Where Simple Baselines Fail: Mapping the Modeling Frontier of Gene Perturbation Prediction

Anna Kalygina [*1] **Alexander Theus** [*1 2] **Marina Esteban-Medina** [1 3 4] **Valentina Boeva** [1 3 5]

Abstract

Whether deep learning models for perturbation prediction outperform simple baselines remains contested. We show that this question is underspecified because aggregate benchmarks mix conditions already explained by simple structure with conditions where reproducible residual signal remains. We introduce *Baseline Saturation*, a per-condition measure of how much of a held-out response is explained by a matched simple baseline. Across nine CRISPR perturbation datasets, Baseline Saturation reveals substantial heterogeneity both between and within generalization regimes: many test conditions are near-solved by baselines, while others remain unsaturated. Stratifying by saturation shows that deep models add value primarily on resistant perturbations, where they recover perturbation-specific signal, but can hurt saturated conditions by over-applying residual structure. We further use inference-time saturation estimates to route between a strong baseline and learned residual modeling, improving performance on complementary unseen-perturbation benchmarks. Simple baselines are therefore not only competitors: their successes identify saturated regimes, and their failures map the modeling frontier of gene perturbation prediction.

1. Introduction

Single-cell perturbation screens measure how genetic or chemical interventions reshape cellular states, enabling systematic studies of gene function, regulatory programs, and drug response (Adamson et al., 2016; Norman et al., 2019; Frangieh et al., 2021; Replogle et al., 2022). These data have motivated models that predict transcriptional responses to unobserved perturbations, cellular contexts, doses, or combinations (Hetzel et al., 2022; Roohani et al., 2024; Cui et al., 2024; Littman et al., 2025; Adduri et al., 2025). Yet the field remains divided on a basic question: do deep learning models outperform simple baselines? Recent evaluations give conflicting answers, with some supporting expressive architectures and others showing that mean-based, additive, or linear predictors are difficult to beat (Miller et al., 2025; Ahlmann-Eltze et al., 2025; Kernfeld et al., 2023; Vollenweider & Bühlmann, 2026).

This tension mirrors a broader lesson from scientific machine learning: more expressive models do not automatically dominate simpler baselines. In time-series forecasting, anomaly detection, tabular learning, and specialized foundation-model benchmarks, strong linear, tree-based, or task-specific baselines have repeatedly challenged much larger pretrained or deep models (Zeng et al., 2023; Grinsztajn et al., 2022; Xu et al., 2025; Zhu et al., 2026). Similar concerns have emerged for single-cell foundation models, where independent evaluations show that pretrained representations do not consistently outperform simpler approaches in zero-shot settings (Kedzierska et al., 2025).

These findings should not be read as evidence that high-capacity models are unnecessary. Instead, they point to a decomposition that is often left implicit in perturbation-response evaluation. Simple baselines encode strong structural priors: shared perturbation programs, cell-type averages, dose scaling, or additivity between single perturbations. For held-out conditions where the baseline already explains most of the assayable signal, an expressive model has little useful structure left to recover and may add unwanted residual variance. When the baseline fails, however, the unexplained residual defines where deep modeling capacity can add value.

We therefore argue that a key modeling frontier in gene perturbation prediction lies in quantifying how much assayable signal is already explained by simple structure, identifying where residual headroom remains, and *routing deep modeling capacity toward the remaining residual*.

To operationalize this idea, we ask a condition-level ques-

[*]Equal contribution [1]ETH Zurich [2]Max Planck Institute for Intelligent Systems [3]Swiss Institute of Bioinformatics [4]ETH AI Center [5]Université Paris Cité. Correspondence to: Anna Kalygina <anna.kalygina@inf.ethz.ch>, Alexander Theus <alexander.theus@inf.ethz.ch>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

tion: before applying an expressive model, how much of the observed perturbation response is already explained by simple structure? Intuitively, if a regime-matched baseline moves the prediction nearly all the way from the control state toward the empirical perturbed state, then the condition is baseline-saturated and little assayable signal remains for a more expressive model to recover. If the baseline explains only a small part of this control-to-perturbation shift, then the remaining discrepancy defines residual headroom for deep modeling.

We formalize this intuition as *Baseline Saturation*: a per-condition measure of the fraction of the empirical control-bounded dynamic range captured by a regime-matched simple baseline. Baseline Saturation quantifies how much of a held-out response is already explained by simple structure, and therefore how much residual headroom remains for more expressive models. We evaluate this framework across nine single-cell CRISPR perturbation datasets, five generalization regimes, and seven calibrated metrics.

Our contributions are:

(i) We establish Baseline Saturation as a regime-aware diagnostic for separating baseline-saturated conditions from unsaturated ones, turning simple baselines into instruments for measuring residual headroom.

(ii) We use Baseline Saturation to quantify the hardness of perturbation-prediction splits, showing that benchmark difficulty is strongly compositional: the same split can contain both near-solved conditions and conditions with substantial residual headroom.

(iii) We show that deep models do not simply collapse to shared baseline-like structure. Instead, they often learn perturbation-specific residual structure, but over-apply it to saturated conditions where the baseline already explains the assayable signal.

(iv) We show, as a proof of concept, that Baseline Saturation can be estimated from inference-time features and used to build saturation-aware predictors that route between a strong baseline and learned residual modeling.

2. Background and Related Work

Gene perturbation prediction. Single-cell perturbation screens combine targeted genetic or chemical interventions with transcriptome-wide readouts, enabling systematic measurement of how perturbations reshape cellular state (Dixit et al., 2016; Adamson et al., 2016; Norman et al., 2019; Frangieh et al., 2021; Replogle et al., 2022). These assays have become a central tool for functional genomics because they link interventions to high-dimensional phenotypes, providing a scalable readout for studying gene function, regulatory programs, drug-response mechanisms, and genetic interactions. The corresponding prediction task is to infer the perturbation-induced expression change relative to control, for an unmeasured perturbation condition. Accurate predictors could help researchers prioritize experiments, explore large perturbation spaces computationally, and move toward virtual-cell models that can simulate intervention effects (Lotfollahi et al., 2023; Roohani et al., 2024; Adduri et al., 2025; Bunne et al., 2024).

Generalization regimes. Because perturbation predictors are intended to support in silico exploration beyond measured conditions, benchmarks typically evaluate whether models can generalize outside the training set. In practice, however, what should count as out-of-distribution generalization is not uniquely defined. Existing benchmarks hold out different axes of the perturbation-response tensor: unseen perturbations, unseen cellular contexts, unseen perturbation–context pairs, unseen doses, or unseen combinations (Lotfollahi et al., 2023; Roohani et al., 2024; Wu et al., 2026; Li et al., 2024; Adduri et al., 2025; Heidari et al., 2026). These regimes correspond to distinct scientific questions, such as whether perturbation effects transfer across genes, cell states, dosages, or combinations. Yet the held-out axis alone does not determine the difficulty of the prediction problem. Thus, evaluating generalization requires more than labeling a test condition as OOD: it requires measuring how much of the held-out response is already explained by simple structure available under the split. We formalize this idea with *Baseline Saturation*, a per-perturbation measure of the fraction of measurable response captured by a regime-matched simple baseline.

Deep models for perturbation response prediction. Motivated by these goals, recent work has proposed expressive models for perturbation response prediction. These include latent generative models that represent perturbations as shifts in cellular state (Lotfollahi et al., 2019; 2023), optimal-transport methods that learn maps between control and perturbed cell distributions (Bunne et al., 2023), graph-based architectures that use gene-gene interaction structure to predict multi-gene perturbation effects (Roohani et al., 2024), and large pretrained or foundation-style models for single-cell representation learning and perturbation prediction (Theodoris et al., 2023; Cui et al., 2024; Theus et al., 2024; Adduri et al., 2025). Recent gene-embedding-based approaches such as PRESAGE instead integrate heterogeneous prior knowledge sources through gene embeddings and use attention-based models to predict perturbation-induced expression changes (Littman et al., 2025). The premise of applying these models is that held-out regimes should require extrapolation beyond simple averages, linear trends, or additive structure: *if the response of an unseen condition were already determined by such structure, little would be gained from a high-capacity model*.

Do deep models beat simple baselines? Ahlmann-Eltze et al. (2025) showed that deep-learning-based perturbation predictors, including single-cell foundation models, often fail to outperform simple linear baselines. Subsequent work has refined, rather than settled, this conclusion. Viñas Torné et al. (2025) argue that standard metrics can overestimate performance by rewarding systematic variation shared across perturbations instead of perturbation-specific effects. Mejia et al. (2025) attribute part of the apparent baseline advantage to metric artifacts and mode collapse, showing that DEG-aware, calibrated metrics can better reward models that recover sparse biological signal. Miller et al. (2025) similarly argue that deep models can outperform uninformative baselines on well-calibrated metrics, while Vollenweider & Bühlmann (2026) emphasize that evaluation must expose biological signal, calibrate performance against empirical bounds, and compare against strong learned baselines. A complementary perspective comes from measurement-error theory. Schulz & Ritter (2026) argue that in biomedical prediction, limited measurement reliability can preferentially obscure nonlinear signal while leaving broad linear structure comparatively intact. This reframes baseline parity: when flexible and linear models tie, the result may reflect measurement-limited observability rather than the absence of nonlinear biology or a failure of model class. Together, these studies show that whether deep models outperform simple predictors depends strongly on noise, metric choice, calibration, and baseline strength. We argue that an additional axis is missing: the saturation composition of the test set. Aggregate scores mix perturbations already explained by simple structure with perturbations where reproducible residual signal remains. This obscures whether poor performance reflects mode collapse toward a shared response, or residual misrouting: the over-application of learned perturbation-specific structure to cases where a simple shared response would have sufficed.

3. Methodology

3.1. Problem setup and evaluation data

Perturbation-response notation. For each cellular context (c) and perturbation (p), we aggregate normalized single-cell expression profiles into a pseudobulk mean $\mu_{c,p} \in \mathbb{R}^G$. We define the perturbation response as the control-centered expression change

$$y_{c,p} = \mu_{c,p} - \mu_{c,\varnothing}, \tag{1}$$

where $\varnothing$ denotes the control condition within the same context. For single-context datasets, we omit the context index and write y_p. A model is trained on the conditions available under a given split and evaluated by comparing its predicted response $\hat{y}_{c,p}$ to the held-out response $y_{c,p}$.

Datasets and preprocessing. We evaluate nine publicly available single-cell CRISPR perturbation datasets: six single-context datasets, Adamson16 (Adamson et al., 2016), Frangieh21 (Frangieh et al., 2021), Norman19 (Norman et al., 2019), Sunshine23 (Sunshine et al., 2023), Wessels23 (Wessels et al., 2023), and Replogle20 (Replogle et al., 2020); and three multi-context datasets, Replogle22 (Replogle et al., 2022), McFaline23 (McFaline-Figueroa et al., 2019), and Jiang24 (Jiang et al., 2024). For Replogle22, we combine the K562 and RPE1 genome-scale Perturb-seq experiments into a single multi-context dataset. All datasets are processed into a common pseudobulk response representation by normalizing counts, log-transforming expression values, aligning genes to the evaluation panel, and centering perturbation profiles relative to matched controls. Dataset-specific details are summarized in Appendix.A.

Generalization regimes. Each dataset is evaluated under the regimes supported by its design: UNSEENPERT, holding out perturbations in observed contexts; UNSEENCELL, holding out cellular contexts with perturbations observed elsewhere; UNSEENPAIR, holding out specific context–perturbation pairs while both axes are observed separately; UNSEENBOTH, holding out both the perturbation and context; and UNSEENCOMBO, holding out perturbation combinations after observing the corresponding single perturbations. Splits are constructed over held-out units rather than individual cells, and cross-validation ensures that each unit appears in the test set at least once. Figure 1 summarizes these regimes and their matched simple baselines. We do not evaluate UNSEENDOSE, which requires perturbation-strength or dose measurements per condition.

3.2. Metrics, controls, and regime-matched baselines

Evaluation metrics. We evaluate predictions at the pseudobulk response level by comparing $\hat{y}_{c,p}$ to the held-out response $y_{c,p}$ for each test condition. Because perturbation prediction can fail in different ways, we use a panel of complementary metrics rather than a single score. These metrics assess response magnitude, correlation, concordance, directionality, perturbation discrimination, and gene-program recovery. Specifically, we use weighted MSE, weighted Pearson correlation, weighted concordance correlation coefficient, fraction of correctly predicted response directions, gene-set recovery scores for up- and down-regulated programs, and cosine rank. Each metric is computed per held-out condition before aggregation across perturbations, folds, datasets, and regimes. See Appendix B.6 for more details.

Positive and negative controls. Raw metric values are not directly comparable across perturbations because the amount of measurable signal differs across conditions. Fol-

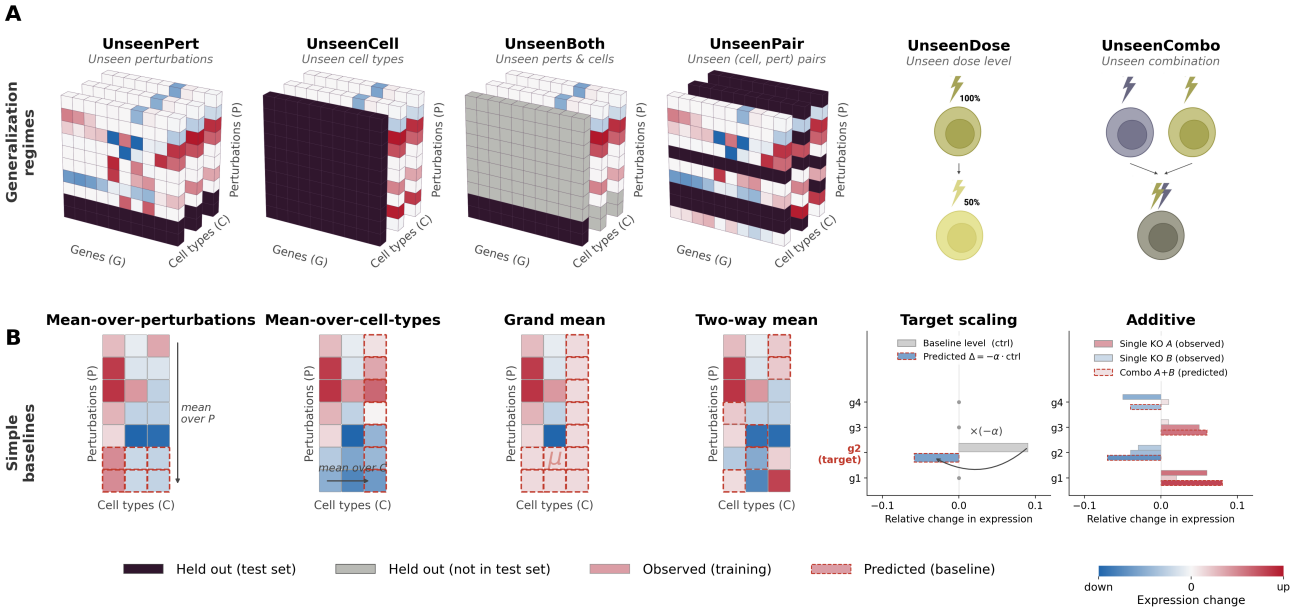

Figure 1. **Prediction regimes and non-learned baselines. A.** Schematic overview of the six prediction regimes used to evaluate perturbation-response prediction and their train/test composition. Each regime holds out a different axis of the perturbation-effect tensor: unseen perturbations (UNSEENPERT), unseen cellular contexts (UNSEENCELL), both unseen perturbations and contexts (UNSEENBOTH), unseen perturbation–context pairs (UNSEENPAIR), unseen dose levels (UNSEENDOSE), and unseen combinatorial perturbations (UNSEENCOMBO). Colored entries denote observed training responses, dark entries denote held-out test responses, and grey entries denote held-out responses that are not part of the test set. **B.** Scenario-matched non-learned baselines quantify simple recoverable structure under each regime. For each held-out setting, the baseline predicts the test response using only the structure available in the training tensor: mean-over-perturbations for UNSEENPERT, mean-over-cell-types for UNSEENCELL, the grand mean for UNSEENBOTH, a two-way mean for UNSEENPAIR, scaled deltas for UNSEENDOSE, and additive single-perturbation effects for UNSEENCOMBO. Blue and red indicate down- and up-regulated expression changes, respectively; dashed outlines indicate baseline-predicted held-out responses.

lowing Miller et al. (2025), we anchor each perturbation-wise metric between two empirical controls. The negative control predicts no perturbation effect, $\hat{y}_{c,p} = 0$, and represents a predictor that ignores the perturbation. The positive control estimates the reproducible upper reference using a split-half or interpolated duplicate of the measured response. Before calibration, metrics are oriented so that larger values indicate better performance; lower-is-better metrics are sign-flipped.

