Fine-Tuning Vision-Language Models for Multimodal Polymer Property Prediction

An Vuong[1] **Minh-Hao Van**[1] **Prateek Verma**[1] **Chen Zhao**[2] **Xintao Wu**[1]
[1]Department of EECS, University of Arkansas
[2]Department of CS, Baylor University
{anv,haovan,prateek,xintaowu}@uark.edu
Chen_Zhao@baylor.edu

Abstract

Vision-Language Models (VLMs) have shown strong performance in tasks like visual question answering and multimodal text generation, but their effectiveness in scientific domains such as materials science remains limited. While some machine learning methods have addressed specific challenges in this field, there is still a lack of foundation models designed for broad tasks like polymer property prediction using multimodal data. In this work, we present a multimodal polymer dataset to fine-tune VLMs through instruction-tuning pairs and assess the impact of multimodality on prediction performance. Our fine-tuned models, using LoRA, outperform unimodal and baseline approaches, demonstrating the benefits of multimodal learning. Additionally, this approach reduces the need to train separate models for different properties, lowering deployment and maintenance costs.

1 Introduction

Vision-Language Models (VLMs) have demonstrated exceptional capabilities in visio-linguistic tasks such as visual question answering (VQA), information extraction, and complex multimodal reasoning. A typical VLM consists of three main components: a vision encoder that extracts visual embeddings from input images, a large language model (LLM) that generates output tokens, and a multimodal projector that maps visual embeddings into a textual space processable by the language model. While both LLMs and VLMs have proven effective for reasoning tasks in general knowledge domains, applying them to specialized scientific areas, such as materials science, remains an open challenge.

In polymer research, SMILES (Simplified Molecular-Input Line-Entry System) Weininger [1], which is a text-based representation of molecular structures, has been extended by introducing asterisks (*) to mark the repeating units, referred to as polymer SMILES (P-SMILES) [2]. Recent studies have explored fine-tuning LLMs for property prediction [3] and combining LLM embeddings with conformational features [4]. Yet, these approaches are either unimodal or rely on separate regressors and therefore lack unified multimodal alignment. Earlier machine learning efforts also contributed to property prediction but often required training separate models for each task, creating fragmented pipelines. To overcome these limitations, we introduce a multimodal polymer dataset tailored for fine-tuning and evaluating VLMs, and investigate their ability to directly predict polymer properties. Specifically: **(1)** Our multimodal dataset is built from computational and experimental sources. Each sample contains a canonicalized P-SMILES, a 2D structure image, molecular descriptors, and property labels. The dataset is structured as VQA pairs, enabling VLMs to predict properties from images, P-SMILES, and molecular descriptors with improved reasoning. **(2)** We evaluate multimodal VLMs on polymer property prediction by fine-tuning Llama-3.2-11B-Vision-Instruct. Using instruction tuning with LoRA [5], the fine-tuned model achieves performance on par with machine learning and deep learning methods that typically require separate models for each property.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI4Mat.

2 Related Works

There have been numerous efforts to integrate AI into materials science research, particularly through the development of deep learning and foundation models for materials discovery and property prediction. This is especially relevant for polymers, where the cost of computational simulations or experimental measurements is often prohibitively high. Huan et al. [6] introduced a dataset of polymer properties, which laid the foundation for the Polymer Genome platform [7] designed to efficiently predict and retrieve polymer properties. Building on this, Doan Tran et al. [8] applied machine learning approaches trained on Polymer Genome data for property prediction. More recently, BERT-based models [9] have been adapted for polymers: Kuenneth and Ramprasad [2] developed PolyBERT, a large-scale representation model trained on millions of polymers, whose embeddings can serve as inputs for property predictors. Similarly, Wang et al. [10] proposed a Transformer-based architecture capable of extracting both 1D representations from P-SMILES and 3D representations from molecular conformations to perform multitask learning, including P-SMILES reconstruction, 3D coordinate generation, and cross-modal fusion. In line with recent trends in deep learning, large language models (LLMs) have also been explored for polymer property prediction [3, 4]. Specifically, Gupta et al. [3] fine-tuned and evaluated text-only LLMs on P-SMILES inputs, while Zhang and Yang [4] combined multimodal embeddings, such as LLM-derived representations, from P-SMILES and Uni-Mol [11] embeddings from polymer structures and then train multilayer perceptrons for property prediction. Despite these advances, a unified multimodal VLM capable of directly predicting properties from multimodal inputs such as images, text, and molecular descriptors remains lacking.

