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

July 12, 2025

Contents

List of Figures

List of Tables

Acronyms

GAN generative adversarial network. 16

LLMs large language models. 7

MLT Multi-task learning. 8, 9, 11

OOD out-of-distribution detection. 10, 12, 15, 17

PBMCs peripheral blood mononuclear cells. 17

scRNA-seq single-cell RNA sequencing. 12, 14, 16

VAE variational autoencoder. 15, 16

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 architecture that predicts perturbed single-cell transcriptomic profiles for multiple perturbations achieving state-of-the-art performance while exhibiting greater scalability and efficiency compared to existing methods.

1 Introduction

The advent of single-cell technologies has enabled the study of the biological 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 [10, 15]. 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 [13].

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), which enables fine-grained analysis of cellular responses [13]. The perturbation response itself is nonlinear 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 [8].

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 [8]. More specifically there is a growing trend toward leveraging large language models (LLMs) in the field. A recent survey by Szalata et al. [30] 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 outperforming transformers [30], 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 [8].

However, the non-linearity of deep learning models presents another challenge in balancing predictive accuracy with interpretability [15]. This trade-off remains a key milestone in the field, and many recent efforts have aimed to address it through causal machine learning approaches such as the GRouNdGAN, sVAE+, and graphVCI [8]. Other interpretable approaches include the usage of SHAP values by UnitedNet [32], integrative gradients by PerturbNet [36], and the approximation of the function of the uninterpretable non-linear decoder with sparse ridge regression as demonstrated by scVIDR [14].

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. The target is to learn a low-dimensional representation that disentangles the core

biological context while removing technical variations.

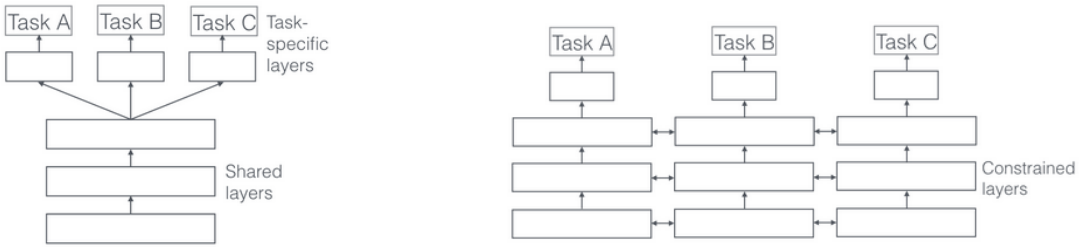
1.1 Multi-task learning

Multi-task learning (MLT) is a machine learning paradigm in which a single model is trained to perform multiple related tasks simultaneously. The central idea is that by sharing representations across tasks, the model can generalize better than if each task were learned in isolation. This approach is inspired by human learning and cognition, where analogy plays a central role in transferring knowledge across domains [11, 37]. From a machine learning perspective, we can view it as a form of inductive bias. It directs the model to prefer the hypothesis that explains more than one tasks, similarly with L1 regularization that leads to a preference for sparse solutions [25]. The degree of benefit depends on the relationship between tasks. If tasks are poorly related, negative transfer may occur, where learning one task harms performance on another [28]. Therefore, understanding task relationships and designing appropriate shared architectures are crucial to the success of MLT.

One of the early-stage motives of MLT is the implicit data augmentation by combining the sources of information from multiple tasks to alleviate the data scarcity problem. This is particularly relevant in single-cell multi-omics protocols [4], where the data is often limited due to the complexity and cost of the experiment protocols. This is also beneficial to single-cell single modality datasets, where the data is limited for a specific number of perturbations.

Other advantages of MLT include the prevention of overfitting while mitigating the data-dependent noise of each task. Noisy data can obscure the underlying patterns, making it difficult for the model to learn meaningful representations. Combining data from multiple tasks provides additional evidence, so the model can distinguish relevant features from irrelevant ones, leading to more robust features [25]. This is particularly important in single-cell perturbation modeling, where the data is often noisy and sparse due to dropout events and other technical limitations.

Regarding the architecture choise of MLT, we need to consider how to process the interplay of the tasks, a concept referred as conditioning [6]. In the context of deep learning, the most prominent approaches are the hard and soft parameter sharing. In hard parameter sharing, the model shares a common set of parameters across all tasks, while keeping a dedicated head for each one (fig. 1a). It is the most common approach and often preferred when tasks are closely related, as it allows for more efficient learning, reducing the risk of overfitting. In soft parameter sharing, each task has its own set of parameters, but they are encouraged to be similar through regularization (fig. 1b). It is more suitable for tasks that are less related, as it allows for more flexibility in task-specific representations, while being less prone to negative transfer [25].

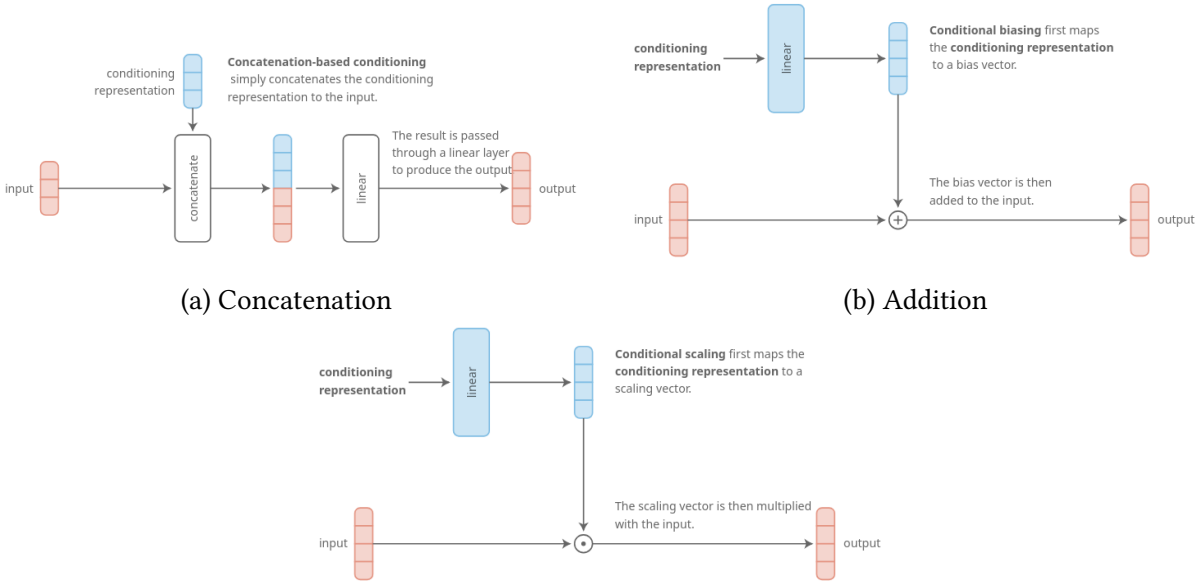


(a) Hard parameter sharing

(b) Soft parameter sharing

Figure 1: [25]

Another approach of conditioning the tasks is the family of feature-wise transformations. In this approach, we can have three different transformations, a) the concatenation, b) the addition, and c) the multiplication. These transformations can be applied in a layer-wise manner, allowing for more flexibility in how tasks are integrated into the model. They can be incorporated either at the initial input of the architecture or at a later stage of the generation process. For the concatenation, given a representation of a task, z, (e.g. one-hot encoded), the input of a layer is concatenated with z, and the output is their linear transformation. For the addition and the multiplication, the conditioning representation is linearly transformed and then added and multiplied to the input respectively. For all these methods, the operations are applied element-wise, hence the name feature-wise transformations [6].



(a) Concatenation

(b) Addition

(c) Multiplication

Figure 2: Methods of integrating conditioning representation [25]

Broadly speaking, MLT methods can be categorized along several dimensions, including the learning paradigm (e.g., supervised or unsupervised), the type of tasks (e.g., classification or regression), and the nature of the input space. In this work, we focus primarily on supervised tasks, which also represent the most widely studied setting in the MLT literature [37]. As we will explore in the next section, our primary task of interest is the prediction of gene expression profiles following a perturbation, which can be formulated as a regression problem. The input consists of a cell's baseline (or "vehicle") gene expression profile, and the output corresponds to the gene expression profile after the perturbation is applied. The perturbation itself is represented by a condition vector, typically implemented as a one-hot encoding that specifies the type of perturbation administered to the cell.

Formally speaking, for supervised tasks with the same input space, we can define m learning tasks $\{T_i\}_{i=1}^{m}$, and $\mathcal{D}_i$ as the accompanying dataset for the ith task, consisting of n_i samples, i.e., $\mathcal{D}_i = \left\{(x_j^i, y_j^i)\right\}_{j=1}^{n_i}$, where x_j^i is the jth input sample and y_j^i is its corresponding label. The loss for the ith task can be formulated as $\mathcal{L}_i(\{\theta_i, \theta_{sp}\}, \mathcal{D}_i)$, where θ_i are the task-specific parameters and θ_{sp} are the shared parameters. The goal of MLT is to learn a set of parameters θ that minimizes the total loss across all tasks, i.e.,

$$\theta = \arg\min_{\theta} \sum_{i=1}^{m} \mathcal{L}_i(\theta_i, \mathcal{D}_i)$$

, where $\theta = \{\theta_{sp}, \theta_1, \ldots, \theta_m\}$ represents the total set of model parameters.

2 Perturbation modeling objectives

In the field of perturbation modeling, there are four main objectives that can be outlined [8, 10, 13].

The prediction of unseen omics signatures and phenotypic changes after a perturbation [1] is applied to either a cell line (bulk omics) or individual cells (single-cell omics) is considered the primary objective, often referred to as OOD. Regarding omics signatures, scGen is a well-known model that has served as a baseline for perturbed single-cell transcriptomics prediction. For phenotypic changes, this could refer to cell viability as a function of drug dosage, typically quantified by IC50 values. DeepDSC [16] is another example, a deep neural network that predicts the drug sensitivity of cancer cell lines based on their gene expression profiles and a compound descriptor. This compound descriptor is represented as Morgan fingerprints[2], capturing the 1D and 2D structure of the compound.

The second objective concerns the prediction of a perturbation's mode of action. This involves identifying the signaling pathways and specific target proteins that are activated or inhibited in response to a given perturbation. Understanding which proteins a drug interacts with, and the downstream cascade of molecular events it initiates, is fundamental for drug discovery and repurposing. DeepDTAGen [27] is a multi-task model designed to address this challenge. It predicts the binding affinity between a drug and a target protein, and also generates novel drug candidates represented as SMILES strings[3], conditioned on a given protein target.

Perturbation interaction prediction, that can be highly beneficial for combinatorial treatments, is treated as the third objective. The goal is to predict how different perturbations interact with each other, which is crucial for understanding drug-drug interactions and potential interlinked side effects. This can involve predicting whether two drugs will have synergistic or antagonistic effects when administered together. For example, the DeepSynergy model predicts a synergy score having as inputs the gene expression profiles of a cell line and the chemical descriptors of two drugs [24]. The synergy score quantifies the deviation of an experimentally observed response surface from one predicted by a theoretical reference model, such as Loewe Additivity [17], Bliss Independence [3], Highest Single Agent (HSA) [31], or the more recent Zero Interaction Potency (ZIP) [35].

The fourth objective involves the prediction of chemical properties. For instance, the task can be formulated to design de novo chemical compounds that can induce a desired perturbed gene expression profile. PerturbNet [36] is a model capable of addressing this challenge. It first

[1] In the case of chemical perturbations, the term perturbation refers to the specific drug applied. However, for the purposes of the out-of-distribution detection (OOD) task, a change in dosage of the same drug is also considered a distinct perturbation. Throughout this study, we treat each unique combination of drug and dosage as a separate perturbation.

[2] Morgan fingerprints, also known as extended-connectivity fingerprints (ECFP), are a type of molecular fingerprint used in cheminformatics to represent the structure of molecules in a computer-readable format. They encode the structural features into a binary vector [19]

[3] SMILES stands for Simplified Molecular Input Line Entry System. It is a text-based representation of molecular structures. For example, ethanol (CH_3CH_2OH) can be represented as the string "CCO" in SMILES format.

compresses the feature spaces of both transcriptomic profiles and chemical structures using autoencoders. These two modalities are then linked via a conditional invertible neural network (cINN). By operating the cINN in reverse, the model enables counterfactual predictions, allowing the exploration of the chemical feature space for perturbations likely to produce a specified gene expression response. In this way, the task serves as a conceptual bridge between biology and chemistry, linking molecular structure to phenotypic effect.

3 Multi-task learning and perturbation modeling

As mentioned by [8, 13], MLT can be considered a powerful ML/DL approach that can be promising to be applied for perturbation modeling. It is worth mentioning that for the NCI-DREAM challenge [5], addressing the drug sensitivity of unseen cell lines, a Bayesian MLT approach was considered to perform the best [26].

In addition to the categorization of perturbation modeling objectives, single-cell analysis tasks, such as gene regulatory network (GRN) inference, cell clustering, and multi-modal integration, can be highly beneficial when incorporated into a MLT framework. These tasks provide complementary biological context that can enhance the performance and interpretability of perturbation modeling. For example, UnitedNet [32] has demonstrated strong performance in cross-modal prediction and cell-type classification by leveraging multi-omics data within an MLT architecture. Similarly, ScPreGAN [33] integrates cell type classification as an auxiliary task to improve the generation of perturbed single-cell transcriptomic profiles, illustrating the value of combining single-cell tasks with perturbation modeling objectives.

Solving multiple tasks together can be challenging when they operate on different levels of granularity. For example, prediction of cell viability or drug sensitivity with IC50 values is considered a population-level task, while prediction of single-cell gene expression after a perturbation is a cell-level task. For the former, available datasets provide information about the gene expression profiles of cell lines, along with the chemical compound and the corresponding IC50 values. However, for single-cell perturbation response prediction, a corresponding population-level phenotype (like IC50) from the same experiment is often not directly available. Solving these tasks simultaneously would require bridging bulk with single-cell omics, taking into account the technical variations between the experimental procedures of data acquisition.

