nature biotechnology

Article

https://doi.org/10.1038/s41587-026-03113-4

TxPert: using multiple knowledge graphs for prediction of transcriptomic perturbation effects

Received: 16 July 2025

Accepted: 31 March 2026

Published online: 01 May 2026

Check for updates

Frederik Wenkel[1,2,4] ✉, Wilson Tu[1,2,4], Cassandra Masschelein[1,2], Hamed Shirzad[1,2], Liam Hodgson[1,2], Ihab Bendidi[1,2], Cian Eastwood[1,2], Shawn T. Whitfield[1,2], Craig Russell[1,2], Yassir El Mesbahi[1,2], Jiarui Ding[3], Marta M. Fay[2], Berton Earnshaw[1,2], Emmanuel Noutahi[1,2] & Alisandra K. Denton[1,2] ✉

Accurately predicting cellular responses to genetic perturbations is essential for understanding disease mechanisms and designing effective therapies. Yet, exhaustively exploring the space of possible perturbations (for example, multigene perturbations or across tissues and cell types) is prohibitively expensive, motivating methods that can generalize to unseen conditions. We present TxPert, a latent-transfer-based deep learning method that uses multiple knowledge graphs of gene (product)–gene (product) relationships to predict transcriptomic perturbation effects. Different knowledge graphs encode complementary information and we show that a combination of graphs derived from biological databases and high-throughput perturbation screens yields the best performance. For predictions of single unseen perturbations, TxPert approaches the performance of split-half experimental reproducibility. For double unseen perturbations and single perturbations in a different cell line, its predictions increase Person Δ for unseen single perturbations by 8–25% over existing methods.

To account for biological complexity, candidate therapeutics are tested across a range of assays in diverse cellular contexts before advancing to model systems such as organoids, xenografts, animals and ultimately human clinical trials. Yet, the vast majority of these candidate therapeutics are unsuccessful, often failing late in development after considerable costs have been incurred. At its core, a therapeutic is a perturbation intended to shift a cell's state, generally from diseased to healthy. Finding the perturbation that will produce the desired effect is central to drug discovery. There is a growing need for computational models that can simulate the effects of perturbation in silico in out-of-distribution (OOD) settings, allowing for highly targeted confirmatory (instead of exploratory) wet lab screens. Such models would reduce the need for exhaustive screening, enable principled extrapolation across cell types and conditions and accelerate the design of effective therapeutic interventions.

Two complementary strategies have emerged to predict perturbation effects in a generalizable OOD setting. The first strategy exploits the inherent compositional nature of cellular responses by training deep generative models to learn latent representations that can be perturbed, enabling in silico predictions across contexts[1,2]. The second uses prior biological knowledge as inductive bias, for example, by incorporating biochemical entity relationship graphs or embedded functional annotations[3,4].

While machine learning models have made major strides in some fields of biology such as protein structure and interaction prediction[5,6], deep models for transcriptomics-focused perturbation biology have often underperformed, sometimes trailing simple or even untrained baselines[7–10]. Notably, a previous study[11] showed that the training set mean or median of all perturbation responses is a better predictor of

[1]Valence Labs, Montréal, Quebec, Canada. [2]Recursion, Salt Lake City, UT, USA. [3]Computer Science, University of British Columbia, Vancouver, British Columbia, Canada. [4]These authors contributed equally: Frederik Wenkel, Wilson Tu. ✉e-mail: frederik@valencelabs.com; ali@valencelabs.com

an individual perturbation response than the outputs from prominent models such as GEARS[4]. In short, the potential of deep learning models in this domain remains an open research question.

In this work, we introduce TxPert as a unifying model for transcriptomics perturbation effect prediction that (1) can be trained broadly across datasets; (2) supports three major OOD tasks in a single framework; and (3) effectively uses prior knowledge and biological context without requiring dataset-specific optimization. In particular, we show that TxPert achieves state-of-the-art performance in predicting unseen single perturbation effects within cell lines, while showing highly competitive performance in predicting the effects of double-gene perturbations and in cross-context generalization where no perturbations have been observed in the test cell line.

Beyond TxPert, we present a modular and extendable training and evaluation framework for transcriptomic perturbation prediction that advances best practices with rigorous benchmarking. Specifically, we introduce batch-appropriate control matching and the recently introduced evaluation task, retrieval[10,12]. Lastly, we contextualize the model's performance with comparisons to multiple baselines, published models and an estimate of split-half experimental reproducibility.

Results

Revisiting metric design for biologically grounded modeling

Modeling of transcriptomic cell profiles after a perturbation is a fundamentally nascent field with many open questions and a notable absence of clear consensus on best practice for data handling and evaluation. Given this uncertainty, we first conducted a theory-informed and data-driven investigation to better understand the data being modeled and thus maximize TxPert's biological impact.

Batch-matched controls are warranted given batch effects and confounding. Experimental batch effects are a well-known challenge in biological data[13,14] where states vary because of the sensitivity and variability of biological systems and their interaction with the environment. To account for this variability, primary experimental studies include carefully designed controls in every batch. However, many deep learning models[4] rely on a global control mean to compute perturbation effects (Δ expression metrics; Methods). In a scenario with both batch effects and confounding between batch and applied perturbation, failure to account for the batch effects can result in mistaking background batch-wise variance for perturbation effects and overestimating model performance, reduce true performance by adding avoidable noise or both. We confirmed that there are substantial batch effects by analyzing the genome-wide Perturb-seq dataset from a previous study[13], where we quantified control–control correlations both within and across batches. Across-batch correlations were significantly lower (U-test, $P = 1.4 \times 10^{-19}$) despite involving more aggregated cells (Fig. 1a). Subsequently, we checked for confounding effects between batch and perturbation. As a library of perturbants is applied to pooled cells in Perturb-seq and transfection, survival and sampling are stochastic processes, confounding cannot be strictly controlled but it can be measured. We found significant association between batch and perturbation ID in every dataset used here (χ^2, 1.1×10^{-4} to $1.0 \times {}^{-98}$ across datasets). To maintain biological validity and reduce noise, given the observed batch effects and confounding, we adopted batch-matched controls for all subsequent model training and evaluation.

Performance evaluation with retrieval metrics and Pearson Δ. Prior studies showed that mean baseline models (averaging perturbation effects across many perturbations) achieve surprisingly strong predictive performance[8,11]. We found that this systematic effect is especially prominent in perturbations of 'essential' genes (Fig. 1b). Moreover, the systematic effect was consistent across diverse datasets and perturbation modalities, including the data from a previous study[15] that used CRISPR activation (CRISPRa) to upregulate target genes, in contrast to the other studies using CRISPR interference (CRISPRi) to downregulate targets (Fig. 1c). Further investigating the genes changing with the systemic effect, we found the functional terms 'vacuole', 'autophagosome membrane' and 'lysosome' among upregulated genes and functions related to cell division and various metabolic processes among downregulated genes (Supplementary Tables 1 and 2). This is consistent with the idea that a perturbed cell undergoes general stress and shifts from growth and uptake to quiescence and recycling. In short, the strong predictive power of the mean baseline is a feature of the data. It reflects general biological responses to perturbation-induced stress or a reduction in fitness, health or growth because of perturbation of an important cellular component, rather than perturbation-specific effects.

Given these findings, we adopted a complementary evaluation approach using retrieval metrics, which measure how effectively a model distinguishes replicate perturbations from other perturbations[10,12]. Retrieval performance depends on both (1) the representation of the perturbation effects and (2) the similarity metric chosen; the choices of these factors vary considerably in the field. Through systematic benchmarking, we found that cosine similarity and Pearson correlation applied directly to Δ profiles yielded the highest retrieval scores (Fig. 1d). In contrast, common practices such as selecting only the top differentially expressed genes substantially reduced retrieval performance compared to unfiltered Pearson Δ (Supplementary Fig. 1). Consequently, we selected Pearson Δ as our primary evaluation metric, complemented by retrieval scores during final testing to holistically assess perturbation-specific signal retention.

TxPert: a transcriptomic perturbation effect prediction framework for OOD tasks

We introduce TxPert, a deep learning framework designed for robust prediction of transcriptional responses to previously unobserved genetic perturbations, including single-gene perturbations, combinations of perturbations and perturbations across new cell types. Inspired by previous efforts[2,3], TxPert relies on latent transfer to achieve strong generalization performance. Specifically, it integrates two complementary modules: a basal state encoder that learns an embedding of the cell without perturbation and a perturbation encoder that learns a representation of perturbation(s) by leveraging informative embeddings from gene interaction networks (Fig. 2). These embeddings are combined (latent transfer) and decoded to predict the resulting log-transformed gene expression profile of perturbed cells.

Through extensive exploration of candidate methods for each module across multiple datasets, we identified architectural choices that effectively use existing biological knowledge for generalizable perturbation prediction. For the basal state model, we explored encoding gene expression using (1) a multilayer perceptron (MLP) and (2) pretrained embeddings from established foundation models. For the perturbation model, we investigated (1) major graph neural network (GNN) architectures (graph attention networks (GATs) and graph transformers (GTs)) and (2) hybrid and multilayer GNN variants that combine different gene interaction graphs (Methods).

