PREDICTING SINGLE-CELL PERTURBATION RESPONSES FOR UNSEEN DRUGS

Leon Hetzel[*][†]
Technical University Munich
Helmholtz Munich

Simon Böhm[*][†]
Helmholtz Munich

Niki Kilbertus[†]
Technical University Munich
Helmholtz Munich

Stephan Günnemann[‡]
Technical University Munich

Mohammad Lotfollahi[†]
Helmholtz Munich

Fabian Theis[†]
Technical University Munich
Helmholtz Munich

ABSTRACT

Single-cell transcriptomics enabled the study of cellular heterogeneity in response to perturbations at the resolution of individual cells. However, scaling high-throughput screens (HTSs) to measure cellular responses for many drugs remains a challenge due to technical limitations and, more importantly, the cost of such multiplexed experiments. Thus, transferring information from routinely performed bulk RNA-seq HTS is required to enrich single-cell data meaningfully. We introduce a new encoder-decoder architecture to study the perturbational effects of unseen drugs. We combine the model with a transfer learning scheme and demonstrate how training on existing bulk RNA-seq HTS datasets can improve generalisation performance. Better generalisation reduces the need for extensive and costly screens at single-cell resolution. We envision that our proposed method will facilitate more efficient experiment designs through its ability to generate in-silico hypotheses, ultimately accelerating targeted drug discovery.

1 INTRODUCTION

Recent advances in single-cell methods allowed the simultaneous analysis of millions of cells, increasing depth and resolution to explore cellular heterogeneity (Sikkema et al., 2022; Han et al., 2020). With single-cell RNA sequencing (scRNA-seq) and high-throughput screens (HTSs) one can now study the impact of different perturbations, i.e., drug-dosage combinations, on the transcriptome at cellular resolution (Yofe et al., 2020; Norman et al., 2019). Unlike conventional HTSs, scRNA-seq HTSs can identify subtle changes in gene expression and cellular heterogeneity, constituting a cornerstone for pharmaceutics and drug discovery (Srivatsan et al., 2020). Nevertheless, these newly introduced sample multiplexing techniques (McGinnis et al., 2019; Stoeckius et al., 2018; Gehring et al., 2018) require expensive library preparation and do not scale to screen thousands of distinct molecules. Even in its most cost-effective version, nuclear hashing, the acquired datasets contain no more than 200 different drugs (Srivatsan et al., 2020).

Consequently, computational methods are required to address the limited exploration power of existing experimental methods and discover promising therapeutic drug candidates. Suitable methods should predict the response to unobserved (combinations of) perturbations. Increasing in difficulty, such tasks may include inter- and extrapolation of dosage values, the generalisation to unobserved (combinations of) drug-covariates (e.g., cell-type), or predictions for unseen drugs. In terms of medical impact, the prediction of unobserved perturbations may be the most desirable, for example for drug repurposing. At the same time, it requires the model to properly capture complex chemical interactions within multiple distinct cellular contexts. Such generalisation capabilities can not yet be learned from single-cell HTSs alone, as they supposedly do not cover the required breadth of chemical interactions. In this work, we leverage information across datasets to alleviate this issue.

[*] contributed equally
[†] first.last@helmholtz-muenchen.de
[‡] guennemann@in.tum.de

We propose a new model that generalises previous work on Fader Networks by Lample et al. (2017) and the CPA model by Lotfollahi et al. (2021) to the challenging scenario of generating counterfactual predictions for unseen compounds. Our method is as flexible and interpretable as CPA but further enables us to leverage lower resolution but higher throughput assays, such as bulk RNA-seq HTSs, to improve the model's generalisation performance on single-cell data (Amodio et al., 2021). Our main contributions are:

1. We introduce chemCPA, a model that incorporates knowledge about the compounds' structure, enabling the prediction of drug perturbations at a single-cell level from molecular representations.
2. We benchmark different molecule encoding networks for their applicability in the presented perturbation context.
3. Finally, we propose and evaluate a transfer learning scheme to leverage HTS bulk RNA-seq data in the setting of both identical and different gene sets between the source (bulk) and target (single-cell) datasets.

2 Related Work

Over the past years, deep learning (DL) has become an essential tool for analysis (Lopez et al., 2020) and interpretation (Rybakov et al., 2020) of scRNA-seq data. Representation learning in particular, has been useful not only for identifying cellular heterogeneity and integration (Gayoso et al., 2022), or mapping query to reference datasets (Lotfollahi et al., 2022), but also in the context of modelling single-cell perturbation responses (Rampášek et al., 2019; Seninge et al., 2021; Lotfollahi et al., 2019; Ji et al., 2021).

Unlike linear models (Dixit et al., 2016; Kamimoto et al., 2020) or mechanistic approaches (Fröhlich et al., 2018; Yuan et al., 2021), DL is suited to capture non-linear cell-type-specific responses and easily scales to genome-wide measurements. Recently, Lotfollahi et al. (2021) introduced the computational perturbation auto-encoder (CPA) for modelling perturbations on scRNA-seq data. CPA does not generalise to unseen compounds, hindering its application to virtual screening of drugs not yet measured via scRNA-seq data, which is required for effective drug discovery.

For bulk RNA-seq data, on the other hand, several methods have been proposed to predict gene expression profiles for de novo chemicals (Pham et al., 2021; Zhu et al., 2021; Umarov et al., 2021). Crucially, the L1000 dataset, introduced by the LINCS programme (Subramanian et al., 2017), greatly facilitated such advances on phenotype-based compound screening. However, it remains unclear how to translate these approaches to single-cell datasets that include significantly fewer compounds and, in many cases, rely on different gene sets.

3 Chemical Compositional Perturbation Autoencoder

We consider a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N = \{(x_i, (d_i, s_i, c_i))\}_{i=1}^N$, where $x_i \in \mathbb{R}^n$ describes the n-dimensional gene expression and y_i an attribute set. For scRNA-seq perturbation data, we usually consider the drug and dosage attributes, $d_i \in \{\text{drugs in } \mathcal{D}\}$ and $s_i \in \mathbb{R}$, respectively, and the cell-line c_i of cell i. Note that this set of attributes $\mathcal{Y}$ depends on the available data and could be extended to covariates such as patient, or species.

A possible approach to predicting counterfactual combinations is to encode a cell's gene expression x_i invariantly from its attributes y_i as a latent vector z_i, called the basal state. Afterwards, z_i can be combined with $z_{\text{attribute}}$ to encode any attribute combination $y_i' \neq y_i$, and decoded back to a gene expression state $\hat{x}_i$ that corresponds to this new set of chosen attributes. To this end, we divide our model into three parts: (1) the gene expression encoder and decoder, (2) the attribute embedders, and (3) the adversarial classifiers, see Figure 1 for an illustration.

