Open Polymer Challenge: Post-Competition Report

Gang Liu[1]   Sobin Alosious[1]   Subhamoy Mahajan[2]   Eric Inae[1]   Yihan Zhu[1]   Yuhan Liu[1]
Renzheng Zhang[1]   Jiaxin Xu[1]   Addison Howard[3]   Ying Li[2]   Tengfei Luo[1]   Meng Jiang[1]

[1]University of Notre Dame   [2]University of Wisconsin–Madison   [3]Kaggle

Corresponding emails: `{gliu7, mjiang2}@nd.edu`

Abstract

Machine learning (ML) offers a powerful path toward discovering sustainable polymer materials, but progress has been limited by the lack of large, high-quality, and openly accessible polymer datasets. The Open Polymer Challenge (OPC) addresses this gap by releasing the first community-developed benchmark for polymer informatics, featuring a dataset with 10K polymers and 5 properties: thermal conductivity, radius of gyration, density, fractional free volume, and glass transition temperature. The challenge centers on multi-task polymer property prediction, a core step in virtual screening pipelines for materials discovery. Participants developed models under realistic constraints that include small data, label imbalance, and heterogeneous simulation sources, using techniques such as feature-based augmentation, transfer learning, self-supervised pretraining, and targeted ensemble strategies. The competition also revealed important lessons about data preparation, distribution shifts, and cross-group simulation consistency, informing best practices for future large-scale polymer datasets. The resulting models, analysis, and released data create a new foundation for molecular AI in polymer science and are expected to accelerate the development of sustainable and energy-efficient materials. Along with the competition, we release the test dataset at `https://www.kaggle.com/datasets/alexliu99/neurips-open-polymer-prediction-2025-test-data`. We also release the data generation pipeline at `https://github.com/sobinalosious/ADEPT`, which is generalized from TC, R_g, and Density to simulate more than 25 properties.

1 Introduction

Machine learning (ML) techniques such as sequence-to-sequence models and graph neural networks have transformed many areas, including chat services and recommender systems [2, 38, 13]. These advances are now extending into the sciences, where even human experts may struggle to fully grasp the complexities [6, 34]. Many scientific discovery efforts have focused on small molecules (*e.g.*, ChEMBL [19], PubChem [28], OGB-LSC [24]) or inorganic crystals (*e.g.*, Materials Project [25], GNoME [34]). However, limited attention has been directed toward *polymers*, which are promising candidates for environmentally friendly and sustainable materials. For example, polymeric membranes are central to separation technologies that can greatly reduce the energy, carbon, and water intensity of many traditional thermally driven separation processes [44].

Polymer discovery has been hindered by the absence of large-scale, open-source datasets [37]. Polymer data generation experimentally is slow because polymer synthesis involves complex polymerization reactions [27] and processing steps [5]. For instance, for gas separation polymers, decades of experimental efforts between 1950 and 2018 produced only about 1,500 polymers with gas permeability measurements, where each major gas type has 400 to 800 reported values. Generative models
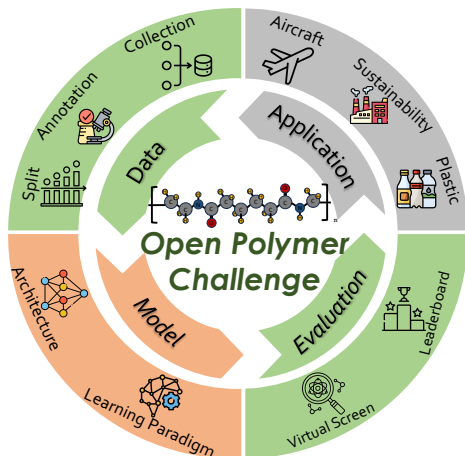
Figure 1: Overview of Open Polymer Challenge from four dimensions.

can propose hypothetical polymer structures [32], but these candidates often lack experimentally validated properties, limiting their practical utility.

The scarcity of polymer datasets has slowed the progress of ML models designed specifically for polymers compared to other areas, such as small molecules and proteins. Motivated by this gap, we launched the community-driven Open Polymer Challenge (OPC), as presented in Figure 1. We prepared polymer data and annotations and set up evaluation metrics to call for new solutions and models for polymer applications. There are five molecular dynamics (MD)-simulated polymer properties (Table 3): thermal conductivity (TC), radius of gyration (R_g), density (Density), fractional free volume (FFV), and glass transition temperature (T_g). To the best of our knowledge, this is the first ML competition dedicated to polymer informatics in both the ML and materials science communities. As summarized in Table 1, the OPC differs from existing competitions in biology, cheminformatics, and materials science, as it introduces new data objects and challenging multi-task prediction problems.

The competition revealed several important lessons about data preparation and property generation. Early in the challenge, a portion of previously generated polymer data that had once been publicly visible was unintentionally reused by participants. This happened even though the data had been removed from GitHub before the competition, but it had already been re-circulated by third parties. After detecting the overlap through SMILES matching on the leaderboard, we released that subset openly and created new polymer structures with new MD based property labels for both the public and private leaderboards. These updates also exposed issues that often arise when simulations come from different groups, where we detail in Section 3.

Although these challenges required careful handling, the competition still succeeded in producing reliable benchmarks and meaningful scientific insights. The competition ran on Kaggle from June 16, 2025 to September 15, 2025, spanning three months. It attracted more than 10,000 registrations, 2,600 participants, 2,000 teams, and 50,000 submissions from over 100 countries. The five prize-winning teams represented a wide range of backgrounds, including undergraduate and graduate students, as well as software engineers with or without prior chemistry experience. Analysis of their solutions shows that high-performing approaches relied on careful domain-specific data curation, fingerprint- and descriptor-based feature engineering, and strong but relatively simple models, such as gradient-boosted trees with rigorous cross-validation. These strategies yielded accurate predictions for T_g, density, FFV, and R_g under small, noisy, and imbalanced data. The resulting best practices and creative ideas, including physics-informed features and structure-aware polymer augmentations, provide concrete guidance for future work in polymer property prediction and molecular AI. We discuss the new ML models and methods in Section 4.

