbed cells as the reference for DEG analysis and Δ-based performance metrics.

4.3. DEG Score-weighted MSE (WMSE)

While many perturbations strongly impact the transcriptome, leading to a large number of DEGs, most perturbations do not (Supplemental Fig. 5E). To address the problem of DEG signal dilution in perturbation modeling, we propose an MSE weighted by the strength of perturbation-specific DEGs (calculated with respect to all other perturbations; "vs Rest") (see 3.3.2). To evaluate the sensitivity of MSE and WMSE to niche DEG signals, we computed both metrics under two scenarios: for each perturbation we compared the perturbation mean (μ^p) with (1) the mean of all perturbed cells (μ^{all}), or (2) μ^{all*} in which we artificially set the gene expression of μ^{all} to be identical to μ^p for the top 25 perturbation-specific DEGs (approximately 0.3% of all genes). Despite the small modification, it alters the most important perturbation-specific signals. As expected, WMSE was significantly more sensitive to the niche DEG signals in μ^{all*} in both datasets compared with MSE (Fig. 3). Overall, these findings highlight the utility of DEG-aware metrics to capture perturbation-specific signals.

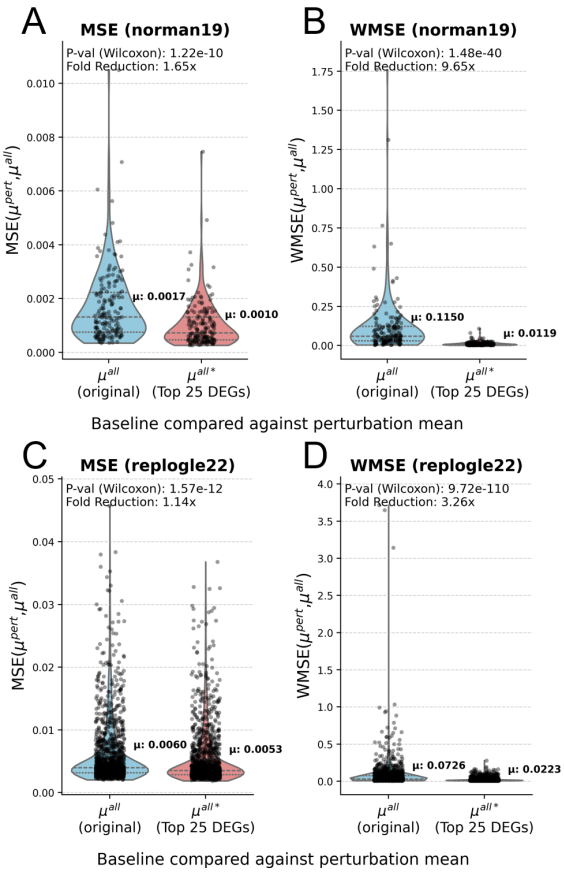

Figure 3. Weighted MSE is sensitive to differences in niche perturbation signals. (A) Violin plot showing MSE between true perturbation profiles μ^p and two possible predictions: the mean baseline μ^{all} and a modified version μ^{all*} which sets the top 25 DEGs to be perfectly predicted. (B) Same as (A) but MSE was weighted by the normalized DEG score (WMSE). (C-D) Same as (A-B) but for the *Replogle22* dataset.

4.4. Improved Metrics with Scale Calibration

As discussed above (see 2.3), common perturbation modeling metrics have multiple drawbacks: (1) MSE/MAE dilutes DEG signals (and filtering to top-k DEGs requires arbitrary cutoffs and cannot measure non-DEG errors), and (2) Pearson(Δ) is sensitive to control bias and cannot measure whether predicted effects have a similar dynamic range to true perturbation effects. The result is that current metrics often cannot detect mode collapse of perturbation models (i.e., they predict μ^{all} regardless of perturbation label). To address these limitations we introduce two metrics (see 3.3): WMSE and DEG score-weighted $R^2(\Delta)$ ($R^2_w(\Delta)$) in which μ^{all} is the Δ reference. Moreover, current benchmarking metrics often lack scale calibration, as negative and positive performance baselines are often not included. To address this limitation, we implement μ^c and μ^{all} as biased and uninformative negative baselines, respectively, and introduce a technical duplicate baseline that simulates ideal model performance (see 3.4).

