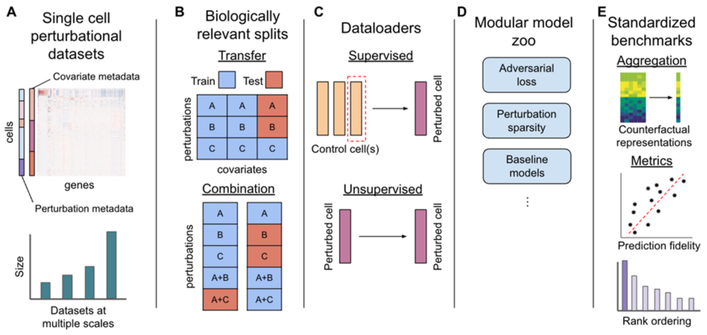
PerturBench: Benchmarking Machine Learning Models for Cellular Perturbation Analysis

Yan Wu*, Esther Wershof*, Sebastian M Schmon*, Marcel Nassar*, Błażej Osiński*, Ridvan Eksi*, Zichao Yan*, Rory Stark, Kun Zhang, Thore Graepel
*these authors contributed equally

ALTOS™

PerturBench Overview



PerturBench provides: **A)** Diverse perturbational dataset at multiple scales. **B)** Biologically relevant splits. **C)** Dataloaders that enable different training strategies. **D)** Modular model development platform/zoo that enable ablation studies. **E)** Novel benchmarks that capture key model behaviors.

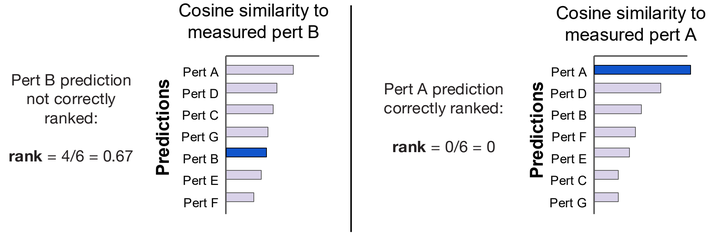
Datasets and benchmarks

Diverse perturbational datasets with single cell RNA-seq readouts

Dataset	Single pert.	Dual pert.	Modality	Primary cells	Biological states	Cells	Task
Srivatsan20	188	0	chemical	✗	3	178,213	*covariate transfer*
Frangieh21	248	0	genetic	✗	3	218,331	*covariate transfer*
Jiang24	219	0	genetic	✗	30	1,628,476	*covariate transfer*
McFalineFigueroa23	525	0	genetic	✗	15	892,800	*covariate transfer*
Norman19	155	131	genetic	✗	1	91,168	*combo prediction*
OP3	144	0	chemical	✓	4	296,147	*covariate transfer*

We provide datasets with both chemical and genetic perturbations, single and combinatorial perturbations, as well as a variety of cell states and dataset sizes. Adding a dataset to PerturBench is as simple as running our preprocessing script and creating a data config file.

Novel rank metric captures key model behavior and use-cases



We benchmark models using traditional measures of model fit including RMSE and cosine similarity. We also introduce a novel rank metric that measures whether the models can rank the perturbations correctly, which helps identify mode/posterior collapse in generative models. Ranking and prioritizing top perturbations is also a key downstream use-case for these models

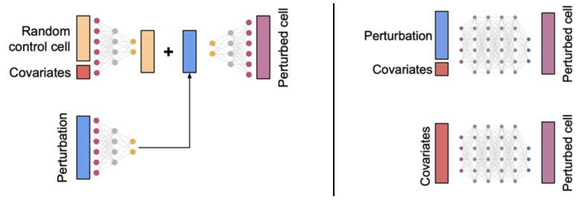
Models and baselines

Published model zoo

Model	Training Mode	Description
CPA*	Disentangling	Adversarial classifier for disentangling latent space
SAMS-VAE*		Sparse perturbation effects in latent space
BioLord*		Partitioned latent space
GEARS	Control matching	Embed perturbations from Gene Ontology and genes from co-expression using graph neural networks
scGPT	Frozen	Foundation model used to generate cell embeddings

We re-implemented (*) and wrapped published models using our model development platform. CPA* (noAdv) and SAMS-VAE* (S) remove the adversarial and sparsity inducing components respectively. Models with (scGPT) in the name use scGPT cell embeddings instead of gene expression