Table 1: Comparison between the proposed challenge and existing machine learning competitions (their latest version) on chemistry and material data.

Challenge	Venue	Domain	Data object	Task	Annotation method
CellSeg	NeurIPS 2022	(1) Biomedicine (2) Cell	Cell Image	Cell segmentation	Manual
Single-cell perturbation	NeurIPS 2021–2023	(1) Drug (2) Cell	Cell & Compound pair	Differential gene expression value	Wet-lab
OGB-LSC (PCQM4Mv2)	KDD 2021 NeurIPS 2022	(1) Quantum (2) Chemistry	Molecule Graph	HOMO-LUMO energy gap	Density functional theory
Open Catalyst Challenge	NeurIPS 2021–2023	(1) Catalyst (2) Materials	Adsorbate Catalyst surface	Adsorption energy	Density functional theory
BELKA challenge	NeurIPS 2024	(1) Drug (2) Biology	Small Molecule	Protein binding affinity	DNA-encoded library technology
Open Polymer Challenge:					
Open Polymer Challenge	Kaggle & NeurIPS 2025	Polymer Materials	Polymer Structure	Thermal, structural, and thermophysical properties	Molecular dynamics

Table 2: Unique polymer structures in the Open Polymer Challenge. "Total" counts all SMILES in each dataset, while "With any label" counts only polymers containing at least one property label.

Dataset	Total Unique SMILES	Unique SMILES With Any Label
Train	7,973	7,973
Public Leaderboard	295	295
Private Leaderboard	3,207	1,354

2 Competition Overview

Problem Definition A polymer X can be represented in two standard formats: (1) a SMILES string, which encodes the molecular structure as a sequence, and (2) a graph, where atoms are nodes and bonds are edges with defined chemical attributes. These two formats are chemically consistent and describe the same polymer structure. In this competition, we released the SMILES strings as polymer identifiers. We used five material properties: thermal conductivity (TC), radius of gyration (R_g), mass density (Density), fractional free volume (FFV), and glass transition temperature (T_g). All properties are obtained through MD simulations.

This challenge curated datasets containing 11,475 unique polymers, among which 9,625 have at least one labeled property. The distributions of structures and properties across the train, public leaderboard, and private leaderboard splits are given in Tables 2 and 3, together with their units and simulation time. Polymers were organized into three subsets: the train set used for model development, the public leaderboard set, and the private leaderboard set. Only the train set was released on Kaggle; both leaderboard sets remain hidden for evaluation.

Competition Metrics We evaluate the models using a weighted Mean Absolute Error (wMAE) across all properties, defined as

$$\text{wMAE} = \frac{1}{|\mathcal{X}|} \sum_{X \in \mathcal{X}} \sum_{i \in \mathcal{I}(X)} w_i \cdot |\hat{y}_i(X) - y_i(X)|, \tag{1}$$

where $\mathcal{X}$ is the set of polymers being evaluated, and $\mathcal{I}(X)$ is the set of property types available for polymer X. The terms $\hat{y}_i(X)$ and $y_i(X)$ are the predicted and true values of the i-th property.

To balance contributions from properties with different scales and varying amounts of available data, we use a reweighting factor

$$w_i = \left(\frac{1}{r_i} \right) \cdot \left(\frac{K \cdot \sqrt{1/n_i}}{\sum_{j=1}^{K} \sqrt{1/n_j}} \right), \tag{2}$$

where n_i is the number of available labels for the i-th property, and $r_i = \max(\mathcal{Y}_i) - \min(\mathcal{Y}_i)$ is its value range estimated from the training data. K is the total number of prediction tasks.

Table 3: Polymer properties in the open polymer challenge. FFV is fractional free volume, T_g is glass transition temperature, TC is thermal conductivity, R_g is radius of gyration, and Density is mass density. CPU hours per sample correspond to the simulation cost for each property. Statistics include minimum, median, and maximum values for the train, public, and private datasets.

| Dataset	Property	Min	Median	Max	Count	Unit	CPU hrs/sample
Train	T_g	-148.030	74.040	472.250	511	°C	2880
	FFV	0.227	0.364	0.777	7030	-	864
	TC	0.047	0.236	0.524	737	$W \cdot m^{-1} \cdot K^{-1}$	1440
	R_g	9.728	15.052	34.673	614	Å	432
	Density	0.749	0.948	1.841	613	$g \cdot cm^{-3}$	432
Public Leaderboard	T_g	-138.802	97.764	359.084	95	°C	2880
	FFV	0.315	0.353	0.656	86	-	864
	TC	0.077	0.244	0.678	239	$W \cdot m^{-1} \cdot K^{-1}$	1440
	R_g	11.208	17.436	32.301	54	Å	432
	Density	0.747	1.057	1.360	244	$g \cdot cm^{-3}$	432
Private Leaderboard	T_g	-86.650	169.670	484.690	166	°C	2880
	FFV	0.306	0.351	0.454	137	-	864
	TC	0.024	0.239	0.954	1165	$W \cdot m^{-1} \cdot K^{-1}$	1440
	R_g	9.445	18.511	35.530	1056	Å	432
	Density	0.135	1.081	1.733	1282	$g \cdot cm^{-3}$	432
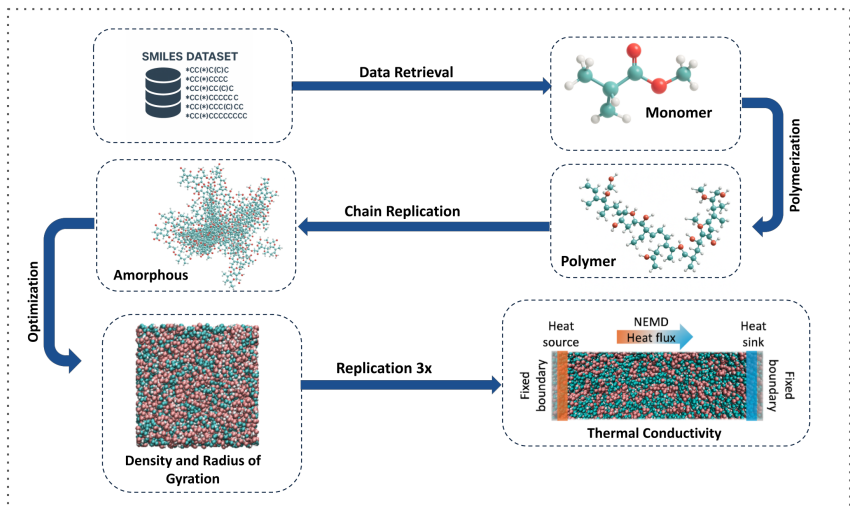
Figure 2: An example of a high-throughput MD workflow for polymer property calculations. The process includes SMILES parsing, monomer-to-polymer chain construction, amorphous packing, multi-stage equilibration, and computation of density, radius of gyration, and thermal conductivity.