While all proposed GNN variants demonstrate strong performance across the explored OOD tasks, we tuned each architecture individually and report the best model variant per task in the main text. We provide further information on model choice after discussing architecture-specific details. For the prediction of unseen single perturbations, we ultimately selected the Exphormer-MG[16,17] GT architecture, which enabled us to simultaneously integrate four complementary graph sources: STRING[18], GO[19], PxMap and TxMap. PxMap and TxMap are proprietary recursion relationship datasets derived from large-scale phenomics (microscopy imaging[20]) and single-cell transcriptomics perturbation screens, respectively[14,21]. Similarly, for predicting double perturbations, GAT-MLG[22,23] achieved the best results, leveraging the complementary information from GO, PxMap

https://doi.org/10.1038/s41587-026-03113-4

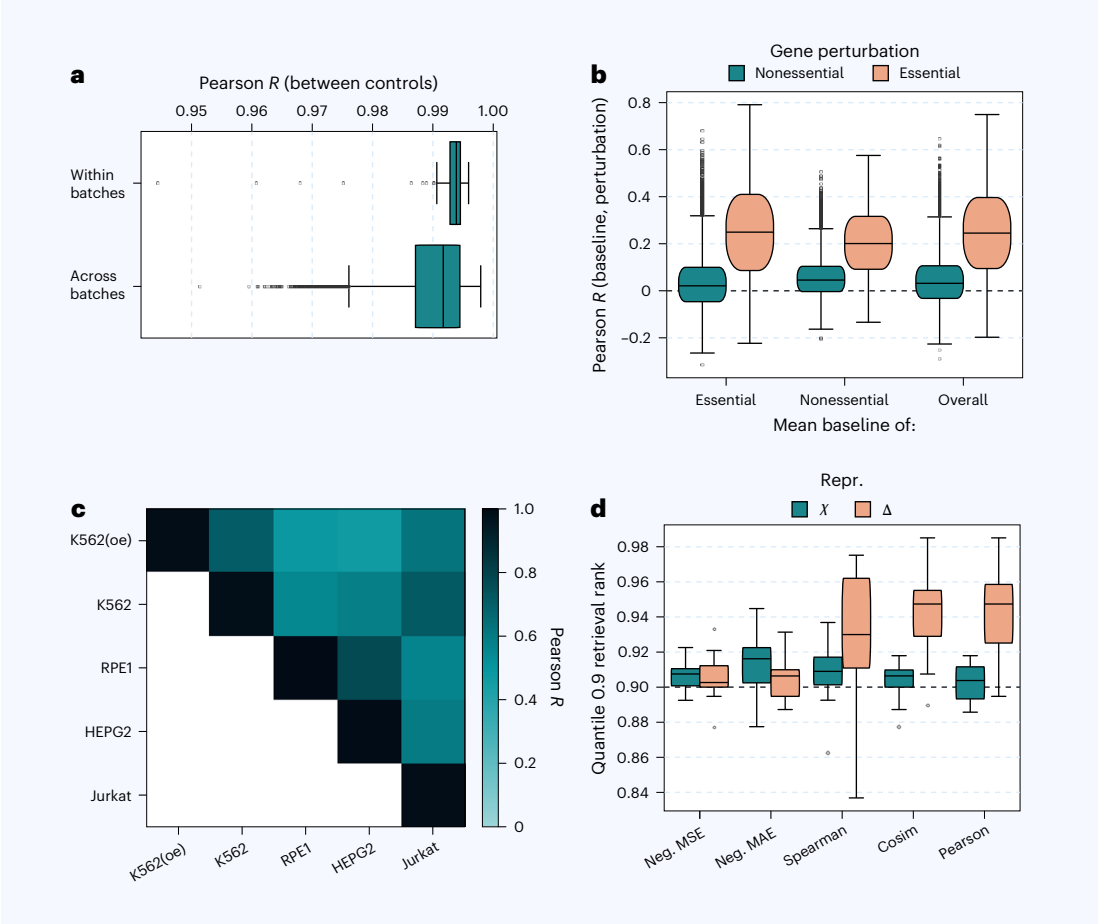

Fig. 1 | Dataset statistics. a, Pearson correlation of aggregated control gene expression profiles within and across experimental batches ($n = 267$ and 3,511, respectively). **b**, Correlation between single perturbations and the mean baseline, that is, the mean Δ calculated over aggregates of essential (as defined previously[13]), nonessential or all genes ($n = 2{,}058$, 7,815 and 9,866, respectively). **c**, Correlation between the mean baseline aggregated (across $n \geq 118$ perturbations) within or between studies and cell types. All data are CRISPRi unless marked as overexpression (oe) data from a previous study[15]. **d**, Normalized retrieval between true perturbant replicates in different biological contexts. Retrieval is calculated on the basis of the indicated expression representations and metrics ($n = 18$). The plotted value is the 0.9 quantile (across all unique perturbants), where expected random performance is 0.9, indicated by the dashed line. MAE, mean absolute error.

and TxMap. Regardless of the specific architectural choices, our proposed framework consistently outperforms simpler learned baselines, existing methods such as GEARS and scLAMBDA, and a statistical general baseline, which predicts perturbation effects on the basis of the strongest combination of additive and mean as appropriate for the task (Methods). This superior performance is observed across various challenging OOD prediction tasks, demonstrating the framework's robustness and versatility.

TxPert substantially outperforms other models at predicting unseen perturbation effects

Predicting the transcriptional response to an unobserved perturbation in a known cell type relies on contextual information, including perturbant-specific and biomolecular interaction signals learned from perturbations observed during training. In this setting, we focus on four well-studied cell lines, each with extensive perturbation data (more than 2,000 perturbation types per cell line): myelogenous leukemia lymphoblasts (K562), retinal pigment epithelial cells (RPE1), liver hepatocellular carcinoma cells (HepG2) and human T lymphocytes (Jurkat)[13,24]. We trained TxPert and existing baseline methods on data from each cell line separately, while leaving out specific perturbations for evaluation. We compare our models results to two existing methods for leveraging biological data. The first, GEARS[4], integrates prior biological knowledge by embedding a gene–gene relationship network derived from Gene Ontology (GO) terms with a GNN architecture and embedding basal state with a coexpression-derived graph and GNN. The second, scLAMBDA, uses inductive biases derived from external textual data. Specifically, it uses genePT—embeddings[25] generated by applying GPT-3.5 to the functional summaries available from the National Center for Biotechnology Information gene database[3]. As illustrated in Fig. 3a, TxPert uniformly outperforms scLAMBDA, GEARS and the general baseline, with GEARS falling below the nonlearned general baseline. Our model is competitive with split-half experimental reproducibility in three of the four cell lines (K562, Jurkat and HepG2). For context, the split-half experimental reproducibility represents the reproducibility achievable by splitting the test set replicates, grouped by batch, in half and comparing the halves. Note that, because of the reduced sample counts, this is not an upper bound but serves as a rigorous benchmark comparable to human-level performance in vision tasks. Using an alternative approach of sampling from a probabilistic model fitted to the test set, we estimated Pearson Δ values from 0.07 to 0.11 higher for an original-sized test set, depending on the dataset (Supplementary Table 4). Retrieval metrics revealed similar relative performance patterns between models, with TxPert again outperforming other methods. However, the gap between split-half experimental reproducibility and TxPert was much larger when measured by retrieval, while the general baseline had the lowest retrieval (Supplementary Fig. 2).

https://doi.org/10.1038/s41587-026-03113-4

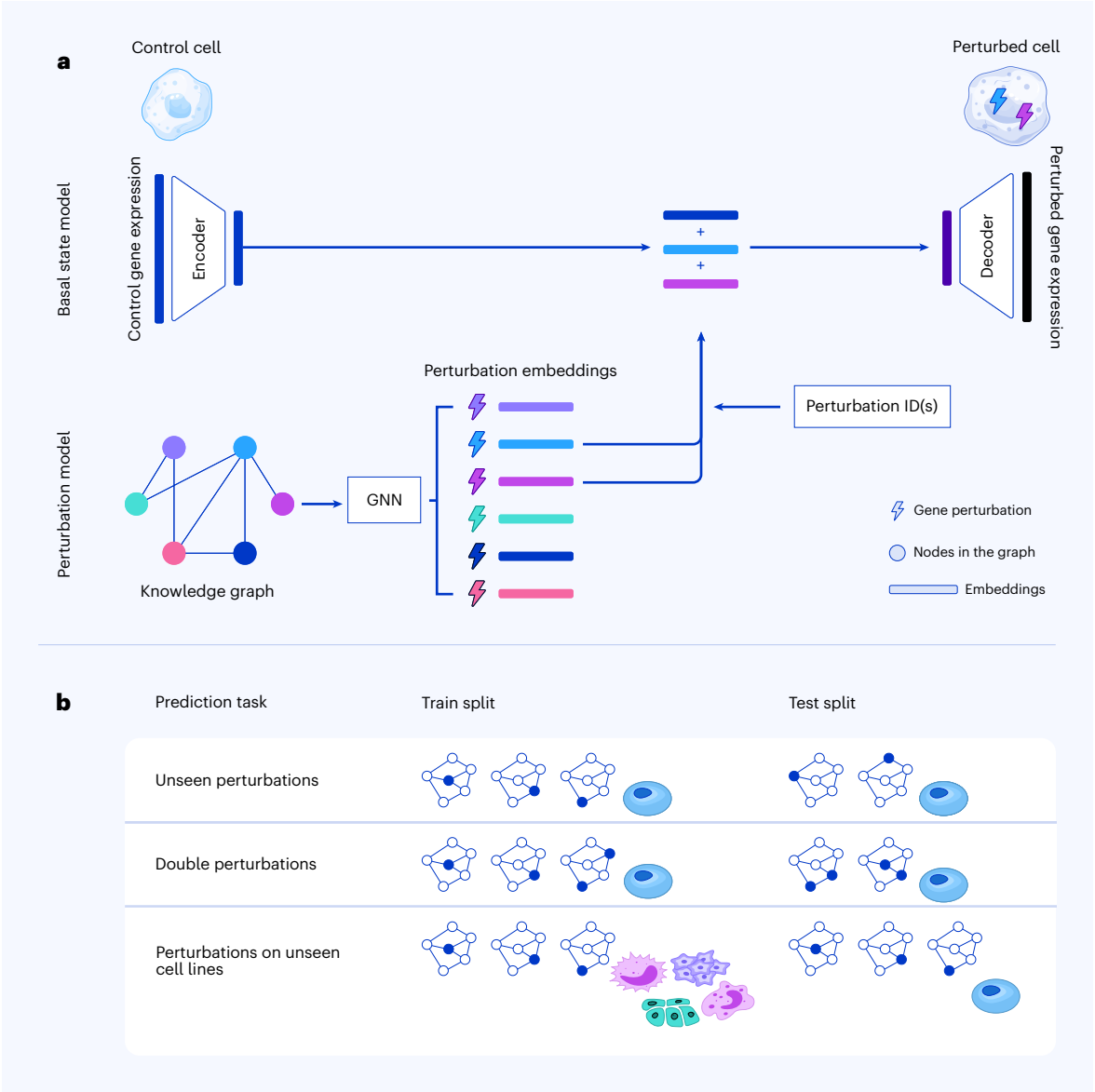

Fig. 2 | TxPert model and task overview. a, The TxPert architecture predicts postperturbation gene expression by combining two modules: (1) a basal state encoder that maps batch-matched control profiles into a latent embedding and (2) a GNN that learns perturbation embeddings from a gene (product)–gene (product) interaction graph. Perturbation embeddings are applied to the basal embedding, and the resulting latent representation is decoded to produce the predicted gene expression profile. **b**, OOD perturbation effect prediction tasks for (1) unseen single perturbations within the training cell line; (2) novel double perturbations, where constituent singles may have been seen during training, within the training cell line; and (3) perturbations within new cell lines not seen during training.

