MODEL SUMMARY

A1.Background on our team

Competition Name: **NeurIPS - Open Polymer Prediction 2025**
Team Name:Ghy HUST CS
Private Leaderboard Score:0.078
Private Leaderboard Place:3rd

Name:hongyu Guo
Location:Wuhan Hubei China
Email:2211460074@qq.com

Name:youheng Yan
Location:Shenyang Liaoning China
Email:3411797175@qq.com

A2. Background on our team

Guo Hongyu

What is your academic/professional background?

Hongyu Guo is a sophomore student at the School of Computer Science and Technology, Huazhong University of Science and Technology.

Do you have any prior experience that helped you succeed in this competition?

Hongyu Guo has systematically self-studied machine learning and deep learning. He is able to use PyTorch to build neural networks, understands model evaluation metrics, and is familiar with model tuning strategies.

What made you decide to enter this competition?

After a year of study, Hongyu Guo wanted to test his learning achievements. At the same time, he has long admired Kaggle's influence in the field of artificial intelligence.

How long did you spend on the competition?

We spent about 2 months on the competition.

If you competed as part of a team, how did you decide to team up?

As the team organizer, Hongyu Guo decided to form a team mainly because he is interested in machine learning and willing to get his hands dirty.

If you competed as part of a team, who did what?

Guo Hongyu was responsible for data preparation, data cleaning, algorithm design, code implementation, model tuning, and ablation studies.

Yan Youheng

What is your academic/professional background?

Youheng Yan is a sophomore student at the School of Software, Northeastern University.

Do you have any prior experience that helped you succeed in this competition?

Youheng Yan has no relevant experience.

What made you decide to enter this competition?

Youheng Yan entered the competition at the encouragement of Hongyu Guo.

How long did you spend on the competition?

We spent about 2 weeks on the competition.

If you competed as part of a team, how did you decide to team up?

As a team member, Youheng Yan participated mainly out of interest and for the purpose of learning.

If you competed as part of a team, who did what?

Yan Youheng was responsible for data mining.

A3. Summary

We tried GNN models,we use GATv2Conv model with 6 layers 384 hidden heads and 8 attention heads.Also,we merged 50 Mogan features in 384 hidden heads.The 50 Mogan features was selected by F regression.

The most important features were selected by GNN.The other features which used in my ML models,like rdkit.descriptor,MACSS or the features lesected by ChemBert were all useless except FP features.

We used Pytorch to build our GNN model,and we used 5-fold CV to train it.It takes us about 10 hours to train our GNN model.

A4. Features Selection / Engineering

What were the most important features?

As our core model is a Graph Neural Network (GNN), the features can be categorized into two levels:

Intrinsic Graph Features: This is the most critical feature set. We converted each polymer's SMILES string into a graph structure. The features for the nodes in the graph include: atomic number, degree, formal charge, number of radical electrons, hybridization, whether it's aromatic, and the total number of connected hydrogen atoms. The features for the edges include bond type and whether the bond is conjugated. The GNN model learns directly from these graph structures, automatically extracting and combining high-dimensional structural features.

Auxiliary Features (Morgan Fingerprints): To enhance the model's expressive power, we calculated 1024-bit Morgan Fingerprints as supplementary features. Instead of generating a single variable importance plot for all 1024 bits, we selected the most relevant bits independently for each prediction task (Tg, FFV, Tc, Density, Rg). For each task, the most important features are the top 50 Morgan fingerprint bits selected based on their f-regression scores. These selected fingerprint bits are then concatenated with the learned graph embedding from the GNN for the final prediction.

How did you select features?

Our feature selection process was specifically applied to the Morgan Fingerprints:

We utilized the SelectKBest method from the Scikit-learn library.

The scoring function used was f_regression, which evaluates the linear relationship between each feature and the continuous target variables.

This selection process was run independently for each of the five target variables, allowing us to identify the 50 most predictive fingerprint bits for each specific task.

The node and edge features for the GNN were predefined based on chemical properties. The GNN itself learns their importance during training, so no manual feature selection was performed on them.

Did you make any important feature transformations?

Yes, we performed two key feature transformations:

SMILES-to-Graph Conversion: The most fundamental transformation was converting the one-dimensional SMILES strings into information-rich graph objects. These graphs contain atoms as nodes, chemical bonds as edges, and their respective chemical attributes, which is the required input format for our GNN model.

