

PAPER

Tensor Factorisation for Polypharmacy Side Effect Prediction

Oliver Lloyd,^{1,*} Yi Liu¹ and Tom R. Gaunt¹¹MRC Integrative Epidemiology Unit, Bristol Medical School, University of Bristol, United Kingdom

*Corresponding author. oliver.lloyd@bristol.ac.uk

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Adverse reactions caused by drug combinations are an increasingly common phenomenon, making their accurate prediction an important challenge in modern medicine. However, the polynomial nature of this problem renders lab-based identification of adverse reactions insufficient. Dozens of computational approaches have therefore been proposed for the task in recent years, with varying degrees of success. One group of methods that has seemingly been under-utilised in this area is tensor factorisation, despite their clear applicability to this type of data. In this work, we apply three such models to a benchmark dataset in order to compare them against established techniques. We find, in contrast to previous reports, that for this task tensor factorisation models are competitive with state-of-the-art graph neural network models and we recommend that future work in this field considers cheaper methods with linear complexity before running costly deep learning processes.

Key words: polypharmacy side effect prediction, adverse drug reaction, knowledge graph embeddings, tensor factorisation

Introduction

The comforts of modern life are causing a demographic shift in populations globally. By 2050, the number of people aged 60+ is projected to more than double [World Health Organisation, 2022], compared to an overall population increase of around 21% [United Nations, 2022]. With this increase, medical events that primarily affect older people will become increasingly important areas of research. Multimorbidity is one such phenomenon, whose prevalence among elderly populations may range from 55-98% [Marengoni et al., 2011]. Closely associated with multimorbidity is another issue: polypharmacy, the taking of two or more medications simultaneously by the same individual. Polypharmacy can be usefully employed in some contexts, for example in order to achieve drug synergism [Tallarida, 2011], where two medications are combined to produce an effect greater than the sum of their individual effects. However, the practise of polypharmacy can also lead to the emergence, via chemical interactions, of adverse drug reactions (ADRs) that are not associated with either drug individually [Ahmed et al., 2014]. ADRs put a large strain on healthcare systems, with a systematic review from 2002 putting their annual cost to the UK National Health Service at £380 million [Wiffen et al., 2002]. Given the general population increase and demographic shift mentioned above, plus inflation, it is very plausible that this annual cost may soon approach £1 billion.

Correct prediction of ADRs related to polypharmacy is a challenging task, owing mainly to the extremely large numbers

involved. In England, an estimated 8.4 million people are taking at least 5 prescribed medications, and a quarter of those are taking 10 or more [UK Government, 2021]. The combinatorial nature of the problem means that testing every possible n -combination of the thousands of commercially available drugs quickly becomes infeasible in wet-lab experiments or clinical trials, even with smaller values of n than are commonly found in real populations. As a result, pre-clinical screening with statistical/computational methods is a necessity for identifying drug combinations of interest [Ryall and Tan, 2015]. Recently, there has been an explosion of interest in graph-embedding methods for solving the problem of polypharmacy side effect (PSE) prediction. This can largely be traced back to the work of Zitnik et al., in which the authors construct a knowledge graph from the following data: drug-target, protein-protein interaction, monopharmacy side effects, and drug-pair side effects [Zitnik et al., 2018]. A portion of the drug-pair side effects are left out to enable out-of-sample prediction and assessment. By constructing the data in this way, the problem is cast as a multirelational link prediction (LP) problem, which, the authors claim, makes theirs the first technique that allows prediction of the *type* of side effect that will occur, rather than a simple binary categorisation or magnitude of effect. The Decagon model itself consists of two components. Firstly, node embeddings for the network are encoded using a graph convolutional model. These embeddings are then passed to a tensor factorisation (TF) decoder which produces a score for a particular side effect r between a drug pair (v_i, v_j) . Finally,

that score is passed to a sigmoid function which outputs a probability that the given triple (v_i, r, v_j) is true.

Commendably, Zitnik et al. made their data publicly available and since then it has been used as a common benchmark in PSE modelling research. Early adopters of the dataset found success with a wide variety of techniques, including kernel ridge regression [Dewulf et al., 2021], semantic predication embedding [Burkhardt et al., 2019], and product of experts models [Malone et al., 2019]. Each of these approaches improved upon the reported scores of Decagon while also being more efficient and easier to interpret. Despite this, as time has progressed we have seen an increasingly homogenised set of methods being applied to the Decagon dataset, with graph neural networks (GNNs) becoming dominant from 2022 onwards. GNNs have been reported to perform well on PSE prediction [Carletti et al., 2021, Zhuang et al., 2023, Saifuddin et al., 2023], particularly when information pertaining to the chemical structure of drugs is incorporated into the training process [Li et al., 2023]. Unfortunately, there is no avoiding the fact that they are expensive to run [Wu et al., 2019], and this problem is set to only get worse for drug-structural models given that larger biologic drugs are becoming more prevalent over time [Senior, 2023].