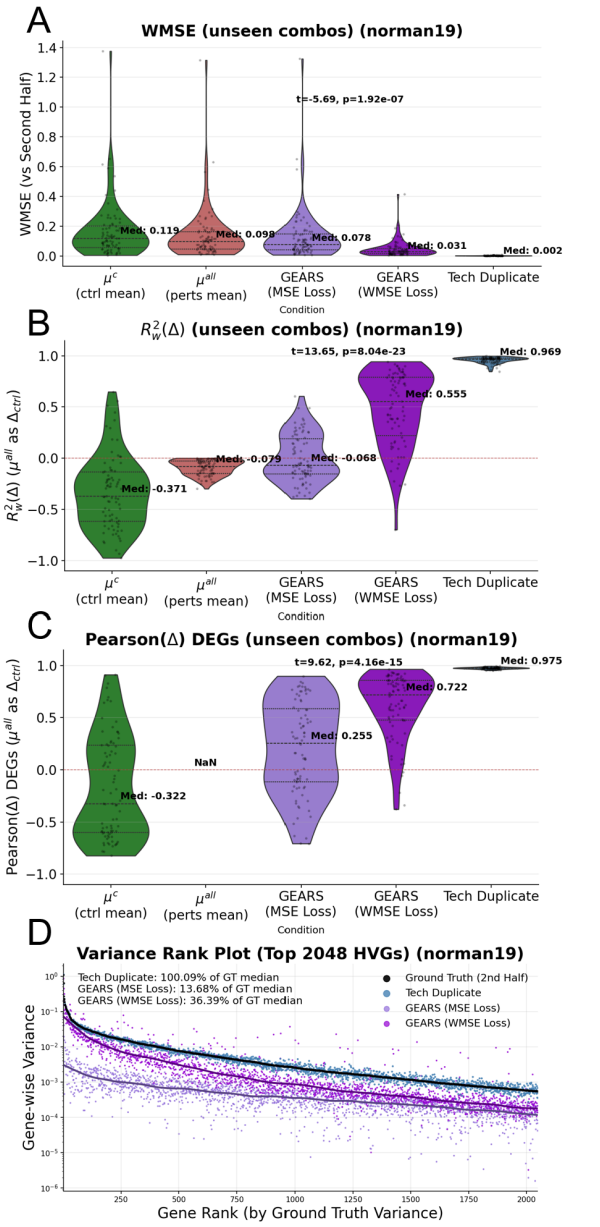

Figure 4. DEG score-weighted loss reduces mode collapse and improves model performance (*Norman19*). (A) WMSE between prediction and ground-truth perturbation mean. X labels: μ^c (control mean), μ^{all} (mean of all perturbed cells), predictions from GEARS model with MSE or WMSE loss, and technical duplicate baseline. Means between GEARS MSE/WMSE compared with paired t-test. (B) Same as (A) but for $R^2_w(\hat{\Delta}^p, \Delta^p)$, the DEG score-weighted R^2 between predicted ($\hat{\Delta}^p$) vs ground-truth perturbation effect (Δ^p). For Δ calculations, μ^{all} is the reference. (C) Same as (B) but with Pearson correlation and filtering to only include perturbation-specific DEGs (vs Rest). (D) Plot showing the top 2048 highly-variable genes ranked by variance in the ground truth pseudobulked dataset.

As shown in Supplemental Fig. 6, our baselines reveal that

DEG-based weighting greatly expands the dynamic range of each metric and reinstates a coherent ranking: the biased control prediction performs worst, the mean baseline has null performance, and the technical duplicate baseline establishes the upper bound. Without weighting, this order collapses; in *Replogle22*, for instance, the technical duplicate performs worst on both MSE and $R^2(\Delta)$ despite being the most informative predictor. The apparent anomaly highlights the signal-dilution problem. Technical duplicate estimates are derived from only half the cells in each perturbation (32 in *Replogle22*) and therefore carry more sampling noise than the population means returned by the baselines. Metrics that treat all genes uniformly reward this noise reduction, even though it is biologically irrelevant. Once genes are re-weighted by their perturbation-specific DEG statistics (WMSE, $R^2_w(\Delta)$), the technical duplicate's superior capture of genuine signal outweighs its higher variance, whereas models that regress toward the dataset mean lose ground. Indeed, for a weak perturbation such as MRPL23, unweighted Δ vectors from the two technical duplicate halves show virtually no correlation, yet restricting the comparison to the five true DEGs restores a strong correspondence, precisely the behavior our weighting scheme is designed to reward (Supplemental Fig. 7).

4.5. MSE vs WMSE Training

With our calibrated metrics, we evaluated the performance of a well-established perturbation model, GEARS (Supplemental Fig. 6). We found that GEARS displays decent performance in the combo-prediction task (*Norman19*), which requires extrapolating from single-gene effects to two-gene combos. However, GEARS struggles to outperform the μ^{all} baseline in the more difficult unseen gene prediction task (*Replogle22*), as it involves zero-shot predictions in a noisier dataset (Supplemental Fig. 6). We hypothesized that, especially for the unseen gene task, GEARS may be experiencing mode collapse during training due to its unweighted MSE loss. To test this, we retrained GEARS with either MSE or WMSE loss on both datasets and evaluated its performance (Fig. 4, Supplemental Fig. 8).

On *Norman19*, the null (μ^{all}) baseline yields strictly negative $R^2_w(\Delta)$ values that cluster near zero; thus any positive score indicates genuine learning. Switching the training objective from MSE to WMSE reduces the test-set WMSE by a factor of 2.5 and lifts the median $R^2_w(\Delta)$ from -0.068 to 0.555, bringing GEARS to within striking distance of the technical duplicate baseline. The same trend appears in Pearson(Δ) computed after filtering to DEGs (vs Rest) (Fig. 4C), a metric whose definition contains no weighting and thus rules out information leakage from the training weights. Gene-wise variance profiles (Fig. 4D) reveal the likely source of this performance gain: WMSE penalizes mode collapse, so predictions recover much of the true perturbation variance instead of shrinking toward the dataset mean.

In the more difficult unseen gene prediction task (*Replogle22*), WMSE again outperforms MSE on every weighted metric (though this improvement is smaller compared to the combo task) (Supplemental Fig. 8). When evaluation is restricted to the top 5% of perturbations ranked by DEG count (Supplemental Fig. 9), WMSE-trained GEARS overtakes the baseline, demonstrating that the model can capture perturbation-specific effects once the biological signal is strong enough. Moreover, we found that gene-wise variances align far more closely with ground truth when WMSE is used (Supplemental Fig. 8D), suggesting again that performance gains likely result from amelioration of mode collapse.

Taken together, these results indicate that DEG-based weighting steers optimization towards sparse, high-variance predictions that better reflect real perturbation effects, while simultaneously aligning with an intuitive calibration in $R^2_w(\Delta)$: null predictors cluster near zero, ordinary learning objectives improve modestly, and WMSE moves performance toward the empirical ceiling set by technical duplicate baseline. The unseen combo task (*Norman19*) is an easier task because information about all single genes is provided to models during training, and thus there is some data leakage when predicting unseen gene combo effects at test time. While this may inflate our estimation of WMSE-driven gains on that task, the parallel improvement in the *Replogle22* zero-shot setting confirms that weighting does provide a genuine generalization advantage. Embedding this inductive bias more explicitly in future architectures should help models remain competitive even when perturbations are weak or data quality is suboptimal.

5. Conclusion

Recent benchmarks reveal that predicting the perturbed dataset mean often performs much better than expected without any learning taking place and often surpasses fitted models' performance on common evaluation metrics. From our analyses on *in silico* and real-world datasets, we traced this behavior to bias in control cells and metric artifacts that reward mode collapse. Our conclusions produced a four-step remedy: (i) use the mean of all perturbed cells to remove systematic control bias in Δ and DEG calculations; (ii) adopt DEG-score weighted metrics (ΔR^2_w, WMSE) that penalize mode collapse while retaining transcriptome-wide coverage; (iii) calibrate all metrics with negative (μ^c), null (μ^{all}), and positive (technical duplicate) baselines; and (iv) implement DEG-aware optimization objectives (e.g., WMSE). Under this protocol, the mean baseline falls to null performance and models that capture perturbation-specific effects rise to the top.