TxPert outperforms existing models, as well as the additive baseline, in predicting the effect of multigene perturbations

Although genome-wide single perturbation datasets are becoming increasingly available, combinatorial perturbation experiments remain prohibitively costly. We compared our model's performance in predicting the effect of double perturbations using the Norman dataset[15], specifically focusing on cases where both individual perturbations were previously seen in isolation during training. At this task, TxPert achieves a slightly higher Pearson Δ than the additive baseline (a specific case of the proposed general baseline, where the expression profile of the unseen double (i,j) is predicted as $\hat{\mathbf{y}}_{(i,j)} = \bar{\mathbf{x}} + \delta_i + \delta_j$, with an appropriate mean estimate of the control in the test set $\bar{\mathbf{x}}$, for example, aligned with respect to cell line and batch effect of the target) with a substantial lead over GEARS and scLAMBDA (Fig. 3b and Supplementary Fig. 3).

TxPert generalizes effectively to predict perturbation effects across cell lines without seen perturbations

The third task is predicting the effect of a seen perturbation in a new biological context (here, a new cell line), which represents a critical test of how generalizable a model can be, as substantial perturbation data exist only for a small fraction of cellular contexts. We performed four leave-one-out experiments, where we held out all perturbation examples from the target cell type but trained on all controls. Neither GEARS nor scLAMBDA originally included in this task; however, we adapted scLAMBDA's implementation to provide a relevant baseline.

We found that TxPert exceeded the general baseline in all four held-out cell lines and consistently outperformed scLAMBDA (Fig. 3c and Supplementary Fig. 4). TxPert exceeded a more challenging nearest-cell-line baseline in two cell lines (K562 and RPE1) while nearly matching its performance in HepG2.

https://doi.org/10.1038/s41587-026-03113-4

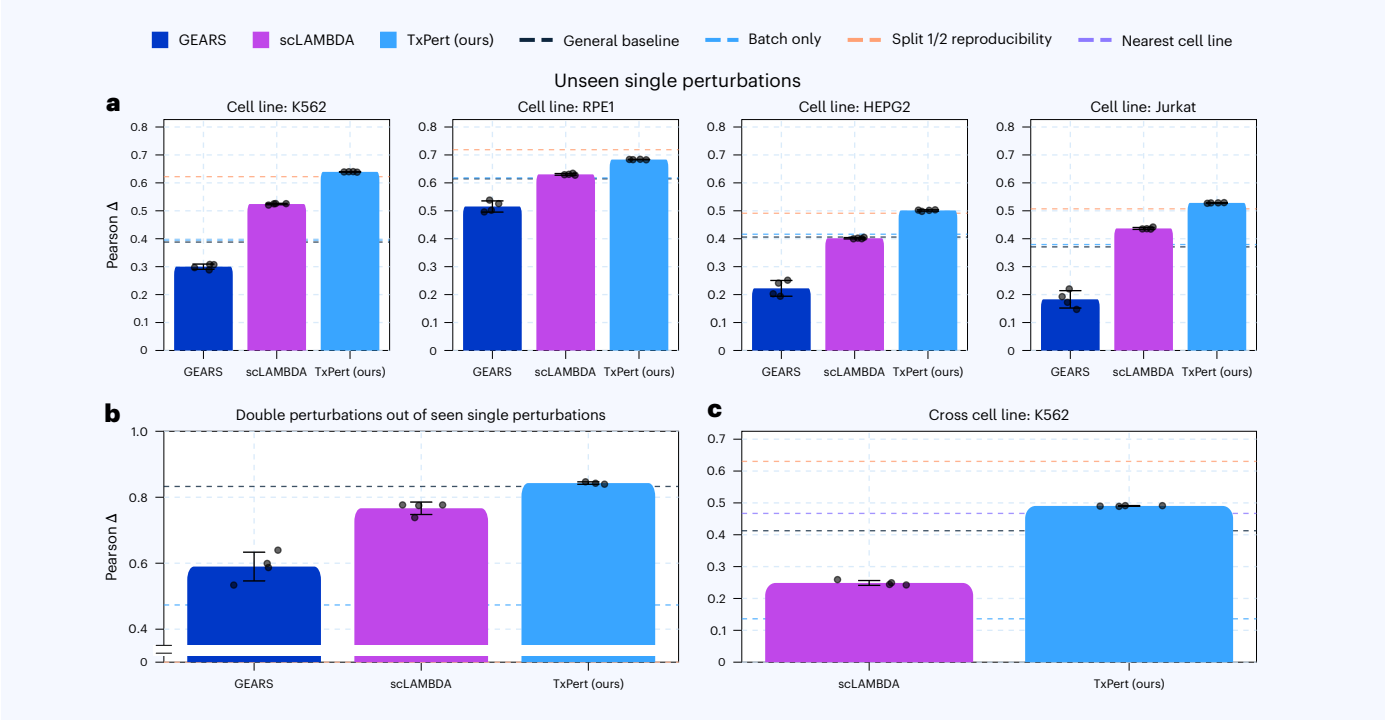

Fig. 3 | Performance comparison across different generalization tasks. We report the mean ± s.d. across multiple training seeds. **a**, Performance of TxPert compared to GEARS and scLAMBDA on predicting unseen single perturbations within a known cell type. Horizontal bars indicate the general baseline, a batch-informed model (capturing potential confounding) and split-half experimental reproducibility, as well as a nearest-cell-line baseline (cross-cell-line task only) (Methods). **b**, Comparison of models in predicting double perturbation effects from known singles. **c**, Comparison of model performance on cross-context generalization of single perturbation effect prediction. For all plots, error bars show the mean and s.d. of scores through training with $n = 4$ different seeds.

Collectively, these results serve as a proof of concept, demonstrating that TxPert can successfully address diverse OOD perturbation prediction tasks. It excels particularly in predicting unseen perturbations, while setting a competitive mark in cross-cell line predictions.

TxPert learns meaningful information from biological knowledge graphs (KGs)

As no definitive, universally accepted biological interaction graph exists, we first empirically compared several graph sources for their utility as predictive priors. We evaluated two curated database-derived graphs (STRING and GO) alongside two graphs derived from genome-wide perturbational screens (PxMap and TxMap). Among these alternatives, the Exphormer configuration of TxPert performed best with the STRING graph (an accumulation of biological knowledge from database, literature and multiple raw sources) followed by the PxMap (derived entirely from high-throughput perturbational screening in primary human umbilical vein endothelial cells (HUVECs)) (Fig. 4b). Next, to confirm the graph's utility, we evaluated the impact of progressively degrading its structure. We randomly rewired the original STRING graph (by randomly changing the source, target or both nodes of each edge) from 0% (original graph) up to 100%, retraining and assessing model performance independently. As the proportion of randomized edges increased, we observed a consistent decline in the test set Pearson Δ for the K562 dataset (Fig. 4a and Supplementary Fig. 5). Similarly, performance was also sensitive to extreme random downsampling (that is, removing) of edges, although performance remained robust until more than 60% of edges were removed (Supplementary Fig. 5), dependent on GNN depth (Supplementary Fig. 6 and Supplementary Note 4). Together, these findings indicate that incorporating accurate biological graph information substantially enhances model performance.

TxPert performance increases when combining multiple KGs

We hypothesized that different biological graphs might provide complementary information and that their combination could yield improved prediction. To this end, we explored four different strategies for integrating multiple graphs: (1) GAT-Hybrid, an extension of the GATv2 model designed to learn from several KGs simultaneously and subsequently combine their information; (2) Exphormer-MG, an extension of the GT architecture adapted for multigraph learning using a union graph methodology; (3) GAT-MLG (multilayered graph), a method that adapts GATv2 to operate on a unified supra-adjacency representation of multiple KGs, enabling message passing across both intralayer and interlayer connections simultaneously; and (4) Hybrid-BMP (bidirectional message passing), a variant of a one layer message passing model that can use both incoming and outgoing edges of a union of adjacency matrices. Detailed descriptions of these models are provided in Supplementary Note 3. For simplicity, we focused on predicting unseen perturbations in K562 cells where Hybrid-BMP achieved the best performance (Fig. 4c).

Moreover, incremental integration of multiple graphs (through Exphormer-MG) starting from STRING consistently improved predictive performance, peaking when all four graphs (STRING, GO, PxMap and TxMap) were combined (t-test, assuming normal distribution, $P < 0.027$ when comparing the best three-source graph versus four-source graph) (Fig. 4d).

Detailed evaluation of model performance

To understand our model's performance in depth, we undertook a detailed analysis and scrutinized factors that could relate to model performance.

