Aristotle University of Thessaloniki
Faculty of Engineering
School of Electrical and Computer Engineering
Department of Electronics and Computer Engineering

Diploma Thesis

Multi-task learning in perturbation modeling

Theodoros Katzalis

Supervisors:
Prof. Pericles Mitkas
Professor at Aristotle University of Thessaloniki

Dr. Fotis Psomopoulos
Senior Researcher at the Institute of Applied Biosciences

June 7, 2025

Abstract

Advanced single-cell technologies have provided new insights on cellular responses to perturbations, with significant potential for translational medicine. However, the inherent complexity of biological systems and the technical limitations of the experimental protocols present challenges for many proposed computational methods to algorithmically capture the perturbation mechanisms. Multi-task learning is one of the methods that have been left unexplored in this field. In this study, we aim to bridge this gap by unraveling its potential in single-cell perturbation modeling. We have developed a multi-task autoencoder achitecture that predicts perturbed transcriptomic profiles for multiple perturbations achieving state-of-the-art performance while exhibiting greater scalabiliy and efficiency compared to existing methods.

Contents

1 Introduction

The advent of single-cell technologies has enabled the study of the cellular heterogeneity at the cellular resolution, opening new avenues for understanding the cellular mechanisms and their responses to perturbations. However, the perturbation space is vast, and experimentally exploring combinations would be infeasible and costly [5, 8]. This has motivated the development of computational methods to model this space, enabling extrapolation to unseen scenarios through in silico experimentation. The field of deciphering and predicting the effects of external stimuli (gene knockouts, drug dosages, temperature changes, etc.) is referred to as perturbation modeling, and it plays a crucial role in disease mechanism discovery and therapeutic target identification [6].

One of the main objectives of perturbation modeling is the out-of-distribution detection (OOD) [3], 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.

Datasets used for perturbation modelling are often highly noisy and sparse due to the inherent limitations of single-cell technologies. For example, dropout events are likely to occur, leading to many zeros in the expression profiles as a failure of detecting lower expression levels. The data is also high-dimensional, typically consisting of thousands of cells profiled across hundreds or thousands of features (e.g., gene expression levels in transcriptomics), whichm on the other way, enables fine-grained analysis of cellular responses [6]. The perturbation response itself is non-linear and complex, depending not only on the nature of the perturbation but also on the cellular context, including cell type, microenvironment, genetic background, and temporal dynamics.

Machine learning methods, particularly deep learning, have shown promise in addressing this complexity by leveraging their generative capacity, made possible by the recent surge in high-throughput single-cell data [3]. More specifically there is a growing trend toward leveraging large language models (LLMs) in the field. A recent survey by Szalata et al. [13] highlights this as a promising yet still immature research direction. Key challenges include the lack of standardized evaluation frameworks, model instabilities, insufficiently diverse datasets, and the absence of sequential structure analogous to positional embeddings in natural language processing. In contrast, autoencoder architectures and their variants have already demonstrated strong performance, while offering notable advantages in terms of resource efficiency and reduced computational complexity.

Based on the core deep learning concept of manifold hypothesis, autoencoder architectures aim to learn a low-dimensional representation of the data, capturing the underlying structure of the perturbation response. This is achieved by the encoder-decoder architecture, where the encoder compresses the input data into a lower-dimensional space, while the decoder attempts to reconstruct the original input. This compression can yield biologically meaningful features, resulting in a more interpretable and efficient representation of the data, which can be useful for downstream tasks such as out-of-distribution detection [3].

However, the non-linearity of deep learning models presents another challenge in balancing predictive accuracy with interpretability. This trade-off remains a key milestone in the field, and many recent efforts have aimed to address it through causal machine learning approaches. Additional limitations in the data space—such as batch effects and confounding covariates—also hinder prediction accuracy. To mitigate these issues and improve generalization, recent studies have focused on integrative single-cell omics approaches, including spatial data integration, to provide a more holistic view of cellular responses [3].

UnitedNet [14] is an explainable framework based on an autoencoder architecture that have addressed the aforementioned limitations while showing the potential of multi-task learning in multi-omics tasks such as cross modal prediction and cell type classification. We aim to extend this approach to perturbation modeling.

2 Current single-cell perturbation modeling methods

In the literature body, there are several approaches for predicting single-cell perturbation responses. An overview of the models on perturbation modeling can be found on this study [3]. To compare our multi-task method, we have chosen the models of scGen [9], scVIDR [7], scPreGAN [15], and scButterfly [1]. One of the key tasks for these models is out-of-distribution detection. A typical dataset for this task consists of transcriptomic profiles obtained from single-cell sequencing technologies (scRNA-seq) from multiple cell types in both control and perturbed conditions. The objective is to predict the perturbed gene expression profile of a held-out (unseen) cell type, given its control-state profile. To achieve this, the model must learn the perturbation effect from the remaining cell types in both conditions. The performance of all of these models will serve as a baseline to evaluate our multi-task learning architectures.

2.1 scGen

scGen's architecture is based on a variational autoencoder (VAE) that learns a probabilistic latent space representation of the gene expression profiles. The perturbation effect is modeled as a vector δ, calculated as the mean of the differences between the latent vectors of the perturbed and control gene expression profiles. Then, the latent perturbed gene expression profile of a held-out cell type, $\hat{z}$, is generated by adding this perturbation vector, δ, to the latent vector of the control profile, z, using $\hat{z} = z + \delta$. Finally, the perturbed gene expression is obtained by decoding the generated latent vector, $\hat{z}$, using the decoder of the VAE. This approach allows for the generation of new perturbed profiles by manipulating the latent space representation using vector arithmetic.

2.2 scVIDR

A key limitation of scGen is the absence of explicit cell-type-specific modeling, which can reduce its ability to generalize to unseen cell types with distinct perturbation responses. scVIDR addresses this by incorporating cell-type-aware perturbation estimation. Rather than computing a single global perturbation vector δ based only on the condition labels, scVIDR fits a linear regression model that captures how perturbation vectors vary across cell types. For each training cell type i, the perturbation vector is defined as $\delta_i = \hat{z}_i - z_i$, where z_i, and $\hat{z}_i$ are the mean latent representation of the control, and perturbed cells of type i respectively. A linear model is then trained to predict $\hat{\delta}_i$ from z_i, i.e., $\hat{\delta}_i = f(z_i)$.

Once trained, this model can predict the perturbation vector δ_A for an unseen cell type A, using only its control-state latent representation z_A, i.e., $\hat{\delta}_A = f(z_A)$. This cell-type-aware prediction improves generalization by allowing the model to tailor the perturbation response based on the control-state context of each cell type.

scVIDR can also predict the gene expression profile for multiple dosages. Similarly, scVIDR fits a linear regression model to predict the perturbation vector $\hat{\delta}_c$ across cell types, but in this case, the conditions are the lowest and the highest dosage. Intermediate dosages are then calculated by log linearly interpolating on the $\hat{\delta}_c$.

Regarding interpretability, the bottleneck of the non-linear mapping from the latent space to the gene expression space is replaced by a linear one, utilizing a sparse linear regression model. This is approximated by a weight matrix $\hat{W}_{VAE}$, with dimensions MxG where M is the number of latent variables and G is the number of genes. Then this matrix is used to examine the contribution of the latent variables to the gene expression profile, using the following equation:

$$\text{gene score} = \hat{\delta}_c^T \hat{W}_{VAE}$$

A higher gene score indicates a bigger change at the expression level of the gene if the dosage increases.

2.3 scPreGAN

scPreGAN integrates an autoencoder with a generative adversarial network (GAN) framework to predict single-cell RNA-seq (scRNA-seq) data under perturbations. The architecture consists of a shared encoder and two generators, one for each condition (control and perturbed). To align the generated distributions with the real data, the model employs two discriminators, each associated with a specific condition.

The encoder, which is shared across both conditions, learns a perturbation-free latent representation that captures high-level biological features common to both states. The generators then incorporate condition-specific perturbation effects to reconstruct the gene expression profiles from the latent space. The discriminators are trained to distinguish between real and generated samples, while the generators are optimized adversarially to produce realistic reconstructions that fool their respective discriminators.

2.4 scButterfly

scButterfly is a generative adversarial model built on a dual-aligned variational autoencoder (VAE) architecture, designed for cross-modal translation in single-cell data. The model has demonstrated strong performance in translating between transcriptomic and chromatin accessibility profiles, as well as between transcriptomic and proteomic data.

Its architecture consists of two VAEs, each pretrained on a specific modality, and a translator component that aligns the latent spaces of the two encoders. The translator is composed of two neural networks, one per modality, each modeling a Gaussian distribution in the latent space. These networks take the encoder's latent representation as input, sample from the modeled distribution, and pass the sample to the decoder of the other modality, enabling cross-modal generation. After VAE pretraining, the translator is trained to align the latent spaces such that biologically meaningful translation across modalities can be achieved.

Although scButterfly isn't primarily designed for perturbation modeling, the study has demonstrated its potential, by treating control and perturbed expression profiles as two modalities. One of its limitations is the narrow evaluation scope, as it has been tested only on the case of human peripheral blood mononuclear cells (PBMCs) stimulated by interferon beta (IFN-b) [7].

3 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 [16]. 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 information, 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.

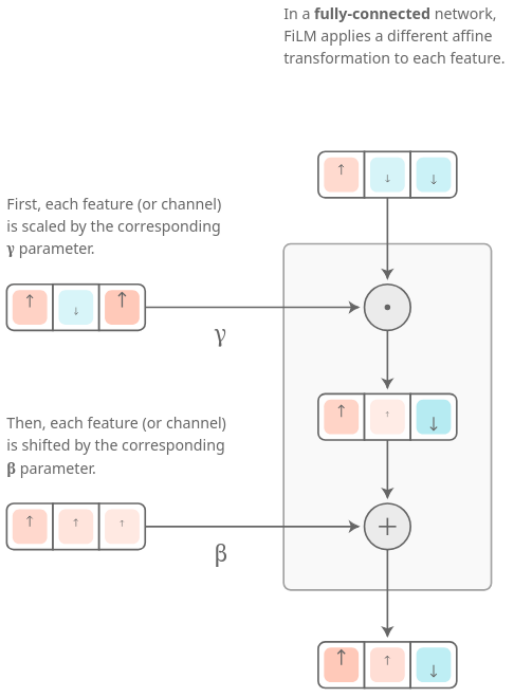


Figure 1: Illustration of the feature-wise transformation [2]

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 vector. Our architecture is built around an autoencoder, where task-specific conditioning —

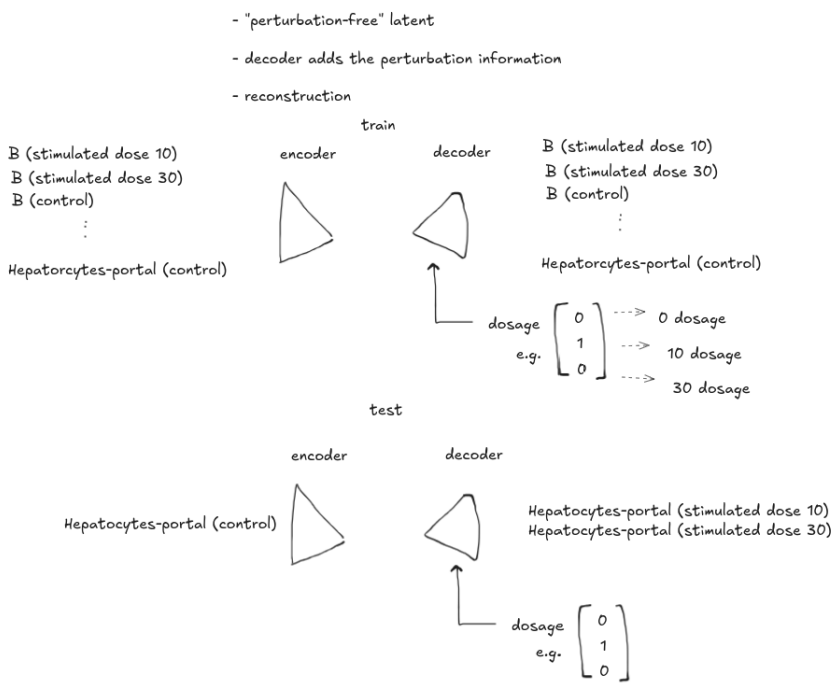


Figure 2: Illustration of the multi-task architecture. The encoder is shared across all tasks, while the decoder is conditioned by the task-specific FiLM layers.

