HeteroGenius: An Improvised 'Intelligence' in Heterogeneous Graph Transformers

Conference Paper · December 2022

DOI: 10.1109/ICMLA55696.2022.00093

4 authors, including:

Nafiz Sadman
Queen's University
20 PUBLICATIONS **106** CITATIONS
SEE PROFILE

Akib Sadmanee
University of Hawai'i at Mānoa
6 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Kishor Datta Gupta
Clark Atlanta University
98 PUBLICATIONS **1,031** CITATIONS
SEE PROFILE

HeteroGenius: An Improvised 'Intelligence' in Heterogeneous Graph Transformers

Nafiz Sadman
Machine Learning
Silicon Orchard Ltd.
Bangladesh
nafiz@siliconorchard.com

Akib Sadmanee
Machine Learning
Silicon Orchard Ltd.
Bangladesh
akib@siliconorchard.com

Kishor Datta Gupta
Department of Cyber-Physical Systems
Clark Atlanta University
Atlanta, Georgia, USA
kgupta@cau.edu

Roy George
Department of Cyber-Physical Systems
Clark Atlanta University
Atlanta, Georgia, USA
rgeorge@cau.edu

***Abstract*—Heterogeneous graphs can capture pragmatic relations between entities (or nodes) better than homogeneous graphs. Heterogeneous graphs are crucial in search and classification problems and can correlate with social network graphs. However, this increases complexity and demands a clear understanding of the relationships, and rankings of the network. The ranking can take the form of various scoring-based systems, or finding the importance of the relations between two entities. In this research, we practice the use of incorporating a meta-edge definition, 'Force', between nodes to embed meaning to its dimensionality. We add this 'Force' to the Heterogeneous Graph Transformer, which we term as 'HeteroGenius', and experimentally demonstrate that the addition increases the overall accuracy by 2%.**

***Index Terms*—Heterogeneous Graphs, Transformers, Information Embedding**

I. Introduction

Graph Neural Networks (GNN) has piqued the curiosity of numerous scholars in recent years. GNNs use a neighborhood aggregation algorithm to generate a node's representation vector by iteratively accumulating and manipulating the representation vectors of its neighbors. One prominent branch of graph learning is based upon the concept of stacking learned "graph convolutional" layers that conduct feature transformation and neighbor aggregation [1], which has resulted in an explosion of versions known as Graph Neural Networks (GNNs) [2]–[4]. To date, GNNs were applied in many applications in different fields like healthcare, natural sciences, recommender systems, and scene comprehensions [5]. Though these applications solve the problems in the homogenous graph domain, large graphs utilized in web-scale classification problems are frequently heterogeneous.

In this paper, we value heterogeneous graphs as a better representation of real-world events compared to homogeneous graphs. Homogeneous graphs are entitled to singular node-edge type and suffer generalizability problems [6]. Using heterogeneous graphs is more pragmatic in the sense that there are various types of entities interacting with each other. They are capable to model complex systems and form hierarchies to support neighborhood importance [7]. Facebook Social Graph [8] is an excellent example of a heterogeneous graph that can profile each user and can be used to track particular user activity. There are many use cases of Heterogeneous GNNs [9], [10] with varied graph embedding techniques. Recent developments have been made using the popular transformer model [11], which had a revolutionary impact on the fields of Artificial Intelligence. We aim to utilize the graph embedding concept introduced by Felfeli et. al [12]. The authors proposed to map homogeneous graphs to physical systems and used Coulomb's Law ($F = k(q_1q_2)/r^2$) to represent node-edge importance. Their method implies that nodes interact by 'forces' derived from the 'potential' that each node creates at the site of other nodes, resulting in a potential gradient that reflects the 'natural' direction of diffusion through the network. However, we redefine this equation by acknowledging the characteristics of a heterogeneous graph. Particularly, we experiment using the Graph Transformer model proposed by Hu et.al [13] on the Open Academic Graph dataset[1].