First, as our model is leveraging prior information, which we know to some degree is both incomplete and biased, we checked for a relation between how well a gene target is known and performance in the

https://doi.org/10.1038/s41587-026-03113-4

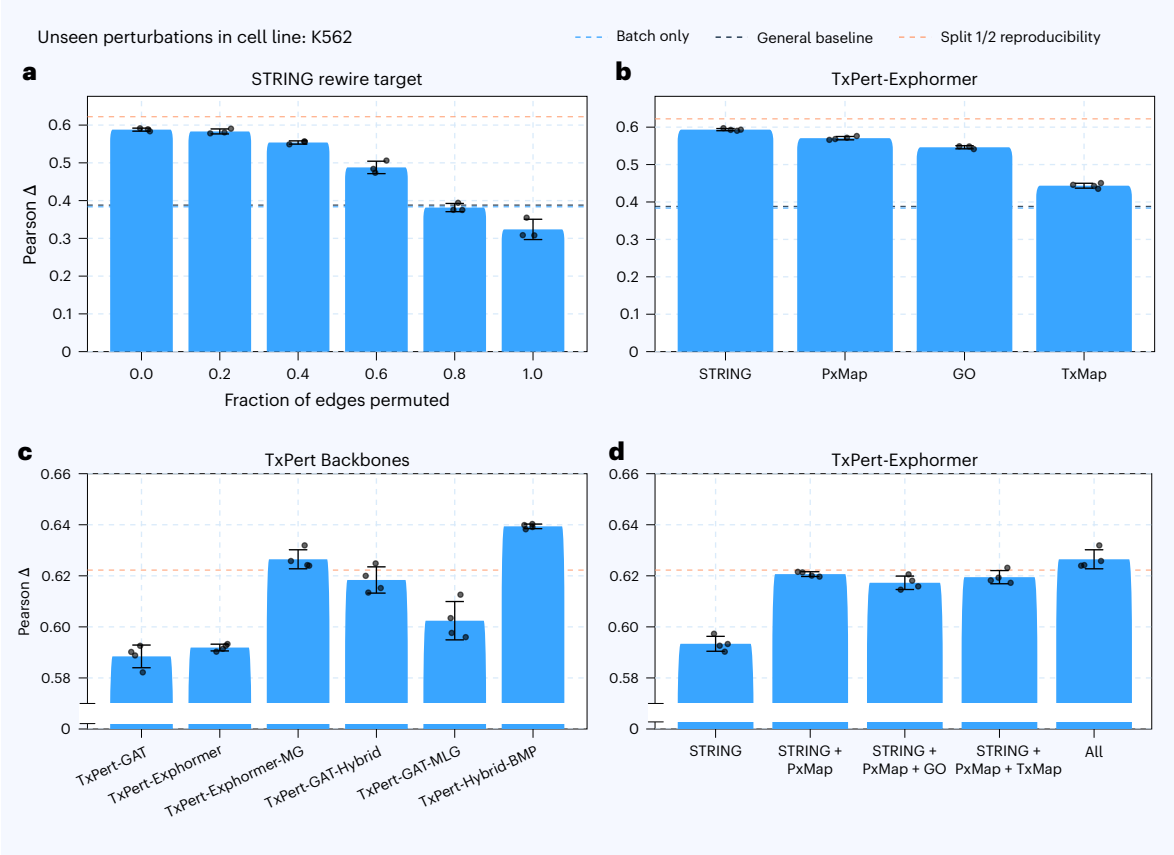

Fig. 4 | Ablation studies for unseen perturbation effect prediction on K562. We report the mean ± s.d. across multiple training seeds. **a**, Performance of TxPert as edges of the STRING graph are progressively rewired. **b**, Performance of TxPert (Exphormer) using individual graphs. **c**, Comparison of various graph integration strategies and architectures. **d**, Performance of TxPert (Exphormer) as multiple KGs (STRING, GO, PxMap and TxMap) are subsequently integrated into the Exphormer architecture (Exphormer-MG). Horizontal bars indicate the general baseline performance, the performance of a learned model making predictions on the basis of batch information (in case of confounding between batch and perturbation) and split-half reproducibility. For all plots, error bars show the mean and s.d. of scores through training with $n = 4$ different seeds.

Hybrid-BMP model. We binned perturbed genes into five knowledge levels by their Pharos knowledge rank[26] (Methods). Unsurprisingly, we see an association between predictive performance and the knowledge of the pertubation target. However, this pattern is also present in general baseline predictions, indicating that a portion of this pattern is driven by perturbant-intrinsic factors such as perturbation effect size. We hypothesized that the perturbation-effect-derived 'maps' would show less bias and encouragingly found that a hybrid STRING + PxMap graph outscored STRING alone across all knowledge levels (Fig. 5a). Overall, both data-intrinsic factors (such as effect size) and biological knowledge factors correlate with model performance (Supplementary Note 5 and Fig. 5b).

Running functional enrichment on the perturbations with the highest and lowest Pearson Δ scores, we found strong enrichments in what our model excelled at, such as protein translation and localization, whereas there were no significant enrichments ($P > 0.05$) in the perturbants our model performed worst on (Supplementary Table 3). While otherwise robust, we did identify one specific failure mode of our model for unseen predictions, namely that the architecture and training strategy do not allow the model to learn the typical downregulation of the unseen perturbation target itself (Fig. 5c,d). While this error is not inconsequential for understanding the model's abilities, we note that many expression forecasting methods assign this value to be equal to the ground-truth value[11]. This analysis into strengths and weaknesses helps increases the applicability of TxPert, by providing early insight into when predictions can be most trusted.

Discussion

The past year has brought a reality check to the promise of foundation models in the transcriptomics perturbation domain, as independent benchmarking studies failed to validate the claimed performance of several high-profile models[7–11,27,28]. In this work, we addressed these concerns through rigorous benchmarking and inclusion of strong baselines, resulting in TxPert, a broadly applicable perturbation model competitive with the rigorous performance level derived from split-half experimental reproducibility on some metrics on unseen perturbations across several datasets. A key factor underpinning our model's success is the effective integration of curated biological databases with large-scale, consistent and unbiased high-throughput screening data combined with first-class graph modeling.

Our reusable framework establishes a strong foundation for iteration and improvement and is positioned to benefit further from new developments in the field. For instance, recently released large perturbation single-cell datasets[29–31] could help improve cross-context generalization. Extending our framework toward few-shot or active learning scenarios is another realistic and promising direction to expand beyond the zero-shot cross-cell-line setting explored here. To generalize to human primary tissues, it will be critical to increase the diversity of the data and train on more than (largely cancer-derived) immortalized cell lines. Crucially, the field can accelerate progress by continuing to adopt, iteratively refine and standardize both task definitions and benchmarks. A key next step in improving further TxPert or other high-performing models lies in the inclusion of metrics that explicitly evaluate the conditionality and specificity of perturbation effects in novel contexts.

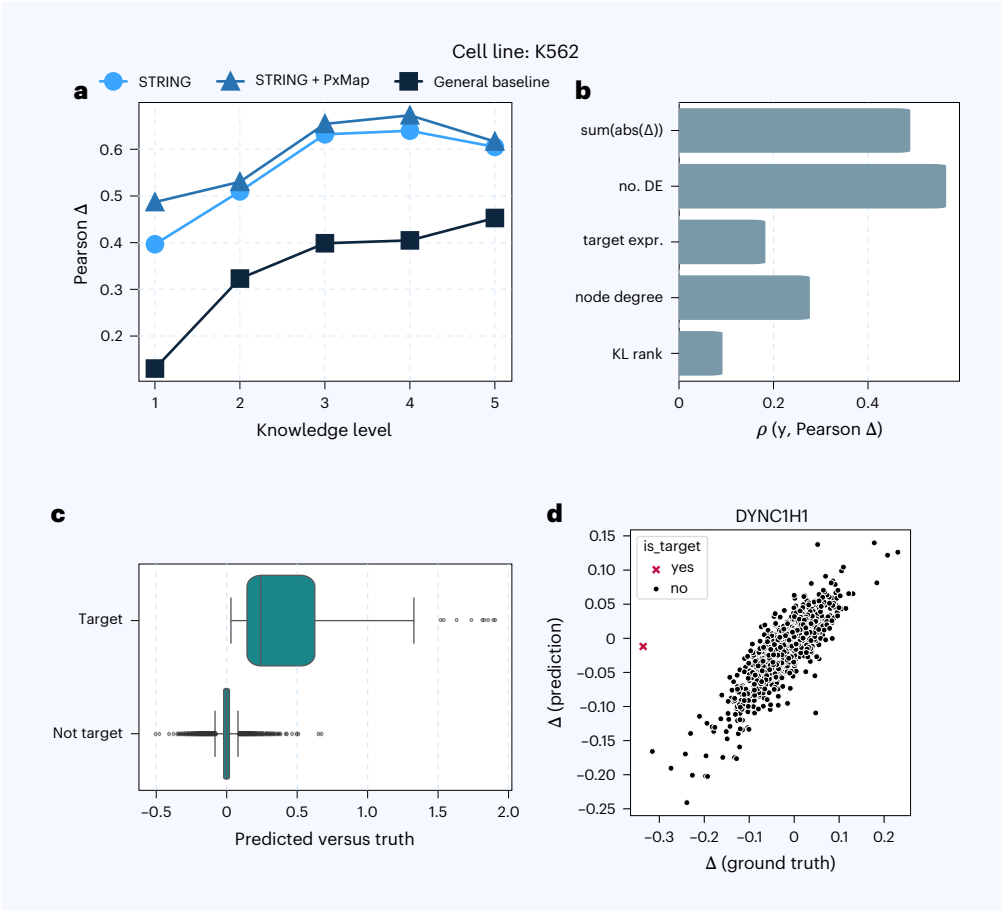

Fig. 5 | Investigation into strengths and weaknesses of our models. a, Breakdown of Pearson *Δ* by the knowledge level (Pharos rank) of the perturbation target gene. **b**, Spearman correlation between performance (Pearson *Δ*) and both data-intrinsic factors (number of differentially expressed genes, sum of absolute *Δ* values and control expression level of the target gene) and biological knowledge factors (degree of perturbed node in graph, Pharos knowledge level) metadata for unique perturbation, which were hypothesized to be related to performance. **c**, Signed error in predicting the expression of genes, when these either are or are not targets ($n = 107$ and 28,997). **d**, Example prediction versus ground truth for all genes when dynein cytoplasmic 1 heavy chain 1 (*DYNC1H1*, ENSG00000197102) is perturbed, showing the target, *DYNC1H1* mRNA, in red. *DYNC1H1* was chosen as an arbitrary but representative example demonstrating the common failure to predict the true downregulation of the perturbation target's mRNA. All analyses on the test set for K562 (ref. 13) were performed for the unseen perturbant task.