On the other hand, the task of predicting bulk-level perturbation responses could be integrated with other population-level tasks such as cell viability, drug sensitivity, synergy prediction, and target/pathway prediction. Datasets that could provide the required information for this bulk analysis by intersecting cell lines include the LINCS L1000 [29] dataset, which consists of 689,831 microarray measurements from 170 different cell lines treated with 20,065 compounds; the Genomics of Drug Sensitivity in Cancer (GDSC) [12], which catalogues genomic profiles of 639 human cancer cell lines and their drug response data to 130 drugs; and the large-scale oncology screen produced by Merck & Co. [22], which includes 23,062 samples, where each sample consists of two compounds and a cell line.

Instead of treating perturbation objectives as independent tasks, the objectives themselves can be formulated within a MLT framework. For example, by defining the prediction of gene expression for a particular perturbation as a task, MLT is implicitly utilized by models such as scVIDR [14] and CODEX [26]. These models aim to perform this task across multiple perturbations using the same model and by encoding the perturbation as a conditional signal. Another example of division of a specific perturbation objective is STAMP [7], a multi-task model that predicts the differential effect of a perturbation relative to the control gene expression profile.

To achieve this, three tasks have been identified: a) which genes are differentially expressed, b) the magnitude of the differential expression, and c) the direction of the differential expression. The model is trained to predict these three tasks simultaneously, allowing for a more comprehensive understanding of the perturbation effect.

4 Method

Based on the previous analysis of potential integration of perturbation objectives, emphasis has been given to the single-cell use case, as it has been the most explored in the literature due to the recent advancements of single-cell technology. More specifically, we define the prediction of the unseen single-cell gene expression after a perturbation is applied as a task, and we will explore designing a model that can achieve this for a set of perturbations. Similarly to the drug-protein study [2], where the prediction of the binding affinity between a drug and a protein corresponds to a different task for each protein, we will treat each perturbation as a separate task.

A typical dataset for this task consists of transcriptomic profiles obtained from single-cell RNA sequencing (scRNA-seq) across multiple biological contexts, such as different cell types, perturbations, dosages, studies, or species, under both control and perturbed conditions. The objective is to predict the perturbed gene expression profile of a held-out context, given its control-state profile and the applied perturbation . To accomplish this, the model learns the effect of perturbations from the remaining (seen) contexts and generalizes this knowledge to unseen ones. This setting enables the evaluation of a model's ability to extrapolate known perturbation effects to new biological or experimental domains and represents the primary objective of perturbation modeling, the OOD.

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 signal that represents the type of perturbation. The perturbation is represented as a one-hot encoded vector, where, for a dataset with N perturbations, the conditioning vector has length N+1, with an extra entry representing the control condition.

About the conditioning of the task, as we have seen at section 1.1, we have explored the application of 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 . This approach is named as FiLM, for Feature-wise Linear Modulation [6, 23]:

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

, where γ, and β are learnable parameters generated by the so called FiLM generator, that takes as input a condition z (e.g. a vector that indicates the task), and x is the input to be transformed. A FiLM layer is the application of the FiLM transformation to the input of a layer, where the param-

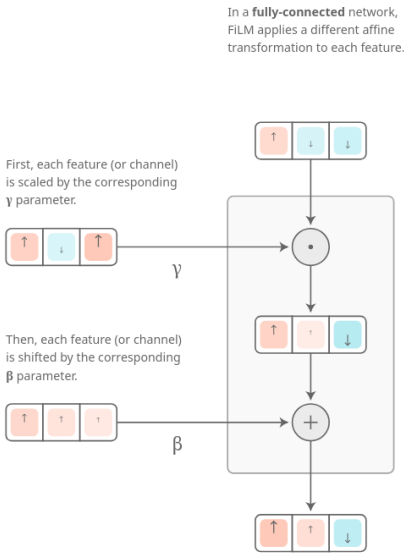


Figure 3: FiLM [6]

eters γ and β can be generated by a common or
layer-specific FiLM generator. Then the result of the transformation is propagated to the rest of the network.

Our baseline architecture is built around an autoencoder, and the conditioning signal is integrated via FiLM layers fused into the decoder (`MTAe`), named as `MTAe`. The modulation parameters γ and β are learned independently for each fusion point with a dedicated FiLM generator. During training, the model learns to reconstruct the perturbed gene expression profiles from the control profiles, while the FiLM layers modulate the decoder's hidden layers based on the perturbation type. 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 and $\hat{x}_i$ are the input and the reconstructed gene expression profile respectively, and N is the number of samples.

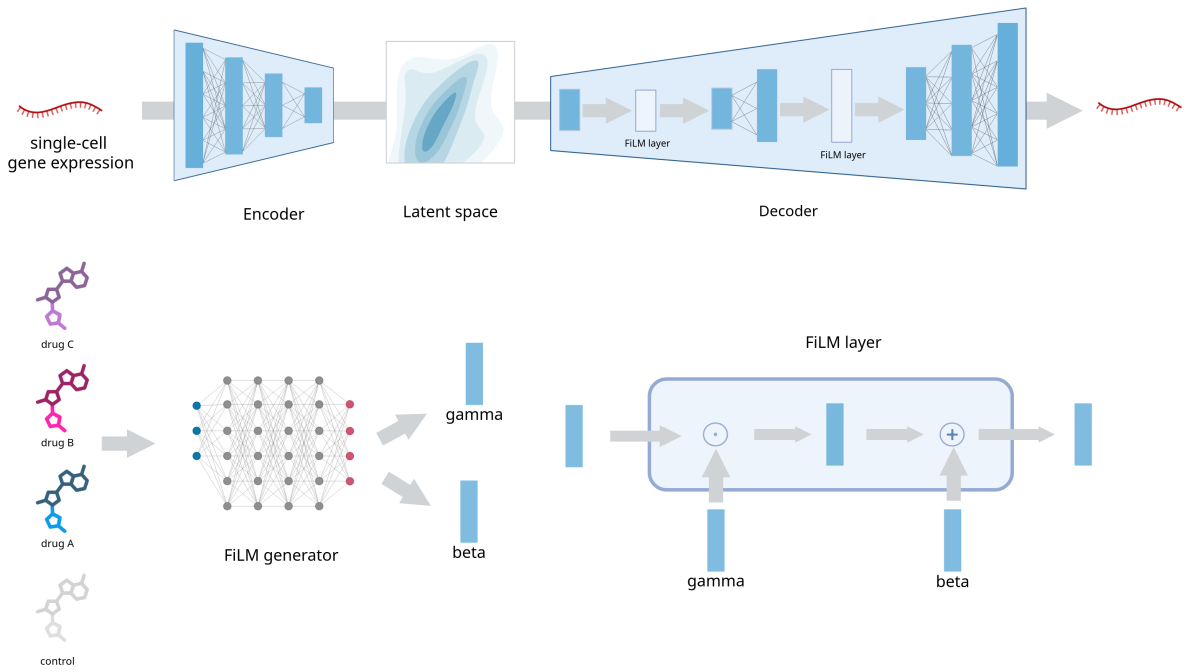


Figure 4: Illustration of the multi-task architecture. The encoder is shared across all tasks, while the decoder is conditioned by the task-specific FiLM layers. In this example, only two film layers are integrated in the first two hidden layers of the decoder. The FiLM generator takes as input a conditioning representation of the perturbations (e.g. one-hot encoded), and outputs the modulation parameters γ, and β for the FiLM layers. Created in `https://BioRender.com`

We have explored several variations of this approach, all of which maintain the decoder architecture with the inclusion of FiLM-based conditioning for all the hidden layers of the

decoder. These variations can be split to three main groups, a) adversarial autoencoders, b) optimal transport, c) Variational Autoencoders (VAEs).

4.1 Adversarial autoencoders (AeAdv)

In our adversarial autoencoder variations, we enforce structure in the latent space via an adversarial loss. The architecture builds on the previously described autoencoder with FiLM layers, extended with a discriminator. The discriminator is trained to distinguish between encoded latent vectors and samples from a target distribution, while the encoder is trained adversarially to fool the discriminator. This encourages the latent space to match the desired target distribution.

In the `MTAeAdv` architecture, we aim to explicitly model a perturbation-free latent space. Here, the discriminator is trained to distinguish between latent representations of control and perturbed gene expression profiles, while the encoder attempts to make them indistinguishable. This encourages the latent space to be agnostic to perturbation.

In contrast, the `MTAeAdvG` architecture enforces a Gaussian prior on the latent space. The discriminator differentiates between latent vectors from the encoder and samples from a fixed multivariate Gaussian, while the encoder learns to match this distribution.

The total loss for the adversarial autoencoder is a combination of the reconstruction loss and the adversarial loss, defined as:

$$\mathcal{L}_{\text{Adv}} = (1 - \lambda)\mathcal{L}_{\text{recon}} + \lambda\mathcal{L}_{\text{adv}}$$

, where λ is a hyperparameter that controls the trade-off between reconstruction and adversarial loss, and $\mathcal{L}_{\text{adv}}$ is the adversarial loss, which can be defined as the binary cross-entropy loss between the discriminator's predictions and the true labels (1 for real samples, 0 for generated samples):

$$\mathcal{L}_{\text{adv}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(D(z_i)) + (1 - y_i)\log(1 - D(z_i))\right]$$

, where $D(z_i)$ is the discriminator's prediction for the ith latent vector z_i, and y_i is the true label (1 for control, 0 for perturbed).

4.2 Optimal transport (OT)

Another set of architectural variations incorporates optimal transport to address the lack of paired samples in scRNA-seq. Because the same cell cannot be sequenced both before and after a perturbation, we lack true one-to-one correspondences between control and perturbed conditions. As a result, modeling must rely on comparing distributions rather than individual cell-level changes.

To mitigate this, we use optimal transport to approximate correspondences between distributions. Specifically, for a given sample from the control distribution, a matching sample from the perturbed distribution is assigned based on OT. This pseudo-pairing allows us to reformulate the training objective: instead of reconstructing the input (as in a standard autoencoder), the model is trained to map control cells to their OT-matched counterparts in 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 paired perturbed gene expression profile, and N is the number of samples.

Compared to the previous architectures, the perturbed gene expression profiles aren't used as inputs to be reconstructed. Instead, they are used as a target distribution to be mapped given the control gene expression profile and the type of the perturbation. 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`.

4.3 Variational Autoencoder (VAE)

The last set of variations involves the inclusion of variational autoencoder (VAE). 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 Current single-cell perturbation modeling methods

Several approaches have been proposed in the literature to address the OOD prediction task. Among these, we selected the following representative methods to benchmark and compare against our multi-task learning architectures: a) scGen [18], b) scVIDR [14], c) scPreGAN [33], and d) scButterfly [4].

5.1 scGen

scGen's architecture is based on a 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.

5.2 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.

5.3 scPreGAN

scPreGAN integrates an autoencoder with a generative adversarial network (GAN) framework to predict 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.

5.4 scButterfly

scButterfly is a generative adversarial model built on a dual-aligned 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) [14].

6 Evaluation

The models described in section 5 serve as baselines for evaluating our multi-task learning architectures on the OOD prediction task. In contrast to our proposed method, these baseline models are typically designed to predict perturbed gene expression for a single, fixed type of perturbation, and lack the ability to condition on varying perturbation types. An exception is scVIDR, which can condition on different dosages of a given drug. This allows a single scVIDR model to predict responses across multiple perturbations (we refer to the dosage-aware version of the model as `vidr-mult`, and the single-perturbation version as `vidr-single`).

Our objective is to explore whether explicitly conditioning on the perturbation type, thus enabling multi-perturbation response prediction, within a multi-task learning framework can lead to improved performance on the OOD prediction task. We extend this evaluation across multiple biological contexts, including cell types, studies, and species, to assess the robustness and generalization capabilities of our approach.

We have used two categories of evaluation metrics to compare the predicted gene expression profiles to the actual perturbed ones: a) baseline metrics, and b) distance metrics. The first ones include the count of differentially expressed genes (DEGs), the R^2 score computed over all the highly variable genes (HVGs), and the R^2 score for the top 100 most variable genes. The distance metrics, designed to capture both point-wise and global differences between the predicted and expected distributions, include: a) euclidean, b) edistance, c) wasserstein, d) mean pairwise, e) mmd. These metrics were computed using the pertpy [9] library. To account for model variability, we repeated each experiment three times using different random seeds (1, 2, and 19193), and report the average performance across runs.

6.1 Single perturbation response prediction

We have evaluated the models on human PBMCs that have been stimulated by IFN-b interferon (Kang et al. [14]). For this dataset, there is only one type of perturbation, and the conditioning signal of our multi-task architectures is a one-hot encoded vector with two entries, one for the control condition and one for the perturbed condition. We have used the provided preprocessed data from scGen study [18], which consists of 18.868 cells, and the most highly variable 6.998 genes.

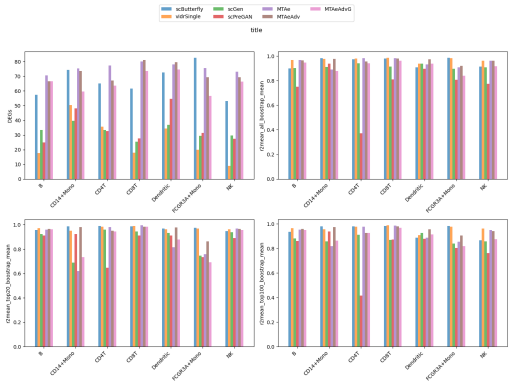
We have tested the models across all the available cell types, training a model from scratch for each use case. To test each one this models, we held-out the perturbed gene expressions of the cell type of interest, and we use the rest to train the model.