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:

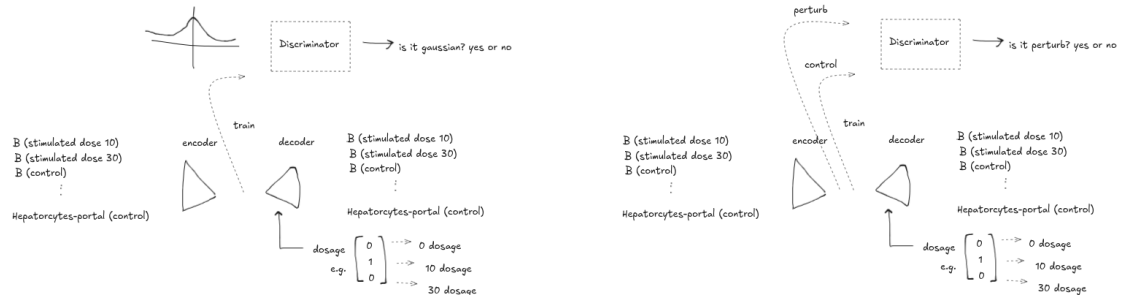
$$\mathcal{L}_{\text{recon}} = \frac{1}{N} \sum_{i=1}^{N} ||x_i - \hat{x}_i||^2$$

, 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. For the `MTAeAdv` architecture, we have attempted to explicitly model a perturbation-free la-



(a) (b)

Figure 3: Adversarial autoencoders

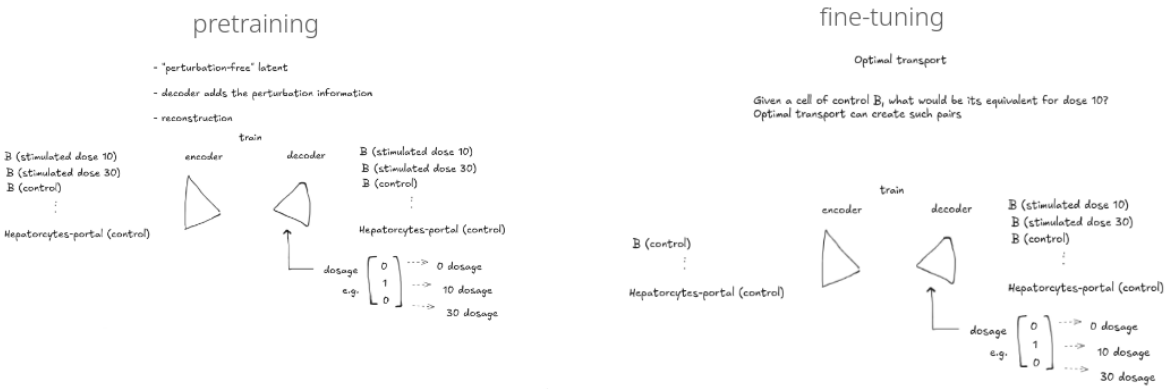


Figure 4: Using optimal transport to fine-tune the MTAe architecture (MTAePlusOT)

tent 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`.

The last set of variations involves the inclusion of Variational Autoencoders (VAEs). The

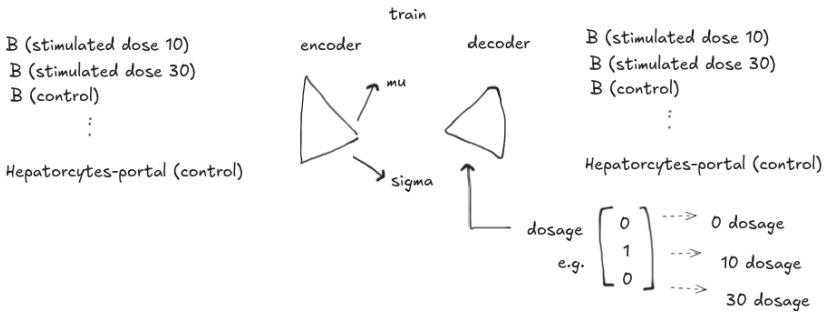


Figure 5: VAE

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.

4 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. [7]), and a multi-perturbation dataset, where liver cells have been stimulated by multiple doses of tetrachlorodibenzo-p-dioxin (TCDD) in vivo (Nault et al. [10, 11]).

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 [10, 11], 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.

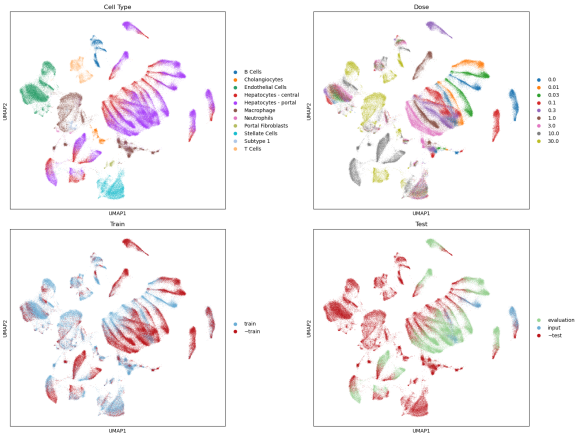
For this comparison, we have used the count of differentially expressed genes (DEGs), the R^2 of all the highly variable genes (HVGs), and the top 100 most variable ones. To complement the evaluation, we have calculated a set of five distance metrics (euclidean, edistance, wasserstein, mean pairwise, 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.

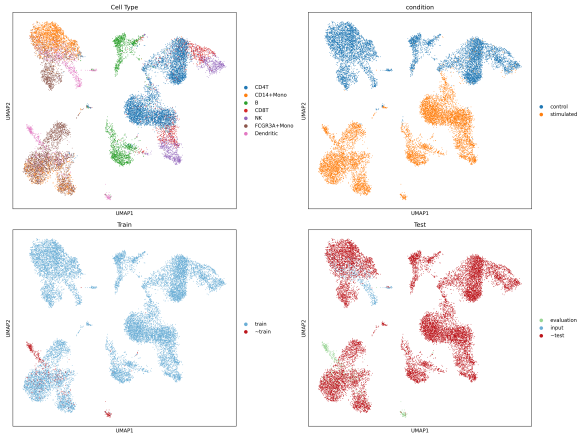
To rank the models, since there could be conflicting cases between metrics, where one model could be better than the other, per model's metric we averaged them across all the experiments. Then we scaled them to the range of 0-1 with the following formula:

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

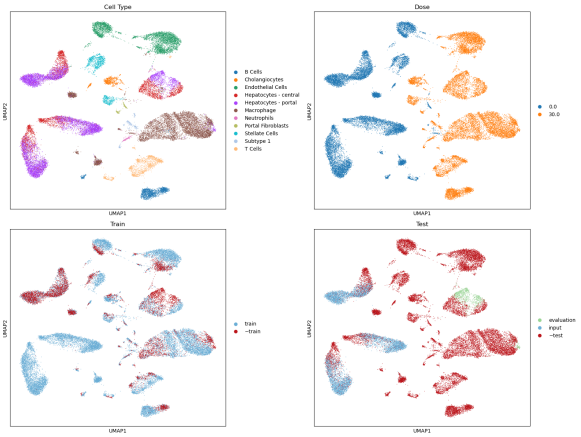
, that can track how a metric deviates from the best one. Then we summed all the metrics, giving a score (penalty) to each model. The model with the lowest score is considered the best one.



(a) Nault et al. [10, 11]



(b) Kang et al. [7]



(c) Example of single dose split of Nault et al. [7] for the dosage $30\mu g/kg$

Figure 6: UMAP representations of data split

5 Results

Initially, we will benchmark only our multi-task variations to filter out the most promising models figs. 10, 15 and 30 to 33. 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. [10, 11] dataset.

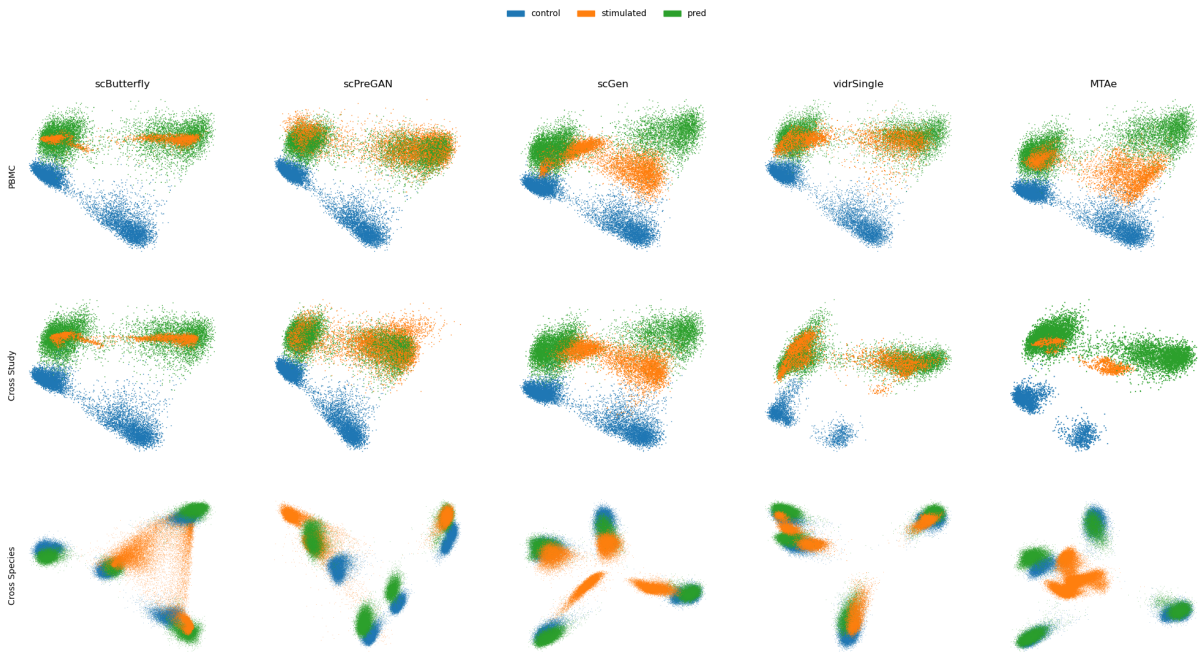


Figure 7: PCA dimensionality reduction of the real unperturbed data, the real perturbed data and the predicted perturbed data.

