Drug Induced Liver Injury Prediction with Injective Molecular Transformer

Geonyeong Choi[1] Hyo Jung Cho[2] Soon Sun Kim[2] Ji Eun Han[2] Jae Youn Cheong[2] Charmgil Hong[1]

***Abstract*— Drug-Induced Liver Injury (DILI), liver damage caused by drugs, represents a significant factor contributing to the failure of clinical trials. Remarkably, the drug development process, which entails an extensive timeline spanning several years and incurring costs of billions of dollars to achieve Food and Drug Administration (FDA) approval, could greatly benefit from early DILI prediction. Furthermore, through the utilization of DILI prediction, clinicians can obtain valuable insights into the potential risks associated with medication, empowering them to make more informed decisions when prescribing drugs to patients. We employ Graph Neural Networks (GNNs) to predict DILI based on drug structures. GNNs consist of node aggregation, which gathers node representations, and graph pooling, which compiles node representations to portray the graph as a single vector. While the graph pooling method built on Set Transformer outperforms existing techniques, we identify a limitation: Set Transformer uses a random seed vector as the query vector that cannot differentiate between graphs of varied structures. Moreover, it potentially lacks expressiveness, as it is randomly defined without prior knowledge and relies on a limited number of seed vectors. To overcome the issues, we introduce Molecular Transformer that employs unique molecular representations as the query vectors. We find that using drug toxicity information extracted from relevant knowledge-bases as the query vector yields the best performance.**

I. INTRODUCTION

Drug-Induced Liver Injury (DILI), damage or injury to the liver caused by medications or drugs, stands as a significant cause of clinical trial failures [4]. Notably, the drug development process, which involves an extensive time frame of approximately 8.3 years and costs around 1.3 billion dollars to bring a drug to FDA approval [18], could greatly benefit from an early prediction of DILI. Furthermore, by utilizing DILI prediction, clinicians can gain valuable insights into the potential risks associated with a medication, enabling them to make more informed decisions when prescribing drugs to patients. This approach empowers clinicians to assess the hepatotoxicity risk of drugs, allowing for more cautious and personalized medication prescriptions. By considering the DILI potential, clinicians can optimize patient safety and tailor treatment plans, selecting alternative drugs or adjusting dosages to minimize the risk of liver injury.

In order to predict DILI risk based on drug structures, we utilize Graph Neural Networks (GNNs) [6]. Demonstrating remarkable performance and potential in a variety of drug molecule-related tasks, including molecular generation and property prediction [12], GNNs create molecular representations based on molecular structural information.

In this paper, we introduce the novel Transformer-based graph pooling technique, called *Molecular Transformer* (M-Transformer), which employs molecular representations as the query vector and utilizes the attention mechanism [16]. More specifically, M-Transformer generates the query vector using drug toxicity knowledge-bases and effectively addresses limitations of previous approaches that rely on a seed vector as the query vector [8].

In Section 2, we review key principles that underlie GNNs and discuss methods relevant to our work. Section 3 sheds light on shortcomings of existing methods and introduces our novel approach. Finally, in Section 4, we compare the performance of our proposed methods with that of existing methods and demonstrate the superiority of our strategy.

II. RELATED WORK

A. Drug-Induced Liver Injury Severity and Toxicity (DILIst)

The Drug-Induced Liver Injury Severity and Toxicity (DILIst) [15] knowledge-base comprises 1,279 drugs, each labeled with its potential to induce DILI. By augmenting precedent knowledge-base DILIrank [3] with other literature featuring at least 350 drugs characterized with human DILI, DILIst compiles by far the largest assembled list of drugs with DILI classification. DILIst contains 768 drugs classified as DILI-Positive and 511 drugs classified as DILI-Negative.

B. Graph Neural Networks

GNNs are a subclass of deep learning models specifically designed to handle graph-structured data such as molecules. Suppose a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with node and edge sets $\mathcal{V}$ and $\mathcal{E}$. Each node $v \in \mathcal{V}$ has the node representation $x_v \in X$ (X denotes the node feature matrix), and each edge $\mathfrak{e}_{vu} \in \mathcal{E}$ connecting nodes v and u has a edge feature $e_{vu} \in E$ (E denotes the edge feature matrix). GNNs primarily consist of two key components: *node aggregation* that updates node representations, and *graph pooling* that formulates the graph representation by combining node representations.