3.1 Gene encoder and decoder

Following Lotfollahi et al. (2021), our model is based on an encoder-decoder architecture combined with adversarial training. The encoding network $E_\theta : \mathbb{R}^n \to \mathbb{R}^l$ is a multi-layer perceptron (MLP)

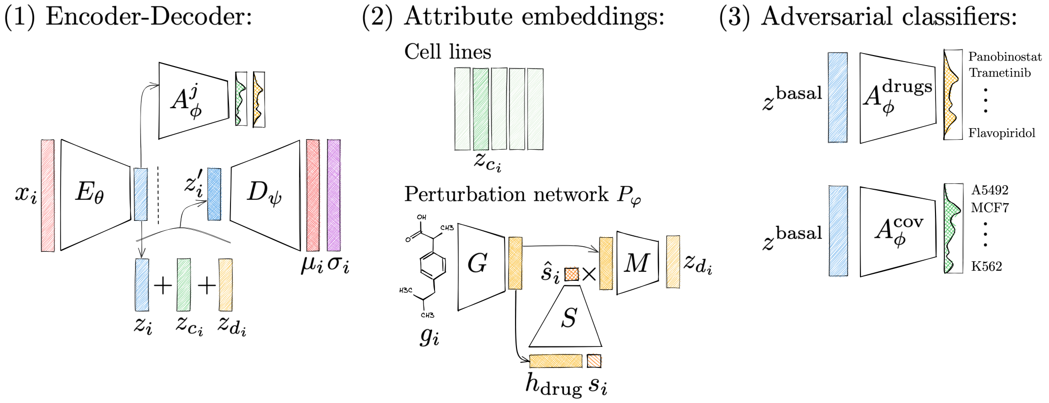


Figure 1: Architecture of chemCPA. The model consists of three parts: (1) the encoder-decoder architecure, (2) the attribute embeddings, and (3) the adversarial classifiers. The molecule encoder G can be any graph- or language-based model as long as it generates fixed-sized embeddings h_{drugs}. The MLPs S and M are trained to map the embeddings to the perturbational latent space. There, z_{d_i} is added to the basal state z_i and the covariate embedding z_{c_i}. In this work, the latter always corresponds to cell lines. The basal state $z_i = E_\theta(x_i)$ is trained to be invariant through adversarial classifiers A^j_ϕ and the decoder D_ψ gives rise to the Gaussian likelihood $P(x_i \,|\, \mu_i, \sigma_i)$.

with parameters θ that maps a measured gene expression state $x_i \in \mathbb{R}^n$ to its l-dimensional latent vector $z_i = E_\theta(x_i)$. Through adversarial classifiers, z_i is trained to not contain any information about its attributes y_i. We assume an additive structure of the perturbation response in the latent space, i.e., for a new set of independently chosen attributes we compute the new latent state as $z'_i = z_i + z_{\text{attribute}}$. The decoder $D_\psi : \mathbb{R}^l \to \mathbb{R}^{2n}$ is also an MLP that takes z'_i as input and computes the component-wise means $\mu = D^\mu_\psi(z')$ and variances $\sigma^2 = D^{\sigma^2}_\psi(z')$ of normal distributions describing the attribute dependent gene expression state. During training, we use a Gaussian likelihood reconstruction loss:

$$\mathcal{L}_{\text{rec}}(\theta, \psi) = \frac{1}{2}\left[\ln\left(\max\left(D^{\sigma^2}_\psi(z'), \epsilon\right)\right) + \frac{\left(D^\mu_\psi(z') - x\right)^2}{\max\left(D^{\sigma^2}_\psi(z'), \epsilon\right)}\right] \quad \text{with} \quad z' = E_\theta(x) + z_{\text{attribute}}\ ,$$

where $\epsilon = 10^{-6}$ is used for stability and $\max$ is evaluated component-wise. Next, we provide intuition about how we can meaningfully interpret latent space arithmetics and how we encode drug and cell-line attributes.

3.2 Attribute embedding

Due to their different nature, we encode the drug attributes and cell-line information separately. For the cell-lines we use the same approach as Lotfollahi et al. (2021), where we optimised a d-dim latent representation z_c for each cell-line c.

For the drugs, we train an embedding network that maps molecular representations—such as its graph or SMILES representation—and the used dosage to its latent perturbation state. The drug perturbation network P_φ consists of the molecule encoder, the dosage scaler, and the perturbation encoder, see Figure 1 (2). First, the molecule encoder $G : \mathcal{G} \to \mathbb{R}^m$ encodes the molecule representation $g_i \in \mathcal{G}$ as a fixed size embedding $h_{\text{drug}} \in \mathbb{R}^m$. Next, the dosage scaler $S : \mathbb{R}^{m+1} \to \mathbb{R}$ maps h_{drug} together with s_i to the scaled dosage value $\hat{s}_i$. In a final step, the perturbation encoder $M : \mathbb{R}^m \to \mathbb{R}^l$ takes the scaled molecular embedding $\hat{s}_i h_{d_i}$ as input and generates the final drug perturbation $z_{d_i} \in \mathbb{R}^l$. Put together, we end up with

$$z_{d_i} = P_\varphi(g_i, s_i) = M\big(S(h_{d_i}, s_i) \times h_{d_i}\big) = M\big(S\big(G(g_i), s_i\big) \times G(g_i)\big) \quad .$$

The molecule encoder G can be any encoding network that maps molecular representations to fixed-size embeddings. Due to the limited number of drugs available in scRNA-seq HTSs, we rely on different pretrained encoding models. In this work, we include GROVER (Rong et al., 2020), a larger transformer-based model, pretrained on tasks ranging from molecule reconstruction to property prediction, JT-VAE (Jin et al., 2018), an autoencoding model acting on commonly occurring graph-level motifs, as well as three simpler models, a GCN (Kipf & Welling, 2016), an MPNN (Gilmer et al., 2017), and a weave model (Kearnes et al., 2016), which are all pretrained and were taken from DGL-LifeSci (Li et al., 2021). As a baseline, we also include RDKit fingerprints, a hash-based embedding that is easily available for any molecular graph (Landrum, 2016).

These design decisions allow us to also generate drug perturbations for molecules that have not been experimentally measured ($d \notin \mathcal{D}$) in an interpretable and semantically meaningful way by computing the latent attribute vector as

$$z_i' = z_i + z_{\text{attribute}} = z_i + z_{d_i} + z_{c_i} \ .$$

Intuitively, we want z_i' to be decoded to a (distribution over) transcriptome(s) that would result from applying the perturbations encoded by z_{d_i}, z_{c_i} to the "basal state" of x_i, i.e., as if no perturbations were made, which we aim to encode by z_i. Since training examples have already been perturbed, we next describe how we "strip z_i from the attribute information" to obtain a basal state representation.

3.3 Adversarial classifiers for invariant basal states

To generate invariant basal states and produce disentangled representations z_i, z_{d_i}, and z_{c_i}, we use adversarial classifiers A_ϕ^{drug} and A_ϕ^{cov}. Both adversary networks $A_\phi^j : \mathbb{R}^l \to \mathbb{R}^{N_j}$ take the latent basal state z_i as input and aim to predict the drug that has been applied to example i as well as its cell-line c_i. While these classifiers are trained to improve classification performance, we also add the classification loss *with a reversed sign* to the training objective for the encoder E_θ. Hence, the encoder attempts to produce a latent representation z_i *which contains no information about the attributes*. We use the cross-entropy loss for both classifiers

$$\mathcal{L}_{\text{class}}^{\text{drugs}} = \text{CE}\big(A_\phi^{\text{drug}}(z_i)\big), d_i) \quad \text{and} \quad \mathcal{L}_{\text{class}}^{\text{cov}} = \text{CE}\big(A_\phi^{\text{cov}}(z_i), c_i\big) \ .$$

To improve the robustness of the adversarial classifiers to measurement noise of gene expressions, we also add a gradient penalty (Gulrajani et al., 2017) to their objective:

$$\mathcal{L}_{\text{pen}}^j = \big\|\partial_{z_i} \sum_k A_\phi^j(z_i)_k\big\|_2 \ .$$

In practice, we alternate SGD update steps between the following competing objectives

$$\mathcal{L}_{\text{AE}}(\theta, \psi, \varphi | \phi) = \mathcal{L}_{\text{rec}}(\theta, \psi, \varphi) - \lambda_{\text{dis}} \sum_j \mathcal{L}_{\text{class}}^j(\theta \,|\, \phi) \quad \text{and}$$

$$\mathcal{L}_{\text{Adv}}(\phi \,|\, \theta) = \sum_j \mathcal{L}_{\text{class}}^j(\phi \,|\, \theta) + \lambda_{\text{pen}} \mathcal{L}_{\text{pen}}^j(\phi) \ ,$$

where λ_{dis} balances the importance of good reconstruction against the encoder E_θ's constraint to generate disentangled basal states z_i. The gradient penalty is weigthed with λ_{pen}.