5.1 Kang et al.

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. [7] 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 2: Kang et al. [7]

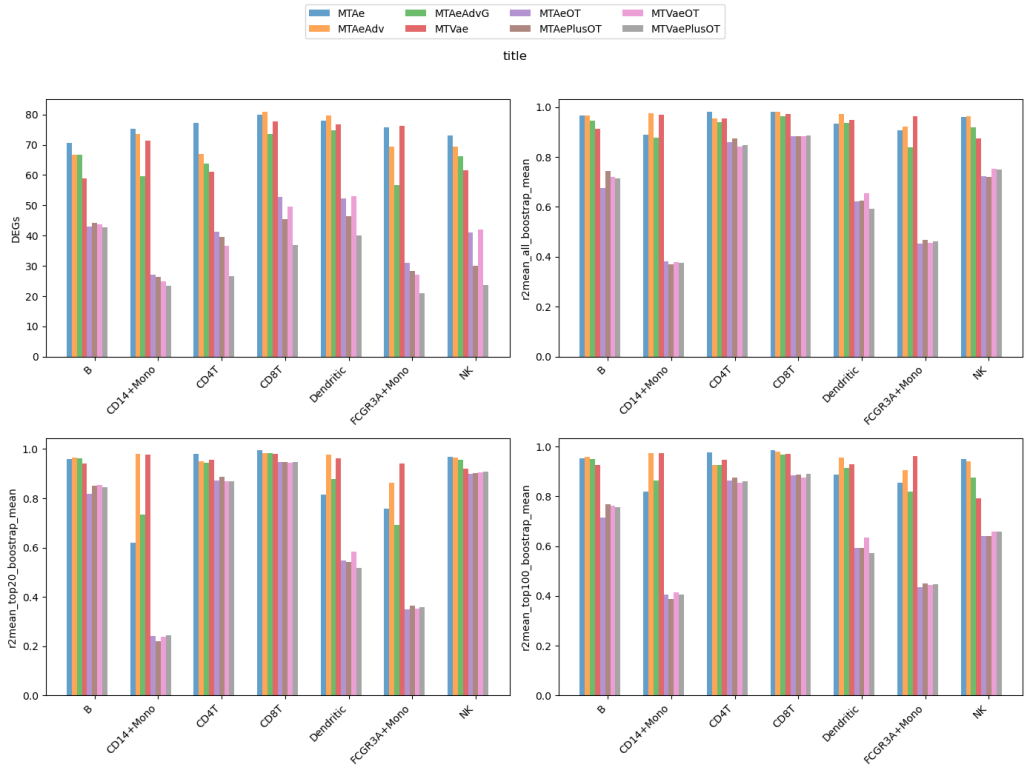


Figure 8: Baseline metrics of multi-task models for the Kang et al. [7] dataset across cell types

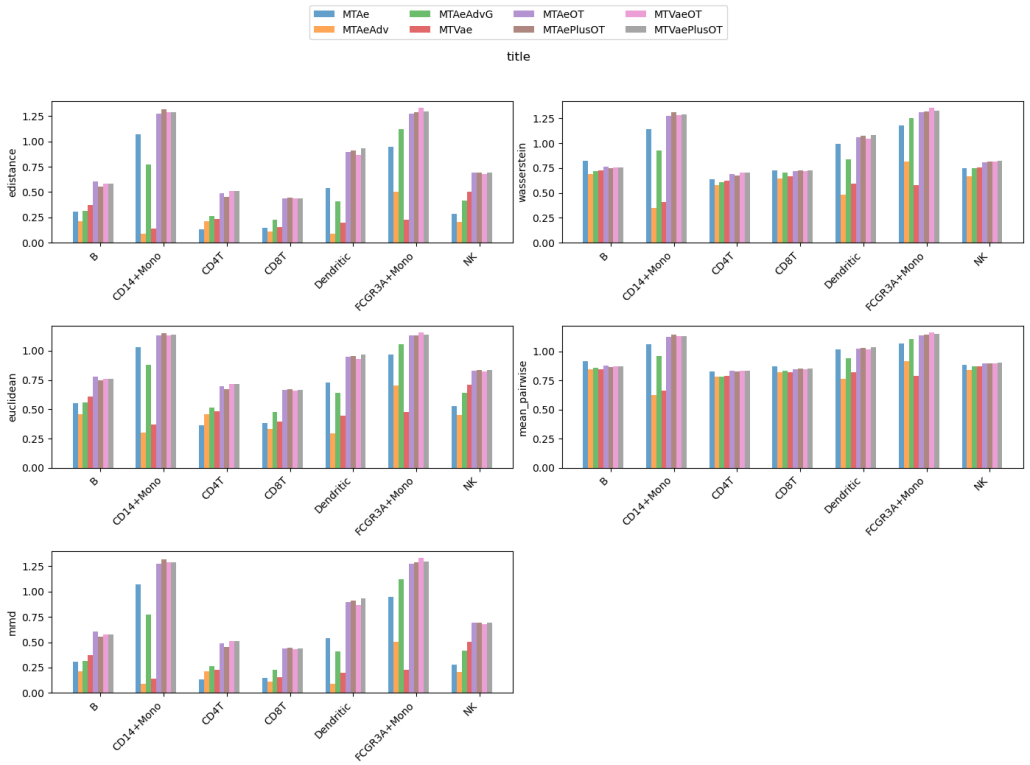


Figure 9: Distance metrics of multi-task models for the Kang et al. [7] dataset across cell types

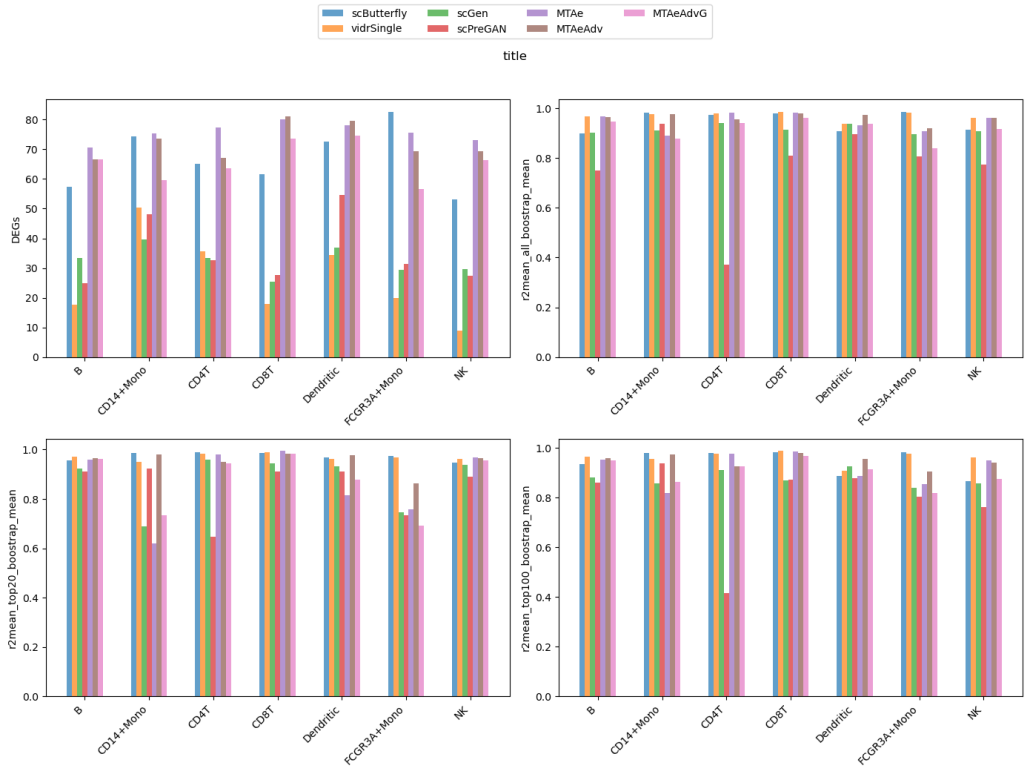


Figure 10: Baseline metrics of multi-task and literature models for the Kang et al. [7] dataset across cell types

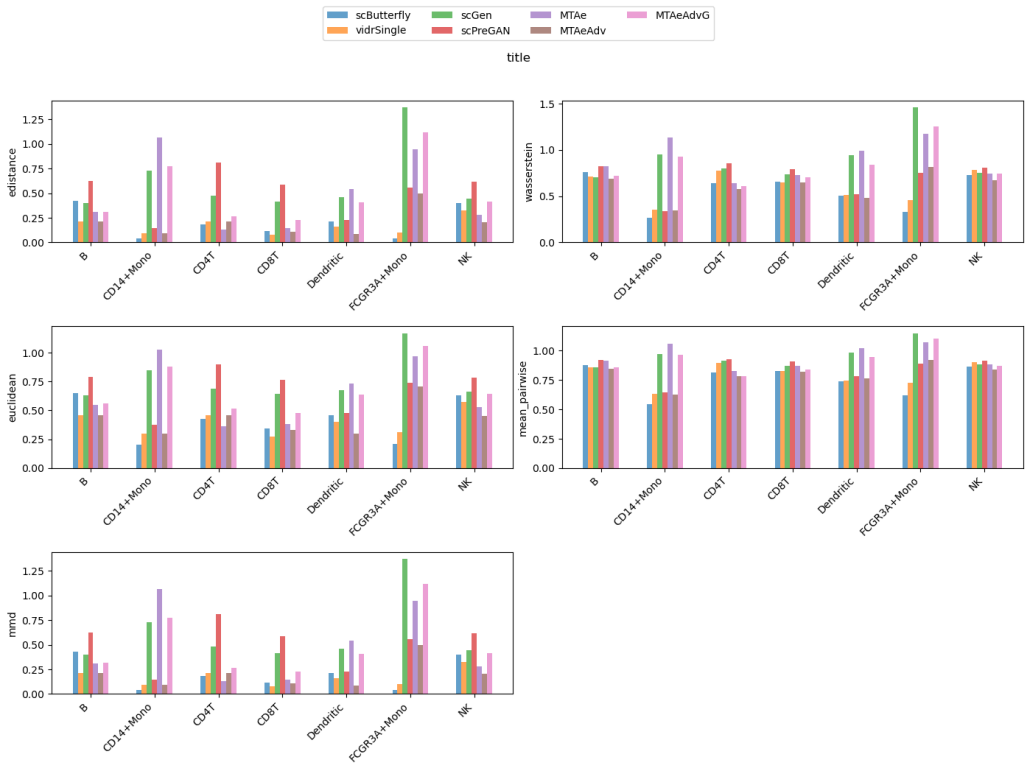


Figure 11: Distance metrics of multi-task and literature models for the Kang et al. [7] dataset across cell types