Although TxPert exhibits strong performance, it represents just an initial step toward developing a new generation of models able to accurately model cellular responses to perturbations across diverse biological context. As the community (1) investigates the availability and importance of prior knowledge for modeling[32] and (2) enables further components, such as compound predictions[33,34], genomic sequence conditioning[35], spatial or intercellular interactions[36] and distributional predictions[37,38], the utility of virtual assays for therapeutic applications will grow. Ultimately, this has the potential to accelerate drug discovery programs, enable a completely new scope in screening and open new frontiers for personalized medicine.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41587-026-03113-4.

References

1. Lotfollahi, M., Wolf, F. A. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nat. Methods* **16**, 715–721 (2019).
2. Lotfollahi, M. et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Mol. Syst. Biol.* **19**, e11517 (2023).
3. Wang, G., Liu, T., Zhao, J., Cheng, Y. & Zhao, H. Modeling and predicting single-cell multi-gene perturbation responses with sclambda. Preprint at bioRxiv https://doi.org/10.1101/2024.12.04.626878 (2024).
4. Roohani, Y., Huang, K. & Leskovec, J. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nat. Biotechnol.* **42**, 927–935 (2024).
5. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
6. Passaro, S. et al. Boltz-2: towards accurate and efficient binding affinity prediction. Preprint at *bioRxiv* https://doi.org/10.1101/2025.06.14.659707 (2025).
7. Bendidi, I. et al. Benchmarking transcriptomics foundation models for perturbation analysis: one PCA still rules them all. Preprint at https://doi.org/10.48550/arXiv.2410.13956 (2024).
8. Wong, D. R., Hill, A. S. & Moccia, R. Simple controls exceed best deep learning algorithms and reveal foundation model effectiveness for predicting genetic perturbations. *Bioinformatics* **41**, btaf317 (2025).

9. Ahlmann-Eltze, C., Huber, W. & Anders, S. Deep learning-based predictions of gene perturbation effects do not yet outperform simple linear methods. *Nat. Methods* **22**, 1657–1661 (2025).
10. Wu, Y. et al. PerturBench: Benchmarking machine learning models for cellular perturbation analysis. Preprint at https://doi.org/10.48550/arXiv.2408.10609 (2025).
11. Kernfeld, E., Yang, Y., Weinstock, J. S., Battle, A. & Cahan, P. A comparison of computational methods for expression forecasting. *Genome Biol.* **26**, 388 (2025).
12. Szałata, A. et al. A benchmark for prediction of transcriptomic responses to chemical perturbations across cell types. In *Proceedings of the 38th International Conference on Neural Information Processing System* (eds Globerson, A. et al.) (NIPS, 2025).
13. Replogle, J. M. et al. Mapping information-rich genotype–phenotype landscapes with genome-scale Perturb-seq. *Cell* **185**, 2559–2575 (2022).
14. Celik, S. et al. Building, benchmarking, and exploring perturbative maps of transcriptional and morphological data. *PLOS Comput. Biol.* **20**, e1012463 (2024).
15. Norman, T. M. et al. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science* **365**, 786–793 (2019).
16. Shirzad, H., Velingker, A., Venkatachalam, B., Sutherland, D. J. & Sinop, A. K. Exphormer: sparse transformers for graphs. In *Proceedings of the 40th International Conference on Machine Learning* (eds Krause, A. et al.) (PMLR, 2023).
17. Shirzad, H. et al. Even sparser graph transformers. In *Proceedings of the 38th International Conference on Neural Information Processing System* (eds Globerson, A. et al.) (NIPS, 2025).
18. Szklarczyk, D. et al. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.* **47**, D607–D613 (2019).
19. Consortium, G. O. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res.* **32**, D258–D261 (2004).
20. Bray, M.-A. et al. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nat. Protoc.* **11**, 1757–1774 (2016).
21. Kraus, O. et al. Masked autoencoders for microscopy ar e scalable learners of cellular biology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (ed. Ceballos, C.) (IEEE, 2024).
22. Zangari, L., Mandaglio, D. & Tagarelli, A. Link prediction on multilayer networks through learning of within-layer and across-layer node-pair structural features and node embedding similarity. In *Proceedings of the ACM Web Conference* (eds Chua, T.-S. & Ngo, C.-W.) (ACM, 2024).
23. Yun, S., Kim, S., Lee, J., Kang, J. & Kim, H. J. Neo-GNNs: neighborhood overlap-aware graph neural networks for link prediction. In *Proceedings of the 35th International Conference on Neural Information Processing Systems* (eds Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. S., Wortman Vaughan, J.) (NIPS, 2021).
24. Nadig, A. et al. Transcriptome-wide analysis of differential expression in perturbation atlases. *Nat. Genet.* **57**, 1228–1237 (2025).
25. Chen, Y. T. & Zou, J. GenePT: a simple but hard-to-beat foundation model for genes and cells built from chatgpt. Preprint at *bioRxiv* https://doi.org/10.1101/2023.10.16.562533 (2023).
26. Sheils, T. K. et al. TCRD and Pharos 2021: mining the human proteome for disease biology. *Nucleic Acids Res.* **49**, D1334–D1346 (2021).
27. Kedzierska, K. Z., Crawford, L., Amini, A. P. & Lu, A. X. Zero-shot evaluation reveals limitations of single-cell foundation models. *Genome Biol.* **26**, 101 (2025).
28. Csendes, G., Sanz, G., Szalay, K. Z. & Szalai, B. Benchmarking foundation cell models for post-perturbation RNA-seq prediction. *BMC Genomics* **26**, 393 (2025).
29. Feng, C. et al. A genome-scale single-cell CRISPRi map of *trans* gene regulation across human pluripotent stem cell lines. *Cell Genom.* **6**, 101076 (2026).
30. Zhang, J. et al. Tahoe-100M: a giga-scale single-cell perturbation atlas for context-dependent gene function and cellular modeling. Preprint at *bioRxiv* https://doi.org/10.1101/2025.02.20.639398 (2025).
31. Huang, A. C. et al. X-Atlas/Orion: genome-wide Perturb-seq datasets via a scalable fix-cryopreserve platform for training dose-dependent biological foundation models. Preprint at *bioRxiv* https://doi.org/10.1101/2025.06.11.659105 (2025).
32. Littman, R. et al. Gene-embedding-based prediction and functional evaluation of perturbation expression responses with PRESAGE. Preprint at *bioRxiv* https://doi.org/10.1101/2025.06.03.657653 (2025).
33. Tong, X. et al. Deep representation learning of chemical-induced transcriptional profile for phenotype-based drug discovery. *Nat. Commun.* **15**, 5378 (2024).
34. Evans, N. J., Mills, G. B., Wu, G., Song, X. & McWeeney, S. Graph structured neural networks for perturbation biology. Preprint at *bioRxiv* https://doi.org/10.1101/2024.02.28.582164 (2024).
35. Rosen, Y. et al. Universal cell embeddings: a foundation model for cell biology. Preprint at *bioRxiv* https://doi.org/10.1101/2023.11.28.568918 (2023).
36. Wen, H. et al. CellPLM: pre-training of cell language model beyond single cells. Preprint at *bioRxiv* https://doi.org/10.1101/2023.10.03.560734 (2024).
37. Klein, D. et al. CellFlow enables generative single-cell phenotype modeling with flow matching. Preprint at *bioRxiv* https://doi.org/10.1101/2025.04.11.648220 (2025).
38. Adduri, A. K. et al. Predicting cellular responses to perturbation across diverse contexts with State. Preprint at *bioRxiv* https://doi.org/10.1101/2025.06.26.661135 (2025).

Methods

TxPert architecture

TxPert predicts the transcriptional response $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^n$ given a set of perturbation tokens $P \subset \mathcal{P} := \{1, \ldots, N\}$ and a basal state representation derived from a control expression profile $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, which has been aligned with the predicted cell with respect to cell line and batch effect. Here, $n \in \mathbb{N}$ denotes the number of experimentally measured genes and $N \in \mathbb{N}$ denotes the total number of observed perturbations in the data. These perturbation tokens (or identifiers) are used to select node representations from a biological gene (product)–gene (product) interaction KG whose embeddings are integrated with the basal state to produce the perturbed expression profile.

To combine the information of the basal state and the perturbations, we first learn latent representations of both, that is, $\mathbf{x} \mapsto \mathbf{s} \in \mathbb{R}^d$ and $P \mapsto \{\mathbf{z}_p \in \mathbb{R}^d : p \in P\}$ for a chosen latent dimension $d \in \mathbb{N}$. Then, we combine the information through latent shift, where a learned decoder g_ϕ predicts the perturbation effect from the given context, that is, $\hat{\mathbf{y}} = g_\phi(\mathbf{s} + \sum_{p \in P} \mathbf{z}_p)$, using the mean squared error (MSE):

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} ||\mathbf{y} - \hat{\mathbf{y}}||_2^2$$

This setup naturally integrates with single, double and more general multiperturbation settings of order $m := |P|$ through the additive and compositional design. More sophisticated combination functions may be used to learn transition functions $\mathbf{s}' = T_\psi(\mathbf{s}, \mathbf{z})$ that allow sequential latent cell state modeling of subsequently applied perturbations, for example, $\hat{\mathbf{y}} = g_\phi(T_\psi(\ldots T_\psi(\mathbf{s}, \mathbf{z}_{\mathbf{p}_1}) \ldots, \mathbf{z}_{\mathbf{p}_m}))$. To obtain $\mathbf{s}$ and $\mathbf{z}_p$, $p \in P$, we use learned encoders, namely the basal state model and the perturbation model, which are discussed below.

Basal state model. The basal state encoder is designed to capture intrinsic cellular attributes, such as cell-cycle stage, cell type and other baseline phenotypic features, by mapping a control cell's gene expression profile $\mathbf{x} \in \mathbb{R}^n$ to a compact, low-dimensional embedding, $\mathbf{s} = f_{\text{basal}}(\mathbf{x}) \in \mathbb{R}^d$. The basal state model was subject to hyperparameter tuning, with an MLP performing best for unseen and double tasks (Supplementary Note 1 and Supplementary Fig. 7 for cross-cell type). The MLP learns a direct deterministic mapping from high-dimensional input gene expression data to a fixed-size embedding space. The MLP architecture offers a simple and computationally efficient framework for representation learning, while still retaining the capacity to model complex, nonlinear dependencies inherent in gene expression data.