Our primary contribution is the redefinition of the Force and introducing it in the attention-heads as well as with the message passing in the Heterogeneous graph transformer. We use the two preprocessed Open Academic Graph's graph dataset[2] ML (Machine Learning) and NN (Neural Network). These are made publicly available for use by Hu et. al. We evaluate the experiments on the paper-field classification task and conclude that the overall accuracy increases by 2%. We named this modified heterogeneous graph transformers as 'HeteroGenius'.

The organization of the paper starts off with a brief introduction to graph jargons in Section II. We present our proposal in Section III, followed by experiment configurations in Section IV. Respective experiment results are discussed in Section V. We conclude our constraints and future plans in Section

[1]https://www.microsoft.com/en-us/research/project/open-academic-graph/
[2]https://github.com/acbull/pyHGT/tree/master/OAG

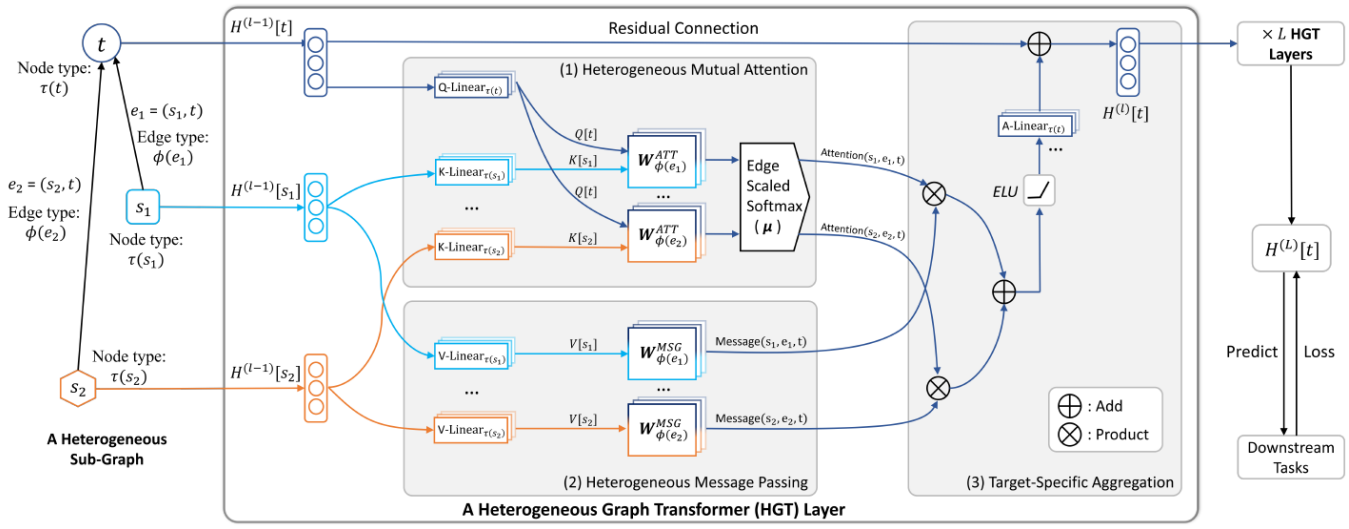


Fig. 1: **Heterogeneous Graph Transformer** [13]: The transformer model encodes the source and target nodes along with the meta path. These are fed to three different functions.

VI and we discuss some prior work that has been done on Heterogeneous graphs using GNNs in Section VII.

II. Preliminaries

A. Graph Neural Networks

A graph is a data structure that is composed of a finite number of vertices and a collection of unordered pairings of these vertices in the case of an undirected graph or a set of ordered pairs in the case of a directed graph. The purpose of GNN is to acquire a state embedding that incorporates labeled notes as well as information about each node's neighborhood in order to forecast the distribution of unlabeled nodes. The state embedding e_s can be defined by

$e_s = f(G_s, G_{co[s]}, e_{ne[s]}, G_{ne[s]})$

where $G_s, G_{co[s]}, e_{ne[s]} \& G_{ne[s]}$ denote the features of s, the features of the edges connecting with s, the embedding state of the neighboring nodes of s, and the features of the neighboring nodes of s respectively. The f function in this case represents the local transition function. GNN employs Banach's Fixed Point Theorem and assumes that the transition function is a contraction map, ensuring that the state vector X of the node eventually converges to a contraction Point. As a result, it is sometimes referred to as the convergence-based technique following the equation below.