This weighting scheme has two goals: (1) normalize for scale so that properties with large numerical ranges do not dominate the loss; (2) adjust for label imbalance through inverse square-root scaling so that rare properties receive appropriate weight.

3 Data Generation Pipeline

Due to the detection of data leakage early in the competition, we generated new evaluation sets for all five properties during the challenge. To prepare new polymer structures, we used three sources. First, we sampled 100 structures from the PI1M database containing one-million virtual polymers [32] that had no overlap with the competition data. Second, we used the PI1M generative model [32] to create 2,000 previously unseen structures. Third, we added 900 experimentally validated polymers that lacked competition labels as a backup pool. In total, we prepared 3,000 candidate structures, forming

a superset from which the final public and private test sets were drawn. This avoided repeated updates and prevented teams from re-running inference after each correction. Two research groups simulated the polymers in parallel, one for TC, Density, and R_g, and the other for T_g and FFV.

There were two rounds of data updates. The first update was public and caused shifts in the public leaderboard after the new data were released. During this step, we identified RDKit version conflicts: about 20 of the 3,000 SMILES strings were valid in the older RDKit versions but invalid in newer ones. We also found a unit mismatch (K <-> °C) in the new T_g values. Both issues were corrected. The second update affected only the private leaderboard. In this stage, we observed a shift in the mean T_g between the training data and the private test set. A detailed check (see Section 3.2) showed that the shift could have arisen from differences between bi-linear and hyperbolic fitting methods, not from incorrect units.

These observations highlight three lessons. First, even brief public release of intermediate data can propagate widely and create leakage. Second, consistency across simulation batches requires careful checks of units, fitting methods, and property definitions. Third, dynamic data generation during a live competition introduces the risk of distribution shifts that cannot be curated or stabilized in advance. Next, we detail the methods and pipelines used in data generation.

3.1 Amorphous Polymer Structure Generation

To construct the amorphous polymer systems, we employed a high-throughput MD workflow, named ADEPT: Automated molecular Dynamics Engine for Polymer simulaTions, consisting of monomer-to-polymer structure generation, amorphous cell packing, and multi-stage equilibration (available at `https://github.com/sobinalosious/ADEPT`). Figure 2 illustrates the overall MD workflow used for polymer generation, amorphous packing, and property calculations. Polymer monomers were represented using simplified molecular input line entry system (SMILES) strings [50]. A Python-based pipeline built on PySIMM [18] was used to generate polymer chains from these monomers. Linear chains were grown using a random-walk polymerization algorithm, followed by methyl (–CH_3) termination at both chain ends. For the density, radius of gyration and TC calculations, the General AMBER Force Field 2 (GAFF2) [48] was applied to assign force field parameters, and Gasteiger charges were used for electrostatics. Each chain contained approximately 600 atoms, and six chains were placed randomly in a periodic simulation box at low initial density to form an amorphous cell. Input files compatible with LAMMPS [40] were automatically generated by the pipeline for subsequent MD simulations.

The systems were then equilibrated through two successive stages: initial relaxation and annealing. During initial relaxation, electrostatic interactions were disabled and Lennard–Jones (LJ) interactions were truncated at 0.3 nm to alleviate large forces from close contacts. The system was equilibrated in the NPT ensemble at 100 K for 2 ps with a 0.1 fs timestep, followed by heating to 1000 K over 1 ns in the NVT ensemble. Next, NPT simulations were conducted at 1000 K and 0.1 atm for 50 ps and subsequently at pressures ramped from 0.1 atm to 500 atm for 1 ns using a 1 fs timestep. SHAKE constraints [43] were applied to maintain covalent bond lengths.

In the annealing stage, electrostatics were re-enabled using the particle–particle–particle–mesh (PPPM) Ewald summation method [21], and the LJ cutoff was increased to 0.8 nm. The system was equilibrated in an NPT ensemble at 1000 K and 1 atm for 2 ps using a 0.1 fs timestep, then cooled to 300 K at a rate of 140 K/ns with SHAKE enabled. Finally, an 8 ns NPT simulation was performed at 300 K and 1 atm with a 1 fs timestep to obtain a well-equilibrated amorphous configuration suitable for subsequent density, radius of gyration, and thermal conductivity calculations (see Fig. 2). The simulation box was allowed to relax but constrained to remain cubic.

During the simulation, data consistency was maintained through a fully automated Python–LAMMPS workflow that standardized system generation, force field assignment, equilibration, and analysis across all polymers. Within each of the two data generation groups, identical simulation parameters and force fields were applied to each system, and input scripts were auto-generated to avoid manual errors. Post-processing was performed using uniform Python routines with checks for trajectory length, thermodynamic stability, and file integrity. Repeated property calculations followed the same analysis scripts to ensure reproducibility and comparability across the dataset.

