Evaluation of deep learning approaches for drug combination synergy prediction in the context of cancer cell lines based on TDC benchmark

Ketan Sarode[1] **Ranajeet Saund**[1] **Amit Agarwal**[1] **Anand Muglikar**[2]
[1]Innoplexus Consulting Services Pvt Ltd, a Partex company
[2]Perpetualblock Technologies Private Limited, a Partex company
ketan.sarode@ics.innoplexus.com

1 Introduction

Cancer in its multiple heterogeneous forms is one of the leading causes for mortality. One of the major challenges in treatment of cancer indications is development of resistance to the available therapies. Development of new therapies is a very challenging and time consuming process. In this scenario, use of drug combination therapy provides multiple advantages. Drug combination therapy can potentially provide advantages in the form of reduced adverse effects/ toxicity (if the individual drugs can be used with reduced dose), overcoming drug resistance and providing synergistic therapeutic effect unattainable with use of the individual drugs. While certain drug combinations can provide synergistic effects, many of them can lead to harmful adverse effects resulting from different drug interactions. Experimental identification of best synergistic combinations is not practically cost effective due to challenges in exploring the huge combinatorial space. Therefore it is crucial to develop computational methods for identification of synergistic drug combinations.

For cancer indications, availability of different cancer cell lines allows experimental measurement of therapeutic effect of individual drugs and combinations of drugs. Such data has been used to quantitate synergistic effects of drug combinations in the form scores like CSS, Bliss, HSA, Loewe, ZIP in portals like DrugComb [1]. Therapeutics Data Commons (TDC) platform provides summarized results of the drug combinations screening experiments in a standardized format ready for use in the AI predictive model building [2, 3]. Specifically the dataset provides data for 128 drug molecules tested on 59 cell lines resulting in a total of 297098 unique drug combination-cell line pairs. For the 297098 drug combinations, the combination synergy score for respective cell lines is provided in the form of five endpoints viz. CSS, Bliss, HSA, Loewe, ZIP [4]. Drug Combination Benchmark Group from Therapeutics Data Commons provides a standardized evaluation framework for testing predictive models built for regression tasks for the five endpoints.

2 Data description and the problem statement

Each sample in the Drug Combination Benchmark Group tasks, has three independent entities viz. Drug 1, Drug 2 and a cancer cell line. For both drugs, their names and their representation in the SMILES string format is provided. For the cancer cell lines, features obtained from NCI's CellMiner interface are provided. For each cell line, 23,808 gene features capturing transcript expression levels averaged from five microarray platforms, 627 microRNA expression features and 3171 proteomic features that capture the abundance levels of a subset of proteins totalling 27606 features are provided [5].

The Drug1-Drug2-Cell_line triplets have following five target endpoints:

1. Drug Combination Sensitivity Score TDC.DrugComb_CSS: It is derived using the relative IC50 values of compounds and the areas under dose-response curves.

Preprint.

2. The Bliss model endpoint TDC.DrugComb_Bliss.
3. The highest single agent endpoint TDC.DrugComb_HSA.
4. The Loewe additivity model endpoint TDC.DrugComb_Loewe.
5. The zero interaction potency endpoint TDC.DrugComb_ZIP

All five target endpoints take real values and therefore the predictive tasks for these endpoints are modeled as regression tasks.

3 Methodology

The tasks in drug combination benchmark need to model the effect of the two drug entities in context of specific cell lines. In our experiments, specific cell lines are represented in the predictive models in the form of 27606 dimensional feature vectors provided with the dataset. For the drug molecules, we explore two different methods of representation. The first method is based on the Conditional Graph Fusion (CongFu) approach for drug synergy prediction which models the drug molecules as graphs [6]. Whereas the second method uses a pre-trained molecular language model that takes SMILES string representations as input and provides dense embeddings for the drug molecules.

3.1 Approach 1: CongFu layers based model

The CongFu approach described in Tsepa, Oleksii, et al. 2023 [6], models the drug molecules as graphs with nodes representing atoms in the molecule and the edges representing bonds between the atoms. The initial node (i.e. atom) and edge (i.e. bond) features are assigned on their identity e.g. based on type of atom/ bond. These initial node/edge embeddings evolve through the GNN (Graph Neural Network) layers. CongFu layer introduces a context propagation approach to fuse context of cell line features with GNN node and edge embeddings. The fused context information is propagated in the graph with the graph update steps. In the end, the bottleneck step in the CongFu layer merges atom level information into a global context. The output of the CongFu layer is the updated dense representations of the two drugs and the cell line context. The original implementation in the Tsepa, Oleksii, et al. 2023 [6] paper models the synergy prediction as binary classification problem. In the current work, we modify the model architecture for the regression task. Also, to make the model architecture explicitly invariant towards order of Drug1, Drug2 representations in the Drug1-Drug2–Context concatenated representation, we take the addition vector of drug embeddings as the representation of the combination.

