Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τηλεπικοινωνιών

Multi-task learning in perturbation modeling

Διπλωματική Εργασία
του
Θεόδωρου Κατζάλη

Επιβλέπων: Όνομα Επίθετο
Καθηγητής Α.Π.Θ.

April 16, 2025

Περιεχόμενα

1 Abstract

With the recent advancements in single-cell technology and the large scale perturbation datasets, the field of perturbation modeling has created an opportunity for a wide variety of computational methods to be leveraged to harness its potential. Multi-task learning is one of the methods that has been left unexplored in this field. In this study we aim to bridge this gap unraveling the potential of multi-task learning in single-cell perturbation modeling.

2 Introduction

The complexity of biological systems have imposed a challenge to capture the underlying mechanisms of cellular heterogeneity. Deciphering the effect of external stimuli (perturbation) at the cellular level, a field referred to as perturbation modeling [5], plays a crucial role in biomedicine and drug discovery. With the recent surge of data generation, machine learning methods aim to understand the effect of perturbations and to extrapolate on unseen events.

An overview of the models on perturbation modeling can be found on this study [3]. One of the main objectives is the out-of-distribution detection, which is the focal point of our study. The task is about predicting the perturbation response of the omics signature of cells with a specific cell type, while having observed the perturbation response of other cell types.

UnitedNet [10] is a multi-task framework that has shown its potential in multi-omics tasks such as cross modal prediction and cell type classification. We aim to extend this approach to perturbation modeling.

3 Current single-cell perturbation modeling methods

In the literature body, there are several approaches for predicting single-cell perturbation responses. To compare our multi-task method, we have chosen the models of scGen [7], scPreGAN [11], scButterfly [1], and scVIDR [6].

scGen projects the gene expression profile to a probabilistic latent space with a VAE. Then, the perturbation effect is modeled with a vector that represents the difference between the control and perturbed gene expression projections in the latent space.

scVIDR builds upon scGen by enhancing the architecture with cell type–specific knowledge. It is capable of predicting cellular responses to multiple chemical perturbations in a dose-dependent manner. As a multi-task model, scVIDR leverages data from various perturbations to improve prediction accuracy across conditions.

scPreGAN is based on a GAN and autoencoder setup. The study aims to decouple the perturbation effect from the latent space, and to apply it to the decoder stage.

scButterfly was not originally designed for perturbation modeling. However, its cross-modal architecture, which includes dual aligned VAEs, can be repurposed for perturbation tasks. Rather than predicting one omic modality from another, the model can treat perturbed and control gene expression profiles as distinct modalities.

The performance of all of these models will serve as a baseline of our multi-task learning architecture.

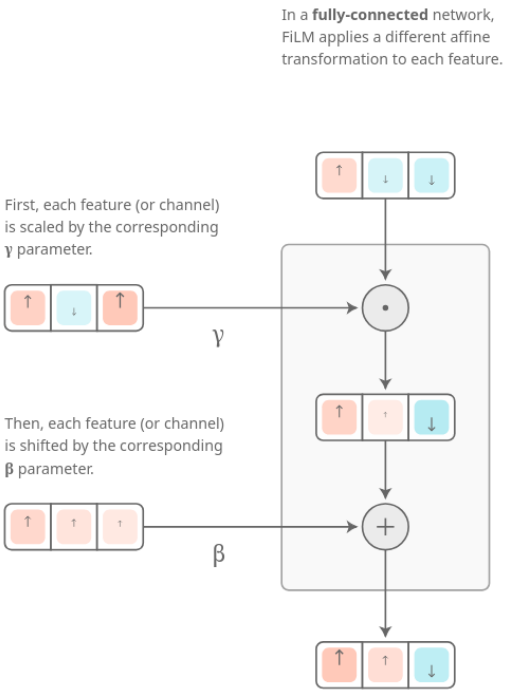


Figure 1: Illustration of the feature-wise transformation [2]

4 Method

Multi-task learning is a machine learning paradigm and its core idea is that training a model to solve multiple tasks can be more effective than training separate models for each specific task [12]. A joint architecture that shares knowledge between the tasks can lead to better generalization. The relationship of the tasks determines the positive or negative transfer to each other and the overall effectiveness of the paradigm.

Defining as a task the prediction of the gene expression given a perturbation, we will explore designing a model that can predict gene expressions after a perturbation for a set of perturbations.

One of the key problems of deep learning methods is the data demand. Another benefit of multi-task learning is the combination of data from multiple sources of informations, especially in perturbation modeling where the data is limited for a specific number of perturbations.

To integrate the tasks, we have explored the application of feature-wise transformations [2]. For this kind of transformation, we have:

$$\text{FiLM(x)} = \gamma(z) \odot x + \beta(z)$$

, where γ, and β are learnable parameters generated by a network that represent a condition z (e.g. a vector that indicates the task), and x is the input.

This particular technique is referred to as conditional affine transformation (a combination of multiplicative and additive conditioning) that shifts and scales the input element-wise. It is efficient in terms of scaling and parameters compared to multi-head architectures, where each task has its dedicated network to generate the output of the task.

In our approach, we aim to decouple the perturbation effect by constructing a perturbation-free latent space, while explicitly modeling the perturbation response through a conditioning

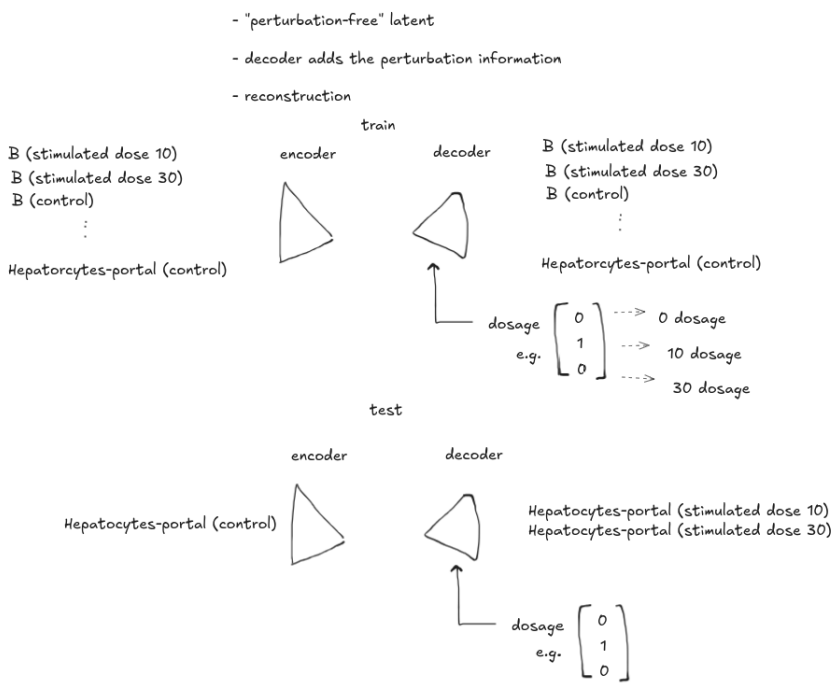


Figure 2: Illustration of the multi-task architecture. The encoder is shared across all tasks, while the decoder is conditioned by the task-specific FiLM layers.

vector. Our architecture is built around an autoencoder, where task-specific conditioning — in our case, the type of perturbation — is integrated via FiLM layers fused into the decoder (`MTAe`). The modulation parameters γ and β are learned independently for each fusion point.

