A Systematic Comparison of Single-Cell Perturbation Response Prediction Models

Lanxiang Li[1,2†], Yue You[1*†], Wenyu Liao[3†], Xueying Fan[4,5,6,7], Shihong Lu[1], Ye Cao[1], Bo Li[1], Wenle Ren[1], Yunlin Fu[1], Jiaming Kong[8], Shuangjia Zheng[9], Jizheng Chen[1,10], Xiaodong Liu[4,5,6,7], Luyi Tian[1,2*]

[1*]Guangzhou National Laboratory, Guangzhou, China.
[2*]GMU-GIBH Joint School of Life Sciences, Guangzhou Medical University, Guangzhou, China.
[3]The University of Hong Kong, Hong Kong, China.
[4]School of Life Sciences, Westlake University, Hangzhou, China.
[5]Research Center for Industries of the Future, Westlake University, Hangzhou, China.
[6]Westlake Laboratory of Life Sciences and Biomedicine, Hangzhou, China.
[7]Westlake Institute for Advanced Study, Hangzhou, China.
[8]Renaissance Era Innovation Technology Co., Ltd, China.
[9]Global Institute of Future Technology, Shanghai Jiaotong University, Shanghai, China.
[10]State Key Laboratory of Respiratory Disease, National Clinical Research Center for Respiratory Disease, Guangzhou Institute of Respiratory Health, the First Affiliated Hospital of Guangzhou Medical University, Guangzhou, Guangdong, China.

*Corresponding author(s). E-mail(s): you_yue@gzlab.ac.cn; tian_luyi@gzlab.ac.cn;
†These authors contributed equally to this work.

Abstract

Predicting single-cell transcriptomes following perturbation is crucial for understanding gene regulation and guiding drug discovery. Yet, the complexity of perturbation effects pose significant challenges for robust predictive modeling. Although recent efforts have introduced foundation models alongside traditional statistical and machine learning approaches, a comprehensive benchmarking study of their predictive performance has been lacking. Here we establish a standardized evaluation framework and systematically assess 9 single-cell perturbation prediction models on 17 diverse datasets, spanning multiple cell types and perturbation modalities. Leveraging a multifaceted suite of 24 evaluation metrics, we find that models often excel in capturing global expression profiles yet struggle to predict the nuanced effects of perturbed genes, or vice versa. Moreover, while foundation models frequently outperform simpler methods, they tend to converge on population averages, struggling to capture heterogeneous cellular responses to perturbations. Taken together, our study highlights both the potential and limitations of single-cell foundation models, identifies opportunities for future improvement, and provide a roadmap for advancing single-cell predictive biology.

Introduction

Single-cell RNA sequencing (scRNA-seq) offers unprecedented resolution for exploring cellular heterogeneity [1, 2]. Multiplexed perturbations together with scRNAseq, such as Perturb-seq and CROP-seq, help uncover how cells respond to diverse stimuli or interventions [3–5]. These approaches advance our understanding of cellular functions, and regulatory networks, and provide crucial insights for drug development and disease treatment [6]. However, current experimental systems are largely optimized for cell lines, with limited applicability to primary or in vivo systems. Furthermore, combinatorial multiplexing experiments remain prohibitively expensive and technically challenging to scale [7, 8].

In this context, in silico approaches that predict post-perturbation cell states across diverse environments and conditions present a promising, scalable alternative, and provide a path to building a virtual cell [9, 10]. Early approaches focused on predicting cellular responses in cell types absent from experimental training data or responses to combinations of previously tested perturbations [11–13]. However, these methods could not generalize to unseen perturbations. Recent advances have incorporated graph-based methods and foundational models that exploit prior knowledge of gene expression relationships, achieving significant improvements in generalizability [14–17].

Among them, single-cell foundation models represent a pivotal breakthrough. By leveraging vast datasets and pre-trained transformer architectures[18], these models learn robust, generalizable relationships between genes and have demonstrated potential across diverse downstream tasks, including transcriptome prediction following perturbations after models fine-tuned on specific datasets [19]. Despite their promise, the field lacks a comprehensive framework to evaluate their capabilities systematically. Compared to tasks such as cell annotation and batch correction, perturbation prediction offers experimentally generated model input/output with well-defined objectives, making it a more robust benchmark for assessing foundation models [15, 20, 21]. Recently, a few studies evaluated single-cell foundation models on perturbation prediction and claimed that foundation models do not outperform simple linear baseline models [22, 23]. However, these findings are limited by the narrow scope of datasets and evaluation frameworks used, raising questions about the robustness and generalizability of their conclusions. A comprehensive benchmarking study is required to evaluate the pros and cons of single-cell foundation models fully.

In light of the gaps listed above, we have systematically benchmarked single-cell perturbation prediction models to date, focusing on foundation models. We curated a diverse benchmarking dataset, consisting of 17 public and in-house datasets that span a wide range of cell types and perturbation conditions. We developed a robust set of evaluation metrics to compare the performance of 9 different models, including foundation models, autoencoders, graph-based architectures, and other deep learning models. While simple linear models can sometimes outperform sophisticated foundation models in specific scenarios, foundation models generally demonstrate superior performance and better generalization ability. Moreover, we reveal that while foundation models outperform earlier methods on many metrics, they still face significant challenges in capturing heterogeneous cellular responses to perturbations.

Results

Benchmarking models and experimental design

In this study, we comprehensively benchmarked 9 single-cell perturbation response prediction models. To evaluate these models' performance, we curated a diverse collection of 16 public datasets (Table 1), encompassing cell lines (e.g., K562, A549), induced pluripotent stem cells (iPSCs), and primary cells. To better evaluate the generalization ability of different models on dissimilar cell types, we also generated a new dataset with K562 and skin fibroblasts treated with the same panel of anti-aging drugs. These datasets span a wide spectrum of perturbation types, including drug treatments, genetic knockouts, and immune response stimuli, allowing us to assess model generalization across varying biological contexts and perturbation types (Fig. 1a, left).

We selected eight representative models, encompassing a variety of computational architectures designed for single-cell perturbation modeling (Fig. 1a, middle). Additionally, we designed a set of linear baseline models based on multilayer perceptron (MLP) to evaluate the performance gain between a sophisticated foundation model and a simple baseline. Due to implementation challenges, certain methods, such as CellOT [24], were not included.

The selected models include:

1. **Gene regulatory network (GRN)-based models** (e.g., GEARS[14], scELMo[25]), which uses regulatory information to predict gene expression changes.
2. **Foundation models** (e.g., scGPT[15], scFoundation[21]), pre-trained on large datasets, enabling generalization to new contexts with minimal retraining.
3. **Variational autoencoders** (VAE) (e.g., scGen[26], trVAE[27]), which capture latent gene expression structures for prediction of perturbation response.
4. **Other deep learning-based methods** (e.g. CPA[28], scPreGAN[29]), which use neural networks to model complex gene expression patterns.

Detailed descriptions of the datasets and models are provided in Table 1 and Supplementary Tables 1 and 2.

Next, we implemented a standardized, multi-dimensional evaluation pipeline encompassing diverse metrics, data subsets, and ranking strategies (Fig. 1a, right). We included four primary metric categories—error, goodness-of-fit, correlation, and distributional similarity while expanding the scope of the evaluation to better capture differences in predictive capacity. Specifically, in addition to evaluating models for the entire set of perturbations, we focused on the top 20 differentially expressed (DE) genes per perturbation, identified by comparing perturbed and control states. Within the four metric categories, error metrics (such as MSE, RMSE, MAE, and L2 loss) quantify how closely predicted values match observed outcomes, reflecting the absolute magnitude of discrepancies. Goodness-of-fit metrics (e.g., R^2) measure the proportion of variance in the data explained by the model. Correlation metrics (e.g. Pearson's correlation, cosine similarity), including a perturbation-specific Pearson correlation measure that accounts for differences relative to the control state, emphasize the

preservation of relative relationships between predicted and true values. Finally, distributional similarity metrics (e.g., Wasserstein distance, MMD) evaluate how closely the overall predicted distribution for each gene matches the observed distribution.

In addition to direct metric comparisons, we employed a ranking-based analysis to further dissect the relative performance of selected models. For each perturbation and evaluation metric, methods were ranked from best to worst. By aggregating these rankings across multiple perturbations, we could quantify, for each pair of models, the percentage of perturbations in which one outperformed the other.

In terms of the models selected, we categorized them into three tasks focusing on their ability in different scenarios, each representing a different aspect of generalization: predicting unseen perturbations, combinatorial perturbation interactions, and responses across unseen cell types. These tasks reflect the varying complexity of perturbation prediction and the diverse experimental contexts in which these models are applied. The evaluation process is shown in Fig. 1b, where the input and output for the training (light blue) and testing (light yellow) phases are outlined. Specifically, we focus on the following.

1. **Unseen Perturbation Prediction** (Fig. 1b, left): In this task, the models were trained to predict the effects of novel, unseen single-gene perturbations within the same cell type or cell line. Using training data that included both unperturbed and known single-gene perturbation responses, the models were tasked with inferring gene expression changes under previously unseen perturbations.
2. **Combinatorial Perturbation Prediction** (Fig. 1b, middle): Building on Task 1, this task introduced additional complexity by requiring the models to predict the interaction effects of combinatorial perturbations. Based on unperturbed and single-gene perturbation data, the models were expected to predict how new combinations of gene perturbations would influence gene expression in the same cell type or cell line.
3. **Cell state transfer** (Fig. 1b, right): This task assessed the models' ability to generalize across different cell types. The models were trained on unperturbed and perturbed data from various cell types to learn gene expression patterns under perturbation. They were then tested on previously unseen cell types, predicting their responses to similar perturbations or drug treatments.

In summary, we systematically evaluated 9 single-cell perturbation prediction methods across three core tasks, utilizing 17 datasets and 9 distinct evaluation metrics to provide a comprehensive comparison of model performance.

Prediction of unseen single-gene perturbations

For predicting responses to unseen single-gene perturbations (Fig. 2a), we evaluated GEARS, scELMo, scFoundation, scGPT, and a baseline model across 10 public datasets using the same four categories of evaluation metrics—error, goodness-of-fit, correlation, and distributional similarity—as in subsequent tasks. Both all genes and the top 20 differentially expressed (DE) genes per perturbation were examined.

In the all-gene scenario, complex models generally achieved lower RMSE than the baseline, although no single model dominated every dataset (Fig. 2b, Fig. S2-S3). However, when assessing population-level distribution alignment via MMD and Wasserstein distance, both the baseline and foundation models struggled, while GEARS and scELMo demonstrated a superior ability to capture heterogeneous cellular responses (Fig. 2c, Fig. S3f-h). Notably, scGPT converges to the population average and has worse distribution alignment as the training epoch increases (Fig. 2d). Pearson correlation (PCC) delta analyses further showed that scFoundation aligned predicted and observed changes more closely than the baseline (Fig. 2e, Fig. S3c).

Focusing on the top 20 DE genes provided additional insight. Here, all complex models outperformed the baseline in MSE, and scFoundation emerged as the top performer (Fig. 2f,g, Fig. S4-8), underscoring its strength in predicting key perturbation signals. Nevertheless, the complexity of biological heterogeneity and perturbation-specific effects remains evident. While conventional metrics like MSE and PCC often reflect dataset composition and gene selection, distribution-based metrics (e.g., MMD) highlight deeper, intrinsic differences in model capabilities. We tried to examine the model's ability to generalize across the different dataset, and found all the model consistenly have poor performance (Fig S9a-c).

Part of the complicated result stems from the data heterogeneity. The ANOVA analysis showed that most of the variation in error and correlation metrics could be explained by perturbation itself (Fig. S9d), and E-distance analysis also showed that different datasets have different perturbation effect sizes (Fig. S9e). Overall, these findings illustrate that while advanced models outperform simple baselines and better capture targeted perturbation signals, challenges in generalization and distributional fidelity persist.

Perturbation interaction prediction

Building upon our previous analysis of single-gene perturbation prediction, we have further assessed the predictive capacity for combinatorial gene perturbations to validate model performance under more complex conditions (Fig. 3a). We utilized three public datasets that contain combinatorial gene perturbations. Similar to our single-gene analysis, we compared the performance of six models (CPA, GEARS, scELMo, scFoundation, scGPT, and Baseline) based on two scenarios: all genes and top 20 DE genes, using four major categories of evaluation metrics. We designed three experimental conditions based on the exposure of gene combinations in the training data: (i) "seen0", where neither gene in the combination was seen during training; (ii) "seen1", where one gene in the combination was present in the training set; and (iii) "seen2", where both genes in the combination were individually observed during training.

Firstly, we evaluated the "seen2" case where both of the perturbations have been seen during the training process (Fig. 3b-e, Fig. S10-15). For the all-gene scenario, the error metrics (Fig. 3b, Fig. S10a-d) revealed that CPA consistently performed poorly across all datasets and conditions, with a higher error than the other models. In contrast, for the remaining models, performance was highly dataset-dependent. The baseline model showed mixed performance, occasionally outperforming more sophisticated methods but failing in certain cases. Other error metrics, goodness-of-fit metrics,

and correlation metrics exhibited similar trends, further reinforcing the observation that CPA consistently underperformed (Fig. S10,11). Interestingly, CPA outperformed other models on the Pearson delta metric (Fig. S11c.d). For distributional similarity metrics, the results were consistent with our single-gene analysis, where scFoundation, scGPT, Baseline, and CPA all failed to outperform GEARS and scELMo (Fig. 3c, Fig. S11g-i). This aligns with the earlier observation that these metrics are more influenced by the model's structural design than the dataset composition.

When focusing on the top 20 DE genes, the results for most metrics were consistent with the trends observed in the all-gene scenario (Fig. S12-14). scFoundation exhibited the best MSE rank across datasets, followed by the baseline model, suggesting its robustness in handling combinatorial perturbations (Fig. 3d). When evaluating PCC on the delta of the top20DE (Fig. 3e, Fig. S13e-g), CPA emerged as the top-performing model, surpassing all others in accurately capturing the relative expression changes. This indicates that CPA excels at capturing the direction and scale of changes, even though its absolute error remains high. In contrast, scGPT and scFoundation struggled in this metric, suggesting that while these models capture global patterns well, their resolution in capturing localized differential responses may be limited.

To further evaluate the generalization ability of these models, we compared their performance under "seen0," "seen1," and "seen2" conditions. For cosine similarity across all genes (Fig. 3f), the seen0 condition consistently resulted in the lowest performance for all models, highlighting the significant challenges models face when generalizing to entirely unseen combinations. Similarly, for the top 20 DE genes, the Wasserstein distance metric showed a comparable trend(Fig. 3g), with seen0 consistently yielding the poorest performance across all models. Additionally, results from other metrics also revealed a slight decline in predictive accuracy as the number of observed genes during training decreased (Fig. S15-16). This may be attributed to the models' insufficient learning of perturbation patterns, particularly under unseen conditions, which significantly limits their ability to capture population heterogeneity and perturbation-specific signals. These findings underscore the importance of incorporating a more diverse range of gene perturbation combinations in training data to enhance model generalization.

Cell state transfer

Next, we investigated the ability of models to predict cell state transitions following perturbations across different cell types. Unlike perturbation prediction for individual genes or gene combinations, cell state transition involves predicting how a different type of cell responds to perturbations (Fig. 4a). To assess model performance in this context, we utilized seven datasets encompassing a broad range of conditions and cell types and evaluated the performance of scPreGAN, scGEN, trVAE and Baseline model based on four different evaluation metrics. Due to the limited availability gene perturbation dataset, we did not include foundation models in the comparison as the current model cannot encode molecular perturbations.

Under the RMSE metric, scGEN and trVAE exhibit similar performance across most datasets, with the Baseline model worse than others on all the datasets (Fig.

4b). These differences may be attributed to the distinct effect size and cell type similarity of the datasets. In terms of other error metrics, goodness-of-fit, and correlation metrics, scGEN and trVAE outperform the other models (Fig. 4c,d, Fig. S17a-i, Fig. S18a-f). This suggests that both models effectively leverage the VAE architecture to model the latent relationships between cell states. For the distribution distance metrics, all the models have worse distribution alignment compared to the original input as the NoPetrub group and baseline model have worse performance (Fig. 4e, Fig. S18g-i). Moreover, we observe that model performance decreases when the predicted cell types are dissimilar to each other (Fig. 4f,g, Fig. S19a-k). Notably, the Baseline model performance gets worse as the cell types are more dissimilar to each other, indicating its weaker generalization capability across dissimilar cell types. In contrast, models like trVAE, by learning the latent structure of cell states, can maintain better predictive ability across different cell types. Additionally, through ANOVA analysis, we identified that the models and cell type were the main factors influencing prediction performance (Fig. S19l).

Overall, trVAE and scGEN demonstrate a clear advantage in handling complex cellular response patterns, particularly in capturing nonlinear transitions of cell states, showcasing strong performance.

Discussion

This study provides a comprehensive evaluation of the performance of single-cell perturbation prediction models, offering an in-depth exploration of their current capabilities and limitations. We designed a novel prediction framework to break down the prediction objectives into 3 tasks: 1) Predicting responses to unseen perturbations; 2) Predicting responses to combinatorial perturbations; 3) Predicting responses to unseen cell types. Our multifaceted evaluation approach, which includes error, goodness-of-fit, correlation, and distributional similarity metrics as well as perturbation-sensitive gene subsets and ranking-based comparisons, enables a more nuanced understanding of model robustness and generalizability.

Our results highlight that performance is highly context-dependent and that no single method excels across all scenarios (Fig. 5). For instance when predicting unseen single-gene perturbations, complex models such as scFoundation surpass simple baselines in capturing essential features—especially when focusing on top differentially expressed genes—yet their ability to align predicted distributions with ground truth varies, as evidenced by MMD-based evaluations. In predicting combinatorial perturbations, prior exposure to related perturbations (seen1, seen2 conditions) can significantly improve model generalization. Foundation models, while expected to generalize better, do not always outperform models like GEARS and scELMo when both perturbations are entirely novel. However, scFoundation's strengths in seen1 and seen2 indicate it can leverage partial prior knowledge effectively. The baseline model, although initially limited, improves as it encounters more perturbations, suggesting that simpler architectures may benefit from additive effects. Models like trVAE and scGEN demonstrate strong performance in transferring perturbation outcomes across cell types, indicating that latent variable models could capture complex cellular transitions.