Knowledge graph embedding (KGE) methods are a broad family of techniques that take heterogenous graphs as input and cast the comprising nodes and relations into a low-dimensional vector space. They have proved useful in several real-world tasks, such as item recommendation [Catherine and Cohen, 2016], drug development and repurposing [Geleta et al., 2021], and knowledge base completion [Bordes et al., 2013]. The application of these techniques to the PSE modelling problem, however, has been somewhat scarce considering their relevance for this type of data, alongside their relatively low cost to run and optimise. This is even more surprising in the light of research which has suggested that simpler models can in fact outperform more complex ones when correctly tuned [Ali et al., 2022]. One possible explanation for this scarcity can be found in the original work of Zitnik et al. - they employed two such methods (RESCAL and DEDICOM) alongside their graph convolutional approach, reporting that these two scored the lowest out of all the methods tested. Even predictions based on the simple method of concatenating drug features achieved substantially higher scores as measured by the three test metrics. Subsequent work has suggested that this might be an overly pessimistic view of the capabilities of KGE methods for this task, with models such as DistMult, ComplEx, SimpleE, and RotatE all matching or surpassing the performance of Decagon [Malone et al., 2019, Kim and Shin, 2023, Dai et al., 2021]. Despite this, the upper limit for such methods (achieved by properly optimising hyperparameters) has yet to be tested. Without knowledge of this limit, we cannot say definitively whether the current state-of-the-art graph neural network methods are actually providing a worthwhile predictive improvement over cheaper KGE methods.

The contributions of this paper are twofold. Primarily, we aim to provide clarity on the true predictive capabilities of KGE methods on the PSE task by running a comprehensive hyperparameter optimisation process on selected models. By putting them through exactly the same assessment process outlined by Decagon [Zitnik et al., 2018], we enable our results to be compared with other research in the field. As a secondary goal, we compare options for incorporating monopharmacy data into the embedding process by repeating the analysis on two graphs where this data is handled differently.

Graph	Meta-nodes	Nodes	Meta-edges	edges
Selfloops	2	19734	11149	5485566
Non-naive	2	19734	964	5310589

Table 1. Node and edge counts for our two constructed variations of the Decagon graph. Meta-edges are the ‘types’ of edges allowed in the graph, such that each meta-edge can be associated with one adjacency matrix. A meta-node is a ‘type’ of node that all follow the same connectivity rules and appear together as one or both axes of an adjacency matrix. For example, an instance of the ‘Drug-Target’ meta-edge can only exist between one ‘Drug’ node and one ‘Gene’ node, and can be represented as a $d \times g$ adjacency matrix in which a value of 1 indicates that drug d_i targets gene g_j , otherwise the value is 0.

Methods

We downloaded the raw data used by Decagon from the Stanford Network Analysis Project <http://snap.stanford.edu/decagon/>. We then prepared the data for LibKGE [Broscheit et al., 2020], which reads graphs as a list of edges in RDF triple (subject, predicate, object) format. There are several ways to approach the conversion of the data to RDF format. In this work we test two such methods, which we have named ‘Selfloops’ and ‘Non-naive’ – the difference between them being the way in which monopharmacy side effect data is included in the resulting graph. The **Selfloops** approach treats these side effects as edges from one drug back to itself (hence the name). The **Non-naive** construction method is equivalent, but, following the example of Zitnik et al., monopharmacy data is instead modelled as n-hot node feature-vectors, with ‘hot’ columns indicating which side effects are associated with a given drug. We performed dimensionality reduction via principal component analysis (PCA) on these features to create a smaller matrix for each possible dimensional size of embeddings. We then modified LibKGE to load these vectors from disk, using them as the starting point for learning node embeddings for this dataset rather than any of its usual stochastic initialisation techniques. Graph statistics for the two networks are listed in table 1, and schemata are shown in figure 1.

All three methods included in this analysis can be classified as some form of TF. ComplEx [Trouillon et al., 2016] uses the Hermitian product to embed entities into the complex space. This (ironically) straightforward approach allows anti-symmetric relations to be modelled with a low-rank decomposition, which, in the real space, would only be possible for symmetric relations. DistMult [Yang et al., 2014], also referred to as ‘bilinear-diag’ by its authors, is similar to the well known TransE model, but uses a multiplicative operation rather than an additive one to combine dyadic vectors. Lastly, SimpleE [Kazemi and Poole, 2018] is an enhancement of the Canonical Polyadic (CP) decomposition for embedding KGs. CP itself was used often in early LP research but has a major shortfall in that it learns head and tail vectors for a given node independently – SimpleE addresses this problem by also considering inverted relations to create dependency between these vectors.

The hyperparameter optimisation space, established in configuration files, was the same for all models. Hyperparameter ranges were based on those from the search performed by the LibKGE developers in their demonstrative paper [Ruffinelli et al., 2020]. We made a few adjustments to the space, notably expanding the number of available options for optimiser and loss function from 2 and 3, to 5 and 4 respectively. Other alterations included adding possible embedding sizes of