3.2 Approach 2: Pre-trained molecular language model for drug features

Recently transformer architecture based language models pre-trained on large corpus of unlabeled data have achieved big performance improvements in downstream natural language processing tasks. This led to efforts in building molecular representation models like ChemBERTa25 trained on SMILES string representation of chemical compounds. MoLFomer-XL is one such recently developed pre-trained molecular language model [7, 8]. It is a transformer encoder model which uses rotary positional embeddings and a linear attention mechanism. It is trained on unlabeled molecules data available through the PubChem and ZINC datasets. We use a publicly available version of MoLFomer-XL trained on 10% ZINC + 10% PubChem data [9]. For the drug combination synergy prediction task, we initialize embeddings of the two drugs using MoLFomer-XL. These drug embeddings and the cell line features are evolved through feed forward neural network layers. The evolved representations of the two drugs are combined using addition operation and are concatenated with the evolved cell lines representation. This concatenated vector is then used to predict target synergy endpoints.

4 Results

For each target synergy endpoint dataset, five models were trained with five different splits of train and validation data set following guidelines of TDC Drug combination benchmark group. Specifically the data splits were generated for the `drugcombo_group` benchmark using the `get_train_valid_split` function provided through python `tdc` package. For all the regression

tasks, the performance metric monitored was Mean Absolute Error (MAE). For each task, the final performance metric generated is the mean MAE across five different train/val splits and the corresponding standard deviation.

The results of model training experiments using the two different approaches as discussed in the methodology section are presented below.

4.1 Approach 1: CongFu layers based model

We trained the CongFu layers based model with the hyperparameter choices provided in Table 1.

Table 1: Hyperparamters choices for CongFu layers based model

Hyperparameter	# Value
Total number of layers	5
GINEconv based feature update layers	3
CongFu layers	2
GNN feature dimension	300
Updated context dimension	512

The model has 17.5 million parameters. Further details of the model parameters are provided in the Appendix A. The peformance metrics for the models trained using this model architecture are presented in Table 2. The table shows the MAE (mean±std.dev., calculated across five different train/validation data splits) obtained for each drug combination synergy endpoint.

Table 2: Results of CongFu layers based model for the tasks in TDC drug combination benchmark

Task	MAE
drugcomb_css	11.786 ± 0.369
drugcomb_css_kidney	10.695 ± 0.508
drugcomb_css_lung	10.637 ± 0.348
drugcomb_css_breast	9.288 ± 0.523
drugcomb_css_hematopoietic_lymphoid	21.04 ± 0.397
drugcomb_css_colon	12.575 ± 0.373
drugcomb_css_prostate	10.403 ± 0.62
drugcomb_css_ovary	10.633 ± 0.536
drugcomb_css_skin	10.761 ± 0.61
drugcomb_css_brain	10.211 ± 0.385
drugcomb_hsa	4.422 ± 0.051
drugcomb_loewe	8.437 ± 0.125
drugcomb_bliss	4.533 ± 0.036
drugcomb_zip	3.799 ± 0.038

4.2 Approach 2: MoLFormer-XL based model

We trained the MoLFormer-XL based model with hyperparameter choices provided in Table 3.

The model has 17.5 million parameters. Further details of the model parameters are provided in the Appendix B. The peformance metrics for the models trained using this model architecture are presented in Table 4. The table shows the MAE (mean±std.dev., calculated across five different train/validation data splits) obtained for each drug combination synergy endpoint.