By averaging the results across all the cell types, we can see that one of our multi-task versions, `MTAe`, achieved the highest DEGs of ~ 75, while achieving comparable R^2 scores

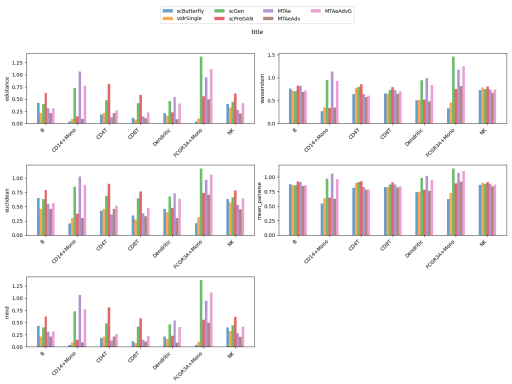
to the scVIDR, and scButterfly, as the best performing ones. The distance metrics, however, show the last ones perform better than our multi-task models, indicating a more representative global structure of the perturbed gene expression profiles. As shown in fig. 5, we have selected ISG15, a marker gene of IFN-b, as an example gene to visualize the distribution of the predicted and expected gene expression profiles across cell types. However, none of the models seem to capture the expected distribution of the gene expression profile.

model	DEGs	R^2_{HVG}	R^2_{HVG20}	R^2_{HVG100}	Euc	Was	E-dist	MPD	MMD
MTAe	**75.714**	0.946	0.871	0.917	0.488	0.892	0.651	0.949	0.488
MTAeAdv	72.381	0.961	0.955	0.948	0.202	0.604	0.429	0.800	0.202
MTAeAdvG	65.905	0.917	0.878	0.901	0.504	0.828	0.681	0.909	0.504
MTAeOT	41.190	0.657	0.668	0.648	0.811	0.947	0.883	0.963	0.811
MTAePlusOT	37.190	0.670	0.674	0.657	0.810	0.951	0.880	0.966	0.810
MTVae	69.095	0.942	0.954	0.928	0.261	0.621	0.499	0.800	0.261
MTVaeOT	39.571	0.669	0.678	0.663	0.813	0.955	0.883	0.966	0.813
MTVaePlusOT	30.619	0.661	0.670	0.655	0.821	0.958	0.888	0.968	0.821
scButterfly	60.727	0.891	0.914	0.889	0.271	**0.601**	0.469	**0.779**	0.271
scGen	32.143	0.910	0.872	0.870	0.627	0.909	0.765	0.946	0.627
scPreGAN	35.750	0.771	0.857	0.799	0.499	0.690	0.682	0.851	0.499
vidrSingle	25.536	**0.970**	**0.971**	**0.961**	**0.182**	0.606	**0.408**	0.797	**0.182**

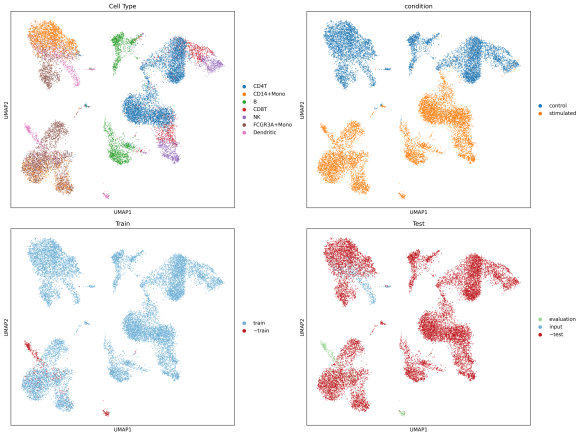
Table 1: Kang et al. [14]



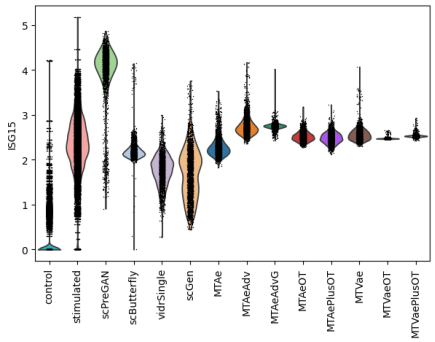
(a) Baseline metrics across cell types



(b) Distance metrics across cell types



(c) UMAP representation of data splitting



(d) Violin plots of gene expression distributions for gene ISG15

Figure 5: Kang et al. [14]

6.2 Multiple perturbations response prediction

As discussed in section 6.1, our multi-task architectures could not leverage shared information across perturbations in the previous setting, as only a single perturbation was available. To more pragmatically assess their performance, we evaluated the models on the dataset introduced by Nault et al. [20, 21], which includes eleven cell types exposed to eight different dosages of TCDD administered to mice.

The dataset was preprocessed using Scanpy [34] by filtering out cells with fewer than 500 total counts and fewer than 720 expressed genes, as well as genes expressed in fewer than 100 cells. The data was log-transformed, for a smoother training process, and the top 5,000 most highly variable genes were selected.

Each dosage level was treated as a distinct perturbation, with the control condition serving as the baseline. The conditioning signal for our multi-task architectures is a one-hot encoded vector of length nine: one entry for each of the eight dosages, and one for the control condition. For evaluation, we trained a separate model for each cell type, holding out the perturbed gene expression profiles of that cell type for testing. The same model was used to predict responses across all dosages, leveraging the model's ability to condition on perturbation identity. In contrast, the baseline models, which do not support conditioning on multiple perturbations, required training a separate model for each dosage. An exception is scVIDR, which is designed to handle multiple dosage levels within a single model.

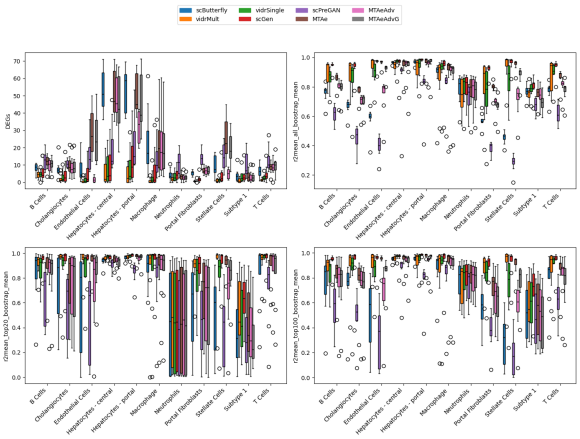
model	DEGs	R^2_{HVG}	R^2_{HVG20}	R^2_{HVG100}	Euc	Was	E-dist	MPD	MMD
MTAe	**20.341**	0.862	0.792	0.833	1.386	1.217	1.116	1.050	1.386
MTAeAdv	13.716	0.792	0.725	0.743	1.128	1.091	1.011	1.017	1.128
MTAeAdvG	18.307	0.808	0.736	0.764	1.164	1.107	1.030	1.029	1.164
MTAeOT	8.652	0.608	0.642	0.590	0.925	1.006	0.951	0.998	0.925
MTAePlusOT	8.519	0.613	0.644	0.596	**0.917**	**1.004**	0.948	0.996	0.917
MTVae	18.981	0.808	0.724	0.753	1.124	1.100	1.005	1.016	1.124
MTVaeOT	8.163	0.614	0.642	0.593	0.929	1.009	0.952	0.998	0.929
MTVaePlusOT	8.701	0.615	0.645	0.597	0.919	1.006	0.948	0.997	**0.919**
scButterfly	16.818	0.740	0.694	0.696	0.984	1.014	**0.944**	**0.990**	0.984
scGen	6.288	**0.915**	**0.863**	**0.897**	2.408	1.229	1.387	1.041	2.408
scPreGAN	14.511	0.599	0.596	0.562	0.972	1.019	0.969	1.000	0.972
vidrMult	2.352	0.870	0.837	0.852	9.295	1.358	2.425	1.054	9.295
vidrSingle	3.795	0.855	0.797	0.824	1.431	1.174	1.118	1.025	1.431

Table 2: Nault et al. [20, 21]

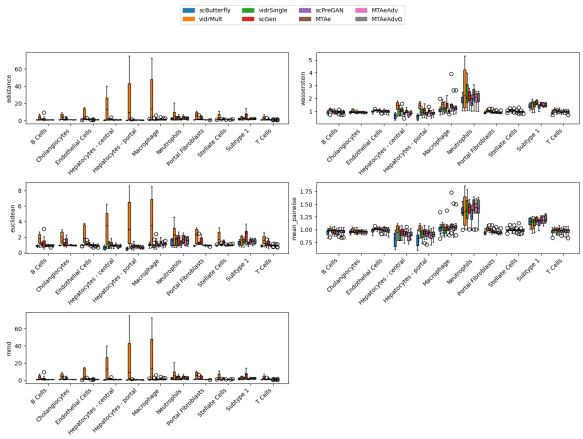
We have averaged the results across all the cell types, and dosages, and similarly with the case of the section 6.1, our multi-task model, `MTAe`, achieved the highest DEGs of $\sim$20. scGen achieved the highest R^2 scores, but with a very low number of DEGs, and scButterfly's performance remained competitive across all metrics. Our optimal transport variations, performed well on the distance metrics, but poorly on the baseline ones.

6.3 Cross-study

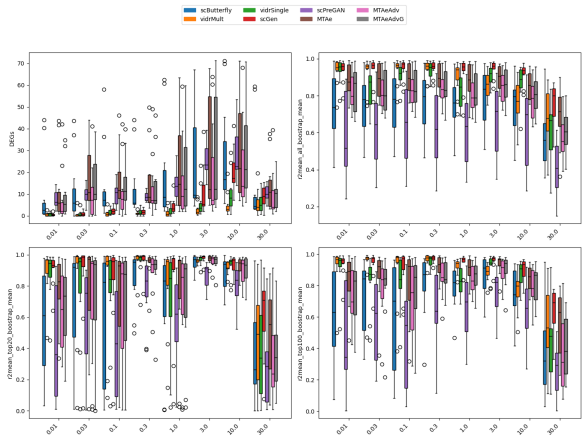
Similarly to the scGen study, robustness against batch effects is a critical aspect of generalization. To evaluate this, we assessed model performance across multiple studies, each potentially introducing technical variation due to differences in experimental protocols, platforms, or sample processing. This cross-study evaluation serves to test the ability of the models to generalize perturbation response predictions beyond dataset-specific biases. By holding out



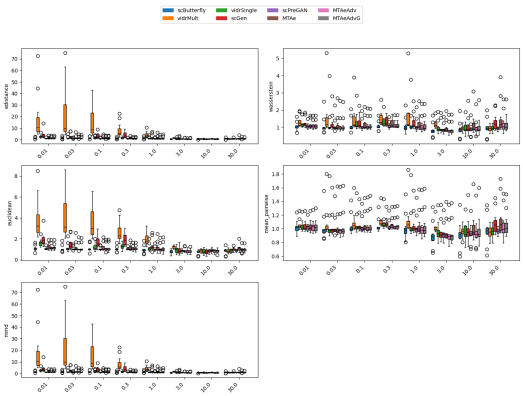
(a) Baseline metrics across cell types



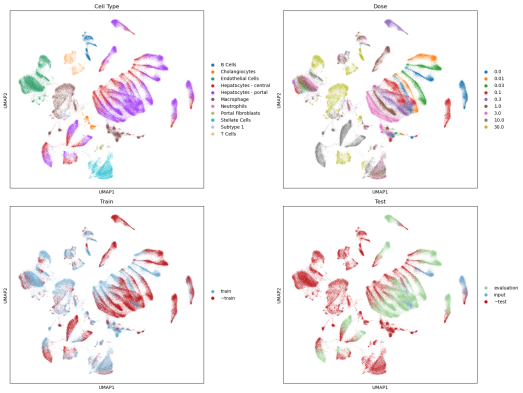
(b) Distance metrics across cell types



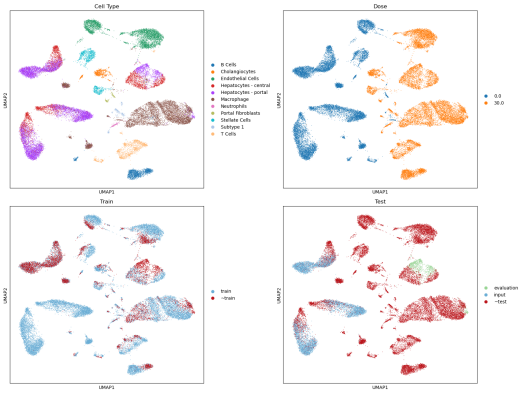
(c) Baseline metrics across dosages



(d) Distance metrics across dosages



(e) UMAP representation of data splitting for all the dosages



(f) UMAP representation of data splitting for the dosage $30\mu g/kg$

Figure 6: Nault et al. [20, 21]

the perturbed profiles of a given study during training and evaluating the model on that study, we simulate an OOD setting with respect to study-level batch effects, thereby assessing the robustness and transferability of each approach across independently generated datasets.

6.4 Cross-species

6.5 Overview

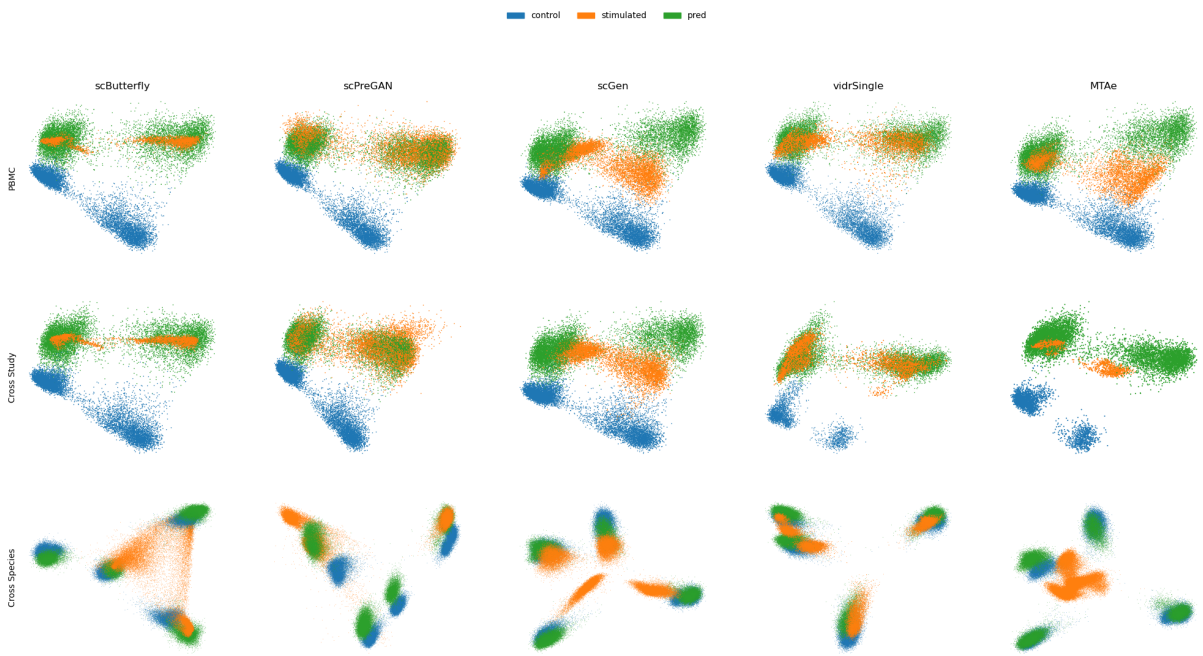


Figure 7: PCA dimensionality reduction of the real unperturbed data, the real perturbed data and the predicted perturbed data.