Broader Impact

Accurate *in silico* perturbation response models can shorten drug-discovery cycles, cut laboratory costs, and reduce animal use by flagging promising candidates before any wet-lab work. Metrics that reward degenerate averages, however, risk elevating brittle models that provide uninformative predictions. By reducing reference bias, implementing calibrated, DEG-aware metrics, and introducing an optimization approach that penalizes mode collapse, our work enables scientists to avoid misleading metric artifacts and steer their resources toward building and evaluating better perturbation response models.

References

Ahlmann-Eltze, C., Huber, W., and Anders, S. Deep learning-based predictions of gene perturbation effects do not yet outperform simple linear methods. *BioRxiv*, pp. 2024–09, 2024.

Bendidi, I., Whitfield, S., Kenyon-Dean, K., Yedder, H. B., Mesbahi, Y. E., Noutahi, E., and Denton, A. K. Benchmarking transcriptomics foundation models for perturbation analysis: one pca still rules them all. *arXiv preprint arXiv:2410.13956*, 2024.

Csendes, G., Sanz, G., Szalay, K. Z., and Szalai, B. Benchmarking foundation cell models for post-perturbation rna-seq prediction. *BMC genomics*, 26(1):393, 2025.

Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., Duan, N., and Wang, B. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21(8):1470–1480, 2024.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerby-Arnon, L., Marjanovic, N. D., Dionne, D., Burks, T., Raychowdhury, R., et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7):1853–1866, 2016.

Gong, J., Hao, M., Cheng, X., Zeng, X., Liu, C., Ma, J., Zhang, X., Wang, T., and Song, L. xtrimogene: an efficient and scalable representation learner for single-cell rna-seq data. *Advances in Neural Information Processing Systems*, 36:69391–69403, 2023.

Hao, M., Gong, J., Zeng, X., Liu, C., Guo, Y., Cheng, X., Wang, T., Ma, J., Zhang, X., and Song, L. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024.

Istrate, A.-M., Li, D., and Karaletsos, T. scgenept: Is language all you need for modeling single-cell perturbations? *bioRxiv*, pp. 2024–10, 2024.

Kamimoto, K., Stringa, B., Hoffmann, C. M., Jindal, K., Solnica-Krezel, L., and Morris, S. A. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614(7949):742–751, 2023.

Li, C., Gao, H., She, Y., Bian, H., Chen, Q., Liu, K., Wei, L., and Zhang, X. Benchmarking ai models for in silico gene perturbation of cells. *bioRxiv*, pp. 2024–12, 2024a.

Li, L., You, Y., Liao, W., Fan, X., Lu, S., Cao, Y., Li, B., Ren, W., Fu, Y., Kong, J., et al. A systematic comparison of single-cell perturbation response prediction models. *bioRxiv*, pp. 2024–12, 2024b.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

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

Lotfollahi, M., Klimovskaia Susmelj, A., De Donno, C., Hetzel, L., Ji, Y., Ibarra, I. L., Srivatsan, S. R., Naghipourfar, M., Daza, R. M., Martin, B., et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular systems biology*, 19(6):e11517, 2023.

Nadig, A., Replogle, J. M., Pogson, A. N., Murthy, M., McCarroll, S. A., Weissman, J. S., Robinson, E. B., and O'Connor, L. J. Transcriptome-wide analysis of differential expression in perturbation atlases. *Nature Genetics*, pp. 1–10, 2025.

Norman, T. M., Horlbeck, M. A., Replogle, J. M., Ge, A. Y., Xu, A., Jost, M., Gilbert, L. A., and Weissman, J. S. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

Peidli, S., Green, T. D., Shen, C., Gross, T., Min, J., Garda, S., Yuan, B., Schumacher, L. J., Taylor-King, J. P., Marks, D. S., et al. scperturb: harmonized single-cell perturbation data. *Nature Methods*, 21(3):531–540, 2024.

Replogle, J. M., Saunders, R. A., Pogson, A. N., Hussmann, J. A., Lenail, A., Guna, A., Mascibroda, L., Wagner, E. J., Adelman, K., Lithwick-Yanai, G., et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

Roohani, Y., Huang, K., and Leskovec, J. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, 42(6):927–935, 2024.

Rosen, Y., Roohani, Y., Agarwal, A., Samotorčan, L., Consortium, T. S., Quake, S. R., and Leskovec, J. Universal

cell embeddings: A foundation model for cell biology. *bioRxiv*, pp. 2023–11, 2023.

Tang, Z., Zhou, M., Zhang, K., and Song, Q. scperb: Predict single-cell perturbation via style transfer-based variational autoencoder. *Journal of Advanced Research*, 2024.

Theodoris, C. V., Xiao, L., Chopra, A., Chaffin, M. D., Al Sayed, Z. R., Hill, M. C., Mantineo, H., Brydon, E. M., Zeng, Z., Liu, X. S., et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Wenteler, A., Occhetta, M., Branson, N., Huebner, M., Curean, V., Dee, W., Connell, W., Hawkins-Hooker, A., Chung, P., Ektefaie, Y., et al. Perteval-scfm: Benchmarking single-cell foundation models for perturbation effect prediction. *bioRxiv*, pp. 2024–10, 2024.

Wolf, F. A., Angerer, P., and Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

Wu, Y., Wershof, E., Schmon, S. M., Nassar, M., Osiński, B., Eksi, R., Zhang, K., and Graepel, T. Perturbench: Benchmarking machine learning models for cellular perturbation analysis. *arXiv preprint arXiv:2408.10609*, 2024.