Despite the context-dependent result on different datasets, foundational models and baseline models consistently face challenges in precisely capturing the heterogeneous response of cells to perturbations, which could be an intrinsic property of cells [30]. Together with a notable observation that some foundational models (such as scGPT) are more sensitive to the increase of training epochs, this leads to excessive compression of the perturbation response distribution. This limitation might stem from current model architectures, which predict gene expression for one cell at a time without capturing the complex distributional heterogeneity of populations. This architecture biases models toward population averages, explaining discrepancies between error metrics and distribution alignment. These findings underscore the need to optimize training processes and develop architectures that better account for response heterogeneity at the population level.

Despite the advances made, this study has limitations. Most datasets used are publicly available and were generated using varied technologies, introducing differences in data quality and batch effects that may influence comparisons. ANOVA analysis confirmed that performance variability arises from the dataset and perturbations, highlighting the complexity and heterogeneity of perturbation effects. More perturbation dataset is required to disentangle these effects. Additionally, the lack of genetic perturbation datasets with diverse cell types limited our ability to assess the cell type transfer capabilities of foundation models fully.

Beyond this study, broader challenges remain in the field. A fundamental limitation is the lack of real ground truth due to the destructive nature of single-cell sequencing methods, that we cannot measure the transcriptome of the same cell before and after perturbation. Another key limitation is the lack of temporal dynamics in existing datasets, which typically include only pre- and post-perturbation time points. Furthermore, current models predominantly focus on transcriptomics, yet integrating additional modalities—such as genomic mutations, epigenomic modifications, and proteomics—is essential for a holistic understanding of perturbation responses[31]. Incorporating language representation around the central dogma (i.e., DNA, RNA, and protein) and molecular representations of perturbations at the atomic level will aid in deciphering the intricate biomolecular interactions within the cell [32], thereby offering a more generalized solution for single-cell perturbation response prediction. Spatial perturbation technology is still in its infancy but could potentially transform the future model from single-cell to tissue level [33].

In general, our study highlights gaps and provides a roadmap for future developments of single-cell perturbation response prediction models. Optimizing training processes, improving model architectures, and incorporating multi-timepoint and multi-modal datasets will collectively drive the evolution of single-cell predictive models. These advancements will deepen our understanding of cellular responses and unlock transformative applications in precision medicine and biotechnology. We anticipate that the benchmarking framework and insights presented here will play a pivotal role in building a virtual cell[34].

Methods

Ethics statement

Human dermal fibroblasts (HDFs) were collected from two male donors with informed consent. The use of HDFs in this study was approved by the Westlake University Ethics Committee (20240909LXD001).

Cell culture and drug treatment

HDFs were cultured in DMEM medium (C11965500BT, Gibco) supplemented with 10% fetal bovine serum (FBS; C04002, VivaCell), 1% non-essential amino acids (11140-050, Gibco), 1 mM GlutaMAX (35050-061, Gibco), 55 μM 2-mercaptoethanol (21985-023, Gibco), 1 mM sodium pyruvate (11360-070, Gibco), and 1% penicillin-streptomycin (15140-122, Gibco). K562 cells, obtained from ATCC, were cultured in RPMI 1640 medium (61870-036, Gibco) with 10% FBS and 1% penicillin-streptomycin. Both K562 cells and HDFs from two donors were seeded at a density of 1×105 cells per well in either 96-well or 12-well plates. K562 cells were rested for 24 hours and HDFs for 48 hours, before drugs were added to each well to reach the specified final concentrations. Half of the medium was refreshed daily. A detailed list of drugs and concentrations used for K562 cells and HDFs is provided in the Supplementary Table 3. K562 cells were collected for single-cell RNA sequencing after 3 days of treatment, while HDFs were collected after 4 days.

Sample multiplexing for single-cell RNA sequencing

Sample multiplexing was conducted using the 3' CellPlex Kit Set A (10X Genomics, 1000261) according to the manufacturer's instructions. Briefly, cells from each well were collected and labeled with unique cell multiplexing oligos, and equal numbers of live cells from 11 drug-treated groups and one untreated group were pooled into a single sample. For HDFs, cells from two donors treated with the same drug were mixed before incubation with multiplexing oligos. In total, two pooled samples were prepared for K562 cells and two for HDFs, followed by library preparation using the 10X Chromium Next GEM Single Cell 3' Kit v3.1 (10X Genomics, 1000123).

Datasets and preprocessing

According to the research conducted by Peidli et al. [8], we acquired the Datlinger [4, 35], Dixit [3], Frangieh [36], Papalexi [37], Replogle [38], Tian [39], and Srivatsan [40] datasets from Zenodo (access link: https://zenodo.org/records/7041849). For the Datlinger and Frangieh datasets, we curated the perturbation_2 subset, excluding perturbations related to "stimulate" and "IFNγ," as well as "Co-culture," while retaining only the CRISPR-Cas9 perturbations. The Sunshine [41] dataset was obtained from the NCBI Gene Expression Omnibus (GEO) under accession number GSE208240, where we filtered for entries classified as "exact_match," "no_match," or "scrambled_pair," and excluded entries that contained non-targeting

sequences in the "guide_identity." For the Srivatsan dataset, we selected perturbations involving the following agents: Tazemetostat, Decitabine, Resminostat, Panobinostat, Navitoclax, Trametinib, and Dasatinib. Similarly, we downloaded the Burkhardt dataset from https://www.kaggle.com/competitions/open-problems-single-cell-perturbations/data, focusing on assessing the effects of Trametinib, Decitabine, Navitoclax, and Dasatinib on perturbation responses. Details of the datasets utilized are summarized in Table 1 and Supplementary Table 1.

Data normalization and log transformation were performed using the scanpy(v1.10.1) toolkit. For datasets involving gene perturbations, we implemented a sample filtering process to ensure adequate coverage and minimize noise. Specifically, we applied stratified subsampling to groups with large sample sizes, limiting the maximum sample count per group to 3,000. Groups with fewer than 50 samples were excluded to reduce potential errors arising from small sample sizes. To enhance the robustness of feature selection, we calculated and retained the top 5,000 highly variable genes. Within datasets containing gene perturbations, we identified perturbed genes based on the "condition" field and extracted their names, thereby creating a subset of genes directly influenced by the experimental conditions. Ultimately, we combined the identified perturbed genes with the highly variable genes to generate the final gene set utilized for data filtering, ensuring the preservation of the most informative genes associated with perturbations. For datasets not involving gene perturbations, we retained only the top 5,000 highly variable genes for filtering.

Additionally, we downloaded the Norman [42] dataset from the scGPT study (https://dataverse.harvard.edu/api/access/datafile/6154020), along with the Kang [43], Haber [44], and Hagai [45] datasets from the scGen study (https://github.com/theislab/scgen-reproducibility/blob/master/code/DataDownloader.py). The Weinreb [46] dataset was sourced from the CellOT study [24] (https://www.research-collection.ethz.ch/handle/20.500.11850/609681). Given that these datasets were appropriately preprocessed in their original studies, no further data processing steps were applied. All data analysis was performed on in-house Linux server with Ubuntu operating system (version 22.04) and Data Science Platform of Guangzhou National Laboratory [47].

The DE genes are calculated using the rank_genes_groups function in scanpy with method='wilcoxon'. then DE genes are ranked by their adjusted p value and the top 20 DE genes are selected and used to subset the gene expression matrix for calculating the evaluation metrics.

Models selection and setup

In this study, we selected and applied a range of advanced modeling approaches to systematically predict and compare single-cell perturbation responses. The models span diverse theoretical frameworks and algorithmic structures, ranging from baseline models to foundation models, GRN-based, VAE-based, and other deep learning-based approaches. These approaches offer strong adaptability and generalization capabilities, allowing effective application across diverse perturbation contexts through training or fine-tuning.

Baseline model

In single-cell perturbation prediction tasks, the baseline model provides a baseline performance that allows us to compare more complex models better. For task 1, we use one-hot encoding to represent the cell's gene expression under perturbation A as p_A. The one-hot encoding of perturbations A is embedded into z_A. The gene expression of the cell under the control (no perturbation) condition is embedded into z_{ctrl}. The gene expression after combined perturbations can be represented as $x' = \text{MLP}(z_A + z_{\text{ctrl}})$. For task 2, we still use one-hot encoding to represent the gene expression of the cell under perturbation A as p_A and perturbation B as p_B. And the one-hot encoding of perturbations A and B are embedded into z_A and z_B, respectively. The gene expression after combined perturbations can be represented as $x' = \text{MLP}(z_A + z_B + z_{\text{ctrl}})$. For Task 3, we train the model for other cells under a single perturbation A, which is defined as $F_A(Z_{\text{other_ctrl}} + p_A)$, where $z_{\text{other_ctrl}}$ is the embedding of the gene expression of other cells under the control (no perturbation) condition, and p_A is the one-hot encoding of perturbation A. For the gene expression of the target cell under perturbation A, it can be represented as $F_A(z_{\text{target_ctrl}} + p_A)$, where $z_{\text{target_ctrl}}$ is the embedding of the gene expression of the target cell under the control condition.

Foundation model-based

The pre-trained models scGPT(v1.0) and scFoundation(v1.0) were employed for the perturbation prediction task, with fine-tuning performed based on the strategy used in the scGPT tutorial. In their approach, a Binary Condition Token was appended at each input gene position to denote the perturbation status of the gene. Besides, we implemented a strategy wherein a non-perturbed control cell was used as the input and a perturbed cell as the target. This was achieved by randomly matching each perturbed cell with a control cell, thereby forming input-target pairs. The models were trained using the default settings recommended in the tutorials, with scGPT undergoing 2 or 5 epochs of training for comparison and scFoundation trained for 2 epochs.

GRN-based

GEARS(v1.0) was utilized to model gene perturbations by seamlessly integrating knowledge graph information with gene embeddings. We followed the tutorial guidelines, employing the default settings as recommended.

For the scELMo(v1.0) model, we followed the tutorial's instructions and incorporated gene embeddings derived from GPT as inputs for gene embedding in GEARS.

VAE-based

The VAE-based approaches, scGen(v2.1.0) [26] and trVAE(v1.0.0) [27], were utilized for single-cell RNA sequencing (scRNA-seq) perturbation prediction.

scGen leverages a variational autoencoder (VAE) framework integrated with vector arithmetic to simulate cellular transitions between conditions, such as control and stimulation. The model encodes gene expression profiles into a latent space, reconstructing them via a decoder while learning difference vectors between cellular states.

This enables scGen to predict unobserved perturbation responses with high generalization. We implemented scGen following the tutorial provided by the authors, adhering to the recommended architecture and hyperparameters.

trVAE builds on the Conditional Variational Autoencoder (CVAE) framework to model perturbation effects under scenarios lacking paired control and perturbed cells. The model encodes cellular states into a shared latent space while applying condition-specific transformations via perturbation vectors. This is optimized using a masked learning objective and Maximum Mean Discrepancy (MMD) regularization to align distributions across conditions. The implementation followed the author's tutorial, with early stopping enabled and the maximum iterations set to 1000, while all other hyperparameters were kept as default.

Other deep learning-based

The Compositional Perturbation Autoencoder (CPA,v0.8.7) [28] combines deep learning flexibility with linear model interpretability to predict cellular responses to perturbations. The model uses an encoder to map gene expression data into a disentangled latent space, removing perturbation-specific information via adversarial training. The latent embedding is then combined with perturbation and covariate embeddings, scaled by a learned dose-response function for cell-specific predictions. The decoder reconstructs gene expression profiles, enabling predictions for unseen perturbations. We followed the authors' tutorial and used default hyperparameters for implementation.

scPreGAN(v1.0) [29] integrates autoencoders with generative adversarial networks (GANs) to predict cellular responses to perturbations. The architecture includes an encoder, two generator networks, and two discriminators. The encoder extracts features, while the generators predict perturbed data from latent vectors. Pre-training combines reconstruction and adversarial losses, optimizing data fidelity and distribution realism. We implemented scPreGAN based on the authors' GitHub tutorial, adhering to default settings.

Evaluation metrics

In evaluating model performance, selecting appropriate assessment metrics is crucial. To achieve a comprehensive evaluation, this study employs four categories of metrics: error, goodness-of-fit, correlation, and distributional similarity. These metrics provide a multi-dimensional perspective on the model's predictive accuracy, explanatory power, and distributional consistency. Specifically, error metrics (such as MSE, RMSE, MAE, and L2) quantify the deviations between predicted and actual values; goodness-of-fit metrics (such as R-squared) assess the model's capacity to account for overall data variability; correlation metrics (including Pearson correlation and cosine similarity) measure the linear relationship between predicted and actual values; and distributional similarity metrics (such as MMD and Wasserstein distance) evaluate the alignment between predicted and actual distributions, capturing nuanced distributional characteristics in high-dimensional data.

Error Metrics

Mean Squared Error(MSE) is a commonly used metric for evaluating model error, quantifying the mean squared difference between predicted and actual values. MSE is particularly suited to assessing prediction bias and overall accuracy, given its sensitivity to larger errors. In calculation, MSE squares the prediction error for each sample and then averages these values. The formula is as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples. **Root Mean Squared Error(RMSE)** is similar to MSE, being the square root of MSE, but its units are consistent with the original data, providing a more intuitive reflection of the magnitude of prediction errors. It is obtained by taking the square root of MSE, defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

Mean Absolute Error(MAE) measures the absolute error between predicted and actual values, is less sensitive to outliers than MSE, and is suitable for scenarios where one does not wish to penalize large errors. MAE was calculated by taking the average of the absolute difference between the predicted and actual values, defined as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

L2 Norm(L2) is used to measure the Euclidean distance between predicted and actual values, serving as an effective tool for describing the error space. It calculates the L2 norm of the difference vector between predicted and actual values, defined as follows:

$$L2 = \sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

Goodness-of-Fit Metric

R-squared(R^2) used to measure a model's ability to explain the total variance, serving as an indicator of model fit. It represents the degree of correlation between predicted and actual values and is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5}$$

In this formula, $\bar{y}$ is the mean of the actual values. The numerator, $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, represents the residual sum of squares (RSS), which is the total squared error. The denominator, $\sum_{i=1}^{n}(y_i - \bar{y})^2$, represents the total sum of squares (TSS) of the actual values.

Correlation Metrics

Pearson's Correlation measures the linear relationship between predicted and actual values, making it suitable for analyzing the strength of association between datasets. It is calculated by dividing the covariance between predicted and actual values by the product of their standard deviations, defined as:

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{6}$$

Cosine similarity measures the directional similarity between two vectors, making it suitable for evaluating the similarity between model predictions and actual results. The cosine similarity between the predicted and actual value vectors is defined as:

$$\text{Cosine Similarity}(\mathbf{y_i}, \hat{\mathbf{y}}_\mathbf{i}) = \frac{y_i \cdot \hat{y}_i}{\|y_i\| \times \|\hat{y}_i\|} = \frac{\sum_{i=1}^{n} y_i \hat{y}_i}{\sqrt{\sum_{i=1}^{n} y_i^2} \times \sqrt{\sum_{i=1}^{n} \hat{y}_i^2}} \tag{7}$$

Distributional Similarity Metrics

Maximum Mean Discrepancy(MMD) is used to measure the distance between two distributions, making it suitable for evaluating the difference between predicted and actual distributions. It calculates the mean discrepancy between the predicted and actual data using a kernel function, defined as:

$$\text{MMD}^2 = \mathbb{E}[k(y_i, y_i')] + \mathbb{E}[k(\hat{y}_i, \hat{y}_i')] - 2\mathbb{E}[k(y_i, \hat{y}_i)] \tag{8}$$

Here, y_i and $\hat{y}_i$ represent samples from two different distributions, while y_i' and $\hat{y}_i'$ are samples within the same distribution. $k(y_i, \hat{y}_i)$ is the kernel function, and we use the RBF kernel in this study.

Wasserstein Distance is used to measure the geographical distance between two probability distributions and is commonly used to assess the similarity between distributions. By solving the optimal transport problem, it calculates the cumulative discrepancy between the two distributions, defined as:

$$W(P, Q) = \inf_{\gamma \in \Pi(P,Q)} \mathbb{E}_{(y_i, \hat{y}_i) \sim \gamma} \left[\|y_i - \hat{y}_i\|\right] \tag{9}$$

In the formula, P and Q are two different probability distributions, and $\gamma \in \Pi(P, Q)$: represent the set of all joint distributions from P to Q.

Calculation of ranking and outperformance score

For each perturbation and performance metric, we assigned a rank to each predictive model. Specifically, all models were evaluated on the same perturbation using a chosen metric (for example, mean squared error [MSE]). Models were sorted either in ascending order (for metrics where lower values indicate better performance, such as MSE, RMSE, MAE, L2, MMD, and Wasserstein distance) or in descending order (for metrics where higher values indicate better performance, such as Pearson's correlation and cosine similarity). The best-performing model (lowest error or highest correlation) was assigned a rank of 1, the second-best model a rank of 2, and so forth. In cases of identical metric values, we applied a dense ranking method, wherein subsequent ranks were incremented by one (i.e., two models tied for first place would both receive rank 1, and the next best model would receive rank 2 rather than 3).

To compare methods against one another across all perturbations, we defined an outperformance score. For two given methods, A and B, we first filtered the dataset to include only perturbations for which both methods had valid rankings. Within each perturbation, we noted which method achieved the better (lower) rank. The outperformance score for method A relative to method B was then computed as the fraction of these perturbations in which method A outperformed method B. Formally:

$$\text{Outperformance}(A, B) = \frac{\sum_p \mathbf{1}(R_{A,p} < R_{B,p})}{N}$$

where $R_{A,p}$ and $R_{B,p}$ are the ranks of methods A and B, respectively, on perturbation p, $\mathbf{1}(\cdot)$ is an indicator function that is 1 if the condition is true and 0 otherwise, and N is the total number of perturbations considered for both methods. This score produces a value between 0 and 1, representing the proportion of perturbations in which one method outperformed the other. By computing these pairwise outperformance scores for every pair of methods, we obtained a comprehensive overview of the relative performance landscape.

E-distance

E-distance is a metric used to measure the difference between two distributions in high-dimensional space, defined as the mean squared Euclidean distance between samples. In this study, we employ E-distance to quantify the high-dimensional distribution differences across different datasets, utilizing scPerturb (v0.1.0) for the calculations. For samples originating from two distributions, we first compute the Euclidean distance between each pair of samples and then calculate the average distance within each distribution as well as the average distance between the two distributions. Prior to these calculations, we standardize the single-cell data, filter out low-expression cells (min_counts=100) and low-expression genes (min_cells=50), and select the top 2000 highly variable genes as features for analysis. Detailed methodologies can be referred to in the scPerturb[8] publication.