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. [20, 21] dataset.

7 Results

7.1 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 3: Score of the models for Kang et al. [14] 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 4: Kang et al. [14]

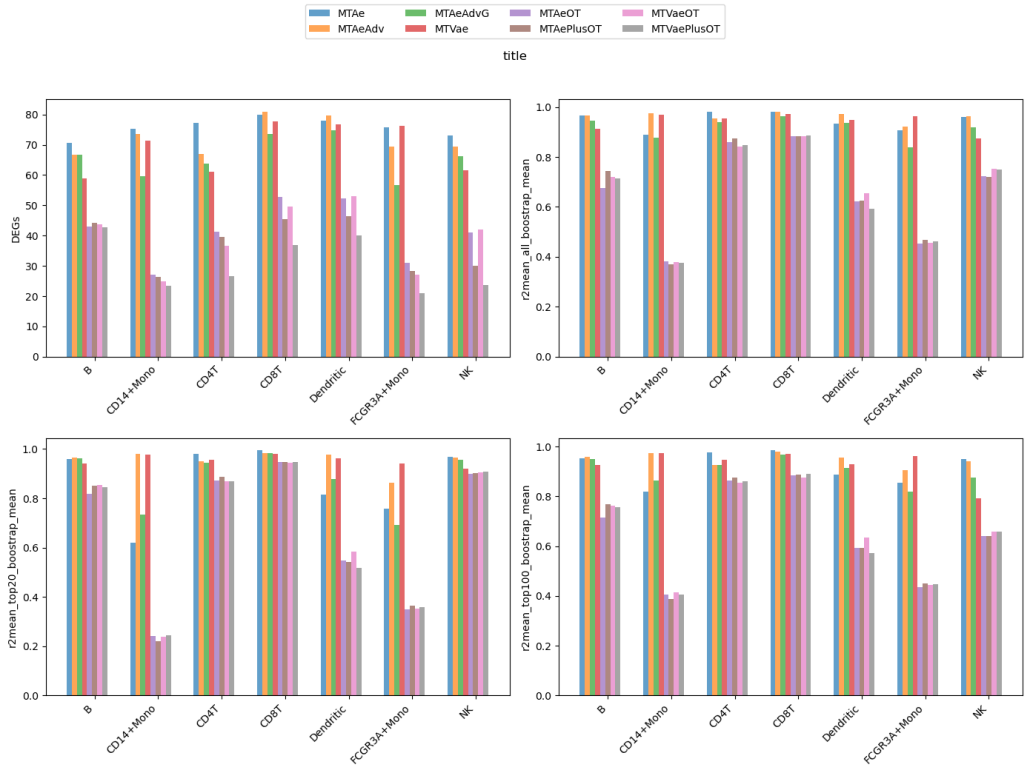


Figure 8: Baseline metrics of multi-task models for the Kang et al. [14] dataset across cell types

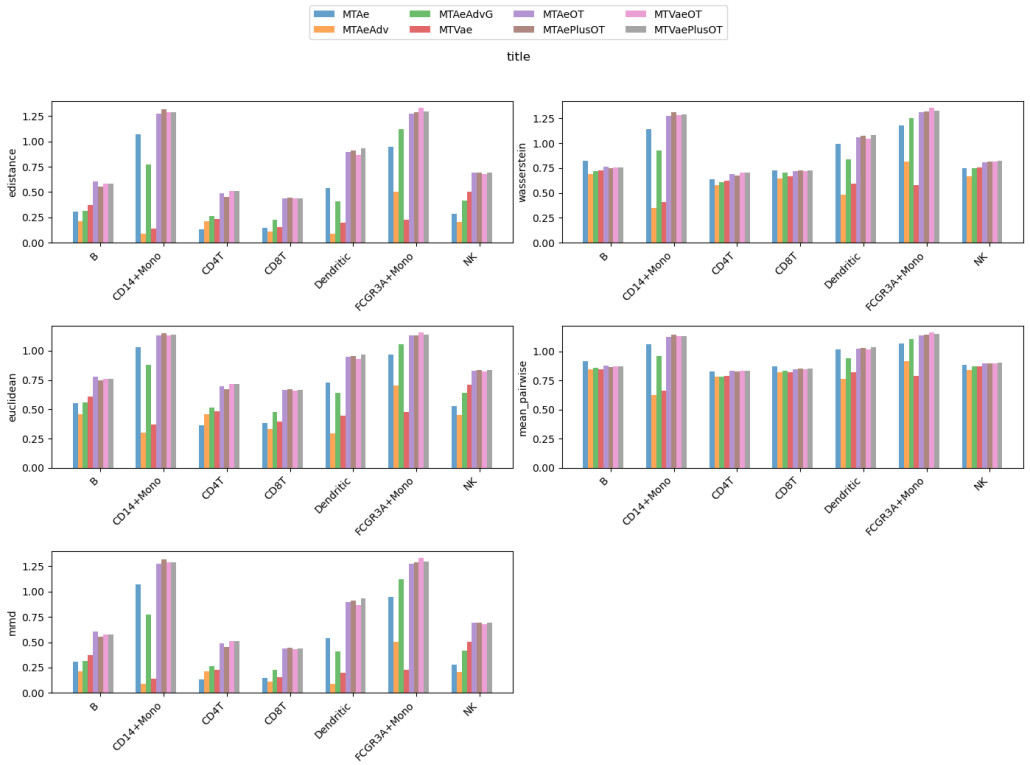


Figure 9: Distance metrics of multi-task models for the Kang et al. [14] dataset across cell types

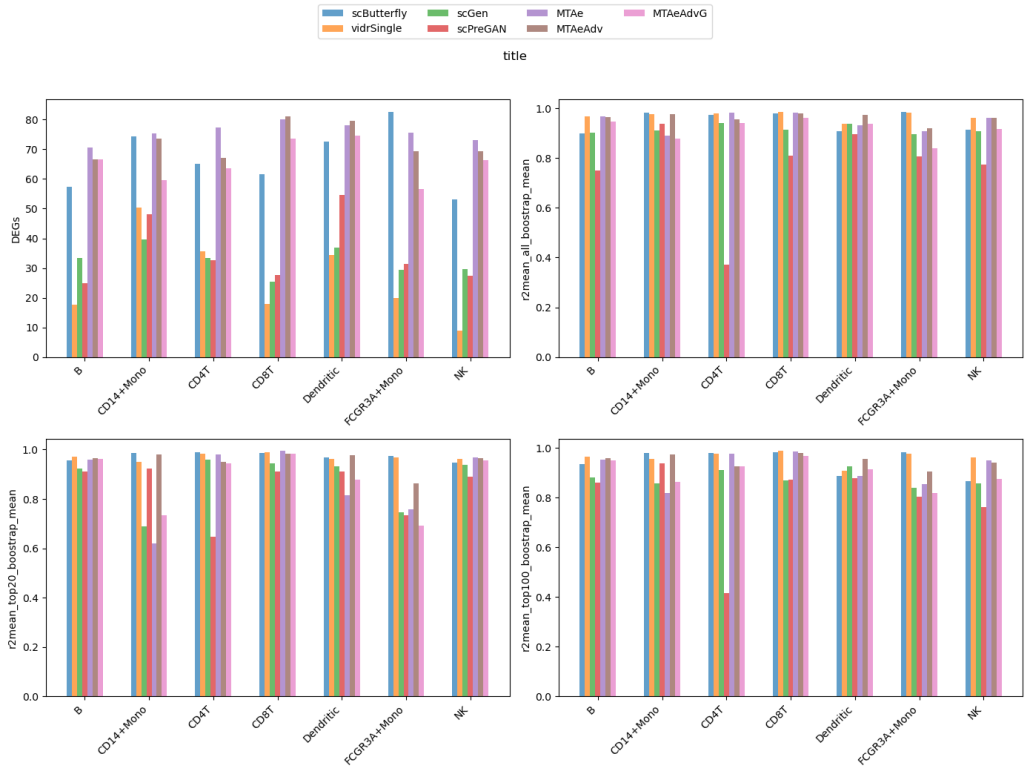


Figure 10: Baseline metrics of multi-task and literature models for the Kang et al. [14] dataset across cell types

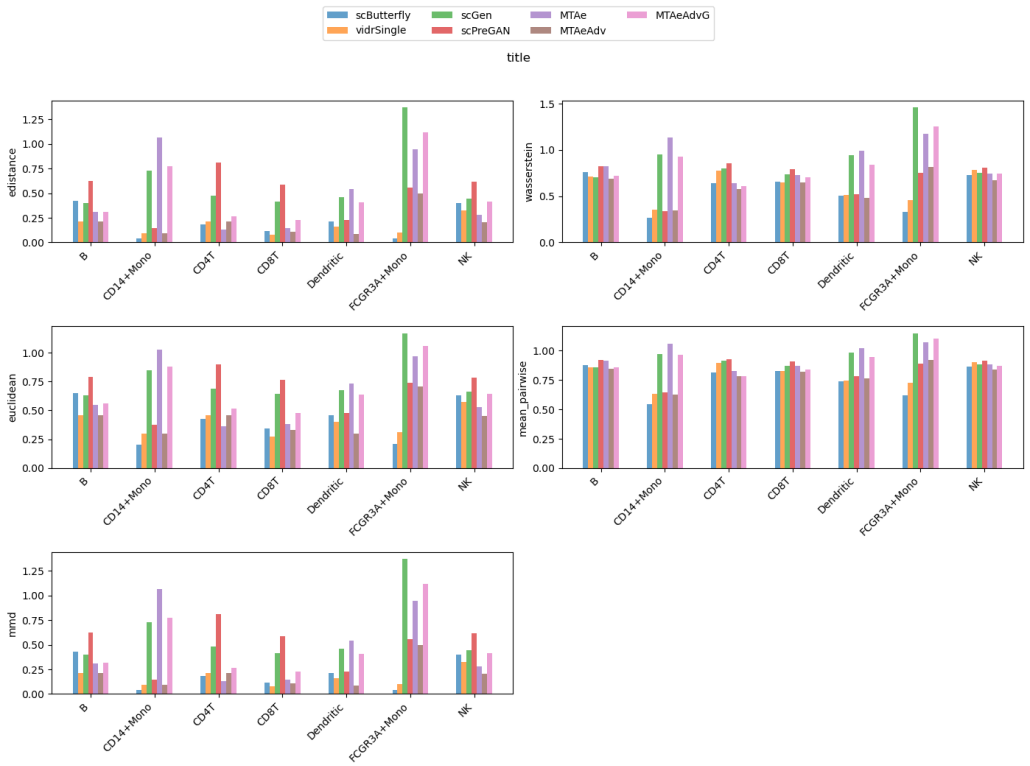


Figure 11: Distance metrics of multi-task and literature models for the Kang et al. [14] dataset across cell types