A. Simulation on Real Data

For every parameter in the simulated data, we designed real data experiments as follows:

- g: In a $\log_2$ sequence from 2-8192, we randomly downsampled the data to N genes.
- n_0: In a $\log_2$ sequence from 1-8192, we randomly downsampled the control cell population to N cells.
- n_p: In a $\log_2$ sequence from 2-256 (for *Norman19*) and 2-64 (for *Replogle22*), we randomly selected N cells for each perturbation label.
- k: In a $\log_2$ sequence from 1-175 (for *Norman19*) and 1-1334 (for *Replogle22*), we randomly selected N perturbations. We repeated downsampling with 10 random seeds.
- β: We calculated the Δ between the mean of all perturbed cells (μ^{all}) and the mean of the control cells (μ^c). We then created synthetic control data by interpolating in equivalent steps of 0.1Δ between μ^{all} (0Δ) and μ^c (1Δ), terminating the interpolation at 2Δ.
- δ: Because there was no real-data equivalent of this simulated parameter, we mimicked it by downsampling the data to include perturbations with variable numbers of detected DEGs. We first ranked perturbations by the number of significant DEGs detected. We then downsampled the data by selecting perturbations in 20% quantile windows of normalized ranks ($0-0.2, 0.1-0.3, ...$) such that $0-0.2$ had the weakest perturbations and $0.8-1.0$ had the strongest.
- ϵ: To evaluate perturbation strength, we downsampled each dataset via the following procedure. For each perturbation, we ranked all genes by their absolute t-test metric score (obtained during DEG calculations). Ranks were binned into 10% quantiles. Data were downsampled to generate one dataset per decile such that each perturbation only contained the genes within the relevant DEG quantile. Thus, for quantile $0-0.1$, the data contained only the least differentially expressed genes within each perturbation, and for $0.9-1.0$ the data contained the most.
- μ_l: Within each perturbation, cells were ranked by library size, and the ranks were binned into deciles. Data were downsampled by selecting only cells belonging to each decile in sequence ($0-0.1, 0.1-0.2, ...$).

Table 3. Pearson correlations between metrics and simulation parameters. Comparison between real data experiments and simulation experiments for all the genes. Correlation values lower than -0.2 are highlighted in **blue** and Correlation values higher than 0.2 are highlighted in **red**.

Metric	Dataset	ϵ	g	n_p	β	δ	n_0	k	μ_l
Pearson Delta	Norman19	-	**0.36**	**0.39**	**0.63**	-0.19	**-0.64**	-0.09	-0.09
	Replogle22	-	0.10	**0.27**	**0.58**	-0.06	**-0.74**	-0.03	0.07
	Simulation	**-0.32**	0.00	**0.28**	**0.54**	**-0.28**	**-0.48**	-0.10	0.03
MSE	Norman19	-	**0.21**	**-0.74**	-	**0.70**	-	-	**-0.42**
	Replogle22	-	0.17	**-0.85**	-	**0.63**	-	-	**-0.58**
	Simulation	**0.45**	**-0.44**	**-0.35**	0.04	**0.42**	-0.01	0.02	-0.17

B. Upper Bounds of $R^2_w(\Delta)$ Under Constant $\hat{\mu}^p = \mu^{all}$ Predictions

Given the definition of the metric:

$$R^2_w(\Delta) = 1 - \frac{\sum_i w_i(\Delta_i - \hat{\Delta}_i)^2}{\sum_i w_i(\Delta_i - \bar{\Delta}_w)^2} \tag{9}$$

$$\bar{\Delta}_w = \sum_{i=1}^{g} w_i \Delta_i \tag{10}$$

$$\Delta_i = \mu_i^p - \mu_i^{all} \tag{11}$$

$$\hat{\Delta}_i = \hat{\mu}_i^p - \mu_i^{all} \tag{12}$$

Under a constant prediction $\hat{\mu}_i^p = \mu_i^{all}$ any predicted delta becomes $\hat{\Delta}_i = 0$ and the overall metric reduces to:

$$R_w^2(\Delta) = 1 - \frac{\sum_i w_i \Delta_i^2}{\sum_i w_i (\Delta_i - \bar{\Delta}_w)^2} \tag{13}$$

$$\tag{14}$$

Which can be easily shown to be negative by expanding $\sum_i w_i (\Delta_i - \bar{\Delta}_w)^2$ as follows:

$$\sum_i w_i (\Delta_i - \bar{\Delta}_w)^2 = \sum_i w_i \Delta_i^2 - 2\bar{\Delta}_w \sum_i w_i \Delta_i + \bar{\Delta}_w^2 \sum_i w_i \tag{15}$$

Because the set $\{w_i\}$ is normalized to add up to one and under the definition of $\bar{\Delta}_w$ the expression can be reduced to:

$$\sum_i w_i (\Delta_i - \bar{\Delta}_w)^2 = \sum_i w_i \Delta_i^2 - 2\bar{\Delta}_w^2 + \bar{\Delta}_w^2 \tag{16}$$

$$= \sum_i w_i \Delta_i^2 - \bar{\Delta}_w^2 \tag{17}$$

Then, rewriting the original metric value under the expansion we get:

$$R_w^2(\Delta) = 1 - \frac{\sum_i w_i \Delta_i^2}{\sum_i w_i \Delta_i^2 - \bar{\Delta}_w^2} \tag{18}$$

$$\tag{19}$$

From which the rightmost fraction is clearly bounded to be positive (the original fraction was between squared quantities) and greater or equal than 1 making $R_w^2(\Delta) \leq 0$ under the constant prediction case for any perturbation.

C. Control Bias and Number of Control Cells (n_0)

- Having observed this control bias, we questioned whether better sampling of the control population might be sufficient to reduce it by better approximating the center of the data (Supplemental Fig. 5F).