Statistical Analysis

All statistical analyses were conducted using Python (v3.12.3) with the pandas (v1.5.3), statsmodels (v0.14.2), and NumPy (1.26.4) libraries. For each task, we performed ANOVA tests to assess the significance of performance differences across various models, perturbation conditions, and datasets. Specifically, a three-way ANOVA was applied using ordinary least squares (OLS) regression to evaluate the effects of perturbation conditions (perturb), datasets (dataset), and models (methods) on each evaluation metric. For each evaluation metric, the following linear model was used:

$$\text{value} \sim C(\text{perturb}) + C(\text{dataset}) + C(\text{methods})$$

where $C()$ denotes that the variables are treated as categorical factors. Post-hoc analysis with Tukey's HSD test was conducted to perform pairwise comparisons between models when a significant effect ($p < 0.05$) was detected. All analyses were performed using Type II Sum of Squares for ANOVA, and results were considered statistically significant at $p < 0.05$.

Perturbation prediction on unseen genes

To assess the model's response to perturbations in single genes that it has not seen, we employ a masking strategy during model training to simulate the absence of specific genes in the training dataset. Specifically, there are multiple choices for perturbed genes in each dataset, and we select some perturbed genes as unseen genes through random sampling. During the training process, the perturbation response data for these genes are completely excluded from the training set, forcing the model to learn based solely on the perturbation features of other genes. Subsequently, the model predicts the perturbation responses for these unseen genes in the test set to evaluate its performance on unknown single-gene perturbations.

Perturbation prediction on combinatorial perturbation

For predicting combinatorial gene perturbations, we employed a random sampling strategy to select multiple gene pairs from each dataset, similar to the GEARS paper. We designed the following three training configurations to evaluate the models' generalization capabilities under varying levels of prior knowledge: Seen0: In this configuration, neither the gene pair nor their individual single-gene perturbations appear in the training data. This setup assesses the model's ability to predict responses to completely novel gene combinations. Seen1: The training set includes single-gene perturbation data for one gene of the pair, while the other gene remains unseen. This configuration tests whether the model can leverage partial prior information to predict the effects of combinatorial perturbations. Seen2: The training data contains single-gene perturbation data for both genes in the pair, but their combined perturbation is not included in the training stage. This setup evaluates the model's capacity to infer combinatorial perturbation responses based on individual gene perturbation information. By randomly selecting gene pairs, we ensured that the perturbation response of each gene combination in the test set was entirely novel to the model, thereby

providing a comprehensive evaluation of its generalization performance in predicting combinatorial gene perturbations.

Perturbation prediction on unseen cell types

In the cell state transfer prediction task, we explored the models' generalization capabilities across different cell types. We trained the models using perturbation data from multiple cell types or cell lines, with the aim of predicting the perturbation responses of a target cell type that was masked during training. Specifically, the training set included perturbation response data from only a subset of cell types, while data from the target cell type were completely excluded. The models were then tasked with predicting the perturbation responses for the unseen target cell type based on the information learned from the training set. This approach allows us to evaluate the models' applicability and robustness across different biological contexts.

Data availability

The raw data of the dataset generated in this study can be accessed through the BioProject service of the China National Center for Bioinformation (https://www.cncb.ac.cn/), with the project registration number PRJCA033051. Except for the data unique to this study, all other datasets used are from public resources. For specific details on their usage, please refer to the "Methods" section of the manuscript. A comprehensive list of datasets and their download sources can be found in Table 1 and Supplementary Table 1.

Code availability

The scripts used in this study are available via GitHub at: https://github.com/TianGzlab/scPerturbBench

Acknowledgements

We acknowledge the support of the Data Science Platform of Guangzhou National Laboratory and the Bio-medical Big Data Operating System (Bio-OS). We thank Lead Healthcare.AI for their help on experiments.

Author contributions

L.T. and Y.Y. designed and supervised the study. L.L., W.L., and Y.Y. performed the baseline model and data analyses with help from Y.C, Y.F, S.Z and J.K. X.F, S.L, B.L, W.R performed experiments to generate in-house datasets with help from J.C and X.L. L.L., W.L., Y.Y., and L.T. drafted the manuscript, with contributions from all authors.

Table legends

Table 1. Overview of datasets used in this study.
Supplementary Table 1. Detailed descriptions of datasets used in this study.
Supplementary Table 2. Overview of perturbation-related models.
Supplementary Table 3. List of drugs and their concentrations used in K562 cells and HDFs.

Figure legends

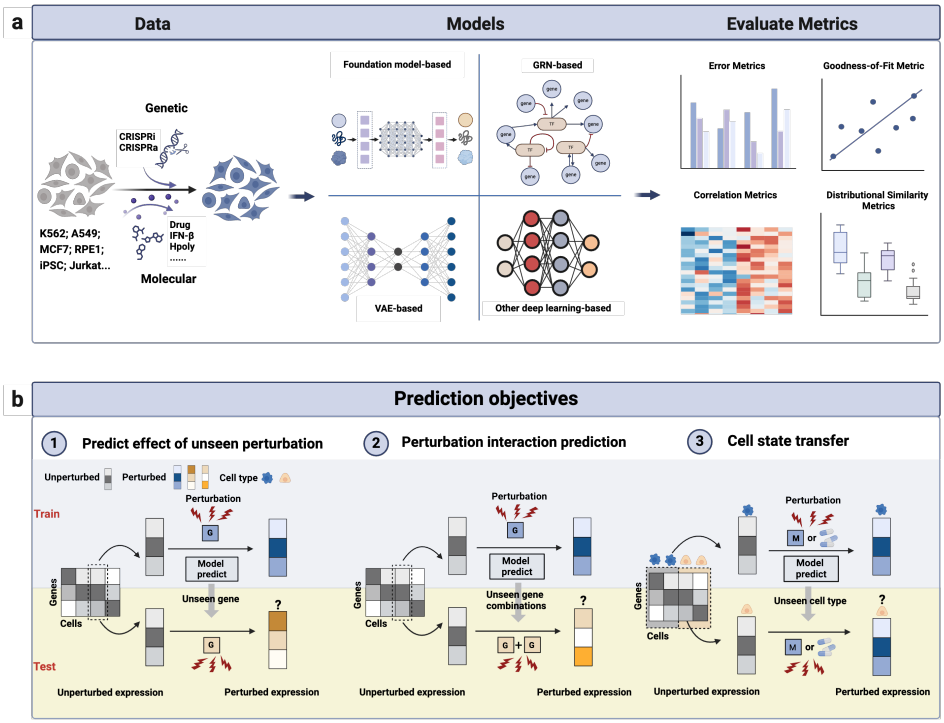

Fig. 1 Overview of the benchmarking framework. a) Data, Models, and Evaluation Metrics: This study includes diverse datasets spanning multiple cell lines under genetic (CRISPRi/CRISPRa) and molecular perturbations (e.g., drugs, IFN-β, Hpoly). A range of models—including Foundation Models, GRN, VAE and other deep learning models are evaluated using error metrics, goodness-of-fit measures, correlation, and distributional similarity metrics. b) Benchmark Prediction objectives: The framework covers three tasks: 1. Predicting responses to unseen single-gene perturbations; 2. Predicting responses to unseen gene combinations; 3. Predicting response on unseen cell types. The blue section (upper part) represents the training stage, while the yellow section (lower part) represents the evaluation stage.

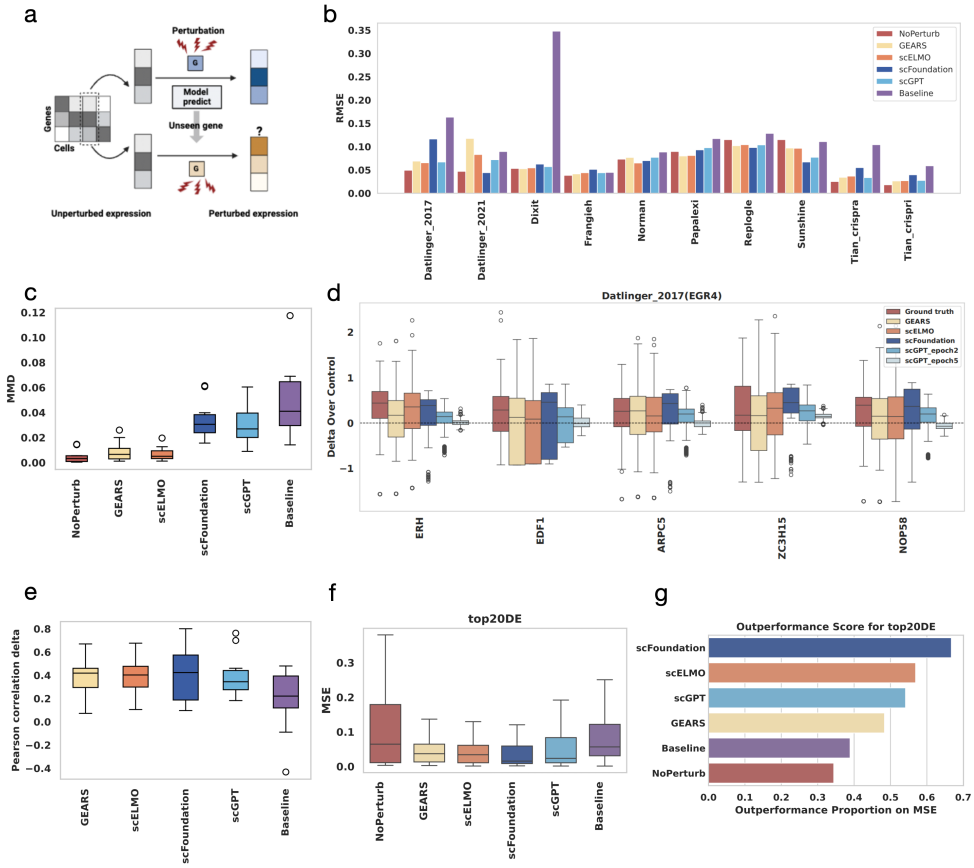

Fig. 2 Evaluation of task 1: Predicting unseen perturbation. a) Schematic representation of the task: predicting responses to perturbations of single gene not previously encountered during training. b) Average RMSE across different datasets for each model. c) Performance of different models on average MMD. d) On the Datlinger_2017 dataset, distribution of predicted and actual gene expression of the top 5 genes. e) Performance of different models on average Pearson correlation delta. f)Performance of different models on the average MSE of top 20 DE. g) Outperformance proportion of different models on MSE for top 20 DE.

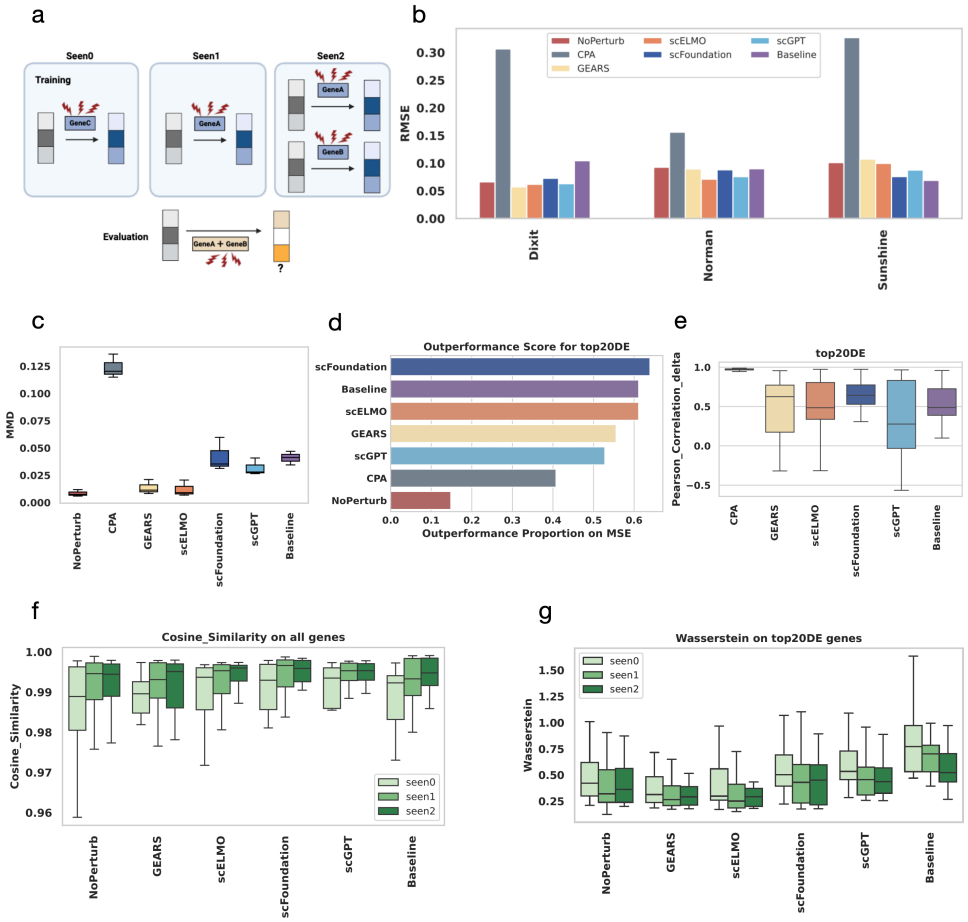

Fig. 3 Evaluation of task 2: Predicting combinatorial gene perturbations. a) Schematic representation of the task: predicting responses to perturbations of combinatorial gene. Three generalization cases represent none of the perturbation (seen0), one of the perturbation (seen1) and both of the perturbation (seen2) are seen by the model during training. Panel b-e were evaluated on the "seen2" condition. b) Average RMSE across different datasets for each model. c) Performance of different models on average MMD. d) Outperformance proportion of different models on MSE for top 20 DE. e) Performance of different models on the average Pearson correlation delta for top 20 DE. f) Box plot showing the distribution of cosine similarity across different conditions (seen0, seen1, and seen2) for various models. g) Distribution of Wasserstein for top 20 DE genes across conditions (seen0, seen1, and seen2) for various models.

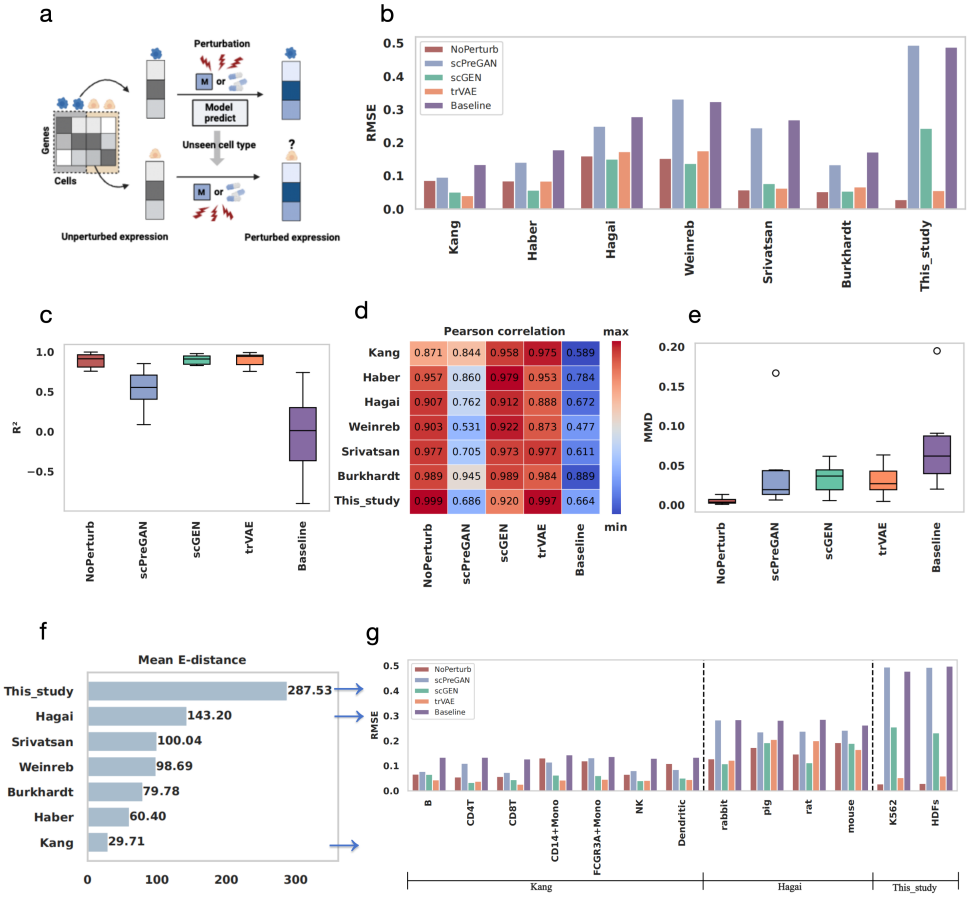

Fig. 4 Evaluation of task 3: cell state transfer. a) Schematic representation of the task: predicting the response of unseen cell types. b) Average RMSE across different datasets for each model. c) Performance of different models on average R squared. d) Heatmap of average Pearson correlation coefficients for each model across different datasets, color scaled by row. e) Performance of different models on average MMD. f) The average E-distance among cell types for each dataset, ranked by values. g) Model performance in predicting the responses of different cell types, evaluated using RMSE on the selected Kang, Hagai, and This_study datasets.