4 Datasets and transfer learning

We use the sci-Plex3 (Srivatsan et al., 2020) and the L1000 (Subramanian et al., 2017) datasets for the single-cell and bulk experiments, respectively. To obtain pretrained chemCPA models, different molecule encoders G are benchmarked in the context of perturbation screens on the L1000 dataset. The sci-Plex3 data is used for the main experiments concerning chemCPA's ability to generalise to unseen drugs.

The L1000 data contains about 1.3 million bulk RNA-seq observations for 978 different genes. It includes measurements for almost 20k different drugs, some of which are FDA-approved, while others are synthetic compounds with no proven effect on any disease. The large size of the L1000 data allows to explore a more diverse space of molecules and, therefore, makes it ideal for pretraining. The sci-Plex3 data is similar in size and contains measurements for 649,340 cells across 7561 drug-sensitive genes. On three human cancer cell lines—A549, MCF7, and K562—single-compound perturbations for 188 drugs at four different dosages are examined. Note that all cell lines and about 150 compounds overlap with the L1000 data. In addition, Srivatsan et al. (2020) assigned to all compounds one of 19 different modes of action (MoA) or pathways. These allow us to assess the embedding quality of different perturbation networks P_φ.

As the decoder network D_ψ assumes normally distributed data, the dataset was first normalised and then $\log(x+1)$-transformed. After preprocessing, we were left with $840{,}677$ (across $17{,}203$ compounds) and $581{,}776$ observations for the L1000 and sci-Plex data, respectively. Depending on the experiment, we further reduced the number of genes included in the single-cell data. In Section 5.2.1, we subsetted both datasets to the same 977 genes found via ensemble gene annotations. In Section 5.2.2, we hypothesize that more than the L1000 genes are required to capture the variability within the single-cell data. To assess whether pretraining on L1000 is still beneficial in this scenario, we included 1023 highly variable genes (HVGs) for the sci-Plex3 data. That is, we consider 2000 genes in total.

For the extended gene set, we perform a version of architecture surgery to match the data dimension $(977 \rightarrow 2000)$ with the models' dimensions. This is realised by additional linear input and output layers for the encoder E_θ and decoder D_ψ, respectively. Moreover, as the computational cost is negligible compared to the generation cost of the perturbation datasets, we did not freeze any weights during the fine-tuning on single-cell data.

5 Experiments

Throughout our experiments, we use the coefficient of determination r^2 as the main performance metric. This score is computed between the actual measurements and the counterfactual predictions on all genes and the 50 most differentially expressed genes (DEGs) per compound. Considering all genes is necessary to evaluate the background and general decoder performance. However, the resulting r^2-scores can get inflated since many genes stay similar to their controls. In contrast, the DEGs capture the differential signal which reflect a drug's effect. For the counterfactual prediction, chemCPA encodes only control observations—treated with DMSO—across all cell lines and combines the resulting basal states with the respective attribute encoding. Note that poor disentanglement will automatically lead to low scores due to computing the test score on the counterfactual prediction.

For a fair comparison, it is essential to account for the disentanglement scores of a trained model. An entangled model would confound the drug and cell line effects in the basal state, effectively becoming a standard autoencoder. In order to classify the degree of disentanglement, we train a separate MLP with four layers over 400 epochs and compute the prediction accuracy given the basal state. This way we obtain the disentanglement score for both drugs and cell lines.

We start with the experiment on the L1000 dataset and then report our results on the sci-Plex3 data.

5.1 Benchmarking drug molecule encoders on the L1000 dataset

Enabled by the flexibility of the molecule encoder G, we investigated what impact the architecture choice has on the performance of chemCPA. For this, we compared multiple pretrained graph-based models whose weights were frozen during the training. Next to predefined RDKit fingerprints, we included a GCN, MPNN, weave model, GROVER model, and a JT-VAE.

The optimal disentanglement scores correspond to the most abundant drug and cell line ratios, which amounts to $< 1\%$ and $\sim 2\%$ on the L1000 data, respectively. Since no model reaches perfect scores, we decided to first subselect on perturbation disentanglement scores $< 20\%$ and then compared the five best-disentangled models concerning the cell lines. The encoder's E_θ, decoder's D_ψ, and adversary networks' A^j_ϕ width and depth parameters were optimised with the RDKit features, and their best combination was chosen and fixed throughout the comparison. The dimensionality of the latent space is always 32.

Table 1 summarises the results of this experiment. The weave model performs much worse than the others, achieving an r^2-score of only 65 ± 8 on DEGs. While the GCN disentangles well, it cannot keep up with the predictive performance of the JT-VAE and GROVER models. Therefore,

Table 1: Summary of chemCPA on the L1000 dataset for different molecule encoders G. All models were trained on the same random split. Reported are the overall disentanglement scores (drug and cell line) and the r^2-scores on the test set.

Model G	Drug	Cell line	Mean r^2 all	Mean r^2 DEGs
GCN	$\mathbf{0.08 \pm 0.03}$	$\mathbf{0.17 \pm 0.01}$	0.92 ± 0.01	0.81 ± 0.05
MPNN	0.10 ± 0.03	0.28 ± 0.07	0.92 ± 0.01	0.82 ± 0.03
GROVER	0.09 ± 0.03	0.19 ± 0.04	0.93 ± 0.01	$\mathbf{0.87 \pm 0.01}$
JT-VAE	$\mathbf{0.08 \pm 0.02}$	0.20 ± 0.04	0.93 ± 0.01	$\mathbf{0.87 \pm 0.01}$
RDKit	0.10 ± 0.04	0.29 ± 0.13	0.93 ± 0.01	0.85 ± 0.03
weave	0.10 ± 0.03	0.29 ± 0.09	0.89 ± 0.02	0.65 ± 0.08

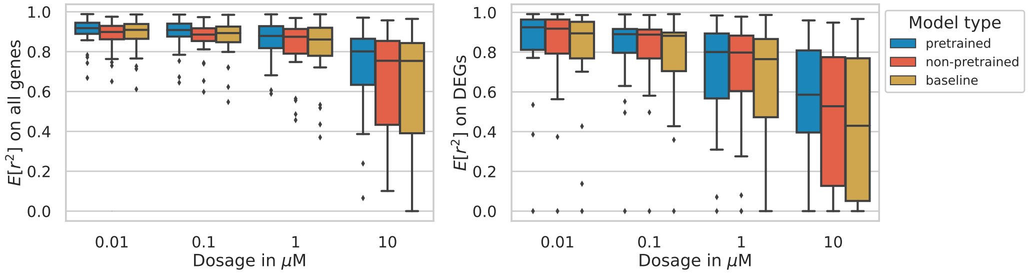


Figure 2: Performance of the pretrained and non-pretrained chemCPA model using GROVER. Comparisons against the baseline are done on both the complete gene set (977 genes) and the compound specific DEGs (50 genes). In both cases, the pretrained model shows the best performance. At $10\,\mu$M on the DEGs, more than 50% of the predictions have an r^2 score > 0.59 while the baseline's median is slightly above 0.4.

we decided to use the best runs—based on the validation score—for GROVER and JT-VAE as initialisation for fine-tuning on the single-cell data. Due to its simplicity and good performance, we also include RDKit features.