- We simulated increasingly higher numbers of control cells and found this reduced the predicting accuracy of the dataset mean (indicating a most robust control) (Supplemental Fig. 5G). However, the simulation also demonstrated that there are diminishing returns from continuing to sample the control population beyond 1000 cells. A similar effect was observed in both the *Norman19* and *Replogle22* dataset (Supplemental Fig. 5H and Supplemental Fig. 5I). These results highlight that, while greater sampling of the control cell population is sufficient to reduce bias, it cannot eliminate it. Thus, metrics which hinge upon the presence of an unbiased control cell population are fundamentally confounded by these effects. This poses a particular challenge for metrics based on deltas (such as the Pearson(Δ)) and DEGs, when deltas and DEGs are calculated with respect to the control cell population. This is a common practice in the field today, as evidenced by the widespread use of these metrics in recent papers (Gong et al., 2023; Istrate et al., 2024; Roohani et al., 2024; Cui et al., 2024; Li et al., 2024a; Wenteler et al., 2024; Tang et al., 2024; Csendes et al., 2025).

D. Number of perturbations (k)

- Another interesting finding of our simulation was a high Pearson(Δ) performance of the mean baseline for truly affected genes under a very low number of perturbations (around 0.6 in the lower k limit on Supplemental Fig. 5K).

- We hypothesize this is explained by the sparsity of the true biological differences when perturbations occur in non-overlapping genes. As exemplified on Supplemental Fig. 5J for a single perturbation, if this perturbation uniquely shows up-regulation of gene 1 and gene 2 and we are under the low k regime, then the mean baseline μ^{all} will pick up some signal from it and correlating $r(\Delta^p, \Delta^{all})$ will yield a positive results. This behavior is direct consequence of Pearson(Δ) focusing on direction changes rather that dynamic range. Note that because of the low probability of gene perturbation in simulation and sparsity of biological signal in the real data, this behavior is the rule rather than the exception.
- Confirming our result and explanation when sub-sampling perturbations in the real data under 10 different seeds we get the same trend when analyzing Pearson (Δ) only on DEGs vs the Rest of perturbations which are a proxy of the real affected genes and are the ones that make every perturbation different from every other (Supplemental Figs. 5L and 5M).
- As the number of perturbations in the dataset increases the probability of overlap between perturbed genes increases while also the pulling effect of a single perturbation on the mean baseline is significantly reduced. In other words μ^{all} is closer to the origin of the plot in Supplemental Fig. 5J reducing artificial performance inflation of the mean baseline as observed with simulations and real data.