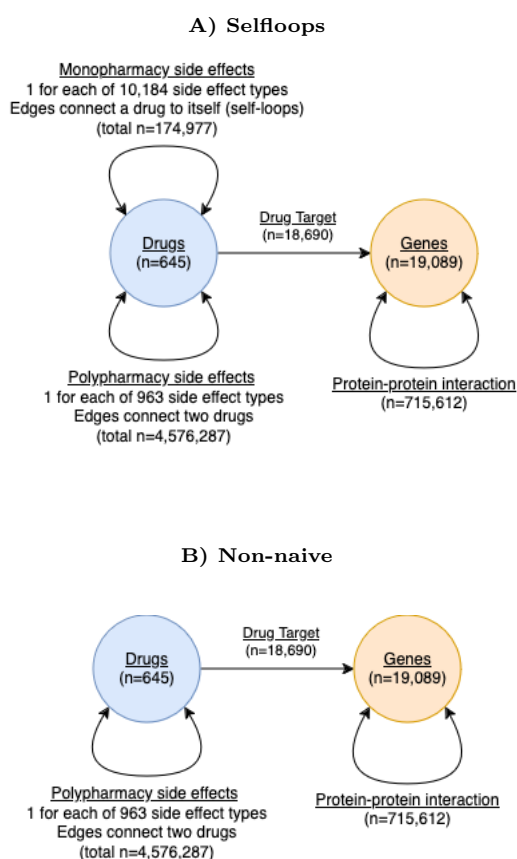


Fig. 1. Schemata for the two graph variants. The only difference between the two graphs relates to the handling of monopharmacy side effect data. The ‘Selfloops’ approach (subfigure A) includes monopharmacy data as self-looping edges from a given drug back to itself, whereas the ‘Non-naive’ graph (subfigure B) doesn’t contain these edges. Instead, monopharmacy associations are modelled as n-hot vectors of length 10,184. At the start of an embedding trial, these vectors are reduced to the selected embedding sized using singular value decomposition and then used as the initialisation points for the drug embeddings. When training on the Selfloops graph, embeddings are initialised by drawing from a random distribution, as is standard in LibKGE.

32 and 64, and reducing the aggressiveness of the learning rate scheduler. The grid searches took place over 100 trials, with the first 50 having values chosen by Sobol sequence and the remaining 50 chosen by Bayesian optimisation through the Ax platform. This Bayesian method works by iteratively fitting surrogate models (Gaussian processes) of trial outcomes against the attempted hyperparameter configurations – these models are used to determine the ‘value’ of choosing a particular new parameterisation by considering both the predicted outcome of that configuration, as well as the corresponding uncertainty. The parameterisation that provides the best value is then chosen for the next trial, and afterwards the process is repeated. All configuration files used to create the experiments are available in our repository <https://doi.org/10.5281/zenodo.10684402>.

We removed a portion of both test graphs, prior to the main analysis, for holdout validation. Following the methodology used by Decagon [Zitnik et al., 2018], the holdout data was created by randomly removing 10% of the edges belonging to each PSE. Since the two test graphs incorporate the same polypharmacy data, the same edges were held out from both

Model	Dataset	AUROC	AUPRC	AP@50
SimplE	Selfloops	0.978	0.971	1.000
SimplE	Non-naive	0.973	0.965	0.980
Complex	Selfloops	0.970	0.963	0.978
Complex	Non-naive	0.967	0.960	0.978
DistMult	Selfloops	0.946	0.930	0.941
DistMult	Non-naive	0.938	0.921	0.932

Table 2. Median performance across 963 polypharmacy side effect types as measured by three metrics for each of our six experiments.

in order to enable fair comparison between the two datasets. This holdout set contains 458,061 edges, leaving a total of 5,761,807 edges in the **Selfloops** graph, and 5,586,830 in the **Non-naive** graph. During training, performance was measured using the standard LP metrics employed in LibKGE, namely mean reciprocal rank (MRR) and hits@k. Performance on the hold-out edges, our measure for overall quality, was assessed with the same metrics used by the Decagon authors - area under receiver-operating characteristic curve (AUROC), area under precision-recall curve (AUPRC), and average precision at 50 (AP@50). These were calculated individually per side effect type. A flowchart of the methodology of this paper is displayed in figure 2.

This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol <http://www.bristol.ac.uk/acrc>. The specific environment was CentOS-7 running Python 3.8.12 with PyTorch 1.7.1, accelerated with CUDA 11.4 on 4× NVIDIA GeForce RTX 2080 Ti.

Results

Results of the assessment are shown in figure 3. All three of our tested KGE methods outperformed Decagon on both of our constructed graphs, with median scores improving on Decagon’s performance by 7.62 - 12.2% for AUROC, 10.7 - 16.8% for AUPRC, and 16.1 - 24.5% for AP@50. Remarkably, the rankings of our six configurations by median score do not change regardless of which metric is used to order them (table 2).

To provide some context to the performance of our best model, SimplE on the **Selfloops** graph, we gathered information about the highest performing methods that have been presented in the literature. The top 10 of these, as ranked by AUPRC, are shown in table 3.

On our investigation of the mechanism for including monopharmacy data into the embedding pipeline, we find that the **Selfloops** graph achieves better median results than the **Non-naive** graph for all three embedding methods. In all cases, however, this improvement is ≤ 0.02 , and in the case of the Complex model measured by AP@50, the difference is as low as 0.0004.