The generation of the amorphous polymer system for FFV and T_g followed a different workflow. Material Studio was used to create polymer chains with ~2000 atoms and 20 identical chains were

packed into a simulation box to the initial amorphous polymer system.[46] The initial system was optimized with polymer consistent force field (PCFF) [45] under a 3D periodic boundary condition using a 21-step equilibration protocol [1]. The LAMMPS simulation package was used for all simulations.

3.2 Property Generation

Density   The density of each polymer system was calculated from NPT simulations conducted at 300 K and 1 atm following the annealing stage. After equilibration, the simulation was continued for a production run, and instantaneous system density was recorded at regular intervals using LAMMPS' built-in `variable density` command. The final reported density corresponds to the time-averaged value over this production window.

Density was evaluated only after long NPT equilibration runs to ensure that each polymer system reached a stable volume and energy. For every trajectory, we identified a stable sampling window and restricted the calculation to the final 3 ns, where volume fluctuations were minimal. Density values were reported only for systems in which the complete simulation finished without interruption or numerical failure.

Radius of Gyration (R_g)   The radius of gyration (R_g) of individual polymer chains was computed during the same NPT production run at 300 K and 1 atm. Per-molecule chunking was performed using `compute chunk/atom molecule`, and R_g values were calculated for each chunk using `compute gyration/chunk`. Time-averaging was carried out using `fix ave/time` over the entire production window. Reported R_g values correspond to the temporal average and are summarized as mean $\pm$ standard deviation over all chains. The radius of gyration is given by

$$R_g = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{r}_i - \mathbf{r}_{\mathrm{cm}}\|^2}, \tag{3}$$

where N is the number of atoms in the chain, $\mathbf{r}_i$ are the atomic positions, and $\mathbf{r}_{\mathrm{cm}}$ is the chain's center of mass. For the radius of gyration, polymer chains were consistently identified using molecule IDs so that atoms were never mixed between different chains. R_{g} values were extracted only from fully equilibrated amorphous cells, avoiding intermediate configurations that were still stretched or compressed. To reduce the impact of thermal noise, instantaneous R_{g} values were averaged over long time windows, providing smooth and representative chain statistics.

Thermal Conductivity (TC)   Thermal conductivity was computed along the x-direction using a non-equilibrium MD (NEMD) approach [33]. After equilibration, the simulation cell was replicated $3 \times 1 \times 1$ along x. Narrow source and sink slabs were defined symmetrically around the box center, and atoms in these slabs were thermostatted using Langevin dynamics at 320 K (source) and 280 K (sink), respectively. The central region evolved under NVE dynamics. A steady-state temperature profile was established by running the simulation for 2×10^7 steps with a 0.25 fs timestep (5 ns total). Temperature profiles were collected by binning the system along x using the commands `compute chunk/atom bin/1d x ...` and `fix ave/chunk`. The temperature gradient $\mathrm{d}T/\mathrm{d}x$ was obtained by fitting only the central linear region. Heat flux J_q was computed from the cumulative energy added to the hot slab and removed from the cold slab, normalized by cross-sectional area. Thermal conductivity κ_x was then determined via Fourier's law,

$$\kappa_x = -\frac{J_q}{\mathrm{d}T/\mathrm{d}x}. \tag{4}$$

Multiple non-overlapping time segments were analyzed independently, and the mean and standard deviation of κ_x were reported.

Thermal conductivity calculations were accepted only after establishing a clear steady state in the non-equilibrium simulations. Heat flux and temperature profiles were monitored, and κ was computed only once a linear, stable temperature gradient was observed. Energy added to the hot region and removed from the cold region was compared to ensure bidirectional balance and to rule out numerical drift. The steady-state portion of each trajectory was further divided into several independent time blocks; κ was calculated for each block, and the reported value corresponds to the block-averaged mean. The temperature-gradient fit was restricted to the central conduction region, excluding noisy bins near the thermostatted zones. Any unphysical negative κ values were discarded during post-processing.
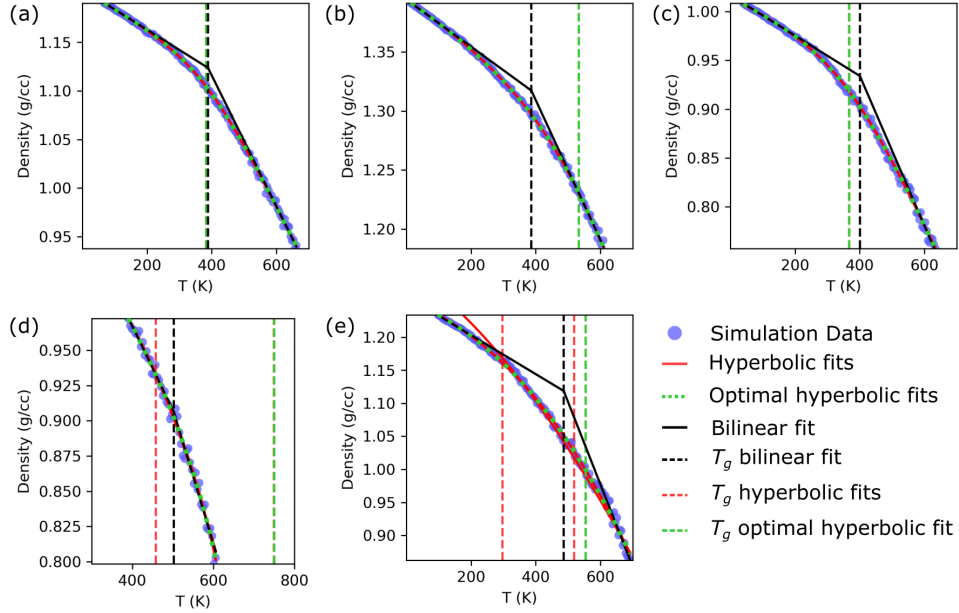
6

Figure 3: Glass transition temperature (T_g) from bi-linear and hyperbolic fits. (a) Early test cases where bi-linear and hyperbolic fits produced identical results. Later verifications showed (c, d) good hyperbolic fits can cause significant deviations in T_g, and (d, e) produce wide range of T_g that is dependent on the constraint bounds on fitting parameters. (d) Best hyperbolic fit can be high.

