# SimGCN for TDC Benchmarks

Suman K. Bera[1], Jason Dent[1], Gurbinder Gill[1], Andrew Stolman[1], Bo Wu[12]

[1] Katana Graph Inc

[2] Colorado School of Mines

February 8, 2022

**Abstract**

In this report, we present a graph neural network based solution, *SimGCN*, for Therapeutics Data Commons (TDC) ADMET benchmark group challenge. The goal of the challenge is to predict various kinds of **A**bsorption, **D**istribution, **M**etabolism, **E**xcretion, and **T**oxicity properties of small molecule drugs, given its structural representation through SMILES strings. Our approach leverages domain specific knowledge to construct a similarity graph of the molecules. We then pose this challenge as a node property prediction task and train a graph neural network on the similarity graph to perform the prediction. We show that our approach leads to top submission in multiple challenges.

## 1 Introduction

Therapeutics Data Commons (TDC) has introduced an ecosystem for AI/ML tasks with the goal of speedy development of effective drugs [4]. As part of this initiative, TDC has launched a suite of challenges for comparing the efficacy of AI/ML methods. One of the challenge deals with predicting various properties of small molecule drugs. For a small molecule drug to be effective, it is required to have appropriate levels of absorption, distribution, metabolism, excretion, and toxicity (ADMET) properties. The ADMET challenge aims at predicting these properties of a set of small molecule drugs given their structural information in the form of SMILES strings.

In the ADMET challenge, there are 22 datasets. For each dataset, TDC uses scaffold split to partition the dataset into training and test sets. The scaffold split simulates the real-world drug discovery scenario where a trained model is applied to a new set of unseen drugs that are structurally distant from the existing ones. Some of the datasets require binary classification tasks and the other require regressions. The choice of metric is AUROC and AUPRC for the binary classification tasks, and Mean Absolute Error (MAE) and Spearman's correlation coefficient for regression tasks.

We present a graph neural networks based solution *SimGCN* for the ADMET challenge. Our approach essentially constructs a graph topology based on domain-knowledge and trains a graph convolution network (GCN) for supervised node classification/regression tasks. At the time of the writing, our submission leads the competitions in 8 categories.

## 2 Methodology

In this section, we discuss our proposed solution SimGCN for solving the small molecule drug property prediction task. SimGCN consists of two main stages. In the first stage, we construct a *threshold similarity graph* among the drug molecules and add RDkit fingerprint based features to the molecules. This similarity graph infuses domain specific knowledge into the prediction pipeline. The second stage consists of training a GNN model on the similarity graph to perform a node property prediction task. We discuss these stages in details below. We denote the entire set of molecules in a dataset by $X$.

**Threshold Similarity Graph Construction.** In the threshold similarity graph, denoted as $G_{\text{sim}}$, we add each drug molecule in $X$ as a node and we connect two molecules by an undirected edge if the *tanimoto similarity* [3] between the *topological fingerprints* [2] of the molecules is above a specified threshold. The threshold serves as a hyper parameter in our model. We treat $G_{\text{sim}}$ as an undirected and unweighted graph. We do not use the value of the tanimoto similarity score in training our models. To construct molecular features, we use RDKit 2D fingerprints. In particular, we use DescriptaStorus [1] to generate RDKit 2d normalized features for each molecule.

**Node Property Prediction Task.** We pose the drug property prediction task as a node property prediction task on $G_{\text{sim}}$. We mask the property labels/values for the test nodes and do not use it during training or hyper-parameter search. For binary classification, we use binary cross entropy loss function. For regression tasks, we use Mean Squared Error for spearman coefficient and L1 loss for MAE.

## 3 Experimental Results

We present our results on the TDC Admet Benchamark datasets in Table 1. Overall, SimGCN has 8 top entries and 2 entries with second position. Currently, there are eight methods in the leaderboard. Five of which are state-of-the-art graph learning based methods and the remianing three are non-graph based methods. Our codes are available at https://github.com/KatanaGraph/SimGCN-TDC.