Figure 12

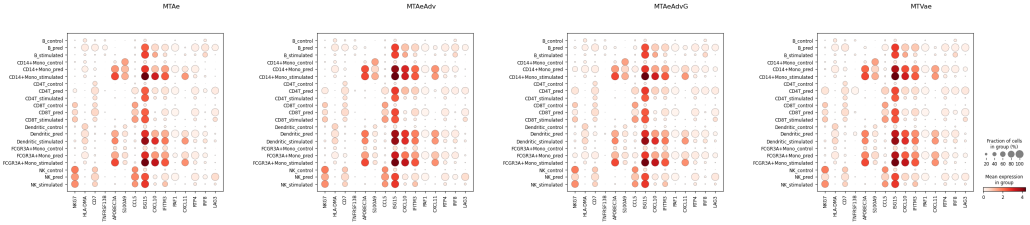


Figure 13

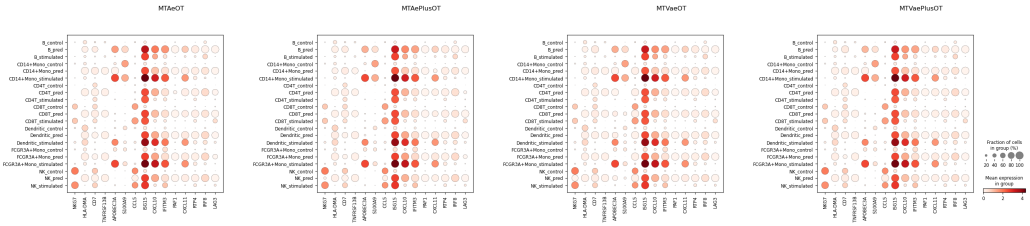


Figure 14

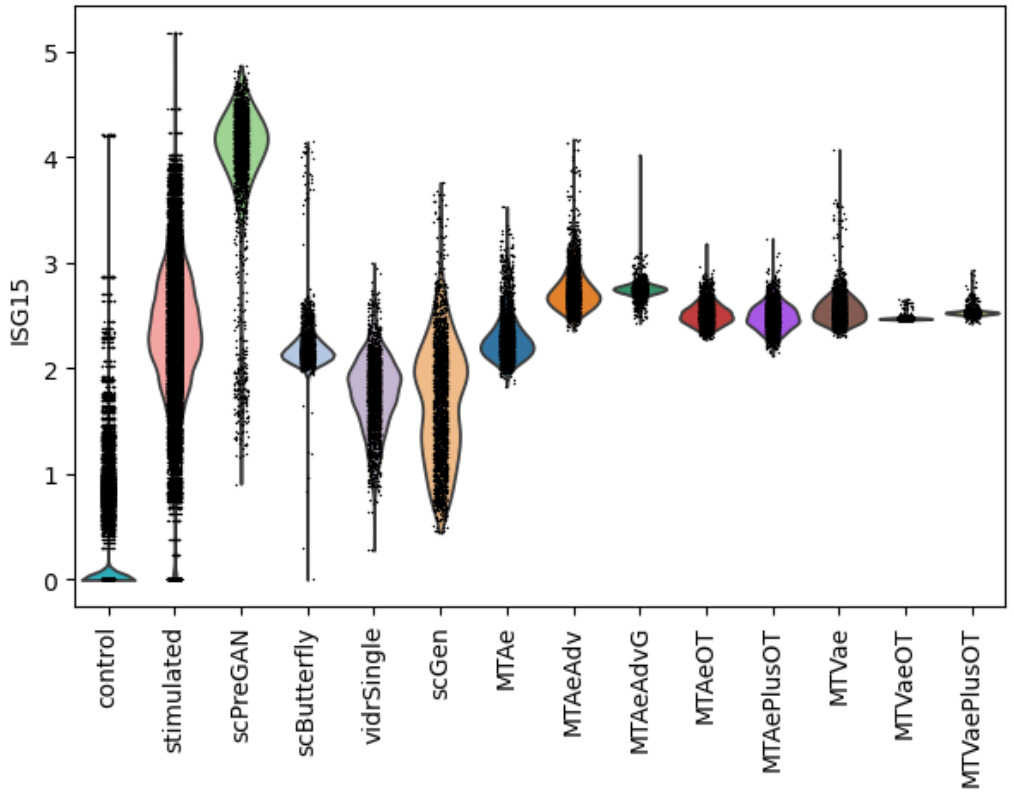


Figure 15

7.2 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 5: 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 6: Score Cross-Study

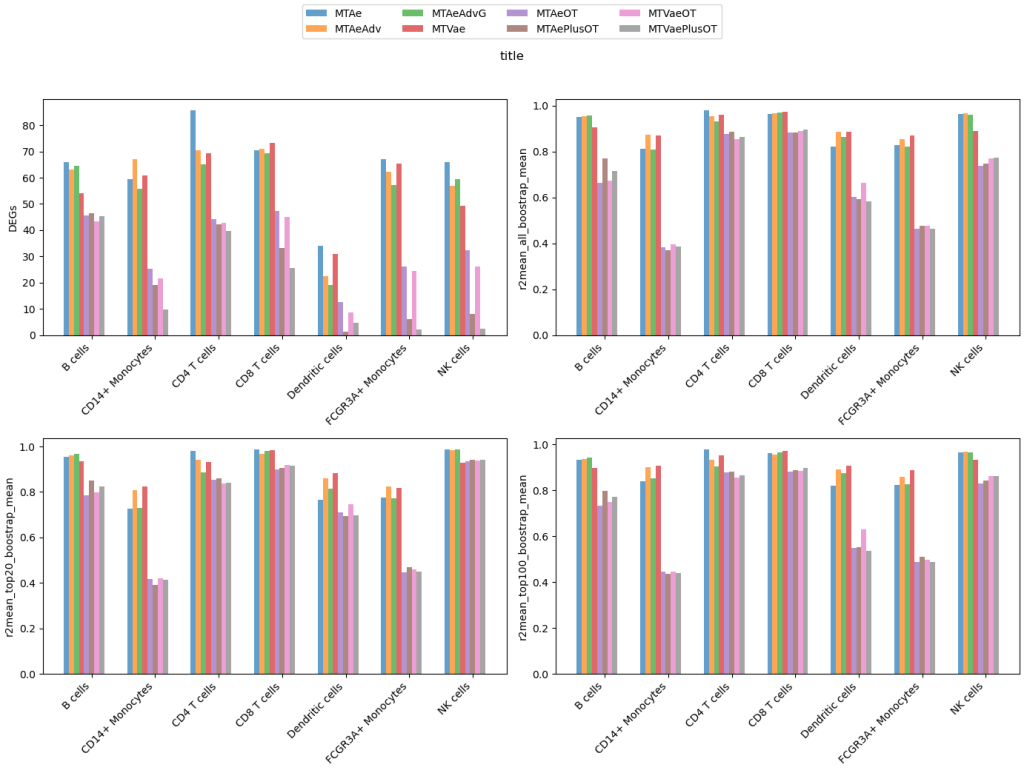


Figure 16: Baseline metrics of multi-task models for the cross-study

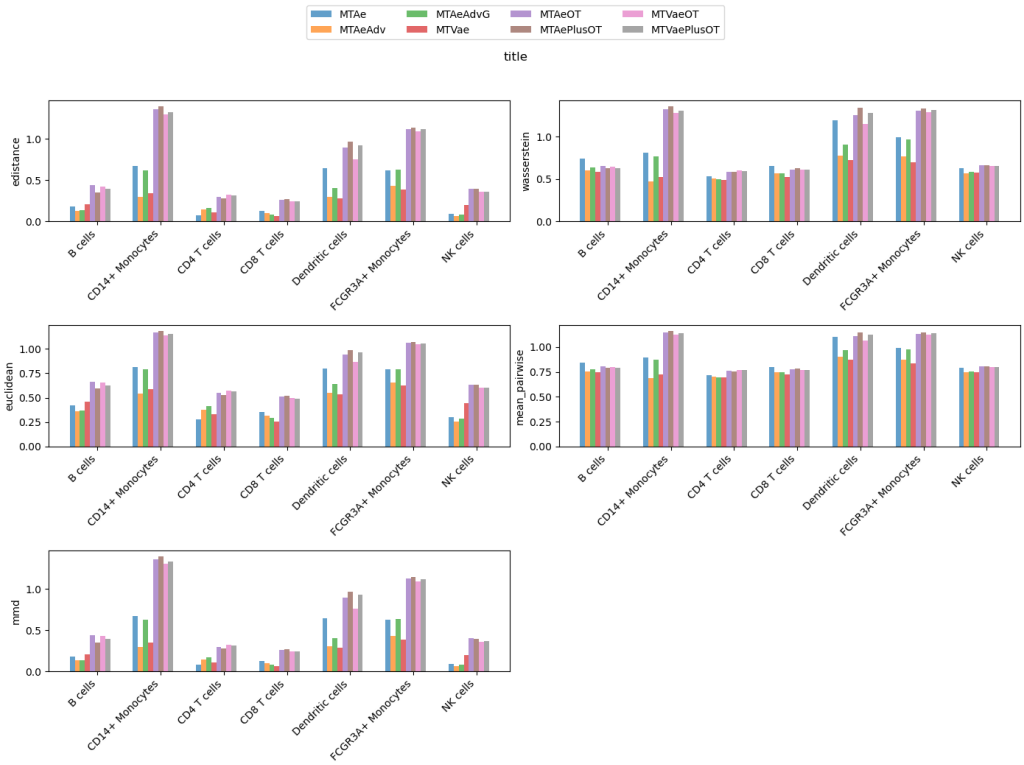


Figure 17: Distance metrics of multi-task models for the cross-study

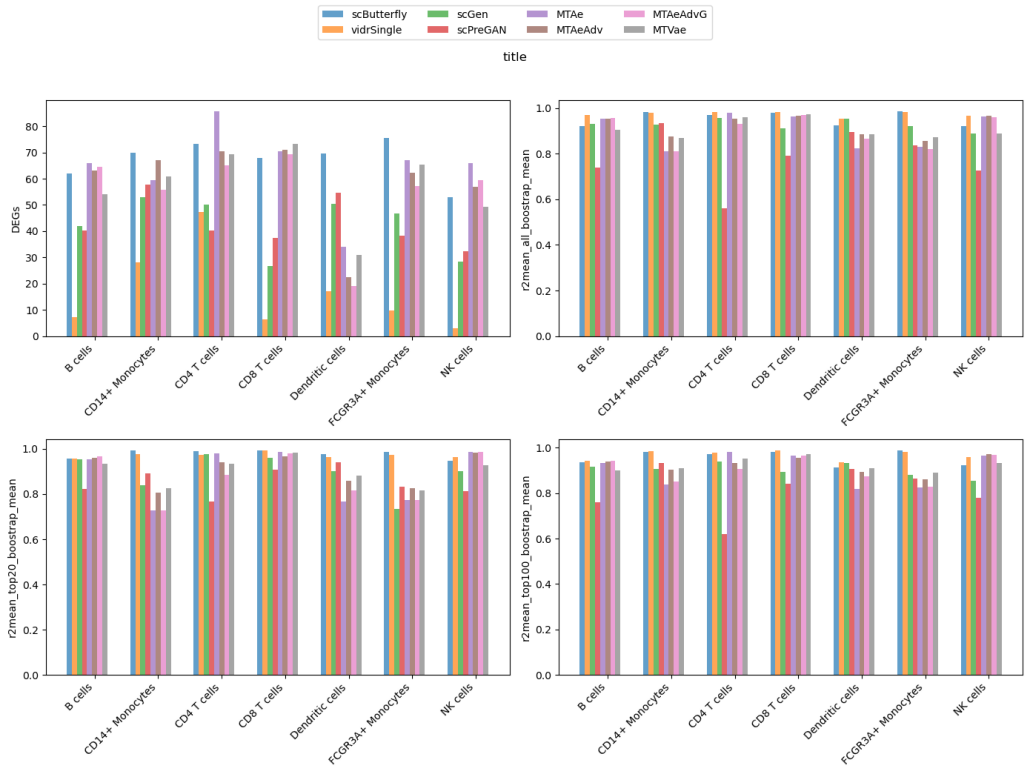


Figure 18: Baseline metrics of multi-task and literature models for the cross-study

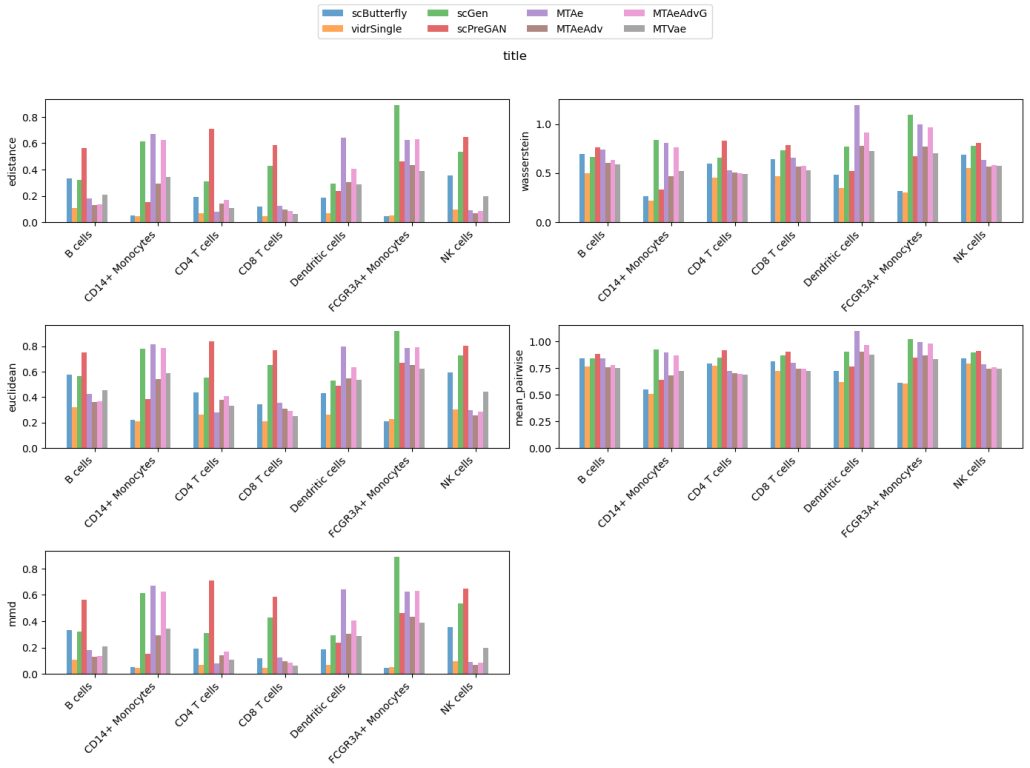


Figure 19: Distance metrics of multi-task and literature models for the cross-study