Node aggregation The node aggregation involves updating node representations by aggregating node representations of neighboring nodes. In this paper, we employ the node aggregation approach adopted by Graph Isomorphism Networks (GINs) [19], which effectively addresses the limited expressiveness of preceding methods [6] by leveraging the

[1]Department of Computer Science and Electrical Engineering, Handong Global University, Pohang, Korea, {gychoi and charmgil}@handong.ac.kr

[2]Department of Gastroenterology, Ajou University School of Medicine, Suwon, Korea {pilgrim8107, soonsunkim, 110518, and jaeyoun620}@aumc.ac.kr

summation of information derived from neighboring nodes. As a result, the node representation $h_v^{(k)}$ formulated in the node aggregation at the k-th layer as follows:

$$h_v^{(l+1)} = \left(h_v^{(l)} + \sum_{u \in \mathcal{N}(v)} \left(h_u^{(l)} + e_{vu} \cdot W_1^{(l)} \right) \right) \cdot W_2^{(l)} \quad (1)$$

where $\mathcal{N}(v)$ represents the set of neighboring nodes of v, $h_v^{(0)} = x_v$ and W is a weight matrix.

Graph Pooling Through the graph pooling, node representations $H^{(K)} \in \mathbb{R}^{N \times d}$ are combined to generate the graph representation $GraphPool(H^{(K)}) = h_g \in \mathbb{R}^d$. Graph pooling necessitates the preservation of specific properties like permutation invariance [8], ensuring output consistency irrespective of input order, and injectiveness [19], the capacity to differentiate among graphs with varied structures. Different approaches to the graph pooling exist, including adding all node representations, known as the sum pooling [19]. However, the sum pooling has a downside of not considering the relative importance of individual nodes [2].

C. Set Transformer

Another approach for the graph pooling is utilizing Transformer [16]. The self-attention mechanism of Transformer involves matrix multiplication of $query\ Q \in \mathbb{R}^{N \times d}$, $key\ K \in \mathbb{R}^{N \times d}$, and $value\ V \in \mathbb{R}^{N \times d}$ ($Att(Q,K,V) = Softmax(QK^T)V$) where K and V are node representations $H^{(K)}$. Set Transformer uses the query matrix with the randomly defined learnable parameter seed vector $Q \in \mathbb{R}^{1 \times d}$.

$$Set\ Transformer(Q,K,V) = LayerNorm(P + rFF(P)) \quad (2)$$

$$\text{where } P = LayerNorm(Q + Att(Q,K,V))$$

where *LayerNorm* and *rFF* denote Layer Normalization [1] and row-wise FeedForward neural networks, respectively.

III. Our Work

A. Motivation: Limitations of Set Transformer

While Set Transformer has been successfully utilized in the graph pooling and has demonstrated good performances in various graph-related tasks [2], we have discerned inherent limitations in this approach.

Injectiveness The attention coefficient results from the matrix multiplication and $Softmax$ function between the seed vector and node representations, $Softmax(QK^T)$. Denoting the attention vector produced by QK^T as $\alpha \in \mathbb{R}^N$, the $Softmax$ operation can be rewritten as: $Softmax(\alpha_i) = \frac{e^{\alpha_i}}{\sum_{j=1}^{N} e^{\alpha_j}}$. Now let $e^{\alpha_i} = z_i$, and $Softmax(\alpha_1)$ can be computed as $\frac{z_1}{(z_1+z_2+\ldots+z_N)}$. If z_1 doubles while maintaining the node proportion, the coefficient for z_1 becomes $\frac{2z_1}{2(z_1+z_2+\ldots+z_N)} = \frac{z_1}{(z_1+z_2+\ldots+z_N)}$. This shows that, even with an increase in the number of nodes in a graph, if the graph maintains a constant node proportion, the overall attention coefficient for distinct node representations remains unchanged (Figure 1).