5.2 Applying chemCPA to single-cell data

For the application to the sci-Plex3 data, we considered two settings. The first setting ignores the technological difference between bulk and single-cell, and the same 977 genes from the L1000 data are used for fine-tuning. The second setting accounts for the variance of single-cell data resulting mostly from its HVGs. We therefore extend the single-cell gene set with 1023 HVGs. Note that on this larger gene set the 50 DEGs per compound become a subset of the HVGs, preventing fine-tuned models from leveraging learned bulk expression directly.

We excluded nine drugs from the training and validation for the data split. These drugs mostly belong to three MoA—epigenetic regulation, tyrosine kinase signalling, and cell cycle regulation—as these were reported among the most effective drugs in the original publication (Srivatsan et al., 2020). We used 4% of the data across all dosages, $10\,$nM, $100\,$nM, $1\,\mu$M, and $10\,\mu$M, for the validation. The drug and cell line thresholds at which we consider a model disentangled were set to 7% (optimal is 3%) and 70% (optimal is 51%) respectively. Across all runs, we observed that pretraining on L1000 leads to better covariate invariance scores: $(88 \pm 16)\%$ for non-pretrained models and $(79 \pm 16)\%$ for fine-tuned models. For the generalisation comparison, we selected the best performing model according to its validation performance for each of the three chemCPA versions: RDKit, GROVER, and JT-VAE.

Table 2: Performance of pretrained and non-pretrained chemCPA models across the three versions of the molecule encoder G for the shared gene set. Since drug effects are stronger for high dosages, the scores are evaluated at a dosage value of 10 μM.

Model G	Type	Mean r^2 all	Mean r^2 DEGs	Median r^2 all	Median r^2 DEGs
	baseline	0.63	0.42	0.75	0.43
GROVER	non-pretrained	0.65	0.47	0.75	0.53
	pretrained	0.73	0.56	0.80	0.59
JT-VAE	non-pretrained	0.59	0.44	0.72	0.50
	pretrained	0.71	0.53	0.79	0.51
RDKit	non-pretrained	0.66	0.47	0.78	0.52
	pretrained	**0.78**	**0.64**	**0.84**	**0.72**

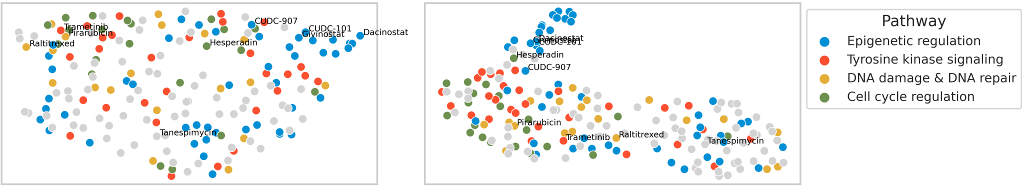
5.2.1 Generalisation to unseen drugs in the setting of shared gene sets

Table 2 shows the test performance of chemCPA for the nine unseen drugs across the three cell lines in the first scenario with identical genes. The reported baseline performance reflects the case when the model predicts the gene expression of a control cell, ignoring the drug's influence. This choice allows to estimate the difference between the treated cell's gene expression and their control (unperturbed) version. The fine-tuned models consistently outperform the baseline and their non-pretrained version at the highest dose with RDKit achieving the highest median score on DEGs.

Figure 2 illustrates that the drugs have a very subtle to no effect at low dosages, which is reflected by the high baseline scores. However, we see how the drug encoding, here GROVER, improves the prediction for high drug dosages. Looking at the predicted phenotype over all genes, the pretrained model has a significant advantage over its non-pretrained version, showing a clear improvement over the baseline. The results look worse when zoomed in to the DEGs. Nevertheless, also in this scenario, the fine-tuned model can explain gene expression values that must result from the drugs' influence. The results for RDKIT and JT-VAE are presented in A.4. In Figure 3, latent perturbations z_d are visualised for the lowest and highest dosage values. The drug perturbations do not show a clear grouping for the lowest dosage, indicating a sparse signal. In contrast, the stronger response in higher dosages is reflected by causing similar drugs with identical MoAs to cluster together, for example, the epigenetic drugs that cause histone deacetylation.

5.2.2 Generalisation to unseen drugs in the setting of extended gene sets

The extension to the larger gene set poses an even greater challenge to the chemCPA model. In Table 3, we show the same analysis as in Section 5.2.1. Strikingly, the advantage of the fine-tuned models translates to this scenario, while the non-pretrained models fail to beat the baseline. A more comprehensive view for all dosages is shown in Figure 4 for the GROVER model, see also A.4 for the RDKit and JT-VAE results.



UMAP of z_d for a dosage of 10 nM

UMAP of z_d for a dosage of 10 μM

Figure 3: Illustration of the the perturbation embedding z_d (GROVER) for the smallest and largest dosages. Drugs induce stronger perturbations with higher dosages which explains the improved clustering for 10 μM according to the drugs' pathways. The nine test compounds are labeled.

Table 3: We show the performance of pretrained and non-pretrained chemCPA models across the three versions of the molecule encoder G on the extended gene set. Since drug effects are stronger for high dosages, the scores are evaluated at a dosage value of 10 μM.

Model G	Type	Mean r^2 all	Mean r^2 DEGs	Median r^2 all	Median r^2 DEGs
	baseline	0.53	0.29	0.74	0.09
GROVER	non-pretrained	0.58	0.31	0.73	0.26
	pretrained	**0.68**	**0.41**	0.79	**0.47**
JT-VAE	non-pretrained	0.60	0.34	**0.80**	0.24
	pretrained	**0.68**	0.40	**0.80**	0.41
RDKit	non-pretrained	0.55	0.34	0.76	0.25
	pretrained	**0.68**	0.40	**0.80**	0.41

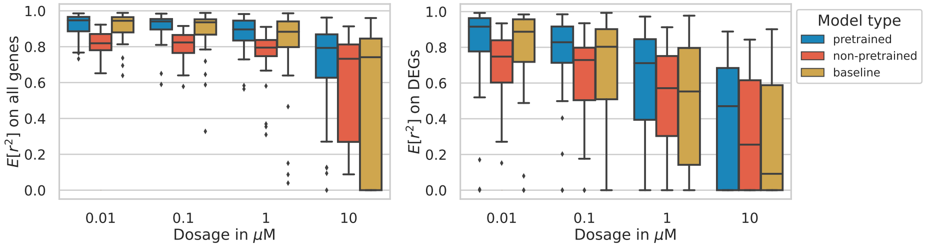


Figure 4: Performance of the pretrained and non-pretrained chemCPA model on the extended gene set using GROVER, see also Figure 2. Again, the pretrained model shows the best performance with the non-pretrained model failing to beat the baseline at lower dosages. At 10 μM on the DEGs, the pretrained model's median is more than 0.3 higher compared to the baseline.