Baseline models

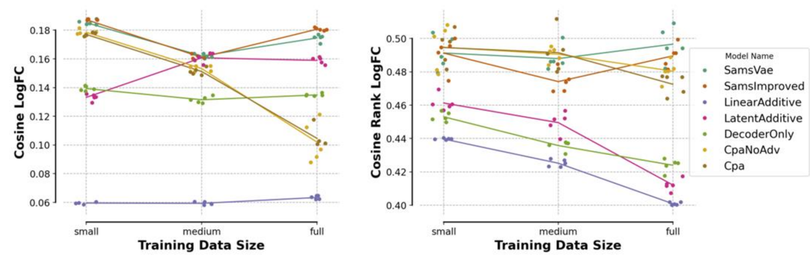


Results

Model	Predicting chemical perturbation effects across cell lines (**Srivatsan20**)		
	Cosine (higher is better)	**Rank** (lower is better)	**MMD** (lower is better)
CPA*	0.38 ± 6E−3	0.15 ± 1E−2	0.53 ± 4E−3
CPA* (noAdv)	0.40 ± 4E−2	**0.09 ± 4E−3**	**0.49 ± 1E−2**
CPA* (scGPT)	0.39 ± 9E−4	0.13 ± 2E−2	-
SAMS-VAE*	0.44 ± 1E−3	0.17 ± 1E−2	0.69 ± 1E−2
SAMS-VAE* (S)	**0.53 ± 1E−2**	0.17 ± 2E−2	0.79 ± 1E−2
BioLord*	0.18 ± 1E−1	0.37 ± 2E−2	4.3 ± 4E−0
Latent Additive	0.45 ± 2E−3	0.13 ± 4E−3	2.0 ± 2E−1
Latent Additive (scGPT)	0.50 ± 4E−3	0.13 ± 7E−3	-
Decoder Only	0.35 ± 5E−3	0.16 ± 1E−2	1.9 ± 5E−3
Covariate Only	0.30 ± 1E−2	0.47 ± 9E−3	-
Linear	0.16 ± 1E−2	0.28 ± 5E−3	0.76 ± 9E−4

We held out 30% of perturbations in each cell line for val/test and optimized model hyperparameters for each unique model/dataset, selecting the best trial with the val split. We then evaluated 4 replicate model runs on the test split to produce the final results.

Data scaling



We assessed performance on a subset of models on the **McFaline-Figueroa23** dataset where we held the val/test splits constant by holding out 70% of perturbations in 3 cell states. We then increased the training dataset by adding more cell states to the training dataset. We re-optimized hyperparameters for every dataset size.

Limitations

- We aimed to reimplement key components of published models and may be missing some elements of the original implementations
- Hyperparameter ranges may not capture the optimal hyperparameters for every model
- Latest model architectures such at CellFlow and STATE not benchmarked in this study

Key Takeaways

- Simple baseline models (Latent Additive, Linear) can outperform complex models for predicting mean perturbation responses
- More complex VAE models better than simple baselines for capturing full heterogeneity of perturbation responses
- Ablation studies show that key model components (i.e. SAMS-VAE sparsity) oftentimes do not improve performance, highlighting the importance of an integrated model development and benchmarking framework, as provided by PerturBench.
- Perturbation representation more important than input gene expression for control matching model performance
- Mode/posterior collapse can fool existing metrics but is captured with our rank metric

Key references

Srivatsan et al (2020), Science. doi.org/10.1126/science.aax6234
Norman et al (2019), Science. doi.org/10.1126/science.aax4438
McFaline-Figueroa et al (2023), Cell Genomics. doi.org/10.1016/j.xgen.2023.100487
Lotfollahi et al (2023), Mol. Syst. Bio. doi.org/10.15252/msb.202211517
Bereket and Karaletsos (2024), NeurIPS. doi.org/10.48550/arXiv.2311.02794
Piran et al (2024). Nat. Biotech. doi.org/10.1038/s41587-023-02079-x
Roohani et al (2024), Nat. Biotech. doi.org/10.1038/s41587-023-01905-6
Cui et al (2024), Nat. Methods. doi.org/10.1038/s41592-024-02201-0
Paszke et al (2019), NeurIPS. doi.org/10.48550/arXiv.1912.01703
Falcon, W. and The PyTorch Lightning team (2019). PyTorch Lightning
Yadan, O. (2019). Hydra - a framework for elegantly configuring complex applications. Github. github.com/facebookresearch/hydra

Preprint

Code

Altos Jobs

Altos is hiring! Scan the Jobs QR code to learn more