The loss is the reconstruction loss of the autoencoder, which is the mean squared error between the input and the output of the decoder:

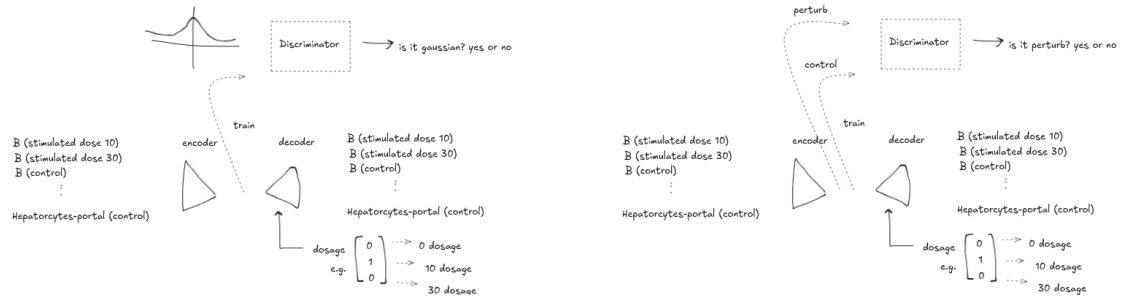
$$\mathcal{L}_{\text{recon}} = \frac{1}{N} \sum_{i=1}^{N} ||x_i - \hat{x}_i||^2$$

, where x_i is the input gene expression profile, $\hat{x}_i$ is the reconstructed gene expression profile, and N is the number of samples.

Regarding data splitting, we hold of the stimulated samples of the cell type of interest as a test set. The controlled ones, along with the rest of the cell types in both conditions of control and stimulated, are used for training. Thus, the autoencoder during training attempts to reconstruct the gene expressions while the condition vector is set accordingly to the type of perturbation. The condition vector is one-hot encoded, and given a dataset with N perturbations, its length is N+1, including the control condition.

We have explored several variations of this approach, all of which maintain the decoder architecture with the inclusion of FiLM-based conditioning. These variations can be split to three main groups, a) adversarial autoencoders, b) optimal transport, c) Variational Autoencoders (VAEs).

Regarding the first ones, we are aiming to enforce a condition in the latent space via an adversarial loss. The architecture consists of the aforementioned autoencoder scheme with the FiLM layers with the addition of the discriminator. The discriminator aims to differentiate between samples of the latent space and a target distribution, while the encoder aims to fool the discriminator via an adversarial loss to enforce the target distribution in the latent space.



(a) (b)

Figure 3: Adversarial autoencoders

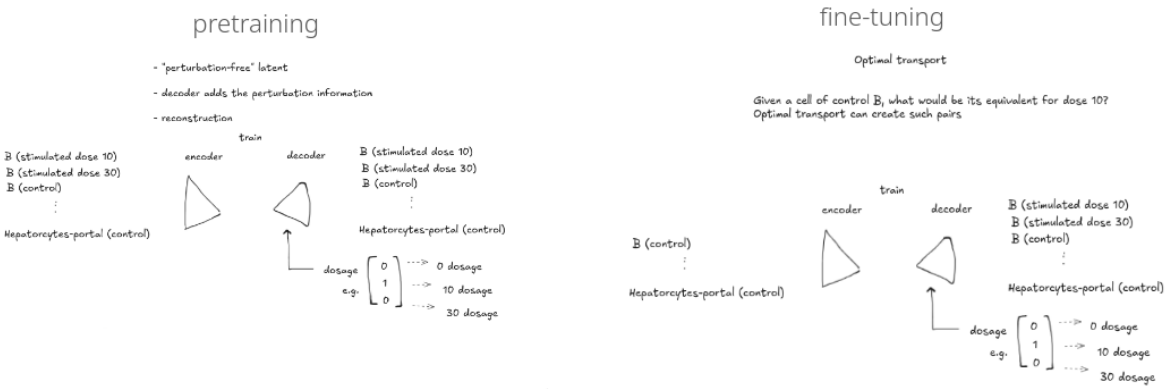


Figure 4: Using optimal transport to fine-tune the MTAe architecture (MTAePlusOT)

For the `MTAeAdv` architecture, we have attempted to explicitly model a perturbation-free latent space, by using a discriminator to differentiate between the control and perturbed gene expression profiles. In that case, the samples were drawn from the latent space. Similarly, the `MTAeAdv` architecture aims to enforce a Gaussian distribution in the latent space, by using a prior Gaussian distribution for the discriminator to sample from.

Another set of variations is the inclusion of optimal transport. In single-cell RNA sequencing, we can't sequence the same cell before and after a perturbation, thus we compare distributions since we lack the pair-wise information. To mitigate this, optimal transport can be used to create these pairs, by sampling from the perturbed distribution and matching it with the sample from the control distribution. Using that technique, instead of reconstructing the input, the goal was, given a sample from the controlled distribution, to predict its pair from the perturbed distribution. The loss is the mean squared error between the input and the output of the decoder, defined as:

$$\mathcal{L}_{\text{OT}} = \frac{1}{N} \sum_{i=1}^{N} ||x_i - \hat{x}_i||^2$$

, where x_i is the input gene expression profile, $\hat{x}_i$ is the pair from the perturbed gene expression profile, and N is the number of samples. Compared to the previous architectures, the perturbed gene expression profiles are not fed in the network and used only to calculate the loss. This approach is named as `MTAeOT`. Additionally, we have attempted to pretrain the model with the `MTAe` architecture and then fine-tune it with the `MTAeOT` architecture. This approach is named as `MTAePlusOT`.

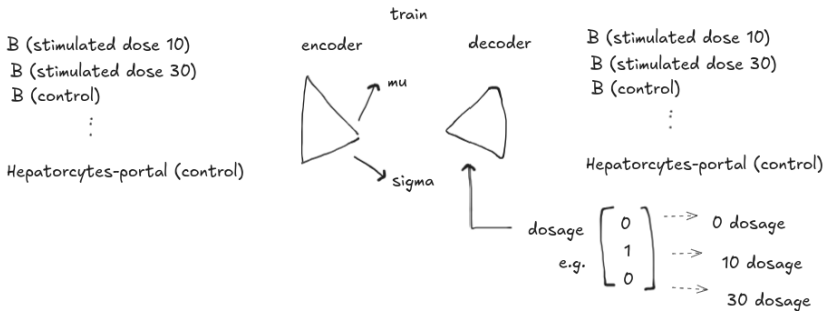


Figure 5: VAE

The last set of variations involves the inclusion of Variational Autoencoders (VAEs). The architecture builds upon the previously described autoencoder framework augmented with FiLM layers, while additionally incorporating a VAE loss to regularize the latent space. The VAE loss is defined as the sum of the reconstruction loss and the Kullback–Leibler (KL) divergence between the learned latent distribution and a standard normal prior:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x) \,\|\, p(z))$$

Here, $q_\phi(z|x)$ is the encoder's approximation of the posterior over latent variables, $p_\theta(x|z)$ is the decoder's likelihood of reconstructing the input, and $p(z) \sim \mathcal{N}(0, I)$ is the prior over latent variables. This model is named as `MTVae`, and as we have described above with the optimal transport use case, we have the `MTVaeOT` and `MTVaePlusOT` architectures.

5 Evaluation

We have tested the models on two datasets, one where human peripheral blood mononuclear cells have been stimulated by IFN-b interferon (Kang et al. [6]), and a multi-perturbation dataset, where liver cells have been stimulated by multiple doses of tetrachlorodibenzo-p-dioxin (TCDD) in vivo (Nault et al. [8, 9]).

The models are evaluated on the unseen cell type, given as input the control gene expression fig. 6. Regarding the single perturbation response models, the scGen, scButterfly, scPreGAN and scVIDR's single-task version, for the multi-perturbation dataset of ten dosages Nault et.al [8,9], we have trained a dedicated model for each dosage. In these cases, the dataset is consisted of only two conditions the control and the perturbed one for a particular dosage. The performance is measured by comparing the predicted gene expression with the actual one.