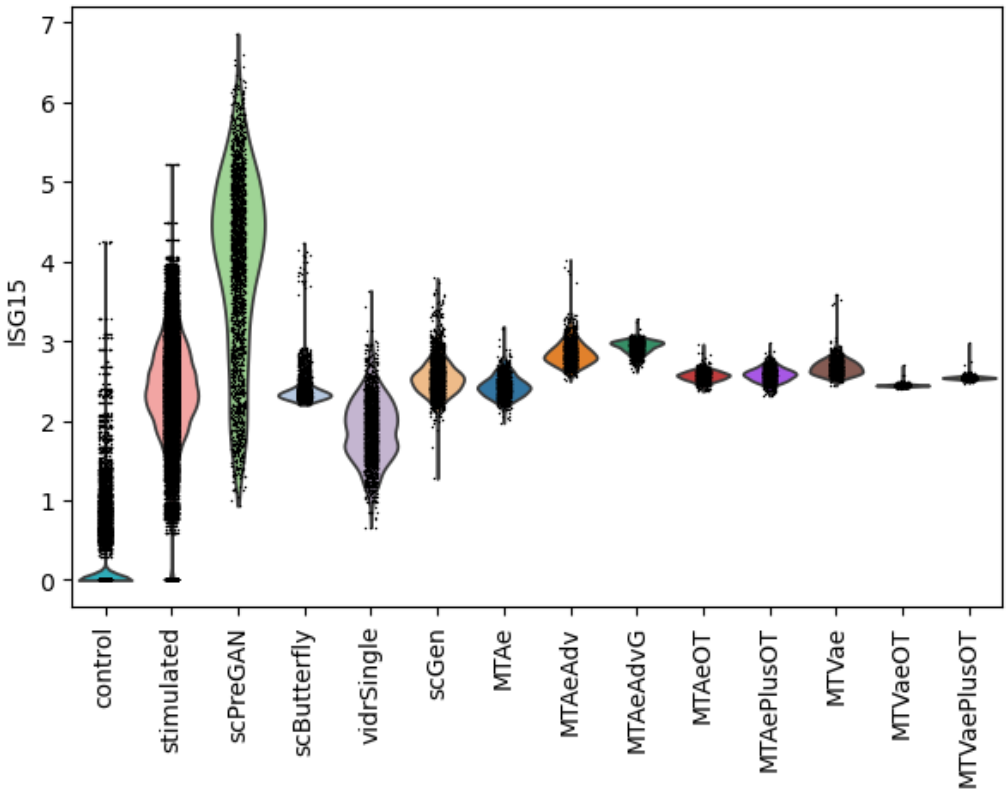

ISG15
0
1
2
3
4
5
6
7
control
stimulated
scPreGAN
scButterfly
vidrSingle
scGen
MTAe
MTAeAdv
MTAeAdvG
MTAeOT
MTAePlusOT
MTVae
MTVaeOT
MTVaePlusOT

Figure 20

7.3 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 7: 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 8: Score Cross-Species

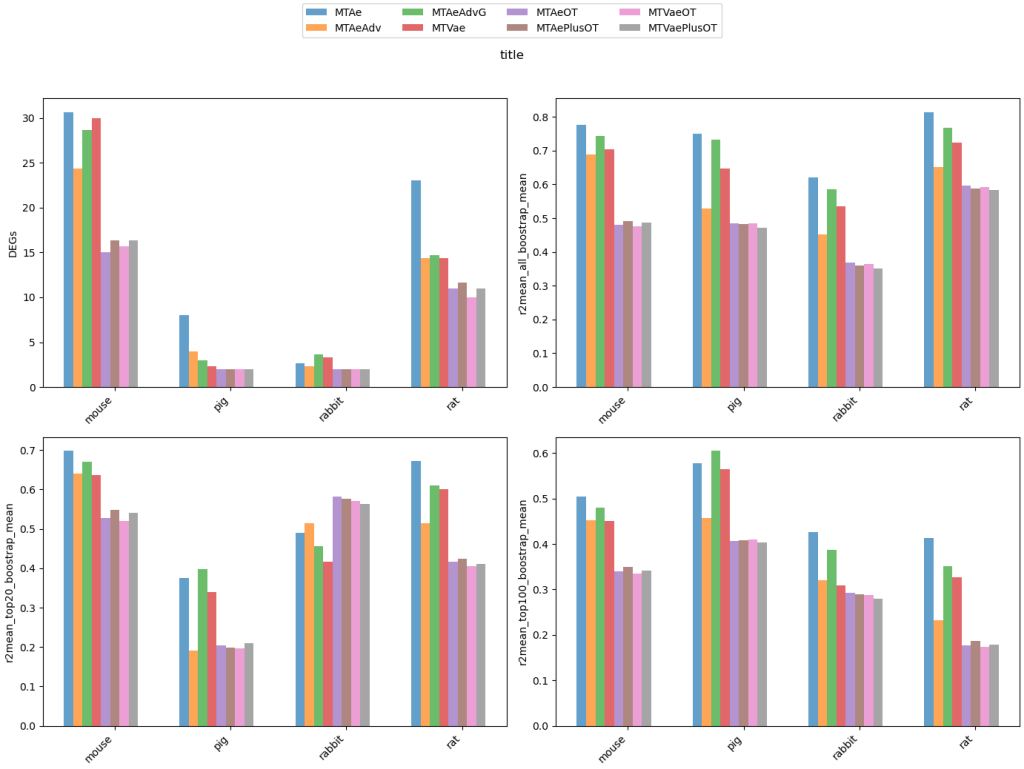


Figure 21: Baseline metrics of multi-task models for the cross-species

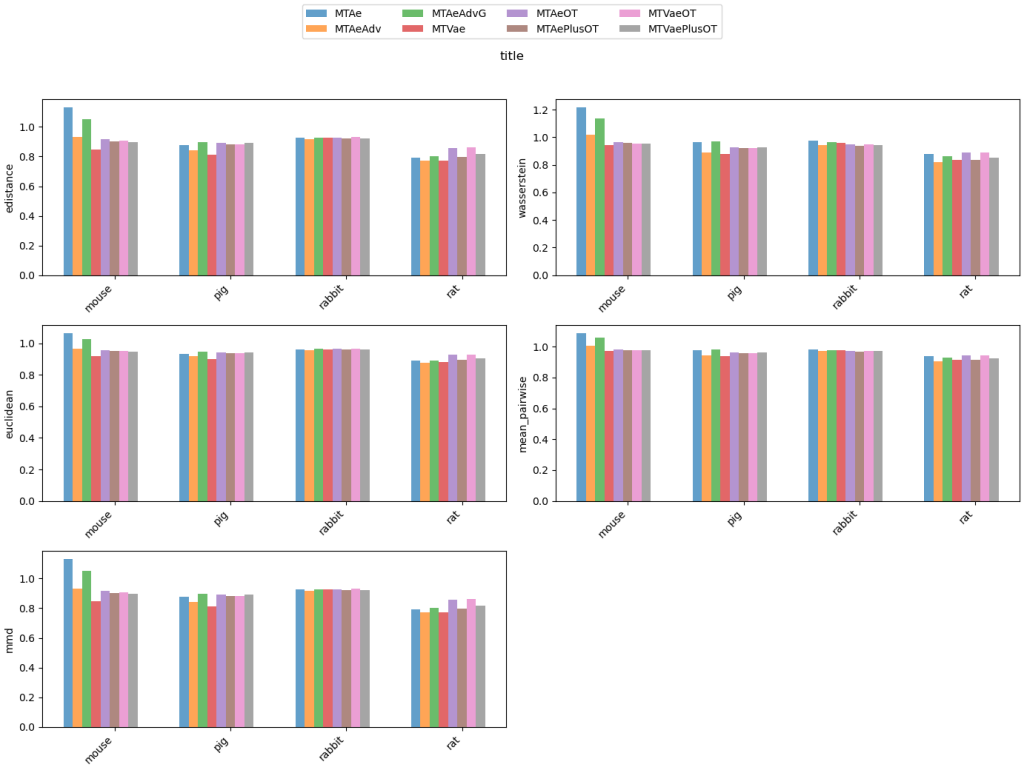


Figure 22: Distance metrics of multi-task models for the cross-species

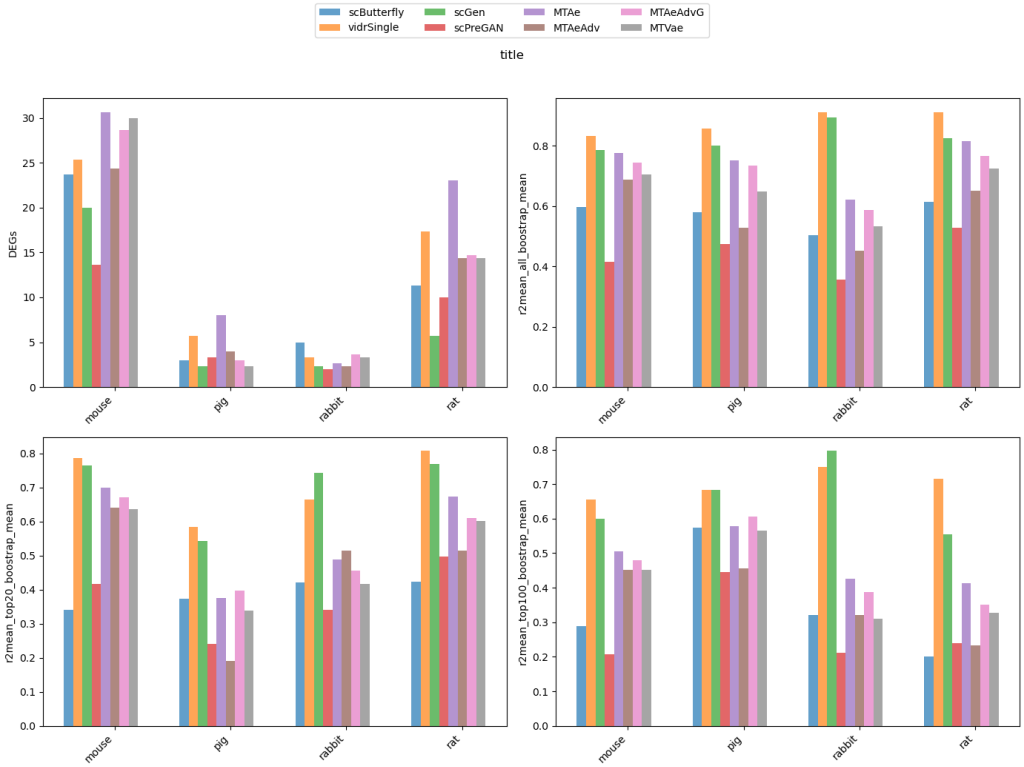


Figure 23: Baseline metrics of multi-task and literature models for the cross-species

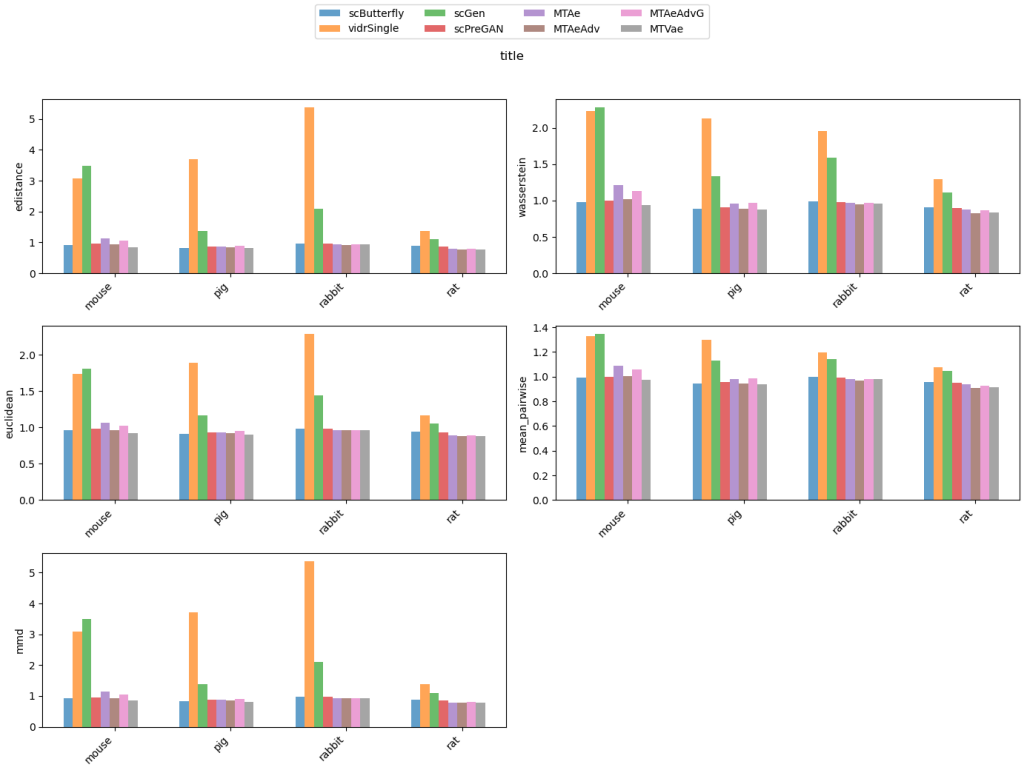


Figure 24: Distance metrics of multi-task and literature models for the cross-species

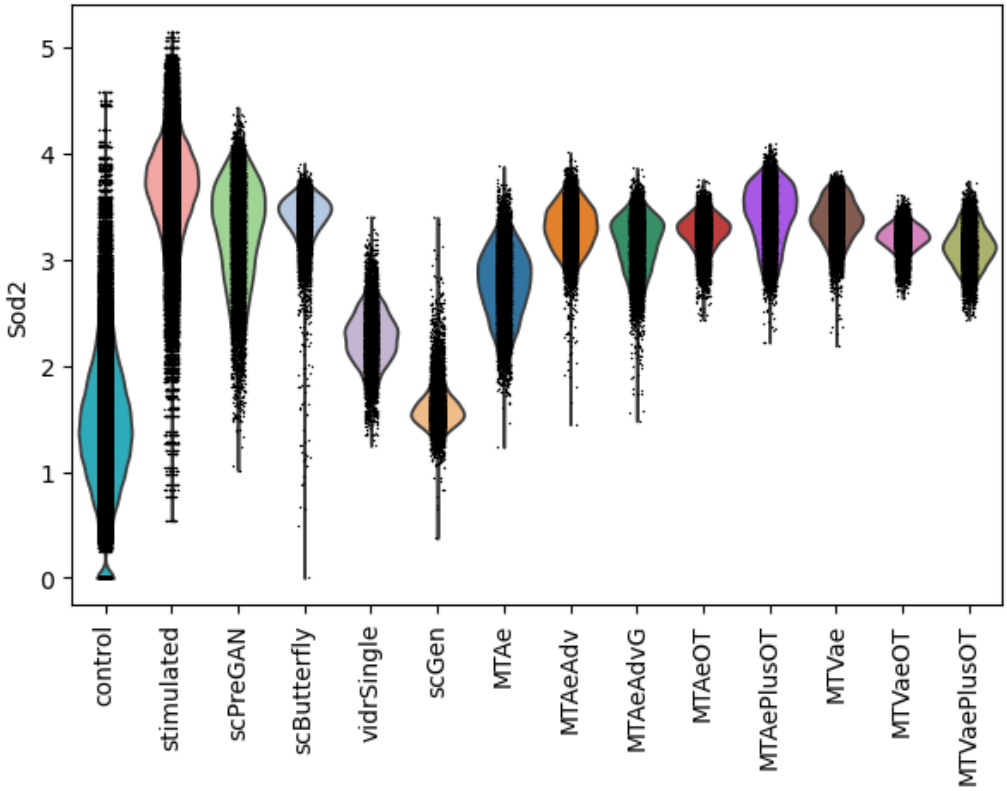

Sod2
0
1
2
3
4
5
control
stimulated
scPreGAN
scButterfly
vidrSingle
scGen
MTAe
MTAeAdv
MTAeAdvG
MTAeOT
MTAePlusOT
MTVae
MTVaeOT
MTVaePlusOT