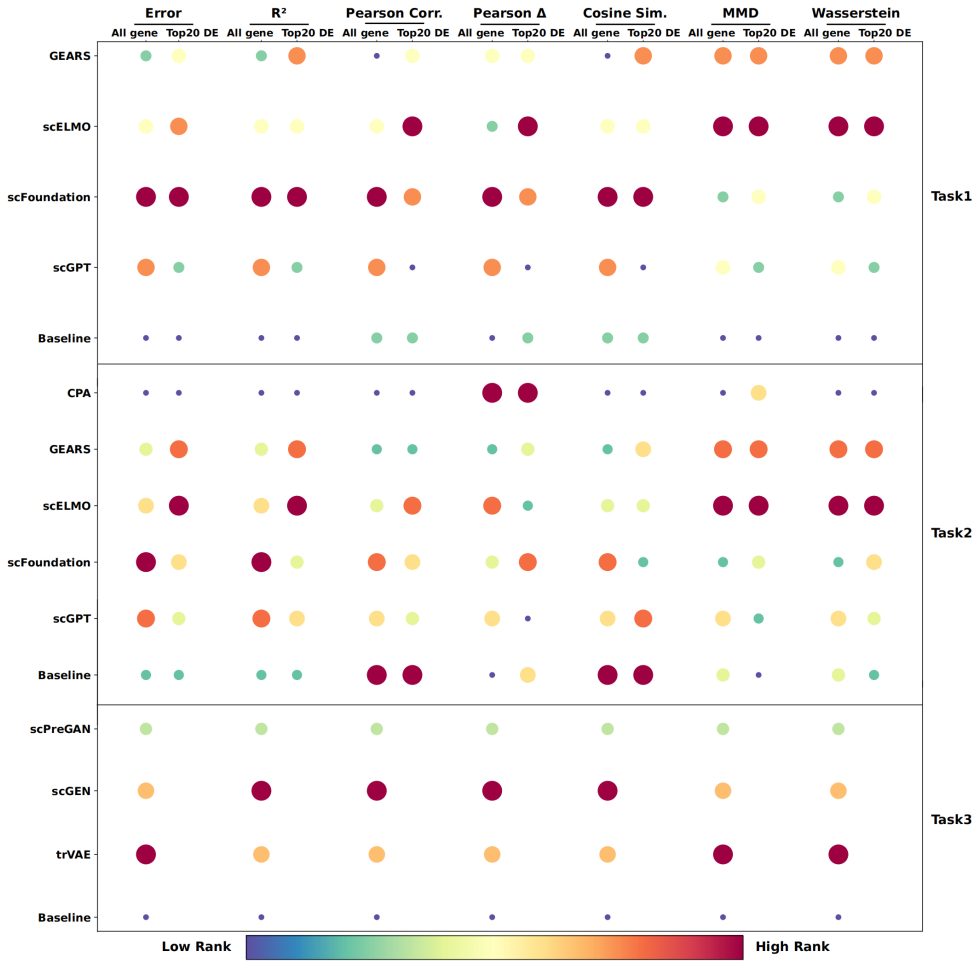

Fig. 5 The ranking of evaluation metrics across different models on different tasks. The size and color intensity of the points reflect the rank of evalutaion metrics.

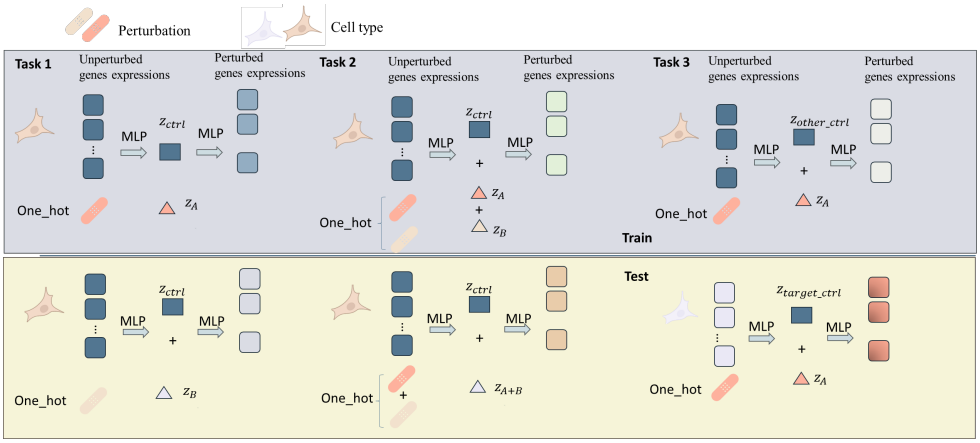

Fig. S1 Summary of baseline model used in this study

Supplementary Fig. 2

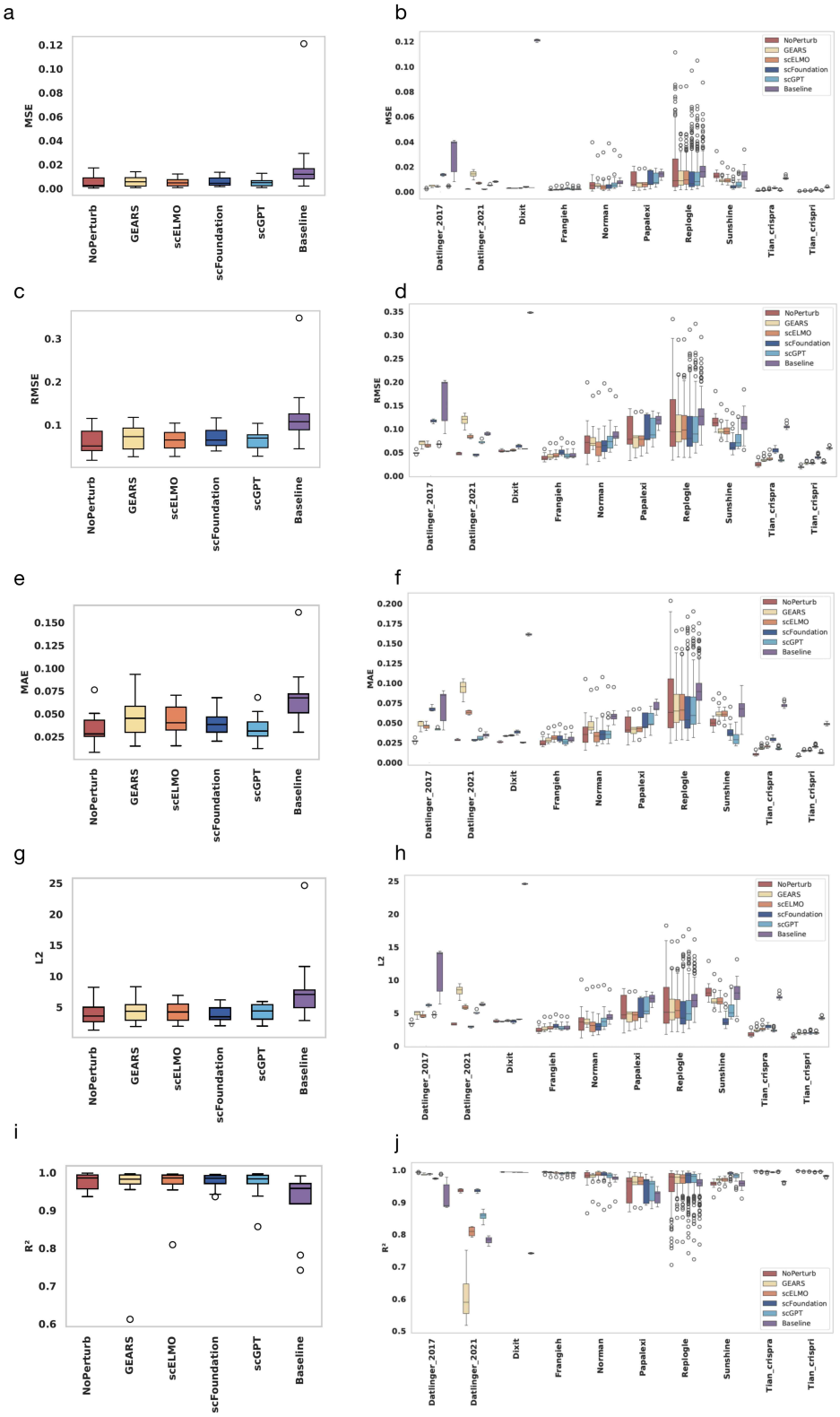

Fig. S2 This figure shows the evaluation result of task 1. a) The MSE distribution of each model over all dataset. b) The MSE distribution of each model on each dataset. c) The RMSE distribution of each model over all dataset. d) The RMSE distribution of each model on each dataset. e) The MAE distribution of each model over all dataset. f) The MAE distribution of each model on each dataset. g) The L2 distribution of each model over all dataset h) The L2 distribution of each model on each dataset. i) The R squared distribution of each model over all dataset. j) The R squared distribution of each model on each dataset.

Supplementary Fig. 3

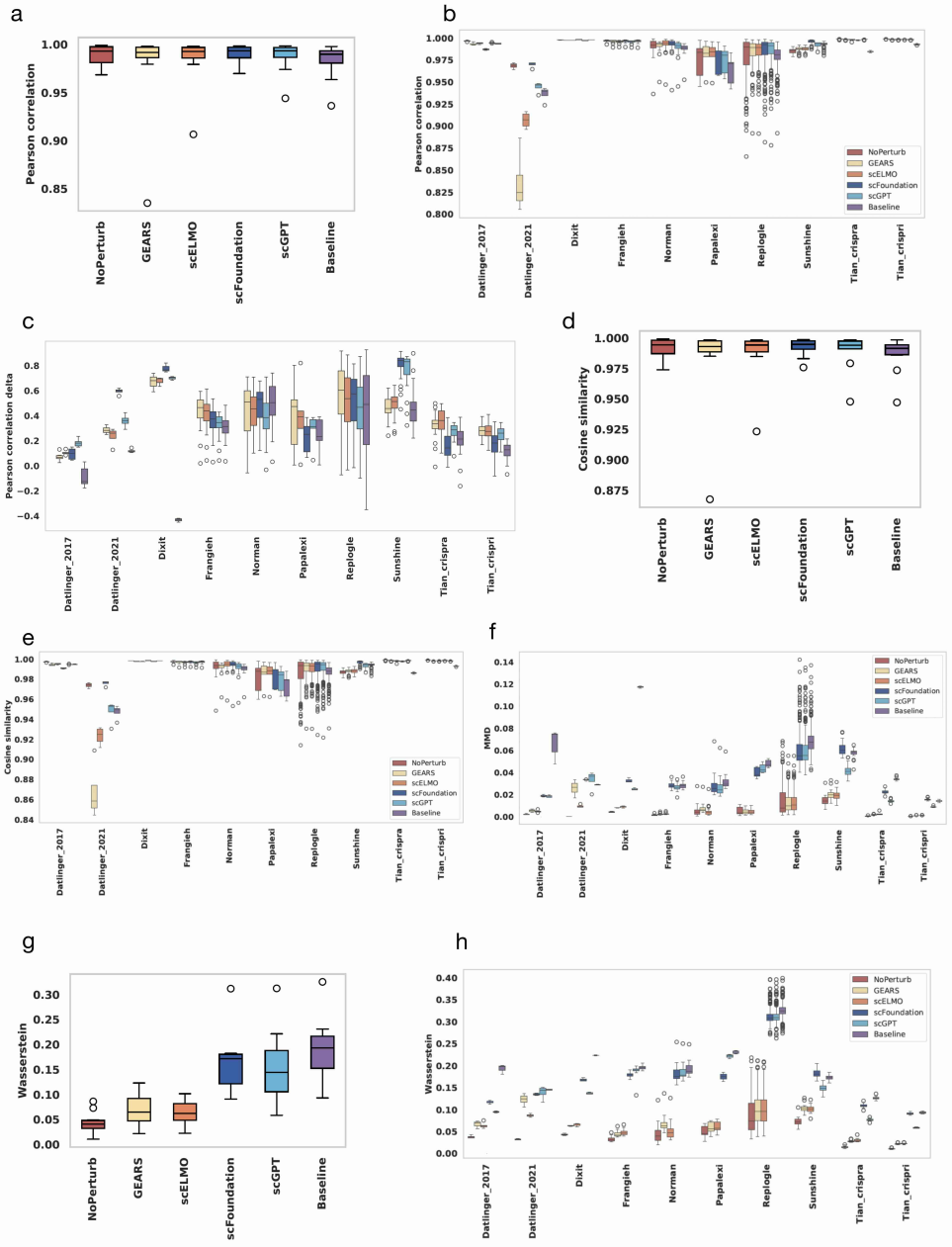

Fig. S3 This figure shows the evaluation result of task 1. a) Distribution of Pearson correlation for each method. b) Distribution of Pearson correlation for each method on each dataset. c) Distribution of Pearson correlation delta for each method on each dataset. d) Distribution of cosine similarity among different methods. e) Distribution of cosine similarity for each method on each dataset. f) Distribution of MMD for each method on each dataset. g) Distribution of Wasserstein for each method. h) Distribution of Wasserstein for each method on each dataset.

Supplementary Fig. 4

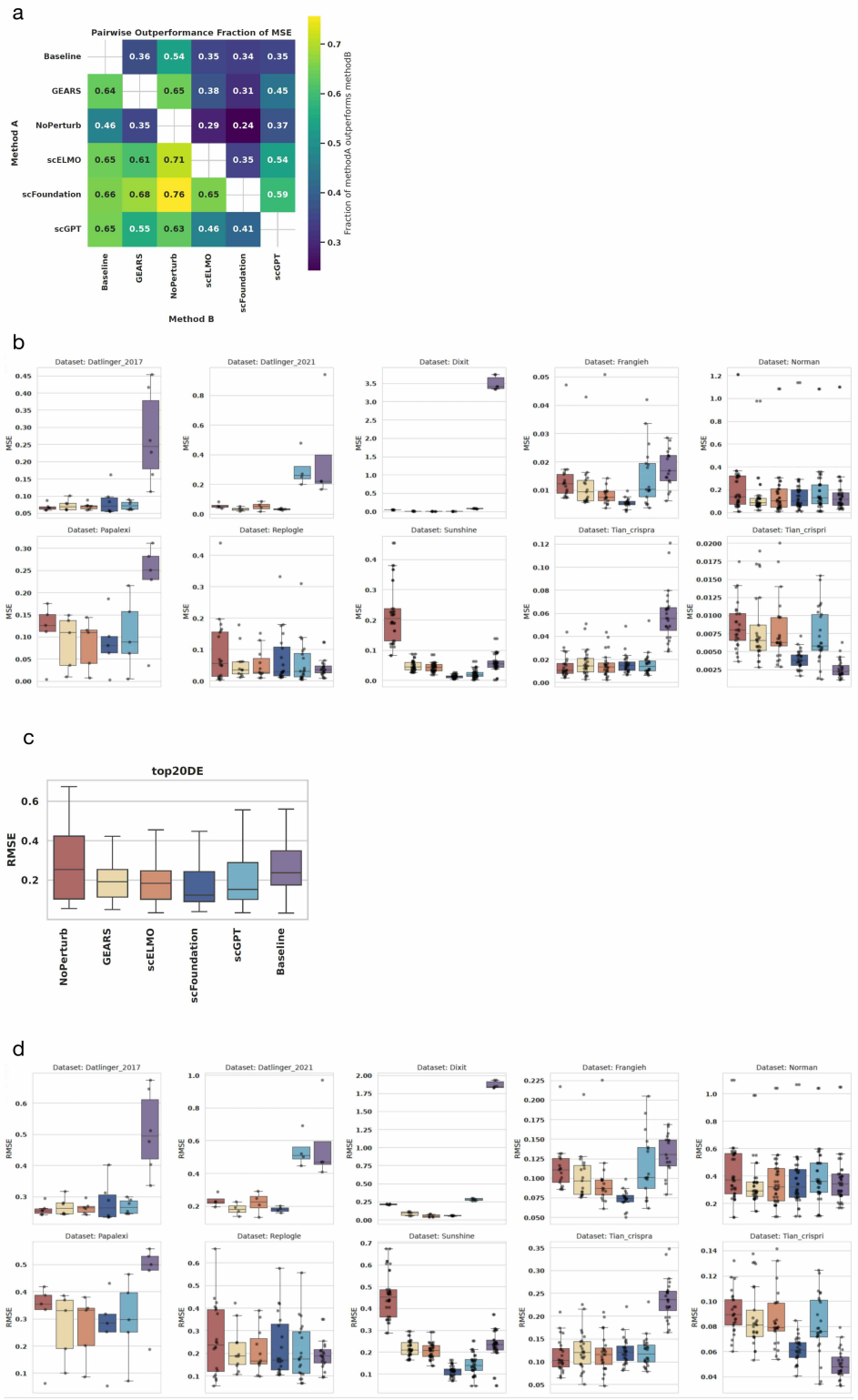

Fig. S4 This figure shows the evaluation result of task 1. a) Pairwise outperformance fraction of MSE on top20 DE genes. The heatmap showing the fraction of methods A outperforms methods B. The color gradient from blue to yellow represents the fraction of outperformance. b) Distribution of MSE on top 20 DE for each method on each dataset. c) Performance of different models on the average RMSE of top 20 DE. d) Distribution of RMSE on top 20 DE for each method on each dataset.

Supplementary Fig. 5

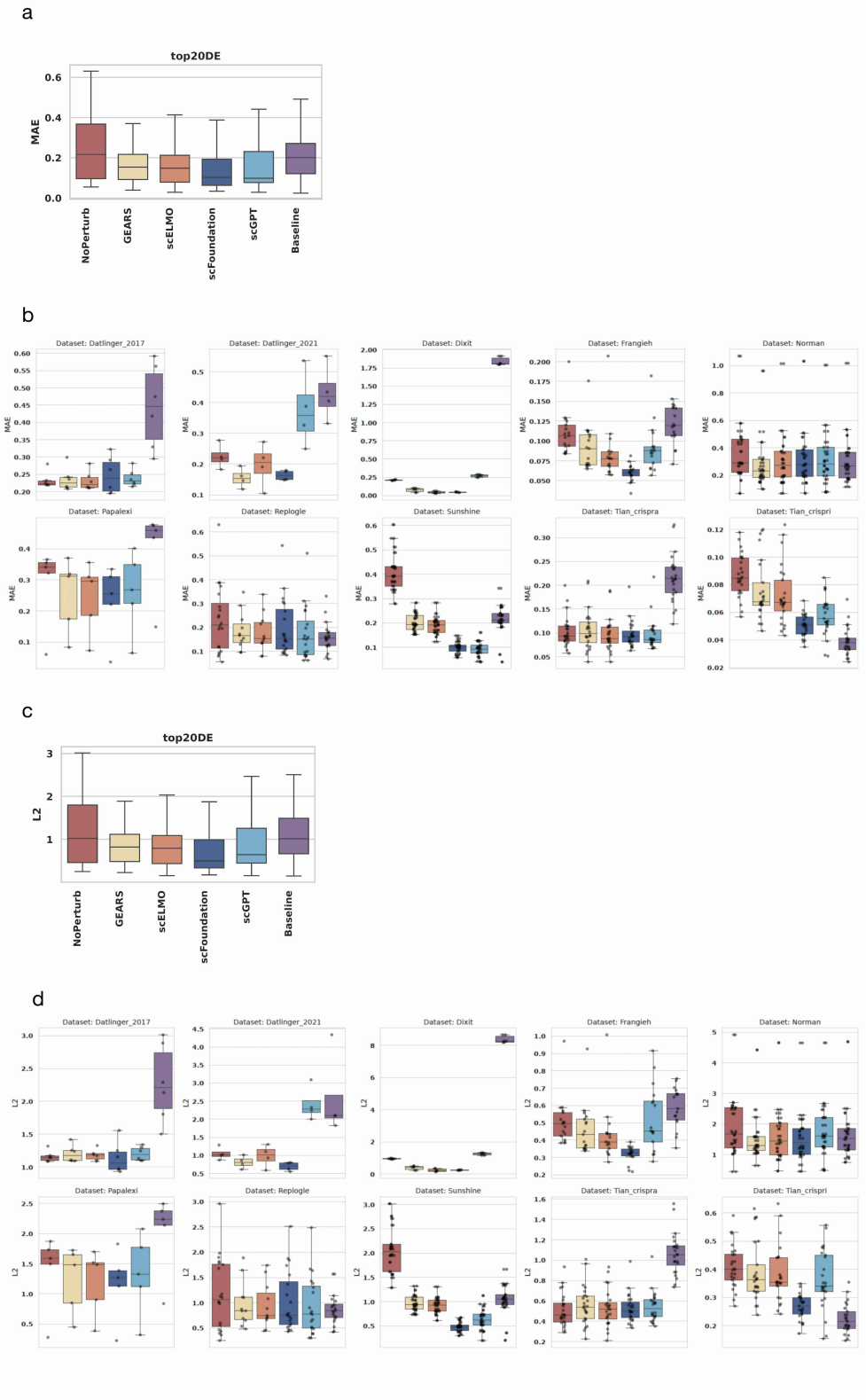

Fig. S5 This figure shows the evaluation result of task 1. a) Performance of different models on the average MAE of top 20 DE. b) Distribution of MAE on top 20 DE for each method on each dataset. c) Performance of different models on the average L2 of top 20 DE. d) Distribution of L2 on top 20 DE for each method on each dataset.