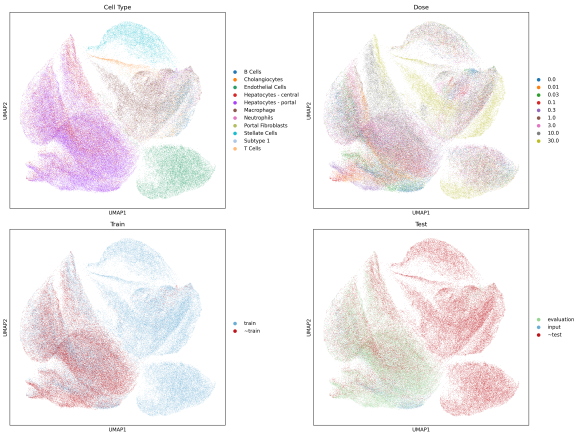
For this comparison, we have used the count of differentially expressed genes (DEGs), the R^2 of all the highly variable genes (HVGs), and the top 100 most variable ones. To complement the evaluation, we have calculated a set of five distance metrics (euclidean, edistance, wasserstein, mean pairwise, mmd) to capture the differences between the expected and predicted perturbed gene expressions in a point-wise and distributional manner using pertpy [4].

To address the randomness of the models, we have performed the experiments three times, with three different seeds 1, 2, 19193, and the metrics have been averaged across experiments.

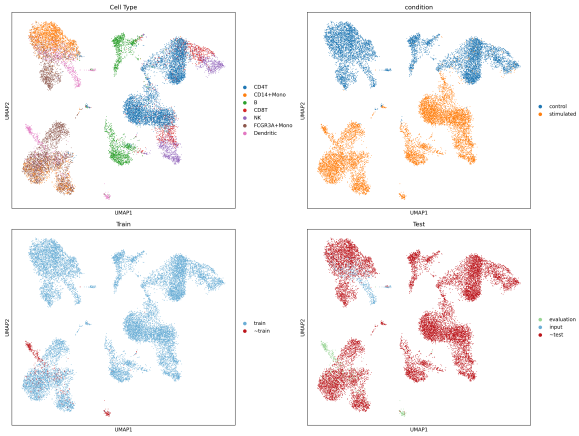
To rank the models, since there could be conflicting cases between metrics, where one model could be better than the other, per model's metric we averaged them across all the experiments. Then we scaled them to the range of 0-1 with the following formula:

$$\frac{\text{current} - \text{best}}{\text{worst} - \text{best}}$$

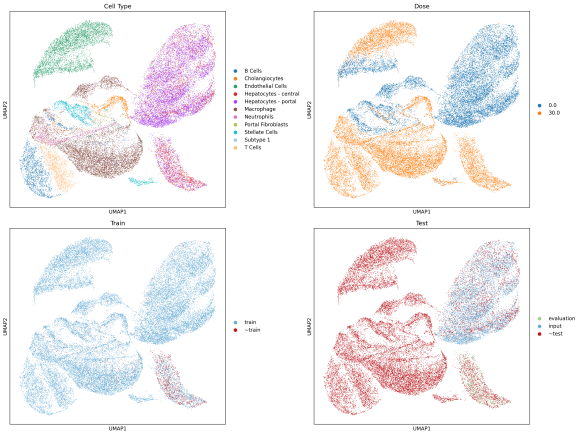
, that can track how a metric deviates from the best one. Then we summed all the metrics, giving a score (penalty) to each model. The model with the lowest score is considered the best one.



(a) Nault et al. [8, 9]



(b) Kang et al. [6]



(c) Example of single dose split of Nault et al. [6] for the dosage $30\mu g/kg$

Figure 6: UMAP representations of data split

6 Results

Initially, we will benchmark only our multi-task variations to filter out the most promising models figs. 13 to 18. As we can see the best performing ones are MTAe, MTAeAdv, and MTAeAdvG. Thus, we will compare them with state-of-the-art literature models such as scButterfly, scVIDR, scPreGAN and scGen.

scVIDR performance drops for DEGs, and distance metrics, but it performs well for the R^2 metrics and stays very consistent, along with scGEN. The multi-task models and scButterfly exhibit greater variability across measurements, but better performance on average. The optimal transport variations performed poorly overall, but were among the best for distance metrics for the Nault et al. [8, 9] dataset.

model	DEGs	R^2_{HVG}	R^2_{HVG20}	R^2_{HVG100}	Euc	Was	E-dist	MPD	MMD
MTAe	**0.000**	0.077	0.330	0.140	0.479	0.815	0.506	0.898	0.479
	(75.714)	(0.946)	(0.871)	(0.917)	(0.488)	(0.892)	(0.651)	(0.949)	(0.488)
MTAeAdv	0.066	0.026	0.053	0.043	0.032	0.006	0.044	0.110	0.032
	(72.381)	(0.961)	(0.955)	(0.948)	(0.202)	(0.604)	(0.429)	(0.800)	(0.202)
MTAeAdvG	0.195	0.168	0.305	0.192	0.503	0.636	0.570	0.689	0.503
	(65.905)	(0.917)	(0.878)	(0.901)	(0.504)	(0.828)	(0.681)	(0.909)	(0.504)
MTAeOT	0.688	1.000	1.000	1.000	0.984	0.969	0.990	0.976	0.984
	(41.190)	(0.657)	(0.668)	(0.648)	(0.811)	(0.947)	(0.883)	(0.963)	(0.811)
MTAePlusOT	0.768	0.960	0.983	0.970	0.982	0.982	0.983	0.988	0.982
	(37.190)	(0.670)	(0.674)	(0.657)	(0.810)	(0.951)	(0.880)	(0.966)	(0.810)
MTVae	0.132	0.088	0.056	0.105	0.125	0.056	0.189	0.114	0.125
	(69.095)	(0.942)	(0.954)	(0.928)	(0.261)	(0.621)	(0.499)	(0.800)	(0.261)
MTVaeOT	0.720	0.961	0.969	0.953	0.988	0.993	0.990	0.992	0.988
	(39.571)	(0.669)	(0.678)	(0.663)	(0.813)	(0.955)	(0.883)	(0.966)	(0.813)
MTVaePlusOT	0.899	0.987	0.994	0.976	1.000	1.000	1.000	1.000	1.000
	(30.619)	(0.661)	(0.670)	(0.655)	(0.821)	(0.958)	(0.888)	(0.968)	(0.821)
scButterfly	0.299	0.251	0.187	0.232	0.140	**0.000**	0.128	**0.000**	0.140
	(60.727)	(0.891)	(0.914)	(0.889)	(0.271)	**(0.601)**	(0.469)	**(0.779)**	(0.271)
scGen	0.868	0.191	0.326	0.290	0.697	0.863	0.744	0.885	0.697
	(32.143)	(0.910)	(0.872)	(0.870)	(0.627)	(0.909)	(0.765)	(0.946)	(0.627)
scPreGAN	0.796	0.634	0.376	0.518	0.496	0.248	0.572	0.381	0.496
	(35.750)	(0.771)	(0.857)	(0.799)	(0.499)	(0.690)	(0.682)	(0.851)	(0.499)
vidrSingle	1.000	**0.000**	**0.000**	**0.000**	**0.000**	0.014	**0.000**	0.096	**0.000**
	(25.536)	**(0.970)**	**(0.971)**	**(0.961)**	**(0.182)**	(0.606)	**(0.408)**	(0.797)	**(0.182)**