Figure 25

7.4 Nault et al.

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 9: Nault et al. [20, 21]

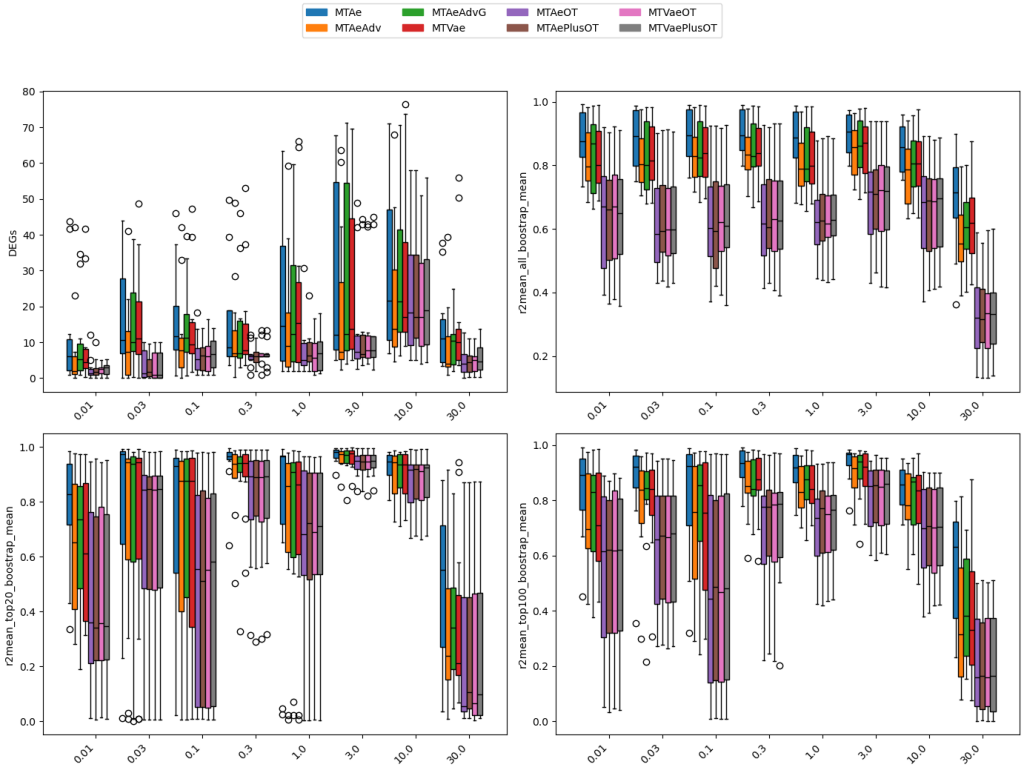


Figure 26: Baseline metrics of multi-task models for the Nault et al. [20, 21] dataset across dosages

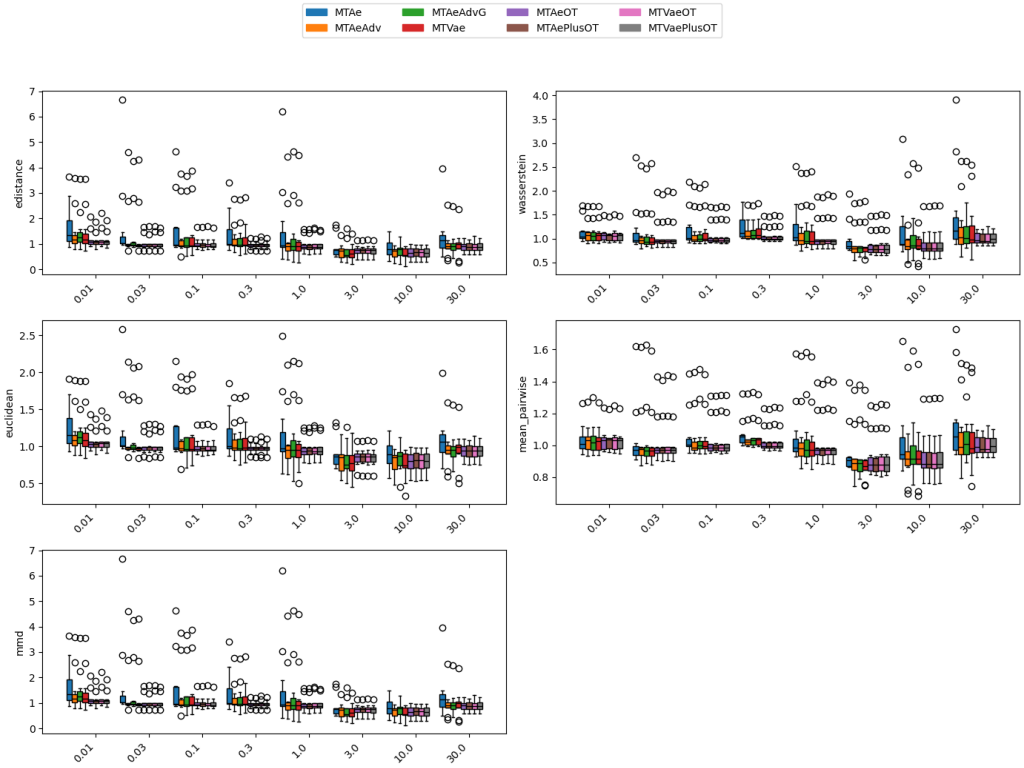


Figure 27: Distance metrics of multi-task models for the Nault et al. [20, 21] dataset across dosages

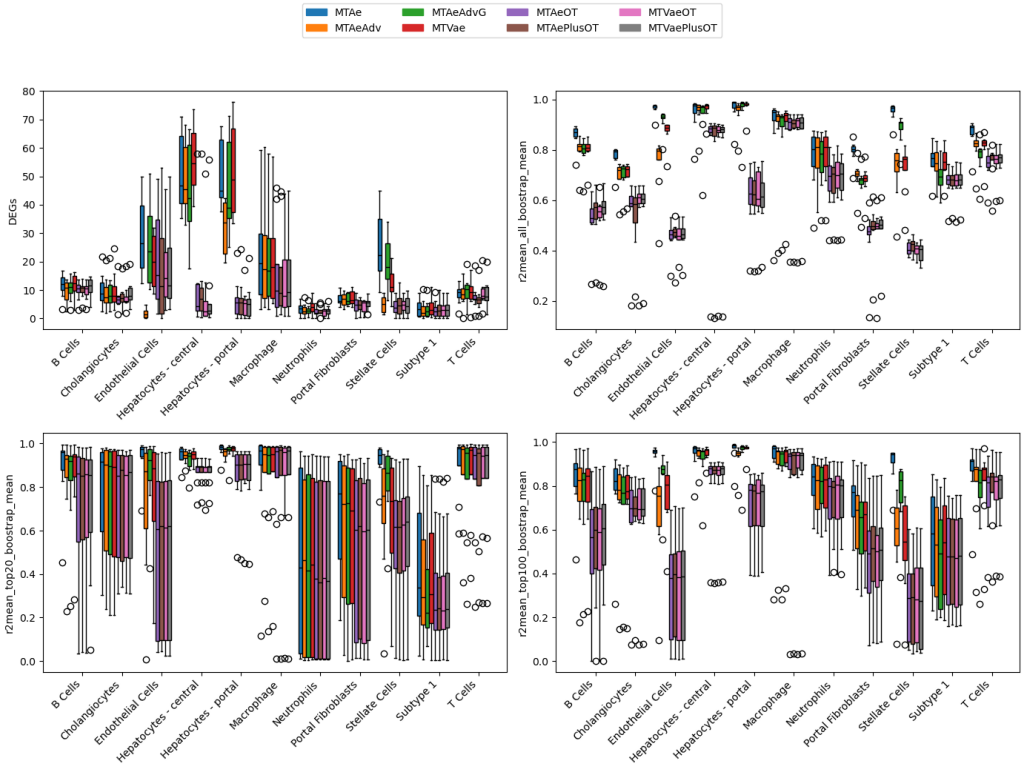


Figure 28: Baseline metrics of multi-task models for the Nault et al. [20, 21] dataset across cell types

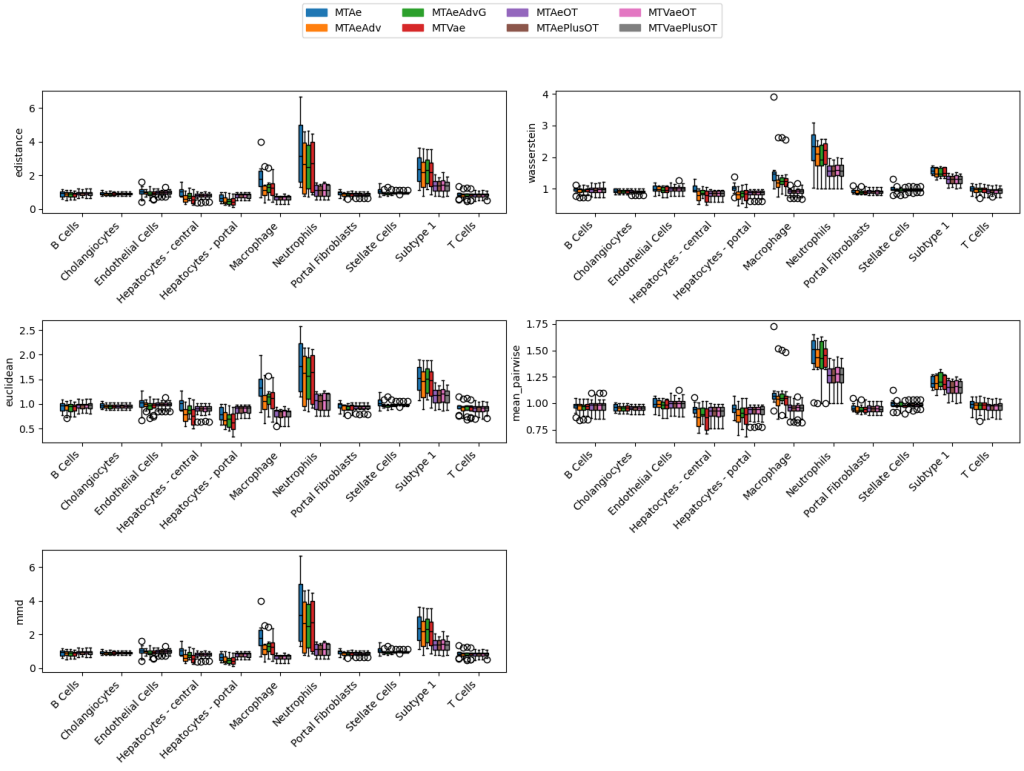


Figure 29: Distance metrics of multi-task models for the Nault et al. [20, 21] dataset across cell types

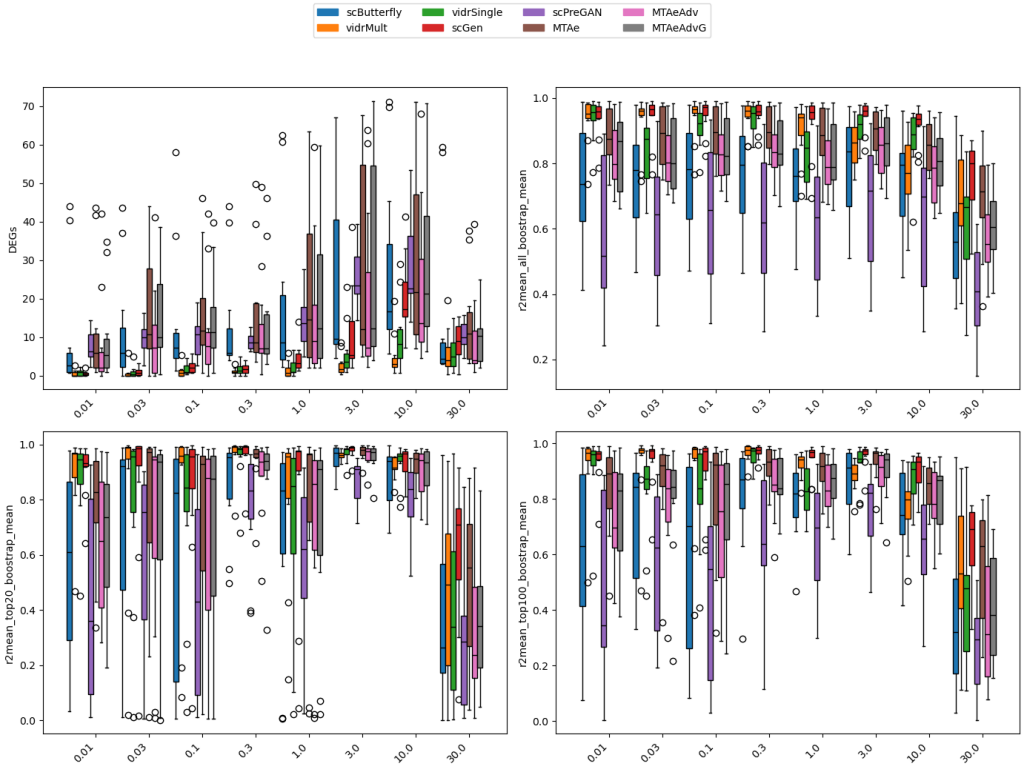


Figure 30: Baseline metrics of multi-task and literature models for the Nault et al. [20, 21] dataset across dosages