Signal-based perturbation filtering. A perturbation is only a meaningful evaluation target if its measured response $y_{c,p}$ is reliable. A common limitation of biological data is measurement and sampling noise, which conceals the underlying structure of the response. A perturbation's effect may be governed by linear or nonlinear structure, but once noise dominates, that structure is hidden and any predictor, whether simple or flexible, becomes uninformative (Schulz & Ritter, 2026). We therefore restrict evaluation to a single, fixed subset of perturbations whose effect is reliably measured, either detectable above noise or reproducible across replicates, and score every metric on this same subset. The first criterion is the signal-to-noise ratio

$$\mathrm{SNR}_{c,p} = \frac{\|y_{c,p}\|}{\mathrm{se}_{c,p}}, \tag{2}$$

the size of the perturbation response relative to its standard error $\mathrm{se}_{c,p} = \sqrt{V_p/n_p + V_\varnothing/n_\varnothing}$, where $V_p, V_\varnothing$ are the total per-cell variances of the perturbed and control populations and $n_p, n_\varnothing$ their cell counts. The second is a reproducibility coefficient $\rho_{c,p}$, the correlation between the response estimated on two independent technical-replicate halves of the cells, which retains weak but consistently directed effects that the SNR alone discards. Both thresholds are calibrated per dataset and context against a control-split null; predictors are then evaluated only on the retained perturbations (see Appendix H for the full derivation, null calibration, the combined rule, and per-dataset retention).

3.3. Regime-matched baselines

For each generalization regime, we empirically estimate the baseline that approximates the signal the best on the aggregated metrics. For UNSEENPERT it is Mean-over-Perturbations with the Target Scaling, for UNSEENCELL it is Mean-over-cell-types with the Target Scaling, for UNSEENBOTH it is Grand mean with the Target Scaling, for

UNSEENPAIR it is Two-way mean with the Target Scaling (for low signal dataset it is Mean-over-perturbations with the Target Scaling), for UNSEENCOMBO it is an Additive baseline. See Appendix for more implementation details.

3.4. Baseline Saturation

Aggregate scores mix held-out conditions that differ in how much modeling remains to be done. Some perturbation responses are already captured by simple structure available under the split, whereas others contain reproducible signal that the matched baseline fails to explain. We quantify this distinction with Baseline Saturation.

For a held-out condition (c, p), let $y_{c,p}$ be the observed perturbation response, 0 the negative-control prediction, $y^{+}_{c,p}$ the positive-control prediction, and $b_{c,p}$ the regime-matched baseline prediction. Raw metric values are reported in their native orientation. For the normalized quantities below, we use an oriented metric $\tilde{m}$, where larger values indicate better performance: $\tilde{m} = m$ for higher-is-better metrics and $\tilde{m} = -m$ for lower-is-better metrics. We define

$$s_{c,p,m} = \mathrm{clip}_{[0,1]}\left(\frac{\tilde{m}(b_{c,p}, y_{c,p}) - \tilde{m}(0, y_{c,p})}{\tilde{m}(y^{+}_{c,p}, y_{c,p}) - \tilde{m}(0, y_{c,p}) + \epsilon}\right). \quad (3)$$

The clipping confines saturation values to the interval $[0, 1]$. Baseline Saturation therefore measures the fraction of the empirical control-bounded dynamic range captured by the baseline. Values near one indicate saturated conditions, where the baseline approaches the reproducible upper reference and little measurable headroom remains. Values near zero indicate baseline-resistant conditions, where the response is measurable but not explained by the simple structure encoded by the baseline.

3.5. Deep learning models for perturbation response prediction

We evaluate three deep perturbation-response models that represent distinct modeling paradigms: GEARS, scGPT, and PRESAGE. GEARS is an established graph-based model for perturbation prediction that uses gene-gene interaction structure to extrapolate to unseen perturbations (Roohani et al., 2024). scGPT represents the foundation-model paradigm, using large-scale pretraining on single-cell data to learn transferable gene and cell representations (Cui et al., 2024). PRESAGE represents a gene-embedding-based approach, combining prior biological knowledge with attention-based prediction of perturbation responses (Littman et al., 2025). Recent evaluations have reported strong performance for PRESAGE across perturbation-response benchmarks (Miller et al., 2025).

3.6. Baseline Saturation-aware prediction

Baseline Saturation suggests a simple inference principle: high-saturation conditions should stay close to the baseline, whereas low-saturation conditions should retain deep-model residuals. We test this by predicting $\hat{s}_{c,p}$ from inference-time features and using it in two ways: post-hoc blending and end-to-end residual gating.

Post-hoc Saturation Blending. This approach leaves the deep model unchanged and only modifies inference. Let $f_{c,p}$ denote the deep model prediction and $b_{c,p}$ the baseline prediction. We train a saturation predictor to estimate $\hat{s}_{c,p} \in [0, 1]$ from validation conditions and use it to interpolate between the two predictions:

$$\hat{y}_{c,p} = (1 - \hat{s}_{c,p}) f_{c,p} + \hat{s}_{c,p} b_{c,p}. \quad (4)$$

The prediction therefore approaches the deep model when the condition is predicted to be baseline-resistant, and approaches the baseline when it is predicted to be saturated. Since the deep model is not retrained, this isolates the effect of saturation-aware inference from changes in model training.

End-to-end Saturation Gating. This approach incorporates saturation awareness into the model training itself. Instead of predicting the response directly, the model predicts a residual $\hat{r}_{c,p}$ around the baseline and scales it by the predicted residual headroom $1 - \hat{s}_{c,p}$:

$$\hat{y}_{c,p} = b_{c,p} + (1 - \hat{s}_{c,p}) \hat{r}_{c,p}. \quad (5)$$

We train the model with the task loss and a saturation-supervised gate loss:

$$\mathcal{L} = \mathcal{L}_{\mathrm{MSE}} + \lambda (\hat{s}_{c,p} - s_{c,p})^2. \quad (6)$$

This encourages residuals to remain active on baseline-resistant conditions and shrink on saturated ones. Unlike the post-hoc blend, the model learns saturation-dependent residuals during training and produces the gated prediction in a single forward pass.

4. Results

4.1. Baseline Saturation reveals heterogeneous benchmark difficulty

Baseline Saturation reveals that perturbation-prediction benchmarks are not uniformly difficult. Each dataset and regime contains a different mixture of noise-limited responses, structure that simple baselines capture, and baseline-resistant signal. At the regime level, UNSEEN-COMBO is the easiest (pooled median Baseline Saturation 0.93, with 63% of perturbations saturated at ≥ 0.66): the

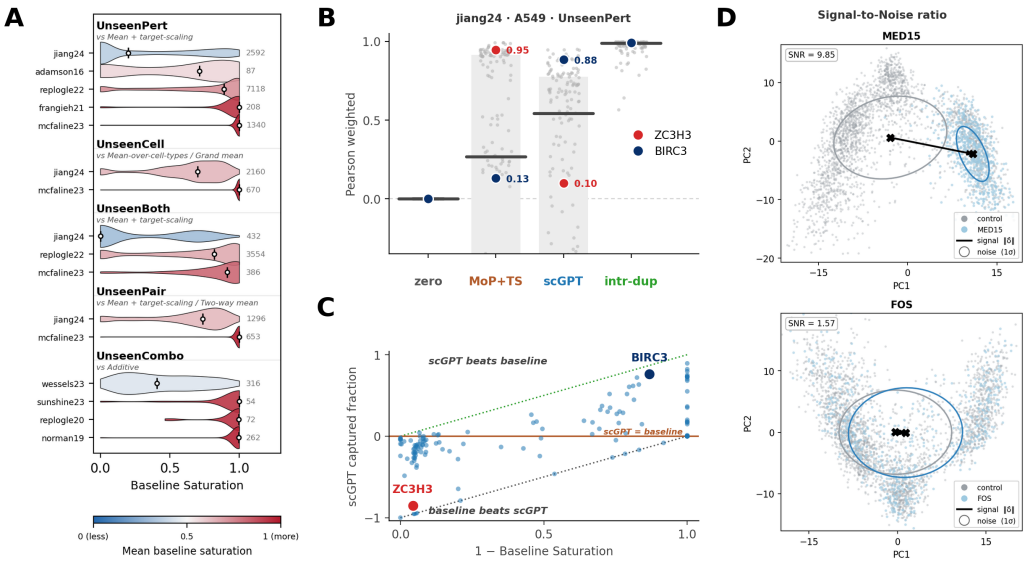

Figure 2. **Baseline Saturation reveals heterogeneous, dataset- and regime-dependent benchmark difficulty. A.** Per-perturbation Baseline Saturation across datasets, grouped by prediction regime (reliability-filtered perturbations, DEG-weighted Pearson). Each horizontal violin is one dataset, coloured by its mean Baseline Saturation (blue, baseline-resistant; red, saturated); the italic sub-label gives the matched simple baseline for that regime. **B.** Per-perturbation prediction accuracy for one example dataset and regime (Jiang24, A549, UNSEENPERT). Each strip is the per-perturbation distribution for four predictors that bracket the achievable range: the negative control (*zero* change), the best simple baseline (mean-over-perturbations with target scaling, *MoP+TS*), a deep model (*scGPT*), and the positive control (intra-experiment replicate, *intr-dup*). Two perturbations are highlighted: ZC3H3 (captured by the baseline) and BIRC3 (baseline-resistant, recovered by scGPT). **C.** For the same setting, the fraction of the remaining baseline-to-replicate signal that scGPT captures (cf) versus the headroom the baseline leaves (1 − Baseline Saturation), one dot per perturbation. Points above cf = 0 are perturbations where the deep model improves on the simple baseline; the dotted lines mark the negative- and positive-control limits. **D.** Signal-to-noise geometry underlying the reliability filter: principal-component embeddings of control and perturbed pseudobulk replicates for a high-SNR perturbation (MED15) and a low-SNR perturbation (FOS); the arrow is the mean perturbation effect (signal) and the ellipse the replicate noise.

additive baseline already captures most combinatorial effects, which quantifies prior observations that deep models often fail to beat additive baselines in this setting (Vollenweider & Bühlmann, 2026). UNSEENBOTH is the hardest (median 0.56, 34% saturated), where neither perturbation nor cell-type averages transfer to the held-out combination. UNSEENCELL, UNSEENPAIR, and UNSEENPERT fall in between (medians 0.77, 0.75, and 0.72). These averages hide the fact that every regime is a mixture (Figure 2A): even UNSEENCOMBO contains 22% baseline-resistant perturbations, and UNSEENBOTH splits almost evenly into easy, moderate, and hard conditions (34, 39, and 28%).

Baseline Saturation varies even more between datasets within the same regime (Figure 2A). In UNSEENPERT, the median ranges from 0.12 in Jiang24 to near 1.0 in McFaline23 and Frangieh21, a wider spread than between regimes. This likely reflects how noisy each study is: when measurements are noisy, the reliability filter retains mostly strong, mean-like responses and the dataset appears saturated, whereas a cleaner dataset such as Jiang24 retains many perturbation-specific responses that simple baselines cannot reproduce.

Baseline Saturation is therefore a joint property of the perturbation biology (how mean-like its effect is) and the dataset (its noise level and the diversity of effects across perturbations). A dataset's average Baseline Saturation summarizes how much non-linear structure it demands. Because mean-based baselines depend on the training split, the same dataset shows different Baseline Saturation when split along different axes; this is one reason model rankings are inconsistent across datasets and regimes.

4.2. Deep models help where baselines leave headroom

The heterogeneity in Baseline Saturation has a direct consequence for model evaluation. A test set mixes easy perturbations, where baselines are near-optimal, with hard ones, where baselines are uninformative, and aggregate metrics weight all of them equally. Because easy perturbations make up most of the test set in most regimes, the aggregate reflects the setting where baselines already solve the problem and a deep model can only add noise. A model that learns real perturbation-specific signal on the hard perturbations can therefore look worse overall, even when it is more useful where modeling is actually needed.

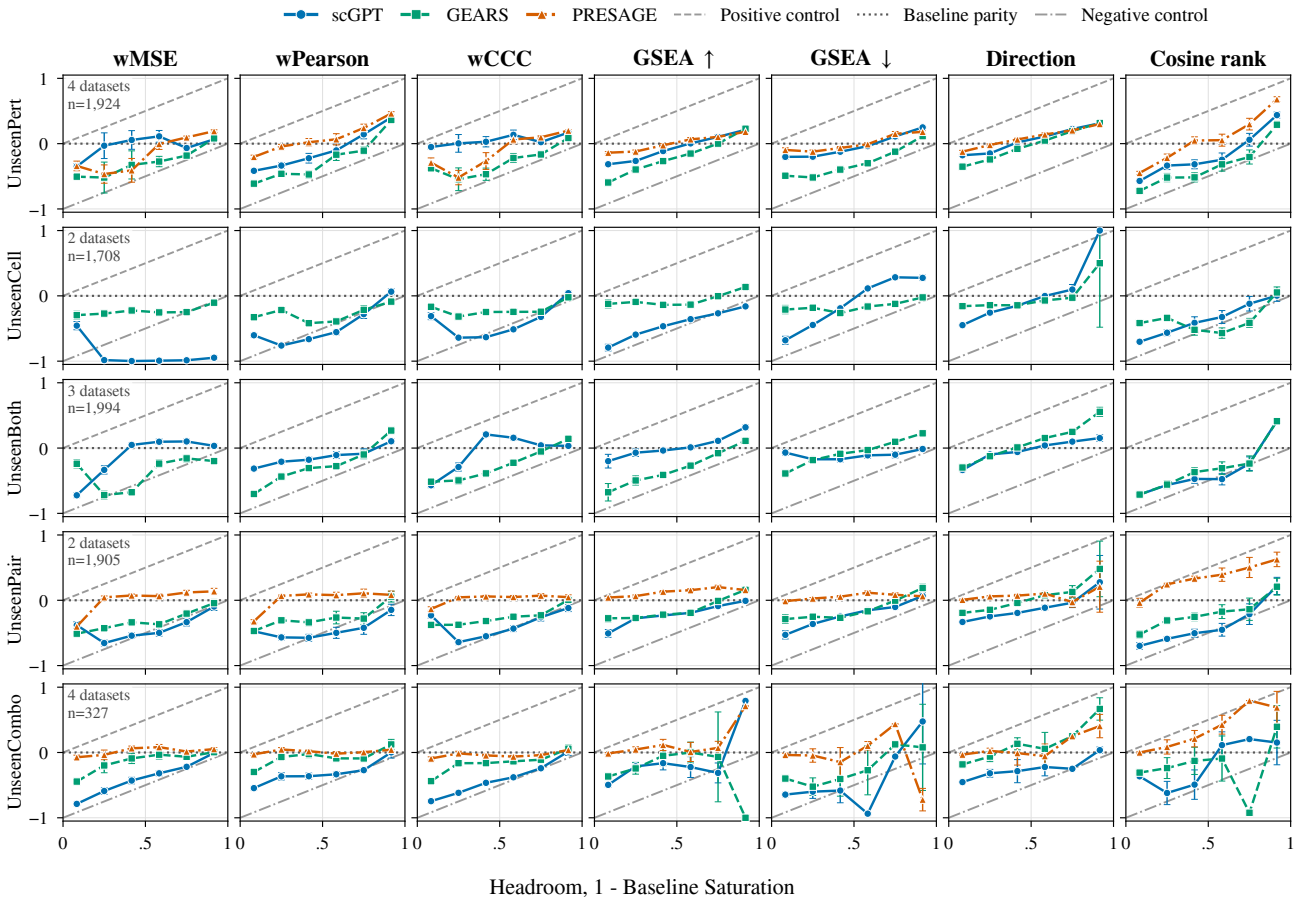

Figure 3. Deep-model capture fraction as a function of empirical baseline headroom across generalization regimes and evaluation metrics. The x-axis shows headroom, $1 - \text{Baseline Saturation}$. The horizontal dotted line denotes baseline parity; values above this line indicate that the deep model improves over the selected empirical baseline. The dashed and dash-dotted diagonals denote the positive and negative controls, respectively, which define the empirical range. Points show binned means and error bars show one standard-error interval.