Fractional Free Volume (FFV)   For FFV calculations, NPT simulations at 300 K and 1 atm were performed for 5,000,000 steps using a timestep of 1 fs, totaling 5 ns. Certain simulations were unstable at 1 fs, for such systems smaller timesteps of 0.75, 0.5, and 0.25 fs were attempted. Polymer systems that were not stable at 0.25 fs were ignored. A total of 10 structures from the last 1,000,000 steps of the simulation was used to evaluate FFV using PoreBlazer 4.0 [16].

PoreBlazer 4.0 can evaluate FFV of two types: total and network-accessible (percolated). For each of these types, the FFV is occupiable with He and N_2, and geometric are evaluated by the software. The reported FFV data was the total geometric FFV (ratio of total `V_G_A^3` and `V_A^3` in the `summary.dat` output file generated by PoreBlazer 4.0) [47]. Initially, the default total N_2 probe FFV was used to determine the FFV. This resulted in several near-zero FFV that were outside the training range. This issue was quickly identified, and accurate geometric FFV were added. The uploaded data had no data inconsistency; All data were generated using Poreblazer 4.0 and geometric FFV for each polymer was determined.

Since FFV are measured using standard software, no data distribution shift are expected. Observations of better prediction accuracy with FFV^2 was again a mere coincidence. Since, $0 < FFV < 1$, squaring FFV, reduces its magnitude. Such strategies might improve model performance if output FFV is known to be lower, however it will perform much worse when output FFVs have higher mean.

Glass Transition Temperature (T_g)   The system was initially equilibrated at 650 K (time constant 0.1 ps) and 1 bar (time constant 1 ps, drag 0.2) for 1 ns. Thereafter, at the same pressure, the system was cooled to 150 K over 10 ns. The density of the polymer system was plotted as a function of the system temperature.

The T_g values from density-temperature plots can be evaluated using two popular methods: (i) bi-linear fit, where two linear asymptotes are evaluated, and their intersection provides the T_g, (ii) hyperbolic fit, where the focus of the hyperbola yields the T_g [39]. For hyperbolic fits, all data are fitted to Eq. 5, where (ρ_0, T_g) is the hyperbola center, and slopes $-a$ and $-(a+b)$ are the asymptotes at low and high temperature regimes, where $a, b > 0$. As $c \to -\infty$, the hyperbola approaches a piecewise linear function (similar to a bilinear fit).c

$$\rho(T) = \rho_0 - \left(a - \frac{b}{2}\right)(T - T_g) - b\sqrt{\frac{(T - T_g)^2}{4} + e^c} \qquad a > 0, b > 0 \qquad (5)$$

There are inherent issues with the bi-linear fit method, as one has to choose an arbitrary range over which the linear fit is performed. The optimal range for linear fits can differ based on the data. Therefore, hyperbolic fits were used to perform a high-throughput evaluation of T_g. Since hyperbolic fits are used over the entire temperature range, only limited constraints are necessary (Eq. 5). While generating the test data for the public leaderboard, consistency tests were performed for several polymers, which yielded good results (Fig. 3 (a)). With the initial consistency, the public leaderboard data were generated using hyperbolic fits. However, comprehensive checks after the Kaggle competition showed inconsistencies between T_g generated with bi-linear fit and hyperbolic fitting methods (Fig. 3 (b)-(e)). Many T_g values evaluated with hyperbolic fits are higher than the corresponding values from the bi-linear fit (Fig. 3(b)-(e)). Another issue was that T_g values from the hyperbolic fits were dependent on the constraints put on T_g. These issues caused some of the shifts in the mean T_g and were not associated with unit conversions. We note that polymers of the order of thousands were considered in the competition but total possibilities of polymers can be billionds-and-billions. In such cases differences in data distribution is unavoidable.

We note that the data distributions were not associated with °C to °F conversions. The public leaderboard data associated with T_g had a mean and standard deviation of $\mu_{pub} = 102.9$ and $\sigma_{pub} = 103.7$, respectively. The private leaderboard data associated with T_g had a mean and standard deviation of $\mu_{pri} = 179.8$ and $\sigma_{pri} = 134.9$, respectively. Let us consider a linear transformation $y = ax + b$ that converts public leaderboard data to private leaderboard. Then,

$$a = \frac{\sigma_{pri}}{\sigma_{pub}} = 1.3$$
$$b = \mu_{pri} - a\mu_{pub} = 46 \qquad (6)$$

For conversions from °C to °F, the expected constants are $a = 1.8$ and $b = 32$, which differ clearly from the observed values.

4 Competition Results and Discussion

4.1 Winner solutions

While each of the top solutions introduced unique innovations, they all converged on similar strategies, including additional data collection and cleaning, data augmentation, T_g label shift correction, fingerprint-based feature selection, and cross-validation. These shared solution features enabled participants to create models that achieved high predictive performance in this competition.

To remedy the limitation of data scarcity in the competition, the majority of the top solutions collected additional polymer data to expand the labeled training sets. Aside from the original training and supplementary data, most participants chose to further incorporate additional data sources of T_g [12, 7, 37] and Density [23, 20]. Additionally, the 1st place solution chose to perform fast MD simulations with neural networks to calculate Density, FFV, and R_g for 1116 polymers from PI1M [32]. These additional data sources aid model training by expanding coverage in the chemical space and reducing overfitting.

Most teams cleaned the data by performing canonicalization and kekulization to clean and deduplicate the training data. However, 3 of the top-10 solutions chose to use non-canonical forms of each polymer SMILES to generate new training data. Some teams used other forms of data augmentation as well, such as the 3rd place representing polymers as multi-monomer chains explicitly or the 4th place substituting functional groups with similar groups. The 1st place chose to perform fairly extensive data filtering, first by filtering training data whose predictions from an ensemble model exceed an error threshold and then by removing out-of-distribution samples, specifically from the TC data. The team found that removing polymers with TC values over 0.402 led to better results.