Each trial, of the 100 per experiment, ran for a variable number of epochs that was determined by an early stopping procedure. This procedure checks the performance of the trial every five epochs starting from 50, ending the trial if the performance has not improved after two consecutive checks. Consequently, to compare the speed of computation between experiments we first standardized the time by dividing it by the total number of epochs run. Figure 4.A shows these values. Although DistMult is the fastest, with a mean time-per-epoch

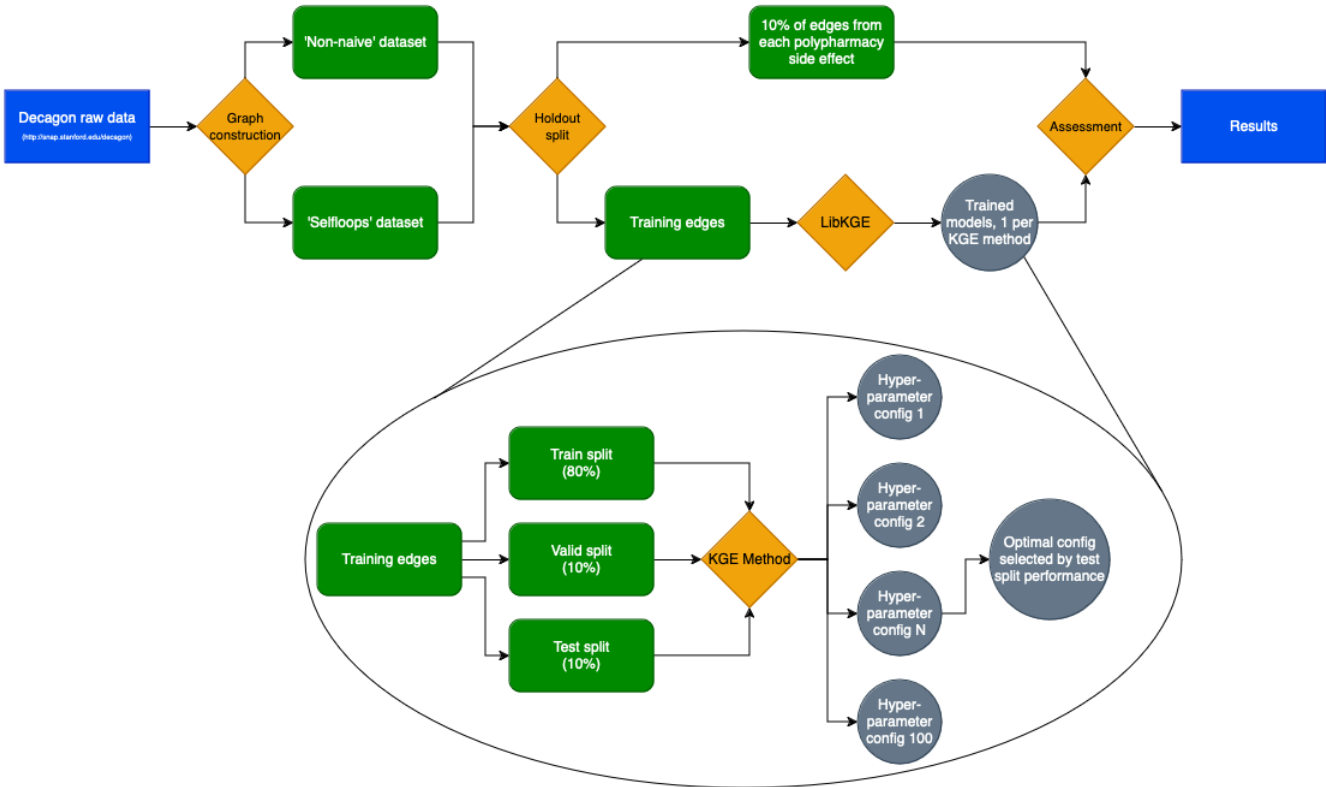


Fig. 2. Workflow for the reported experiment. The datasets that are created from the raw data differ only in their handling of monopharmacy data. The holdout split is the same for both because they contain the same polypharmacy data. Both are then put through the workflow on the right hand side of the ‘holdout split’ node, and results compared. The lower diagram offers a ‘zoomed in’ look at the learning process in LibKGE that is not displayed in the upper diagram. This learning process happens once for each of the 6 dataset-model combinations. Key: blue = raw data; yellow = software/code; green = processed data; grey = trained model.

Model	AUPRC	Mechanism	Extra data	Citation
Carletti’s	0.998	GAT	None	[Carletti et al., 2021]
MS-ADR	0.983	GCN	Enzyme and transporter	[Zhuang et al., 2022]
ADGCL	0.980	GNN	Enzyme and transporter	[Zhuang et al., 2023]
GS-ADR	0.972	GAT	Enzyme and transporter	[Zhuang and Wang, 2021]
SimpleE (this work)	0.971	Tensor factorisation	None	-
SimVec	0.968	Auto-Encoder	Chemical substructure	[Lukashina et al., 2022]
HyGNN	0.965	GCN	Chemical substructure	[Saifuddin et al., 2023]
HLP	0.965	GNN	Chemical substructure	[Vaida and Purcell, 2019]
DeepDrug	0.960	GNN	Chemical/protein substructure	[Yin et al., 2023]
M2GCN	0.953	GNN	None	[Liu et al., 2023]

Table 3. Top 10 performing models by AUPRC on the Decagon dataset. GAT = Graph Attention Network, GCN = Graph Convolutional Network, GNN = Graph Neural Network.

of 165 seconds on **Selfloops** and 183 on **Non-naive**, we observe surprising consistency overall with only a small gap from DistMult to the other models. The slowest overall was SimpleE on **Non-naive**, achieving an average epoch completion time of 219 seconds. Figure 4.B shows the mean number of epochs (plus standard error) that the trials ran for in each experiment. The DistMult and SimpleE models had comparable epochs-per-trial between the two datasets, with the former always stopping at the minimum count of 55. Interestingly, ComplEx was about 30% faster on **Non-naive**, with a mean epoch time of 81 seconds versus 115 for the slightly larger graph **Selfloops**.