3 Multimodal Polymer Data Generation

3.1 Polymer SMILES Data

Kaggle Polymer Challenge (Kaggle). We collect the data from the Kaggle Open Polymer Prediction 2025 [12], including 7,973 P-SMILES with five properties: glass transition temperature (Tg), fractional free volume (FFV), thermal conductivity (Tc), density, and radius of gyration (Rg). Besides the main dataset, there are four supplementary datasets provided in this challenge, but we only use three of them: the first with 874 P-SMILES and Tc values, the third with 46 P-SMILES and Tg values, and the fourth with 862 P-SMILES and FFV values. The second supplementary dataset, having 7,174 P-SMILES without any property values, is not used in our study.

Data preprocessing. To integrate the supplementary datasets with the main Kaggle dataset, we first canonicalize P-SMILES strings in all datasets to obtain a unique representation for each polymer. We then remove duplicates from the supplementary datasets by checking against the main dataset. A polymer is considered a duplicate if it has the same canonical P-SMILES and the same property values as a polymer already included in the main dataset. Following this process, we sequentially merge the main dataset with the first, third and fourth datasets. Duplicates are removed if found.

As a result, the final dataset consists of 8,963 P-SMILES, each with a varying number of ground-truth properties. Table 3 in Appendix B summarizes the data statistics, including missing values across the five properties. Finally, we split the dataset into training and testing sets in a 90/10 ratio based on canonical P-SMILES, ensuring no polymer appears in both the training and test sets. The split results in 7,950 polymers for training and 1,013 for testing.

To evaluate model's performance on unknown dataset, we use **Glass Transition Temperature dataset (GTT)**, which contains 662 P-SMILES with Tg values introduced by Choi et al. [13]. We apply the same preprocessing steps to GTT and then compare it with the final dataset to filter out duplicates, removing 563 duplicates and retaining 99 polymers with Tg values.

3.2 Multimodal Features Generation

For both the training and testing sets, we generate a 2D image of each polymer from its canonical P-SMILES using RDKit [14], at a resolution of 1120×1120 pixels. We also compute 217 molecular descriptors from each P-SMILES with the RDKit Python package. Of these, 17 descriptors were selected by domain experts as meaningful features for LLMs to predict the five target properties.

Instruction prefix: You are a polymer expert.
Task: Given a polymer image `<image>`, its P-SMILES string `<P-SMILES>`, and its molecular descriptors: `<descriptor-1:value>`, `<descriptor-2:value>`, ..., `<descriptor-17:value>`.
Predict the `<property type>` of the polymer in `<unit>`.
Answer: `<property type>:<property value> <unit>`

Figure 1: An example of our instruction-tuning sample

3.3 Instruction-tuning Dataset

We construct an instruction-tuning dataset to predict one property type at a time. For polymers with multiple ground-truth properties, each data sample is decomposed into separate instruction-tuning samples that share the same canonical P-SMILES representation but are assigned different prompts, each requesting the prediction of a single property type with an available ground-truth value. After this decomposition, the training set expands to 9,097 samples and the testing set to 1,442 samples. Each sample is then structured as a question–answer pair, where the question corresponds to the prompt and the answer contains the ground-truth property value. The prompt is generated by randomly combining one of 20 instruction prefixes with one of 20 prediction templates, into which the canonicalized P-SMILES, the 2D image produced with RDKit, the 17 selected descriptors, and the specific property type to be predicted are inserted. The answer is standardized to a consistent format for each property type. Figure 1 shows an example of our instruction-tuning sample.

4 Experiments

4.1 Models

Our models. We fine-tune Llama-3.2-11B-Vision-Instruct (LVision) using the LoRA [5] technique with rank $= 16$ and $\alpha = 16$. The model is fine-tuned for 12 epochs with a learning rate of 0.0001 and a batch size of 8.

Baselines. To validate our approach, we compare the fine-tuned LVision against three baseline groups: (1) LLM-based models, (2) ML models using molecular descriptors, and (3) ML models using PolyBERT-derived representations [2]. For the LLM baselines, we fine-tune a text-only version of Llama-3.1-8B-Instruct (LText) using the same settings as LVision and also evaluate the original (non-fine-tuned) LVision and LText. For the descriptor-based ML group (+RDKit), we train Multi-layer Perceptron (MLP), Support Vector Regressor (SVR), Random Forest (RF), and Linear Regression (LinR) models using 17 molecular descriptors. For the PolyBERT-based ML group (+PolyBERT), we extract representations from a pretrained PolyBERT and train the same ML models. Each property is predicted using a separate model, resulting in five models for five properties.