Figure 12

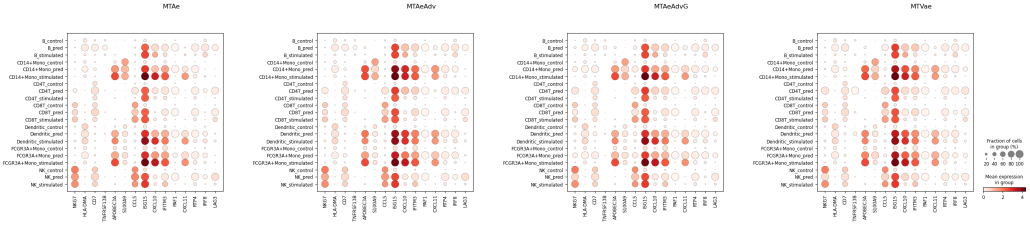


Figure 13

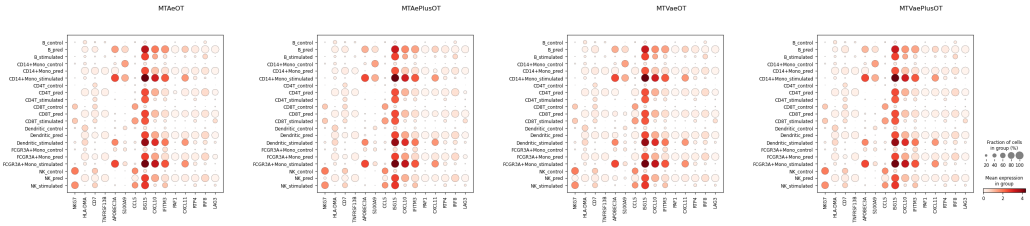


Figure 14

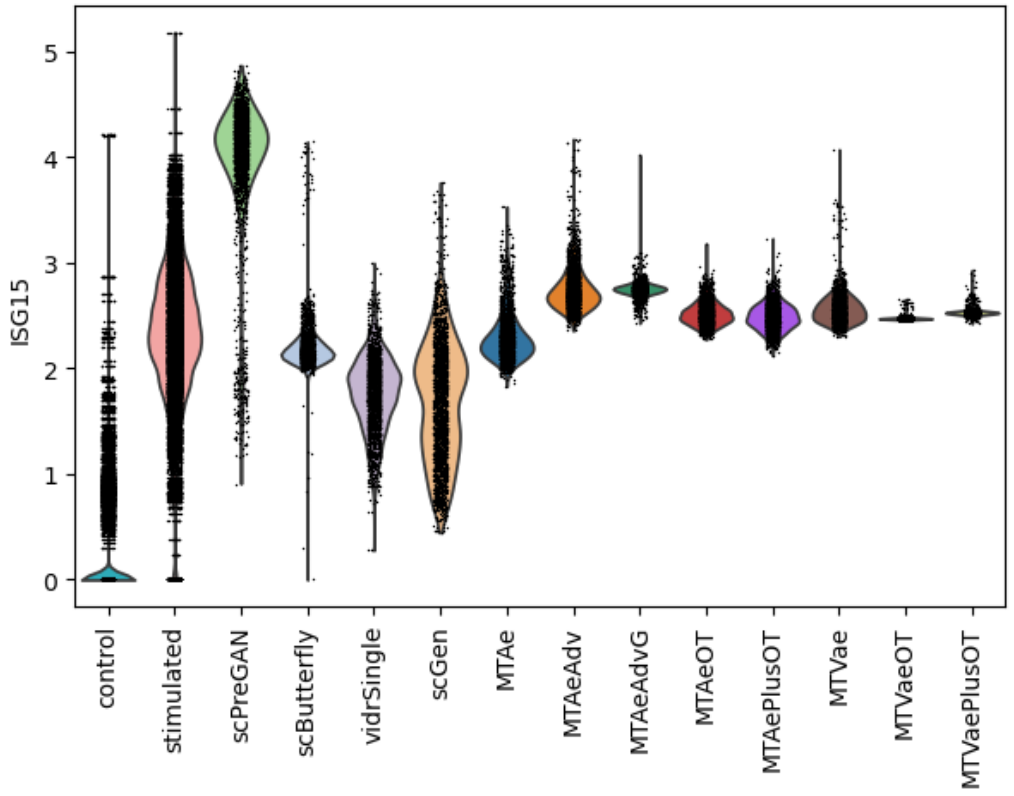


Figure 15

5.2 Cross-study

model	DEGs	R^2_{HVG}	R^2_{HVG20}	R^2_{HVG100}	Euc	Was	E-dist	MPD	MMD
MTAe	0.066	0.224	0.369	0.223	0.446	0.733	0.528	0.746	0.446
	(64.048)	(0.902)	(0.882)	(0.904)	(0.344)	(0.794)	(0.538)	(0.876)	(0.344)
MTAeAdv	0.166	0.161	0.277	0.158	0.228	0.383	0.337	0.347	0.228
	(59.000)	(0.922)	(0.906)	(0.922)	(0.210)	(0.610)	(0.436)	(0.773)	(0.210)
MTAeAdvG	0.230	0.227	0.393	0.217	0.382	0.563	0.476	0.558	0.382
	(55.762)	(0.901)	(0.876)	(0.906)	(0.305)	(0.704)	(0.510)	(0.828)	(0.305)
MTAeOT	0.674	1.000	1.000	1.000	0.992	0.964	1.000	0.970	0.992
	(33.381)	(0.658)	(0.720)	(0.687)	(0.683)	(0.916)	(0.789)	(0.934)	(0.683)
MTAePlusOT	0.893	0.945	0.961	0.946	1.000	1.000	0.996	1.000	1.000
	(22.333)	(0.675)	(0.730)	(0.702)	(0.687)	(0.935)	(0.787)	(0.942)	(0.687)
MTVae	0.194	0.208	0.298	0.154	0.258	0.346	0.383	0.305	0.258
	(57.619)	(0.907)	(0.901)	(0.923)	(0.228)	(0.590)	(0.461)	(0.763)	(0.228)
MTVaeOT	0.737	0.948	0.958	0.935	0.930	0.915	0.960	0.924	0.930
	(30.238)	(0.674)	(0.731)	(0.705)	(0.644)	(0.890)	(0.768)	(0.922)	(0.644)
MTVaePlusOT	0.970	0.967	0.977	0.972	0.973	0.960	0.982	0.960	0.973
	(18.476)	(0.668)	(0.726)	(0.694)	(0.670)	(0.914)	(0.780)	(0.932)	(0.670)
scButterfly	0.000	0.060	0.000	0.037	0.184	0.226	0.273	0.214	0.184
	(67.381)	(0.954)	(0.977)	(0.956)	(0.182)	(0.527)	(0.402)	(0.739)	(0.182)
scGen	0.495	0.147	0.317	0.225	0.673	0.729	0.787	0.840	0.673
	(42.429)	(0.927)	(0.896)	(0.903)	(0.485)	(0.792)	(0.676)	(0.901)	(0.485)
scPreGAN	0.483	0.602	0.480	0.544	0.667	0.505	0.780	0.597	0.667
	(43.000)	(0.783)	(0.854)	(0.814)	(0.481)	(0.674)	(0.672)	(0.838)	(0.481)
vidrSingle	1.000	0.000	0.022	0.000	0.000	0.000	0.000	0.000	0.000
	(16.952)	(0.973)	(0.971)	(0.966)	(0.068)	(0.408)	(0.257)	(0.684)	(0.068)

Table 3: Cross-study

model	score	baseline score	distance score
vidrSingle	1.022251	1.022251	0.000000
scButterfly	1.177201	0.097338	1.079863
MTAeAdv	2.287284	0.763252	1.524032
MTVae	2.403579	0.853503	1.550076
MTAeAdvG	3.430162	1.068459	2.361703
MTAe	3.780031	0.881922	2.898110
scGen	4.886600	1.184490	3.702110
scPreGAN	5.326055	2.109282	3.216773
MTVaeOT	8.237092	3.578456	4.658636
MTAeOT	8.593334	3.674221	4.919113
MTVaePlusOT	8.732940	3.885585	4.847355
MTAePlusOT	8.741812	3.745904	4.995908