Stratifying by Baseline Saturation separates these two regimes. The capture fraction of all three deep models rises steadily with perturbation difficulty (Figure 3): in UNSEENPERT, scGPT, GEARS, and PRESAGE beat the hardest baseline on 59 to 70% of hard perturbations across the seven metrics, but on only 18 to 28% of easy ones, and the same gradient holds across regimes. The tiered raw scores tell the same story (Table 1): the deep models lead on the hard tier and fall back to or below the baseline on the easy tier.

This shows that the deep models do learn perturbation-specific structure, since they beat the baseline exactly where it leaves headroom: scGPT recovers baseline-resistant perturbations such as BIRC3 that the baseline misses, while the baseline already captures perturbations such as ZC3H3 (Figure 2B,C). Their weak aggregate performance comes from over-applying that structure to saturated perturbations, where the baseline already captures the response and the added flexibility only injects variance. Baseline Saturation makes this visible by separating the perturbations a model should defer to the baseline from those where its capacity is worth spending.

4.3. Saturation-aware predictors improve performance

As outlined in Section 3.6, we introduce two mechanisms for directing deep learning capacity toward perturbations that are not already explained by a strong empirical baseline: *Post-hoc Saturation Blending* and *End-to-end Saturation Gating*. Post-hoc Saturation Blending combines a trained PRESAGE predictor with the Mean + target scaling baseline using an estimated saturation score, whereas End-to-end Saturation Gating learns this routing jointly during model training. As a proof of concept, we evaluate both methods on the `jiang24` and `replogle22` UnseenPert benchmarks. These datasets represent complementary regimes: `replogle22` is substantially baseline-saturated, while `jiang24` retains substantial headroom for deep models.

Table 1. **Absolute predictor performance stratified by Baseline Saturation across five generalization scenarios.** Median of the raw per-perturbation metric, pooled across datasets within each scenario, over the NSR$\vee\rho$ reproducible-signal inclusion set. *Baseline* is the empirically-strongest simple baseline per dataset$\times$scenario (UNSEENPERT: Mean+TargetScaling; UNSEENCELL: Mean-over-cell-types, Mean-over-perturbations-and-cell-types; UNSEENBOTH: Mean+TargetScaling; UNSEENPAIR: Mean+TargetScaling, Two-way-mean; UNSEENCOMBO: Additive). **Bold** marks the best predictor per metric (excluding *Zero* and *Tech-dup*; only when it beats the *Zero* negative control). Dir. acc. (fraction of correct direction) is evaluated over each perturbation's significant DEGs; perturbations with no reproducible DEGs contribute 0, so it is low for weak-signal datasets.

	UNSEENPERT			UNSEENCELL			UNSEENBOTH			UNSEENPAIR			UNSEENCOMBO		
	$MSE_{wt}\downarrow$	$Pears_{wt}\uparrow$	Dir. acc. $\uparrow$	$MSE_{wt}\downarrow$	$Pears_{wt}\uparrow$	Dir. acc. $\uparrow$	$MSE_{wt}\downarrow$	$Pears_{wt}\uparrow$	Dir. acc. $\uparrow$	$MSE_{wt}\downarrow$	$Pears_{wt}\uparrow$	Dir. acc. $\uparrow$	$MSE_{wt}\downarrow$	$Pears_{wt}\uparrow$	Dir. acc. $\uparrow$
Hard (Baseline Saturation <0.33; $n=1354$; 164; 552; 244; 69)															
Zero	0.041	0.00	0.00	0.039	0.00	0.00	0.041	0.00	0.00	0.034	0.00	0.00	0.320	0.00	0.00
Baseline	0.045	-0.03	0.35	**0.037**	0.09	**0.65**	0.043	-0.00	0.33	0.036	0.09	0.49	0.288	**0.18**	**0.80**
Ridge	0.045	-0.11	0.34	**0.037**	0.10	0.64	**0.030**	0.43	0.75	0.036	-0.06	0.43	0.307	0.07	0.67
LinearAdditive	0.045	-0.11	0.34	**0.037**	**0.11**	**0.65**	0.033	**0.50**	**0.78**	0.038	0.02	0.48	0.307	0.07	0.67
PRESAGE	**0.034**	**0.19**	**0.52**	—	—	—	—	—	—	**0.030**	**0.19**	**0.55**	0.289	0.17	0.75
scGPT	0.040	0.04	0.41	0.178	-0.09	0.47	0.047	-0.03	0.34	0.038	-0.27	0.33	0.313	-0.00	0.17
GEARS	0.042	-0.03	0.44	0.040	-0.11	0.48	0.054	0.01	0.49	0.040	-0.08	0.44	**0.286**	0.11	0.75
Tech-dup	0.005	0.95	1.00	0.001	0.98	1.00	0.009	0.89	1.00	0.001	0.98	1.00	0.010	0.99	1.00
Moderate ($0.33\leq$ Baseline Saturation <0.66; $n=979$; 451; 768; 410; 47)															
Zero	0.057	0.00	0.00	0.072	0.00	0.00	0.062	0.00	0.00	0.049	0.00	0.00	0.186	0.00	0.00
Baseline	0.047	0.58	0.77	**0.045**	**0.61**	**0.75**	0.049	0.59	0.00	0.033	0.60	0.75	0.144	**0.51**	0.86
Ridge	0.049	0.59	0.75	0.048	0.60	0.74	**0.034**	**0.71**	0.00	0.031	0.60	0.75	0.174	0.35	0.62
LinearAdditive	0.049	0.59	0.75	**0.045**	**0.61**	**0.75**	**0.034**	0.67	0.00	0.033	0.60	0.75	0.174	0.35	0.62
PRESAGE	**0.038**	**0.66**	**0.81**	—	—	—	—	—	—	**0.028**	**0.69**	**0.84**	**0.143**	**0.51**	**0.89**
scGPT	0.046	0.44	0.69	0.268	0.01	0.47	0.045	0.51	0.00	0.051	0.03	0.52	0.183	0.16	0.50
GEARS	0.055	0.20	0.54	0.058	0.33	0.57	0.068	0.14	0.00	0.045	0.33	0.58	0.152	0.44	0.75
Tech-dup	0.007	0.94	1.00	0.001	0.99	1.00	0.012	0.83	0.00	0.001	0.99	1.00	0.008	0.98	1.00
Easy (Baseline Saturation ≥ 0.66; $n=2796$; 1093; 674; 1251; 211)															
Zero	0.018	0.00	0.00	0.005	0.00	0.00	0.025	0.00	0.00	0.006	0.00	0.00	0.155	0.00	0.00
Baseline	**0.015**	**0.46**	0.00	0.005	**0.23**	0.00	**0.017**	**0.51**	0.67	0.006	**0.55**	0.00	**0.019**	**0.96**	**0.98**
Ridge	0.020	0.27	0.00	0.005	0.13	0.00	0.020	0.48	0.92	0.006	0.44	0.00	0.116	0.58	0.68
LinearAdditive	0.020	0.27	0.00	0.005	0.10	0.00	0.022	0.50	**0.96**	0.006	0.41	0.00	0.116	0.58	0.68
PRESAGE	0.019	0.23	0.00	—	—	—	—	—	—	0.006	0.42	0.00	**0.019**	0.94	0.97
scGPT	**0.015**	0.38	0.00	0.020	0.02	0.00	0.026	0.07	0.00	0.006	0.13	0.00	0.128	0.46	0.57
GEARS	0.022	0.09	0.00	0.005	0.12	0.00	0.041	0.05	0.00	0.006	0.16	0.00	0.071	0.75	0.83
Tech-dup	0.009	0.53	0.00	0.003	0.20	0.00	0.010	0.68	1.00	0.003	0.73	0.00	0.004	0.99	1.00

Figure 4 summarizes aggregate performance across SNR-kept perturbations. On `jiang24`, mean + target scaling is weak, and all deep predictors substantially improve over it on correlation-based metrics. End-to-end Saturation Gating gives the strongest overall performance, improving wPearson from 0.329 for PRESAGE to 0.409, while also improving wCCC from 0.168 to 0.185 and reducing cosine rank from 0.244 to 0.177. This indicates that the gating objective does not merely protect against saturated regimes, but can also preserve or improve performance when substantial headroom remains.

The `replogle22` dataset presents the opposite challenge: the Mean + target scaling baseline is already strong, making it harder for a deep model to add value. In this regime, vanilla PRESAGE underperforms the baseline on wPearson (0.418 vs. 0.457), while Post-hoc Saturation Blending partially recovers performance (0.515). End-to-end Saturation Gating performs best, reaching 0.581 wPearson and 0.405 wCCC, compared with 0.291 wCCC for PRESAGE and Post-hoc Saturation Blending.

Together, these results suggest that saturation-aware predictors have potential to recover both operating regimes: they can retain the gains of deep models when baseline headroom is large, and they avoid degradation when the empirical baseline is already highly saturated. In particular, End-to-end Saturation Gating provides the most consistent improvements across datasets and metrics, supporting the use of baseline-aware training objectives rather than purely post-hoc correction.

5. Discussion

A saturation-conditioned view of model comparison. Our results show that apparently conflicting conclusions about deep learning and simple baselines can hold simultaneously. In baseline-saturated regimes, simple predictors capture most of the signal, leaving little measurable headroom for expressive models. In more unsaturated regimes, reproducible perturbation-specific structure remains beyond these predictors, and expressive models can add value. Baseline Saturation therefore acts as a conditioning variable for model comparison: the same model can appear competitive or uncompetitive depending on the saturation composition of the test set. A key question therefore becomes where simple baselines fail to explain the assayable signal, and what characterizes this boundary.

Distribution shift is not the same as modeling demand. Common split labels describe what is held out from training, but not how difficult the held-out response is to predict. A distributionally novel setting can be easy if the response follows simple structure available in the training data. Conversely, a standard unseen-perturbation split can be difficult when held-out responses deviate from the shared perturbation program. For example, combinatorial prediction can be

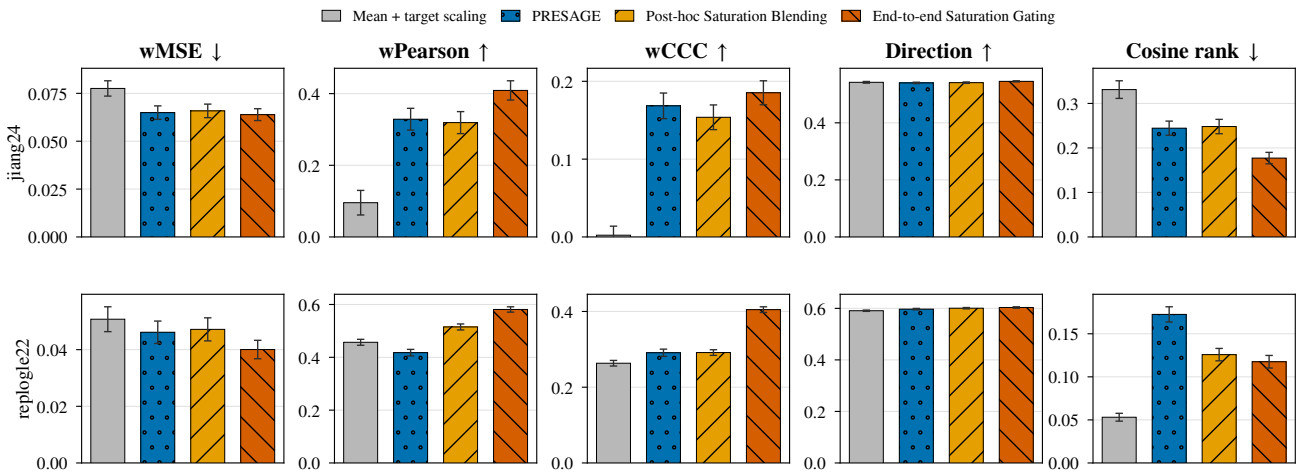

Figure 4. Aggregate performance of the baseline and saturation-aware PRESAGE variants on the `jiang24` and `replogle22` UnseenPert benchmarks. Bars show mean performance over SNR-kept perturbations. Error bars show approximate 95% confidence intervals for the mean. The baseline is Mean + target scaling. Post-hoc Saturation Blending applies saturation-aware blending after training, while End-to-end Saturation Gating learns saturation-aware gating during training. Arrows indicate whether lower or higher values are better for each metric.

nominally out-of-distribution while still being largely saturated by an additive baseline, whereas unseen-perturbation prediction can retain substantial headroom when the mean response is uninformative. Baseline Saturation complements split labels by measuring how much residual signal remains after the regime-matched baseline has been applied.

Residual over-application as a model failure mode. Stratifying by Baseline Saturation also changes how we interpret deep-model failures. Poor aggregate performance does not necessarily imply that deep models collapse to shared baseline-like structure. In our experiments, deep models often improve exactly where baselines leave headroom, indicating that they recover perturbation-specific residual structure. Their failures arise when this residual structure is applied to saturated conditions, where the baseline already captures the assayable response. This suggests that a key challenge is not only learning residual structure, but also estimating when that residual should be used.

Limitations and future work. Baseline Saturation depends on the quality of the positive control. When few cells are available per perturbation, the split-half replicate is noisy, which can inflate saturation estimates. We mitigate this through DEG-weighted and masked metrics that concentrate signal above the noise floor and verify that DRF embeds the noise floor on pseudobulk data (Appendix B.6). GSEA metrics require a minimum number of differentially expressed genes per perturbation; for datasets with weak perturbation effects (notably McFaline23 and Frangieh21), many perturbations did not meet this threshold and were evaluated without GSEA, reducing the metric coverage for those datasets. Our analysis evaluates three DL architectures (scGPT, GEARS, PRESAGE); other models may behave differently. The biological characterization identifies what predicts resistance (deviation from the shared response) but not why specific genes produce unique responses; incorporating gene regulatory network structure is an important next step. Finally, we use non-learned baselines as the primary reference; the frontier narrows further with learned baselines (Ridge, Linear Additive), suggesting that much of the remaining headroom is capturable by linear models.

6. Conclusion

Perturbation prediction benchmarks mix conditions whose assayable signal is already captured by simple structure with conditions where reproducible residual signal remains. Baseline Saturation makes this mixture explicit: it quantifies the solved portion of a benchmark, identifies where residual headroom remains, and provides a signal for routing deep modeling capacity toward that residual. Rather than treating simple baselines only as competitors, our results show that they can serve as instruments for mapping and exploiting the modeling frontier of gene perturbation prediction.

Acknowledgments and Disclosure of Funding

We thank Florian Barkmann, Michael Vollenweider, and Vera Milovanović for the helpful discussions and comments. Alexander Theus is supported by the Max Planck ETH Center for Learning Systems.

Impact Statement

This work advances evaluation and modeling methodology for single-cell perturbation prediction. Its main societal impact is indirect: better calibrated benchmarks may help researchers identify when computational predictions are reliable, where additional experiments are needed, and how to prioritize perturbations for follow-up studies. The proposed framework does not introduce new biological interventions, clinical recommendations, or deployable decision systems. As with any perturbation-prediction method, misuse could arise if model outputs are treated as experimentally validated effects. We therefore emphasize that Baseline Saturation is intended as a diagnostic and evaluation tool, and that predictions should be interpreted together with empirical controls, uncertainty, and downstream experimental validation.

References

Adamson, B., Norman, T. M., Jost, M., Cho, M. Y., Nuñez, J. K., Chen, Y., Villalta, J. E., Gilbert, L. A., Horlbeck, M. A., Hein, M. Y., et al. A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882, 2016.

Adduri, A. K., Gautam, D., Bevilacqua, B., Imran, A., Shah, R., Naghipourfar, M., Teyssier, N., Ilango, R., Nagaraj, S., Dong, M., et al. Predicting cellular responses to perturbation across diverse contexts with state. *BioRxiv*, pp. 2025–06, 2025.

Ahlmann-Eltze, C., Huber, W., and Anders, S. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22(8):1657–1661, 2025.

Bunne, C., Stark, S. G., Gut, G., Del Castillo, J. S., Levesque, M., Lehmann, K.-V., Pelkmans, L., Krause, A., and Rätsch, G. Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768, 2023.

Bunne, C., Roohani, Y., Rosen, Y., Gupta, A., Zhang, X., Roed, M., Alexandrov, T., AlQuraishi, M., Brennan, P., Burkhardt, D. B., et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024.

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.

Frangieh, C. J., Melms, J. C., Thakore, P. I., Geiger-Schuller, K. R., Ho, P., Luoma, A. M., Cleary, B., Jerby-Arnon, L., Malu, S., Cuoco, M. S., et al. Multimodal pooled perturb-cite-seq screens in patient models define mechanisms of cancer immune evasion. *Nature genetics*, 53(3):332–341, 2021.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on tabular data? arxiv. *arXiv preprint arXiv:2207.08815*, 2022.