4.2 Metrics

In our experiments, we report the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), since the task involves predicting continuous property values. To evaluate performance across multiple properties, we also report the Weighted Mean Absolute Error (wMAE), as introduced in the Kaggle challenge [12]. Details of how these metrics are calculated are provided in Appendix E.

4.3 Results and Discussion

We evaluate the models on the Kaggle polymer dataset across five property targets (Tg, FFV, Tc, density, and Rg) and report MAE, MAPE, and wMAE. For external validation, we evaluate Tg prediction on the GTT dataset. Following our protocol, VLM results are averaged over five inference runs, while ML baselines are averaged over five independently trained instances. Results for MAE and wMAE are reported in Table 1, while MAPE results are provided in Table 2. Due to space limitations, Table 2 is included in Appendix A. Details on fine-tuning resources and execution time are reported in Appendix C.

Table 1: Comparison of fine-tuned LVision and LText models with baseline approaches for predicting five polymer properties. Results are reported as mean and standard deviation of MAE, with the best performance for each property highlighted in bold.

Model	Kaggle						GTT
	Tg↓	FFV↓ $\times 10^{-2}$	Tc↓ $\times 10^{-2}$	Density↓ $\times 10^{-2}$	Rg↓	wMAE↓ $\times 10^{-2}$	Tg↓
MLP + RDKit	$\mathbf{46.5_{3.7}}$	$6.6_{1.5}$	$14.6_{4.9}$	$17.9_{4.5}$	$3.7_{0.2}$	$12.1_{1.2}$	$\mathbf{61.4_{7.7}}$
SVR + RDKit	$92.8_{0.0}$	$2.3_{0.0}$	$3.8_{0.0}$	$5.9_{0.0}$	$3.3_{0.0}$	$6.6_{0.0}$	$97.9_{0.0}$
RF + RDKit	$55.9_{0.2}$	$1.2_{0.0}$	$3.9_{0.0}$	$7.7_{0.0}$	$3.1_{0.0}$	$5.4_{0.0}$	$67.3_{0.1}$
LinR + RDKit	$54.9_{0.0}$	$1.4_{0.0}$	$3.7_{0.0}$	$6.4_{0.0}$	$3.1_{0.0}$	$5.3_{0.0}$	$62.2_{0.0}$
MLP + PolyBERT	$67.9_{2.0}$	$1.9_{0.3}$	$5.4_{0.7}$	$7.8_{0.6}$	$\mathbf{1.8_{0.0}}$	$5.4_{0.2}$	$67.5_{2.7}$
SVR + PolyBERT	$75.2_{0.0}$	$1.9_{0.0}$	$3.6_{0.0}$	$4.7_{0.0}$	$\mathbf{1.8_{0.0}}$	$4.9_{0.0}$	$77.9_{0.0}$
RF + PolyBERT	$64.5_{0.3}$	$1.2_{0.0}$	$\mathbf{3.1_{0.0}}$	$6.9_{0.0}$	$2.5_{0.0}$	$4.9_{0.0}$	$67.8_{0.2}$
LinR + PolyBERT	$168.8_{0.0}$	$\mathbf{1.0_{0.0}}$	$13.1_{0.0}$	$6.7_{0.0}$	$6.1_{0.0}$	$10.1_{0.0}$	$161.1_{0.0}$
Original LText	$100.2_{1.3}$	$4.9_{0.2}$	$7.9_{0.2}$	$14.3_{0.5}$	$5.9_{0.3}$	$12.0_{0.2}$	$104.7_{1.8}$
Original LVision	$90.3_{1.8}$	$4.4_{0.0}$	$8.1_{0.2}$	$9.6_{0.2}$	$7.2_{0.4}$	$11.6_{0.3}$	$97.4_{1.3}$
Fine-tuned LText	$60.0_{3.5}$	$\mathbf{1.0_{0.0}}$	$5.7_{0.5}$	$4.0_{0.1}$	$2.8_{0.1}$	$4.8_{0.1}$	$70.7_{3.3}$
Fine-tuned LVision	$58.0_{2.7}$	$\mathbf{1.0_{0.0}}$	$3.6_{0.1}$	$\mathbf{3.3_{0.1}}$	$2.3_{0.0}$	$\mathbf{4.1_{0.1}}$	$67.7_{2.5}$