We observed varying hyperparameter configurations of the optimal models from each experiment. Either 1vsAll or KvsAll was the chosen method in all cases for the sampling of negative training examples, with the top (SimpleE-Selfloops) and bottom two (DistMult) models using 1vsAll and the others using KvsAll. The chosen loss functions matched this pattern exactly - Binary Cross-Entropy (BSE) was used under the KvsAll strategy, and Kullback-Leibler divergence (KL) used under 1vsAll. Adamax was the modal optimiser, chosen in four out of six cases. The two exceptions, SimpleE-Selfloops and ComplEx-Selfloops, used Adam and Adadelta respectively. Dimensionality also varied between the embeddings, with half

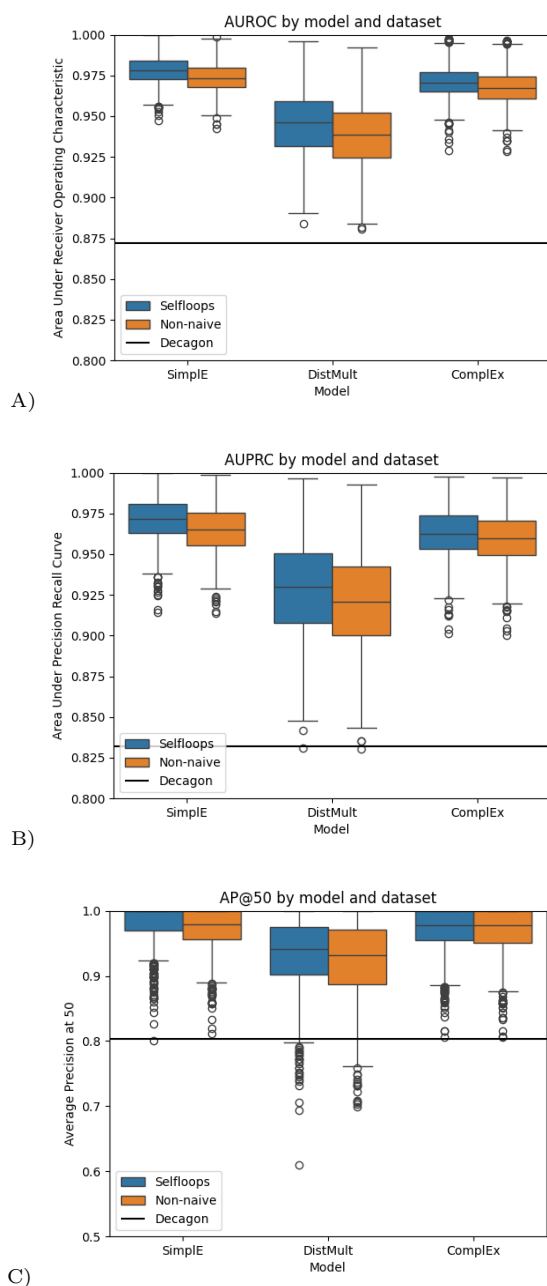


Fig. 3. Out-of-sample prediction performance over 963 side effects as measured by three metrics: A) Area Under Receiver Operating Characteristic; B) Area Under Precision Recall Curve; C) Average Precision at 50. Results are presented for each of 6 experiments, comprising 3 models (ComplEx, DistMult, SimpleE) run on two datasets (Non-naive, Selfloops). Decagon's performance on the same data is indicated by the horizontal line.

the models opting for $m = 128$, two selecting $m = 256$, and just one using $m = 512$. Since **Non-naive** initialises embedding values with principal components of the n -hot monopharmacy side effect vector, the 'weight initialisation' hyperparameter was only relevant for models on the **Selfloops** dataset. Within these, ComplEx and DistMult simply drew values from a normal distribution, whereas SimpleE subsequently applied the 'Xavier' [Glorot and Bengio, 2010] modification to these values.

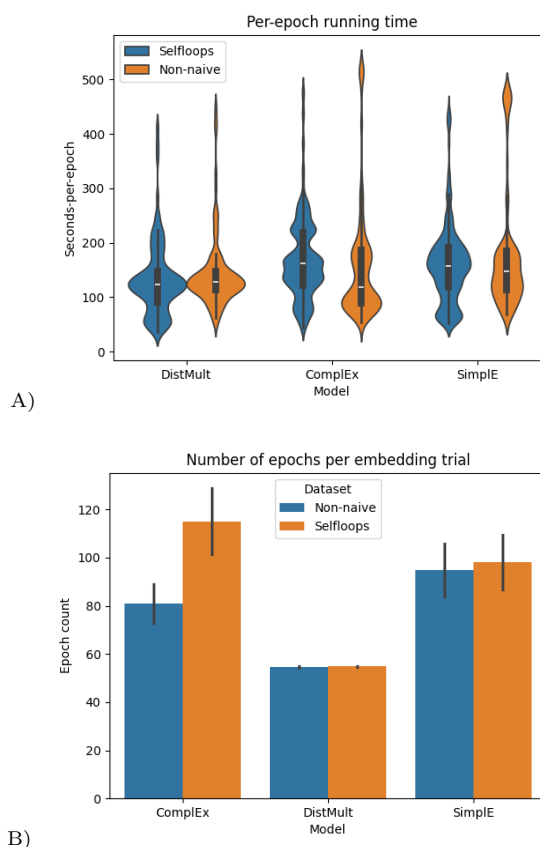


Fig. 4. Running time information for the six experiments. Subfigure A is a violin plot showing the distribution of compute time per-epoch. Subfigure B shows a bar plot of the mean number of epochs per-trial, with a black bar representing the standard error. An early stopping procedure was used to determine the maximum number of epochs in a trial, with a minimum of 55 and a maximum of 500.