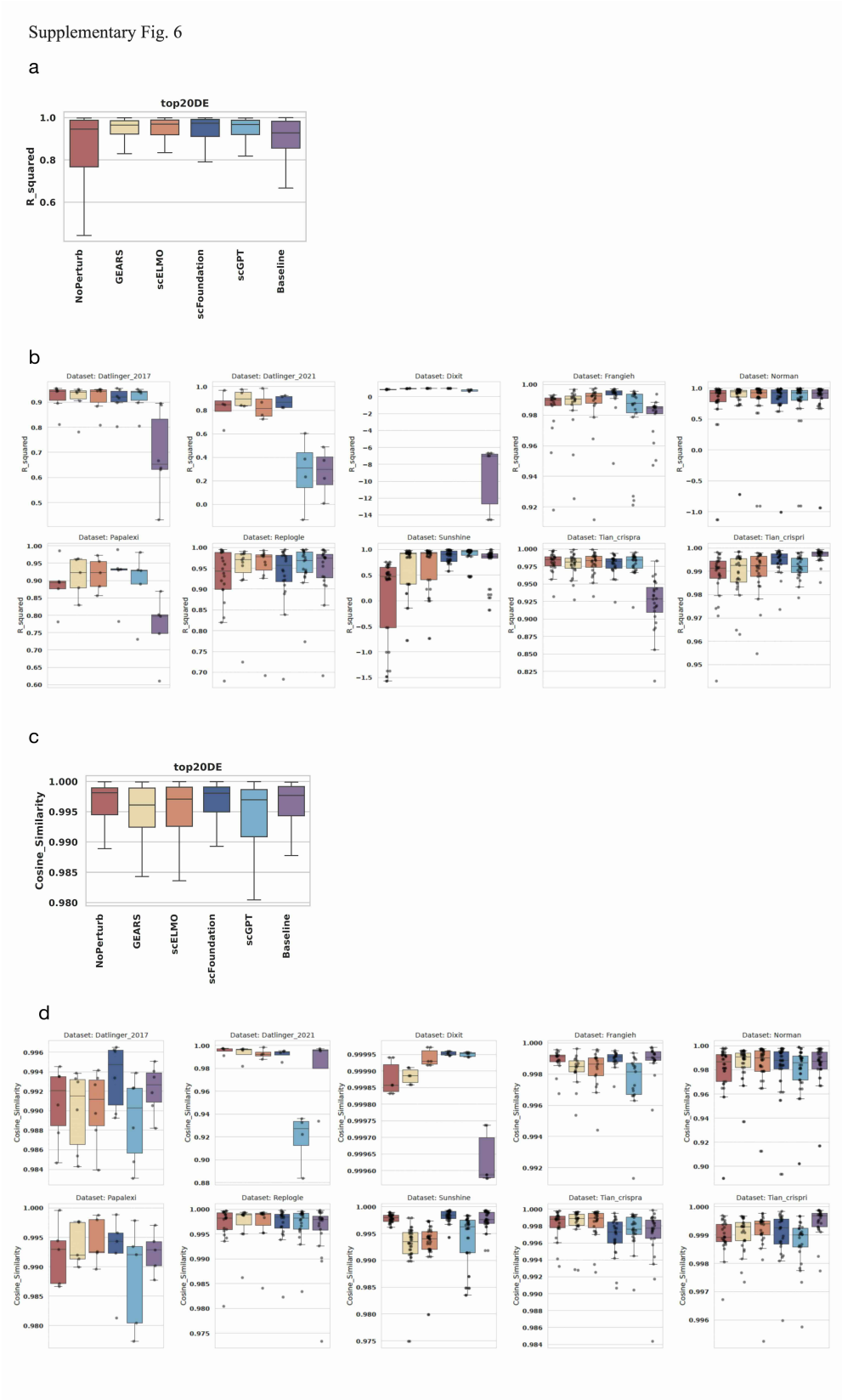

Fig. S6 This figure shows the evaluation result of task 1. a) Performance of different models on the average R squared of top 20 DE. b) Distribution of R squared on top 20 DE for each method on each dataset. c) Performance of different models on the average cosine similarity of top 20 DE. d) Distribution of cosine similarity on top 20 DE for each method on each dataset.

Supplementary Fig. 7

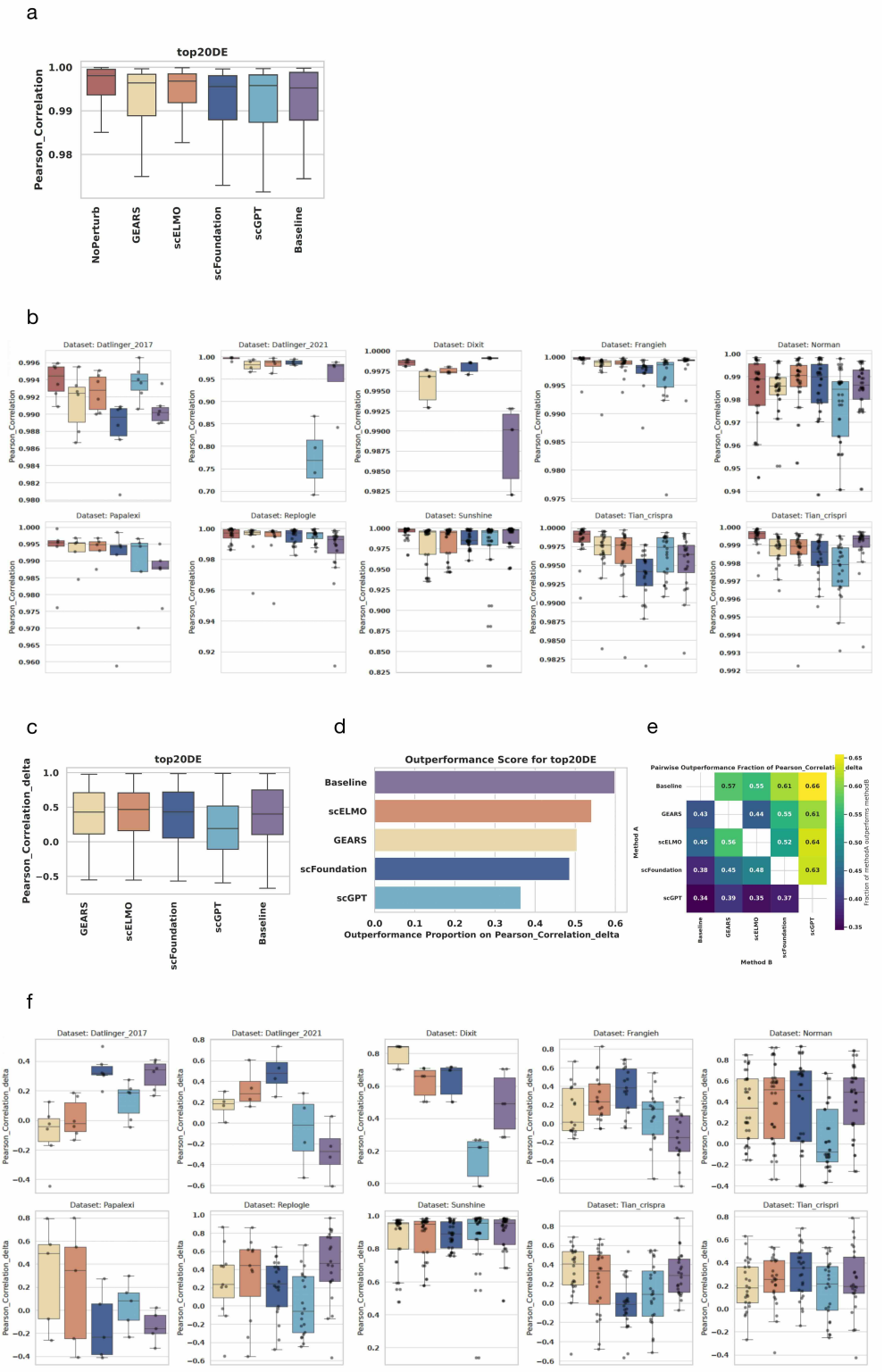

Fig. S7 This figure shows the evaluation result of task 1. a) Performance of different models on the average Pearson correlation of top 20 DE. b) Distribution of Pearson correlation on top 20 DE for each method on each dataset. c) Performance of different models on the average Pearson correlation delta of top 20 DE. d) Outperformance proportion of different models on Pearson correlation delta for top 20 DE. e) Pairwise outperformance fraction of Pearson correlation delta on top20 DE. f) Distribution of Pearson correlation delta on top 20 DE for each method on each dataset.

Supplementary Fig. 8

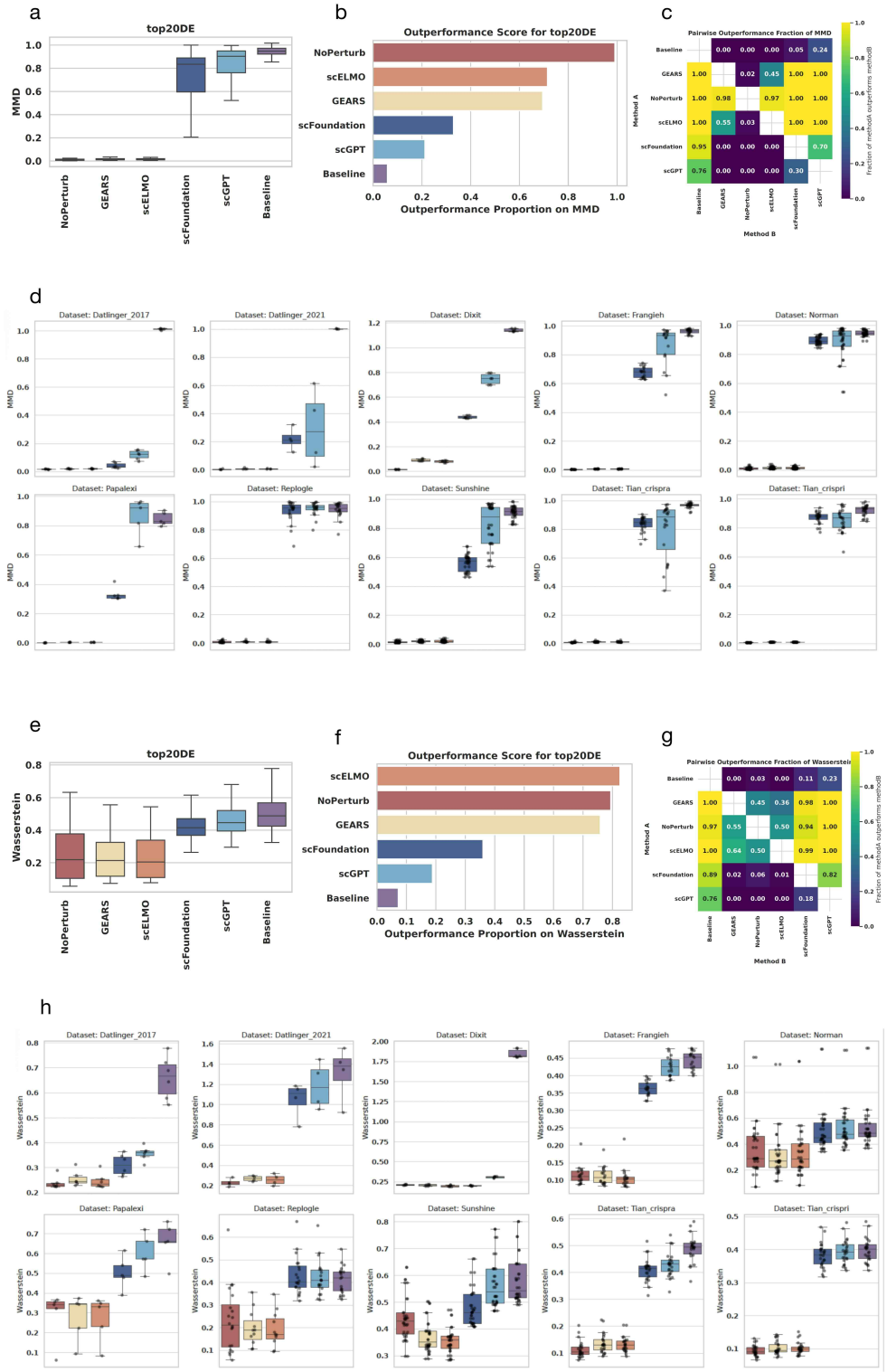

Fig. S8 This figure shows the evaluation result of task 1. a) Performance of different models on the average MMD of top 20 DE. b) Outperformance proportion of different models on MMD for top 20 DE. c) Pairwise outperformance fraction of MMD on top20 DE. d) Distribution of MMD on top 20 DE for each method on each dataset. e) Performance of different models on the average Wasserstein of top 20 DE. f) Outperformance proportion of different models on Wasserstein for top 20 DE. g) Pairwise outperformance fraction of Wasserstein on top20 DE. h) Distribution of Wasserstein on top 20 DE for each method on each dataset.

Supplementary Fig. 9

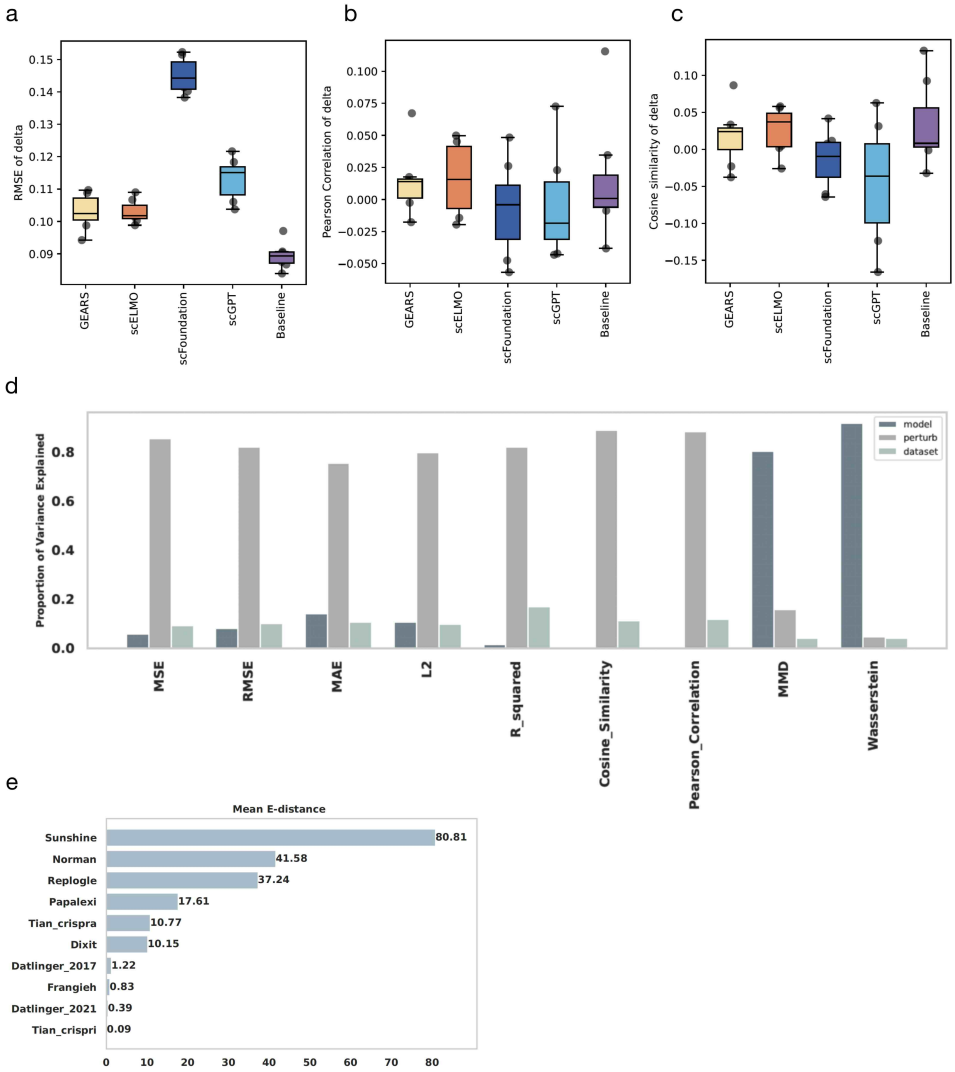

Fig. S9 This figure shows the evaluation result of task 1. a-c) Boxplots showing the generalization performance of different models across datasets, measured by (a) RMSE delta, (b) Pearson correlation delta, and (c) cosine similarity delta. d) A bar plot depicting the ANOVA results for each evaluation metric, with variance components decomposed into contributions from models, perturbations, and datasets. e) Mean E-distance for each dataset between perturbations and control.