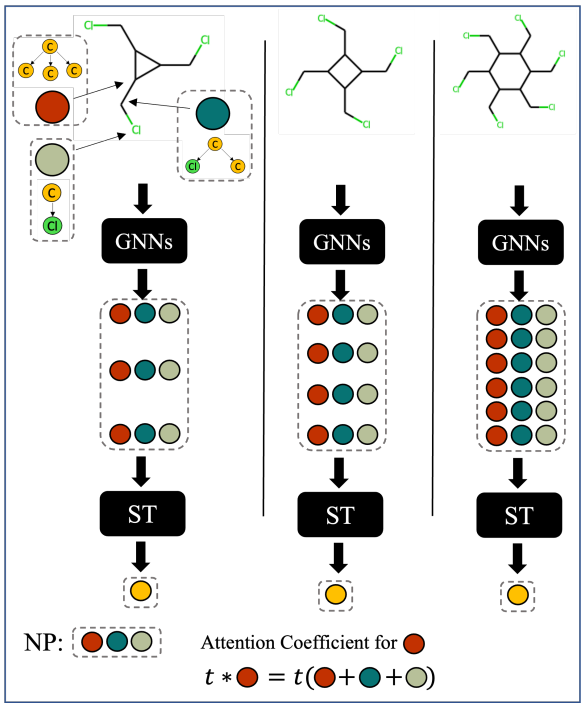

Fig. 1. Cases where graph representations produced by Set Transformer are identical due to same node proportion. ST and NP stands for Set Transformer and the node proportion, respectively.

Randomness It has been observed that initializing a model with prior knowledge, rather than randomly, results in better performance [7]. However, in the case of Set Transformer, the randomly initialized seed vector used as *query*, which is defined independently of the input graph or related tasks.

Diversity DILIst comprises more than 1,200 molecules, each with an average of 56 atoms. The node representations produced via node aggregations in GNNs process encode information pertaining to the subgraphs of each molecule. When Set Transformer using a seed vector carries out attention over node representations, it has to process more than 60k subgraphs through a single seed vector.

B. Proposed Method: M-Transformer

Addressing the limitations of Set Transformer identified in the previous subsection, we propose M-Transformer. Rather than utilizing a seed vector with random initial values, M-Transformer employs the graph representation as $query$.

Unlike Set Transformer, M-Transformer can distinguish graphs with different structures by using unique graph representation. Assuming two graphs G_1 and G_2 with distinct structures, in Set Transformer Q_1 and Q_2 used as *queries* are identical in G_1 and G_2 ($Q_1 = Q_2$). However, in M-Transformer, *queries* Q_1 and Q_2 differ across the two graphs ($Q_1 \neq Q_2$). As a result, even if the vector $h_t = Att(Q,K,V)$ is the same in two graphs, the self-loop in the M-Transformer could lead to two distinct graph representations ($Q_1 + h_t \neq Q_2 + h_t$). Hence, M-Transformer embodies the injective graph pooling function, enabling differentiation between two graphs with disparate structures.

TABLE I
SET TRANSFORMER VS M-TRANSFORMER

	Set Transformer	**Molecular Transformer**
Injectiveness	X	O
Initialization of the Query Vector	Random	Molecular features
Diversity of the Query Vector	Maximum K	# of molecules (N)

Furthermore, M-Transformer uses the graph representation as $query$, providing the molecule-specific representation different from that of a seed vector with random initial values. Lastly, rather than relying on a single seed vector, M-Transformer utilizes molecular representations that are uniquely created for each graph. As a result, one seed vector no longer performs attention on all the atoms of every molecule. Instead, the graph representations generated by N graphs perform attention on each atom. Table I summarizes differences between Set Transformer and M-Transformer.

C. Query for M-Transformer

With the vanilla M-Transformer, *query* is the molecular representation generated through the sum pooling, the injective graph pooling function [19]. However, the molecular representation from the sum pooling has a drawback, in that it relies on the optimization of node representations during the initial stage of training. To overcome this issue, we adopt and use the molecule- or drug-level features extracted from two relevant knowledge-bases as $queries$.

Drug Toxicitiy knowledge-bases We assume that drug toxicity correlates with DILI and can greatly assist in identifying drug structures associated with DILI. To extract toxicity information of drugs comprising DILIst, we utilized PubChem BioAssay (PCBA) knowledge-bases [17], which contains 128 bioassays measured across 400k compounds, and ToxCast knowledge-bases [11], containing qualitative results from over 600 experiments conducted on 8k compounds. We trained separate GNNs on each label using these knowledge-bases and used the predictions as *query*.

ATC code The Anatomical Therapeutic Chemical Classification System (ATC) code [10] categorizes drugs based on their mode of action and chemical properties. Given that the manifestations of DILI can vary depending on the usage of the drug, employing ATC codes as $query$ can effectively encode specific parts of a drug associated with DILI. We converted the individual levels of the ATC code into one-hot encodings and applied them to fully connected neural networks in sequential order, from level 1 to level 4.

IV. EXPERIMENTS

Among the 1,279 drugs in DILIst, we excluded those that are not small molecules, such as antibodies, and those without an ATC code. After these exclusions, we performed predictions on the remaining 1,002 drugs.