$H = F(E, G)$

The matrices holding all the states and all the features are denoted by E and G, respectively, and the global transition function is denoted by F. The equation may be rewritten using Banach's Fixed Point Theorem as

$H^{t+1} = F(E^t, G)$

If we consider O to be the local output function, then the output $output_s$ is defined as

$output_s = O(E_s, G_s)$

where E_s is the state embedding and G_s are the features of s.

B. Node Embedding

Embeddings record the graph topology, node connections, and any other important information as numerical values. In a graph-structure, how this information is retrieved, is determined by the network-related questions we ask. A possible approach to this problem is to build embedding in such a way that the node embedding of two nodes are similar in some way if they happen to be similar in the real network. In other words, one can decide to form embedding based on the principle of similarity. Nodes that are similar in the network will have similar embedding.

C. Attention based Graphs

Attention is an approach to implementing a brain action of selectively concentrating on a few relevant things while ignoring others in deep neural networks. Proposed by Bahdanu et. al [14], attention layers assist in adding extra (important) weights to certain vectors. While attention is mostly used to determine the importance of word vectors, it can also be utilized to understand the importance of certain nodes. The authors [15] propose graph attention networks (GATs), which are new neural network designs that operate on graph-structured data and use masked self-attention layers to overcome the drawbacks of prior techniques based on graph convolutions or approximations. They implicitly enable specifying different weights to different nodes in a neighborhood by stacking layers in which nodes are able to attend over their neighborhoods' features, without requiring any kind of costly matrix operation (such as inversion) or requiring prior knowledge of the graph structure.

D. Heterogeneous Graph Transformer

A heterogeneous graph is a sort of data structure that has different categories of items or various types of relationships. A heterogeneous graph,