Heidari, M., Karimpour, M., Srivatsa, S., and Montazeri, H. Evaluating single-cell perturbation response models is far from straightforward. *bioRxiv*, pp. 2026–02, 2026.

Hetzel, L., Boehm, S., Kilbertus, N., Günnemann, S., Theis, F., et al. Predicting cellular responses to novel drug perturbations at a single-cell resolution. *Advances in Neural Information Processing Systems*, 35:26711–26722, 2022.

Jiang, Q., Chen, S., Chen, X., and Jiang, R. scpram accurately predicts single-cell gene expression perturbation response based on attention mechanism. *Bioinformatics*, 40(5):btae265, 2024.

Kedzierska, K. Z., Crawford, L., Amini, A. P., and Lu, A. X. Zero-shot evaluation reveals limitations of single-cell foundation models. *Genome Biology*, 26(1):101, 2025.

Kernfeld, E., Yang, Y., Weinstock, J. S., Battle, A., and Cahan, P. A systematic comparison of computational methods for expression forecasting. *BioRxiv*, pp. 2023–07, 2023.

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, 2024.

Littman, R., Levine, J., Maleki, S., Lee, Y., Ermakov, V., Qiu, L., Wu, A., Huang, K., Lopez, R., Scalia, G., et al. Gene-embedding-based prediction and functional evaluation of perturbation expression responses with presage. *bioRxiv*, pp. 2025–06, 2025.

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): MSB202211517, 2023.

McFaline-Figueroa, J. L., Hill, A. J., Qiu, X., Jackson, D., Shendure, J., and Trapnell, C. A pooled single-cell genetic screen identifies regulatory checkpoints in the continuum of the epithelial-to-mesenchymal transition. *Nature genetics*, 51(9):1389–1398, 2019.

Mejia, G. M., Miller, H. E., Leblanc, F. J., Wang, B., Swain, B., and Camillo, L. P. d. L. Diversity by design: Addressing mode collapse improves scrna-seq perturbation modeling on well-calibrated metrics. *arXiv preprint arXiv:2506.22641*, 2025.

Miller, H. E., Mejia, G. M., Leblanc, F. J., Wang, B., Swain, B., and Camillo, L. P. d. L. Deep learning-based genetic perturbation models do outperform uninformative baselines on well-calibrated metrics. *bioRxiv*, pp. 2025–10, 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.

Replogle, J. M., Norman, T. M., Xu, A., Hussmann, J. A., Chen, J., Cogan, J. Z., Meer, E. J., Terry, J. M., Riordan, D. P., Srinivas, N., et al. Combinatorial single-cell crispr screens by direct guide rna capture and targeted sequencing. *Nature biotechnology*, 38(8):954–961, 2020.

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.

Schulz, M.-A. and Ritter, K. Measurement noise limits the advantage of nonlinear models over linear models in biomedical prediction, 2026. URL https://arxiv.org/abs/2606.18420.

Sunshine, S., Puschnik, A. S., Replogle, J. M., Laurie, M. T., Liu, J., Zha, B. S., Nuñez, J. K., Byrum, J. R., McMorrow, A. H., Frieman, M. B., et al. Systematic functional interrogation of sars-cov-2 host factors using perturb-seq. *Nature communications*, 14(1):6245, 2023.

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.

Theus, A., Barkmann, F., Wissel, D., and Boeva, V. Cancerfoundation: A single-cell rna sequencing foundation model to decipher drug resistance in cancer. *bioRxiv*, pp. 2024–11, 2024.

Viñas Torné, R., Wiatrak, M., Piran, Z., Fan, S., Jiang, L., Teichmann, S. A., Nitzan, M., and Brbić, M. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, pp. 1–10, 2025.

Vollenweider, M. and Bühlmann, P. Signal, bounds, and baselines: Principles for evaluating virtual cell perturbation models. *bioRxiv*, pp. 2026–04, 2026.

Wessels, H.-H., Méndez-Mancilla, A., Hao, Y., Papalexi, E., Mauck III, W. M., Lu, L., Morris, J. A., Mimitou, E. P., Smibert, P., Sanjana, N. E., et al. Efficient combinatorial targeting of rna transcripts in single cells with cas13 rna perturb-seq. *Nature methods*, 20(1):86–94, 2023.

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

Wu, Y., Wershof, E., Schmon, S., Nassar, M., Osiński, B., Eksi, R., Yan, Z., Stark, R., Zhang, K., and Graepel, T. Perturbench: Benchmarking machine learning models for cellular perturbation analysis. *Advances in Neural Information Processing Systems*, 38, 2026.

Xu, Z., Gupta, R., Cheng, W., Shen, A., Shen, J., Talwalkar, A., and Khodak, M. Specialized foundation models struggle to beat supervised baselines. In *International Conference on Learning Representations*, volume 2025, pp. 102908–102933, 2025.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zhu, X., Carpentier, L., and Verbeke, M. When foundation models are one-liners: Limitations and future directions for time series anomaly detection. In *The Fourteenth International Conference on Learning Representations*, 2026.

A. Datasets

Table 2. Perturbation datasets used in this work. The first six datasets have a single cell type; the last three (replogle22, mcfaline23, jiang24) span multiple cell types and contribute one row per prediction regime. **Cells** counts all cells in the evaluated panel, controls included. **Perturbations** are the perturbation units present after preprocessing (S: single-gene, C: combinatorial). **Evaluated** is the number of held-out test units scored by the benchmark. For the single-cell-type datasets this is the held-out test perturbations in the overlapping-perturbation set scored identically across every method (Sec. A.4): for UNSEENPERT each single-gene perturbation is held out exactly once across the five folds, whereas for UNSEENCOMBO only the combinations are held out (single-gene perturbations and controls always stay in training). For the multi-cell-type datasets the test unit is a (cell type, perturbation) pair, and **Evaluated** counts all such held-out pairs aggregated over folds (the full held-out set, not restricted to cross-method overlap). **Folds**: number of cross-validation folds; single-gene datasets use a $70/10/20$ train/validation/test split of perturbations per fold, combination datasets use 2-fold CV over the held-out combinations. Counts use the Miller et al. (2025)-aligned gene panel where available and the union panel otherwise. Splits and perturbation names for the single-cell-type datasets are matched to Miller et al. (2025).

Dataset	Cell type	Pert. method	Cells	Perturbations	Evaluated	Regime	Folds	Source
adamson16 (Adamson et al., 2016)	K562	CRISPRi	55 021	89 S	75	UNSEENPERT	5 (70/10/20)	Zenodo 10044268
frangieh21 (Frangieh et al., 2021)	Melanoma	CRISPR KO (Cas9)	47 571	240 S	209	UNSEENPERT	5 (70/10/20)	Zenodo 7041849
norman19 (Norman et al., 2019)	K562	CRISPRa	88 528	105 S + 131 C	124 C	UNSEENCOMBO	2	Zenodo 7041849
sunshine23 (Sunshine et al., 2023)	Calu-3	CRISPRi	31 441	179 S + 33 C	31 C	UNSEENCOMBO	2	Zenodo 10044268
wessels23 (Wessels et al., 2023)	Monocyte (THP-1)	Cas13d knockdown	22 794	28 S + 158 C	142 C	UNSEENCOMBO	2	Zenodo 13350497
replogle20 (Replogle et al., 2020)	K562	CRISPRi	26 938	42 S + 40 C	35 C	UNSEENCOMBO	2	GEO GSE146194
replogle22 (Replogle et al., 2022)	K562, RPE1	CRISPRi	380 833	1 991 S	3 982	UNSEENPERT	5	Zenodo 10044268
					2 956	UNSEENBOTH	5	
mcfaline23 (McFaline-Figueroa et al., 2019)	A172, T98G, U87MG	CRISPRi	153 011	511 S	1 533	UNSEENPERT	5	HF perturbench
					1 533	UNSEENCELL	3	
					819	UNSEENBOTH	5	
					1 533	UNSEENPAIR	5	
jiang24 (Jiang et al., 2024)	A549, BxPC3, HAP1, HT29, K562, MCF7	CRISPRi	872 030	216 S	1 296	UNSEENPERT	5	HF perturbench
					1 080	UNSEENCELL	5	
					216	UNSEENBOTH	5	
					1 296	UNSEENPAIR	5	

A.1. Data Preprocessing

All datasets use the same pipeline with a fixed random seed (`42`). Each dataset is downloaded and converted to a common format. Duplicate gene symbols are dropped. The perturbation, cell type, and donor columns are renamed to shared names. Perturbation labels are then cleaned. Guide suffixes are stripped. Non-targeting guides become a single control label. Empty and sentinel labels are dropped. We remove cells with fewer than 200 detected genes. We remove genes seen in fewer than 3 cells, except perturbation target genes. Raw counts are saved. Expression is normalized to 10^4 counts per cell and $\log(1 + x)$-transformed. In the datasets with multiple cell types (for example mcfaline23) we only consider perturbations that are present in every cell type: a perturbation is scored only when it has at least four cells in both technical-duplicate halves of every cell type, so that each evaluated perturbation has a matched control and ground-truth profile in all cell types. For replogle22 the two cell types come from two separately released datasets, K562 and RPE1, which we concatenate on their shared genes and then restrict to the perturbations measured in both.

Cells are then downsampled per perturbation. Perturbations with at least 12 cells are kept. Each perturbation is capped at the rounded mean number of cells per perturbation. Controls are capped at 8192 cells. For datasets with several cell types this is done inside each cell type. The gene panel is the union of the top 8192 highly variable genes (scanpy default flavor) and all measured target genes. Differential expression is computed for each half with an overestimated-variance t-test against the rest, which matches scanpy `t-test_overestim_var`. Genes with a Benjamini-Hochberg adjusted $p < 0.05$ are called significant. The first half also gives per-perturbation DEG masks, directions, and weights.

A.2. Evaluation Setup

Within each perturbation, and within each cell type when there are several, cells are split in half at random. The first half gives the ground truth signal that predictions are scored against. The second half gives the technical duplicate noise ceiling. Folds are then assigned for each scenario (UNSEENPERT, UNSEENCELL, UNSEENBOTH, UNSEENPAIR, UNSEENDOSE, UNSEENCOMBO).

We evaluate with cross-validation, so every perturbation is held out and scored once. For single-gene perturbation datasets we use 5-fold cross-validation, where in each fold about 70% of the perturbations are train, 10% are validation, and 20% are test. The held-out perturbations are unseen during training. We match the perturbation names and the folds to Miller et al. (2025) so that their results and ours can be compared directly, and because the random seed is set right before the

perturbations are shuffled we reproduce these single-gene folds exactly. For UNSEENCOMBO regime datasets we use 2-fold cross-validation. Here all single-gene perturbations and controls stay in train, the combinations are the held-out unit, and each combination is in the test set of one of the two folds, with a small held-out part used for validation.

A.3. Models

We benchmark three Deep Learning models: scGPT (Cui et al., 2024), PRESAGE (Littman et al., 2025), and GEARS (Roohani et al., 2024). All models were trained using their default hyperparameters within the model-specific Docker container provided by the Miller et al. (2025) codebase.

A.4. Incomplete model coverage

The deep-learning models cover neither the full gene panel nor the same set of perturbations. We make two decisions so that every method is scored on an identical set of genes and an identical set of perturbations.

Missing genes. scGPT does not emit a prediction for every gene in the panel; on wessels23 it covers 7 453 of the 8 210 genes ($\sim$9% missing), with comparable gaps on the other datasets. Miller et al. (2025) score scGPT only on the genes it predicts and leave the remainder undefined, so scGPT is evaluated on a strictly easier gene subset than the other methods; because the missing genes are disproportionately high-weight DEGs, this deflates its weighted error. Vollenweider & Bühlmann (2026) instead impute the missing genes with the dataset-mean baseline so that all methods share one gene set; under their delta-vs-mean metrics this imputation contributes exactly zero signal. Our metrics are delta-vs-control, for which the analogous uninformative value is the control profile itself, so *we impute every missing gene with a zero delta* (equivalently, the ZERO baseline). On the genes it does not predict scGPT then ties ZERO and can only gain a margin over the baselines on the genes it actually predicts—the honest reading of a declined prediction.

Missing perturbations. The methods also resolve different perturbations, so averaging a metric over each method's own coverage would compare them on different sets. We instead score every method on the *overlapping-perturbation set*—the perturbations that all methods resolve—and report every metric on it. A perturbation leaves this set when (i) its target is stored under an outdated symbol that no longer matches the panel, so the target-aware baselines cannot resolve it (we do not map aliases to recover these); (ii) its target is not in scGPT's modeled gene set, or is modeled but unsupported as a perturbation (scGPT resolves the fewest); (iii) its target is absent from GEARS's prior graph; or (iv) a method predicts it in a different fold than the one we assigned. Unlike the missing-gene case we do *not* impute a missing perturbation with a baseline, as that would credit a method with a baseline's score for a perturbation it never modeled. For example, adamson16 has 89 single-gene perturbations (some prior work reports $\sim$91 (Miller et al., 2025; Vollenweider & Bühlmann, 2026) by also counting the control and applying no minimum-cell filter); requiring coverage by every method leaves 75, as scGPT supports 76 and GEARS's prior graph excludes one more.

B. Methods

B.1. Operation space

All predictors, models and baselines alike, operate on the perturbation response y: the difference between perturbed and unperturbed (control) pseudobulk expression, represented as a single vector per cell type per perturbation.

Pseudobulk. Ground-truth profiles are pre-processed into pseudobulk vectors $\mu_{c,p}$ by averaging single-cell expression within each (cell type, perturbation) condition.

Response. The response $y_{c,p} = \mu_{c,p} - \mu_{c,\varnothing}$ is the difference between the perturbed pseudobulk and the matched control pseudobulk.

B.2. Baselines

Let $\boldsymbol{y} \in \mathbb{R}^{C\times P\times G}$ denote the tensor of perturbation effects, where C is the number of cell types, P the number of perturbations, and G the number of genes. Entry $y_{c,p,g}$ is the log-fold-change of gene g in cell type c under perturbation p relative to the unperturbed control:

$$y_{c,p,g} \;=\; \mu_{c,p,g} \;-\; \mu_{c,\varnothing,g},$$

where $\mu_{c,p}$ denotes the pseudobulk (cell-averaged) log-normalized expression of condition (c, p) and $\varnothing$ the control condition. We denote the set of training (cell type, perturbation) pairs by $\mathcal{T}$, and write $\mathcal{C}_{\text{train}}$ and $\mathcal{P}_{\text{train}}$ for the sets of training cell-type and perturbation indices, respectively.

B.3. Controls

Controls bracket the performance scale: the *negative control* defines the floor (no biological signal), and *positive controls* define the ceiling (best achievable given finite sampling noise).

Negative control: Zero. Predicts no perturbation effect whatsoever:

$$\hat{y}^{\text{Zero}}_{c,p,g} \;=\; 0.$$

Positive control: Technical duplicate (Tech-dup). The N_{cell} cells available for each (cell type, perturbation) condition are randomly split into two disjoint halves A and B of size $\lfloor N_{\text{cell}}/2 \rfloor$. Pseudobulk deltas are computed independently from each half:

$$y^A_{c,p,g} = \mu^A_{c,p,g} - \mu^A_{c,\varnothing,g}, \qquad y^B_{c,p,g} = \mu^B_{c,p,g} - \mu^B_{c,\varnothing,g}.$$

Half A serves as the ground-truth target and half B as the prediction. Because both halves measure the same biological condition, the only discrepancy arises from finite cell-sampling noise. Tech-dup therefore estimates the performance ceiling imposed by stochastic cell-to-cell variability.

Positive control: Interpolated duplicate (Interp-dup). A smoother alternative that combines the Tech-dup signal on differentially expressed genes (DEGs) with the Mean-over-Perturbations baseline elsewhere:

$$\hat{y}^{\text{Interp}}_{c,p,g} \;=\; m_{p,g}\, y^B_{c,p,g} \;+\; (1 - m_{p,g})\, \hat{y}^{\text{MoP}}_{c,p,g},$$

where $m_{p,g} \in \{0, 1\}$ is a binary DEG mask (1 if gene g is differentially expressed under perturbation p, determined by t-test with Benjamini–Hochberg correction or magnitude thresholding). This control is less noisy than Tech-dup on the majority of genes with negligible perturbation effects, while preserving the biological signal on DEGs.