Table 1: **Comparison with the Leaderboard on the TDC ADMET Benchmark Group.** As required by TDC, average and standard deviation across five runs are reported. Arrows (↑, ↓) indicate the direction of better performance. The best method is highlighted and the number of parameters are reported.

| TDC ADMET Challenge | | Current Top Entry | | | Our Method: SimGCN | | |
|---|---|---|---|---|---|---|---|
| Dataset | Metric | Method | Score | # Params. | Score | # Params. | Rank |
| **TDC.Pgp** (↑) | AUROC | AttrMasking | **0.929**±0.006 | 2067K | **0.929** ±0.01 | 1103K | Tied 1st |
| **TDC.Bioav** (↑) | AUROC | RDKit2D | 0.672±0.021 | 633K | **0.748** ±0.033 | 1103K | 1st |
| **TDC.BBB** (↑) | AUROC | ContextPred | 0.897±0.004 | 2067K | **0.901** ±0.007 | 1103K | 1st |
| **TDC.VD** (↑) | Spearman | RDKit2D | 0.561±0.025 | 633K | **0.582** ±0.031 | 1103K | 1st |
| **TDC.CYP3A4-S** (↑) | AUROC | CNN | **0.662**±0.031 | 227K | 0.64 ±0.016 | 1103K | 2nd |
| **TDC.CYP2C9-S** (↑) | AUPRC | ContextPred | 0.392±0.026 | 2067K | **0.433** ±0.017 | 281K | 1st |
| **TDC.Half_Life** (↑) | Spearman | Morgan | 0.329±0.083 | 1477K | **0.392** ±0.065 | 1103K | 1st |
| **TDC.CL-Micro** (↑) | Spearman | RDKit2D | 0.586±0.014 | 633K | **0.597** ±0.025 | 281K | 1st |
| **TDC.hERG** (↑) | AUROC | RDKit2D | 0.841±0.020 | 633K | **0.874** ±0.014 | 1103K | 1st |
| **TDC.DILI** (↑) | AUROC | AttrMasking | **0.919**±0.008 | 2067K | 0.909±.011 | 1103K | 2nd |

We look for the best hyperparameters using the validation data on the following search space: threshold in $\{0.6, 0.65, 0.7\}$, learning rate in $\{0.01, 0.001\}$, gnn hidden dimension in $\{64, 256\}$, number of layers in $\{2, 3\}$, and dropout rate in $\{0.1, 0.2\}$. We observe two particularly potent settings of hyperparameters; we list them in Listing 1 and Listing 2.

**Listing 1** 3 layers with 256 neurons each, and 0.2 dropout.

```
{
  "thres": 0.7,
  "sim_gnn_batch_size": 256,
  "sim_gnn_lr": 0.001,
  "sim_gnn_weight_decay": 0.001,
  "sim_gnn_patience": 45
  "sim_gnn_num_layers": 3,
  "sim_gnn_hidden_channels": 256,
  "sim_gnn_dropout": 0.2,
  "sim_gnn_predictor_hidden_feats": 1024,
  "sim_gnn_batchnorm": true,
  }
```

**Listing 2** 2 layers with 64 neurons each, and dropout 0.1.

```
{
  "thres": 0.7,
  "sim_gnn_batch_size": 256,
  "sim_gnn_lr": 0.001,
  "sim_gnn_weight_decay": 0.001,
  "sim_gnn_patience": 45
  "sim_gnn_num_layers": 2,
  "sim_gnn_hidden_channels": 64,
  "sim_gnn_dropout": 0.1,
  "sim_gnn_predictor_hidden_feats": 1024,
  "sim_gnn_batchnorm": true,

  }
```

# References


[1] Descriptastorus. https://github.com/bp-kelley/descriptastorus. Accessed: 02.06.2022. 2

[2] Rdkit: Open-source cheminformatics. https://www.rdkit.org/. 2020.03.1 (Q1 2020) Release. 2

[3] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):1–13, 2015. 2

[4] 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. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021. 1