We believe that the poor performance of the non-pretrained model can be attributed to two things. First, perturbations are subtle, and since we are predicting the whole phenotype, a typical reconstruction loss might not be sufficient to capture the drugs' effect. A loss that explicitly accounts for the DEGs might help in that regard. Second, the sci-Plex3 data is the first of its kind, and technological noise is still an issue. When evaluated over the whole training set, the baseline achieves r^2-scores higher than 65% for more than 96% of the observations. This sparsity holds the risk of overfitting for non-pretrained models, whereas the fine-tuned models are more robust to noise.

In addition, the drug-dose combinations for which the examined cells deviate from the controls' phenotypes correspond to less than 20% of the compounds in the training set. Accordingly, the effective number of drugs for which chemCPA can learn perturbations reduces to fewer than 30 drugs. This is another indication for the limitation of the sci-Plex3 dataset, preventing us to identify clear differences between the molecule encoders. Nevertheless, chemCPA was partly able to successfully predict the response when the baseline failed, see Figure 5 for two examples, and we believe that it will benefit from higher-quality scRNA-seq HTSs in the future.

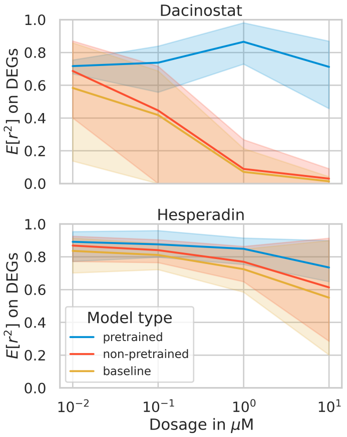


Figure 5: Perturbation prediction for Dacinostat and Hesperadin for chemCPA across all three cell-lines for the identical gene set using RDKit as molecule encoder.

6 Conclusion

In this paper, we introduced chemCPA, a model for predicting cellular phenotypes for unseen drug perturbations by encoding the drugs' molecular structures. Enabled by the flexible architecture of chemCPA, we evaluated different molecule encoding networks and found that readily available molecule features such as RDKit fingerprints yield competitive results. Applied to single-cell data, we demonstrated how pretraining on bulk HTSs improves chemCPA's generalisation performance. With new scRNA-seq HTSs, chemCPA can become a powerful aid in the drug screening and drug discovery process.

Acknowledgments

L.H. is thankful for valuable feedback from Fabiola Curion and Carlo De Donno. L.H. acknowledges support by the Helmholtz Association under the joint research school Munich School for Data Science - MUDS". This work was supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036B. F.J.T. acknowledges support by the BMBF (01IS18053A) and the Helmholtz Associations Initiative and Networking Fund through Helmholtz AI (ZT-I-PF-5-01) and sparse2big (ZT-I-007).

References

Matthew Amodio, Dennis Shung, Daniel B Burkhardt, Patrick Wong, Michael Simonov, Yu Yamamoto, David van Dijk, Francis Perry Wilson, Akiko Iwasaki, and Smita Krishnaswamy. Generating hard-to-obtain information from easy-to-obtain information: applications in drug discovery and clinical inference. *Patterns*, 2(7):100288, 2021.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7): 1853–1866, 2016.

Fabian Fröhlich, Thomas Kessler, Daniel Weindl, Alexey Shadrin, Leonard Schmiester, Hendrik Hache, Artur Muradyan, Moritz Schütte, Ji-Hyun Lim, Matthias Heinig, et al. Efficient parameter estimation enables the prediction of drug response using a mechanistic pan-cancer pathway model. *Cell systems*, 7(6):567–579, 2018.

Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, et al. A python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, pp. 1–4, 2022.

Jase Gehring, Jong Hwee Park, Sisi Chen, Matthew Thomson, and Lior Pachter. Highly multiplexed single-cell rna-seq for defining cell population and transcriptional spaces. *BioRxiv*, pp. 315333, 2018.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Xiaoping Han, Ziming Zhou, Lijiang Fei, Huiyu Sun, Renying Wang, Yao Chen, Haide Chen, Jingjing Wang, Huanna Tang, Wenhao Ge, et al. Construction of a human cell landscape at single-cell level. *Nature*, 581(7808):303–309, 2020.

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

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Kenji Kamimoto, Christy M Hoffmann, and Samantha A Morris. Celloracle: Dissecting cell identity via network inference and in silico gene perturbation. *bioRxiv*, 2020.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8): 595–608, 2016.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. *Advances in neural information processing systems*, 30, 2017.

Greg Landrum. Rdkit: Open-source cheminformatics software. 2016. URL `https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4`.

Mufei Li, Jinjing Zhou, Jiajing Hu, Wenxuan Fan, Yangkang Zhang, Yaxin Gu, and George Karypis. Dgl-lifesci: An open-source toolkit for deep learning on graphs in life science. *ACS Omega*, 2021.

Romain Lopez, Adam Gayoso, and Nir Yosef. Enhancing scientific discoveries in molecular biology with deep generative models. *Molecular Systems Biology*, 16(9):e9198, 2020.

Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 16(8):715–721, 2019.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Yuge Ji, Ignacio L Ibarra, F Alexander Wolf, Nafissa Yakubova, Fabian J Theis, and David Lopez-Paz. Learning interpretable cellular responses to complex perturbations in high-throughput screens. *bioRxiv*, 2021.

Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, et al. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology*, 40(1):121–130, 2022.

Christopher S McGinnis, David M Patterson, Juliane Winkler, Daniel N Conrad, Marco Y Hein, Vasudha Srivastava, Jennifer L Hu, Lyndsay M Murrow, Jonathan S Weissman, Zena Werb, et al. Multi-seq: sample multiplexing for single-cell rna sequencing using lipid-tagged indices. *Nature methods*, 16(7):619–626, 2019.

Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

Thai-Hoang Pham, Yue Qiu, Jucheng Zeng, Lei Xie, and Ping Zhang. A deep learning framework for high-throughput mechanism-driven phenotype compound screening and its application to covid-19 drug repurposing. *Nature machine intelligence*, 3(3):247–257, 2021.

Ladislav Rampášek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr. vae: improving drug response prediction via modeling of drug perturbation effects. *Bioinformatics*, 35(19):3743–3751, 2019.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

Sergei Rybakov, Mohammad Lotfollahi, Fabian J Theis, and F Alexander Wolf. Learning interpretable latent autoencoder representations with annotations of feature sets. In *Machine Learning in Computational Biology (MLCB) meeting*. Cold Spring Harbor Laboratory, 2020.

Lucas Seninge, Ioannis Anastopoulos, Hongxu Ding, and Joshua Stuart. Vega is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nature communications*, 12(1):1–9, 2021.