Discussion

The predictive performance reported here ranks among the best to date - in fact, there are only four models that claim to achieve greater AUPRC (table 3). The highest of these four, Carletti et al.'s model, extends Decagon's architecture by adding a Relation Attention Module (RAM) which "increases the expressiveness of the network by weighting the contribution of the messages exchanged over different relations". This relatively simple modification to the Decagon method reportedly causes a drastic improvement in prediction performance, raising both AUROC and AUPRC to 0.998 from 0.872 and 0.832 respectively. However, the authors provide no publicly available repository to enable reproduction of their work and, whilst they did send archived code on request, the lack of documentation meant we were unable to run it. The absence of the original coder from their team also meant that they could provide no technical assistance and as a result we were unable to reproduce their reported model performance. The 2nd to 4th ranked methods all come from the same pair of authors, Luhe Zhuang and Hong Wang of Shandong Normal University. All three of their approaches use the same dataset - a modified version of Decagon's graph, with enzyme and transporter data added and 97 drug nodes removed. It is unclear which drug nodes were removed and no motivation for doing so is given.

The earliest presented of their models, GS-ADR, focuses on the concept of signed edges in the graph, with the paper reporting that consideration of this results in ‘more effective’ feature representation of drugs [Zhuang and Wang, 2021]. MS-ADR, published in 2022, also uses signed edges but additionally considers four different biomedical ‘views’ (enzyme, indication, side effect, and transporter) when convolving information across the graph [Zhuang et al., 2022]. ADGCL is the latest of their models and is similar to MS-ADR but makes use of an Implicit GNN rather than a convolutional encoder, finding that comparable performance is achieved [Zhuang et al., 2023]. As with Carletti et al.’s work, none of these three papers offer a publicly available repository of the code used to enable replication of their findings. Although AUPRC was not measured in their paper, it is well worth mentioning the work of Li et al., who report an AUROC of 0.999 on the Decagon dataset by incorporating drug atomic structure into the network [Li et al., 2023]. In this case, drugs are viewed as both singular graphs, where edges are bonds between atomic nodes, and also bipartite graphs, where potentially interacting drug pairs have edges from each atom to all atoms in the other molecule. By incorporating both the ‘inter’ and ‘intra’ drug views into a co-attention decoder, Li et al. achieve what is arguably the current state of the art when it comes to predictive performance on the Decagon data.

One large benefit of TF methods is their relative simplicity. ComplEx, DistMult, and Simple all have linear complexity with respect to the size of the embeddings produced [Rossi et al., 2021]. Even if we take the results of the papers mentioned above at face value, they are achieved by employing GNNs which have at least quadratic complexity, if not cubic or higher [Wu et al., 2021]. This is a disadvantage for two main reasons. Firstly, it hinders scalability - the inclusion of additional data into the base Decagon graph is a common theme in the surrounding literature, with seven out of the top ten performing models (by AUPRC) bringing in at least one more set of edges (3). One has to imagine that prediction performance could be maximised by including as many relevant data sources as possible, but as the size of the graph grows GNNs will become prohibitively expensive in computational terms. Secondly, a higher cost to training means that optimisation of hyperparameters is more difficult. Under a grid search scenario, far more hyperparameter combinations can be tested when the model of interest has reduced complexity, so an optimal configuration for the model can be found more easily and better prediction can be achieved at lower cost. This factor may well explain the surprisingly good performance of the models in the work presented here, as for each model-dataset pair we were able to choose the best from 100 trials based on the empirical evidence of their performance on the test split within LibKGE.

On the other hand, GNNs do have a definite advantage when it comes to inductive reasoning. TF methods, as they currently exist, simply cannot make inferences about unseen nodes and would require the entire graph to be re-embedded if predictions about new drugs were required. The Decagon dataset is far from exhaustive with its 645 drugs, and even if it were, we will continue to see new drugs developed over time, all of which will need to be assessed for possible PSEs. This is the largest drawback of the models employed in this work, and is something that would certainly need to be addressed before they could be used in any translational sense. Another limitation of both sets of techniques is the restriction to drug dyads, when, in real life, patients may often be co-prescribed three or more drugs [UK Government, 2021]. Unique interactions can occur in such cases,

e.g. when taking skeletal muscle relaxants [Chen et al., 2022], so a clinically useful ‘polypharmacy’ side effect prediction model might need to be able to consider more than just the ‘duopharmacy’ case. Hypergraph embedding models, such as those used by HyGNN [Saifuddin et al., 2023] and HLP [Vaida and Purcell, 2019], seem the most promising candidates to solve this issue because, by definition, hyperedges can connect any number of nodes. However, research in this direction has so far been scarce, probably due to the lack of available training data available for such a task.