The final data consideration that most teams addressed was the shift in the T_g labels between the training and leaderboard test data. Teams chose to scale the data by manually selecting a constant

8

bias term or train a linear regressor to find the optimal weight and bias terms. The fourth-place team found that by using a quantile regression target, which guides the model to learn from values higher or lower in the label distribution (for example, using an objective percentile $\alpha = 0.85$), they could improve predictive accuracy for T_g and R_g. These data processing strategies led to the success of these top solutions.

Feature Selection   The features often selected by the top solutions were fingerprints and molecular descriptors. All solutions in the top-10 used Morgan fingerprints [42]. RDKit descriptors [30], MACCS [14], Atom Pair [9], Topological Torsion [36], Mordred descriptors [35], and ECFP [42] were also used, listed in descending order of use in feature selection. The top-3 solutions used graph features, although only the 3rd place team used a graph neural network to represent the polymer. The 5th place used FFV and TC prediction values as features to predict T_g, R_g, and Density, the rationale being that FFV and TC data were more abundant and were closely linked to the other properties according to their observations. The 1st place was unique among the top solutions for utilizing uncommon features, including shape-based descriptors, Gasteiger charge, element composition, bond type ratios, and polyBERT embeddings. The 1st and the 3rd places explicitly mentioned filtering features by those that were most predictive. Despite the advances in sequence processing and graph learning, SMILES embeddings and graphs were not commonly used, each only appearing in one of the top-10 solutions.

Model and Training   Modeling strategies were surprisingly diverse. While most of the top solutions used tree-based models such as XGBoost [10], LightGBM [26], CatBoost [41], some used other models. The 1st place used an ensemble of polyBERT [29], AutoGluon [15], and Uni-Mol [4], the 3rd place used a GAT, and the 5th place implemented a physics-based mathematical model with their tree ensemble. For model training, all solutions used 5-fold cross-validation. Notably, the 1st place performed BERT pretraining on self-supervised synthetic labels before finetuning.

4.2   Best Practice in Polymer Property Prediction

A key challenge in the competition was the inherently small scale, imbalanced, and noisy nature of the available polymer property datasets. With only a few thousand samples, the top 200 solutions repeatedly demonstrated that domain-specific data curation, the use of simple, well-tuned models combined with complementary signals or restrained ensembling, and disciplined learning strategies were key to achieving robust generalization.

Mitigating small-data constraints.   A recurring strength among high-performing solutions was their careful adaptation to the limited and noisy nature of polymer data. Efficient techniques included canonicalization (mapping each SMILES to a unique standard representation) and kekulization (explicitly assigning single and double bonds to aromatic systems), which generated chemically valid sequences without altering molecular identity. In contrast, aggressive augmentation such as random stereoisomer or tautomer enumeration (listing all valid stereochemical variants) frequently led to overfitting, with the 3rd place team explicitly reporting its ineffectiveness for GNNs. Empirically, top solutions avoided unvalidated augmentation, instead focusing on representation-level diversity grounded in chemical invariance and test-time robustness.

To expand structural expressiveness in small-data regimes, top-performing teams extensively constructed feature pools, combining classical descriptors (e.g., RDKit [30], Mordred [35]) with diverse fingerprints (e.g., Morgan/ECFP [42], MACCS [14], Atom Pair [9], Topological Torsion [36]), and in some cases, 3D geometric or MD-derived attributes. Feature counts often exceeded thousands, necessitating rigorous feature selection. The choices of different data strategies among top teams are illustrated in Figure 4.

A common pipeline involved training preliminary tree-based models (e.g., XGBoost [10], Light-GBM [26], CatBoost [41]) to compute per-family importance scores and retain only a top-N subset before fitting the final models. Automated hyperparameter search (e.g., Optuna [3]) was also used to tune the number of retained features, drop entire feature categories, or adjust fingerprint dimensionality. Notably, the 10th place team implemented a robust multi-stage strategy: statistical gating, keeping descriptors only if they passed correlation or mutual-information tests with the target, followed by collinearity pruning to remove redundant signals. Such workflows preserved chemical signal while improving robustness under sparse supervision.
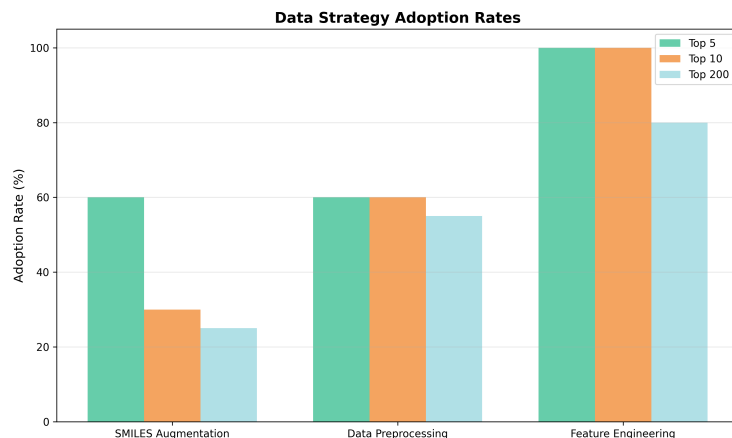
Figure 4: Choices of data strategies among Top 5, Top 10, and Top 200 leaderboard teams. Feature engineering (e.g., custom descriptors, fingerprints) was universally adopted in the top tiers, while use of SMILES augmentation showed decreasing adoption beyond Top 10.