E. Supplemental figures

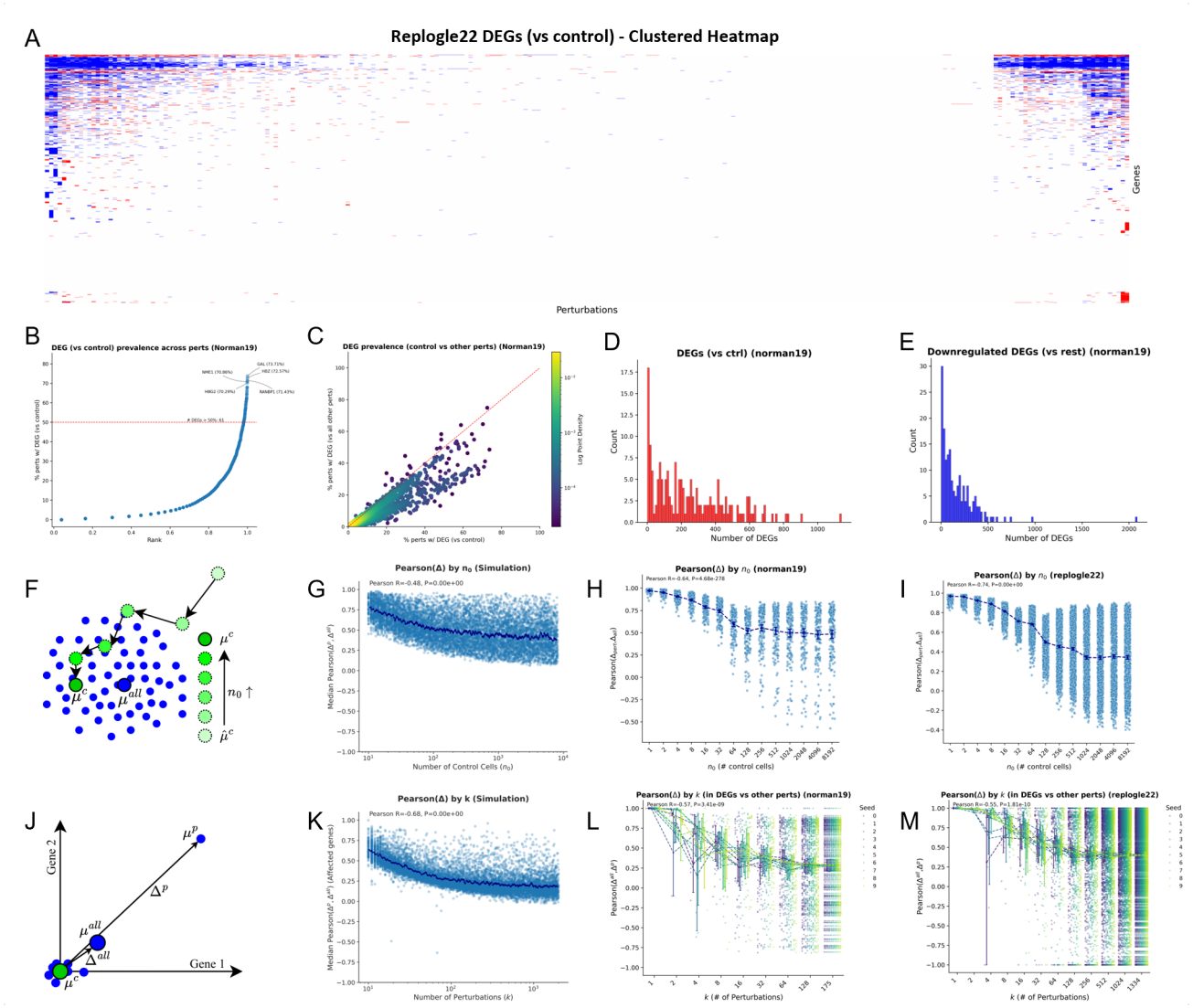

Figure 5. Supplemental figure accompanying Figure 2. (A) Trinary (up, down, or unchanged) clustermap of significant differentially expressed genes of every perturbation against the control population for the *Replogle22* dataset. Genes and perturbations were down-sampled randomly (to 256 and 2048 respectively) due to restrictions in plotting software. (B) Rank plot showing the percentage of perturbations detecting each DEG (vs Control) in the *Norman19* dataset. Annotation shows the top shared genes and the number of genes shared by most perturbations (61). (C) Scatter plot (with density histogram) showing the percentage of perturbations detecting each DEG (vs Control) compared with the percentage of perturbations detecting each DEG (vs all other perturbations). (D) Histogram showing distribution in terms of number of significant DEGs per perturbation in the *Norman19* dataset. (E) Same as (D) but for DEGs calculated with respect to all other perturbations (vs Rest). (F) Diagram showing the effect of increasing the number of control cells (n_0) on improving estimation of the control mean (μ^c) also reducing systematic bias. (G) Plot showing the effect of increasing control cell number (n_0) on Pearson(Δ^p,Δ^{all}), which is the similarity between $\mu^p - \mu^c$ and $\mu^{all} - \mu^c$, in simulated data. (H-I) Same as (G), but in real datasets *Norman19* and *Replogle22* respectively. (J) Plot illustrating the biasing of μ^{all} by a strong perturbation μ^p. The effect of this bias is to increase the similarity in direction between Δ^p and Δ^{all}, especially when the dataset contains fewer perturbations in the first place to moderate this single-perturbation influence. (K) Plot showing the effect of number of perturbations in a simulated dataset (k) on the Pearson(Δ^p,Δ^{all}). (L-M), same as (K) except in real datasets and with 10 random seeds for selection of k perturbations.

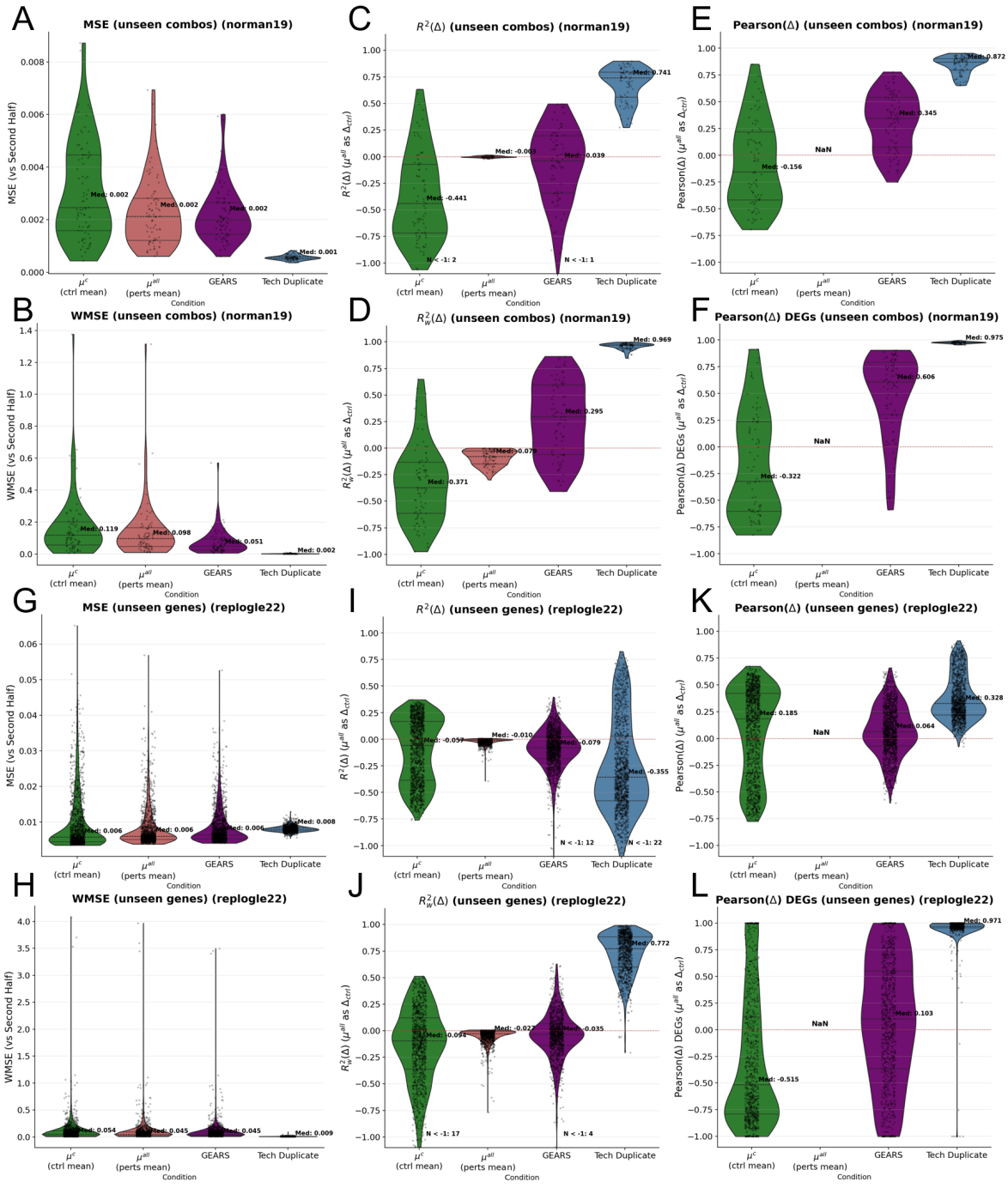

Figure 6. DEG-aware metrics, elimination of control bias, and addition of negative and positive baselines provide greater sensitivity and calibration to assess perturbation response model performance. (A) In the *Norman19* dataset, MSE between ground truth and baselines or model predictions. μ^c (control cell mean), μ^{all}, GEARS model predictions, and technical duplicate baseline are shown. (B) Same as (A) but measuring error with DEG score-weighted MSE (WMSE) instead of MSE. (C-D) Same as (A-B) but using $R^2(\Delta^p, \hat{\Delta}^p)$ and DEG score-weighted $R^2_w(\Delta^p, \hat{\Delta}^p)$ as the error metric, where $\hat{\Delta}^p$ is $\hat{\mu}^p - \mu^{all}$ and Δ^p is $\mu^p - \mu^{all}$. (E-F) Same as (C-D) except with Pearson instead of R^2 and filtering for DEGs (per perturbation) instead of weighting by DEG score. (G-L) Same as (A-F) but for the *Replogle22* dataset in which the task was prediction of unseen single genes.