Table 1: Score of the models for Kang et al. [6] along with the actual value in parenthesis

model	DEGs	R^2_{HVG}	R^2_{HVG20}	R^2_{HVG100}	Euc	Was	E-dist	MPD	MMD
MTAe	**0.000**	0.167	0.267	0.189	0.056	0.603	0.116	0.945	0.056
	(20.341)	(0.862)	(0.792)	(0.833)	(1.386)	(1.217)	(1.116)	(1.050)	(1.386)
MTAeAdv	0.368	0.388	0.518	0.458	0.025	0.246	0.045	0.426	0.025
	(13.716)	(0.792)	(0.725)	(0.743)	(1.128)	(1.091)	(1.011)	(1.017)	(1.128)
MTAeAdvG	0.113	0.339	0.477	0.396	0.029	0.293	0.058	0.607	0.029
	(18.307)	(0.808)	(0.736)	(0.764)	(1.164)	(1.107)	(1.030)	(1.029)	(1.164)
MTAeOT	0.650	0.969	0.829	0.916	0.001	0.008	0.005	0.122	0.001
	(8.652)	(0.608)	(0.642)	(0.590)	(0.925)	(1.006)	(0.951)	(0.998)	(0.925)
MTAePlusOT	0.657	0.956	0.822	0.897	**0.000**	**0.000**	0.002	0.099	0.000
	(8.519)	(0.613)	(0.644)	(0.596)	**(0.917)**	**(1.004)**	(0.948)	(0.996)	(0917)
MTVae	0.076	0.339	0.523	0.428	0.025	0.273	0.041	0.413	0.025
	(18.981)	(0.808)	(0.724)	(0.753)	(1.124)	(1.100)	(1.005)	(1.016)	(1.124)
MTVaeOT	0.677	0.953	0.830	0.906	0.001	0.016	0.006	0.132	0.001
	(8.163)	(0.614)	(0.642)	(0.593)	(0.929)	(1.009)	(0.952)	(0.998)	(0.929)
MTVaePlusOT	0.647	0.948	0.816	0.894	0.000	0.008	0.003	0.112	**0.000**
	(8.701)	(0.615)	(0.645)	(0.597)	(0.919)	(1.006)	(0.948)	(0.997)	**(0.919)**
scButterfly	0.196	0.553	0.633	0.600	0.008	0.029	**0.000**	**0.000**	0.008
	(16.818)	(0.740)	(0.694)	(0.696)	(0.984)	(1.014)	**(0.944)**	**(0.990)**	(0.984)
scGen	0.781	**0.000**	**0.000**	**0.000**	0.178	0.637	0.299	0.805	0.178
	(6.288)	**(0.915)**	**(0.863)**	**(0.897)**	(2.408)	(1.229)	(1.387)	(1.041)	(2.408)
scPreGAN	0.324	1.000	1.000	1.000	0.007	0.042	0.017	0.163	0.007
	(14.511)	(0.599)	(0.596)	(0.562)	(0.972)	(1.019)	(0.969)	(1.000)	(0.972)
vidrMult	1.000	0.143	0.099	0.133	1.000	1.000	1.000	1.000	1.000
	(2.352)	(0.870)	(0.837)	(0.852)	(9.295)	(1.358)	(2.425)	(1.054)	(9.295)
vidrSingle	0.920	0.191	0.247	0.216	0.061	0.480	0.117	0.544	0.061
	(3.795)	(0.855)	(0.797)	(0.824)	(1.431)	(1.174)	(1.118)	(1.025)	(1.431)

Table 2: Score of the models for Nault et al. [8, 9] along with the actual value in parenthesis

model	score	baseline score	distance score
MTAeAdv	0.414099	0.188957	0.225142
MTVae	0.989313	0.381226	0.608087
vidrSingle	1.109933	1.000000	0.109933
scButterfly	1.375249	0.968395	0.406855
MTAe	3.723979	0.546259	3.177721
MTAeAdvG	3.762873	0.860886	2.901987
scPreGAN	4.519561	2.325642	2.193919
scGen	5.559246	1.674543	3.884703
MTVaeOT	8.552051	3.602547	4.949504
MTAeOT	8.590746	3.688019	4.902727
MTAePlusOT	8.598116	3.680466	4.917650
MTVaePlusOT	8.855812	3.855812	5.000000

Table 3: Kang et al. [6]

model	score	baseline score	distance score
scButterfly	2.026767	1.981569	0.045198
MTVae	2.142595	1.365570	0.777025
MTAeAdvG	2.341641	1.324716	1.016925
MTAe	2.398554	0.622043	1.776511
MTAeAdv	2.498785	1.731406	0.767379
vidrSingle	2.837696	1.573823	1.263873
scGen	2.878990	0.781217	2.097772
MTVaePlusOT	3.428329	3.305106	0.123224
MTAePlusOT	3.433685	3.332175	0.101511
MTAeOT	3.500992	3.363746	0.137247
MTVaeOT	3.522593	3.365641	0.156953
scPreGAN	3.559583	3.324068	0.235515
vidrMult	6.375357	1.375357	5.000000

Table 4: Nault et al. [8, 9]

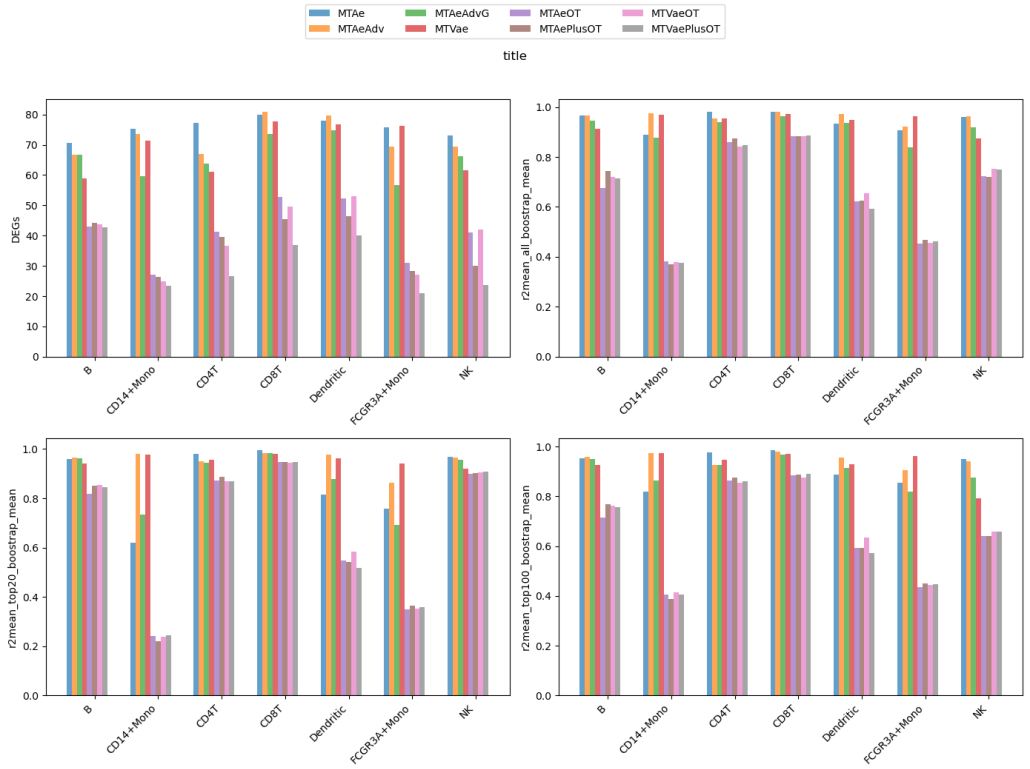


Figure 7: Baseline metrics of multi-task models for the Kang et al. [6] dataset across cell types