In this work we observed a slight improvement in predictive performance when including monopharmacy information as self-looping edges in the graph, compared to using it to initialise embedding vectors. This improvement held for all metrics and models, with the exception of ComplEx where an AP@50 score of 0.978 was achieved on both graphs. The level of improvement was at best 0.02, however, so the self-looping form of the dataset should only be considered ‘better’ if the overall efficiency of the model is not noticeably hampered by the addition of the 174,977 extra edges. As shown in figure 4.A, runtime-per-epoch was not notably different between the two datasets when using the DistMult or Simple models. ComplEx did take 34 seconds longer on average (81 vs 115) when embedding **Selfloops**, so in this particular case it may actually be worthwhile to include dimensionally-reduced monopharmacy data as embedding initialisation points, as was done by Zitnik et al.. Generally speaking though, we can conclude that the monopharmacy side effect data should be modelled as graph edges instead when working with TF models.

In summary, we have shown here that TF embedding methods can rival state-of-the-art GNNs on the task of PSE prediction, even when trained with less data. Given the nature of the challenge, scalability should be a primary concern for anyone developing such models, and methods with linear complexity provide a huge advantage in this regard. They also bring the benefit of lower electricity consumption and therefore a reduced carbon footprint. This is, of course, an important consideration in the context of climate change. Zitnik et al. reported very poor performance of their only tested TF models, with both RESCAL and DEDICOM being outperformed even by a simple predictor. This may well explain the corresponding lack of application by later researchers. However, when set up with optimal hyperparameters and run with modern loss functions and sampling strategies such as those implemented in LibKGE, we have demonstrated that they do compete with the state-of-the-art. If the hurdle of inductive reasoning can be cleared, TF methods could be the most cost-effective way to tackle PSE prediction.

Competing interests

T.R.G. receives funding from Biogen and GSK for unrelated research.

Author contributions statement

Oliver Lloyd: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization.

Yi Liu: Conceptualization, Writing - Review & Editing, Supervision.

Tom R. Gaunt: Conceptualization, Writing - Review

& Editing, Supervision, Project administration, Funding acquisition

Acknowledgments

The authors thank Patrick Rubin-Delanchy for his helpful input.

References

- B. Ahmed, K. Nanji, R. Mujeeb, and M. J. Patel. Effects of polypharmacy on adverse drug reactions among geriatric outpatients at a tertiary care hospital in karachi: A prospective cohort study. *PLoS ONE*, 9, 11 2014. ISSN 19326203. doi: 10.1371/journal.pone.0112133.
- M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, and J. Lehmann. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8825–8845, 2022. doi: 10.1109/TPAMI.2021.3124805.
- A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data, 2013.
- S. Broscheit, D. Ruffinelli, A. Kochsiek, P. Betz, and R. Gemulla. Libkge a knowledge graph embedding library for reproducible research, 2020.
- H. A. Burkhardt, D. Subramanian, J. Mower, and T. Cohen. Characterizing the utilization of the problem list for pediatric trauma care. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2019:992–1001, 2019. ISSN 1942597X.
- V. Carletti, P. Foggia, A. Greco, A. Roberto, and M. Vento. Predicting polypharmacy side effects through a relation-wise graph attention network. volume 12644 LNCS, pages 119–128. Springer Science and Business Media Deutschland GmbH, 2021. ISBN 9783030739720. doi: 10.1007/978-3-030-73973-7_12.
- R. Catherine and W. Cohen. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. pages 325–332. Association for Computing Machinery, Inc, 9 2016. ISBN 9781450340359. doi: 10.1145/2959100.2959131.
- C. Chen, S. Hennessy, C. M. Brensinger, G. K. Dawwas, E. K. Acton, W. B. Bilker, S. P. Chung, S. Dublin, J. R. Horn, T. A. Miano, T. P. P. Nguyen, S. E. Soprano, and C. E. Leonard. Skeletal muscle relaxant drug–drug–drug interactions and unintentional traumatic injury: Screening to detect three-way drug interaction signals. *British Journal of Clinical Pharmacology*, 88:4773–4783, 11 2022. ISSN 13652125. doi: 10.1111/bcp.15395.
- Y. Dai, C. Guo, W. Guo, and C. Eickhoff. Drug–drug interaction prediction with wasserstein adversarial autoencoder-based knowledge graph embeddings. *Briefings in Bioinformatics*, 22:1–16, 7 2021. ISSN 1467-5463. doi: 10.1093/bib/bbaa256.
- P. Dewulf, M. Stock, and B. D. Baets. Cold-start problems in data-driven prediction of drug–drug interaction effects. *Pharmaceuticals*, 14:429, 5 2021. ISSN 1424-8247. doi: 10.3390/ph14050429.
- D. Geleta, A. Nikolov, G. Edwards, A. Gogleva, R. Jackson, E. Jansson, A. Lamov, S. Nilsson, M. Pettersson, V. Poroshin, B. Rozemberczki, T. Scrivener, M. Ughetto, and E. Papa. Biological insights knowledge graph: an integrated knowledge graph to support drug development. *bioRxiv*, page 2021.10.28.466262, 11 2021. doi: 10.1101/2021.10.28.466262.
- X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.
- S. M. Kazemi and D. Poole. Simple embedding for link prediction in knowledge graphs. *Advances in Neural Information Processing Systems*, 2018-December:4284–4295, 2 2018.
- J. Kim and M. Shin. A knowledge graph embedding approach for polypharmacy side effects prediction. *Applied Sciences*, 13:2842, 2 2023. ISSN 2076-3417. doi: 10.3390/app13052842.
- Z. Li, S. Zhu, B. Shao, X. Zeng, T. Wang, and T.-Y. Liu. Dsn-ddi: an accurate and generalized framework for drug–drug interaction prediction by dual-view representation learning. *Briefings in Bioinformatics*, 24, 1 2023. ISSN 1467-5463. doi: 10.1093/bib/bbac597.
- Q. Liu, E. Yao, C. Liu, X. Zhou, Y. Li, and M. Xu. M2gcn: multi-modal graph convolutional network for modeling polypharmacy side effects. *Applied Intelligence*, 53: 6814–6825, 3 2023. ISSN 15737497. doi: 10.1007/s10489-022-03839-z.
- N. Lukashina, E. Kartysheva, O. Spjuth, E. Virko, and A. Shpilman. Simvec: predicting polypharmacy side effects for new drugs. *Journal of Cheminformatics*, 14:49, 12 2022. ISSN 17582946. doi: 10.1186/s13321-022-00632-5.
- B. Malone, A. García-Durán, and M. Niepert. Knowledge graph completion to predict polypharmacy side effects. volume 11371 LNBI, pages 144–149. Springer Verlag, 2019. ISBN 9783030060152. doi: 10.1007/978-3-030-06016-9_14.
- A. Marengoni, S. Angleman, R. Melis, F. Mangialasche, A. Karp, A. Garmen, B. Meinow, and L. Fratiglioni. Aging with multimorbidity: A systematic review of the literature, 9 2011. ISSN 15681637.
- A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, 15, 1 2021. ISSN 1556472X. doi: 10.1145/3424672.
- D. Ruffinelli, S. Broscheit, and R. Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. *International Conference on Learning Representations*, 2020.
- K. A. Ryall and A. C. Tan. Systems biology approaches for advancing the discovery of effective drug combinations rajarshi guha. *Journal of Cheminformatics*, 7:7, 12 2015. ISSN 17582946. doi: 10.1186/s13321-015-0055-9.
- K. M. Saifuddin, B. Bumgardner, F. Tanvir, and E. Akbas. Hygcn: Drug-drug interaction prediction via hypergraph neural network. volume 2023-April, pages 1503–1516. IEEE Computer Society, 2023. ISBN 9798350322279. doi: 10.1109/ICDE55515.2023.00119.
- M. Senior. Fresh from the biotech pipeline: fewer approvals, but biologics gain share. *Nature Biotechnology*, 41:174–182, 2 2023. ISSN 15461696. doi: 10.1038/S41587-022-01630-6.
- R. J. Tallarida. Quantitative methods for assessing drug synergism, 11 2011. ISSN 19476019.
- T. Trouillon, J. Welbl, S. Riedel, U. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. *[International Conference on Machine Learning Logo] Proceedings of Machine Learning Research*, 48:2071–2080, 6 2016. ISSN 1938-7228.