TABLE II
PERFORMANCES OF THE MODELS FOR DILI PREDICTIONS

	$Seed$	SUM	ATC	$Drug\_Tox$
$Only$	-	0.667 (0.04)	0.676 (0.04)	0.645 (0.04)
Cat	-	-	0.691 (0.03)	0.676 (0.03)
Att	0.547 (0.04)	0.675 (0.05)	**0.693 (0.06)**	**0.695 (0.06)**

A. Setup

For constructions of GNNs, we used PyTorch Geometric (PyG) [5]. Hyperparameters of GNNs include the number of GNN layers $\in \{4,20\}$, batch size $\in \{64,128,256\}$, activation function $\in \{$ReLU, GELU$\}$, 256 hidden units, learning rate $\in \{$1e-04, 1e-05, 1e-06$\}$, and epochs $= 100$. Fully Connected Neural Networks to predict DILI with graph representations consist of 4 layers. Atomic number and chirality are used as atomic features, and bond type is used as bond features. We used the ADAM optimizer [14], binary cross-entropy loss, and the Area Under the Receiver Operating Characteristic (AUROC) as an evaluation metric. We evaluated the performance on the test set where the validation loss was minimized. We conducted 5-fold cross-validation and recorded the average value and standard deviation.

B. Results of the DILIst Prediction

Table II presents the performance of the compared methods in our experiment. As a baseline, we recorded DILI prediction outcomes using the graph representation derived from the sum pooling, ATC code, and drug toxicity information, as shown in the first row ($Only$). We also concatenated graph representations (Cat) from the sum pooling and ATC codes (or drug toxicity) to compare the effectiveness of Transformer (Att). In terms of the columns on Table II, *Seed*, *SUM*, *ATC*, and *Drug_Tox* indicate a seed vector, the sum pooling, ATC codes, drug toxicity, respectively.

In our experimental evaluations, we compared the effectiveness of using molecular representations against a seed vector as $query$ in Transformer. The results clearly indicated that employing the molecular representation as $query$ could result in a performance enhancement. First, by observing the improved performance when using graph representations produced by the sum pooling as $query$ in M-Transformer, we demonstrated that adjusting node weights to focus on specific parts of the graph is more effective than the sum pooling that equally weights all nodes.

Additionally, we concatenated drug toxicity (or ATC code) into the graph representation derived from the sum pooling and observed a performance improvement. This confirms that augmenting the graph representations obtained from GNNs with supplementary information positively impacts model decisions, thereby improving prediction accuracy.

We further compared the results of employing a simple concatenation approach against using M-Transformer.

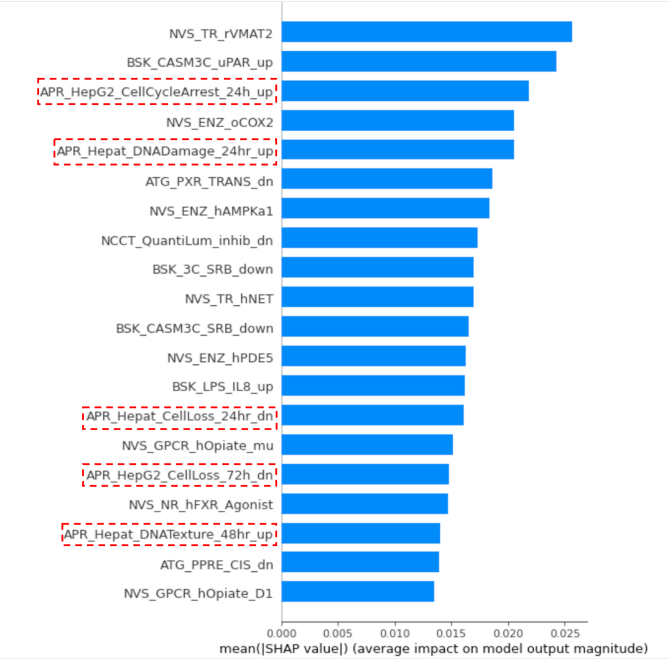

Fig. 2. Shapley value measurement results for drug toxicity information

We observed an improvement in performance when M-Transformer was employed, suggesting that the structural advantages of M-Transformer surpass those of concatenation methods. Finally, we found that using drug toxicity information as the $query$ in M-Transformer resulted in the highest performance. This finding underscores the critical role that drug toxicity information plays in identifying the substructures related to DILI in pharmaceutical compounds.