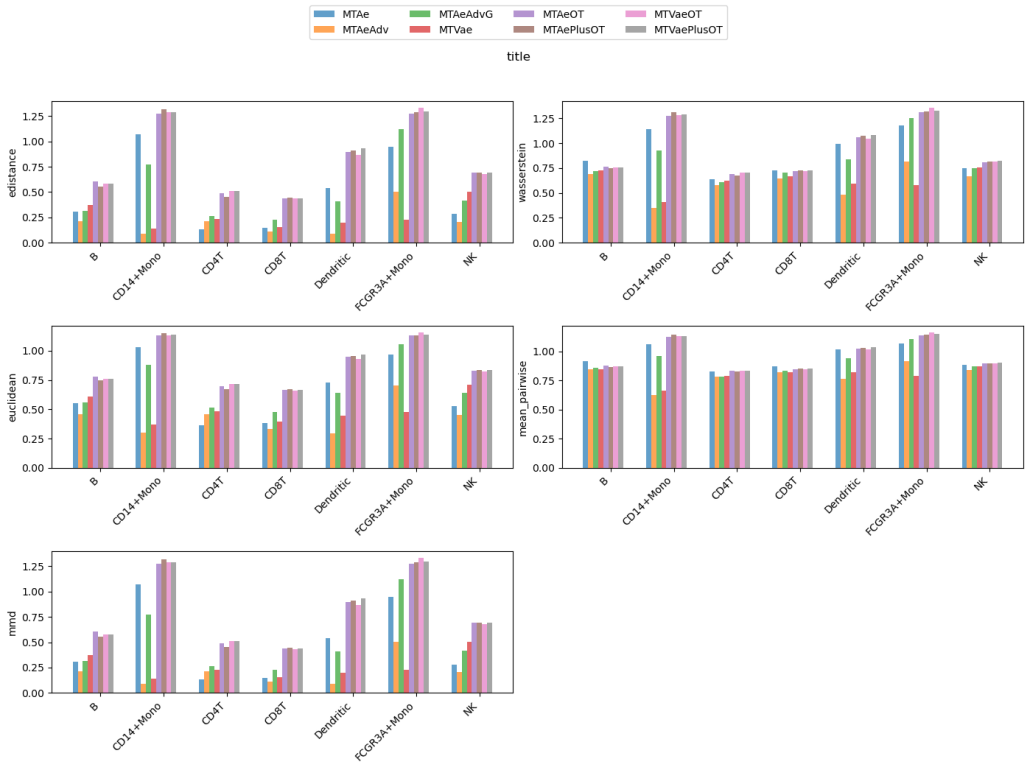


Figure 8: Distance metrics of multi-task models for the Kang et al. [6] dataset across cell types

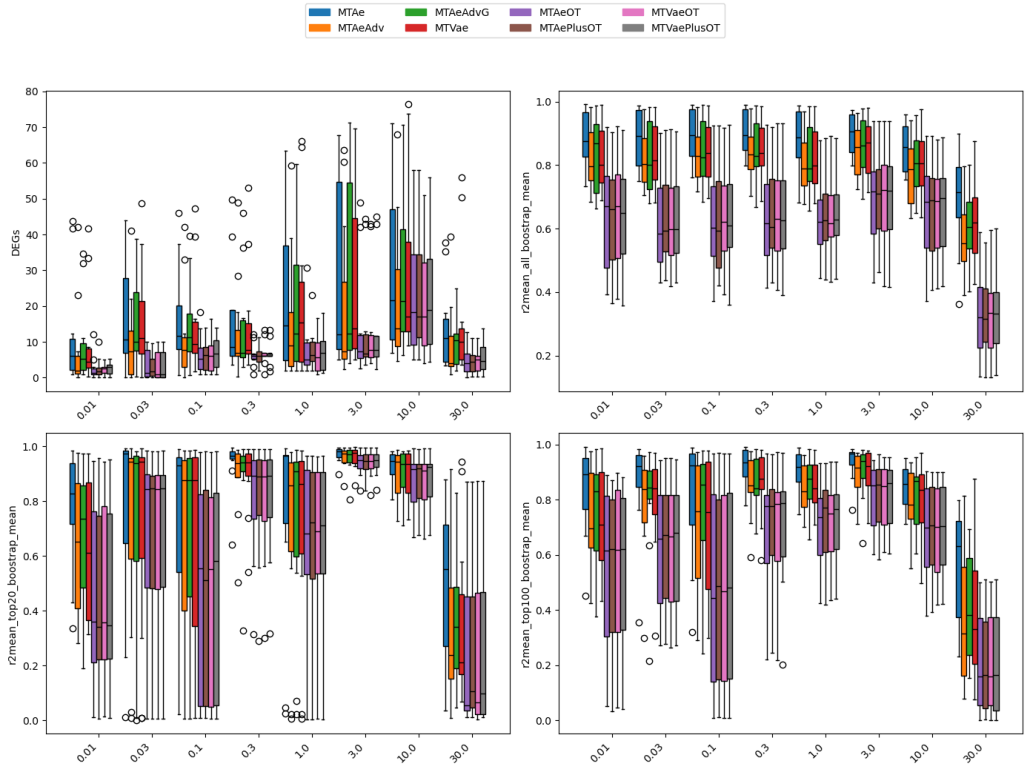


Figure 9: Baseline metrics of multi-task models for the Nault et al. [8,9] dataset across dosages

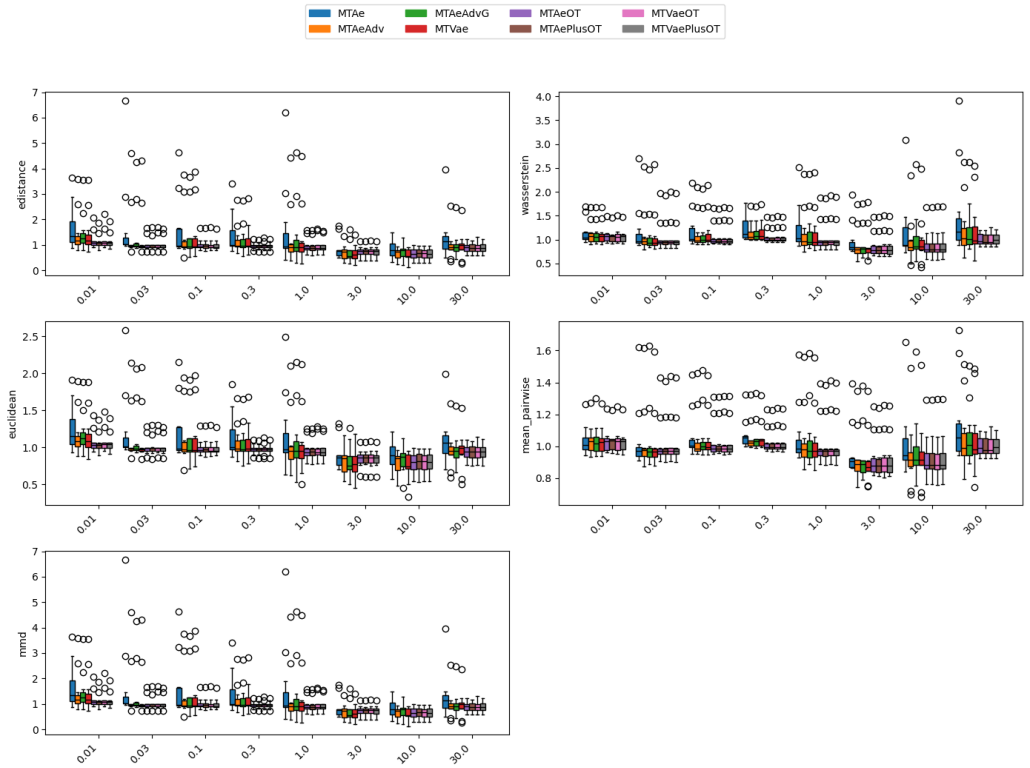


Figure 10: Distance metrics of multi-task models for the Nault et al. [8,9] dataset across dosages