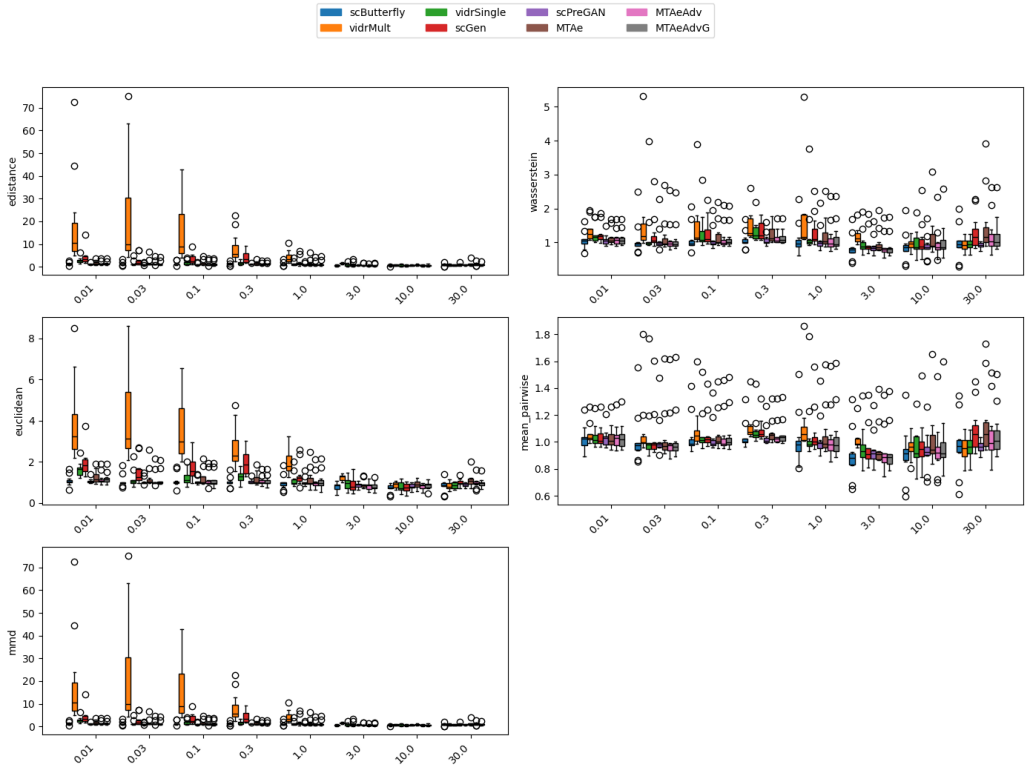


Figure 31: Distance metrics of multi-task and literature models for the Nault et al. [20, 21] dataset across dosages

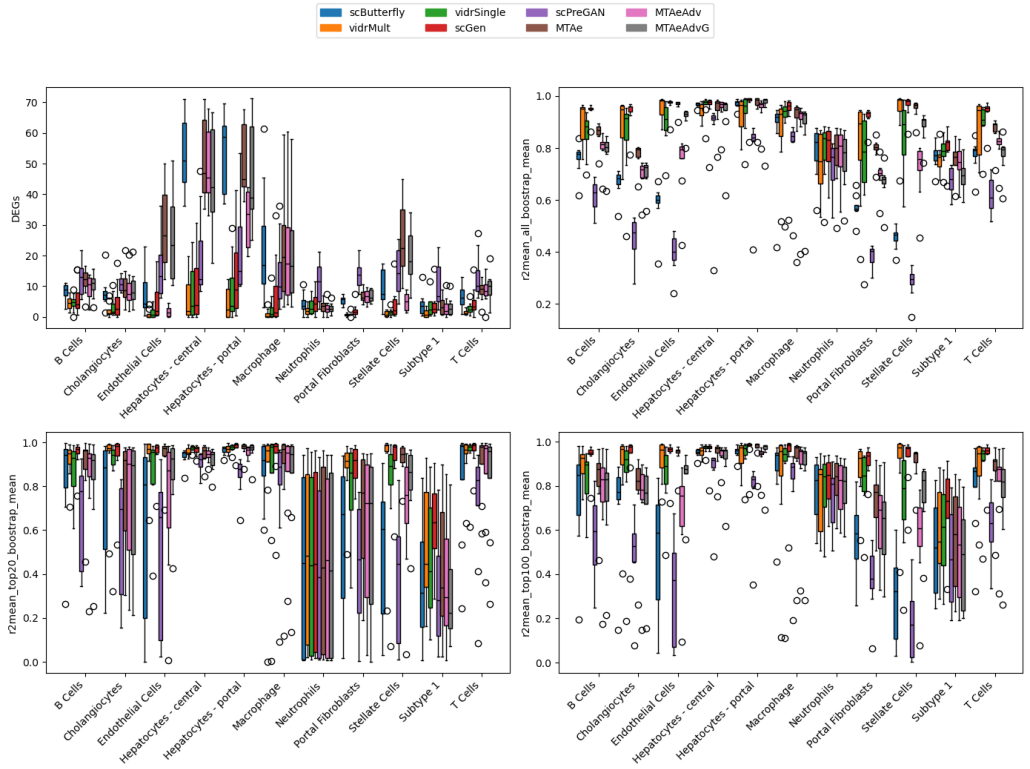


Figure 32: Baseline metrics of multi-task and literature models for the Nault et al. [20, 21] dataset across cell types

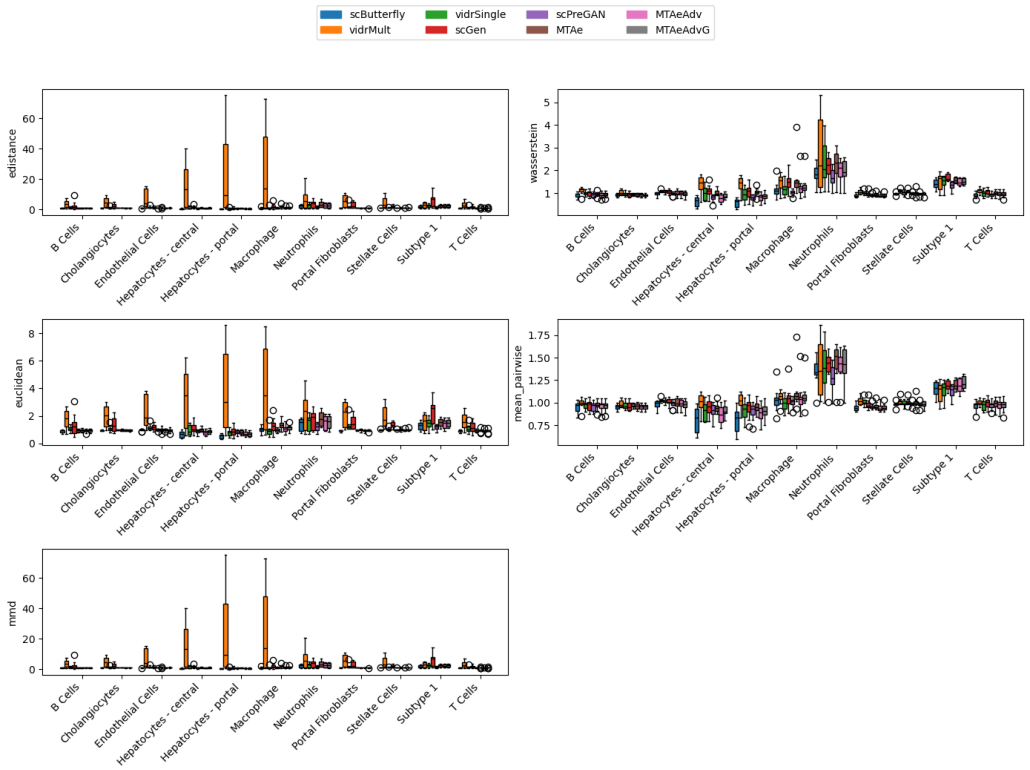


Figure 33: Distance metrics of multi-task and literature models for the Nault et al. [20, 21] dataset across cell types

7.5 Knowledge transfer

which tasks and why they are important?

7.6 TODO

- interpretability
- explainability
- integration with multiple omics

8 Interpretability

9 Conclusion and future work

We have validated the potential of scButterfly to perturbation modeling with the multi-dosage dataset of Nault et al. [20, 21], an addition of the author's study that is based only on the dataset of Kang et al. [14]. 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.

Another significant benefit of our model is its lower complexity while scaling to a larger number of perturbations (scalability) [1].

It should be noted that one of the limitations of our methods is the transductive learning with respect to the perturbations. The perturbation signal is one-hot encoded, thus limiting the model to generalize to unseen ones. Thus, future work could include to explore how inductive learning with respect to the perturbations could be integrated to our architecture, enabling extrapolation to unseen perturbations.

10 Code availability

The code for the models and the experiments is available at `https://github.com/thodkatz/thesis`.

References

[1] Stephan Allenspach, Jan A. Hiss, and Gisbert Schneider. Neural multi-task learning in drug design. 6(2):124–137.

[2] Stephan Allenspach, Jan A Hiss, and Gisbert Schneider. Neural multi-task learning in drug design. *Nature Machine Intelligence*, 6(2):124–137, 2024.

[3] Chester I Bliss. The toxicity of poisons applied jointly 1. *Annals of applied biology*, 26(3):585–615, 1939.

[4] 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.

[5] James C Costello, Laura M Heiser, Elisabeth Georgii, Mehmet Gönen, Michael P Menden, Nicholas J Wang, Mukesh Bansal, Muhammad Ammad-Ud-Din, Petteri Hintsanen, Suleiman A Khan, et al. A community effort to assess and improve drug sensitivity prediction algorithms. *Nature biotechnology*, 32(12):1202–1212, 2014.

[6] 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.

[7] Yicheng Gao, Zhiting Wei, Kejing Dong, Jingya Yang, Guohui Chuai, and Qi Liu. Toward subtask decomposition-based learning and benchmarking for genetic perturbation outcome prediction and beyond.

[8] 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.

[9] 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.

[10] 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.

[11] Douglas R Hofstadter. Analogy as the core of cognition. *The analogical mind: Perspectives from cognitive science*, pages 499–538, 2001.

[12] Francesco Iorio, Theo A Knijnenburg, Daniel J Vis, Graham R Bignell, Michael P Menden, Michael Schubert, Nanne Aben, Emanuel Gonçalves, Syd Barthorpe, Howard Lightfoot, et al. A landscape of pharmacogenomic interactions in cancer. *Cell*, 166(3):740–754, 2016.

[13] 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.

[14] 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.

[15] 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.

[16] Min Li, Yake Wang, Ruiqing Zheng, Xinghua Shi, Yaohang Li, Fang-Xiang Wu, and Jianxin Wang. Deepdsc: a deep learning method to predict drug sensitivity of cancer cell lines. *IEEE/ACM transactions on computational biology and bioinformatics*, 18(2):575–582, 2019.

[17] S Loewe. The problem of synergism and antagonism of combined drugs. *Arzneimittelforschung*, 3(6):285–290, 1953.

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

[19] Harry L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

[20] 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.

[21] 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.

[22] Jennifer O'Neil, Yair Benita, Igor Feldman, Melissa Chenard, Brian Roberts, Yaping Liu, Jing Li, Astrid Kral, Serguei Lejnine, Andrey Loboda, et al. An unbiased oncology compound screen to identify novel combination strategies. *Molecular cancer therapeutics*, 15(6):1155–1162, 2016.

[23] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[24] Kristina Preuer, Richard PI Lewis, Sepp Hochreiter, Andreas Bender, Krishna C Bulusu, and Günter Klambauer. Deepsynergy: predicting anti-cancer drug synergy with deep learning. *Bioinformatics*, 34(9):1538–1546, 2018.

[25] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks.

[26] Stefan Schrod, Helena U Zacharias, Tim Beißbarth, Anne-Christin Hauschild, and Michael Altenbuchinger. Codex: Counterfactual deep learning for the in silico exploration of cancer cell line perturbations. *Bioinformatics*, 40(Supplement_1):i91–i99, 2024.

[27] Pir Masoom Shah, Huimin Zhu, Zhangli Lu, Kaili Wang, Jing Tang, and Min Li. Deepdtagen: a multitask deep learning framework for drug-target affinity prediction and target-aware drugs generation. *Nature Communications*, 16(1):1–15, 2025.

[28] Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which Tasks Should Be Learned Together in Multi-task Learning?

[29] Aravind Subramanian, Rajiv Narayan, Steven M Corsello, David D Peck, Ted E Natoli, Xiaodong Lu, Joshua Gould, John F Davis, Andrew A Tubelli, Jacob K Asiedu, et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452, 2017.

[30] 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.

[31] Xu Tan, Long Hu, Lovelace J Luquette III, Geng Gao, Yifang Liu, Hongjing Qu, Ruibin Xi, Zhi John Lu, Peter J Park, and Stephen J Elledge. Systematic identification of synergistic drug pairs targeting hiv. *Nature biotechnology*, 30(11):1125–1130, 2012.

[32] 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.

[33] 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.

[34] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

[35] Bhagwan Yadav, Krister Wennerberg, Tero Aittokallio, and Jing Tang. Searching for drug synergy in complex dose–response landscapes using an interaction potency model. *Computational and structural biotechnology journal*, 13:504–513, 2015.

[36] Hengshi Yu and Joshua D. Welch. PerturbNet predicts single-cell responses to unseen chemical and genetic perturbations.

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