Table 3: Hyperparamters choices for MoLFormer based model

Hyperparameter	# Value
Drug embedding dimension	768
Drug/Context FFN, hidden layer 1 dimension	600
Drug/Context FFN, hidden layer 2 dimension	300
Drug+Context evolved feature dimension	300

Table 4: Results of MoLFormer based model for the tasks in TDC drug combination benchmark

Task	MAE
drugcomb_css	11.79 ± 0.079
drugcomb_css_kidney	10.859 ± 0.193
drugcomb_css_lung	10.991 ± 0.105
drugcomb_css_breast	9.691 ± 0.039
drugcomb_css_hematopoietic_lymphoid	17.896 ± 0.346
drugcomb_css_colon	12.592 ± 0.079
drugcomb_css_prostate	10.941 ± 0.078
drugcomb_css_ovary	11.211 ± 0.103
drugcomb_css_skin	11.204 ± 0.136
drugcomb_css_brain	10.771 ± 0.087
drugcomb_hsa	4.108 ± 0.013
drugcomb_loewe	7.94 ± 0.023
drugcomb_bliss	4.256 ± 0.01
drugcomb_zip	3.673 ± 0.019

4.3 Hardware configuration

Machine with Intel i9-13900 processor 62GB RAM and NVIDIA RTX A6000 GPU with 48GB VRAM was used for the model training experiments.

5 Discussion

The two approaches evaluated in this work demonstrate the utility of two different representation techniques for the drug molecules. Both these approaches fall into the category of deep learning techniques which avoid tedious steps of feature selection and feature engineering. The GNN architecture provides the advantage of explicitly representing topological characteristics of the molecules which are only implicitly represented in the string-based representations. This makes them a good candidate for molecular property prediction tasks. In a GNN based approach, the model learns the molecular representations from scratch based on the 128 molecules in the dataset. Whereas the MoLFormer-XL based approach benefits from the representation learnt on the basis of pre-training on 100M molecules [10]. This highlights strengths of pretrained foundation models in addressing downstream tasks. The graph based foundation models are an emerging topic and it will be exciting to see how their performance compares with transformer based foundation models. In future work, we plan to explore more suitable approaches for combining multi-entity representations.

References

[1] Bulat Zagidullin, Jehad Aldahdooh, Shuyu Zheng, Wenyu Wang, Yinyin Wang, Joseph Saad, Alina Malyutina, Mohieddin Jafari, Ziaurrehman Tanoli, Alberto Pessia, et al. Drugcomb: an integrative cancer drug combination data portal. *Nucleic acids research*, 47(W1):W43–W51, 2019.

[2] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*, 2021.

[3] Therapeutics data commons - tdc. https://tdcommons.ai/. Accessed: 2024-07-18.

[4] Drug combination benchmark group - tdc. https://tdcommons.ai/benchmark/drugcombo_group/overview/. Accessed: 2024-07-18.

[5] Drug synergy prediction - tdc. https://tdcommons.ai/multi_pred_tasks/drugsyn/. Accessed: 2024-07-18.

[6] Oleksii Tsepa, Bohdan Naida, Anna Goldenberg, and Bo Wang. Congfu: Conditional graph fusion for drug synergy prediction. *arXiv preprint arXiv:2305.14517*, 2023.

[7] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties, 2021.

[8] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022.

[9] ibm/molformer-xl-both-10pct. https://huggingface.co/ibm/MoLFormer-XL-both-10pct. Accessed: 2024-07-18.

[10] Ibm/molformer: Repository for molformer. https://github.com/IBM/molformer. Accessed: 2024-07-18.