Overall, Fine-tuned LVision achieves the best aggregate performance on the Kaggle benchmark, attaining the lowest wMAE of 0.041 (Table 1) and the best MAPE across FFV, Tc, and Density (Table 2). Fine-tuned LText performs competitively but trails LVision, underscoring the value of visual information in this study. ML baselines trained on RDKit descriptors and PolyBERT embeddings achieve advancements for some targets, yet still underperform compared to Fine-tuned LVision on the overall wMAE. Specifically, Fine-tuned LVision outperforms the second-best Fine-tuned LText (0.048), as well as all descriptor-based and embedding-based baselines. Fine-tuned LVision also achieves the best MAE for FFV (0.010) and Density (0.033), outperforming the next-best entries.

Fine-tuned models substantially improve over their original counterparts. For LVision, wMAE decreases from 0.117 (Original LVision) to 0.041; for LText, from 0.120 (Original LText) to 0.048. In terms of multimodal gains, Fine-tuned LText narrows the gap but still lags behind Fine-tuned LVision across all five MAE metrics, highlighting the added value of visual input beyond P-SMILES and descriptors. On unseen Tg evaluation (GTT), Fine-tuned LVision achieves an MAE of 67.7, improving over Fine-tuned LText (70.7). Comparisons to descriptor-based baselines on GTT required additional checks for unit consistency, data curation, and distribution shift. Compared to training and deploying five separate regressors (one per property), a single fine-tuned VLM delivers competitive or superior accuracy through a unified interface, reducing per-property model management while leveraging multimodal cues (structure image + P-SMILES + descriptors).

The improved performance of Fine-tuned LVision is promising and not unexpected. Prior work has shown that deep learning on 2D molecular depictions can match or outperform string-based representations such as SMILES by enabling augmentation, transfer learning, and visual feature extraction [15, 16]. While a 2D depiction generated by RDKit from a SMILES string does not inherently contain more chemical information than the SMILES itself [17, 18], it provides a visually structured representation that appears to enhance how a Vision–Language Model processes and interprets molecular structure.

5 Conclusion

We present a multimodal instruction-tuning dataset for polymer property prediction and fine-tune a Vision-Language Model using images, P-SMILES, and molecular descriptors. We compare its performance to baselines, including traditional ML with RDKit or PolyBERT features, and Llama-based vision and text-only models. Our fine-tuned Llama-3.2-11B-Vision-Instruct achieves competitive performance, highlighting the value of multimodal inputs. Moreover, it eliminates the need for separate per-property models, thereby reducing deployment and maintenance costs.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Institute of General Medical Sciences of National Institutes of Health under award P20GM139768, and the Arkansas Integrative Metabolic Research Center at the University of Arkansas.

References

[1] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28 (1):31–36, 1988. doi: 10.1021/ci00057a005.

[2] Christopher Kuenneth and Rampi Ramprasad. polybert: a chemical language model to enable fully machine-driven ultrafast polymer informatics. *Nature communications*, 14(1):4099, 2023.

[3] Sonakshi Gupta, Akhlak Mahmood, Shivank Shukla, and Rampi Ramprasad. Benchmarking large language models for polymer property predictions. *arXiv preprint arXiv:2506.02129*, 2025.

[4] Tianren Zhang and Dai-Bei Yang. Multimodal machine learning with large language embedding model for polymer property prediction. *arXiv preprint arXiv:2503.22962*, 2025.

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[6] Tran Doan Huan, Arun Mannodi-Kanakkithodi, Chiho Kim, Vinit Sharma, Ghanshyam Pilania, and Rampi Ramprasad. A polymer dataset for accelerated property prediction and design. *Scientific data*, 3(1):1–10, 2016.

[7] Chiho Kim, Anand Chandrasekaran, Tran Doan Huan, Deya Das, and Rampi Ramprasad. Polymer genome: a data-powered polymer informatics platform for property predictions. *The Journal of Physical Chemistry C*, 122(31):17575–17585, 2018.