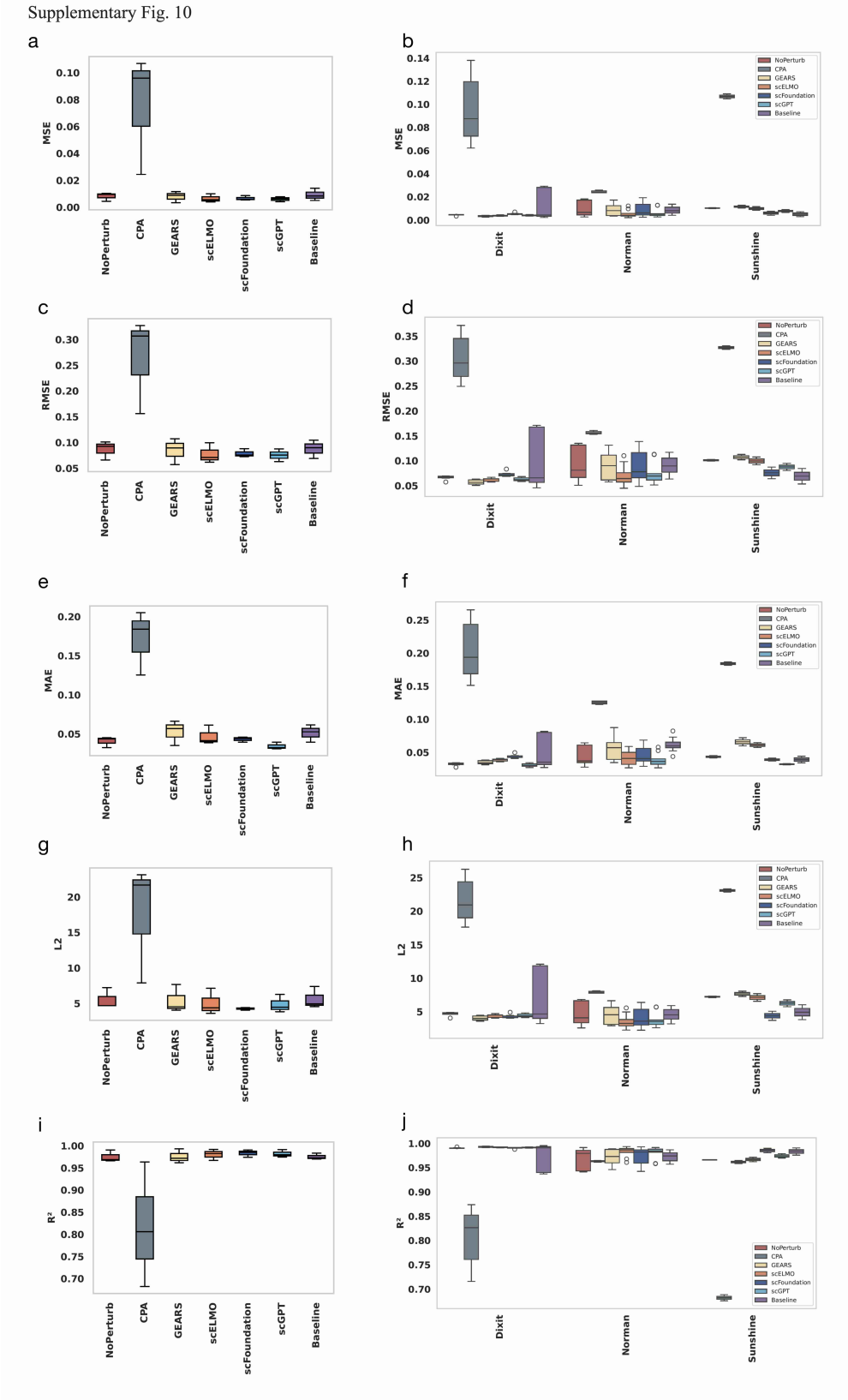

Fig. S10 This figure shows the evaluation result of task 2. a) Distribution of MSE for each method. b) Distribution of MSE for each method on each dataset. c) Distribution of RMSE for each method. d) Distribution of RMSE for each method on each dataset. e) Distribution of MAE for each method. f) Distribution of MAE for each method on each dataset. g) Distribution of L2 for each method. h) Distribution of L2 for each method on each dataset. i) Distribution of R squared for each method. j) Distribution of R squared for each method on each dataset.

Supplementary Fig. 11

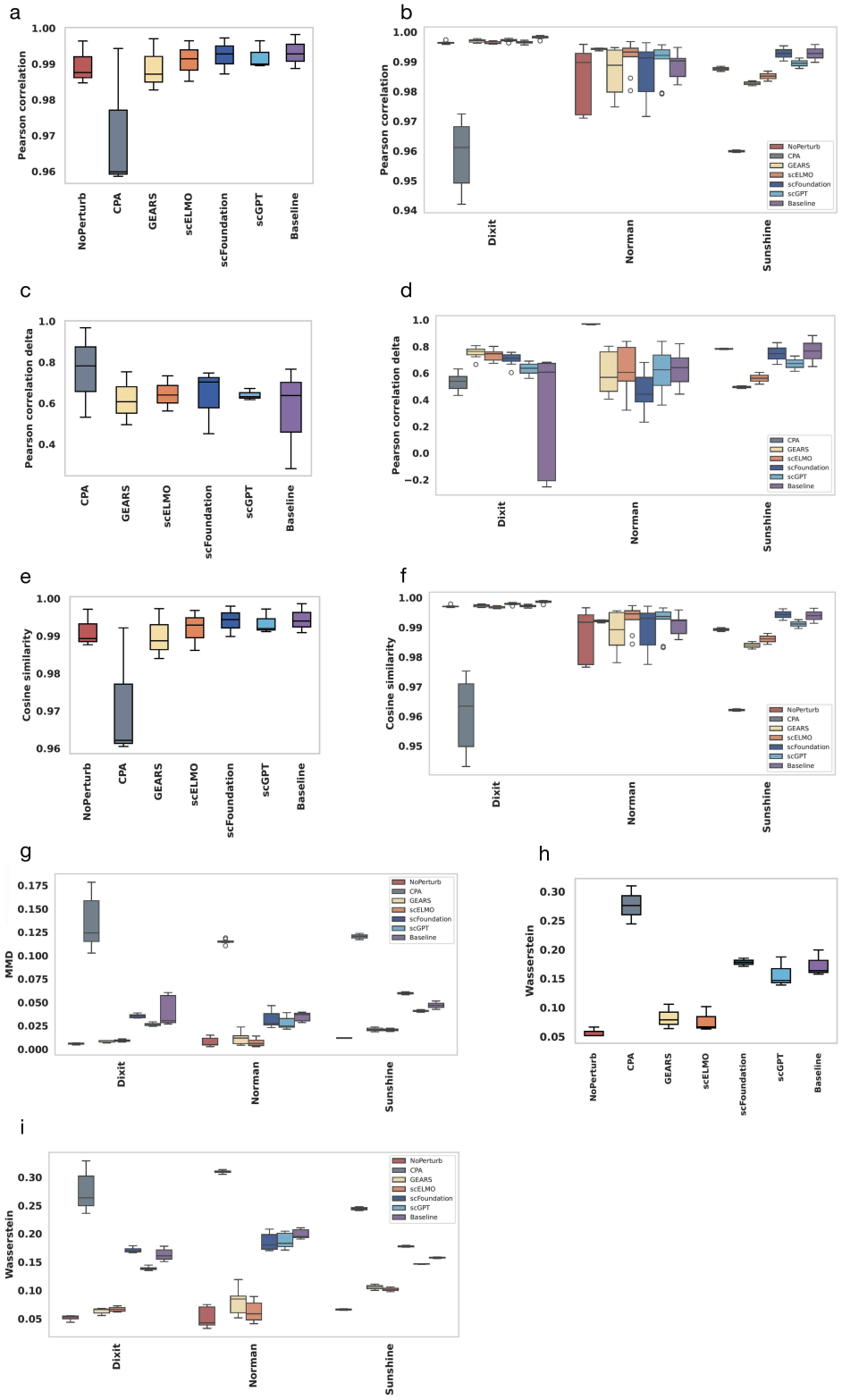

Fig. S11 This figure shows the evaluation result of task 2. a) Distribution of Pearson correlation for each method. b) Distribution of Pearson correlation for each method on each dataset. c) Distribution of Pearson correlation delta for each method. d) Distribution of Pearson correlation delta for each method on each dataset. e) Distribution of cosine similarity for each method. f) Distribution of cosine similarity for each method on each dataset. g) Distribution of MMD for each method on each dataset on each dataset. h) Distribution of Wasserstein for each method. i) Distribution of Wasserstein for each method on each dataset.

Supplementary Fig. 12

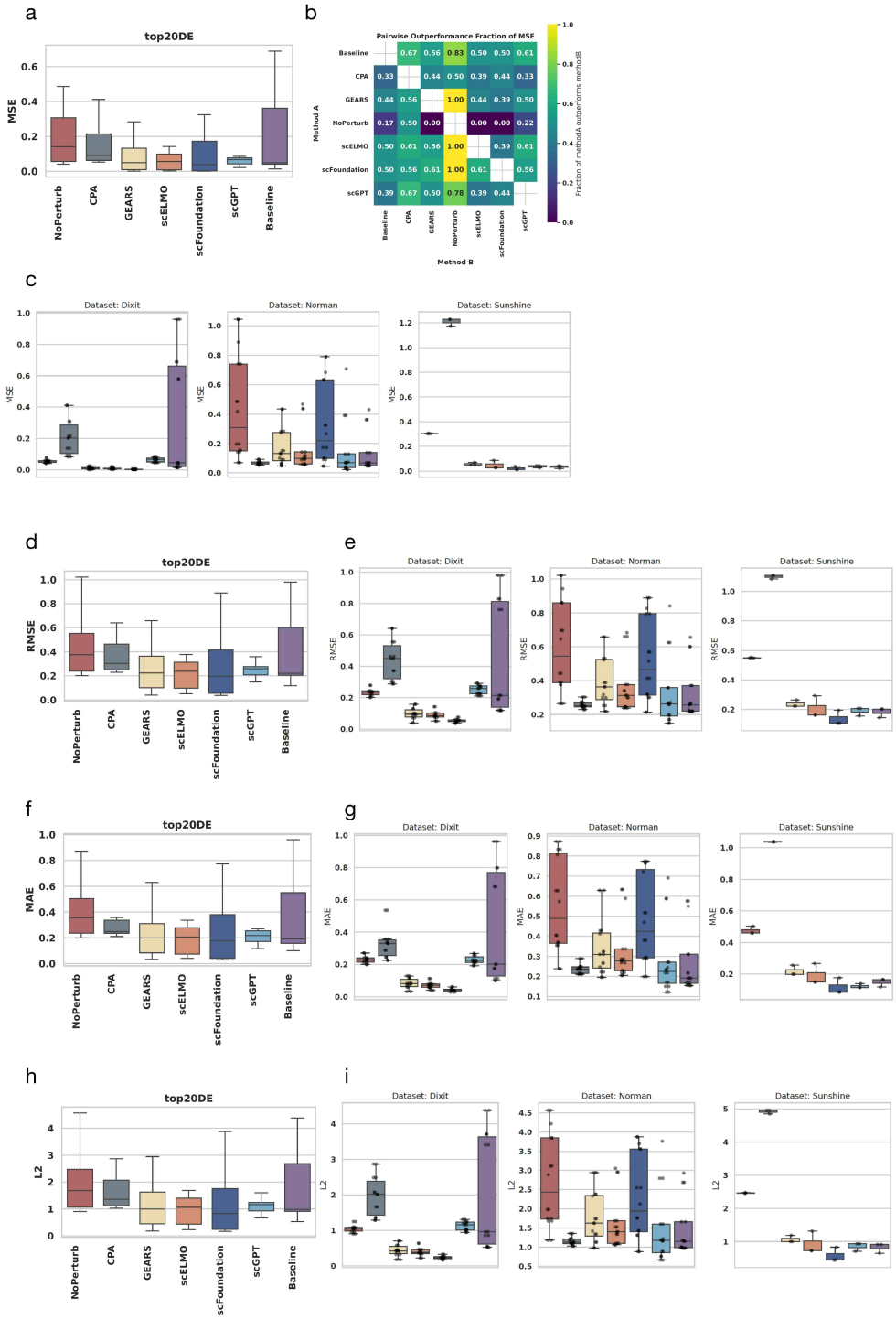

Fig. S12 This figure shows the evaluation result of task 2. a) Performance of different models on the average MSE of top 20 DE. b) Pairwise outperformance fraction of MSE on top20 DE genes. c) Distribution of MSE on top 20 DE for each method on each dataset. d) Performance of different models on the average RMSE of top 20 DE. e) Distribution of RMSE on top 20 DE for each method on each dataset. f) Performance of different models on the average MAE of top 20 DE. g) Distribution of MAE on top 20 DE for each method on each dataset. h) Performance of different models on the average L2 of top 20 DE. i) Distribution of L2 on top 20 DE for each method on each dataset.

Supplementary Fig. 13

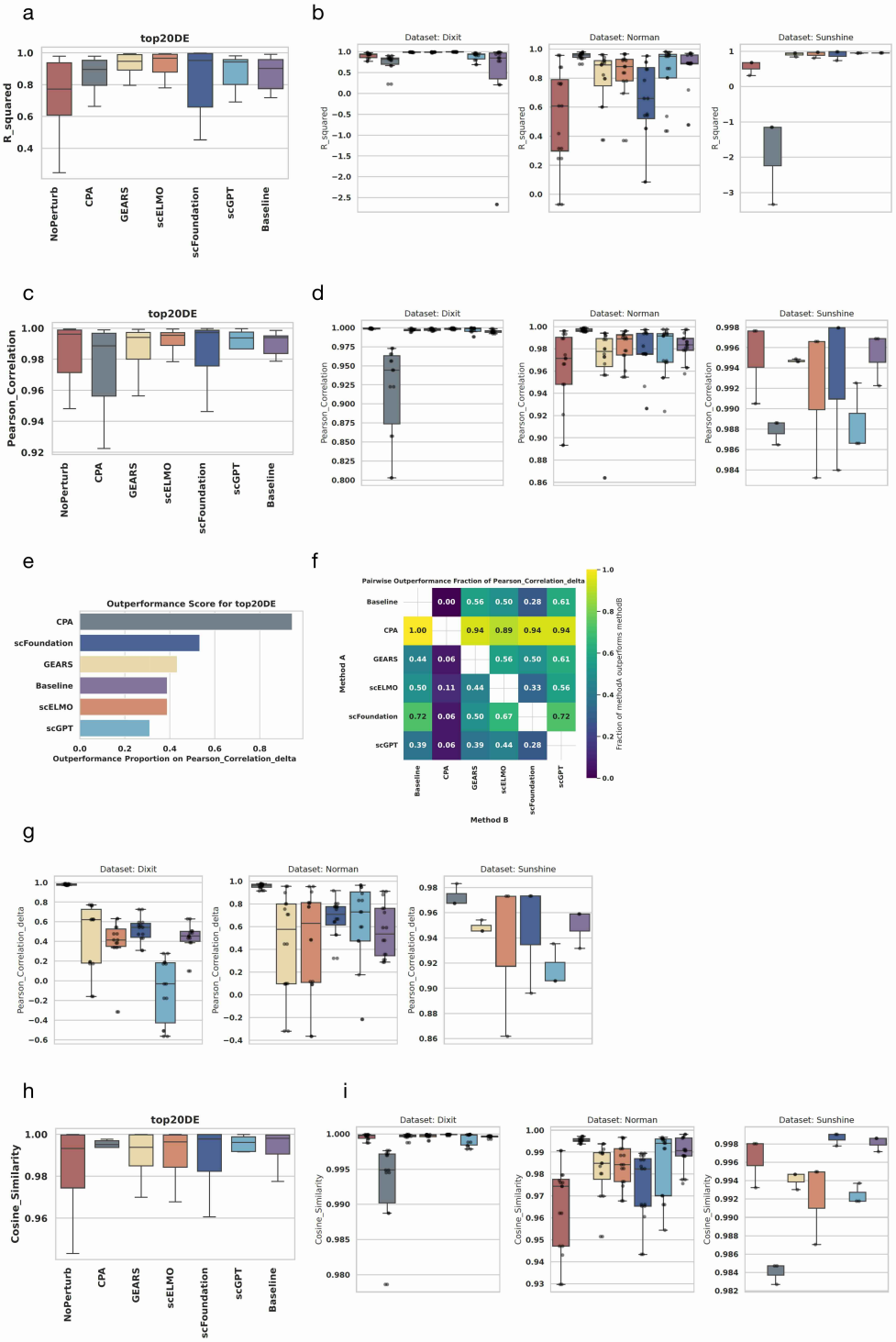

Fig. S13 This figure shows the evaluation result of task 2. a) Performance of different models on the average R squared of top 20 DE. b) Distribution of R squared on top 20 DE for each method on each dataset. c) Performance of different models on the average Pearson correlation of top 20 DE. d) Distribution of Pearson correlation on top 20 DE for each method on each dataset. e) Outperformance proportion of different models on Pearson correlation delta for top 20 DE. f) Pairwise outperformance fraction of Pearson correlation delta on top20 DE. g) Distribution of Pearson correlation delta on top 20 DE for each method on each dataset. h) Performance of different models on the average cosine similarity of top 20 DE. i) Distribution of cosine similarity on top 20 DE for each method on each dataset.