A Approach 1 model parameters

Modules	Parameters
model.x_embedding1.weight	35700
model.x_embedding2.weight	900
model.basic_layers.0.graph_update.gnn.mlp.0.weight	180000
model.basic_layers.0.graph_update.gnn.mlp.0.bias	600
model.basic_layers.0.graph_update.gnn.mlp.3.weight	180000
model.basic_layers.0.graph_update.gnn.mlp.3.bias	300
model.basic_layers.0.graph_update.gnn.edge_embedding1.weight	1500
model.basic_layers.0.graph_update.gnn.edge_embedding2.weight	900
model.basic_layers.0.graph_update.batch_norm.weight	300
model.basic_layers.0.graph_update.batch_norm.bias	300
model.basic_layers.1.graph_update.gnn.mlp.0.weight	180000
model.basic_layers.1.graph_update.gnn.mlp.0.bias	600
model.basic_layers.1.graph_update.gnn.mlp.3.weight	180000
model.basic_layers.1.graph_update.gnn.mlp.3.bias	300
model.basic_layers.1.graph_update.gnn.edge_embedding1.weight	1500
model.basic_layers.1.graph_update.gnn.edge_embedding2.weight	900
model.basic_layers.1.graph_update.batch_norm.weight	300
model.basic_layers.1.graph_update.batch_norm.bias	300
model.basic_layers.2.graph_update.gnn.mlp.0.weight	180000
model.basic_layers.2.graph_update.gnn.mlp.0.bias	600
model.basic_layers.2.graph_update.gnn.mlp.3.weight	180000
model.basic_layers.2.graph_update.gnn.mlp.3.bias	300
model.basic_layers.2.graph_update.gnn.edge_embedding1.weight	1500
model.basic_layers.2.graph_update.gnn.edge_embedding2.weight	900
model.basic_layers.2.graph_update.batch_norm.weight	300
model.basic_layers.2.graph_update.batch_norm.bias	300
model.congfu_layers.0.context_propagation.context_linear.weight	90000
model.congfu_layers.0.context_propagation.context_linear.bias	300
model.congfu_layers.0.context_propagation.x_linear.weight	90000
model.congfu_layers.0.graph_update.gnn.mlp.0.weight	180000
model.congfu_layers.0.graph_update.gnn.mlp.0.bias	600
model.congfu_layers.0.graph_update.gnn.mlp.3.weight	180000
model.congfu_layers.0.graph_update.gnn.mlp.3.bias	300
model.congfu_layers.0.graph_update.gnn.edge_embedding1.weight	1500
model.congfu_layers.0.graph_update.gnn.edge_embedding2.weight	900

model.congfu_layers.0.graph_update.batch_norm.weight	300
model.congfu_layers.0.graph_update.batch_norm.bias	300
model.congfu_layers.0.bottleneck.gnn.att_src	300
model.congfu_layers.0.bottleneck.gnn.att_dst	300
model.congfu_layers.0.bottleneck.gnn.bias	300
model.congfu_layers.0.bottleneck.gnn.lin_src.weight	90000
model.congfu_layers.0.bottleneck.gnn.lin_dst.weight	90000
model.congfu_layers.1.context_propagation.context_linear.weight	90000
model.congfu_layers.1.context_propagation.context_linear.bias	300
model.congfu_layers.1.context_propagation.x_linear.weight	90000
model.congfu_layers.1.graph_update.gnn.mlp.0.weight	180000
model.congfu_layers.1.graph_update.gnn.mlp.0.bias	600
model.congfu_layers.1.graph_update.gnn.mlp.3.weight	180000
model.congfu_layers.1.graph_update.gnn.mlp.3.bias	300
model.congfu_layers.1.graph_update.gnn.edge_embedding1.weight	1500
model.congfu_layers.1.graph_update.gnn.edge_embedding2.weight	900
model.congfu_layers.1.graph_update.batch_norm.weight	300
model.congfu_layers.1.graph_update.batch_norm.bias	300
model.congfu_layers.1.bottleneck.gnn.att_src	300
model.congfu_layers.1.bottleneck.gnn.att_dst	300
model.congfu_layers.1.bottleneck.gnn.bias	300
model.congfu_layers.1.bottleneck.gnn.lin_src.weight	90000
model.congfu_layers.1.bottleneck.gnn.lin_dst.weight	90000
model.context_encoder.0.fc.weight	14134272
model.context_encoder.0.fc.bias	512
model.context_encoder.1.weight	153600
model.context_encoder.1.bias	300
model.output_transformation.0.fc.weight	153600
model.output_transformation.0.fc.bias	512
model.output_transformation.1.weight	131072
model.output_transformation.1.bias	256
model.reshape_drug1drug2.weight	131072
model.reshape_drug1drug2.bias	512
model.mlp.0.fc.weight	207872
model.mlp.0.fc.bias	256
model.mlp.1.fc.weight	32768
model.mlp.1.fc.bias	128
model.mlp.2.fc.weight	8192
model.mlp.2.fc.bias	64
model.mlp.3.weight	64
model.mlp.3.bias	1

Total Trainable Params: 17533553

B Approach 2 model parameters

Modules	Parameters
model.lin1.weight	460800
model.lin1.bias	600
model.lin2.weight	180000
model.lin2.bias	300
model.lin3.weight	16563600
model.lin3.bias	600
model.lin4.weight	180000
model.lin4.bias	300
model.lin5.weight	180000

```
|  model.lin5.bias  |    300     |
| model.lin6.weight |    300     |
|  model.lin6.bias  |     1      |
+-------------------+------------+
Total Trainable Params: 17566801
```