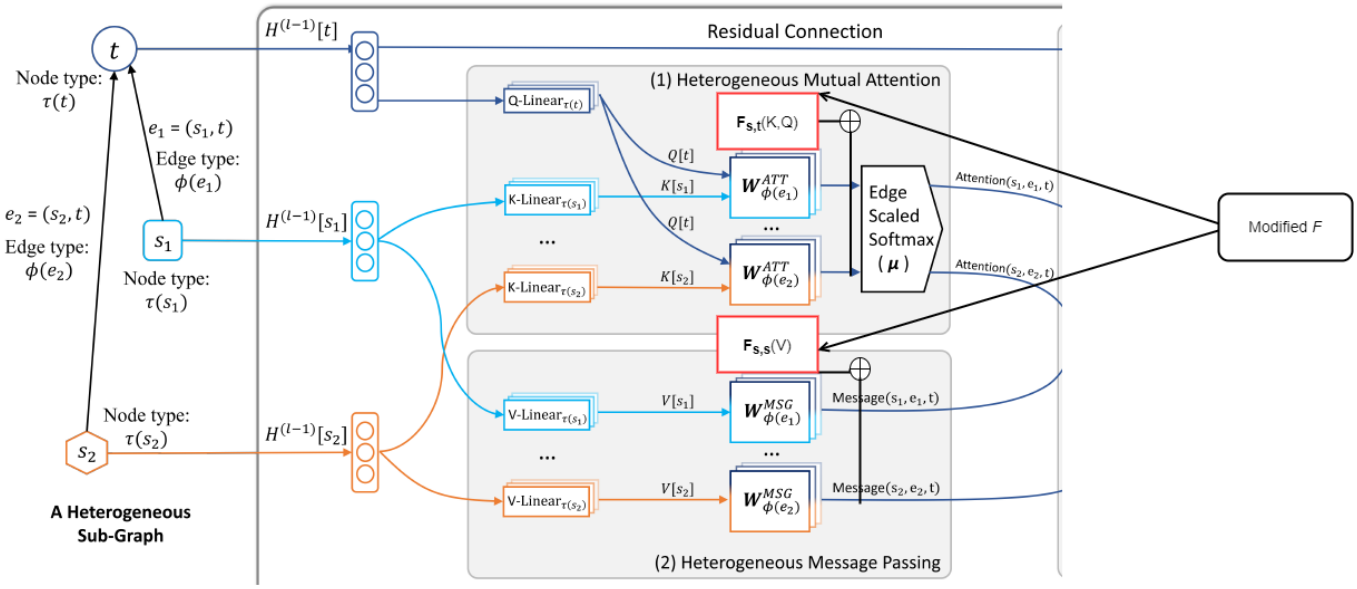


Fig. 2: **HeteroGenius:** Introducing modified Force, F, to Heterogeneous Mutual Attention weights and Heterogeneous Message Passing weights.

$G = (V, E)$

where V is an item and E is a set of links. A heterogeneous graph can be expressed in terms of a node-to-type mapper and a link-to-type mapper function. A node-to-type mapper function can be expressed as

$\phi : V \rightarrow A$

A link-to-type mapper function can be defined as

$\psi : \epsilon \rightarrow R$

Here, A and R denote the sets of predefined object types and link types respectively.

Heterogeneous Graph Transformer (HGT) is proposed by Hu et. al [13]. The authors tried to solve the problem of homogeneity in graphs and introduce heterogeneous graph transformers for Web graph data. The authors used relative temporal encoding into HGT to handle time-dependent features and formed a mini-batch sampling algorithm on heterogeneous graphs(HGSampling) to deal with big data. The graph heterogeneity is addressed by implementing node-and-edge-type dependent attention mechanism, where each edge $e =< s, t >$. (e = edge, s = node 1, t = node 2, where node 1 and node 2 are of different types). Meta relations $<(s), (e), (t)>$ among nodes are used to compute weight matrices for attention on the edges. These attentions create a meta path for messages passing across layers. The relative temporal encoding (RTE) helps the heterogeneous graph understand temporal dependencies. The architecture from their paper is shown in Figure 1. There are three main modules in the transformer model: Heterogeneous Mutual Attention, Heterogeneous Message Passing, and Target Specific Aggregation. The HGT can capture complex and time-dependent relations among entities. The transformer-attention architecture incorporates high-order heterogeneous neighbor information, which automatically learns the importance of implicit meta paths.

III. Literature Review

Graph Neural Networks (GNN) [16]–[18] have become dominant for graph embedding and graph mining. Some GNN applications including using convolution neural networks [4], [19] and the relational graph convolutional networks (RGCN) [9] to model knowledge graphs. The addition of attention to GNNs led to interesting inventions like Graph Attention Network (GAT) [15] These allow the processing of large graphs and motivate the model to converge on the important graph data. Heterogeneous Graph Attention Network by Wang et. al [10] is the extension of GAT where weights are varied depending on the meta path [20]. The authors used high-level semantic attention to separate and add information from the meta paths. The gap in this research is the absence of capturing common and specific patterns of different relationships using equal or even fewer parameters. Hu et. al [13] proposed to use an attention-based modified transformer model that can incorporate high-order graph data and can encode temporal features using Relative Temporal Encoding (RTE). Our work is takes inspired by this invention, and our aim is to introduce a functional improvement to their proposed methodology, making it a more robust system. As best of our knowledge from an extensive literature review, there was no similar approach taken.

IV. Proposal: Redefining F

Felfeli et. al [12] addressed the problem of influence maximization which states that every node in a graph can be thought as interacting particles. These interactions are coined as Forces and are expressed by Equation 1.