C. Shapley value analysis for DILI prediction results

Upon identifying that drug toxicity information is the most effective as $query$, we evaluated a DILI prediction model that only uses drug toxicity information to analyze the relationship between drug toxicity and DILI. We used Shapley Value [13] to assess the importance of variables and observe their influence on model decision-making. Interestingly, we found that among the 20 most influential variables, five variables were directly tied to hepatotoxicity. Furthermore, two of the top five variables specifically pertained to hepatotoxicity. This discovery not only highlights critical roles that liver-related toxicity plays in identifying the substructures of drugs related to DILI but also justifies that the model generated through our work effectively makes accurate judgments.

Surprisingly, factors considered unrelated to hepatotoxicity appeared to play an important role in determining hepatotoxicity. For instance, one of the highly ranked variables, "NVS_TR_rVMAT2", is a protein in the nervous system involved in regulating the release of monoamine neurotransmitters and dopamine [9]. Given that the mechanisms underlying most DILI cases remain elusive [15], our findings suggest that examining such influential factors in drug toxicity could provide a new approach for future research.

V. Conclusion

In this study, we have successfully demonstrated that the incorporation of drug toxicity information within the Transformer architecture significantly enhances the accuracy of predictions compared to previous methods. We believe that our research represents a significant step forward in the quest for improved pharmaceutical safety and efficacy. We expect future work that builds on our findings will continue to drive advancements in the field.

ACKNOWLEDGMENT

This work was supported by the National IT Industry Promotion Agency (NIPA) grant funded by the Korea government (MSIT) - No.S0252-21-1001, Development of AI Precision Medical Solution (Doctor Answer 2.0).

References

[1] Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
[2] Baek, J., Kang, M., Hwang, S.J.: Accurate learning of graph representations with graph multiset pooling. arXiv preprint arXiv:2102.11533 (2021)
[3] Chen, M., Suzuki, A., Thakkar, S., Yu, K., Hu, C., Tong, W.: Dilirank: the largest reference drug list ranked by the risk for developing drug-induced liver injury in humans. Drug Discov Today **21**(4) (2016)
[4] Chen, M., Vijay, V., Shi, Q., Liu, Z., Fang, H., Tong, W.: Fda-approved drug labeling for the study of drug-induced liver injury. Drug discovery today **16**(15-16), 697–703 (2011)
[5] Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428 (2019)
[6] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
[7] Kumar, S.K.: On weight initialization in deep neural networks. arXiv preprint arXiv:1704.08863 (2017)
[8] Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., Teh, Y.W.: Set transformer: A framework for attention-based permutation-invariant neural networks. In: International conference on machine learning. pp. 3744–3753. PMLR (2019)
[9] Miller, G.W., Gainetdinov, R.R., Levey, A.I., Caron, M.G.: Dopamine transporters and neuronal injury. Trends in pharmacological sciences **20**(10), 424–429 (1999)
[10] Miller, G., Britt, H.: A new drug classification for computer systems: the atc extension code. International journal of bio-medical computing **40**(2), 121–124 (1995)
[11] Richard, A.M., Judson, R.S., Houck, K.A., Grulke, C.M., Volarath, P., Thillainadarajah, I., Yang, C., Rathman, J., Martin, M.T., Wambaugh, J.F., et al.: Toxcast chemical landscape: paving the road to 21st century toxicology. Chemical research in toxicology **29**(8), 1225–1251 (2016)
[12] Sun, M., Zhao, S., Gilvary, C., Elemento, O., Zhou, J., Wang, F.: Graph convolutional networks for computational drug development and discovery. Briefings in bioinformatics **21**(3), 919–935 (2020)
[13] Sundararajan, M., Najmi, A.: The many shapley values for model explanation. In: International conference on machine learning. pp. 9269–9278. PMLR (2020)
[14] Tato, A., Nkambou, R.: Improving adam optimizer (2018)
[15] Thakkar, S., Li, T., Liu, Z., Wu, L., Roberts, R., Tong, W.: Drug-induced liver injury severity and toxicity (dilist): binary classification of 1279 drugs by human hepatotoxicity. Drug discovery today (2020)
[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
[17] Wang, Y., Bryant, S.H., Cheng, T., Wang, J., Gindulyte, A., Shoemaker, B.A., Thiessen, P.A., He, S., Zhang, J.: Pubchem bioassay: 2017 update. Nucleic acids research **45**(D1), D955–D963 (2017)
[18] Wouters, O.J., McKee, M., Luyten, J.: Estimated research and development investment needed to bring a new medicine to market, 2009-2018. Jama **323**(9), 844–853 (2020)
[19] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)