Lisa Sikkema, Daniel C Strobl, Luke Zappia, Elo Madissoon, Nikolay S Markov, Laure-Emmanuelle Zaragosi, Meshal Ansari, Marie-Jeanne Arguel, Leonie Apperloo, Christophe Becavin, Marijn Berg, Evgeny Chichelnitskiy, Mei-i Chung, Antoine Collin, Aurore CA Gay, Baharak Hooshiar Kashani, Manu Jain, Theodore Kapellos, Tessa M Kole, Christoph H Mayr, Michael Papen, von, Lance Peter, Ciro Ramirez-Suastegui, Janine Schniering, Chase J Taylor, Thomas Walzthoeni, Chuan Xu, Linh T Bui, Carlo Donno, de, Leander Dony, Minzhe Guo, Austin J Gutierrez, Lukas Heumos, Ni Huang, Ignacio L Ibarra, Nathan D Jackson, Preetish Kadur Lakshminarasimha Murthy, Mohammad Lotfollahi, Tracy Tabib, Carlos Talavera-Lopez, Kyle J Travaglini, Anna Wilbrey-Clark, Kaylee B Worlock, Masahiro Yoshida, Tushar J Desai, Oliver Eickelberg, Christine Falk, Naftali Kaminski, Mark A Krasnow, Robert Lafyatis, Marko Z Nikoli, Joseph E Powell, Jayaraj Rajagopal, Orit Rozenblatt-Rosen, Max A Seibold, Dean Sheppard, Douglas P Shepherd, Sarah A Teichmann, Alexander M Tsankov, Jeffrey Whitsett, Yan Xu, Nicholas E Banovich, Pascal Barbry, Thu E Duong, Kerstin B Meyer, Jonathan A Kropski, Dana Pe'er, Herbert B Schiller, Purushothama Rao Tata, Joachim L Schultze, Alexander V Misharin, Martijn C Nawijn, Malte D Luecken, and Fabian J Theis. An integrated cell atlas of the human lung in health and disease. *bioRxiv*, 2022. doi: 10.1101/2022.03.10.483747. URL `https://www.biorxiv.org/content/early/2022/03/11/2022.03.10.483747`.

Sanjay R Srivatsan, José L McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A Pliner, Dana L Jackson, Riza M Daza, Lena Christiansen, et al. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473):45–51, 2020.

Marlon Stoeckius, Shiwei Zheng, Brian Houck-Loomis, Stephanie Hao, Bertrand Z Yeung, William M Mauck, Peter Smibert, and Rahul Satija. Cell hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. *Genome biology*, 19(1):1–12, 2018.

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.

Ramzan Umarov, Yu Li, and Erik Arner. Deepcellstate: An autoencoder-based framework for predicting cell type specific transcriptional states induced by drug treatment. *PLoS Computational Biology*, 17(10):e1009465, 2021.

Ido Yofe, Rony Dahan, and Ido Amit. Single-cell genomic approaches for developing the next generation of immunotherapies. *Nature medicine*, 26(2):171–177, 2020.

Bo Yuan, Ciyue Shen, Augustin Luna, Anil Korkut, Debora S Marks, John Ingraham, and Chris Sander. Cellbox: interpretable machine learning for perturbation biology with application to the design of cancer combination therapy. *Cell systems*, 12(2):128–140, 2021.

Jie Zhu, Jingxiang Wang, Xin Wang, Mingjing Gao, Bingbing Guo, Miaomiao Gao, Jiarui Liu, Yanqiu Yu, Liang Wang, Weikaixin Kong, et al. Prediction of drug efficacy from transcriptional profiles with deep learning. *Nature biotechnology*, 39(11):1444–1452, 2021.

A Appendix

A.1 Attribute embedding

Table 4: Details on pretrained models for the molecule encoder G.

Molecule encoder G	Embedding dim h_{drug}	Pretrained
RDKit	200	–
GROVER	3400	authors
JT-VAE	56	ZINC, L1000, sci-Plex3
GCN	128	PCBA
MPNN	128	PCBA
weave	128	PCBA

A.2 Counterfactual prediction

1. To compute counterfactual predictions, we obtain basal states z_i for all control observations present in the test set. For each combination of drug, dose, and cell line in the test set, we compute the latent attribute state $z_{\text{attribute}}$ and combine it with all z_i. Subsequently, we compute the mean per gene across all predictions and likewise for the real measurements. As a result, we obtain two n dimensional vectors, where n is the number of genes (977 or 2000), for which we compute the r^2 score. Taken together, we get one score per combination.

A.3 Additional information on the L1000 experiment

1. For infos on the RDKit sweep and resulting best run, see Table 5 and Table 6.
2. Architectures for best configuration of the perturbation networks P_φ and adversary classifiers are presented in Table 7.
3. For details on the performance of the best runs, see Table 8.

Table 5: Fixed Parameters for the RDKit sweep in the L1000 dataset.

Parameter	Value
num_epochs	1500
dataset_type	lincs
decoder_activation	linear
model	rdkit

A.4 Additional information on the sci-Plex3 experiments

1. The optimisation was performed similarly to the presented sweeps in Table 6 and Table 7 for the perturbation network and adversary parameters for 10 samples each per category.
2. Boxplot results for RDKit, see Figures 6 and 8, and JT-VAE, see Figures 7 and 9.
3. Paired t-tests were performed for both settings, see Table 10 for the shared gene set and Table 10 for the extended gene set.
4. More examples on the performance with respect to specific drugs are presented in Figure 10, Figure 11, Figure 12, and Figure 13.

Table 6: Random parameters for the RDKit sweep in the L1000 dataset.

Parameter	Type	Values	Best config
samples	fixed	25	NaN
dim	choice	{64, 32}	32
dosers_width	choice	{64, 256, 128, 512}	64
dosers_depth	choice	{1, 2, 3}	1
dosers_lr	loguniform	$[1 \times 10^{-4}, 1 \times 10^{-2}]$	5.61×10^{-4}
dosers_wd	loguniform	$[1 \times 10^{-8}, 1 \times 10^{-5}]$	1.33×10^{-7}
autoencoder_width	choice	{128, 256, 512}	256
autoencoder_depth	choice	{3, 4, 5}	4
autoencoder_lr	loguniform	$[1 \times 10^{-4}, 1 \times 10^{-2}]$	1.12×10^{-3}
autoencoder_wd	loguniform	$[1 \times 10^{-8}, 1 \times 10^{-5}]$	3.75×10^{-7}
adversary_width	choice	{64, 256, 128}	128
adversary_depth	choice	{2, 3, 4}	3
adversary_lr	loguniform	$[5 \times 10^{-5}, 1 \times 10^{-2}]$	8.06×10^{-4}
adversary_wd	loguniform	$[1 \times 10^{-8}, 1 \times 10^{-3}]$	4.0×10^{-6}
adversary_steps	choice	{2, 3}	2
reg_adversary	loguniform	[5, 100]	24.1
penalty_adversary	loguniform	[1, 10]	3.35
batch_size	choice	{32, 64, 128}	128
step_size_lr	choice	{200, 50, 100}	100
embedding_encoder_width	choice	{128, 256, 512}	128
embedding_encoder_depth	choice	{2, 3, 4}	3

Table 7: Presented are the best configurations per molecule encoder from 18 random hyperparamter samples similar to the one presented in Table 6.