B.4. Non-learned baselines

Each evaluation regime has a corresponding MSE-optimal simple-rule baseline, i.e., the conditional expectation $\mathbb{E}[y \mid \text{info available at test time}]$. Table 3 summarizes the correspondence.

Table 3. MSE-optimal non-learned baseline for each scenario.

Evaluation regime	Known at test time	Optimal predictor	Baseline
UNSEENPERT	Cell type only	$\mathbb{E}[y \mid c]$	Mean-over-Perturbations
UNSEENCELL	Perturbation only	$\mathbb{E}[y \mid p]$	Mean-over-Cell-Types
UNSEENBOTH	Neither	$\mathbb{E}[y]$	Grand Mean
UNSEENPAIR	Both (unseen combination)	$\mathbb{E}[y \mid c] + \mathbb{E}[y \mid p] - \mathbb{E}[y]$	Two-way Mean
UNSEENCOMBO	Single-perturbation effects	$y_A + y_B$	Additive

We do not evaluate UNSEENDOSE, which requires per-perturbation dose or knockdown-strength measurements that the datasets here do not provide; it is omitted from the baseline table above.

Mean-over-Perturbations (MoP). Averages over all training perturbations within each cell type:

$$\hat{y}^{\text{MoP}}_{c,p,g} \;=\; \frac{1}{|\mathcal{P}_{\text{train}}|} \sum_{p' \in \mathcal{P}_{\text{train}}} y_{c,p',g}.$$

For unseen cell types ($c \notin \mathcal{C}_{\text{train}}$), falls back to the global mean across all training cell types and perturbations.

Mean-over-Cell-Types (MoCT). Averages over all training cell types for each perturbation:

$$\hat{y}^{\text{MoCT}}_{c,p,g} = \frac{1}{|\mathcal{C}_{\text{train}}|} \sum_{c' \in \mathcal{C}_{\text{train}}} y_{c',p,g}.$$

Requires that each test perturbation was observed during training (otherwise undefined).

Grand Mean. A single prediction vector for all (cell type, perturbation) pairs:

$$\hat{y}^{\text{GM}}_{c,p,g} = \frac{1}{|\mathcal{C}_{\text{train}}|\,|\mathcal{P}_{\text{train}}|} \sum_{c' \in \mathcal{C}_{\text{train}}} \sum_{p' \in \mathcal{P}_{\text{train}}} y_{c',p',g}.$$

Two-way Mean. An additive row–column decomposition of the training tensor:

$$\hat{y}^{\text{2W}}_{c,p,g} = \mu_g + \alpha_{c,g} + \beta_{p,g},$$

where μ_g is the grand mean, $\alpha_{c,g}$ is the cell-type effect, and $\beta_{p,g}$ is the perturbation effect:

$$\begin{aligned}
\mu_g &= \frac{1}{|\mathcal{T}|} \sum_{(c',p') \in \mathcal{T}} y_{c',p',g}, \\
\alpha_{c,g} &= \frac{1}{|\{p' : (c,p') \in \mathcal{T}\}|} \sum_{\{p' : (c,p') \in \mathcal{T}\}} y_{c,p',g} - \mu_g, \\
\beta_{p,g} &= \frac{1}{|\{c' : (c',p) \in \mathcal{T}\}|} \sum_{\{c' : (c',p) \in \mathcal{T}\}} y_{c',p,g} - \mu_g.
\end{aligned} \tag{7}$$

This is equivalent to MoP + MoCT − Grand Mean and is the MSE-optimal predictor when both cell type and perturbation identity are known but their specific combination was not observed during training (UNSEENPAIR).

Additive (for combinatorial perturbations). For a combinatorial perturbation (A, B) whose constituent singles are observed:

$$\hat{y}^{\text{Add}}_{c,(A,B),g} = y_{c,A,g} + y_{c,B,g}.$$

Assumes perturbation effects compose without epistasis, i.e., the joint effect equals the sum of marginal effects (UNSEENCOMBO).

Target Scaling. Each regime-matched mean baseline is paired with a *target-gene correction* that models the perturbation's direct effect on its own target gene(s), following the implementation of Vollenweider & Bühlmann (2026). For a perturbation p with target-gene set $\mathcal{G}_p$,

$$\hat{y}^{\text{TS}}_{c,p,g} = \begin{cases} -\,\alpha\,\mu_{c,\varnothing,g}, & g \in \mathcal{G}_p, \\ 0, & \text{otherwise}, \end{cases}$$

where $\mu_{c,\varnothing,g}$ is the control expression of gene g in context c and α is a single global scalar, shared across all perturbations and contexts, fit by ordinary least squares over the training target triples:

$$\alpha = -\frac{\sum_{(c,p,g):\, g \in \mathcal{G}_p} y_{c,p,g}\,\mu_{c,\varnothing,g}}{\sum_{(c,p,g):\, g \in \mathcal{G}_p} \mu^2_{c,\varnothing,g}}.$$

The regime-matched baselines reported in the main text are the composite **Mean + Target Scaling**: a regime mean ($\hat{y}^{\text{MoP}}$, $\hat{y}^{\text{MoCT}}$, $\hat{y}^{\text{GM}}$, or $\hat{y}^{\text{2W}}$) on all genes, with the term above added on the resolvable target genes (where a target is unresolvable the correction is omitted and the prediction falls back to the mean). For strong knockdowns $\alpha \approx 1$, driving the target gene toward zero expression ($y \approx -\mu_{c,\varnothing}$).

B.5. Learned baselines

Learned baselines use the training split to fit parameters, while remaining far simpler than full neural perturbation-prediction models. All use the same multi-hot feature encoding: $\mathbf{x} = [\mathbf{e}_p;\, \mathbf{e}_c] \in \{0,1\}^{P+C}$, where $\mathbf{e}_p$ and $\mathbf{e}_c$ are one-hot indicators for the perturbation and cell type, respectively.

Linear Additive. A linear model on the multi-hot encoding, trained by mini-batch SGD on MSE loss:

$$\hat{y}_g \;=\; \mathbf{w}_g^\top \mathbf{x} + c_g,$$

or equivalently $\hat{\boldsymbol{y}} = \mathbf{W}^\top \mathbf{x} + \mathbf{c}$, with $\mathbf{W} \in \mathbb{R}^{(P+C)\times G}$ and $\mathbf{c} \in \mathbb{R}^G$.

Ridge. The same multi-hot feature encoding but solved in closed form with ℓ_2 regularization:

$$\mathbf{W} = \left(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}\right)^{-1} \mathbf{X}^\top \mathbf{Y}_c,$$

where $\mathbf{X} \in \{0,1\}^{N\times(P+C)}$ is the design matrix over all $N = |\mathcal{T}|$ training samples, $\mathbf{Y}_c$ is the mean-centered target matrix, and λ controls regularization strength.

Latent Additive. Three two-layer MLPs (ReLU activations, hidden dimension h, latent dimension d) that enforce additivity in a learned latent space:

$$\mathbf{z} = f_{\text{gene}}(\mathbf{x}^{\text{ctrl}}) + f_{\text{pert}}(\mathbf{x}), \qquad \hat{\boldsymbol{y}} = f_{\text{dec}}(\mathbf{z}),$$

where $f_{\text{gene}} : \mathbb{R}^G \to \mathbb{R}^d$ encodes the control expression profile, $f_{\text{pert}} : \{0,1\}^{P+C} \to \mathbb{R}^d$ encodes perturbation identity, and $f_{\text{dec}} : \mathbb{R}^d \to \mathbb{R}^G$ decodes the predicted delta. By summing in latent space, the model can learn nonlinear gene-expression features while preserving an additive perturbation structure. Trained end-to-end via mini-batch SGD on MSE loss ($h = 256$, $d = 128$).

B.6. Performance metrics - full list

We evaluate every predictor against ground truth using a catalog of metrics drawn from the perturbation-prediction literature. Throughout this appendix we fix notation as follows. Let $k = (c, p)$ index a test item (cell type c, perturbation p), let G denote the number of genes, and let $g \in \{1, \ldots, G\}$ index a gene. We write $\hat{y}_{k,g}$ for the predicted response and $y_{k,g}$ for the ground-truth response (log-fold-change with respect to the unperturbed control). Each metric below is defined as a per-row scalar $M(\hat{\boldsymbol{y}}_k, \boldsymbol{y}_k)$ acting on a single test item (or, for cross-row metrics, on the full $K \times G$ tensor). Where relevant, $w_g \geq 0$ denotes a per-gene weight and $\mathbf{1}[\cdot]$ an indicator. A gene mask $m_{k,g} \in \{0,1\}$ may restrict the sum to a subset of genes; unless stated otherwise, unmasked genes contribute uniformly ($w_g = 1$, $m_{k,g} = 1$ for all g). All modifiers (masking, weighting, stratification) are deferred to Section C, and every base metric below may be combined with any modifier listed there.

B.6.1. Point-wise distribution metrics

Element-wise comparisons between predicted and ground-truth deltas.

Mean Squared Error (MSE). Lower is better; perfect value 0.

$$\text{MSE}_k \;=\; \frac{\sum_g w_g\, m_{k,g} \left(\hat{y}_{k,g} - y_{k,g}\right)^2}{\sum_g w_g\, m_{k,g}}.$$

Mean Absolute Error (MAE). Lower is better; perfect value 0.

$$\text{MAE}_k \;=\; \frac{\sum_g w_g\, m_{k,g} \left|\hat{y}_{k,g} - y_{k,g}\right|}{\sum_g w_g\, m_{k,g}}.$$

Fold-change Gap. Mean absolute error averaged across four equally populated bins of $|y_{k,g}|$ (quartiles of the ground-truth magnitude distribution). Let $B_0, \ldots, B_3$ denote the quartile index sets of $|y_{k,\cdot}|$.

$$\text{FCG}_k \;=\; \frac{1}{4} \sum_{b=0}^{3} \frac{1}{|B_b|} \sum_{g \in B_b} \left|\hat{y}_{k,g} - y_{k,g}\right|.$$

This reweights error so that the high-magnitude tail (a few strongly affected genes) is not drowned by the bulk of unaffected genes. Lower is better; perfect value 0.

B.6.2. CORRELATION METRICS

Shape- or rank-alignment between prediction and target, ignoring absolute scale. Let weighted moments be $\bar{a}_k = \sum_g w_g m_{k,g}\, a_{k,g} \big/ \sum_g w_g m_{k,g}$ for any quantity a.

Pearson correlation. Higher is better; perfect value 1.

$$r_k \;=\; \frac{\sum_g w_g m_{k,g}\,\big(\hat{y}_{k,g} - \bar{\hat{y}}_k\big)\big(y_{k,g} - \bar{y}_k\big)}{\sqrt{\sum_g w_g m_{k,g}\big(\hat{y}_{k,g} - \bar{\hat{y}}_k\big)^2 \;\cdot\; \sum_g w_g m_{k,g}\big(y_{k,g} - \bar{y}_k\big)^2 + \varepsilon}}.$$

Centered R^2. The textbook coefficient of determination, with the target mean as null:

$$R^2_{\mathrm{c},\,k} \;=\; 1 \;-\; \frac{\sum_g w_g m_{k,g}\big(y_{k,g} - \hat{y}_{k,g}\big)^2}{\sum_g w_g m_{k,g}\big(y_{k,g} - \bar{y}_k\big)^2 + \varepsilon}.$$

$R^2_{\mathrm{c}} = 0$ corresponds to predicting the per-row mean. Higher is better; perfect value 1; unbounded below.

Concordance Correlation Coefficient (CCC). Lin's concordance coefficient penalises both scale and location mismatch:

$$\mathrm{CCC}_k \;=\; \frac{2\,\mathrm{cov}_w(\hat{y}_k, y_k)}{\mathrm{var}_w(\hat{y}_k) + \mathrm{var}_w(y_k) + \big(\bar{\hat{y}}_k - \bar{y}_k\big)^2 + \varepsilon},$$

where cov_w and var_w denote the weighted covariance and variance under $w_g m_{k,g}$. Higher is better; perfect value 1.

B.6.3. DISCRIMINATION METRICS

Whether a prediction's vector retains enough identity to distinguish its own perturbation from others (PDS) or to flag which genes respond to the perturbation (effect-size AUROC).

Perturbation Discrimination Score (PDS). For each predicted profile $\hat{\boldsymbol{y}}_k$ we find the nearest ground-truth profile $\boldsymbol{y}_j$ under a chosen distance d and ask whether $j = k$:

$$\mathrm{PDS} \;=\; \frac{1}{K}\sum_{k=1}^{K} \mathbf{1}\Big[k = \arg\min_j\; d\big(\hat{\boldsymbol{y}}_k, \boldsymbol{y}_j\big)\Big].$$

We evaluate three distances (yielding `pds_cosine`, `pds_l2`, `pds_l1`):

$$\begin{aligned} d_{\cos}(u,v) \;&=\; 1 - \frac{u^\top v}{\|u\|_2\,\|v\|_2 + \varepsilon}, \\ d_{L_2}(u,v) \;&=\; \|u - v\|_2, \\ d_{L_1}(u,v) \;&=\; \|u - v\|_1. \end{aligned}$$

If $K > K_{\max}$ ($K_{\max} = 200$ by default), a random subsample of $K_{\max}$ items is drawn with a fixed seed. Higher is better; perfect value 1.

Effect-size AUROC. Measures the separability of responding vs non-responding genes under the prediction. For each row, the binary label is $z_{k,g} = \mathbf{1}[\,|y_{k,g}| > \theta\,]$ and the score is $s_{k,g} = |\hat{y}_{k,g}|$ (threshold $\theta = 0.5$ by default). Using midrank-based Mann–Whitney U with $n_k^+ = \sum_g z_{k,g}$ and $n_k^- = G - n_k^+$:

$$\mathrm{AUROC}_k \;=\; \frac{U_k}{n_k^+\, n_k^- + \varepsilon}, \qquad U_k \;=\; \sum_g \mathrm{rank}(s_{k,\cdot})_g\, z_{k,g} \;-\; \tfrac{1}{2} n_k^+\big(n_k^+ - 1\big).$$

Rows with $n_k^+ = 0$ or $n_k^- = 0$ return 0.5. Higher is better; perfect value 1.

B.6.4. EFFECT-SIZE METRICS

Recovery of the *shape* of the per-perturbation response—its direction and magnitude distribution—without requiring per-gene alignment.

Fraction of correct direction. Among genes with a non-zero ground-truth delta, the fraction on which the predicted sign agrees:

$$\mathrm{FCD}_k \;=\; \frac{\big|\{g : y_{k,g} \neq 0,\ \mathrm{sign}(\hat{y}_{k,g}) = \mathrm{sign}(y_{k,g})\}\big|}{\big|\{g : y_{k,g} \neq 0\}\big|}.$$

Rows with fewer than $n_{\min}$ signed genes return NaN (default $n_{\min} = 5$). Higher is better; perfect value 1.

Variance-ratio log error. A global (cross-row) diagnostic that penalises prediction variance collapse or inflation:

$$\mathrm{VRLE} \;=\; \underset{g\,:\,\mathrm{Var}_k(y_{k,g}) \geq \sigma^2_{\min}}{\mathrm{median}} \left|\log \frac{\mathrm{Var}_k(\hat{y}_{k,g}) + \varepsilon}{\mathrm{Var}_k(y_{k,g}) + \varepsilon}\right|,$$

where Var_k is the variance across test items for a fixed gene and $\sigma^2_{\min} = 10^{-4}$ is a variance floor excluding genes whose ground-truth variance is numerically negligible. Lower is better; perfect value 0.

B.6.5. ENERGY DISTANCE METRICS

Cross-row distances comparing the empirical distribution of predicted profiles to that of ground-truth profiles, treating the K rows as samples from two distributions on $\mathbb{R}^G$.

E-distance. Szekely's energy distance in the full gene space:

$$\mathrm{Edist}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \;=\; \frac{2}{K^2}\sum_{i,j}\big\|\hat{\boldsymbol{y}}_i - \boldsymbol{y}_j\big\|_2 \;-\; \frac{1}{K^2}\sum_{i,i'}\big\|\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_{i'}\big\|_2 \;-\; \frac{1}{K^2}\sum_{j,j'}\big\|\boldsymbol{y}_j - \boldsymbol{y}_{j'}\big\|_2.$$

Zero iff the two empirical distributions coincide. Lower is better; perfect value 0.