	Nodes						Edges										
	Paper	**Author**	**Venue**	**Affiliation**	**Field**	**Total**	**p-p**	**p-a**	**p-v**	**p-f**	**a-p**	**a-aff**	**v-p**	**aff-a**	**f-f**	**f-p**	**Total**
NN	18911	32307	2008	3445	9540	66211	14788	14052	18911	79570	36667	32307	2117	3445	9181	9540	220578
ML	90012	109423	3226	5455	19028	227144	78392	72136	90012	377698	136712	109423	3492	5455	18523	19028	910871
CS	544244	510189	6934	9079	45717	1116163	510306	1183804	544244	2360646	678501	510189	7716	9079	45653	45717	3765855

TABLE I: Node and Edge counts in NN, ML and CS datasets. Here p-p, p-a, p-v, p-f, a-p, a-aff, v-p, aff-a, f-f, and f-p denote Paper-Paper, Paper-Author, Paper-Venue, Paper-Field, Author-Paper, Author-Affiliation, Venue-Paper, Affiliation-Author, Field-Field, and Field-Paper edges respectively.

$$F = k(i)k(j)/[r(i,j)^2] \tag{1}$$

where where 'i' and 'j' represents two homogeneous nodes and 'r' represents the randomized shortest path between those two nodes. 'k' represents the degree of each node. The authors delve further into the cost of graph traversal depending on the Forces of each node. However, the scope of this research is based on Equation 1 and modifying it to our objective.

As discussed in Section II D, a Heterogeneous graph 'G' can have source nodes 's', and target nodes 't'. It inherits all the properties of a graph with the addition of having multi-type node interactions, e = ¡s,t¿. These nodes are embedded and fed to the transformer model shown previously in Figure 1. Our proposal is to modify the force 'F' and redefine it according to Equation 2.

$$F_{s,t}[HGT] = \sigma\Sigma k(s)k(t)/(||emb_s|| - ||emb_t||)^2 \tag{2}$$

where 's' and 't' represents source and target(which can be another source). k(s) and k(t) represent the number of edges of the source and target nodes. After multiplying them we divide the value by the square of the difference in the norms of their respective embedding 'emb'. We pass the resultant force through a sigmoid function to scale the force between 0 and 1.

The Heterogeneous Mutual Attention and Heterogeneous Message passing module consist of two important functions: Mutual attention heads and multi-head message. Each with its own weights, 'W' generated within the functions. The mutual attention constructs its overall weights from the Key-Query pair of the source and target nodes, whereas the message function leverages all source node values. Our goal is to add the Force, *'F'* to each of the weights of those corresponding functions as described in Equations 3 and 4. The addition is pictured in Figure 2

$$MutualAtt = (K^i(s) \bullet W \bullet Q^i(t)) \bullet \frac{\mu\tau(s), \phi(e), \tau(t)}{\sqrt{d}} \oplus \mathbf{F} \tag{3}$$

$$Message = V^i_{\tau(s)} \bullet W \oplus \mathbf{F} \tag{4}$$

Equations exclusive of *'F'* are detailed by Hu et. al [13] in Equation 3 and 4 of their research paper. It is explicitly to be noted that *'F'* is added to each of the weights of the W-matrix, i.e., pointwise addition, since *'F'* is a scalar value. Therefore pointwise addition will normalize the weight distribution of the overall outcome. Addition of Force at this point makes sense that *'F'* itself can be thought of as a weight, or a bias towards more potential nodes, i.e, nodes with greater interaction hold more value in contributing to the neighborhood and thus can help in particular classification tasks and reduce graph path traversal costs.

V. Experiments

This section describes our experimental setup and procedure that we used to evaluate the performance of our added force function to the HGT model. We employ the Open Academic Graph (OAG) dataset as our experimental foundation for this evaluation.

A. Experimental Dataset

We use the 2 variant adaptations of the standard OAG dataset, namely NN and ML for evaluating our Force bias. NN is a subset of the original OAG dataset, consisting of more than 178 million nodes and 2.236 billion edges, is the collection of all the neural network-related papers ranging from 1900 to 2019. ML is also a subset of the largest publicly available heterogeneous academic dataset containing Machine Learning related papers from 1900 to 2019. The CS adaptation of the OAG dataset contains all the papers published in the Computer Science domain. However, for resource constraints, our experiments were confined within NN and ML adaptations.