Parameter	GROVER	MPNN	RDKit
dosers_width	512	64	64
dosers_depth	2	2	3
dosers_lr	5.61×10^{-4}	1.58×10^{-3}	1.12×10^{-3}
dosers_wd	1.33×10^{-7}	6.25×10^{-7}	3.75×10^{-7}
embedding_encoder_width	512	128	128
embedding_encoder_depth	3	4	4

Parameter	weave	JT-VAE	GCN
dosers_width	512	64	512
dosers_depth	2	2	2
dosers_lr	1.12×10^{-3}	2.05×10^{-4}	2.05×10^{-4}
dosers_wd	2.94×10^{-8}	2.94×10^{-8}	1.33×10^{-6}
embedding_encoder_width	128	256	128
embedding_encoder_depth	3	4	3

Table 8: Performance of the best runs on L1000 for different molecule encoders G

Model G	Drug	Cell line	Mean r^2 all	Mean r^2 DEGs	Mean r^2 DEGs [val]
GCN	0.11	0.16	0.92	0.84	0.83
MPNN	0.07	0.24	0.94	0.87	0.84
GROVER	0.07	0.16	0.94	0.88	0.86
JT-VAE	0.06	0.15	0.94	0.88	0.85
RDKit	0.08	0.15	0.93	0.86	0.85
weave	0.09	0.20	0.91	0.74	0.72

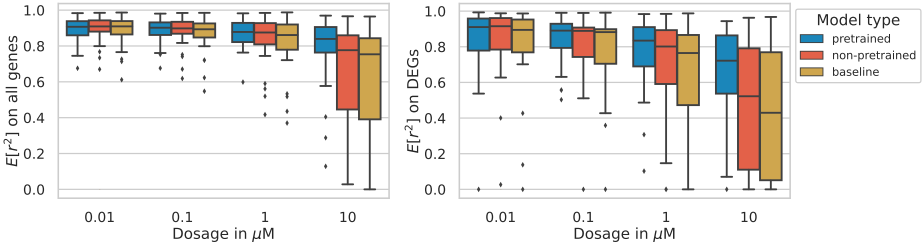


Figure 6: Performance of the pretrained and non-pretrained chemCPA model using RDKit. Comparisons against the baseline are done on both the complete gene set (977 genes) and the compound specific DEGs (50 genes).

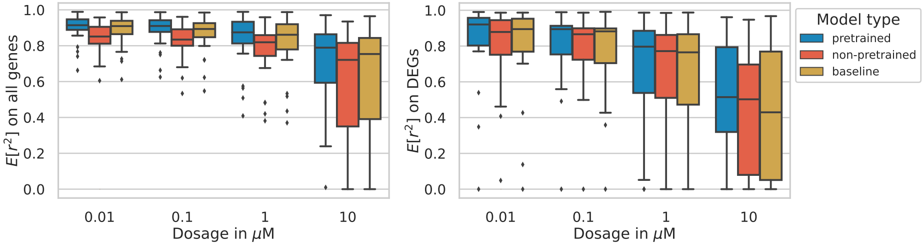


Figure 7: Performance of the pretrained and non-pretrained chemCPA model using JT-VAE. Comparisons against the baseline are done on both the complete gene set (977 genes) and the compound specific DEGs (50 genes).

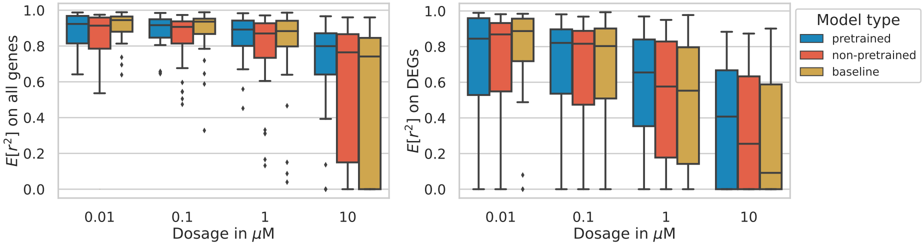


Figure 8: Performance of the pretrained and non-pretrained chemCPA model on the extended gene set using RDKit, see also Figure 6.

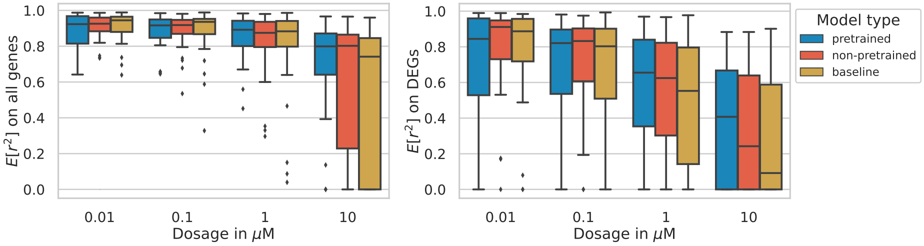


Figure 9: Performance of the pretrained and non-pretrained chemCPA model on the extended gene set using JT-VAE, see also Figure 7.

Table 9: Significance test for a dosage of $10\,\mu$M on the shared gene set using the paired t-test.

Model G	Against	Gene set	p-value
rdkit	baseline	all genes	0.0014
rdkit	baseline	DEGs	0.0007
rdkit	non-pretrained	all genes	0.0058
rdkit	non-pretrained	DEGs	0.0051
grover	baseline	all genes	0.0009
grover	baseline	DEGs	0.0003
grover	non-pretrained	all genes	0.0025
grover	non-pretrained	DEGs	0.0067
jtvae	baseline	all genes	0.0047
jtvae	baseline	DEGs	0.0021
jtvae	non-pretrained	all genes	0.0001
jtvae	non-pretrained	DEGs	0.0086

Table 10: Significance test for a dosage of $10\,\mu$M on the extended gene set using the paired t-test.

Model G	Against	Gene set	p-value
rdkit	baseline	all genes	0.0030
rdkit	baseline	DEGs	0.0009
rdkit	non-pretrained	all genes	0.0137
rdkit	non-pretrained	DEGs	0.0370
grover	baseline	all genes	0.0029
grover	baseline	DEGs	0.0048
grover	non-pretrained	all genes	0.0137
grover	non-pretrained	DEGs	0.0434
jtvae	baseline	all genes	0.0030
jtvae	baseline	DEGs	0.0009
jtvae	non-pretrained	all genes	0.0324
jtvae	non-pretrained	DEGs	0.0230

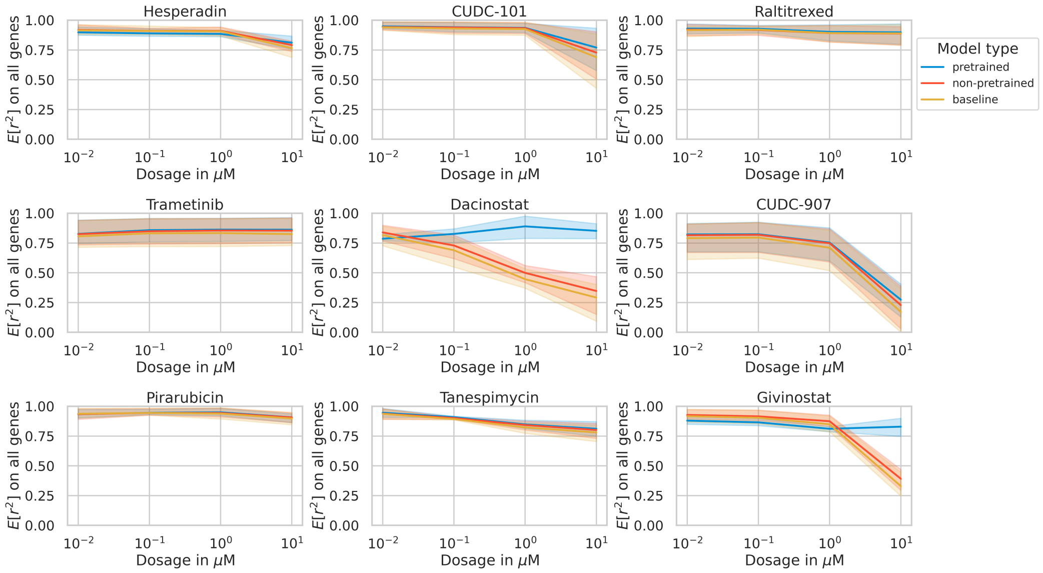


Figure 10: Drug-wise comparison between the baseline, pretrained and non-pretrained models using RDKit for all nine drugs in the test set considering all genes for the shared gene set.