Table 4: Score Cross-Study

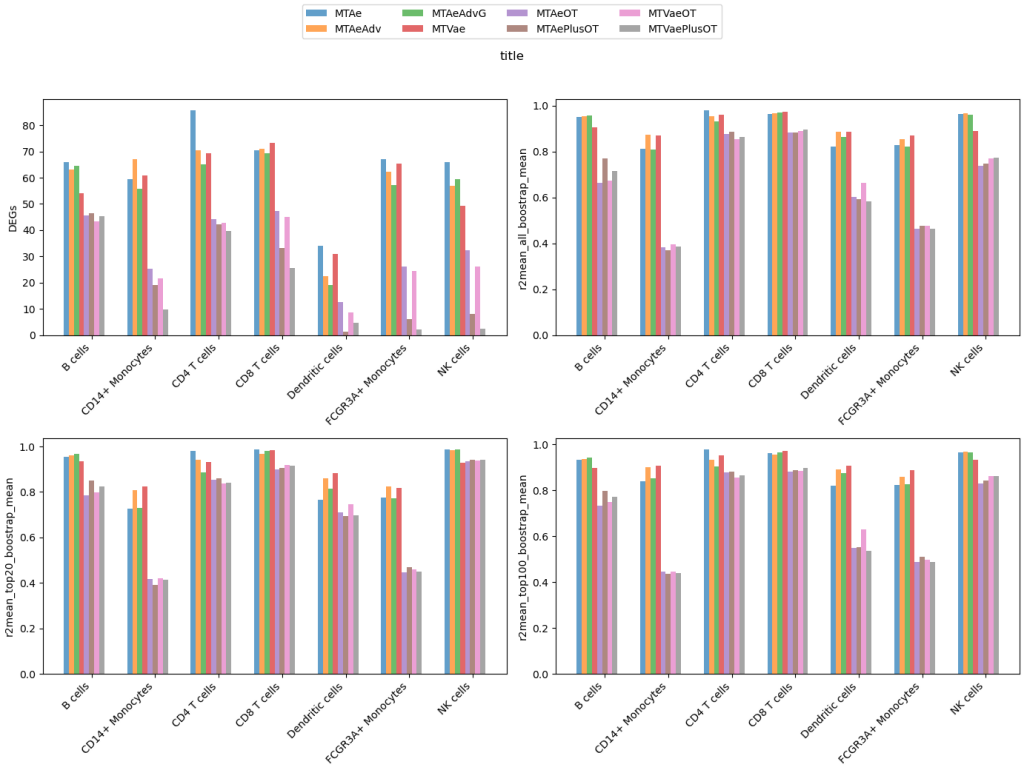


Figure 16: Baseline metrics of multi-task models for the cross-study

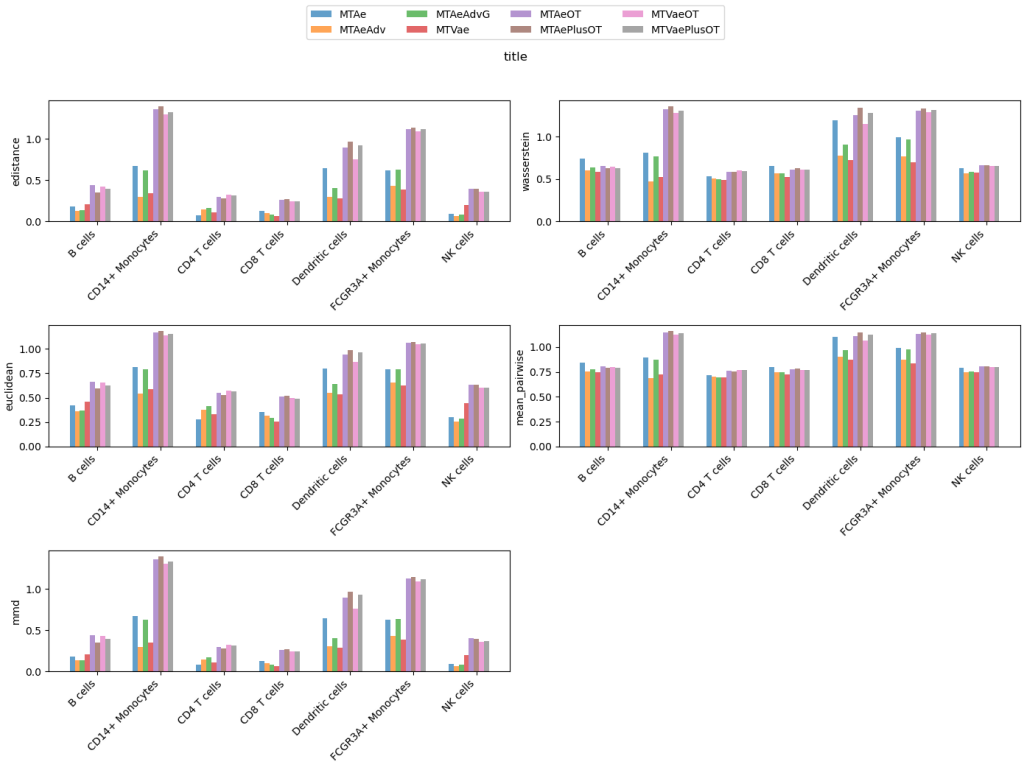


Figure 17: Distance metrics of multi-task models for the cross-study

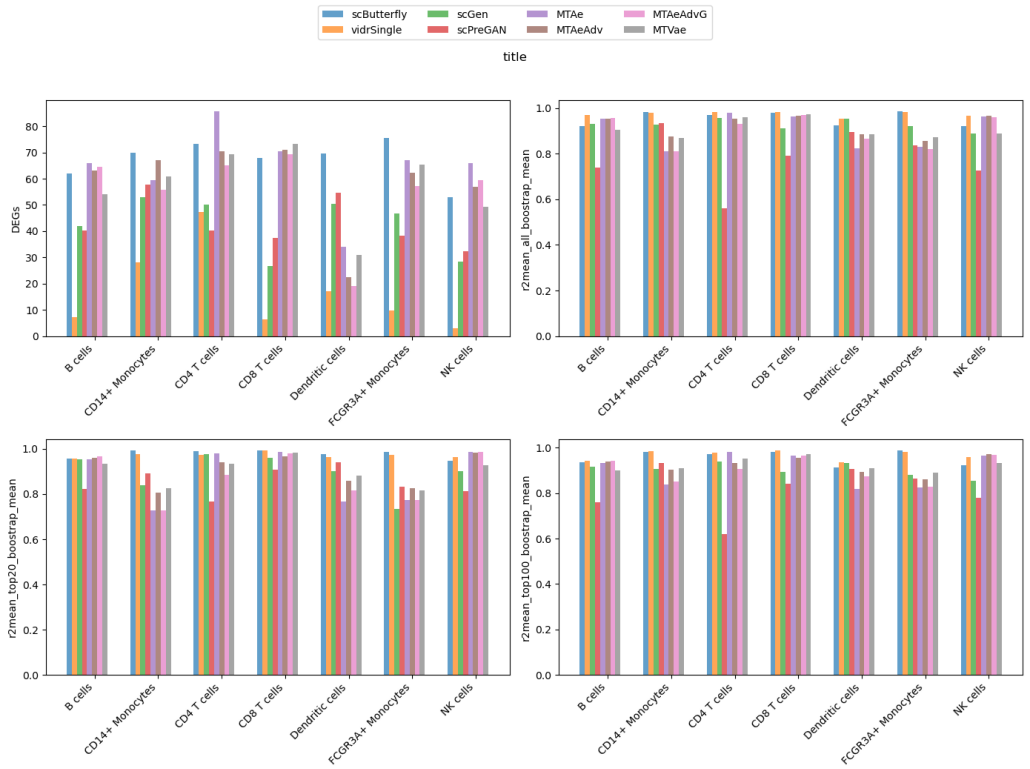


Figure 18: Baseline metrics of multi-task and literature models for the cross-study

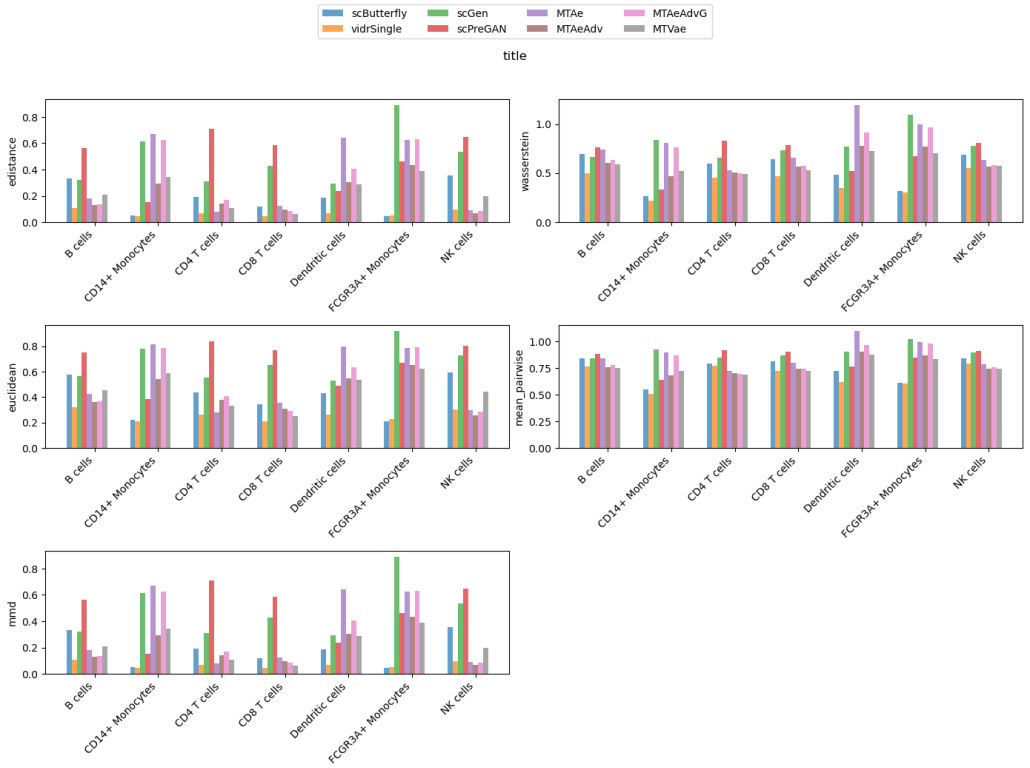


Figure 19: Distance metrics of multi-task and literature models for the cross-study

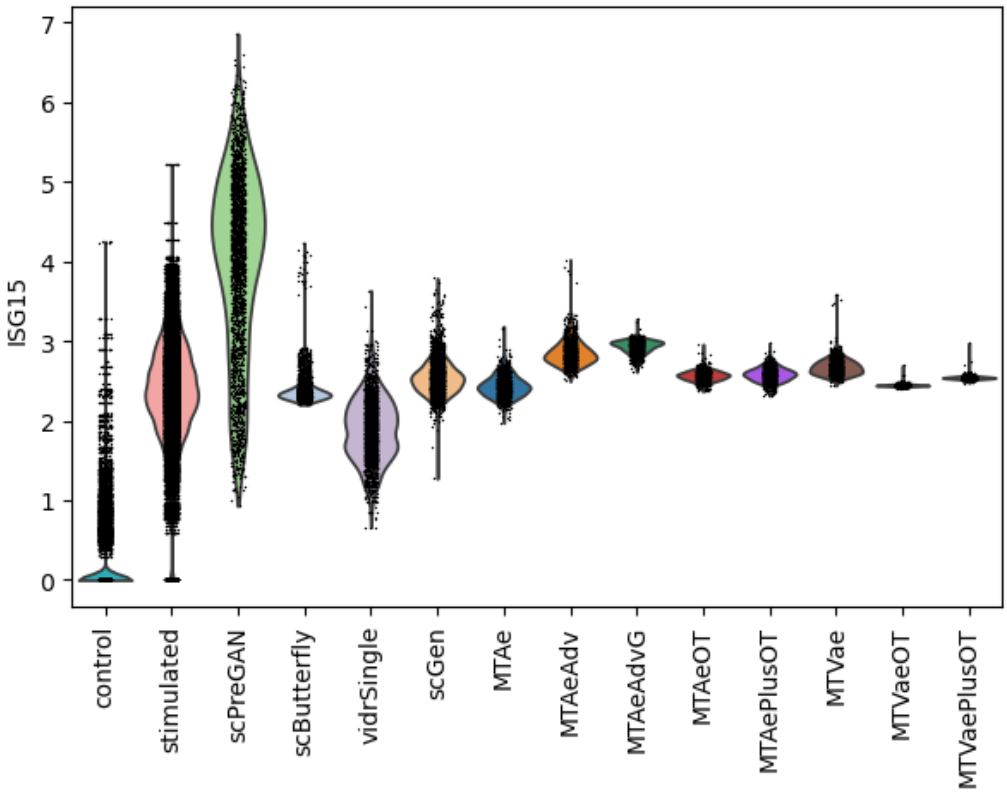

ISG15
0
1
2
3
4
5
6
7
control
stimulated
scPreGAN
scButterfly
vidrSingle
scGen
MTAe
MTAeAdv
MTAeAdvG
MTAeOT
MTAePlusOT
MTVae
MTVaeOT
MTVaePlusOT

Figure 20

5.3 Cross-species

model	DEGs	R^2_{HVG}	R^2_{HVG20}	R^2_{HVG100}	Euc	Was	E-dist	MPD	MMD
MTAe	0.000	0.316	0.451	0.519	0.036	0.104	0.055	0.161	0.036
	(16.083)	(0.740)	(0.559)	(0.481)	(0.930)	(1.008)	(0.962)	(0.995)	(0.930)
MTAeAdv	0.547	0.686	0.731	0.790	0.010	0.015	0.016	0.020	0.010
	(11.250)	(0.579)	(0.465)	(0.365)	(0.865)	(0.919)	(0.929)	(0.957)	(0.865)
MTAeAdvG	0.406	0.391	0.528	0.577	0.032	0.082	0.049	0.132	0.032
	(12.500)	(0.708)	(0.533)	(0.456)	(0.921)	(0.985)	(0.958)	(0.987)	(0.921)
MTAeOT	0.972	0.908	0.827	0.934	0.023	0.029	0.038	0.053	0.023
	(7.500)	(0.483)	(0.432)	(0.304)	(0.899)	(0.932)	(0.948)	(0.966)	(0.899)
MTAePlusOT	0.915	0.915	0.815	0.923	0.014	0.010	0.023	0.017	0.014
	(8.000)	(0.480)	(0.436)	(0.309)	(0.876)	(0.913)	(0.936)	(0.956)	(0.876)
MTVae	0.406	0.518	0.631	0.677	0.000	0.000	0.000	0.000	0.000
	(12.500)	(0.652)	(0.498)	(0.413)	(0.840)	(0.903)	(0.916)	(0.951)	(0.840)
MTVaeOT	0.981	0.917	0.855	0.940	0.022	0.026	0.035	0.046	0.022
	(7.417)	(0.479)	(0.423)	(0.302)	(0.895)	(0.929)	(0.946)	(0.964)	(0.895)
MTVaePlusOT	0.934	0.932	0.830	0.942	0.017	0.016	0.028	0.028	0.017
	(7.833)	(0.473)	(0.431)	(0.301)	(0.883)	(0.919)	(0.940)	(0.959)	(0.883)
scButterfly	0.604	0.700	0.956	0.835	0.023	0.039	0.038	0.074	0.023
	(10.750)	(0.574)	(0.389)	(0.346)	(0.899)	(0.942)	(0.948)	(0.971)	(0.899)
scGen	0.962	0.118	0.016	0.100	0.461	0.672	0.529	0.779	0.461
	(7.583)	(0.826)	(0.705)	(0.658)	(2.014)	(1.576)	(1.367)	(1.165)	(2.014)
scPreGAN	1.000	1.000	1.000	1.000	0.029	0.042	0.047	0.079	0.029
	(7.250)	(0.443)	(0.374)	(0.276)	(0.914)	(0.945)	(0.955)	(0.973)	(0.914)
vidrSingle	0.358	0.000	0.000	0.000	1.000	1.000	1.000	1.000	1.000
	(12.917)	(0.878)	(0.711)	(0.701)	(3.386)	(1.905)	(1.769)	(1.225)	(3.386)

Table 5: Cross-species

model	score	baseline score	distance score
MTAe	1.676506	1.285196	0.391310
MTAeAdvG	2.227590	1.901575	0.326015
MTVae	2.232088	2.232088	0.000000
MTAeAdv	2.826189	2.754406	0.071783
scButterfly	3.291590	3.094464	0.197126
MTAePlusOT	3.646820	3.568396	0.078425
MTVaePlusOT	3.743715	3.637807	0.105908
MTAeOT	3.806643	3.640329	0.166315
MTVaeOT	3.845143	3.693929	0.151214
scGen	4.098110	1.196070	2.902040
scPreGAN	4.225735	4.000000	0.225735
vidrSingle	5.358491	0.358491	5.000000