E-distance in PCA space. Identical to EDIST but computed on a PCA projection of rank n_{PC} (default 50). If μ and $V_{n_{\mathrm{PC}}}$ are the mean and principal-component matrix fit on the ground-truth tensor, we project $\tilde{\boldsymbol{y}}_i = (\boldsymbol{y}_i - \mu)V_{n_{\mathrm{PC}}}^{\top}$ and similarly for $\hat{\boldsymbol{y}}$, then

$$\mathrm{Edist}_{\mathrm{PCA}} \;=\; \mathrm{Edist}(\tilde{\hat{\boldsymbol{y}}}, \tilde{\boldsymbol{y}}).$$

This reduces gene-space isotropy and emphasises the directions of genuine biological variation.

B.6.6. DE RECALL METRICS

Whether the top of the predicted ranking is enriched for genes that the ground truth flags as differentially expressed (DEGs).

GSEA enrichment score. For each row k, let $S_k \subseteq \{1, \ldots, G\}$ be the signed DEG set and let $\pi_k \in \mathfrak{S}_G$ be the permutation that sorts $\{\hat{y}_{k,g}\}_{g=1}^{G}$ in descending order. Walking the ranked list from rank 1 to rank G, each step contributes

$$r_{k,j} \;=\; \begin{cases} +1/|S_k| & \text{if } \pi_k(j) \in S_k, \\ -1/(G - |S_k|) & \text{otherwise,} \end{cases}$$

and the enrichment score is the maximum of the running cumulative sum:

$$\mathrm{ES}_k \;=\; \max_{1 \leq j \leq G} \sum_{i=1}^{j} r_{k,i}.$$

Rows with $|S_k| < n_{\text{set,min}}$ (default 5) return NaN. Higher is better; perfect value 1.

GSEA-up. Uses $S_k = \{g : m^{\text{deg}}_{k,g} = 1 \text{ and } y_{k,g} > 0\}$ and ranks predictions in descending order: rewards predictions that place upregulated DEGs at the *top*.

GSEA-down. Uses $S_k = \{g : m^{\text{deg}}_{k,g} = 1 \text{ and } y_{k,g} < 0\}$ and ranks predictions in ascending order ($\arg\text{sort}(-\hat{y})$ descending), rewarding predictions that place downregulated DEGs at the top. In both cases m^{deg} is the per-row DEG mask described in Section C.

C. Metric modifiers: DEGs, weighting, and masking

Composable base metrics (MSE, Pearson, CCC, etc.) can be combined with modifiers that restrict the gene set or reweight genes. This section describes the modifiers used in the evaluation pipeline.

C.1. Differentially expressed genes (DEGs)

DEGs are computed per perturbation from the second-half single-cell split (held out from pseudobulk ground truth) using scanpy's `rank_genes_groups` with `method='t-test_overestim_var'` and `reference='rest'` (Wolf et al., 2018). This performs a Welch t-test with overestimated variance (conservative) comparing perturbed cells against the rest of the dataset, with Benjamini–Hochberg correction for multiple testing. A gene is classified as a DEG for perturbation k if its adjusted p-value satisfies $p_{\text{adj}} < 0.05$.

From the DEG analysis we derive three arrays stored in the HDF5 file:

DEG mask. A binary matrix $m^{\text{deg}}_{k,g} \in \{0, 1\}$ indicating whether gene g is differentially expressed under perturbation k ($p_{\text{adj}} < 0.05$). Used by GSEA metrics and fraction correct direction to restrict evaluation to genes with detectable perturbation effects.

DEG directions. A signed matrix $d^{\text{deg}}_{k,g} \in \{-1, 0, +1\}$: $+1$ if gene g is significantly upregulated (positive t-score and $p_{\text{adj}} < 0.05$), -1 if significantly downregulated, and 0 otherwise. Used by fraction correct direction to define the ground-truth sign of change.

Per-perturbation weights. For DEG-weighted metrics (denoted by the `_weighted` suffix), each gene receives a weight derived from the t-test score:

$$w_{k,g} = \left(\frac{|t_{k,g}| - \min_g |t_{k,g}|}{\max_g |t_{k,g}| - \min_g |t_{k,g}|} \right)^2,$$

where $t_{k,g}$ is the t-score for gene g under perturbation k. This min–max normalization followed by squaring concentrates weight on the most strongly affected genes while retaining contributions from moderately affected ones. When the range of $|t|$-scores is zero (constant across genes), all weights are set to zero. The weighted metric is then:

$$M_k^{\text{weighted}} = \frac{\sum_g w_{k,g} \, f(\hat{y}_{k,g}, y_{k,g})}{\sum_g w_{k,g}},$$

where f is the base metric's per-gene function (e.g., squared error for MSE, or the contribution to the correlation numerator for Pearson).

C.2. Variance-based weighting

For variance-weighted metrics (denoted `_var_weighted`), genes are weighted by the inverse of their cross-perturbation variance:

$$w_g^{\text{var}} = \frac{1}{1 + \text{Var}_k(y_{k,g})},$$

where Var_k is computed across all test perturbations. This downweights highly variable genes (which tend to be noisy) and upweights genes with stable expression changes.

C.3. Gene masking

Gene masks restrict the metric computation to a subset of genes:

- **Top-k most affected genes per perturbation** (`_top_200`): for each perturbation k, retain only the $k = 200$ genes with the largest $|y_{k,g}|$. This focuses evaluation on genes with the strongest ground-truth response.
- **Top-k expressed genes globally** (`_top_1000_expressed`): retain the 1,000 genes with the highest mean expression across control cells. This restricts evaluation to well-measured genes.

D. Baseline Saturation

Baseline Saturation is measured per perturbation and operates on per-item performance metric values. Let p index a test perturbation (or, for multi-cell-type datasets, a test item (c, p)). For a given evaluation metric, we write neg_p, pos_p, and baseline_p for the metric values of the negative control, positive control, and regime-matched baseline on perturbation p.

Positive-control selection. We evaluate two positive controls: Tech-duplicate (y^B, the split-half replicate) and Interp-duplicate (a smoothed variant that uses the Tech-duplicate signal on DEGs and MoP elsewhere; see §B.2). For each metric, we select the positive control whose median value is better (higher for higher-is-better, lower for lower-is-better); this selection is performed once per metric across all test items.

Definition. Using an oriented metric $\tilde{m}$ (larger is better: $\tilde{m} = m$ for higher-is-better metrics and $\tilde{m} = -m$ for lower-is-better metrics), the per-item Baseline Saturation is the single control-anchored ratio

$$s_{c,p,m} = \text{clip}_{[0,1]}\left(\frac{\tilde{m}(b_{c,p}, y_{c,p}) - \tilde{m}(0, y_{c,p})}{\tilde{m}(y^{+}_{c,p}, y_{c,p}) - \tilde{m}(0, y_{c,p}) + \varepsilon}\right), \qquad \varepsilon = 10^{-8}, \tag{8}$$

matching the main-text definition, where $b_{c,p}$ is the regime-matched baseline prediction, 0 the negative control (Zero), and $y^{+}_{c,p}$ the positive control.

Baseline pool. Baseline Saturation is computed only over the non-learned simple-rule baselines: Mean-over-Perturbations, Mean-over-Cell-Types, Grand Mean, Two-way Mean, and Additive (combinatorial perturbations only), each with the Target Scaling correction of §B.4. Learned models (Linear Additive, Ridge, Latent Additive) are excluded.

Best-baseline selection. For each metric, Baseline Saturation is computed independently for every baseline in the pool; the regime-matched baseline reported is the one maximizing $\text{mean}(s_{c,p,m})$ across all retained perturbations (a global choice per metric, not per perturbation).

Aggregation. Per-perturbation Baseline Saturation values are clipped to $[0, 1]$ and aggregated by the median across retained perturbations.

E. Existing Assets and Licenses

F. Compute resources

Experiments were run on an internal GPU cluster using up to 8 NVIDIA A100 GPUs with 80GB memory each. CPU preprocessing, pseudobulk construction, baseline evaluation, metric computation, and figure generation were run on standard cluster CPU nodes with 32–64 CPU cores and 256–512GB RAM. Across datasets, raw inputs, intermediate pseudobulk files, model outputs, and generated figures required approximately 1–2TB of storage.

Table 4. Existing datasets and model assets used in this work. We do not redistribute raw datasets, pretrained weights, or external model assets; users should obtain them from the original sources.

Asset	Type	Source / identifier	License or usage terms	Redistributed?
Adamson16	Dataset	NCBI GEO: GSE90546	GEO terms[a]	No
Frangieh21	Dataset	Broad Single Cell Portal: SCP1064; raw data via DUOS-000124	SCP / DUOS terms	No
Norman19	Dataset	NCBI GEO: GSE133344	GEO terms[a]	No
Wessels23	Dataset	NCBI GEO: GSE213957	GEO terms[a]	No
Replogle20	Dataset	NCBI GEO: GSE146194	GEO terms[a]	No
Replogle22	Dataset	Figshare+; DOI in original source	CC BY 4.0	No
McFaline23	Dataset	NCBI GEO: GSE114687	GEO terms[a]	No
Jiang24 / PerturBench	Dataset / benchmark source	PerturBench processed datasets via Hugging Face / Lamin	Original data terms; PerturBench code: BSD 3-Clause	No
scGPT	Model / code	`bowang-lab/scGPT`	MIT; checkpoint terms separate	No
GEARS	Model / code	`snap-stanford/GEARS`	MIT	No
PRESAGE	Model / code	`Genentech/PRESAGE`	Genentech Non-Commercial Software License v1.0	No

[a]*GEO terms* denotes the NCBI GEO public-access terms, subject to any submitter-retained rights. For controlled-access or portal-hosted datasets, users should follow the terms provided by the original data source.

G. Raw metric scores

Table 5. Adamson16

	$\mathrm{MSE}_{\mathrm{wt}}$	$\mathrm{Pearson}_{\mathrm{wt}}$	$\mathrm{CCC}_{\mathrm{wt}}$	Frac. dir.	Cos. rank	$\mathrm{GSEA}_{\uparrow}$	$\mathrm{GSEA}_{\downarrow}$
UNSEENPERT ($n=80$, 5f)							
Zero	0.0302	0.000	0.000	0.000	0.500	0.090	0.079
Tech-dup	0.0012	0.981	0.978	0.969	0.000	0.882	0.869
Interp-dup	0.0018	0.968	0.965	0.969	0.000	0.916	0.906
MoP	0.0324	0.103	0.042	0.379	0.500	0.419	0.164
Target Scaling	0.0178	0.561	0.315	0.379	0.000	0.419	0.244
GEARS	0.0236	0.010	0.004	0.354	0.438	0.375	0.166
scGPT	0.0148	0.360	0.262	0.392	0.143	0.338	0.183
PRESAGE	0.0295	0.128	0.061	0.411	0.118	0.461	0.269
Ridge	0.0324	0.103	0.042	0.379	0.500	0.419	0.164
Bilinear Ridge	0.0248	0.254	0.112	0.473	0.000	0.534	0.292
Linear Add.	0.0324	0.103	0.042	0.379	0.500	0.419	0.164
Correlation	0.0490	0.199	0.119	0.614	0.559	0.113	0.225
Latent Add.	0.0285	0.101	0.019	0.389	0.529	0.420	0.163

Table 6. Frangieh21

	MSE_{wt}	$\text{Pearson}_{\text{wt}}$	CCC_{wt}	Frac. dir.	Cos. rank	$\text{GSEA}_{\uparrow}$	$\text{GSEA}_{\downarrow}$
UNSEENPERT ($n=39$, 5f)							
Zero	0.0037	0.000	0.000	0.000	0.500	0.211	—
Tech-dup	0.0035	0.215	0.188	1.000	0.213	1.000	—
Interp-dup	0.0030	0.280	0.101	1.000	0.330	1.000	—
MoP	0.0033	0.180	0.067	0.500	0.500	0.099	—
Target Scaling	0.0032	0.273	0.108	1.000	0.032	0.099	—
GEARS	0.0045	0.037	0.021	0.000	0.564	0.214	—
scGPT	0.0041	0.167	0.120	1.000	0.209	—	—
PRESAGE	0.0034	0.161	0.059	0.500	0.444	0.200	—
Ridge	0.0033	0.180	0.067	0.500	0.500	0.099	—
Bilinear Ridge	0.0031	0.288	0.103	1.000	0.000	0.955	—
Linear Add.	0.0033	0.180	0.067	0.500	0.500	0.099	—
Correlation	0.0062	0.043	0.017	1.000	0.415	0.420	—
Latent Add.	0.0035	0.181	0.023	0.250	0.489	0.100	—

Table 7. Jiang24

	MSE_{wt}	$\text{Pearson}_{\text{wt}}$	CCC_{wt}	Frac. dir.	Cos. rank	$\text{GSEA}_{\uparrow}$	$\text{GSEA}_{\downarrow}$
UNSEENPERT ($n=216$, 5f)							
Zero	0.0560	0.000	0.000	0.000	0.500	0.066	0.071
Tech-dup	0.0012	0.988	0.988	1.000	0.000	0.942	0.928
Interp-dup	0.0015	0.987	0.986	1.000	0.000	0.971	0.969
MoP	0.0555	0.114	0.025	0.510	0.500	0.420	0.298
Target Scaling	0.0552	0.134	0.034	0.513	0.143	0.420	0.302
GEARS	0.0552	0.026	0.007	0.500	0.405	0.307	0.185
scGPT	0.1505	-0.014	-0.008	0.497	0.571	0.039	0.492
PRESAGE	0.0434	0.534	0.182	0.664	0.190	0.506	0.373
Ridge	0.0555	0.110	0.025	0.511	0.500	0.420	0.298
Bilinear Ridge	0.0534	0.609	0.077	0.704	0.071	0.684	0.413
Linear Add.	0.0555	0.114	0.025	0.510	0.500	0.420	0.298
Correlation	0.0621	-0.160	-0.009	0.372	0.667	0.032	0.057
Latent Add.	0.0574	-0.190	-0.016	0.486	0.524	0.386	0.140
UNSEENCELL ($n=216$, 5f)							
Zero	0.0569	0.000	0.000	0.000	0.500	0.058	0.059
Tech-dup	0.0010	0.991	0.990	1.000	0.000	0.938	0.924
Interp-dup	0.0013	0.991	0.990	1.000	0.000	0.972	0.969
MoCT	0.0332	0.639	0.508	0.746	0.074	0.458	0.426
Target Scaling	0.0557	-0.134	-0.013	0.439	0.186	0.320	0.075
GEARS	0.1412	0.105	0.082	0.584	0.253	0.128	0.569
scGPT	0.1849	-0.116	-0.077	0.437	0.498	0.081	0.347
PRESAGE	0.1749	0.184	0.148	0.624	0.117	0.104	0.698
Ridge	0.0385	0.638	0.434	0.750	0.086	0.451	0.388
Bilinear Ridge	0.0616	-0.323	-0.121	0.403	0.844	0.147	0.127
Linear Add.	0.0332	0.638	0.508	0.752	0.081	0.454	0.394
Correlation	0.0651	-0.147	-0.007	0.432	0.681	0.060	0.100
Latent Add.	0.0559	-0.144	-0.011	0.441	0.519	0.317	0.071
UNSEENBOTH ($n=213$, 5f)							
Zero	0.0529	0.000	0.000	0.000	0.500	0.056	0.064
Tech-dup	0.0010	0.991	0.991	1.000	0.000	0.938	0.927
Interp-dup	0.0012	0.991	0.990	1.000	0.000	0.975	0.970
Grand Mean	0.0552	0.020	0.002	0.441	0.500	0.295	0.082
Target Scaling	0.0552	0.062	0.006	0.441	0.214	0.295	0.083
GEARS	0.1162	0.180	0.126	0.589	0.238	0.133	0.605
scGPT	0.1532	0.013	0.013	0.513	0.595	0.040	0.478
PRESAGE	0.1289	0.259	0.214	0.647	0.119	0.167	0.676
Ridge	0.0516	0.057	0.007	0.474	0.500	0.271	0.101
Bilinear Ridge	0.0579	-0.157	-0.010	0.439	0.667	0.280	0.099
Linear Add.	0.0516	0.058	0.007	0.474	0.500	0.271	0.101
Correlation	0.0558	-0.131	-0.003	0.418	0.667	0.058	0.095
Latent Add.	0.0550	0.016	0.001	0.442	0.548	0.296	0.082
UNSEENPAIR ($n=216$, 5f)							
Zero	0.0535	0.000	0.000	0.000	0.500	0.064	0.071
Tech-dup	0.0012	0.988	0.987	1.000	0.000	0.942	0.928
Interp-dup	0.0014	0.987	0.986	1.000	0.000	0.973	0.971
MoP	0.0540	0.004	0.001	0.500	0.500	0.401	0.285
MoCT	0.0278	0.628	0.509	0.750	0.088	0.447	0.418
Two-way Mean	0.0276	0.671	0.541	0.762	0.091	0.508	0.524
GEARS	0.0370	0.658	0.400	0.755	0.189	0.558	0.481
scGPT	0.1473	-0.017	-0.012	0.495	0.500	0.032	0.495
PRESAGE	0.1654	0.172	0.122	0.600	0.195	0.128	0.602
Ridge	0.0280	0.674	0.479	0.759	0.091	0.528	0.528
Bilinear Ridge	0.0762	-0.043	-0.040	0.623	0.281	0.511	0.470
Linear Add.	0.0266	0.675	0.543	0.758	0.091	0.516	0.525
Correlation	0.0623	-0.133	-0.003	0.393	0.682	0.034	0.079
Latent Add.	0.0567	-0.144	-0.013	0.484	0.511	0.412	0.130