- UK Government. National overprescribing review report, 2021. URL <https://www.gov.uk/government/publications/national-overprescribing-review-report>.
- United Nations. Population, 2022. URL <https://www.un.org/en/global-issues/population>.
- M. Vaida and K. Purcell. Hypergraph link prediction: Learning drug interaction networks embeddings. pages 1860–1865. Institute of Electrical and Electronics Engineers Inc., 12 2019. ISBN 9781728145495. doi: 10.1109/ICMLA.2019.00299.
- P. Wiffen, M. Gill, J. Edwards, and A. Moore. Adverse drug reactions in hospital patients: a systematic review of the prospective and retrospective studies. *Bandolier*, 2002.
- World Health Organisation. Ageing and health, 2022. URL <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health>.
- Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4–24, 2019. URL <https://api.semanticscholar.org/CorpusID:57375753>.
- Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32:4–24, 1 2021. ISSN 21622388. doi: 10.1109/TNNLS.2020.2978386.
- B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng. Learning multi-relational semantics using neural-embedding models. *arXiv Preprint*, 11 2014. URL <http://arxiv.org/abs/1411.4072>.
- Q. Yin, X. Cao, R. Fan, Q. Liu, R. Jiang, and W. Zeng. Deepdrug: A general graph-based deep learning framework for drug-drug interactions and drug-target interactions prediction. *Quantitative Biology*, 0:0, 9 2023. ISSN 2095-4689. doi: 10.15302/j-qb-022-0320.
- L. Zhuang and H. Wang. Drug-drug adverse reactions prediction based on signed network. pages 277–281. Institute of Electrical and Electronics Engineers Inc., 2021. ISBN 9781665406796. doi: 10.1109/ITME53901.2021.00064.
- L. Zhuang, H. Wang, W. Li, T. Liu, S. Han, and H. Zhang. Ms-adr: predicting drug–drug adverse reactions base on multi-source heterogeneous convolutional signed network. *Soft Computing*, 26:11795–11807, 11 2022. ISSN 14337479. doi: 10.1007/s00500-022-06951-x.
- L. Zhuang, H. Wang, M. Hua, W. Li, and H. Zhang. Predicting drug-drug adverse reactions via multi-view graph contrastive representation model. *Applied Intelligence*, 53:17411–17428, 7 2023. ISSN 15737497. doi: 10.1007/s10489-022-04372-9.
- M. Zitnik, M. Agrawal, and J. Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34:i457–i466, 7 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty294.