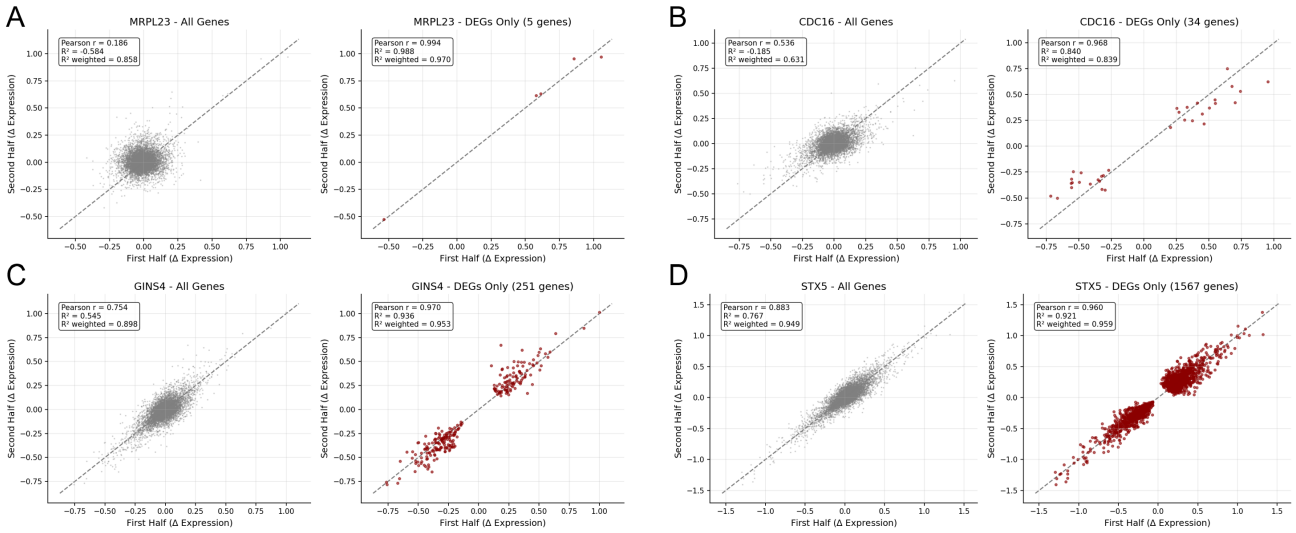

Figure 7. Effect of perturbation strength (measured by number of significant DEGs) on $R^2_w(\Delta^p, \hat{\Delta}^p)$ metric in technical duplicate baseline. (A) Scatter plots showing correlation between perturbation effects ($\Delta = \mu^p - \mu^{all}$) when using the first half of the data to predict the second half for MRPL23 with and without filtering for only MRPL23-specific DEGs. DEG-weighted and regular R^2 shown, along with Pearson correlation. (B) Same as (A) for CDC16. (C) Same as (A) for GINS4. (D) Same as (A) for STX5.