Basal information matching and aggregation. An important aspect of modeling the basal state is the alignment of the control with the predicted perturbed cell. Note that experimental protocols can vary widely between different data sources. Therefore, we randomly sample this control according to the same cell line and dataset or experimental protocol. Furthermore, we explore basal state matching, where the control cell is selected to closely match the batch metadata of the perturbed sample. As this matching is not unique, we randomly sample one such appropriate control. Lastly, we experiment with basal state averaging, where instead of a single control, we compute the average expression profile across all matching controls for a given cell line and/or batch. This produces a more stable estimate of the basal state. Both strategies consistently improved model performance in our experiments.

Encoders. Beyond MLP encoders on raw gene expression profiles, we explored multiple transcriptomics foundation model embeddings to obtain a latent representation of the basal state. Specifically, we experiment with scGPT, and scVI pretrained on the Joung dataset[39]. We also include a variant with no basal state encoder, where the (latent) basal state space is represented directly by the raw expression profile (that is, $\mathbf{s} := \mathbf{x}$). In this configuration, the perturbation decoder learns a Δ vector from the perturbation embedding, which is added to the control profile: $\mathbf{x} \in \mathbb{R}^n$, that is, $\hat{\mathbf{y}} = \mathbf{x} + g_\phi(\sum_{p \in P} \mathbf{z}_p)$. This resembles the formulation of the general baseline, with a trained model predicting the Δ instead of a hand-crafted heuristic. Unsurprisingly, this variant shines most in settings with limited data availability for learning robust basal state representations, for example, perturbation effect prediction in cell lines without seen perturbations.

Perturbation model. We rely on GNNs that use biological KGs capturing gene (product)–gene (product) interactions to learn informative embeddings for gene perturbations. The GNN learns a matrix of node embeddings associated with the perturbation tokens $\{1, \ldots, N\} \mapsto \mathbf{Z} \in \mathbb{R}^{N \times d}$, where $N \in \mathbb{N}$ is the number of perturbations relevant to the investigated task. Each row (or node representation) of this matrix represents the latent encoding of a specific perturbation, that is, $\mathbf{z}_p \in \mathbb{R}^d$ required for the latent shift is the pth row of $\mathbf{Z}$. More specifically, we first associate each perturbation $p \in \{1, \ldots, N\}$ with a randomly initialized input node embedding $\mathbf{h}_p^0 \in \mathbb{R}^{d_0}, d_0 \in \mathbb{N}$ that are consolidated in the input node feature matrix $\mathbf{H}^0 \in \mathbb{R}^{N \times d_0}$. During training, those node features are (1) treated as model parameters that are learned using backpropagation and (2) subsequently refined through the message passing of an L-layer GNN, that is, $\mathbf{H}^\ell = \texttt{LAYER}_{\theta_\ell}(\mathbf{H}_{\ell-1}), 0 \leq \ell \leq L$ and $\mathbf{Z} := \mathbf{H}^L$. This allows the model to characterize perturbation effects on the basis of known relationships from KGs such as GO[19], STRING[18] or proprietary data sources.

Real-world KGs present inherent imperfections; they often contain noisy or incorrect edges (false positives), suffer from missing connections (false negatives) and may originate from diverse sources offering multiple, sometimes conflicting, perspectives. The effective use of such complex data necessitates GNN architectures specifically chosen to address these challenges.

To this end, we select and adapt two fundamental GNN approaches. First, to handle noisy edges, we use attention-based models such as the GAT[40,41]. The ability of GAT to dynamically (re)weight neighbor importance provides much needed robustness by effectively downweighting less credible connections, which is a major difference relative to non-attention-based methods such as the simple graph convolution[42] used in GEARS. Second, to address graph incompleteness and capture long-range dependencies, we use GTs, specifically Exphormer[16,17]. Its capacity for attention beyond immediate graph neighbors allows it to potentially model implicit relationships and bridge structural gaps.

Furthermore, it proved crucial for the presented tasks to use multiple KGs that offer complementary and reinforcing perspectives on the task-related biology. For this, we explore architectures designed for synergistic learning from diverse sources. This includes extending GAT to GAT-Hybrid (allowing for node-level attention weighting of information from different KGs), introducing our provenance-aware Exphormer-MG variant and developing GAT-MLG, a multilayer extension of GAT that uses a supra-adjacency representation to effectively integrate information across multiple biological networks.

In Supplementary Note 3, we provide rigorous details on the relevant graph representation learning used for encoding perturbations, the proposed models and the techniques applied to take advantage of complementary information from multiple KGs. In our experimental setup, the learned embeddings for the perturbed gene(s) from the GNN were extracted and combined with a basal state representation to predict the resulting gene expression profile. This comparative analysis allowed us to investigate how different GNN strategies (attention, flexible connectivity and multigraph fusion) perform when learning from the complexities of biological KGs.

Training and evaluation framework

Algorithm 1. TxPert training algorithm

Require: Pert. cells y, control cells x, biological prior graph $G = (V, E)$ with perturbations $\mathcal{P} \subset V$

Ensure: Minimize MSE loss between predicted and true perturbed cell measurements

Initialize input perturbation embeddings $\{\mathbf{h}_v^0\}_{v \in V}$ randomly
for each training step **do**
  Sample a batch of perturbed cell profiles: $\{\mathbf{y}_i\}_{i=1}^B \subset y$
  Sample corresponding control cells from the same experimental batches: $\{\mathbf{x}_i\}_{i=1}^B \subset x$
  Enrich perturbation embeddings using graph prior: $\{\mathbf{z}_v\}_{v \in V} \leftarrow \mathtt{GNN}(G, \{\mathbf{h}_v^0\}_{v \in V})$
  for each sample in batch **do**
    Encode control cells into basal latent space: $\mathbf{b}_i \leftarrow \mathtt{MLP}_{\mathrm{basal}}(\mathbf{x}_i)$
    Retrieve perturbations $P_i \subset \mathcal{P}$ associated with target $\mathbf{y}_i$
    Combine control and perturbation embeddings: $\hat{\mathbf{z}}_i \leftarrow \mathtt{COM}(\mathbf{b}_i, \{\mathbf{z}_p : p \in P_i\})$
    Decode to predicted perturbed profile: $\hat{\mathbf{y}}_i \leftarrow \mathtt{MLP}_{\mathrm{dec}}(\hat{\mathbf{z}}_i)$
    Compute loss for each sample: $\mathcal{L}_i \leftarrow \mathtt{MSE}(\hat{\mathbf{y}}_i, \mathbf{y}_i)$
  end for
  Compute total loss over batch: $\mathcal{L} \leftarrow \sum_{i=1}^B \mathcal{L}_i$
  Backpropagate and update model parameters
end for

Data splits. The data were split into training, validation and test sets through grouping by perturbation such that distinct sets of unseen perturbations were reserved both the validation and test sets with target ratios of 0.5625, 0.1875 and 0.25 for the training, validation and test sets, respectively. Moreover, for the cross-cell-type task, the test set was a reserved cell type with only control cells included during training with a breakdown into seen and unseen perturbations therein. As an exception, for the doubles task on the Norman dataset, predefined splits were loaded from the GEARS setup.

Optimal hyperparameters for each model were selected based on the validation Pearson Δ metric. Only metrics on the test set are reported.

Metric definitions. All metrics are reported as weighted averages, that is, the mean of the mean across cells subjected to each unique perturbation, unless otherwise specified.

Expression value representations and Δ. Where not otherwise specified, expression values $\mathbf{x} \in x \cup y$, are represented as log1p-transformed and library size-normalized counts (with target library size of 4,000); that is, for a raw count $\mathbf{x}_{\mathrm{raw}} \in \mathbb{R}^n$, we define

$$\mathbf{x} := \log\left(1 + 4000 \cdot \frac{\mathbf{x}_{\mathrm{raw}}}{\| \mathbf{x}_{\mathrm{raw}} \|_1}\right).$$

The other representation used is a Δ representation, which is centered on batch-matched controls. Specifically, for each perturbed cell expression $\mathbf{y}_i \in y$ with cell line c and batch b, the expression is transformed to

$$\delta_i := \mathbf{y}_i - \bar{\mathbf{x}}_{c,b},$$

where $\bar{x}_{c,b}$ represents the mean expression of control cells $\mathbf{x} \in x$ with batch b and cell line c.

Pearson Δ. Slightly modified from the metric 'Pearson correlation (Δ expression)' from the GEARS manuscript, Pearson Δ calculates the correlation between predicted and observed log fold change versus batch-matched control mean,

$$\mathrm{Pearson}\Delta(p) := \mathrm{Pearson}(\hat{\delta}_p, \delta_p),$$

where $\hat{\delta}_p$ and δ_p are the batch-matched control centering of the prediction and ground truth, respectively, averaged across replicates of certain perturbation $p \in \mathcal{P}$. For simplicity, we define this and following metrics for single perturbations $p \in \mathcal{P}$ but note that analogous formulations are appropriate for multiple perturbations $P \subset \mathcal{P}$. The results across all predicted perturbation effects are then averaged to obtain an overall performance estimate.

Note that, for the GEARS model only, we report the exact 'Pearson correlation (Δ expression)' from the GEARS code base instead. We confirmed that any differences between 'Pearson correlation (Δ expression)' and our 'Pearson Δ' were much smaller in practice than the differences between models.

Retrieval. We use two variants of the retrieval rank metric that score a prediction's similarity to the ground truth not overall but relative to other perturbations. These metrics are the same as rank average from PerturBench[10], except that they focus on similarity with a perfect score of 1, a random score of 0.5 and perfect anticorrelated prediction score of 0:

$$\mathrm{Retrieval} := \frac{1}{N} \sum_{p \in \mathcal{P}} \mathrm{rank}(\hat{\delta}_p),$$

$$\mathrm{rank}(\hat{\delta}_p) := \frac{1}{N-1} \sum_{\substack{q \in \mathcal{P} \\ q \neq p}} \mathbf{1}_{\{\mathrm{Pearson}(\hat{\delta}_p, \delta_p) \geq \mathrm{Pearson}(\hat{\delta}_p, \delta_q)\}}.$$

For 'normalized' retrieval, the perturbation count $N := |\mathcal{P}|$ matches the original experiment, whereas, for 'fast retrieval', for computational efficiency, a seeded random reference set of only 100 perturbations is used, with the addition of the query perturbant p when not in the reference set (thus, $N \in \{100, 101\}$. Similar to Person Δ, we report the averaged performance across all perturbations.