There are five node types in the dataset: 'Paper,' 'Author,' 'Field,' 'Venue,' and 'Institute'. The 'Field' nodes are further classified into six levels, L0 through L5, and are arranged using a hierarchical tree. As a result, we distinguish the 'Paper–Field' edges that correspond to the field level. Furthermore, the dataset distinguishes between author orders (i.e., first author, last author, and others) and venue types (i.e., journal, conference, and preprint). Here the 'Self' edge type refers to the self-loop connection. Table I shows the node and edge distribution in the NN, ML and CS datasets.

B. Experimental setup

We train and compare 4 different models for this experiment. we train 2 models with the legacy HGT model as the authors (Hu et. al [13]) are yet to report their performance on the ML and NN variants of the OAG dataset. We train the other 2 models by adding the Force F with the weight matrices. We demonstrate our experimental procedure in Figure 3

We train the legacy and the proposed models with the same datasets, namely NN and ML, for 50 epochs using the AdamW optimizer [21] with a learning rate of 0.001.

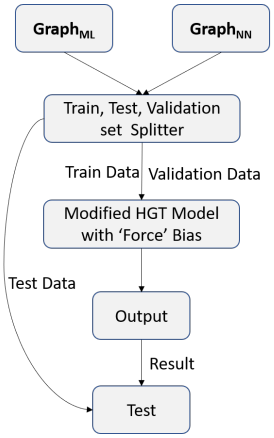


Fig. 3: Experiment procedure

VI. Result Analysis

In this section, we present our experimental results. Table II represents the performance of the HGT model after adding the Force bias. We use 2 different metrics to evaluate the model. Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR). NDCG is a ranking-based metric that is a popular method for measuring the quality of a set of search results. It can be defined as

$$NDCG_{pos} = \frac{DCG_{pos}}{idealDCG}$$

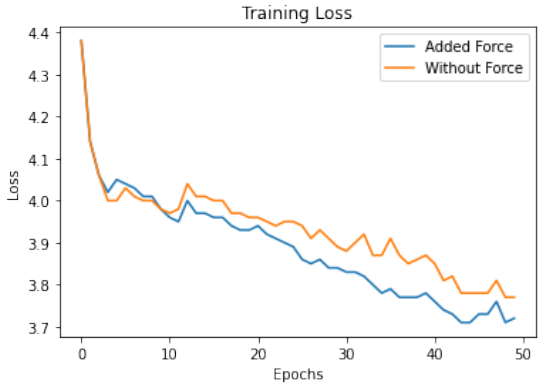
Where DCG is the Discounted cumulative gain computed with the relevance of the results. MRR is also a rank-aware evaluation metric. This technique places a strong emphasis on the first relevant element in the list. It is best suited for specific queries.

	Best Test NDCG	Best Test MRR
Legacy$_{NN}$	**0.5040**	**1.0000**
Proposed$_{NN}$	**0.5040**	**1.0000**
Legacy$_{ML}$	0.2754	0.2412
Proposed$_{ML}$	**0.2828**	**0.2600**

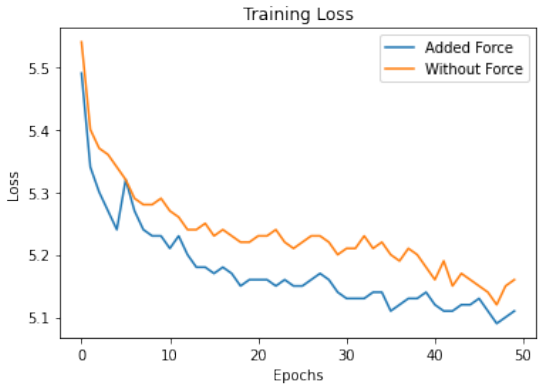
TABLE II: Experiment Results: Both metrics are considered as higher the better

According to Table II, after adding the Force bias, the proposed model outperforms its predecessor by 0.44% for the ML dataset on NDCG score and by 1.88% on the MRR score. They are on par with the legacy model according to when we experiment on the NN dataset. We inferred from this observation that on a bigger dataset, the Force bias has a bigger impact.