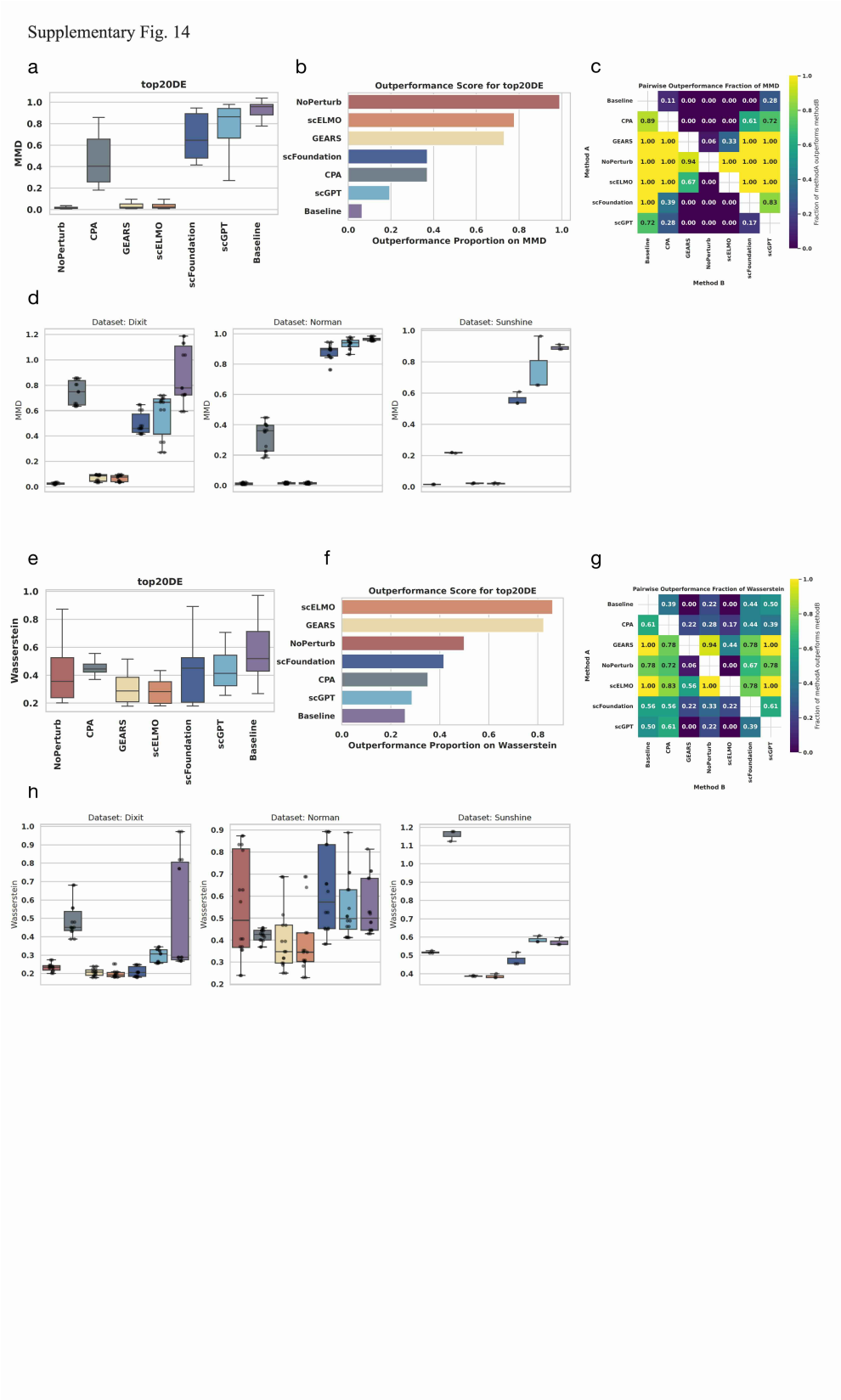

Fig. S14 This figure shows the evaluation result of task 2. a) Performance of different models on the average MMD of top 20 DE. b) Outperformance proportion of different models on MMD for top 20 DE. c) Pairwise outperformance fraction of MMD on top20 DE. d) Distribution of MMD on top 20 DE for each method on each dataset. e) Performance of different models on the average Wasserstein of top 20 DE. f) Outperformance proportion of different models on Wasserstein for top 20 DE. g) Pairwise outperformance fraction of Wasserstein on top20 DE. h) Distribution of Wasserstein on top 20 DE for each method on each dataset.

Supplementary Fig. 15

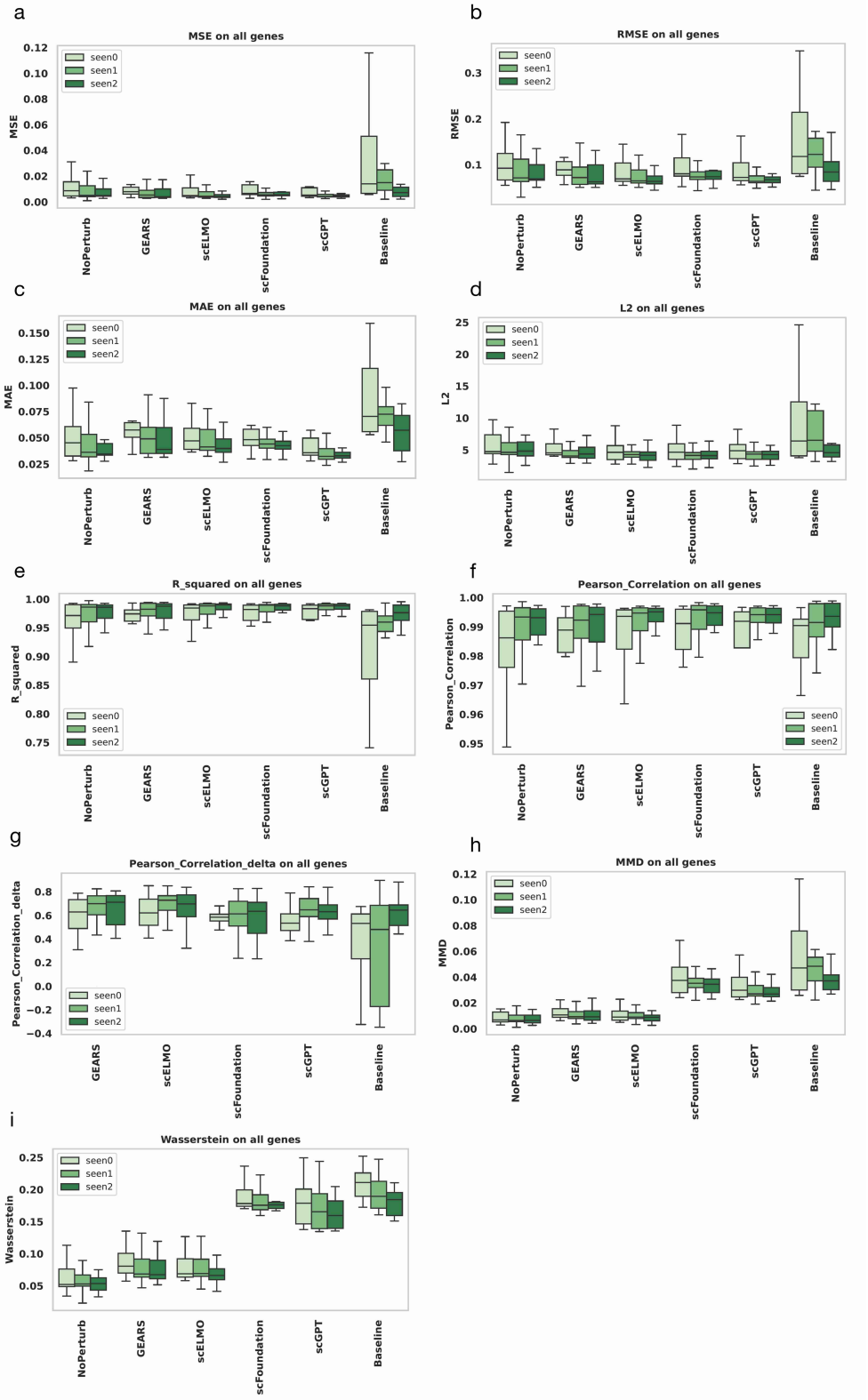

Fig. S15 This figure shows the evaluation result of task 2. a) Distribution of MSE across conditions (seen0, seen1, and seen2) for various models. b) Similar to panel a, but showing RMSE. c) Similar to panel a, but showing MAE. d) Similar to panel a, but showing L2. e) Similar to panel a, but showing R squared. f) Similar to panel a, but showing Pearson correlation. g) Similar to panel a, but showing Pearson correlation delta. h) Similar to panel a, but showing MMD. i) Similar to panel a, but showing Wasserstein.

Supplementary Fig. 16

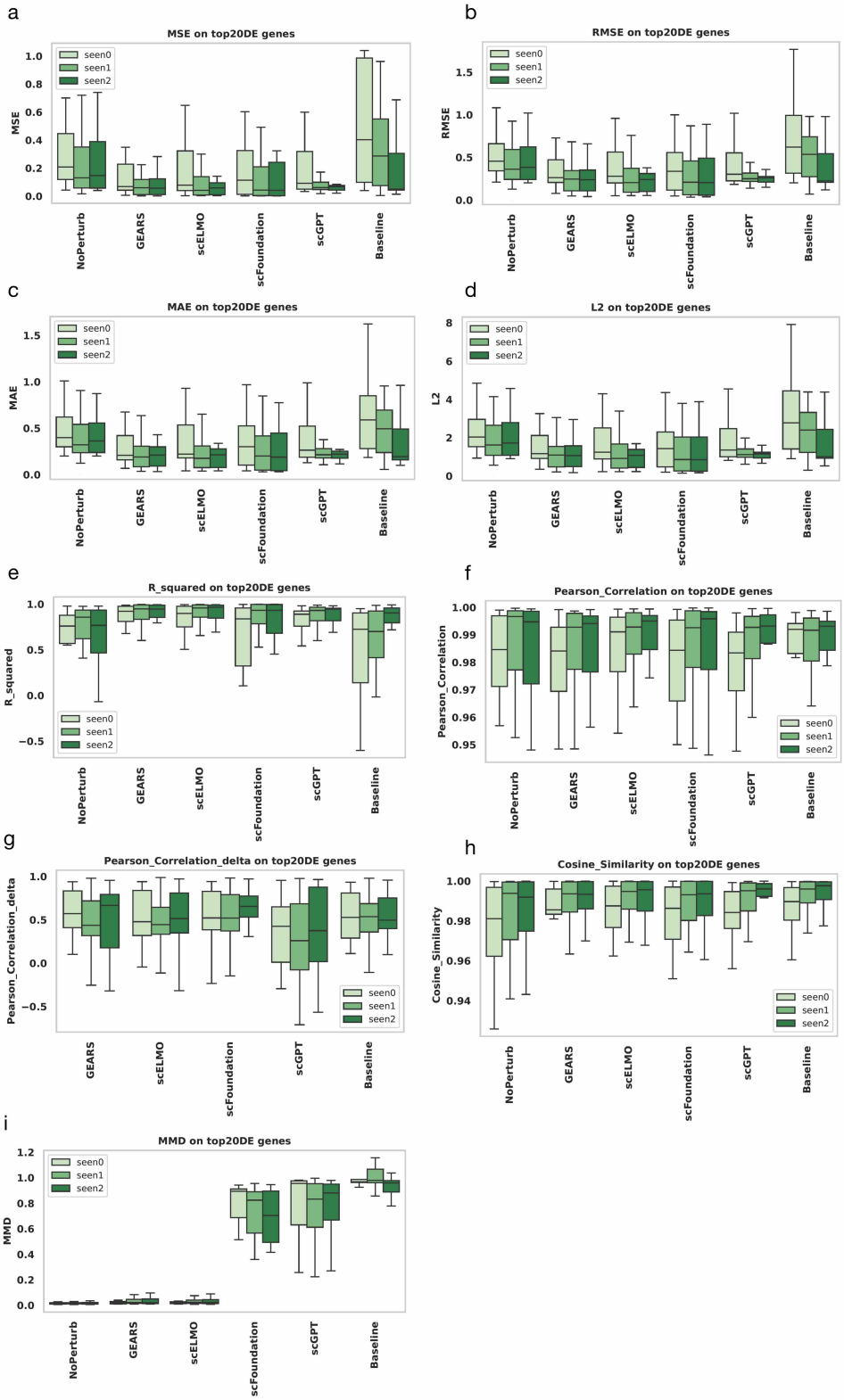

Fig. S16 This figure shows the evaluation result of task 2. a) Distribution of MSE for top 20 DE genes across conditions (seen0, seen1, and seen2) for various models. b) Similar to panel a, but showing RMSE. c) Similar to panel a, but showing MAE. d) Similar to panel a, but showing L2. e) Similar to panel a, but showing R squared. f) Similar to panel a, but showing Pearson correlation. g) Similar to panel a, but showing Pearson correlation delta. h) Similar to panel a, but showing cosine similarity. i) Similar to panel a, but showing MMD.

Supplementary Fig. 17

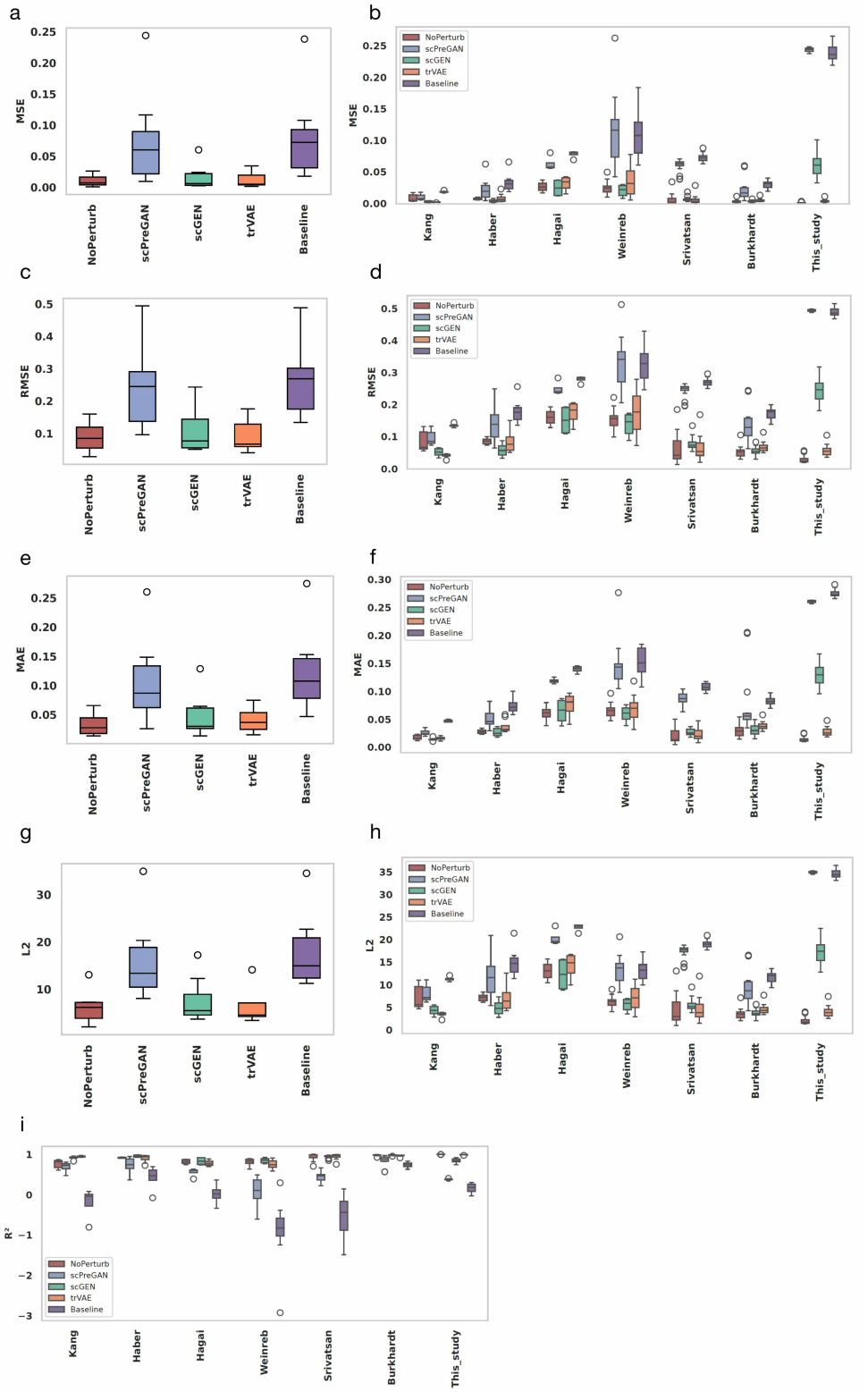

Fig. S17 This figure shows the evaluation result of task 3. a) Distribution of MSE for each method. b) Distribution of MSE for each method on each dataset. c) Distribution of RMSE for each method. d) Distribution of RMSE for each method on each dataset. e) Distribution of MAE for each method. f) Distribution of MAE for each method on each dataset. g) Distribution of L2 for each method. h) Distribution of L2 for each method on each dataset. i) Distribution of R squared for each method on each dataset.

Supplementary Fig. 18

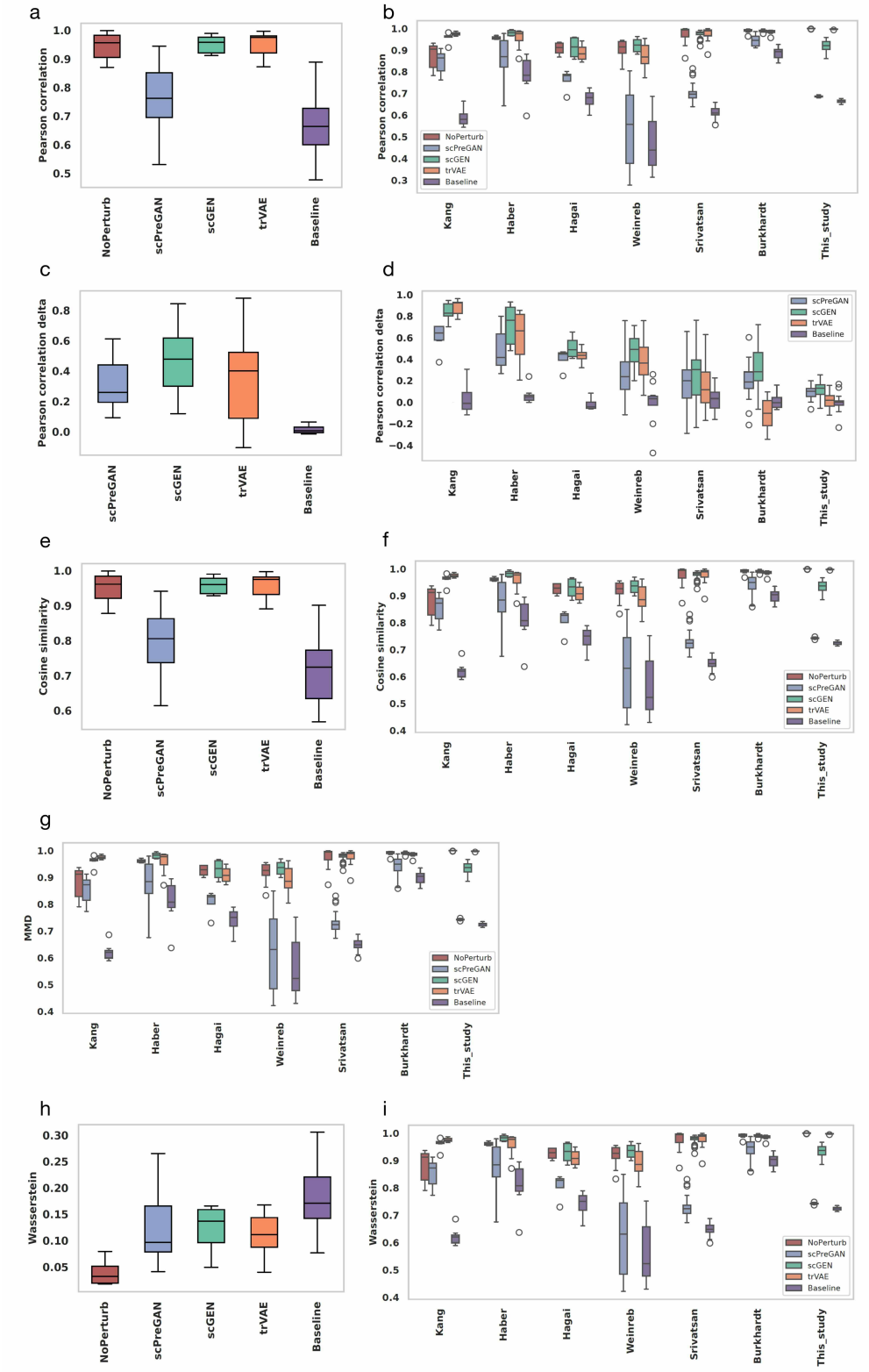

Fig. S18 This figure shows the evaluation result of task 3. a) Distribution of Pearson correlation for each method. b) Distribution of Pearson correlation for each method on each dataset. c) Distribution of Pearson correlation delta for each method. d) Distribution of Pearson correlation delta for each method on each dataset. e) Distribution of cosine similarity among different methods. f) Distribution of cosine similarity for each method on each dataset. g) Distribution of MMD for each method on each dataset. h) Distribution of Wasserstein for each method. i) Distribution of Wasserstein for each method on each dataset.