Nonlearned general baseline. To establish a performance floor, we implement a nonlearned general baseline model that predicts expression profiles using mean values observed in the training data. This baseline uses an additive approach that combines the following:

- The mean test cell type control expression profile
- Either the perturbation-specific mean changes (for seen perturbations) or the global perturbation mean (for unseen perturbations)
- When multiple cell lines are present in the training set, we either use a weighted average according to the number of samples per cell line or the perturbation-specific mean changes from the most similar cell line (nearest-cell-line baseline). Here, similarity is determined on the basis of mean correlation of shared perturbation Δ values between the test and candidate cell line.

Consider a multiset of training samples $\mathrm{Train} \subset \mathcal{P}_{\mathrm{train}} \times \mathcal{C}_{\mathrm{train}} \times \mathcal{B}_{\mathrm{train}}$ consisting of combinations of perturbation(s), cell line(s) and batch effect(s) with a multiset test defined analogously. Consider a perturbation p such that $(p, c_p, b_p) \in$ test with cell line c_p and batch effect b_p. Implicitly, a (c_p, b_p) map is associated with a set of control cell profiles in that context.

If there exists $(p, c, b) \in$ train, we have

$$\hat{\mathbf{y}}_{(p,c_p,b_p)} = \bar{\mathbf{x}}_{(c_p,b_p)} + \frac{1}{|\{(q,c,b) \in \mathrm{Train} : q = p\}|} \sum_{\substack{(q,c,b) \in \mathrm{Train} \\ q=p}} \mathbf{y}_{(p,c,b)} - \bar{\mathbf{x}}_{(c,b)}.$$

Otherwise, we use the global Δ across perturbations observed in the training set, that is,

$$\hat{\mathbf{y}}_{(p,c_p,b_p)} = \bar{\mathbf{x}}_{(c_p,b_p)} + \frac{1}{|\text{Train}|}\sum_{(q,c,b)\in\text{Train}} \mathbf{y}_{(p,c,b)} - \bar{\mathbf{x}}_{(c,b)}.$$

For multiple perturbations, this baseline is implemented to initially attempt to use samples where the exact perturbation configuration is present. Otherwise, the perturbation is split into its components and each component is sequentially added to the test control mean according to the above method, adding a local Δ estimate if available and resorting to a global Δ otherwise.

Experimental reproducibility estimation: split-half validation and sample-based extension. As Perturb-seq is a destructive assay, we cannot observe the same cell in both perturbed and unperturbed states. This necessitates focusing on distribution means rather than individual cell accuracies. To approximate experimental reproducibility, we first use a split-half validation approach:

For each combination of perturbation(s), cell line context and batch, we apply three operations:

1. Divide test cells into two roughly equal halves
2. Calculate mean expression profiles for each half
3. Measure the agreement between these means using various metrics

To account for the randomness in choosing the half-split, we repeat the experiment across multiple seeded runs and report average performance. This provides a performance benchmark analogous to human-level reproducibility, which is called accuracy in other machine learning domains.

Consider the set of expression profiles $S \subset \mathcal{Y}$ for a fixed perturbation cell line context and batch (p, c, b) in the test set:

$$S' \subseteq S : |S'| \approx |S|/2$$

$$\bar{S}_1 = \frac{1}{|S'|}\sum_{\mathbf{y}\in S'} \mathbf{y}$$

$$\bar{S}_2 = \frac{1}{|S\setminus S'|}\sum_{\mathbf{y}\in S\setminus S'} \mathbf{y}.$$

We then report

$$\text{Reproduce}(p,c,b) = \text{Metric}(\bar{S}_1, \bar{S}_2),$$

where `metric` represents any of our evaluation metrics, for example, Pearson Δ, `Retrieval` or `MSE`. Theoretically, the split-half experimental reproducibility is not expected to establish an upper bound for performance of all models at test time because it operates on a different test set (only using half for prediction and testing, respectively). However, it empirically proves to be useful as a competitive (but still theoretically reachable) mark to beat.

As split-half reproducibility is likely an underestimate because of a reduced (halved) number of replicates, we also introduce a sample-based approach that gives an estimate for the reproducibility of the full-size dataset. Using the original count matrix, we calculate the per-batch probability distribution over genes (multinomial distribution maximum-likelihood estimator) for each perturbation. We then sample these distributions to generate two datasets that have the same number of observations (that is, cells) as the original dataset, but with stochastically resampled counts. These are then log1p-normalized and subset to the HVGs in the same way as the original dataset, before calculating the experiment reproducibility as described above. A comparison of split-half and sampled reproducibility is shown in Supplementary Table 4.

Data

Perturb-seq data sources. We demonstrate the efficacy of our approach across a range of datasets, including CRISPRi (gene knockdown) of ~2,000 essential genes in K562 and RPE1 cell lines from a previous study[13] (also used in GEARS[4]) and similarly designed CRISPRi experiments in Jurkat and HEPG2 cell lines from another previous study[24]. Furthermore, we implement the Norman[15] dataset with 94 unique single and 110 unique double CRISPRa (gene overexpression) perturbations respectively in the K562 cell line.

Graphs: sourcing and processing. The graphs used as inductive bias in this work can be classified into two main categories: (1) curated publicly available biological knowledge and (2) large-scale perturbation screens.

The curated graphs from category 1 include the GO graph, first used by GEARS, which is constructed by assigning edges between nodes that have a high Jaccard Index in their GO terms[19], the STRING graph[18] and Reactome[43].

Category 2 graphs are generated from large-scale perturbation screens including DepMap[44] and Perturb-seq[45]. These are extensive datasets linking genetic perturbation to either morphological or transcriptomic outcomes, which can offer particularly crucial insights into cellular responses to stimuli. To translate these experimental screens into graphs, we use derived embeddings to represent the genes and cell lines in a high-dimensional space, allowing for the analysis of relationships and identification of dependencies.

To curate these graphs, we first compute the pairwise similarity score between all combinations of genes. This means that, for each pair of genes (g_i, g_j), we compute the cosine similarity between their (aggregated) embeddings $\mathbf{x}_{g_i}$ and $\mathbf{x}_{g_j}$. Cosine similarity is computed as follows:

$$\text{cosine similarity}(\mathbf{x}_{g_i}, \mathbf{x}_{g_j}) = \frac{\mathbf{x}_{g_i}\cdot\mathbf{x}_{g_j}}{\|\mathbf{x}_{g_i}\|\|\mathbf{x}_{g_j}\|} = \frac{\sum_{k=1}^{n}\mathbf{x}_{g_ik}\mathbf{x}_{g_jk}}{\sqrt{\sum_{k=1}^{n}\mathbf{x}_{g_ik}^2}\cdot\sqrt{\sum_{k=1}^{n}\mathbf{x}_{g_jk}^2}}$$

where

- $\mathbf{x}_{g_i}\cdot\mathbf{x}_{g_j}$ represents the dot product of the vectors
- $\|\mathbf{x}_{g_i}\|$ and $\|\mathbf{x}_{g_i}\|$ represent the Euclidean norms (magnitudes) of vectors $\mathbf{x}_{g_i}$ and $\mathbf{x}_{g_j}$, respectively
- $\mathbf{x}_{g_ik}$ and $\mathbf{x}_{g_jk}$ are the individual components of vectors $\mathbf{x}_{g_ik}$ and $\mathbf{x}_{g_jk}$.

These cosine similarities are converted to their absolute values because the difference between highly cosine negative and highly cosine positive does not translate literally to the signed weight of the edge in the graph.

We additionally use proprietary data from internal genome-wide perturbation screens, where we measure the similarity of perturbation effect using both microscopy imaging and transcriptomics in various cell types.

Filtering configurations were optimized empirically. We found that the most performant configuration involved selecting for the top 1% of edges by (absolute) weight for screen-based graphs. For all other graph types, we (additionally) filtered for no more than 20 incoming nodes by target. Edge direction was assigned arbitrarily for undirected edges.

Data understanding. Additional methods related to specific analyses are described below.

Pharos knowledge rank. The Pharos initiative consolidates a variety of statistics relating to how researched and well known specific genes are[26]. Starting from this, we ranked knowledge levels as the mean of the rank of the Pharos Pubmed score and the rank of the Pharos negative log novelty score to create a single Pharos knowledge rank. We used this rank to break down and compare to the performance of models and understand potential bias. The 'knowledge levels' 0, 1, 2 and 3 correspond to the following bins of the Pharos knowledge rank:

- knowledge level 0 (least characterized): 0–0.2
- knowledge level 1: 0.2–0.4
- knowledge level 2: 0.4–0.6
- knowledge level 3: 0.6–0.8
- knowledge level 3 (most characterized): 0.8–1.

Within versus across. In investigating the correlations between controls and mean baselines, we compared 'within'-context correlation to 'across'-context correlation. Generally, before calculating either, all examples were first split into two mutually exclusive halves, A and B, where within-context correlation is a comparison of A versus B, in each context, while across-context correlation is a comparison of an arbitrary half of one context to another context. The only exception is across-batch controls in Fig. 1a, for which, to make a conservative estimate of across-batch variance, full batches were aggregated without splitting. For batch comparison, individual control cells were split and aggregated; for the mean baselines, the δ of perturbant replicate cells was preaggregated and then split (such that the halves had nonoverlapping perturbations).

Functional enrichment. To achieve a descriptive biological summary of the actual gene expression changes in the mean baseline, we first calculated a meta mean baseline (mean of all cell types and datasets, using the intersect of provided expressed genes) and defined upregulated and downregulated genes as having a remaining $\delta > 0.05$ or $\delta < -0.05$, respectively. We then ran functional enrichment testing separately on each on these sets (versus the background of all genes in the dataset intersect) using the GOATOOLS package[46].