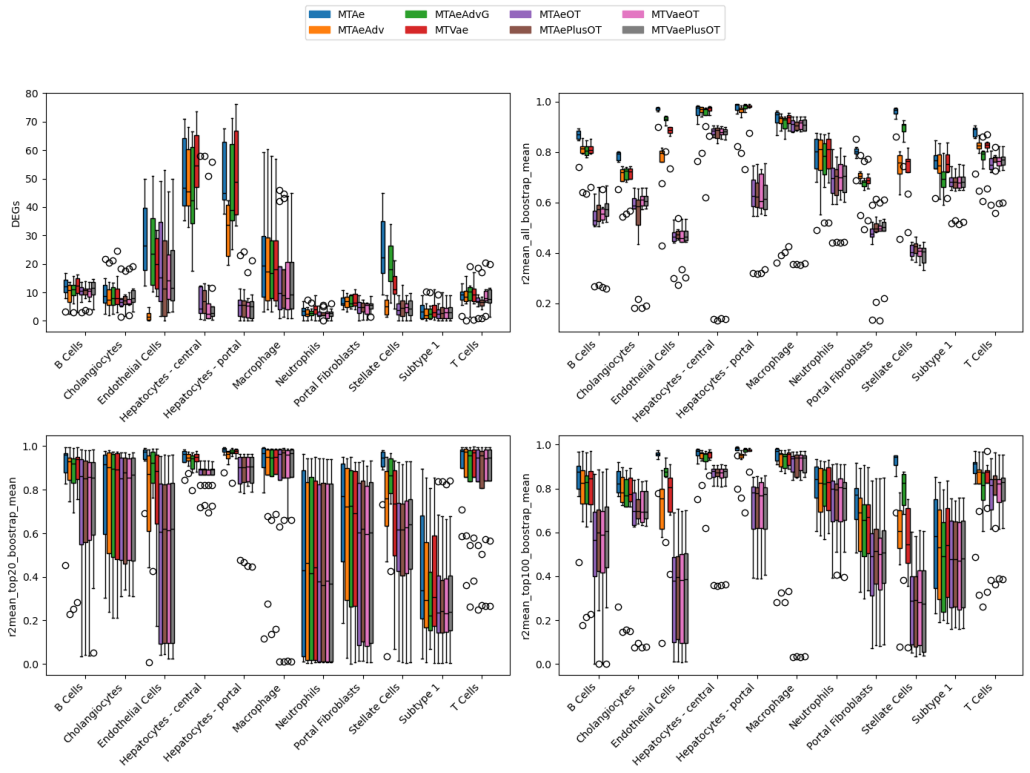


Figure 11: Baseline metrics of multi-task models for the Nault et al. [8, 9] dataset across cell types

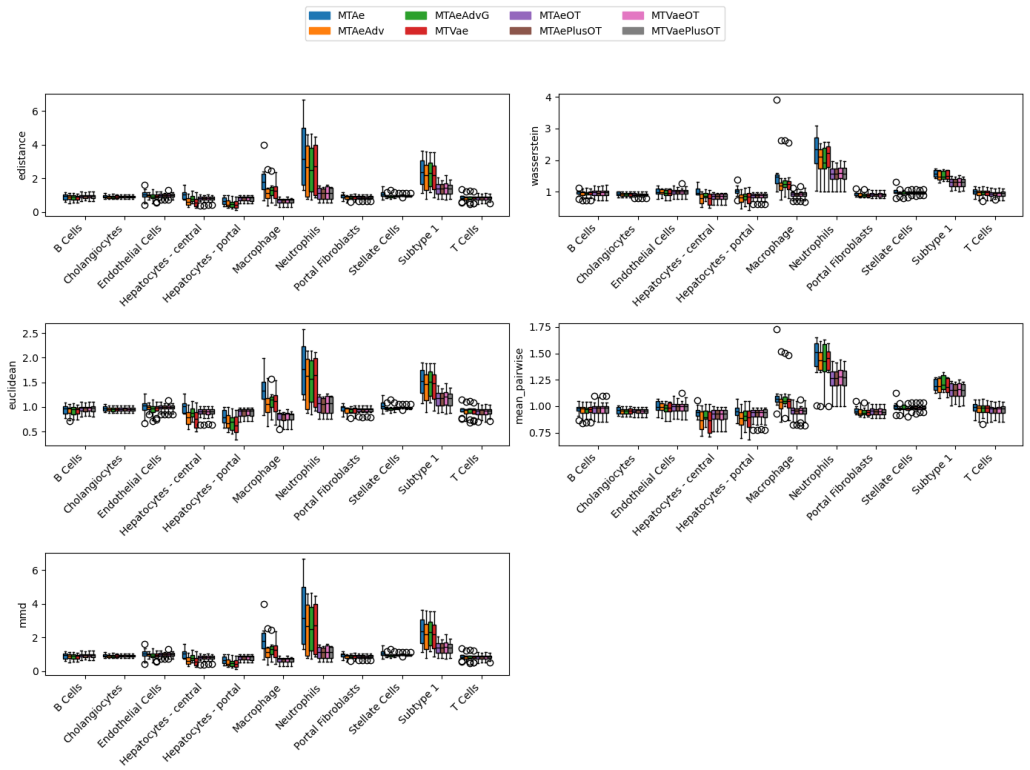


Figure 12: Distance metrics of multi-task models for the Nault et al. [8, 9] dataset across cell types

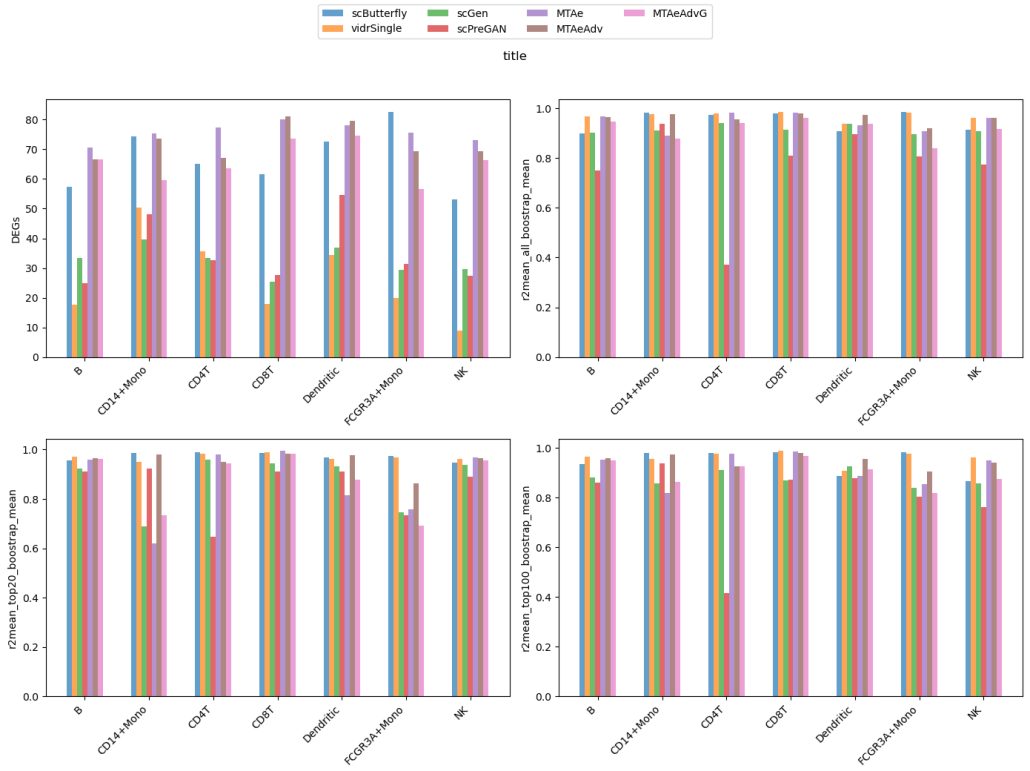


Figure 13: Baseline metrics of multi-task and literature models for the Kang et al. [6] dataset across cell types