Supplementary Fig. 19

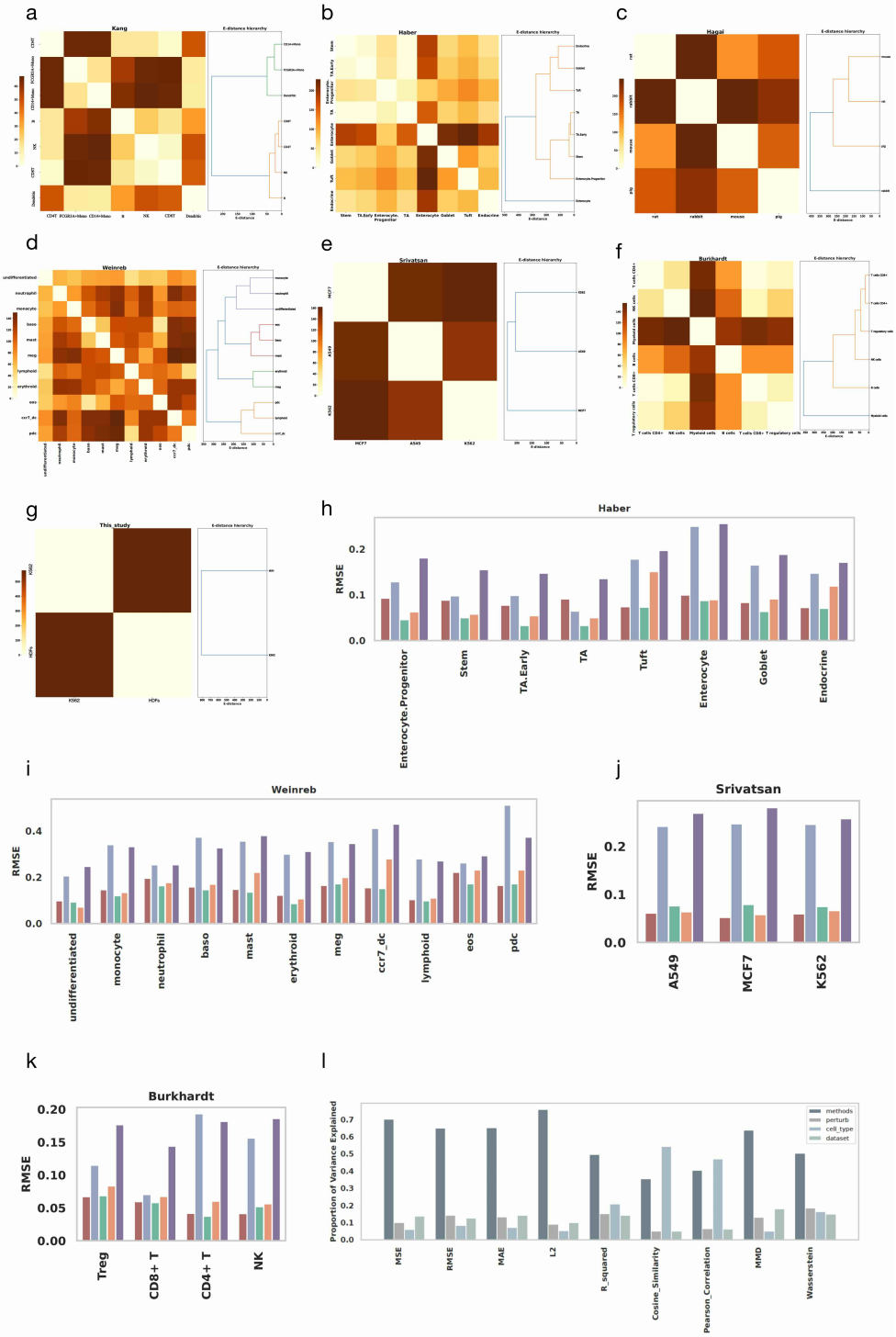

Fig. S19 This figure shows the evaluation result of task 3. a-g) The E-distances between cell types in each dataset (a: Kang, b: Haber, c: Hagai, d: Weinreb, e: Srivatsan, f:Burkhardt, g: This_study) are visualized using heatmaps and hierarchical clustering dendrograms. The heatmaps display the square E-distances between cell types based on their principal component analysis (PCA) representations. Darker colors indicate greater E-distances, reflecting less similarity between cell types, whereas lighter colors indicate smaller E-distances, reflecting greater similarity. The dendrograms illustrate the hierarchical clustering of cell types based on these distances. h-k) Model performance in predicting the responses of different cell types across datasets (h: Haber, i: Weinreb, j: Srivatsan, k: Burkhardt), evaluated using RMSE. l) A bar plot depicting the ANOVA results for each evaluation metric, with variance components decomposed into contributions from methods, perturbations, cell type and datasets.

Competing interests

Although not directly related to this paper, X.L. is a cofounder of iCamuno Biotherapeutics. The remaining authors declare no competing interests.

References

[1] Rood, J.E., Wynne, S., Robson, L., Hupalowska, A., Randell, J., Teichmann, S.A., Regev, A.: The Human Cell Atlas from a cell census to a unified foundation model. Nature (2024) https://doi.org/10.1038/s41586-024-08338-4

[2] Xu, X., Hua, X., Mo, H., Hu, S., Song, J.: Single-cell RNA sequencing to identify cellular heterogeneity and targets in cardiovascular diseases: from bench to bedside. Basic Research in Cardiology **118**(1), 7 (2023) https://doi.org/10.1007/s00395-022-00972-1

[3] 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)

[4] Datlinger, P., Rendeiro, A.F., Schmidl, C., Krausgruber, T., Traxler, P., Klughammer, J., Schuster, L.C., Kuchler, A., Alpar, D., Bock, C.: Pooled crispr screening with single-cell transcriptome readout. Nature methods **14**(3), 297–301 (2017)

[5] Duan, B., Zhou, C., Zhu, C., Yu, Y., Li, G., Zhang, S., Zhang, C., Ye, X., Ma, H., Qu, S., *et al.*: Model-based understanding of single-cell crispr screening. Nature communications **10**(1), 2233 (2019)

[6] Nelson, M.R., Tipney, H., Painter, J.L., Shen, J., Nicoletti, P., Shen, Y., Floratos, A., Sham, P.C., Li, M.J., Wang, J., *et al.*: The support of human genetic evidence for approved drug indications. Nature genetics **47**(8), 856–860 (2015)

[7] Fang, X., Lloyd, C.J., Palsson, B.O.: Reconstructing organisms in silico: genome-scale models and their emerging applications. Nature Reviews Microbiology **18**(12), 731–743 (2020)

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

[9] Bunne, C., Roohani, Y., Rosen, Y., Gupta, A., Zhang, X., Roed, M., Alexandrov, T., AlQuraishi, M., Brennan, P., Burkhardt, D.B., Califano, A., Cool, J., Dernburg, A.F., Ewing, K., Fox, E.B., Haury, M., Herr, A.E., Horvitz, E., Hsu, P.D., Jain, V., Johnson, G.R., Kalil, T., Kelley, D.R., Kelley, S.O., Kreshuk, A., Mitchison, T., Otte, S., Shendure, J., Sofroniew, N.J., Theis, F., Theodoris, C.V.,

Upadhyayula, S., Valer, M., Wang, B., Xing, E., Yeung-Levy, S., Zitnik, M., Karaletsos, T., Regev, A., Lundberg, E., Leskovec, J., Quake, S.R.: How to Build the Virtual Cell with Artificial Intelligence: Priorities and Opportunities (2024). https://arxiv.org/abs/2409.11654

[10] Quake, S.R.: The cellular dogma. Cell **187**(23), 6421–6423 (2024) https://doi.org/10.1016/j.cell.2024.10.029 . Publisher: Elsevier. Accessed 2024-12-02

[11] Fiers, M.W., Minnoye, L., Aibar, S., Bravo González-Blas, C., Kalender Atak, Z., Aerts, S.: Mapping gene regulatory networks from single-cell omics data. Briefings in functional genomics **17**(4), 246–254 (2018)

[12] Lopez, R., Gayoso, A., Yosef, N.: Enhancing scientific discoveries in molecular biology with deep generative models. Molecular systems biology **16**(9), 9198 (2020)

[13] Ji, Y., Lotfollahi, M., Wolf, F.A., Theis, F.J.: Machine learning for perturbational single-cell omics. Cell Systems **12**(6), 522–537 (2021)

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

[15] Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., Duan, N., Wang, B.: scgpt: toward building a foundation model for single-cell multi-omics using generative ai. Nature Methods, 1–11 (2024)

[16] Rood, J.E., Hupalowska, A., Regev, A.: Toward a foundation model of causal cell and tissue biology with a perturbation cell and tissue atlas. Cell **187**(17), 4520–4545 (2024)

[17] Gavriilidis, G.I., Vasileiou, V., Orfanou, A., Ishaque, N., Psomopoulos, F.: A mini-review on perturbation modelling across single-cell omic modalities. Computational and Structural Biotechnology Journal (2024)

[18] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2019). https://arxiv.org/abs/1810.04805

[19] 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)

[20] Shao, X., Yang, H., Zhuang, X., Liao, J., Yang, P., Cheng, J., Lu, X., Chen, H., Fan, X.: scdeepsort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network. Nucleic acids research **49**(21), 122–122 (2021)

[21] Hao, M., Gong, J., Zeng, X., Liu, C., Guo, Y., Cheng, X., Wang, T., Ma, J., Zhang, X., Song, L.: Large-scale foundation model on single-cell transcriptomics. Nature Methods, 1–11 (2024)

[22] Wenteler, A., Occhetta, M., Branson, N., Huebner, M., Curean, V., Dee, W.T., Connell, W.T., Hawkins-Hooker, A., Chung, S.P., Ektefaie, Y., Gallagher-Syed, A., Córdova, C.M.V.: Perteval-scfm: Benchmarking single-cell foundation models for perturbation effect prediction. bioRxiv (2024)

[23] Wu, Y., Wershof, E., Schmon, S.M., Nassar, M., Osi'nski, B.z., Eksi, R., Zhang, K., Graepel, T.: Perturbench: Benchmarking machine learning models for cellular perturbation analysis. ArXiv **abs/2408.10609** (2024)

[24] Bunne, C., Stark, S.G., Gut, G., Castillo, J.S., Levesque, M., Lehmann, K.-V., Pelkmans, L., Krause, A., Rätsch, G.: Learning single-cell perturbation responses using neural optimal transport. Nature Methods, 1–10 (2023) https://doi.org/10.1038/s41592-023-01969-x . Publisher: Nature Publishing Group. Accessed 2023-11-05

[25] Liu, T., Chen, T., Zheng, W., Luo, X., Zhao, H.: scelmo: Embeddings from language models are good learners for single-cell data analysis. bioRxiv, 2023–12 (2023)

[26] scgen predicts single-cell perturbation responses. Nature methods **16**(8), 715–721 (2019)

[27] Lotfollahi, M., Naghipourfar, M., Theis, F.J., Wolf, F.A.: Conditional out-of-distribution generation for unpaired data using transfer vae. Bioinformatics **36**(Supplement_2), 610–617 (2020)

[28] 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), 11517 (2023)

[29] Wei, X., Dong, J., Wang, F.: scpregan, a deep generative model for predicting the response of single-cell expression to perturbation. Bioinformatics **38**(13), 3377–3384 (2022)

[30] Goyal, Y., Busch, G.T., Pillai, M., Li, J., Boe, R.H., Grody, E.I., Chelvanambi, M., Dardani, I.P., Emert, B., Bodkin, N., Braun, J., Fingerman, D., Kaur, A., Jain, N., Ravindran, P.T., Mellis, I.A., Kiani, K., Alicea, G.M., Fane, M.E., Ahmed, S.S., Li, H., Chen, Y., Chai, C., Kaster, J., Witt, R.G., Lazcano, R., Ingram, D.R., Johnson, S.B., Wani, K., Dunagin, M.C., Lazar, A.J., Weeraratna, A.T., Wargo, J.A., Herlyn, M., Raj, A.: Diverse clonal fates emerge upon drug treatment of homogeneous cancer cells. Nature **620**(7974), 651–659 (2023) https://doi.org/10.

1038/s41586-023-06342-8 . Publisher: Nature Publishing Group. Accessed 2024-12-03

[31] Moon, H.C., Herschl, M.H., Pawluk, A., Konermann, S., Hsu, P.D.: A combinatorial domain screening platform reveals epigenetic effector interactions for transcriptional perturbation. bioRxiv (2024) https://doi.org/10.1101/2024.10.28.620683 https://www.biorxiv.org/content/early/2024/11/05/2024.10.28.620683.full.pdf

[32] Zheng, S., Rao, J., Zhang, J., Zhou, L., Xie, J., Cohen, E., Lu, W., Li, C., Yang, Y.: Cross-modal graph contrastive learning with cellular images. Advanced Science **11**(32), 2404845 (2024)

[33] Saunders, R.A., Allen, W.E., Pan, X., Sandhu, J., Lu, J., Lau, T.K., Smolyar, K., Sullivan, Z.A., Dulac, C., Weissman, J.S., Zhuang, X.: A platform for multimodal in vivo pooled genetic screens reveals regulators of liver function (2024) https://doi.org/10.1101/2024.11.18.624217 . Publisher: Cold Spring Harbor Laboratory Section: New Results. Accessed 2024-12-03

[34] Metzner, E., Southard, K.M., Norman, T.M.: Multiome Perturb-seq unlocks scalable discovery of integrated perturbation effects on the transcriptome and epigenome (2024) https://doi.org/10.1101/2024.07.26.605307 . Publisher: Cold Spring Harbor Laboratory Section: New Results. Accessed 2024-08-29

[35] Datlinger, P., Rendeiro, A.F., Boenke, T., Senekowitsch, M., Krausgruber, T., Barreca, D., Bock, C.: Ultra-high-throughput single-cell rna sequencing and perturbation screening with combinatorial fluidic indexing. Nature methods **18**(6), 635–642 (2021)

[36] 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)

[37] Papalexi, E., Mimitou, E.P., Butler, A.W., Foster, S., Bracken, B., Mauck III, W.M., Wessels, H.-H., Hao, Y., Yeung, B.Z., Smibert, P., *et al.*: Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens. Nature genetics **53**(3), 322–331 (2021)

[38] 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)

[39] Tian, R., Abarientos, A., Hong, J., Hashemi, S.H., Yan, R., Dräger, N., Leng, K., Nalls, M.A., Singleton, A.B., Xu, K., *et al.*: Genome-wide crispri/a screens in human neurons link lysosomal failure to ferroptosis. Nature neuroscience **24**(7),

1020–1034 (2021)

[40] Srivatsan, S.R., McFaline-Figueroa, J.L., Ramani, V., Saunders, L., Cao, J., Packer, J., Pliner, H.A., Jackson, D.L., Daza, R.M., Christiansen, L., *et al.*: Massively multiplex chemical transcriptomics at single-cell resolution. Science **367**(6473), 45–51 (2020)

[41] 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)

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

[43] Kang, H.M., Subramaniam, M., Targ, S., Nguyen, M., Maliskova, L., McCarthy, E., Wan, E., Wong, S., Byrnes, L., Lanata, C.M., *et al.*: Multiplexed droplet single-cell rna-sequencing using natural genetic variation. Nature biotechnology **36**(1), 89–94 (2018)

[44] Haber, A.L., Biton, M., Rogel, N., Herbst, R.H., Shekhar, K., Smillie, C., Burgin, G., Delorey, T.M., Howitt, M.R., Katz, Y., *et al.*: A single-cell survey of the small intestinal epithelium. Nature **551**(7680), 333–339 (2017)

[45] Hagai, T., Chen, X., Miragaia, R.J., Rostom, R., Gomes, T., Kunowska, N., Henriksson, J., Park, J.-E., Proserpio, V., Donati, G., *et al.*: Gene expression variability across cells and species shapes innate immunity. Nature **563**(7730), 197–202 (2018)

[46] Weinreb, C., Rodriguez-Fraticelli, A., Camargo, F.D., Klein, A.M.: Lineage tracing on transcriptional landscapes links state to fate during differentiation. Science **367**(6479), 3381 (2020)

[47] Liu, J.-L., Yang, L.-G., Xiao, Q.-Y., Li, Z.-Q., Cheng, J.-Y., Chen, Z.-R., Zhou, J.-W., Wan, X., Chang, T.-H., Zhang, X., Li, Y.-X.: Bio-medical Big Data Operating System (Bio-OS): An Integrated Data Mining Environment for Data Intensive Scientific Research (2024) https://doi.org/10.1101/2024.10.17.612997 . Publisher: Cold Spring Harbor Laboratory Section: New Results. Accessed 2024-12-03