Metric selection through retrieval. To avoid confusion and distraction caused by reporting many similarly performing or perhaps slightly contradicting metrics, we first performed an empirical 'test of the test metrics'. We adapted the evaluation method pioneered by a previous study[12] to work for our data and task. In short, this method uses cross-context retrieval of a perturbation as a way to judge whether a representation and metric together allow the retention and comparison of details necessary to distinguish perturbations. In our case, we modified the method to work on nonaggregated single-cell expression profiles (as this is the input and output of our model) and ran retrieval across the essential perturbation set on core cell types (K562, RPE1, HEPG2 and Jurkat). For each retrieval calculation, instead of aggregating, we first randomly sampled one cell of each perturbation. We ran three replicates on each of the cell × cell pairings for $n = 3 \times 6 = 18$ total estimates per perturbation. We report the 0.9 quantile, to focus on active perturbants; however, similar patterns were observed at other thresholds.

Note that the choice to focus on single cells excluded use of the representation selected by a previous study[12], namely the signed *P* value. To derisk this, we ran a preliminary analysis on the exact setup described previously[12]. We were only able to reproduce their results when making choices that would have limited the extensibility of our data and training setup; in particular, we found the high performance of the signed *P* value to rely on performing a global fit (and, thus, using a global estimate for gene-wise variance) across all contexts for determining differential expression.

We focused our selection of representations and metrics especially common in the perturbational transcriptomics literature but acknowledge the current omission of count-based representation and metrics.

General

Model development and analysis were performed with Python 3.12 and PyTorch 2.6.0. Plotting was performed with a combination of seaborn 0.13.2 and Matplotlib 3.10.8. Box-and-whisker plots used seaborn defaults, where the box represents the 0.25–0.75 quantiles, and the center line the median. The whiskers extend to the furthest observed data point within 1.5× the nearest interquartile range.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All data used here are already publicly available, with the exception of the PxMap and TxMap graphs, which are Recursion proprietary assets and not made available. Preprocessed data are available from Zenodo (https://doi.org/10.5281/zenodo.15420279)[47].

Code availability

The code to reproduce results with public data is available from GitHub (https://github.com/valence-labs/TxPert), including weights for top model variants using only public data.

References

39. Joung, J. et al. A transcription factor atlas of directed differentiation. *Cell* **186**, 209–229 (2023).
40. Veličković, P. et al. Graph attention networks. Preprint at https://doi.org/10.48550/arXiv.1710.10903 (2018).
41. Brody, S., Alon, U. & Yahav, E. How attentive are graph attention networks? Preprint at https://doi.org/10.48550/arXiv.2105.14491 (2021).
42. Wu, F. et al. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) (PMLR, 2019).
43. Milacic, M. et al. The Reactome pathway knowledgebase 2024. *Nucleic Acids Res.* **52**, D672–D678 (2024).
44. Tsherniak, A. et al. Defining a cancer dependency map. *Cell* **170**, 564–576 (2017).
45. Dixit, A. et al. Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell* **167**, 1853–1866 (2016).
46. Klopfenstein, D. V. et al. GOATOOLS: a Python library for Gene Ontology analyses. *Sci. Rep.* **8**, 10872 (2018).
47. Tu, W., Wenkel, F. & Denton, A. K. TxPert: leveraging biochemical relationships for out-of-distribution transcriptomic perturbation prediction. *Zenodo* https://doi.org/10.5281/zenodo.15420279 (2025).

Acknowledgements

We thank J. Hsu and H. Salam for their assistance with the figure design.

Author contributions

Conceptualization, F.W., W.T., C.M. and A.D. Data curation, C.M. and S.W. Formal analysis, A.D. Investigation, F.W., W.T., H.S., C.M., C.E., I.B., C.R. and A.D. Methodology, F.W., W.T., C.M., H.S., C.E., I.B. and C.R. Project administration, F.W. and A.D. Resources, B.E. Software, F.W., W.T., C.M., H.S., C.E., I.B., C.R., L.H., Y.E.M. and A.D. Supervision, F.W., J.D., M.F., B.E., E.N. and A.D. Validation, F.W., W.T., C.E. and A.D. Visualization, F.W., W.T., C.M., H.S., S.W., I.B., L.H. and A.D. Writing—original draft, F.W., W.T., C.M., H.S., C.E. and A.D. Writing—review and editing, F.W., W.T., C.M., H.S., S.W., I.B., L.H., M.F., E.N. and A.D.

Competing interests

All authors, except J.D., are currently or were employed by Recursion Pharmaceuticals during preparation of this manuscript. F.W., C.E., S.T.W., C.R., Y.M., M.M.F., B.E., E.N. and A.K.D. either currently hold or held equity of Recursion Pharmaceuticals during preparation of this manuscript.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-026-03113-4.

Correspondence and requests for materials should be addressed to Frederik Wenkel or Alisandra K. Denton.

Peer review information *Nature Biotechnology* thanks Eric Kernfeld, Luke O'Connor and Bence Szalai for their contribution to the peer review of this work. Peer reviewer reports are available.

nature portfolio

Corresponding author(s): Alisandra Denton
Frederik Wenkel

Last updated by author(s): Mar 24, 2026

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☐	☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Code is available at https://github.com/valence-labs/TxPert (v0.1.0). No other data was collected
Data analysis	Data analysis was performed in python 3.12, with further package versions as recorded here: https://github.com/valence-labs/TxPert/blob/main/pyproject.toml

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All data used here is already publicly available; with the exception of
the PxMap and TxMap graphs, which are Recursion proprietary assets and not made available.

Pre-processed and model ready data is automatically downloaded via the code below.
The code to reproduce results with public data is available at \url{https://github.com/valence-labs/TxPert}.

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	*Use the terms sex (biological attribute) and gender (shaped by social and cultural circumstances) carefully in order to avoid confusing both terms. Indicate if findings apply to only one sex or gender; describe whether sex and gender were considered in study design; whether sex and/or gender was determined based on self-reporting or assigned and methods used.* *Provide in the source data disaggregated sex and gender data, where this information has been collected, and if consent has been obtained for sharing of individual-level data; provide overall numbers in this Reporting Summary. Please state if this information has not been collected.* *Report sex- and gender-based analyses where performed, justify reasons for lack of sex- and gender-based analysis.*
Reporting on race, ethnicity, or other socially relevant groupings	*Please specify the socially constructed or socially relevant categorization variable(s) used in your manuscript and explain why they were used. Please note that such variables should not be used as proxies for other socially constructed/relevant variables (for example, race or ethnicity should not be used as a proxy for socioeconomic status).* *Provide clear definitions of the relevant terms used, how they were provided (by the participants/respondents, the researchers, or third parties), and the method(s) used to classify people into the different categories (e.g. self-report, census or administrative data, social media data, etc.)* *Please provide details about how you controlled for confounding variables in your analyses.*
Population characteristics	*Describe the covariate-relevant population characteristics of the human research participants (e.g. age, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write "See above."*
Recruitment	*Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.*
Ethics oversight	*Identify the organization(s) that approved the study protocol.*

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Training and evaluation was performed across five core datasets, including three different task setups, and with multiple metrics. The relative order of models by their performance was extremely robust across data and tasks. The one exception in model ranking, a single subset (no single seen) on the Norman data, had just 9 unique double perturbations, including repeated singles, thus making results on this exact split quite stochastic. These datasets were selected in large part for their critical mass and comparability, but also for their coverage of different tasks (e.g. including the Norman dataset for double perturbations). As the core conclusions hold on every dataset, every task (and all but one subset), we consider this sufficient replication, even without a sample size calculation. The final 'sample' accounted for in the study is perturbations (i.e. a collection of cells receiving a guide targeting the same gene). These were core to defining 'out of distribution'. For the unseen singles task: the splits were grouped by the covariate perturbation, such that all replicates for a unique perturbation were assigned to test, validation, or training; for the double perturbation task, this was extended to pairs of perturbations; while subsets were calculated based upon whether the comprising singles were seen in training. For cross cell type generalization, the perturbed cells were grouped by cell type for splitting, while the perturbations were stratified allowing for a seen perturbation, not-seen-perturbed cell line task. Unseen assignments were random, doubles assignments were taken from the GEARS paper for comparability, and cross cell type was performed as leave one out. The perturbations, post filtering and excluding controls, varied from 283 in the Norman data, to 2057 in Replogle K562, and 2393 in the remaining three datasets. This data was used but not created in this study, so no sample size calculation could be performed.
Data exclusions	Perturbations were filtered to those with at least 25 cells, 50 differentially expressed genes and , for cases where the CRISPRi perturbation target was measured to have positive expression in the control, perturbed target expression below 70% of control. Additional filtering of graphs was performed as described in methods.
Replication	Model training was performed with 5 seeds, with 5 to balance investment in compute with achieving a robust performance estimate. Some times a crashed run resulted in just 4 complete runs, and to provide a single 'n' in line with journal requirements, figures have been randomly

down sampled to match the least replicated configuration per plot; this down sampling did not change any conclusions. As down sampling further would not change conclusions; we consider this sufficient replication even without a sample size calculation.

The data itself, comes with up to thousands of perturbations and up to hundreds of cells per perturbation; as described in the original data-producing studies.

Randomization

Randomization was used extensively, from splits, to matching of control perturbed cells within a batch to model initialization. This was performed with numpy and torch random, as applicable.

Blinding

N / A, as no data was collected in this study, further regarding analysis 'best' was determined by non-subjective metrics that compared the closeness of predictions to a ground truth transcriptomic profile

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks

Report on the source of all seed stocks or other plant material used. If applicable, state the seed stock centre and catalogue number. If plant specimens were collected from the field, describe the collection location, date and sampling procedures.

Novel plant genotypes

Describe the methods by which all novel plant genotypes were produced. This includes those generated by transgenic approaches, gene editing, chemical/radiation-based mutagenesis and hybridization. For transgenic lines, describe the transformation method, the number of independent lines analyzed and the generation upon which experiments were performed. For gene-edited lines, describe the editor used, the endogenous sequence targeted for editing, the targeting guide RNA sequence (if applicable) and how the editor was applied.

Authentication

Describe any authentication procedures for each seed stock used or novel genotype generated. Describe any experiments used to assess the effect of a mutation and, where applicable, how potential secondary effects (e.g. second site T-DNA insertions, mosiacism, off-target gene editing) were examined.