[8] Huan Doan Tran, Chiho Kim, Lihua Chen, Anand Chandrasekaran, Rohit Batra, Shruti Venkatram, Deepak Kamal, Jordan P Lightstone, Rishi Gurnani, Pranav Shetty, et al. Machine-learning predictions of polymer properties with polymer genome. *Journal of Applied Physics*, 128(17), 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[10] Fanmeng Wang, Wentao Guo, Minjie Cheng, Shen Yuan, Hongteng Xu, and Zhifeng Gao. Predicting polymer properties based on multimodal multitask pretraining. *arXiv e-prints*, pages arXiv–2406, 2024.

[11] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. 2023.

[12] Gang Liu, Jiaxin Xu, Eric Inae, Yihan Zhu, Ying Li, Tengfei Luo, Meng Jiang, Yao Yan, Walter Reade, Sohier Dane, Addison Howard, and María Cruz. Neurips - open polymer prediction 2025. `https://kaggle.com/competitions/neurips-open-polymer-prediction-2025`, 2025.

[13] Soyeon Choi, Jungho Lee, Jaehyun Seo, Seung Won Han, Sung Ho Lee, Jeong-Hwan Seo, and Chaok Seok. Automated bigsmiles conversion workflow and dataset for homopolymeric macromolecules. *Scientific Data*, 11(1), 2024. doi: 10.1038/s41597-024-03212-4.

[14] RDKit. Rdkit: Open-source cheminformatics. `https://www.rdkit.org`. Accessed: 2025-08-22.

[15] Garrett B Goh, Charles Siegel, Abhinav Vishnu, Nathan O Hodas, and Nathan Baker. Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models. *arXiv preprint arXiv:1706.06689*, 2017.

[16] Matthew R Wilkinson, Uriel Martinez-Hernandez, Chick C Wilson, and Bernardo Castro-Dominguez. Images of chemical structures as molecular representations for deep learning. *Journal of Materials Research*, 37(14):2293–2303, 2022.

[17] Noel M O'Boyle. Towards a universal smiles representation-a standard method to generate canonical smiles based on the inchi. *Journal of cheminformatics*, 4(1):22, 2012.

[18] Mario Krenn, Qianxiang Ai, Senja Barthel, Nessa Carson, Angelo Frei, Nathan C Frey, Pascal Friederich, Théophile Gaudin, Alberto Alexander Gayle, Kevin Maik Jablonka, et al. Selfies and the future of molecular string representations. *Patterns*, 3(10), 2022.

[19] U.S. Environmental Protection Agency. Molecular descriptors guide. Technical Report Version 1.02, U.S. Environmental Protection Agency, National Center for Computational Toxicology, 2007. URL `https://www.epa.gov/sites/default/files/2015-05/documents/moleculardescriptorsguide-v102.pdf`.

[20] Datagrok. Molecular descriptors. URL `https://datagrok.ai/help/datagrok/solutions/domains/chem/descriptors`.

A Results on MAPE

Table 2: Comparison of fine-tuned LVision and LText models with baseline approaches for predicting five polymer properties. Results are reported as mean and standard deviation of MAPE, with the best performance for each property highlighted in bold.

Model	Kaggle					GTT
	Tg	FFV	Tc	Density	Rg	Tg
MLP + RDKit	$118.2_{8.3}$	$18.2_{3.9}$	$70.1_{23.3}$	$17.4_{4.7}$	$23.0_{1.6}$	$\mathbf{75.4_{4.9}}$
SVR + RDKit	$190.5_{0.0}$	$6.3_{0.0}$	$17.1_{0.0}$	$5.7_{0.0}$	$18.8_{0.0}$	$122.4_{0.0}$
RF + RDKit	$116.3_{0.5}$	$3.1_{0.0}$	$18.6_{0.0}$	$7.3_{0.0}$	$18.9_{0.0}$	$86.5_{0.4}$
LinR + RDKit	$147.3_{0.0}$	$3.9_{0.0}$	$16.9_{0.0}$	$6.2_{0.0}$	$18.6_{0.0}$	$87.2_{0.0}$
MLP + PolyBERT	$133.1_{12.6}$	$5.2_{0.8}$	$25.5_{3.6}$	$7.6_{0.7}$	$\mathbf{9.7_{0.1}}$	$113.1_{21.8}$
SVR + PolyBERT	$152.1_{0.0}$	$5.1_{0.0}$	$16.9_{0.0}$	$4.5_{0.0}$	$9.9_{0.0}$	$92.2_{0.0}$
RF + PolyBERT	$142.0_{1.2}$	$3.1_{0.0}$	$14.7_{0.0}$	$6.6_{0.0}$	$14.9_{0.0}$	$97.2_{0.3}$
LinR + PolyBERT	$543.2_{0.0}$	$2.7_{0.0}$	$64.1_{0.0}$	$6.9_{0.0}$	$38.7_{0.0}$	$337.2_{0.0}$
Original LText	$178.1_{12.0}$	$13.1_{0.4}$	$34.1_{1.0}$	$14.2_{0.5}$	$33.8_{1.3}$	$100.5_{3.7}$
Original LVision	$142.3_{10.3}$	$11.7_{0.1}$	$36.8_{1.4}$	$9.7_{0.2}$	$45.6_{1.7}$	$105.1_{3.6}$
Fine-tuned LText	$\mathbf{99.8_{22.5}}$	$\mathbf{2.6_{0.1}}$	$27.3_{2.1}$	$3.8_{0.1}$	$15.9_{0.5}$	$105.1_{6.8}$
Fine-tuned LVision	$119.5_{14.9}$	$\mathbf{2.6_{0.0}}$	$\mathbf{14.4_{0.7}}$	$\mathbf{3.2_{0.1}}$	$12.0_{0.2}$	$78.5_{8.4}$