Table 6: Score Cross-Species

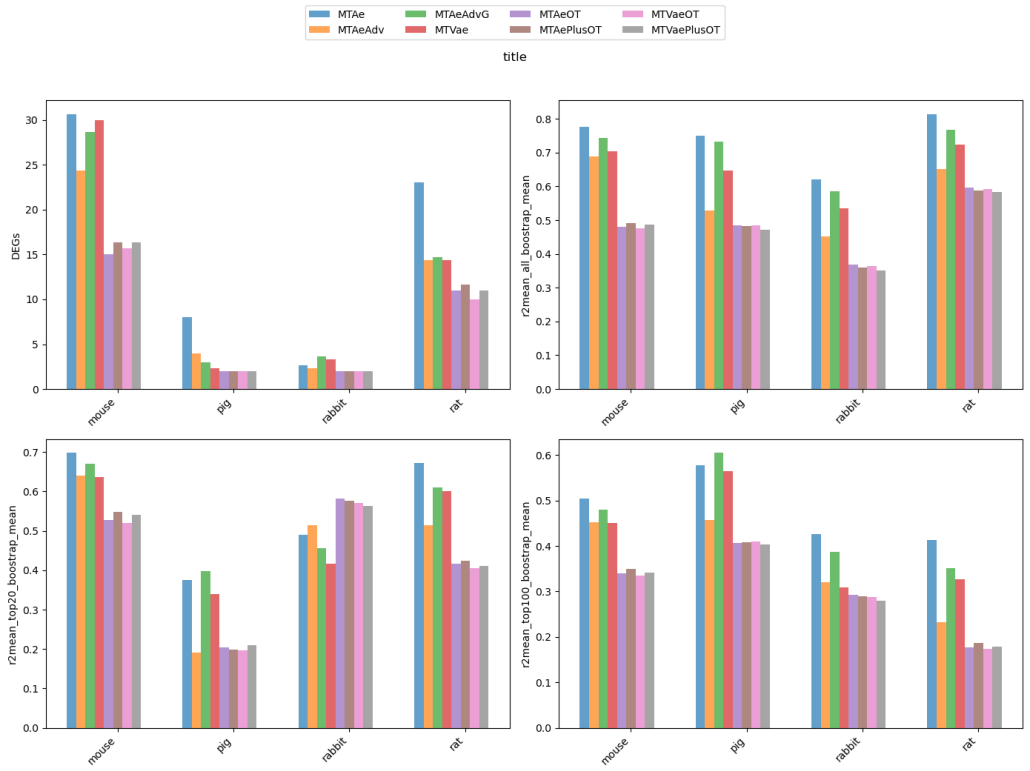


Figure 21: Baseline metrics of multi-task models for the cross-species

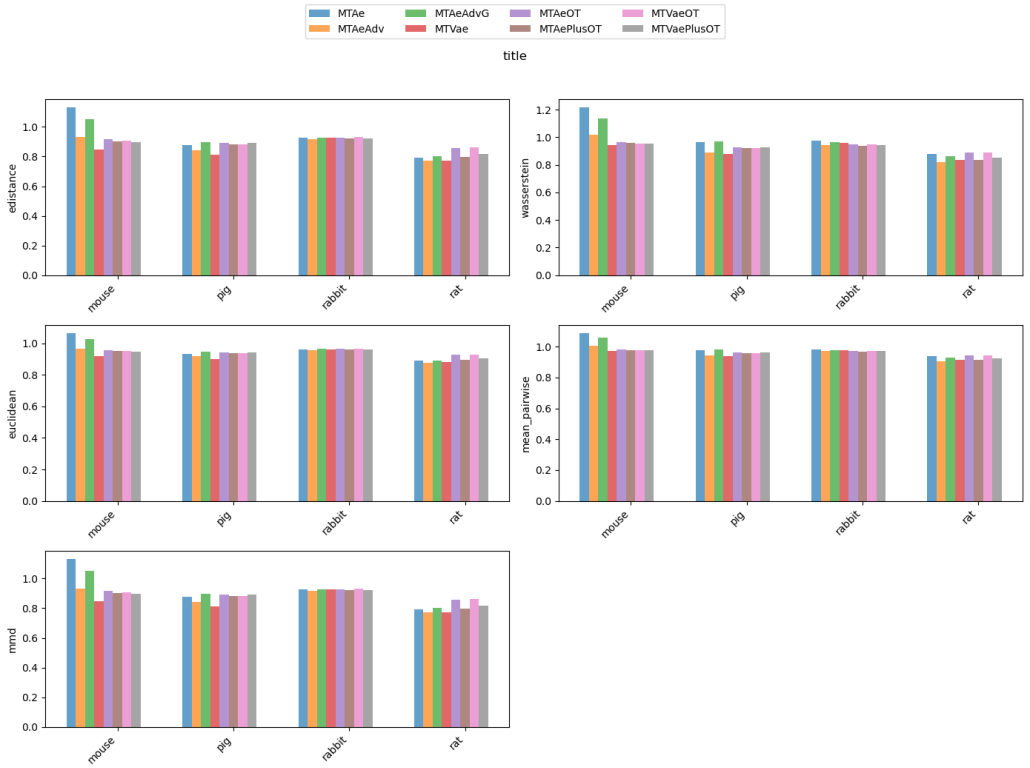


Figure 22: Distance metrics of multi-task models for the cross-species

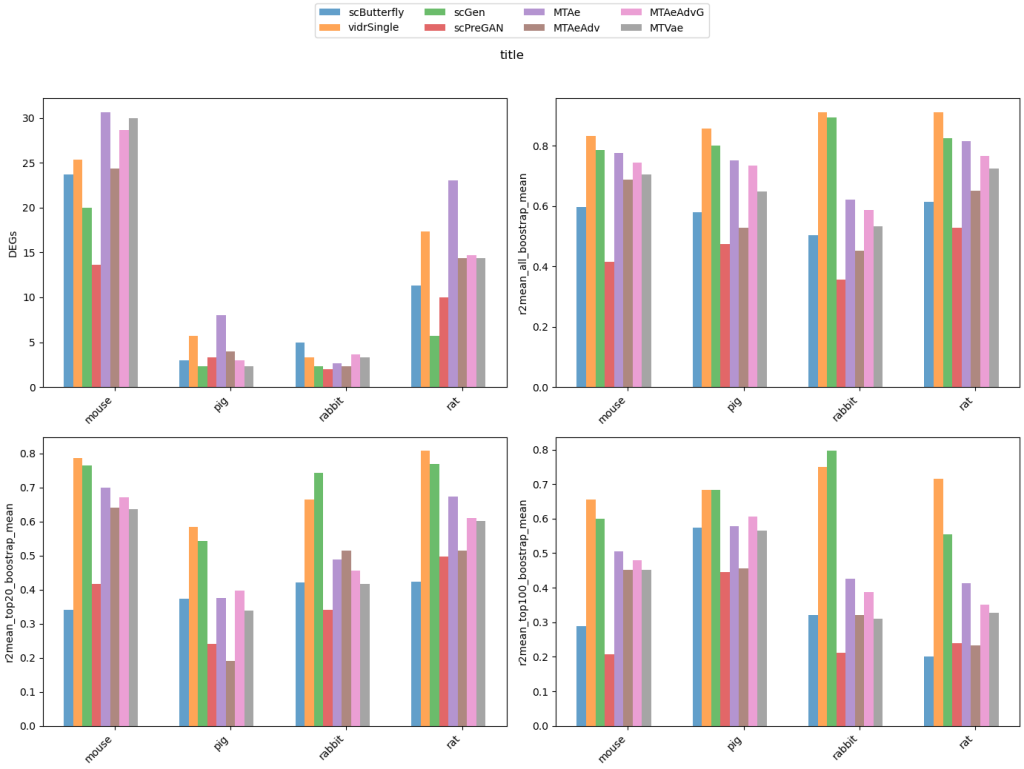


Figure 23: Baseline metrics of multi-task and literature models for the cross-species

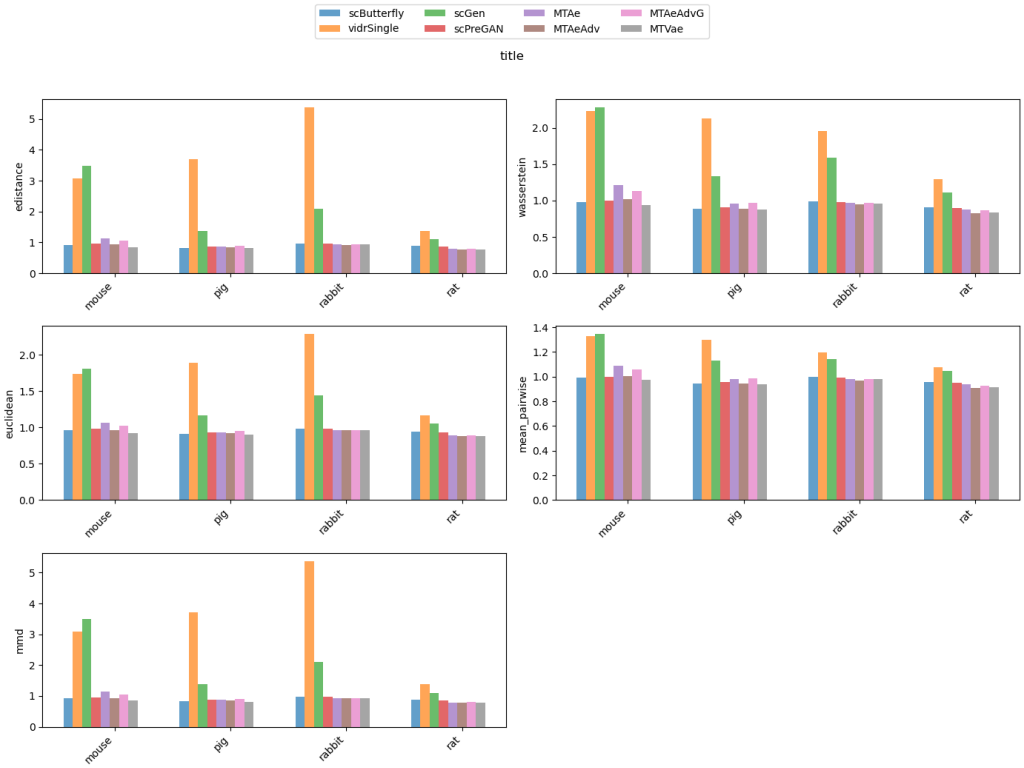


Figure 24: Distance metrics of multi-task and literature models for the cross-species

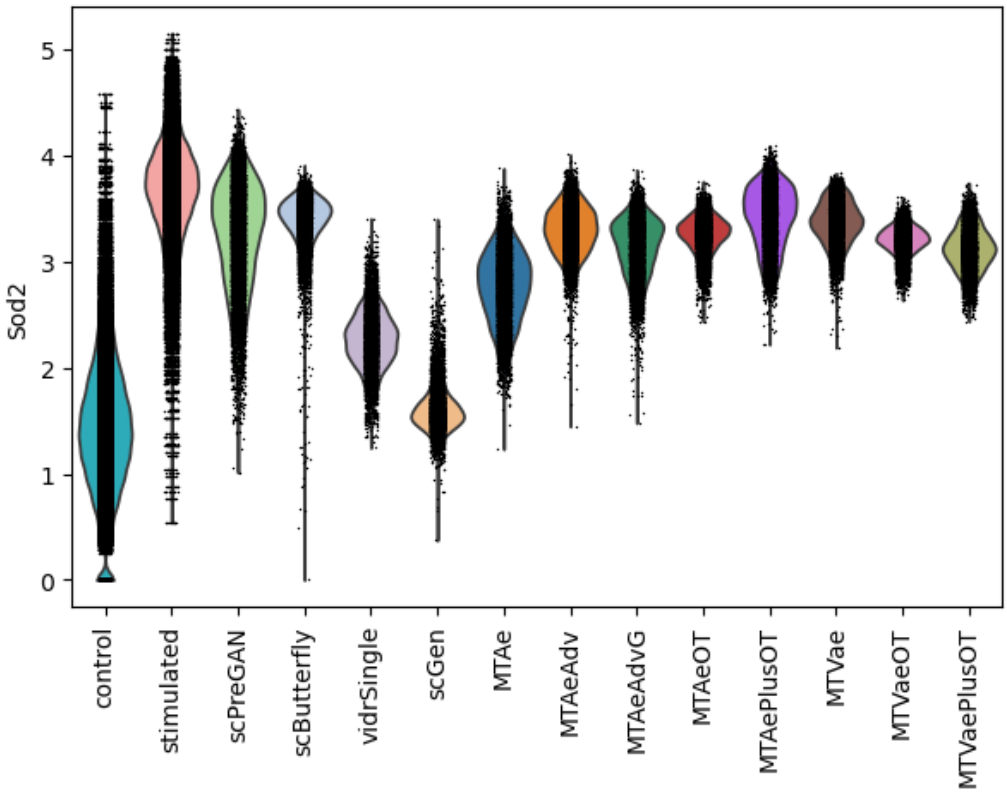

Sod2
0
1
2
3
4
5
control
stimulated
scPreGAN
scButterfly
vidrSingle
scGen
MTAe
MTAeAdv
MTAeAdvG
MTAeOT
MTAePlusOT
MTVae
MTVaeOT
MTVaePlusOT