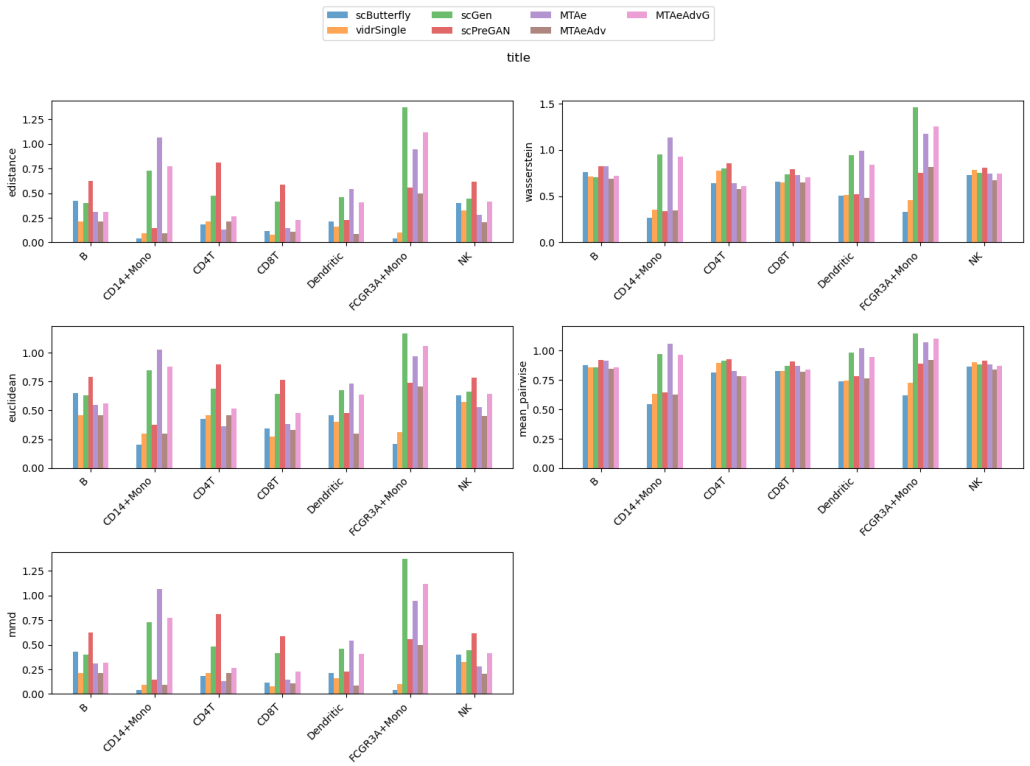


Figure 14: Distance metrics of multi-task and literature models for the Kang et al. [6] dataset across cell types

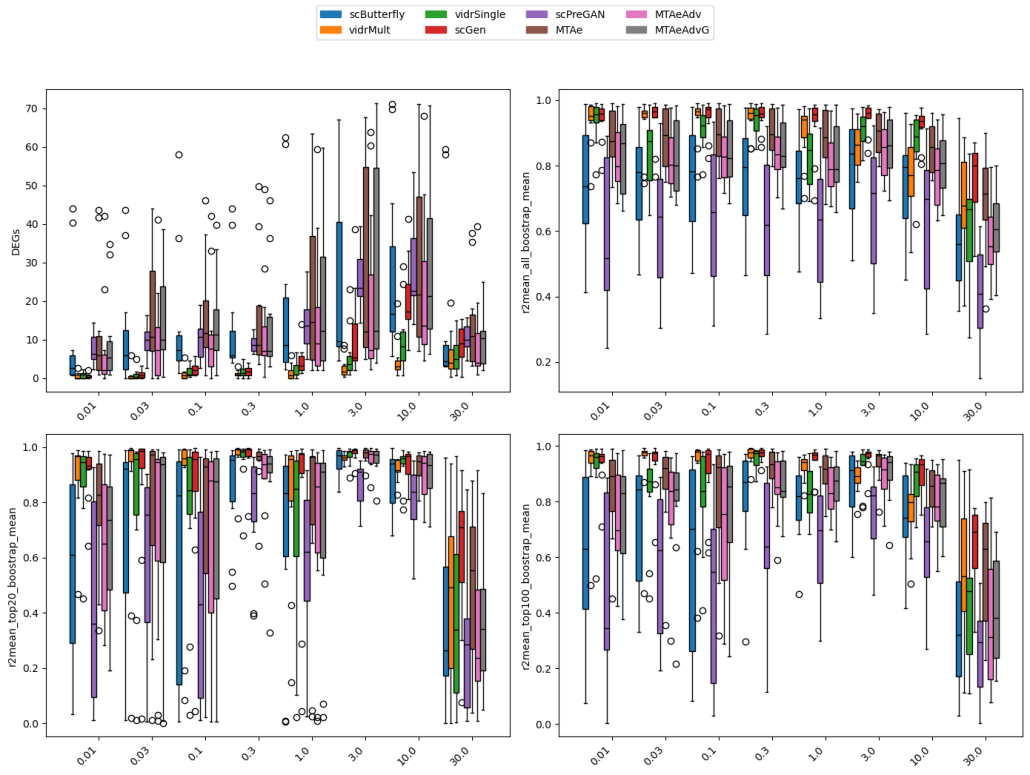


Figure 15: Baseline metrics of multi-task and literature models for the Nault et al. [8,9] dataset across dosages

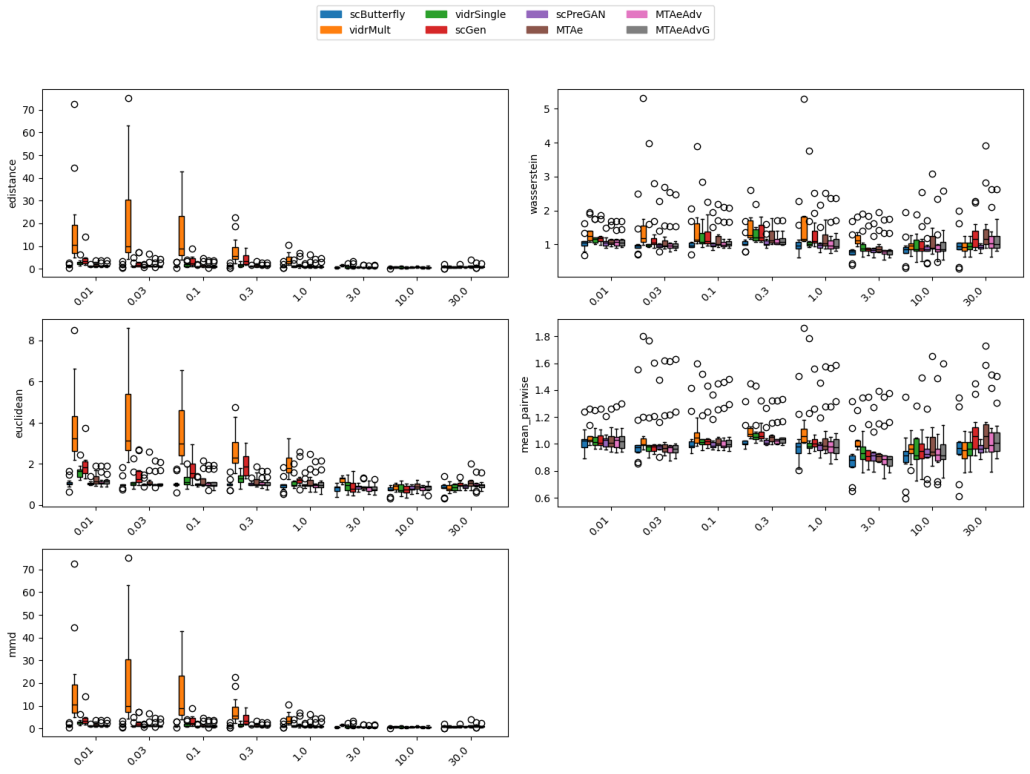


Figure 16: Distance metrics of multi-task and literature models for the Nault et al. [8,9] dataset across dosages