B Dataset Statistics

Table 3: Number of samples for each property

	Tg	FFV	Tc	Density	Rg
Missing count	8400	1071	8106	8350	8349
Missing ratio	93.72%	11.95%	90.44%	93.16%	93.14%

C Fine-tuning Resources and Execution Time

We perform fine-tuning and evaluation on an H200 GPU with 141 GB of RAM. Fine-tuning LVision takes about 21 hours for 12 epochs with 1120×1120 images, while fine-tuning LText takes about one hour under the same settings.

D Molecular Descriptor Features

Here are the 17 molecular descriptors used in our dataset (descriptor descriptions adapted from [19, 20]):

- **MolWt**: Molecular weight
- **MolLogP**: Octanol–water partition coefficient (logP)
- **BalabanJ**: Balaban's J topological index
- **Chi0**: Zero-order molecular connectivity index
- **Chi1**: First-order molecular connectivity index
- **HallKierAlpha**: Hall–Kier alpha parameter
- **LabuteASA**: Labute's Approximate Surface Area
- **TPSA**: The polar surface area of a molecule based upon fragments.
- **FractionCSP3**: The fraction of C atoms that are SP3 hybridized

- **HeavyAtomCount**: The number of heavy atoms
- **NHOHCount**: The number of NHs or OHs
- **NOCount**: The number of nitrogens and oxygens
- **NumAliphaticRings**: The number of aliphatic rings
- **NumAmideBonds**: The number of amide bonds
- **NumAromaticRings**: The number of aromatic rings
- **NumRotatableBonds**: The number of rotatable bonds
- **NumSaturatedRings**: The number of saturated rings

E Metrics

MAE. The Mean Absolute Error (MAE) measures the average absolute difference between the predicted and ground truth values of a single property type:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|, \tag{1}$$

where n denotes the number of available ground-truth values for a property type under evaluation, y_i and $\hat{y}_i$ represent the ground-truth and predicted values of the i-th polymer, respectively.

MAPE. The Mean Absolute Percentage Error (MAPE) measures the average absolute percentage error between the predicted and ground truth values of a single property type:

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{|y_i|} \tag{2}$$

wMAE. Weighted Mean Absolute Error (wMAE) is the evaluation metric used in the Kaggle contest [12] to evaluate the overall prediction performance across five properties:

$$\text{wMAE} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{\mathcal{I}_i} w_k \cdot \left|\hat{y}_i^k - y_i^k\right|, \tag{3}$$

$$w_k = \left(\frac{1}{r_k}\right)\cdot\left(\frac{K\cdot\sqrt{1/n_k}}{\sum_{j=1}^{K}\sqrt{1/n_j}}\right), \tag{4}$$

where $\mathcal{I}_i$ denotes the set of property types of the i-th polymer, and $\hat{y}_i^k$ and y_i^k are the predicted and ground-truth values of the property k of polymer i-th, respectively. Moreover, w_k is reweighting factor for each property where n_k denotes the number of samples having k-th property, K is the number of property types, and $r_k = \max(y^k) - \min(y^k)$ represents the estimated value range of the k-th property based on the test data.