Chemical Data Augmentation: We implemented an important function which named augment_repeat_units. This function expands a SMILES string representing a monomer (e.g., *C-C*) into an oligomer chain containing multiple repeat units (e.g., *C-C-C-C-C-C*). This effectively created new data samples that are structurally closer to real polymers, augmenting our training set.

Did you find any interesting interactions between features?

Yes. Our model architecture was specifically designed to discover such interactions. In the final prediction head of the model, we fused features from two sources:

Global Graph Embedding: This vector is generated by the GATv2 network after processing the entire molecular graph and represents the polymer's overall topology and chemical structure.

Local Fingerprint Features: These are the pre-selected 50 Morgan fingerprint bits that are most relevant to a specific task.

By concatenating these two feature sets, the model is forced to learn how to combine macroscopic, global structural information with microscopic information represented by specific chemical sub-structures (the fingerprints). This interaction was crucial for improving prediction accuracy across the different tasks.

Did you use external data? (if permitted)

Yes, we did not use any external data. We find many external data but at last,we found only about 7000 Tg data were useful.

A5. Training Method(s)

What training methods did you use?

We employed a multi-task, multi-stage training methodology based on Graph Neural Networks:

Model Architecture: The core of our solution is a **Graph Attention Network v2 (GATv2)** with residual connections. The model was designed to predict all five target properties simultaneously.

Training Framework: We used 5-fold Cross-Validation to train and evaluate our model. This ensures robustness and provides a more reliable performance estimate.

Loss Function: We implemented and utilized a custom Weighted Mean Absolute Error (wMAE) Loss function. This loss function is identical to the official competition metric, ensuring that the model training was directly optimizing for the final score.

Two-Stage Training Process: Within each fold, we first trained the model on the training set and used an **Early Stopping** strategy on the validation set to find the optimal number of epochs. Subsequently, we merged the training and validation sets of that fold and re-trained a new model from scratch on this combined data for the exact number of epochs determined earlier. This ensures the final model for each fold leverages all available information.

Output Calibration: After training in each fold, we also trained a simple Linear Regression model for each prediction task to calibrate the raw outputs of the GNN, aiming to reduce systematic bias.

Did you ensemble the models?

Yes. Our final submission is an ensemble of the 5 models generated from the 5-fold cross-validation.

If you did ensemble, how did you weight the different models?

We used a straightforward and robust simple averaging approach. The final prediction is the arithmetic mean of the predictions from the five models. Each model was given an equal weight (i.e., 20%).

A6. Interesting findings

What was the most important trick you used?

We believe our most important strategy was the synergistic combination of two techniques: Chemical Data Augmentation and a Task-Oriented Hybrid Feature Engineering approach.

Chemical Data Augmentation (Repeat Unit Expansion): Before training, we programmatically "expanded" the SMILES strings that represented polymer monomers. For instance, a monomer like *A* was chained together to form an oligomer like *A-A-A*. This simple procedure had a profound impact: it not only significantly augmented the size of our training set but, more crucially, it generated training samples that were structurally more representative of real-world long-chain polymers. This allowed our Graph Neural Network (GNN) to learn how repeating units connect and influence the overall structure, rather than just seeing isolated monomers, leading to a more generalizable feature representation.

Task-Oriented Hybrid Feature Engineering: We did not rely solely on the end-to-end learning capabilities of the GNN. Recognizing that different physical properties (e.g., Tg vs. Rg) likely depend on different chemical sub-structures, we built a hybrid model:

GNN Component: Responsible for learning the polymer's global topology and overall structural features, generating a "graph embedding."

Fingerprint Component: For each prediction target (Tg, FFV, etc.), we **independently** used f-regression to select the top 50 most important Morgan fingerprint bits.

A7. Model Execution Time

How long did it take to train the model?

The entire 5-fold cross-validation training process took approximately 10 hours to complete on a single NVIDIA RTX 4070 ti super GPU.

This total time includes both training stages for each of the 5 folds:

1. Training with early stopping on a validation set to determine the optimal number of epochs.
2. Re-training the model on the full data for that fold for the optimal number of epochs.

How long did it take to generate predictions with the model?

Generating predictions for the entire test set was very efficient, taking approximately **30** seconds.

This time includes loading the 5 trained models from the cross-validation, preprocessing the test data (graph conversion and fingerprint calculation), and computing the final ensembled average predictions.

A8. References

NeurIPS - Open Polymer Prediction 2025 | Kaggle