Table 8. McFaline23

	MSE_{wt}	$\text{Pearson}_{\text{wt}}$	CCC_{wt}	Frac. dir.	Cos. rank	$\text{GSEA}_{\uparrow}$	$\text{GSEA}_{\downarrow}$
UNSEENPERT ($n=511$, 5f)							
Zero	0.0038	0.000	0.000	0.000	0.500	—	—
Tech-dup	0.0050	0.038	0.034	1.000	0.426	—	—
Interp-dup	0.0037	0.082	0.016	1.000	0.500	—	—
MoP	0.0037	0.082	0.016	0.000	0.500	—	—
Target Scaling	0.0037	0.087	0.017	1.000	0.255	—	—
GEARS	0.0039	0.034	0.015	1.000	0.490	—	—
scGPT	0.0091	0.009	0.008	1.000	0.510	—	—
PRESAGE	0.0085	0.013	0.013	0.000	0.510	—	—
Ridge	0.0037	0.082	0.016	0.000	0.500	—	—
Bilinear Ridge	0.0037	0.113	0.022	0.000	0.049	—	—
Linear Add.	0.0037	0.082	0.016	0.000	0.500	—	—
Correlation	0.0038	0.009	0.000	1.000	0.450	—	—
Latent Add.	0.0038	0.081	0.012	0.000	0.505	—	—
UNSEENCELL ($n=511$, 3f)							
Zero	0.0033	0.000	0.000	0.000	0.500	—	—
Tech-dup	0.0044	0.029	0.026	1.000	0.414	—	—
Interp-dup	0.0033	0.043	0.010	1.000	0.500	—	—
MoCT	0.0051	0.012	0.011	1.000	0.475	—	—
Target Scaling	0.0033	0.049	0.012	1.000	0.271	—	—
GEARS	0.0035	0.022	0.010	0.000	0.514	—	—
scGPT	0.0088	0.005	0.005	1.000	0.505	—	—
PRESAGE	0.0186	0.004	0.003	1.000	0.487	—	—
Ridge	0.0038	0.020	0.010	1.000	0.471	—	—
Bilinear Ridge	0.0033	0.031	0.008	1.000	0.500	—	—
Linear Add.	0.0051	0.012	0.011	1.000	0.475	—	—
Correlation	0.0033	0.007	0.000	1.000	0.459	—	—
Latent Add.	0.0033	0.042	0.006	0.000	0.497	—	—
UNSEENBOTH ($n=511$, 5f)							
Zero	0.0034	0.000	0.000	—	0.500	—	—
Tech-dup	0.0045	0.033	0.030	—	0.401	—	—
Interp-dup	0.0034	0.044	0.011	—	0.500	—	—
Grand Mean	0.0034	0.044	0.011	—	0.500	—	—
Target Scaling	0.0034	0.048	0.013	—	0.272	—	—
GEARS	0.0036	0.023	0.010	—	0.475	—	—
scGPT	0.0094	0.003	0.003	—	0.505	—	—
PRESAGE	0.0161	0.009	0.007	—	0.485	—	—
Ridge	0.0034	0.044	0.011	—	0.500	—	—
Bilinear Ridge	0.0034	0.044	0.011	—	0.500	—	—
Linear Add.	0.0034	0.044	0.011	—	0.500	—	—
Correlation	0.0034	0.008	0.000	—	0.446	—	—
Latent Add.	0.0034	0.044	0.006	—	0.510	—	—
UNSEENPAIR ($n=511$, 5f)							
Zero	0.0038	0.000	0.000	0.000	0.500	—	—
Tech-dup	0.0051	0.039	0.036	1.000	0.402	—	—
Interp-dup	0.0037	0.085	0.015	1.000	0.500	—	—
MoP	0.0037	0.084	0.015	0.000	0.500	—	—
MoCT	0.0047	0.005	0.004	1.000	0.478	—	—
Two-way Mean	0.0046	0.019	0.011	1.000	0.467	—	—
GEARS	0.0039	0.034	0.014	1.000	0.481	—	—
scGPT	0.0086	0.011	0.010	1.000	0.549	—	—
PRESAGE	0.0161	0.008	0.007	0.000	0.505	—	—
Ridge	0.0040	0.030	0.016	1.000	0.463	—	—
Bilinear Ridge	0.0040	0.026	0.007	0.000	0.500	—	—
Linear Add.	0.0047	0.021	0.016	1.000	0.473	—	—
Correlation	0.0039	0.012	0.000	1.000	0.471	—	—
Latent Add.	0.0037	0.078	0.012	0.000	0.480	—	—

Table 9. Norman19

	MSE_{wt}	$\text{Pearson}_{\text{wt}}$	CCC_{wt}	Frac. dir.	Cos. rank	$\text{GSEA}_{\uparrow}$	$\text{GSEA}_{\downarrow}$
UNSEENCOMBO ($n = 125$, 2f)							
Zero	0.0993	0.000	0.000	0.000	0.500	0.098	0.090
Tech-dup	0.0025	0.986	0.984	1.000	0.000	0.951	0.943
Interp-dup	0.0036	0.978	0.976	1.000	0.000	0.980	0.966
Additive	0.0090	0.960	0.941	0.997	0.000	0.936	0.923
Matching Mean	0.0251	0.955	0.754	0.990	0.000	0.918	0.910
Target Scaling	0.0708	0.576	0.219	0.703	0.000	0.588	0.394
Global Epistasis	0.0105	0.956	0.909	0.983	0.000	0.914	0.899
GEARS	0.0334	0.751	0.622	0.831	0.141	0.608	0.620
scGPT	0.0735	0.581	0.299	0.635	0.024	0.472	0.472
PRESAGE	0.0118	0.949	0.918	0.993	0.000	0.918	0.913
Ridge	0.0764	0.529	0.170	0.684	0.500	0.560	0.394
Bilinear Ridge	0.0623	0.780	0.339	0.811	0.000	0.737	0.572
Linear Add.	0.0764	0.529	0.170	0.684	0.500	0.560	0.394
Correlation	0.1153	-0.484	-0.072	0.194	0.794	0.000	0.022
Latent Add.	0.0915	0.528	0.059	0.666	0.461	0.557	0.373

Table 10. Replogle20

	MSE_{wt}	$\text{Pearson}_{\text{wt}}$	CCC_{wt}	Frac. dir.	Cos. rank	$\text{GSEA}_{\uparrow}$	$\text{GSEA}_{\downarrow}$
UNSEENCOMBO ($n = 39$, 2f)							
Zero	10.4635	0.000	0.000	0.000	0.500	0.306	0.206
Tech-dup	0.0045	1.000	1.000	0.979	0.000	0.951	0.923
Interp-dup	0.0059	1.000	1.000	0.979	0.000	0.966	0.954
Additive	0.0751	0.996	0.993	0.957	0.000	0.899	0.873
Matching Mean	2.7901	0.996	0.727	0.963	0.000	0.893	0.872
Target Scaling	9.7519	0.601	0.045	0.792	0.000	0.778	0.654
Global Epistasis	0.3703	0.996	0.968	0.904	0.000	0.924	0.860
GEARS	4.6582	0.957	0.558	0.953	0.000	0.906	0.689
scGPT	0.0115	0.312	0.169	0.727	0.366	0.468	0.539
PRESAGE	0.4949	0.987	0.958	0.990	0.000	0.877	0.932
Ridge	9.7584	0.620	0.042	0.792	0.500	0.778	0.636
Bilinear Ridge	7.3811	0.980	0.244	0.833	0.000	0.885	0.738
Linear Add.	9.7584	0.620	0.042	0.792	0.500	0.778	0.636
Correlation	9.6266	0.341	0.018	0.750	0.197	0.220	0.487
Latent Add.	10.2452	0.623	0.014	0.792	0.500	0.771	0.629

Table 11. Replogle22

	MSE_{wt}	$\text{Pearson}_{\text{wt}}$	CCC_{wt}	Frac. dir.	Cos. rank	$\text{GSEA}_{\uparrow}$	$\text{GSEA}_{\downarrow}$
UNSEENPERT ($n = 1467$, 5f)							
Zero	0.0315	0.000	0.000	0.000	0.500	0.165	0.164
Tech-dup	0.0114	0.719	0.695	1.000	0.010	0.942	0.950
Interp-dup	0.0169	0.632	0.450	1.000	0.005	0.987	0.981
MoP	0.0330	0.149	0.039	0.738	0.500	0.349	0.542
Target Scaling	0.0287	0.358	0.154	0.832	0.008	0.349	0.562
GEARS	0.0337	0.060	0.033	0.621	0.484	0.184	0.479
scGPT	0.0639	0.025	0.023	0.600	0.488	0.248	0.330
PRESAGE	0.0546	0.102	0.095	0.818	0.200	0.377	0.598
Ridge	0.0330	0.149	0.039	0.738	0.500	0.350	0.542
Bilinear Ridge	0.0300	0.097	0.043	0.711	0.078	0.385	0.450
Linear Add.	0.0330	0.149	0.039	0.738	0.500	0.349	0.542
Correlation	0.0364	0.124	0.036	0.721	0.458	0.274	0.175
Latent Add.	0.0330	0.143	0.036	0.734	0.497	0.343	0.535
UNSEENBOTH ($n = 1066$, 5f)							
Zero	0.0599	0.000	0.000	0.000	0.500	0.210	0.159
Tech-dup	0.0146	0.781	0.761	1.000	0.010	0.959	0.936
Interp-dup	0.0217	0.548	0.304	1.000	0.071	0.992	0.986
Grand Mean	0.0577	0.242	0.036	0.607	0.500	0.195	0.489
Target Scaling	0.0485	0.321	0.058	0.679	0.144	0.195	0.500
GEARS	0.0544	0.180	0.071	0.667	0.441	0.148	0.584
scGPT	0.0973	0.037	0.030	0.558	0.494	0.270	0.398
PRESAGE	0.1869	0.016	0.014	0.645	0.294	0.281	0.458
Ridge	0.0577	0.242	0.036	0.607	0.500	0.195	0.489
Bilinear Ridge	0.0577	0.242	0.036	0.607	0.500	0.195	0.489
Linear Add.	0.0577	0.242	0.036	0.607	0.500	0.195	0.489
Correlation	0.0599	0.107	0.016	0.636	0.455	0.325	0.148
Latent Add.	0.0579	0.242	0.033	0.602	0.490	0.193	0.485

Table 12. Sunshine23

	MSE_{wt}	$\text{Pearson}_{\text{wt}}$	CCC_{wt}	Frac. dir.	Cos. rank	$\text{GSEA}_{\uparrow}$	$\text{GSEA}_{\downarrow}$
UNSEENCOMBO ($n = 13$, 2f)							
Zero	0.0564	0.000	0.000	0.000	0.500	0.204	0.130
Tech-dup	0.0234	0.679	0.660	1.000	0.096	0.996	0.992
Interp-dup	0.0309	0.475	0.265	1.000	0.221	0.999	0.999
Additive	0.0299	0.636	0.551	1.000	0.110	0.995	0.977
Matching Mean	0.0290	0.624	0.384	1.000	0.096	0.995	0.974
Target Scaling	0.0537	0.292	0.057	1.000	0.127	0.748	0.434
Global Epistasis	0.0344	0.517	0.294	1.000	0.160	0.990	0.930
GEARS	0.0585	0.015	0.023	0.166	0.615	0.721	0.152
scGPT	0.0495	0.223	0.074	0.917	0.150	0.725	0.186
PRESAGE	0.0326	0.496	0.291	1.000	0.050	0.995	0.992
Ridge	0.0537	0.272	0.022	0.763	0.500	0.748	0.424
Bilinear Ridge	0.0485	0.653	0.105	0.962	0.000	0.990	0.832
Linear Add.	0.0537	0.272	0.022	0.763	0.500	0.748	0.424
Correlation	0.0639	0.100	0.003	1.000	0.425	0.447	0.744
Latent Add.	0.0555	0.276	0.007	0.769	0.517	0.736	0.412

Table 13. Wessels23

	MSE_{wt}	$\text{Pearson}_{\text{wt}}$	CCC_{wt}	Frac. dir.	Cos. rank	$\text{GSEA}_{\uparrow}$	$\text{GSEA}_{\downarrow}$
UNSEENCOMBO ($n = 152$, 2f)							
Zero	0.2153	0.000	0.000	0.000	0.500	0.133	0.138
Tech-dup	0.0082	0.983	0.981	1.000	0.000	0.976	0.899
Interp-dup	0.0144	0.972	0.967	1.000	0.000	0.994	0.948
Additive	0.1463	0.373	0.297	0.886	0.032	0.802	0.815
Matching Mean	0.1719	0.342	0.169	0.875	0.051	0.798	0.808
Target Scaling	0.2003	0.222	0.067	0.736	0.058	0.655	0.254
Global Epistasis	0.1798	0.282	0.118	0.699	0.013	0.716	0.410
GEARS	0.1564	0.313	0.158	0.883	0.135	0.733	0.594
scGPT	0.0515	0.426	0.202	0.650	0.400	0.831	0.167
PRESAGE	0.1511	0.381	0.201	0.902	0.000	0.805	0.795
Ridge	0.2003	0.222	0.067	0.736	0.500	0.655	0.254
Bilinear Ridge	0.1955	0.251	0.079	0.774	0.000	0.689	0.293
Linear Add.	0.2003	0.222	0.067	0.736	0.500	0.655	0.254
Correlation	0.2149	0.141	0.036	0.706	0.391	0.420	0.368
Latent Add.	0.2026	0.216	0.025	0.643	0.538	0.663	0.240

H. Signal-based perturbation filtering

Aggregate scores are only meaningful over perturbations whose ground-truth response can be trusted. We select, once and independently of any metric, a fixed subset of test conditions whose measured response is reliably estimated, and report every metric on that same subset. This appendix gives the statistics, their calibration, and the effect on the evaluated set.

H.1. Reliability, not effect size, is the right axis

We score a prediction against a measured pseudobulk response $y_{c,p} = \mu_{c,p} - \mu_{c,\varnothing}$, which is an estimate $y_{c,p} = \delta_{c,p} + \epsilon_{c,p}$ of the true effect $\delta_{c,p}$ corrupted by finite-sample noise $\epsilon_{c,p}$. For many perturbations $\epsilon_{c,p}$ dominates, so the "ground truth" is unreliable and scoring against it is uninformative. Measurement noise does not merely add error; it preferentially erases structure—a degree-k interaction is attenuated by the k-th power of the measurement reliability while a linear term is attenuated only once (Schulz & Ritter, 2026)—so at low reliability the recoverable signal vanishes whatever the underlying biology, and a flexible model has nothing left to exploit beyond a simple baseline. These are exactly the conditions on which the positive control's split-half prediction degrades to noise and the achievable dynamic range collapses (Mejia et al., 2025): a perturbation can look baseline-resistant not because it hides recoverable non-linear structure but because its measured response is unreliable. Separating these two is the purpose of the gate. Three cases must not be conflated:

- **precise null** — true effect ≈ 0, measured precisely (both replicate halves agree it is ≈ 0): a model *should* predict ≈ 0, so this is a legitimate, if easy, target;
- **unreliable** — effect unknown because the estimate is too noisy or heterogeneous to trust: an illegitimate target, regardless of its apparent magnitude;
- **precise effect** — large and reliably measured: the target we most want to keep.

Effect-size filters (a fixed signal-to-noise cut, the energy distance, or its permutation E-test) threshold on *magnitude* and therefore delete the precise null together with the noise, while telling us nothing about whether a large effect is reproducible. The reliability of a measurement is a property of its *precision*, not its size. We thus build the gate from two effect-size–decoupled statistics, each yielding one scalar per condition so that the retained subset is identical across all metrics.