Modeling strategies under small-data conditions emphasized robustness over complexity. Tree-based methods (e.g., LightGBM [26], XGBoost [10], CatBoost [41], TabPFN [22]) consistently served as robust baselines, appearing in all top-5 solutions. Simple configurations with well-engineered features proved stronger than complex neural networks. The 14th-place team observed an inverse correlation between model complexity and performance. While several teams experimented with pretrained transformers (e.g., ChemBERTa [11], ModernBERT [49], CodeBERT [17]) or GNN methods (e.g., GATv2Conv [8], GREA [31]), these models required careful pretraining or ensembling to succeed. Most top teams employed lightweight ensembles, such as target-wise averaging of model outputs rather than deep fusion, to maintain stability. AutoGluon [15] was also adopted in some solutions as an alternative to manually constructed model stacks for ensemble training under resource constraints. These modeling choices are quantitatively summarized in Figure 5, which highlights the prevalence of tree-based methods and the more selective use of neural architectures in top-ranking solutions.
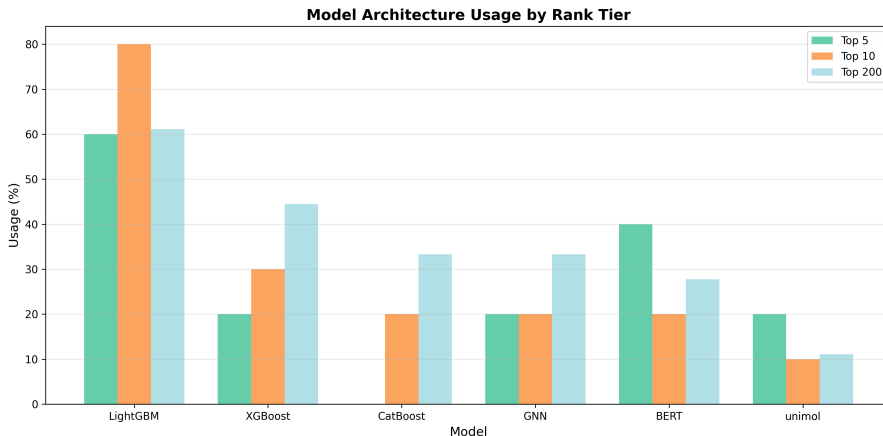

Figure 5: Model architecture usage among Top 5, Top 10, and Top 200 leaderboard teams. Tree-based models, especially LightGBM, remained dominant in top positions.

Addressing data imbalance and distribution skew. The challenge's multi-target setting introduced heterogeneity across properties: differing scales, label densities, and noise levels. To address this, top teams employed complementary strategies as follows:

Several teams leveraged external datasets from public resources to enrich under-sampled regions. Rather than direct concatenation, they performed rigorous distributional checks, including examining

value ranges, SMILES consistency, and MAE gaps on overlapping entries. Retained datasets were harmonized via dataset-specific linear transformation, bin-wise resampling, or MAE-driven calibration. Simple regressions on overlapping samples were often sufficient to estimate optimal slope and intercept corrections. Once aligned, supplemental data boosted model performance.

Stratified K-fold validation with per-target quantile binning was widely adopted to ensure balanced folds and reliable error estimates [51]. Most teams trained separate single-task models for each target to isolate noise from underrepresented targets. This setup was often paired with out-of-fold predictions to construct meta-ensembles without leakage. To further reduce systematic bias, lightweight post-processing techniques—including isotonic regression, mean alignment, property-specific clipping, or fold-wise linear calibration—were employed, especially for tail-heavy properties where raw models tended to underpredict extremes.

For distribution shifts, participants applied target-wise normalization—for example, max-scaling, or applying heuristic offsets ($+20°C$ to $+40°C$) based on cross-validation feedback and leaderboard probing. Similar corrections were also used for the Density property and R_g, yielding more stable and calibrated predictions. For example, the 3rd-place team used fold-wise regression calibration, improving scores by 5.5%. These findings highlighted the importance of error calibration, fold-aware correction, and diagnostic probing for robustness under real-world shifts.

4.3 Creative Strategies and Open Challenges

Beyond the dominant themes of data cleaning and ensemble modeling, several participants explored ideas that offered promising directions for advancing polymer informatics.

Structure-aware augmentations grounded in polymer semantics. One idea was the recognition that monomer-level SMILES representations often lacked enough topological context to capture repeating patterns in real polymers. Some participants (the 3rd place and the 200th place) proposed SMILES level augmentation through polymer chain extension by connecting the polymerization points to generate plausible dimers or trimers. They suggested that chain extension could reflect the physical repeat structure of polymers rather than only their static composition, and that it could improve performance relative to isomer enumeration for GNN models. This approach aligned with the repetition augmentation introduced in a recent study on repetition invariant representations [52].

Integrating physics priors through hybrid modeling. While most solutions relied on neural networks, select participants incorporated domain knowledge into both input features and prediction logic. The 5th place team blended physics-inspired descriptors with ML-based predictions through hierarchical modeling. FFV and TC were first predicted and then used as inputs for Density and R_g. Others manually constructed feature bundles that captured rigidity, hydrogen bonding, and packing heuristics, which are mechanistic cues often missed by learned representations. These methods showed that domain knowledge can improve performance even when models remain simple.

Persistent challenge: data shift. Distributional robustness remained a major difficulty in polymer property prediction, especially when datasets were sourced from heterogeneous simulations or experimental protocols. Polymer annotations are sensitive to force field parameters and post-processing pipelines, and even small methodological differences can produce meaningful shifts that reduce model generalization. This issue was clearly seen during the competition. Several teams identified systematic distribution shift, especially in T_g, as the most influential factor that affected leaderboard outcomes. For example, the 1st place solution applied a calibrated offset proportional to the standard deviation ($T_g \leftarrow T_g + \sigma \times 0.5644$). These corrections, however, depended heavily on post-hoc public leaderboard probing and manual calibration rather than systematic diagnostics, showing the fragility of current pipelines when hidden shifts are present.