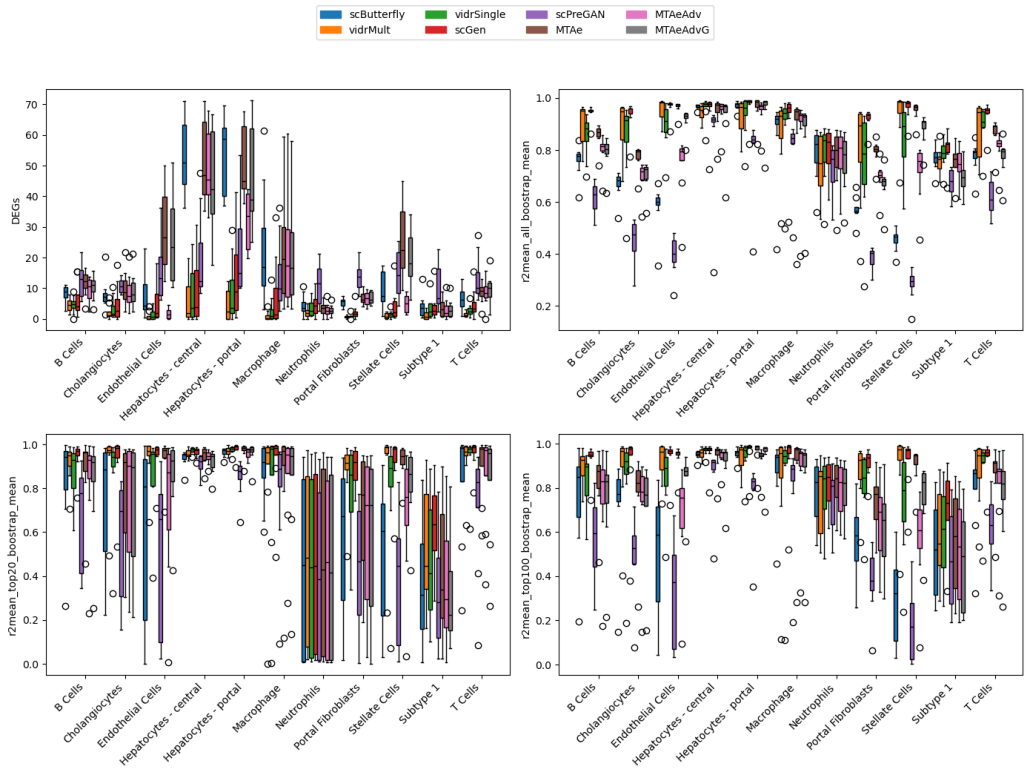


Figure 17: Baseline metrics of multi-task and literature models for the Nault et al. [8,9] dataset across cell types

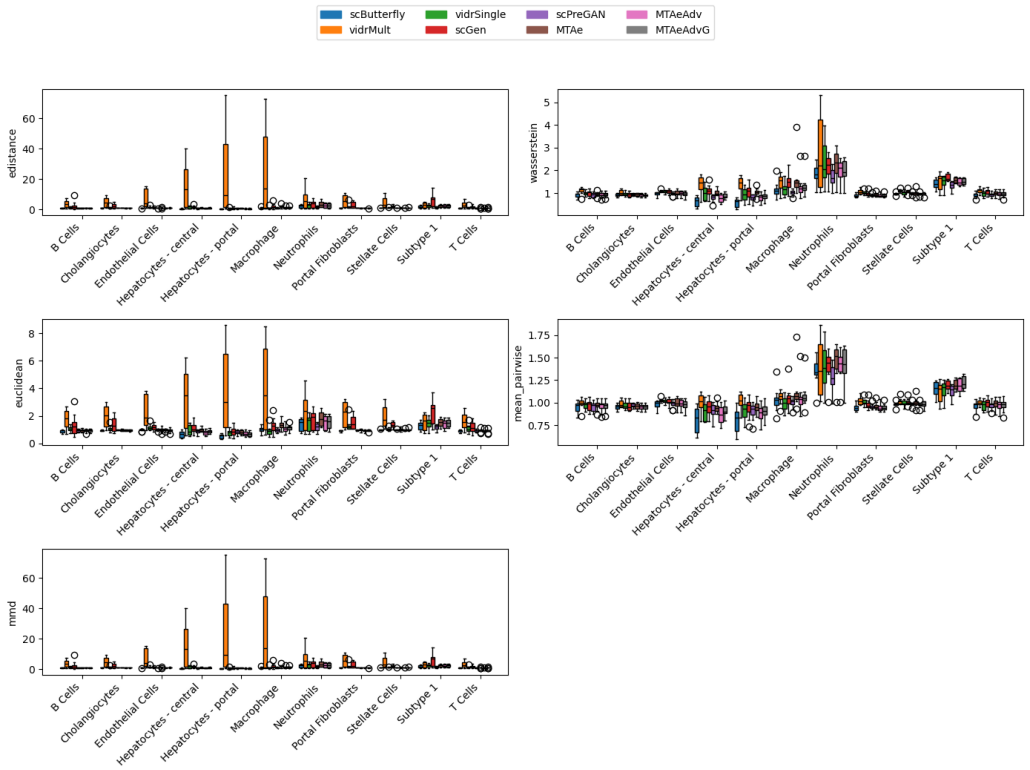


Figure 18: Distance metrics of multi-task and literature models for the Nault et al. [8,9] dataset across cell types

6.1 Knowledge transfer

which tasks and why they are important?

6.2 TODO

- batch effect
- interpretability
- explainability
- integration with multiple omics

7 Conclusions

We have validated the potential of scButterfly to perturbation modeling with the multi-dosage dataset of Nault et al. [8, 9], an addition of the author's study that is based only on the dataset of Kang et al. [6]. We have proposed multi-task architectures that can be used for perturbation modeling, and we have benchmarked them against the state-of-the-art models in the field. The results show that our models outperform or are comparable to the state-of-the-art models in the field.

8 Future work

Α Ακρωνύμια και συντομογραφίες

LAN Local Area Network

References

[1] Yichuan Cao, Xiamiao Zhao, Songming Tang, Qun Jiang, Sijie Li, Siyu Li, and Shengquan Chen. scButterfly: A versatile single-cell cross-modality translation method via dual-aligned variational autoencoders. 15(1):2973.

[2] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 2018. https://distill.pub/2018/feature-wise-transformations.

[3] George I. Gavriilidis, Vasileios Vasileiou, Aspasia Orfanou, Naveed Ishaque, and Fotis Psomopoulos. A mini-review on perturbation modelling across single-cell omic modalities. 23:1886–1896.

[4] Lukas Heumos, Yuge Ji, Lilly May, Tessa Green, Xinyue Zhang, Xichen Wu, Johannes Ostner, Stefan Peidli, Antonia Schumacher, Karin Hrovatin, et al. Pertpy: an end-to-end framework for perturbation analysis. *bioRxiv*, pages 2024–08, 2024.

[5] Yuge Ji, Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. Machine learning for perturbational single-cell omics. *Cell Systems*, 12(6):522–537, June 2021.

[6] Omar Kana, Rance Nault, David Filipovic, Daniel Marri, Tim Zacharewski, and Sudin Bhattacharya. Generative modeling of single-cell gene expression for dose-dependent chemical perturbations. 4(8):100817.

[7] Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. scGen predicts single-cell perturbation responses. 16(8):715–721.

[8] Rance Nault, Kelly A Fader, Sudin Bhattacharya, and Tim R Zacharewski. Single-nuclei rna sequencing assessment of the hepatic effects of 2, 3, 7, 8-tetrachlorodibenzo-p-dioxin. *Cellular and Molecular Gastroenterology and Hepatology*, 11(1):147–159, 2021.

[9] Rance Nault, Satabdi Saha, Sudin Bhattacharya, Jack Dodson, Samiran Sinha, Tapabrata Maiti, and Tim Zacharewski. Benchmarking of a bayesian single cell rnaseq differential gene expression test for dose–response study designs. *Nucleic acids research*, 50(8):e48–e48, 2022.

[10] Xin Tang, Jiawei Zhang, Yichun He, Xinhe Zhang, Zuwan Lin, Sebastian Partarrieu, Emma Bou Hanna, Zhaolin Ren, Hao Shen, Yuhong Yang, Xiao Wang, Na Li, Jie Ding, and Jia Liu. Explainable multi-task learning for multi-modality biological data analysis. 14(1):2546.

[11] Xiajie Wei, Jiayi Dong, and Fei Wang. scPreGAN, a deep generative model for predicting the response of single-cell expression to perturbation. 38(13):3377–3384.

[12] Yu Zhang and Qiang Yang. A Survey on Multi-Task Learning.