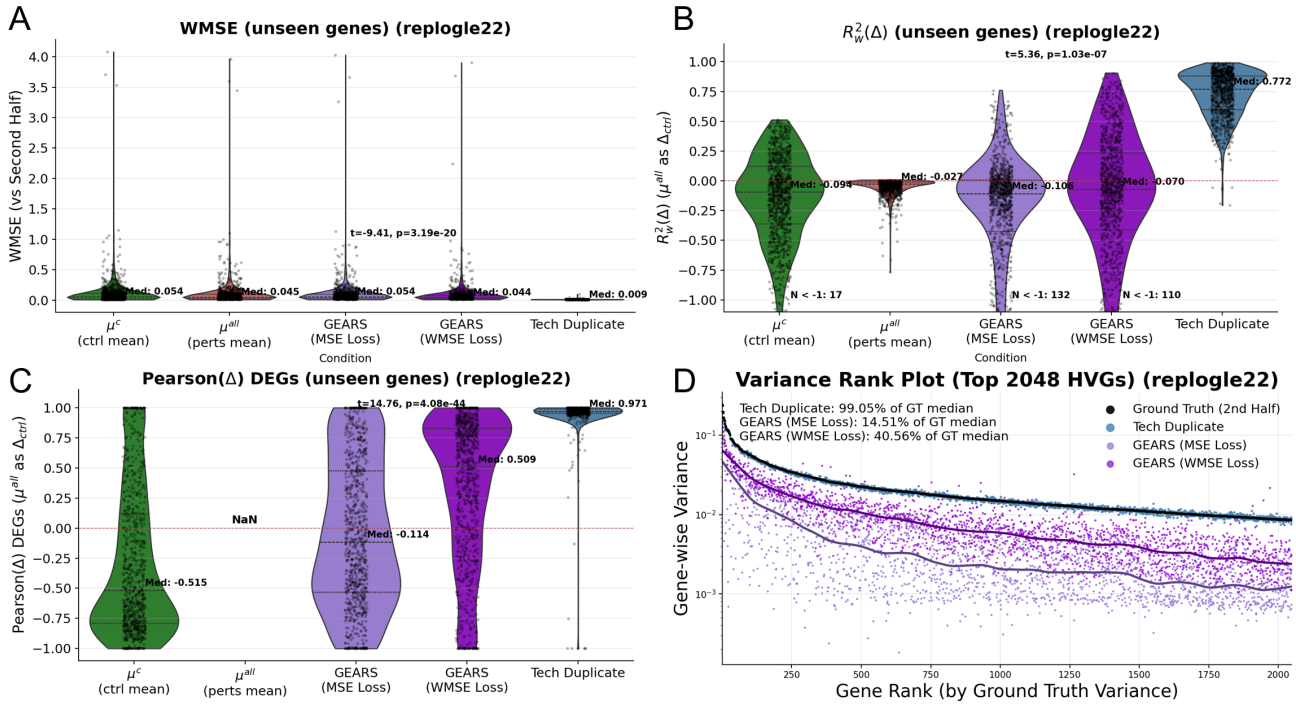

Figure 8. Supplemental figure accompanying Fig. 4. DEG score-weighted loss reduces mode collapse and improves model performance (*Replogle22*). (A) WMSE between prediction and ground-truth perturbation mean (μ^p). X labels: μ^c (control mean), μ^{all} (mean of all perturbed cells), predictions from GEARS model with MSE or WMSE loss, and technical duplicate baseline. Means between GEARS MSE/WMSE compared with paired t-test. (B) Same as (A) but for $R^2_w(\hat{\Delta}^p, \Delta^p)$, the DEG score-weighted R^2 between predicted ($\hat{\Delta}^p$) vs ground-truth perturbation effect (Δ^p). For Δ calculations, μ^{all} is the reference. (C) Same as (B) but with Pearson correlation and filtering to only include perturbation-specific DEGs (vs Rest). (D) Plot showing the top 2048 highly-variable genes ranked by variance in the ground truth pseudobulked dataset. Includes variances for the Technical Duplicate, GEARS MSE/WMSE predictions.

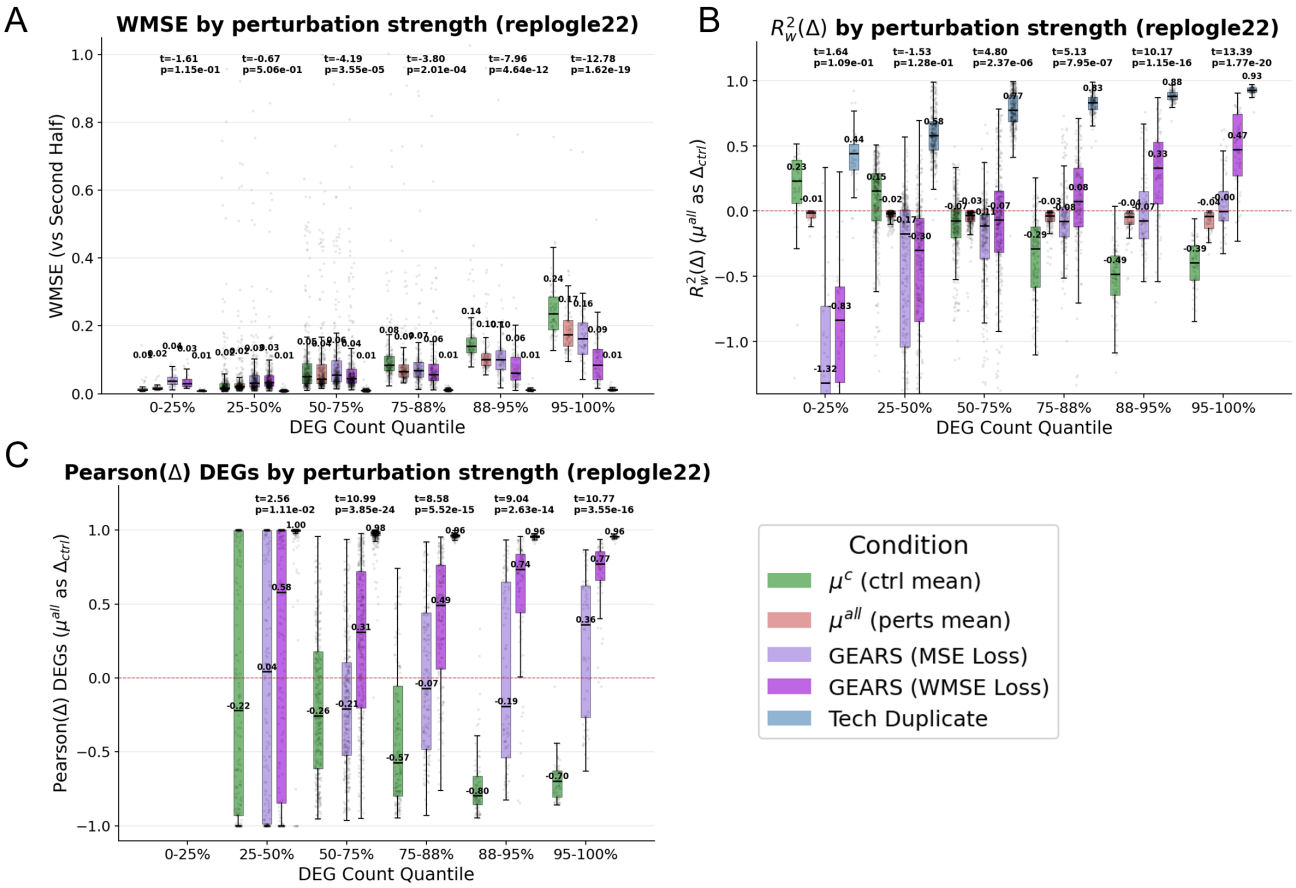

Figure 9. DEG score-weighted loss improves model performance on unseen gene prediction task, especially for stronger perturbations. (A) Performance of baselines and GEARS perturbation prediction (with MSE or WMSE loss), compared using WMSE metric vs ground truth perturbation mean (μ^p), grouped by the quantile range of perturbations tested (quantile ranges based on number of DEGs for each perturbation). Paired t-test conducted for each quantile range between GEARS with MSE vs GEARS with WMSE loss, with t-score and p value shown on plot. Median of each prediction within each quantile range also shown. (B) Same as (A) but for $R^2_w(\Delta^p, \hat{\Delta}^p)$ (DEG score-weighted R2 between predicted vs ground-truth perturbation effect). (C) Same as (B) but for Pearson correlation with data filtered to only include perturbation-specific DEGs (vs Rest). Note that the 0-25% quantile is missing because there were no DEGs for perturbations in this quantile.