Figure 4 illustrates that when the Force bias is included, the training converges significantly quickly. Figure 4a depicts the HGT model's convergence rate on the NN dataset, whereas Figure 4b depicts the same for the ML dataset. During training, we compute the model's NDCG score with respect to a validation data split for each epoch. Figure 5 demonstrates the model's performance on the NDCG measure. The NDCG scores of the HGT model on the NN dataset for each epoch



(a) NN Data



(b) ML Data

Fig. 4: Training Loss Convergence

are shown in 5a, and the same for the ML dataset in 5b. While training the model for both datasets, we can see that the suggested model has a better NDCG score on the validation dataset.

VII. Conclusion

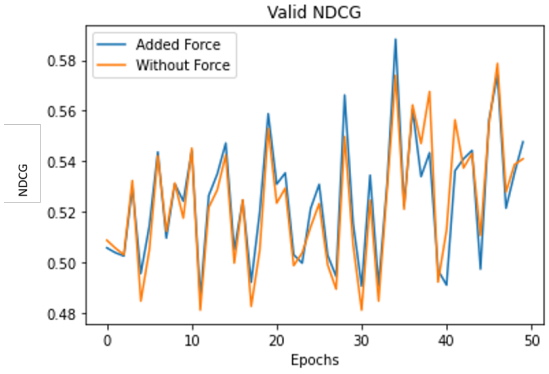
In this research, we aimed to enhance the Heterogeneous Graph Transformer(HGT) framework proposed by Hu et. al [13] by introducing the physical concept of Coulomb's Law which is used by Felfeli et. al in [12]. Our modification or re-definition of force helps the model gain 0.44% better accuracy on the NDCG metric and 1.88% on the MRR score for the ML data. It also helps the training to converge significantly faster. Though we train and test the enhanced model on only one type of heterogeneous dataset (OAG Dataset), it has a bigger scope when it comes to deep learning.

In the future, we aim to use the enhanced HGT model on other various dataset and introduce the cost of path traversal that can optimize search problems even further.

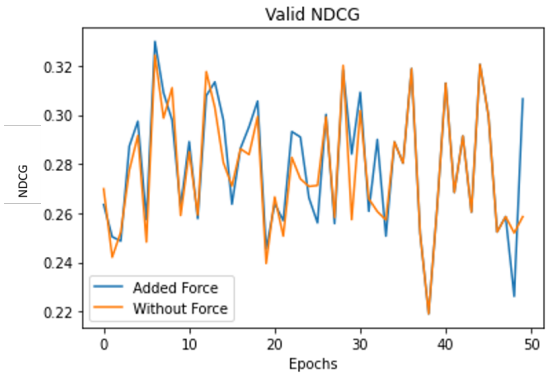
References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[2] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.



(a) NN Data



(b) ML Data

Fig. 5: Validation NDCG scoring

[3] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5453–5462.

[4] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[5] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2018.

[6] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.

[7] Y. Sun and J. Han, "Mining heterogeneous information networks: principles and methodologies," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 1–159, 2012.

[8] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, "The anatomy of the facebook social graph," *arXiv preprint arXiv:1111.4503*, 2011.

[9] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European semantic web conference*. Springer, 2018, pp. 593–607.

[10] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The world wide web conference*, 2019, pp. 2022–2032.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] Z. Felfli, R. George, K. Shujaee, and M. Kerwat, "Potential-driven model for influence maximization in social networks," *IEEE Access*, vol. 8, pp. 189 786–189 795, 2020.

[13] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of The Web Conference 2020*, 2020, pp. 2704–2710.

[14] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[16] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE international joint conference on neural networks*, vol. 2, no. 2005, 2005, pp. 729–734.

[17] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[18] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[19] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[20] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[21] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.