Figure 25

5.4 Nault et al.

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 7: Score of the models for Nault et al. [10, 11] along with the actual value in parenthesis

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 8: Nault et al. [10, 11]

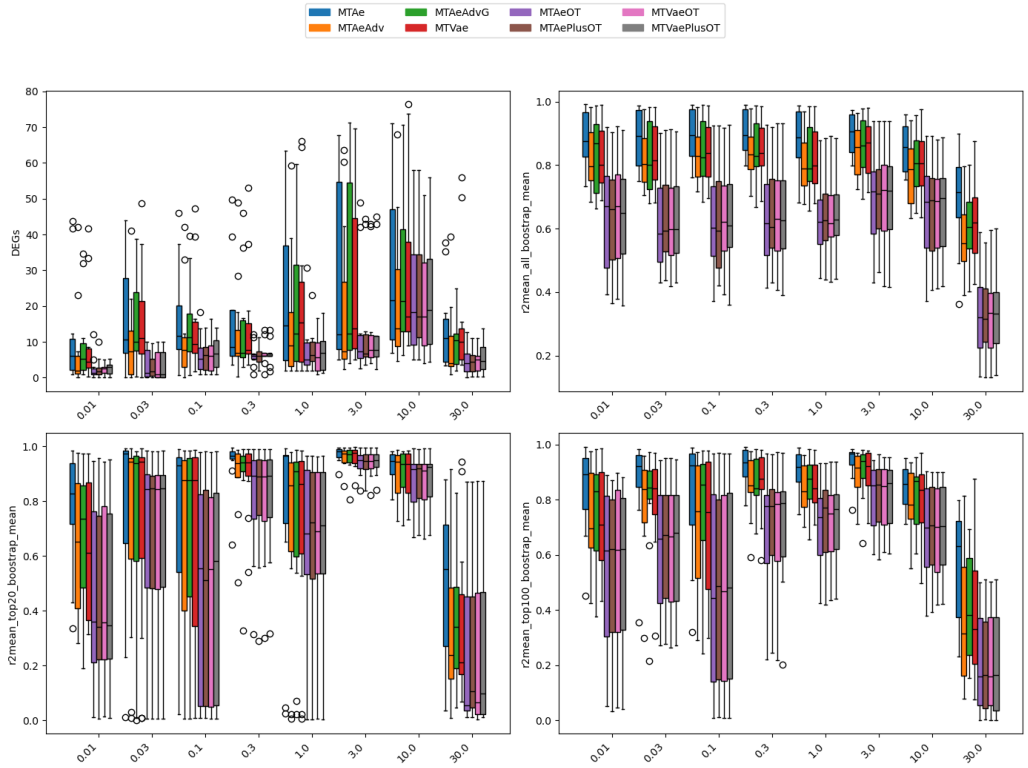


Figure 26: Baseline metrics of multi-task models for the Nault et al. [10, 11] dataset across dosages

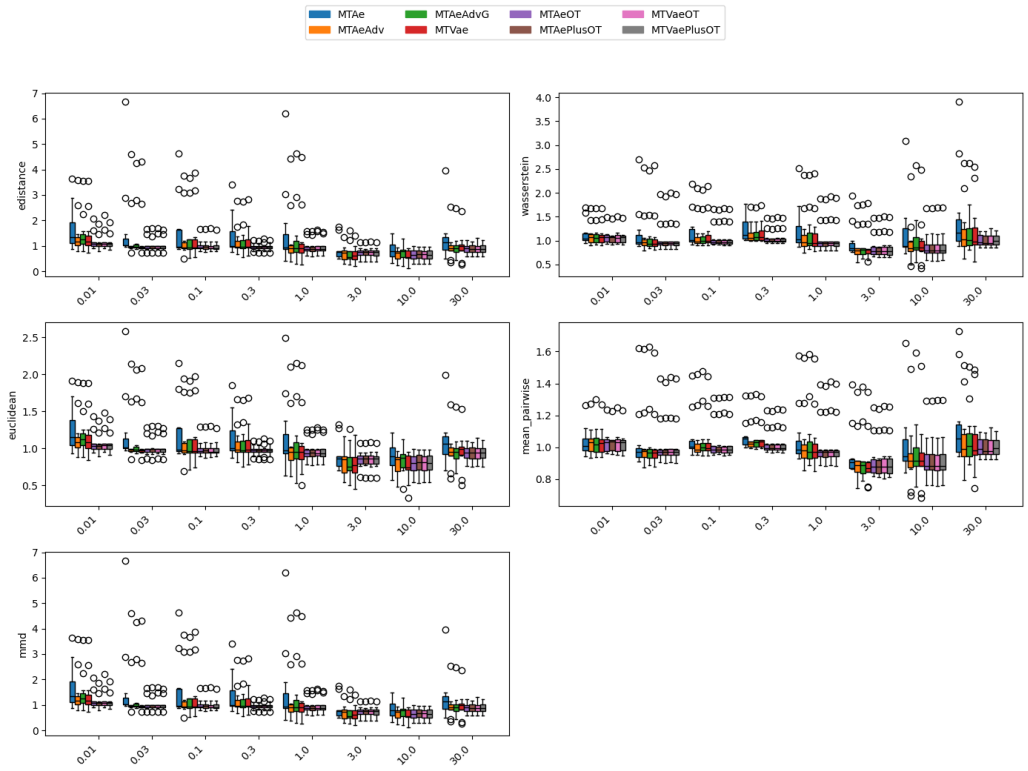


Figure 27: Distance metrics of multi-task models for the Nault et al. [10, 11] dataset across dosages

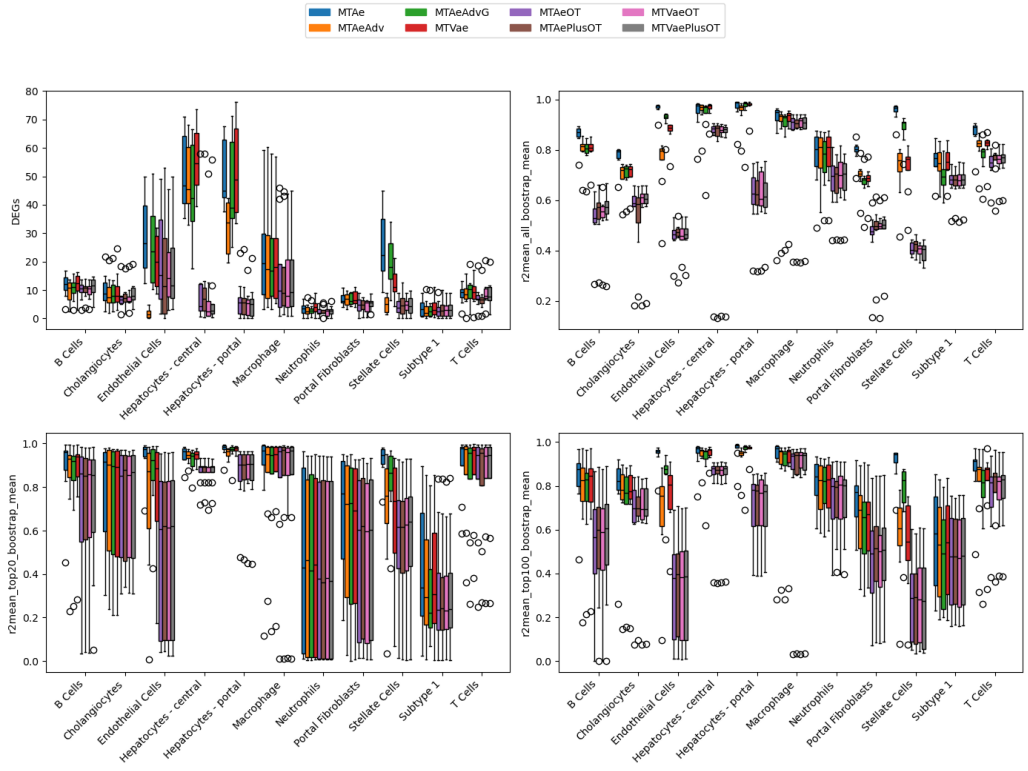


Figure 28: Baseline metrics of multi-task models for the Nault et al. [10, 11] dataset across cell types

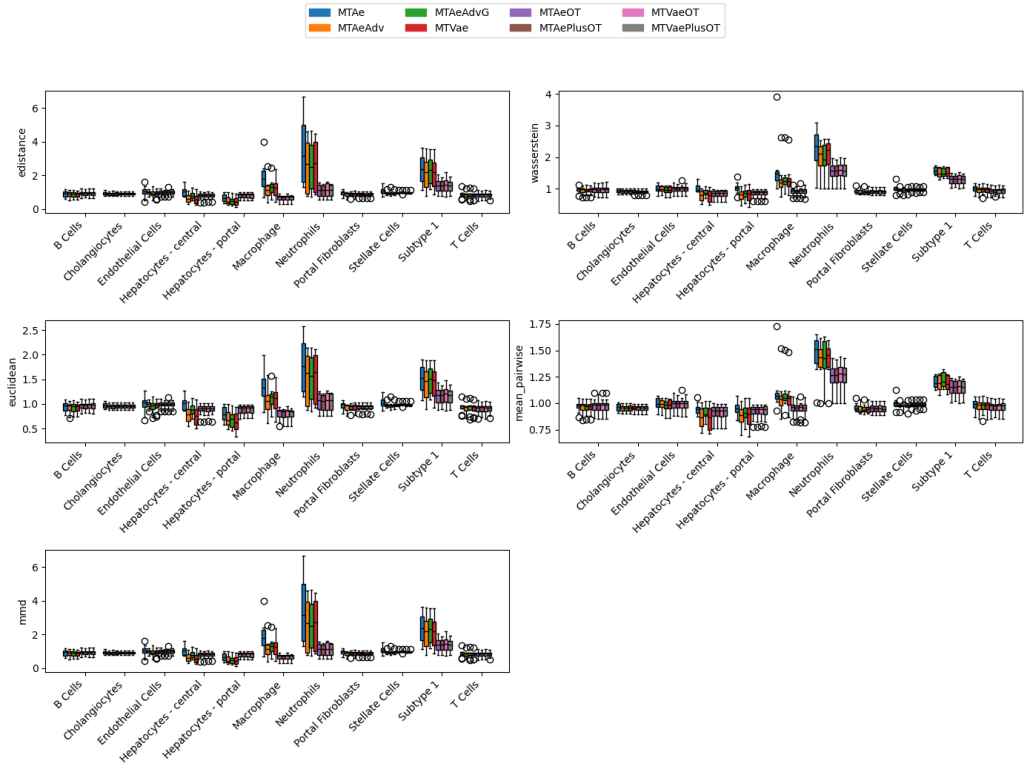


Figure 29: Distance metrics of multi-task models for the Nault et al. [10, 11] dataset across cell types