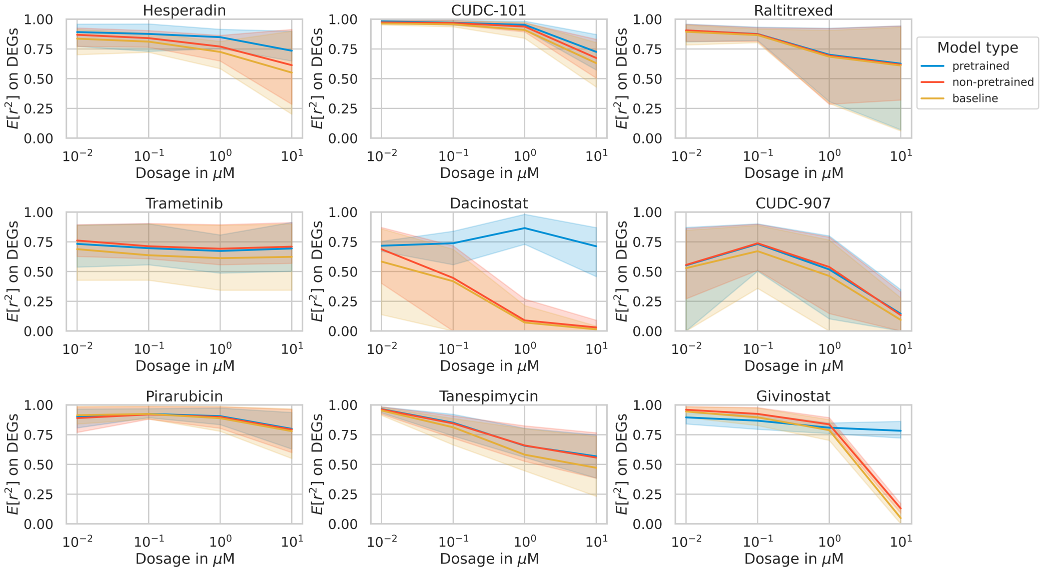


Figure 11: Drug-wise comparison between the baseline, pretrained and non-pretrained models using RDKit for all nine drugs in the test set considering the DEGs for the shared gene set.

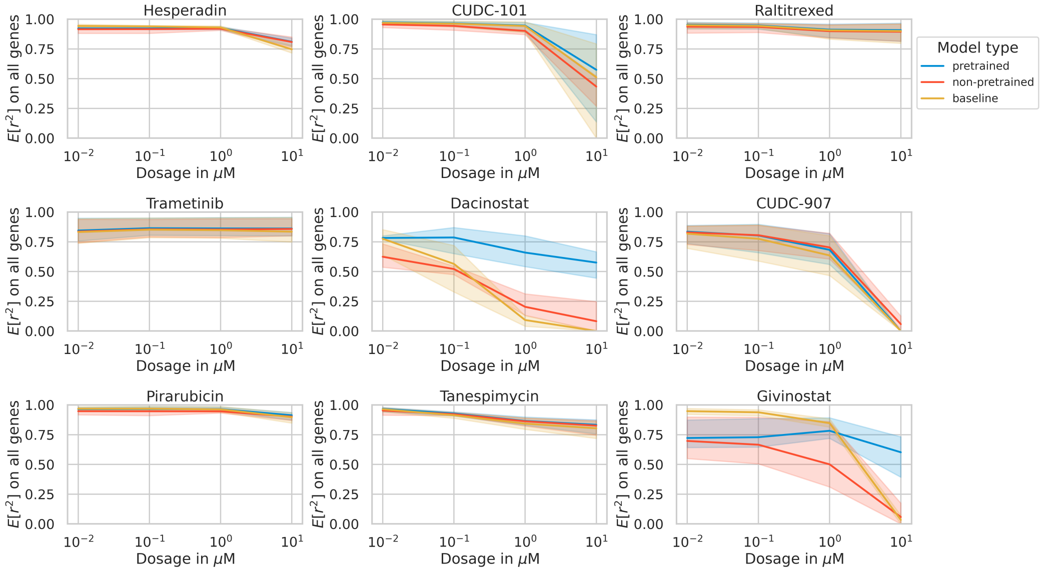


Figure 12: Drug-wise comparison between the baseline, pretrained and non-pretrained models using RDKit for all nine drugs in the test set considering all genes for the extended gene set.

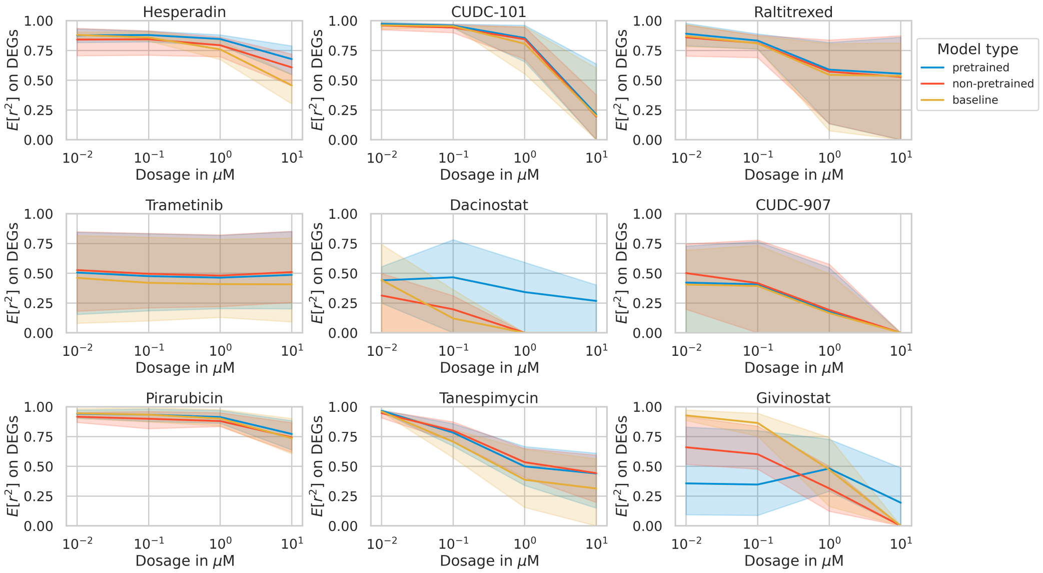


Figure 13: Drug-wise comparison between the baseline, pretrained and non-pretrained models using RDKit for all nine drugs in the test set considering the DEGs for the extended gene set.