H.2. Detectability: the noise-scaled response (NSR)

For a condition with n_p perturbed and $n_\varnothing$ control cells, the pseudobulk difference $y_{c,p} = \bar{x}^{\,p} - \bar{x}^{\,\varnothing}$ is an unbiased estimate of $\delta_{c,p}$ because the two cell pools are independent. Under the isotropic squared-Euclidean geometry used throughout, its expected squared error collapses to two scalars — only variance *traces* enter, never a covariance matrix:

$$\mathbb{E}\left\|y_{c,p} - \delta_{c,p}\right\|^2 \;=\; \frac{V_p}{n_p} + \frac{V_\varnothing}{n_\varnothing} \;=:\; \text{se}^2_{c,p}, \qquad V \;=\; \text{tr}\,\Sigma \;=\; \sum_g \text{Var}_g, \tag{9}$$

where V_p and $V_\varnothing$ are the total per-cell variances of the perturbed and control populations, estimated with the unbiased plug-in $\hat{V} = \frac{1}{n-1}\sum_i \|x_i - \bar{x}\|^2$ (one scalar per cell, computed in a single streaming pass over the count matrix). The detectability statistic is the response size relative to its standard error,

$$\mathrm{NSR}_{c,p} \;=\; \frac{\|y_{c,p}\|}{\mathrm{se}_{c,p}}. \tag{10}$$

Because $\|y_{c,p}\|$ is positively biased ($\mathbb{E}\|y_{c,p}\|^2 = \|\delta_{c,p}\|^2 + \mathrm{se}^2_{c,p}$), we also record the debiased size $\sqrt{\max(0,\, \|y_{c,p}\|^2 - \mathrm{se}^2_{c,p})} \approx \|\delta_{c,p}\|$. Equation (10) measures the same per-perturbation signal-to-noise that a per-metric positive-control normalization approximates — here measured directly from the two populations' cell-level variances, metric-free, and used to decide *inclusion* rather than to *score*. A relabeled control (a "fake" perturbation) has $\mathrm{NSR} \approx 1$; a strong effect has $\mathrm{NSR} \gg 1$. The cost of using detectability alone is that it also discards genuinely *weak* effects, which motivates the second axis.

H.3. Reproducibility: split-half direction agreement (ρ)

We split each condition's cells into two technical halves and form independent response estimates $y^{(1)}_{c,p}, y^{(2)}_{c,p}$. The reproducibility coefficient is their across-gene correlation,

$$\rho_{c,p} \;=\; \mathrm{corr}_g\big(y^{(1)}_{c,p},\, y^{(2)}_{c,p}\big), \tag{11}$$

which measures whether the *direction pattern* of the response reproduces, independently of its magnitude. A fake perturbation has no reproducible pattern ($\rho \approx 0$), whereas a small but consistently directed real effect has $\rho > 0$; ρ therefore retains weak-but-reproducible conditions that (10) drops. We deliberately use the *direction* correlation rather than the split-half magnitude disagreement $D = \|y^{(1)}_{c,p} - y^{(2)}_{c,p}\|$: a fake is a *reproducibly null* measurement (both halves ≈ 0, hence small D), so a "$D \leq$ floor" rule would *keep* fakes. A literal zero-effect perturbation has no direction to reproduce and is statistically indistinguishable from a fake; any reproducibility gate therefore drops it, which is harmless because the negative-control (predict-zero) baseline already scores it trivially. Genuine non-targeting guides are removed by annotation upstream, not by this gate.

H.4. Null calibration

Both thresholds are calibrated per dataset and per context against a control-split null, so that the cutoff adapts to each dataset's depth and noise. For each context we repeatedly draw a size-matched group of control cells (matched to the perturbation cell-count distribution; 300 draws), relabel it as a pseudo-perturbation against the remaining control cells, and recompute NSR and ρ exactly as above. These draws carry pure technical noise with zero perturbation effect, so their 95th percentile is the cutoff at a 5% false-positive rate. The NSR null concentrates near 1 (cutoffs ≈ 1.04–1.14 across datasets) and the ρ null near 0 (cutoffs ≈ 0.05–0.28).

H.5. The combined rule

A condition is retained iff it clears *either* cutoff,

$$\text{keep } (c,p) \quad\Longleftrightarrow\quad \mathrm{NSR}_{c,p} > q^{\mathrm{NSR}}_{0.95} \;\text{ OR }\; \rho_{c,p} > q^{\rho}_{0.95}, \tag{12}$$

which drops only the joint low-detectability, low-reproducibility corner — the noise/fake region. The two axes partition conditions into four quadrants (Figure 5): *both* (detectable and reproducible), *NSR-only* (large but pattern-noisy), *ρ-only* (weak but reproducible — rescued by (12)), and *neither* (dropped). A fake lands at $(\mathrm{NSR} \approx 1, \rho \approx 0)$ and is removed.

H.6. Effect on the evaluated set

Table 14 reports retention across the ten datasets. The retained fraction tracks signal strength: high-signal datasets keep essentially all conditions, while the noise-floor datasets (McFaline23, X-Atlas/Orion, Sunshine23) shed most of theirs — exactly the perturbations whose measured response is within its own sampling noise. The reproducibility axis rescues 590 conditions in total (e.g. 478 in X-Atlas/Orion, 66 in McFaline23) that detectability alone would drop, confirming that the second axis is not redundant. Figure 6 shows that the dropped conditions cluster in the low-cell-count, small-effect corner, as intended.

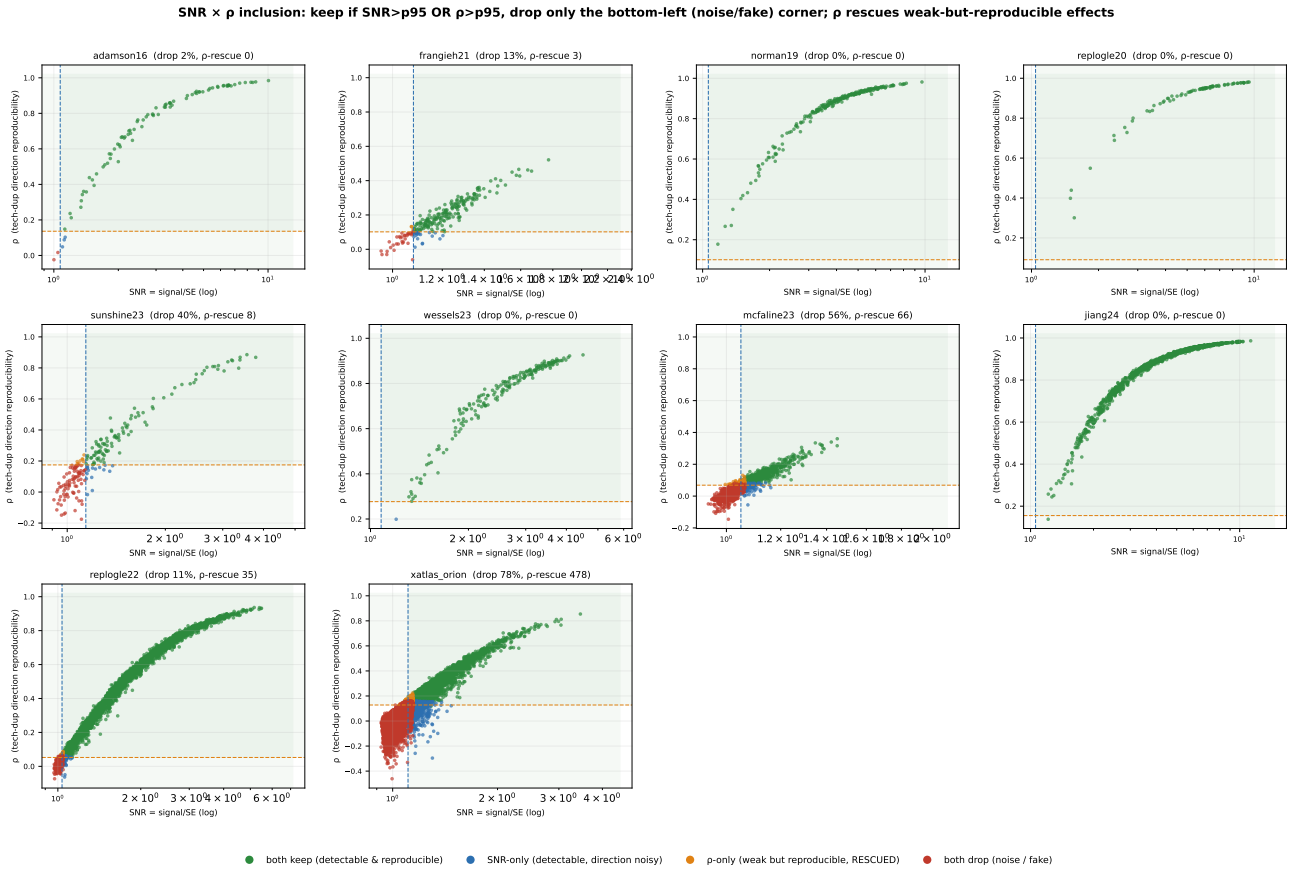

Figure 5. Inclusion geometry per dataset. Each point is a (context, perturbation) condition placed by detectability ($\mathrm{NSR} = \|y\|/\mathrm{se}$, x-axis, log) and split-half direction reproducibility (ρ, y-axis); dashed lines are the per-context control-split 95th-percentile cutoffs. The rule keeps everything except the bottom-left (noise/fake) corner; the orange ρ-only points are weak-but-reproducible conditions rescued by the reproducibility axis that a detectability-only filter would discard.

H.7. Robustness and agreement with alternative inclusion rules

The signal filter is a single, metric-*independent* inclusion set, computed once on the response vector, that fixes *which* conditions every metric is scored on, guaranteeing a constant, comparable evaluation set across all metrics. The selection in (12) agrees closely with the calibrated energy-distance E-test and with a detectability-only cut on the conditions they all decide (Figure 7); we adopt the $\mathrm{NSR} \vee \rho$ rule because, unlike the magnitude-based alternatives, it retains weak-but-reproducible effects and treats a precisely measured small response as a valid target.

Table 14. Per-dataset retention under the $\mathrm{NSR} > q_{0.95}$ OR $\rho > q_{0.95}$ rule. n is the number of evaluated (context, perturbation) units; "ρ-rescued" counts conditions kept by reproducibility despite failing detectability. Datasets are ordered by retained fraction.

Dataset	Ctx.	n	Kept	ρ-rescued	Keep %
Norman19	1	236	236	0	100.0
Replogle20	1	77	77	0	100.0
Jiang24	6	1,296	1,296	0	100.0
Wessels23	1	186	186	0	100.0
Adamson16	1	89	87	0	97.8
Replogle22	2	3,982	3,559	35	89.4
Frangieh21	1	240	208	3	86.7
Sunshine23	1	212	128	8	60.4
McFaline23	3	1,533	670	66	43.7
X-Atlas/Orion	2	35,916	8,035	478	22.4

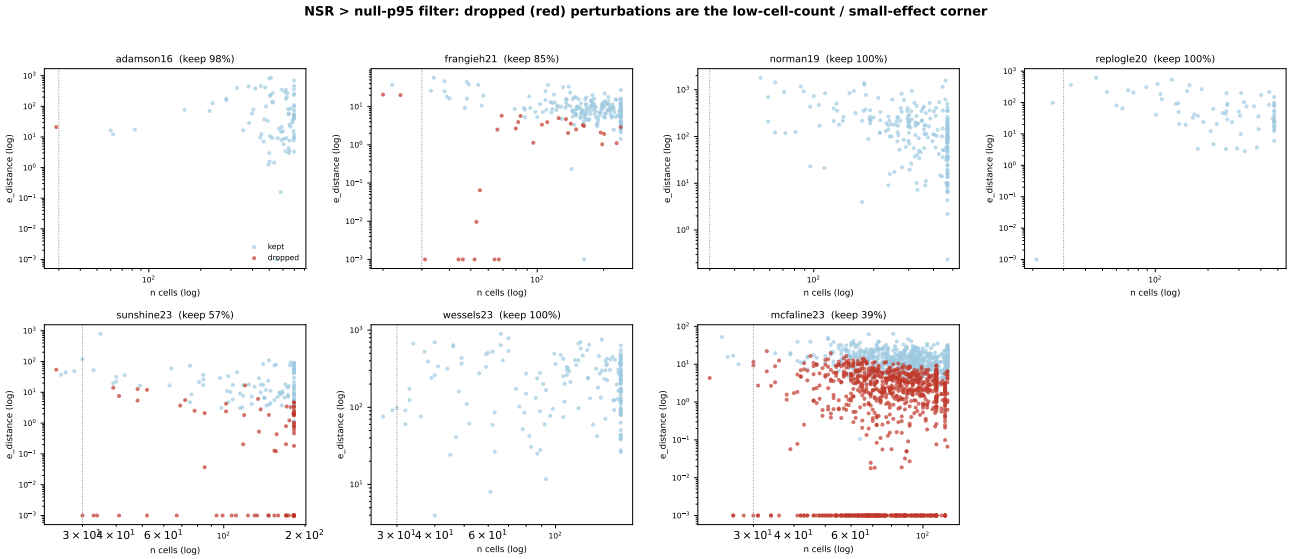

Figure 6. Dropped (red) versus retained (blue) conditions per dataset, by cell count (x, log) and effect size (y, log). Dropped conditions concentrate in the low-cell-count, small-effect corner (the dotted line marks 30 cells), confirming the filter removes under-powered and noise-floor perturbations rather than genuine strong effects.

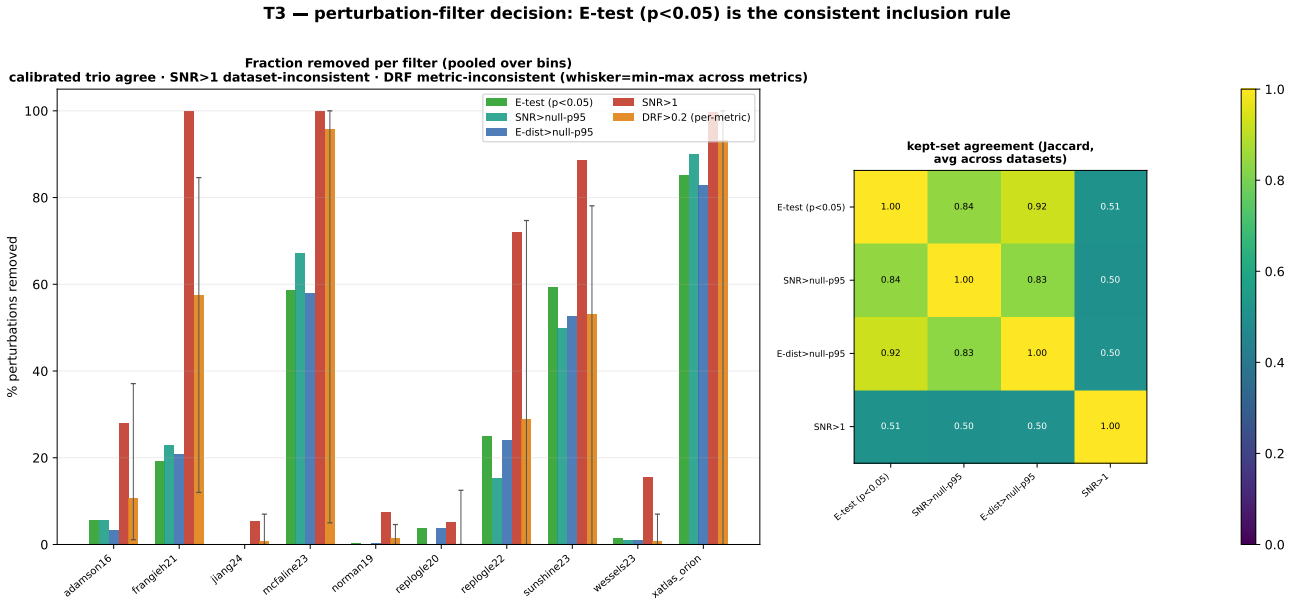

Figure 7. Agreement of keep/drop decisions across candidate inclusion rules (energy-distance E-test, detectability $\mathrm{NSR} > q_{0.95}$, energy distance $> q_{0.95}$, a fixed $\mathrm{NSR} > 1$, and the combined $\mathrm{NSR} \vee \rho$ rule). The calibrated rules largely concur; the combined rule differs by additionally retaining weak-but-reproducible conditions.