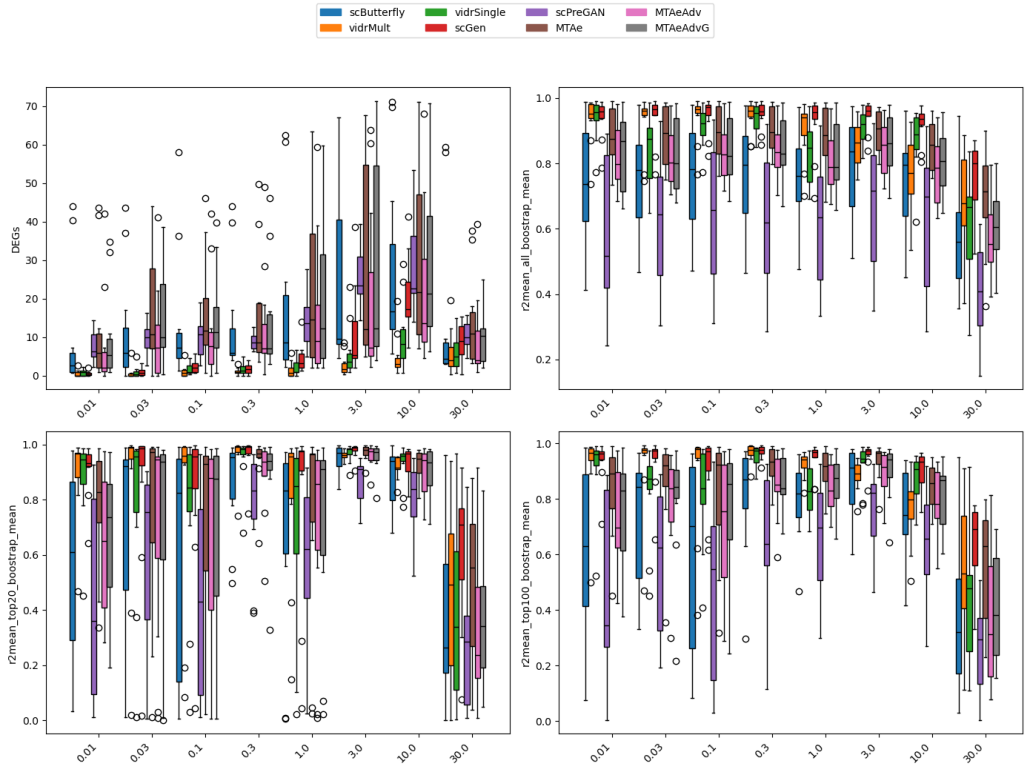


Figure 30: Baseline metrics of multi-task and literature models for the Nault et al. [10, 11] dataset across dosages

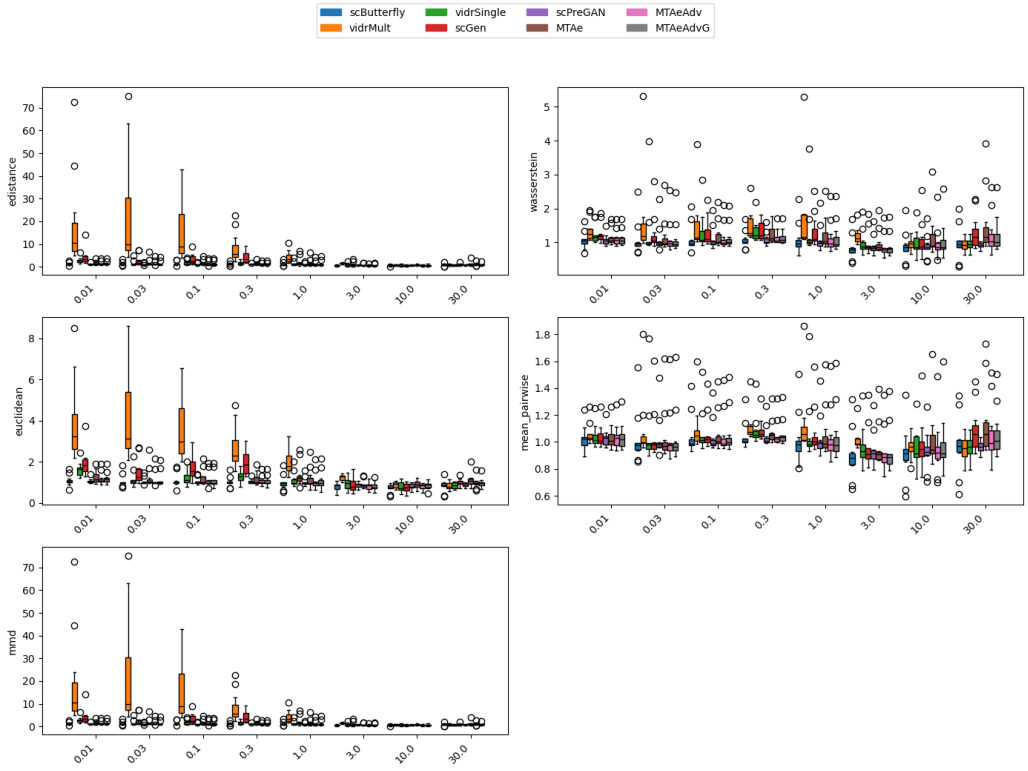


Figure 31: Distance metrics of multi-task and literature models for the Nault et al. [10, 11] dataset across dosages

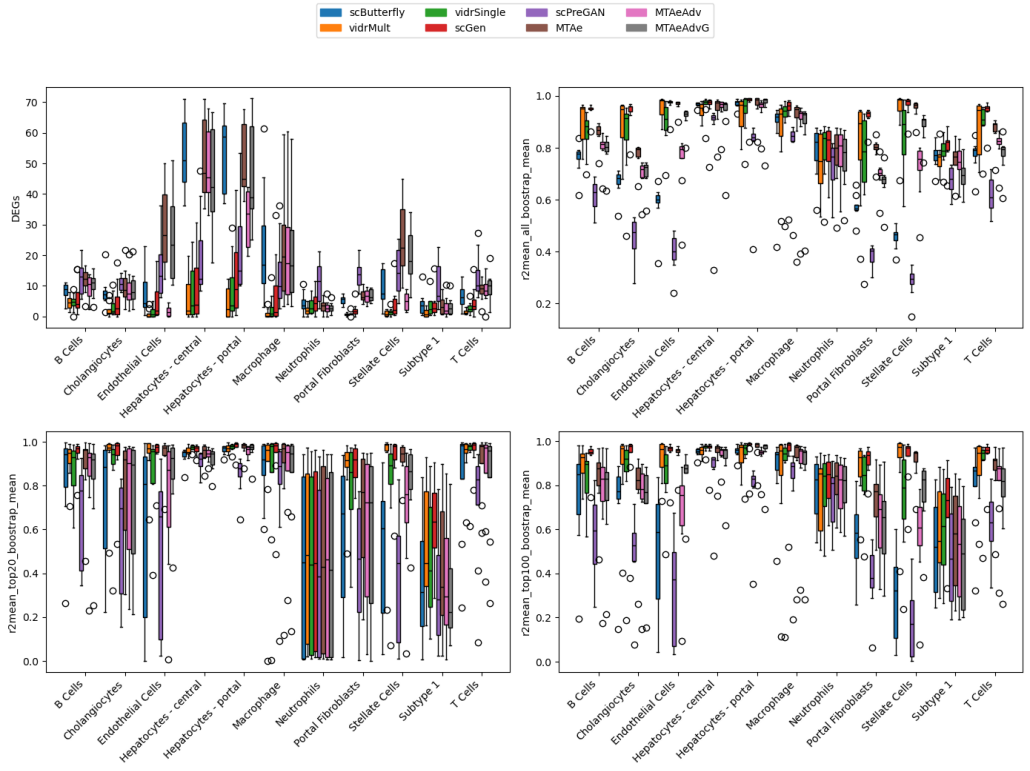


Figure 32: Baseline metrics of multi-task and literature models for the Nault et al. [10, 11] dataset across cell types

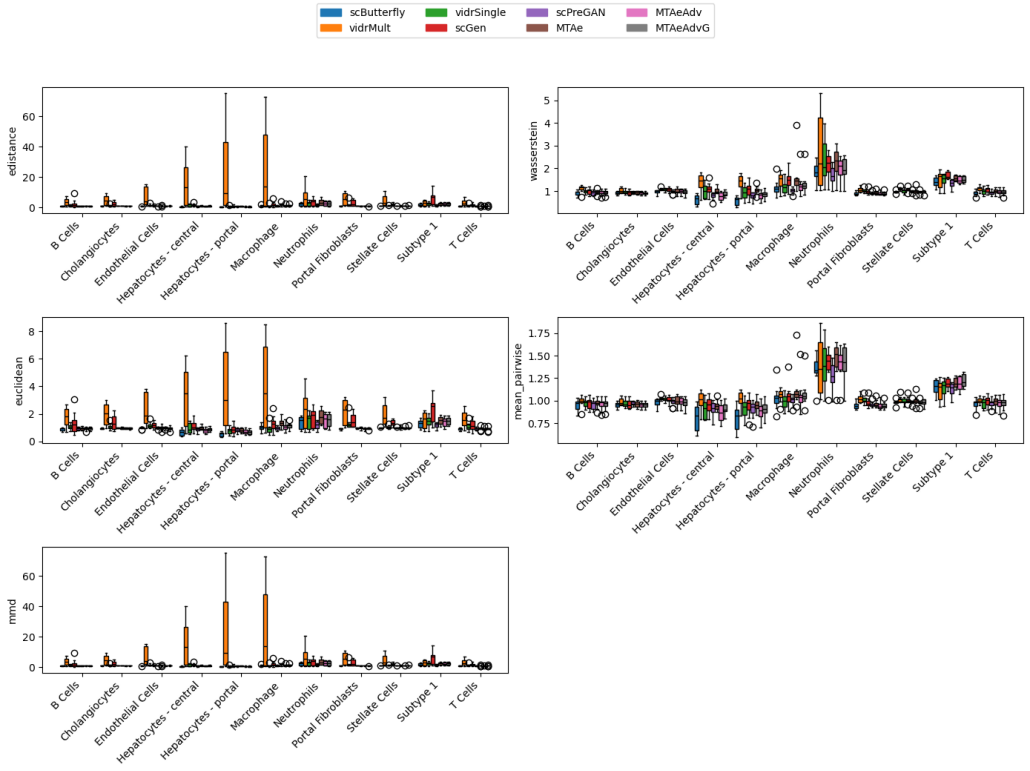


Figure 33: Distance metrics of multi-task and literature models for the Nault et al. [10, 11] dataset across cell types

5.5 Knowledge transfer

which tasks and why they are important?

5.6 TODO

- interpretability
- explainability
- integration with multiple omics

6 Conclusions

We have validated the potential of scButterfly to perturbation modeling with the multi-dosage dataset of Nault et al. [10, 11], an addition of the author's study that is based only on the dataset of Kang et al. [7]. 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.

7 Future work

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] Lukas Heumos, Anna C Schaar, Christopher Lance, Anastasia Litinetskaya, Felix Drost, Luke Zappia, Malte D Lücken, Daniel C Strobl, Juan Henao, Fabiola Curion, et al. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, 24(8):550–572, 2023.

[6] 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.

[7] 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.

[8] 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. *Patterns*, 4(8), 2023.

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

[10] 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.

[11] 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.

[12] Valentine Svensson, Eduardo da Veiga Beltrame, and Lior Pachter. A curated database reveals trends in single-cell transcriptomics. *Database*, 2020:baaa073, 2020.

[13] Artur Szałata, Karin Hrovatin, Sören Becker, Alejandro Tejada-Lapuerta, Haotian Cui, Bo Wang, and Fabian J Theis. Transformers in single-cell omics: a review and new perspectives. *Nature methods*, 21(8):1430–1443, 2024.

[14] 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.

[15] 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.

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

A Acronyms

LAN Local Area Network

LLM Large Language Model

DEG Differentially Expressed Gene

HVG Highly Variable Gene

TCDD tetrachlorodibenzo-p-dioxin

OOD Out-of-Distribution