5 Conclusion

The Open Polymer Challenge created the first large-scale, community-driven benchmark for polymer property prediction and revealed both the promise and the current limitations of molecular AI in polymer science. Participants demonstrated that careful data curation, targeted augmentation, and strong but simple models can reach high accuracy even under small, noisy, and imbalanced datasets. The competition also exposed critical challenges in data generation, distribution shifts, and cross-group simulation consistency, emphasizing the need for transparent pipelines and standardized

polymer datasets. Beyond the leaderboard results, the creative strategies and practical insights from hundreds of teams provide a foundation for future advances in polymer informatics. We hope that this benchmark, together with the released data, code, and analysis, will support sustained progress in molecular AI for materials discovery and motivate further collaboration between the ML and polymer science communities.

References

[1] Lauren J. Abbott, Kyle E. Hart, and Coray M. Colina. Polymatic: a generalized simulated polymerization algorithm for amorphous polymers. Theoretical Chemistry Accounts, 132 (1334), 2013.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2623–2631, 2019. doi: 10.1145/3292500.3330701.

[4] Eric Alcaide, Zhifeng Gao, Guolin Ke, Yaqi Li, Linfeng Zhang, Hang Zheng, and Gengmo Zhou. Uni-mol docking v2: Towards realistic and accurate binding pose prediction, 2024. URL https://arxiv.org/abs/2405.11769.

[5] Donald G. Baird. Polymer processing. In Robert A. Meyers, editor, Encyclopedia of Physical Science and Technology (Third Edition), pages 611–643. Academic Press, New York, third edition edition, 2003. ISBN 978-0-12-227410-7. doi: https://doi.org/10.1016/B0-12-227410-5/00593-7. URL https://www.sciencedirect.com/science/article/pii/B0122274105005937.

[6] Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. Nature, 624(7992):570–578, 2023.

[7] Claudia Borredon, Luis A Miccio, Silvina Cerveny, and Gustavo A Schwartz. Characterising the glass transition temperature-structure relationship through a recurrent neural network. Journal of Non-Crystalline Solids: X, 18:100185, 2023.

[8] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks?, 2022. URL https://arxiv.org/abs/2105.14491.

[9] Alice Capecchi, Daniel Probst, and Jean-Louis Reymond. One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. Journal of Cheminformatics, 12, 2020. URL https://api.semanticscholar.org/CorpusID:219589988.

[10] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, page 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

[11] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction, 2020. URL https://arxiv.org/abs/2010.09885.

[12] Sunho Choi, Joonbum Lee, Jangwon Seo, Sung Won Han, Sang Hyun Lee, Ji-Hun Seo, and Junhee Seok. Automated bigsmiles conversion workflow and dataset for homopolymeric macromolecules. Scientific data, 11(1):371, 2024.

[13] Gabriele Corso, Hannes Stark, Stefanie Jegelka, Tommi Jaakkola, and Regina Barzilay. Graph neural networks. Nature Reviews Methods Primers, 4(1):17, 2024.

[14] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of mdl keys for use in drug discovery. Journal of chemical information and computer sciences, 42 6:1273–80, 2002. URL https://api.semanticscholar.org/CorpusID:22752474.

[15] Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data, 2020. URL https://arxiv.org/abs/2003.06505.

[16] Lev Sarkisov*Rocio Bueno-PerezMythili SutharsonDavid Fairen-Jimenez. Materials informatics with poreblazer v4.0 and the csd mof database. Chemistry of Materials, 32(23), 2020.

[17] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. Codebert: A pre-trained model for programming and natural languages, 2020. URL https://arxiv.org/abs/2002.08155.

[18] Michael E Fortunato and Coray M Colina. pysimm: A python package for simulation of molecular systems. SoftwareX, 6:7–12, 2017.

[19] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. Nucleic acids research, 40(D1):D1100–D1107, 2012.

[20] Yoshihiro Hayashi, Junichiro Shiomi, Junko Morikawa, and Ryo Yoshida. Radonpy: automated physical property calculation using all-atom classical molecular dynamics simulations for polymer informatics. npj Computational Materials, 8(1):222, 2022.

[21] Roger W Hockney and James W Eastwood. Computer simulation using particles. crc Press, 2021.

[22] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second, 2023. URL https://arxiv.org/abs/2207.01848.

[23] Bingyin Hu, Anqi Lin, and L Catherine Brinson. Chemprops: A restful api enabled database for composite polymer name standardization. Journal of cheminformatics, 13(1):22, 2021.

[24] Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. Ogb-lsc: A large-scale challenge for machine learning on graphs. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021.

[25] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. APL materials, 1(1), 2013.

[26] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30, 2017.

[27] Matthew K Kiesewetter, Eun Ji Shin, James L Hedrick, and Robert M Waymouth. Organocatalysis: opportunities and challenges for polymer synthesis. Macromolecules, 43(5):2093–2107, 2010.

[28] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. Nucleic acids research, 47(D1):D1102–D1109, 2019.

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

[30] Greg Landrum. Rdkit: Open-source cheminformatics. http://www.rdkit.org, 2013.

[31] Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 1069–1078, 2022.

[32] Ruimin Ma and Tengfei Luo. Pi1m: a benchmark database for polymer informatics. Journal of Chemical Information and Modeling, 60(10):4684–4690, 2020.

[33] Ruimin Ma, Hanfeng Zhang, Jiaxin Xu, Luning Sun, Yoshihiro Hayashi, Ryo Yoshida, Junichiro Shiomi, Jian xun Wang, and Tengfei Luo. Machine learning-assisted exploration of thermally conductive polymers based on high-throughput molecular dynamics simulations. Materials Today Physics, 28:100850, 2022. ISSN 2542-5293.

[34] Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. Nature, 624(7990):80–85, 2023.

[35] Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. Journal of Cheminformatics, 10, 2018. URL `https://api.semanticscholar.org/CorpusID:3726588`.

[36] Nilakantan, R, Bauman, N, JS Dixon, and R Venkataraghavan. Topological torsion: a new molecular descriptor for sar applications. Journal of Chemical Information and Computer Sciences, 27(2):82–85, 1987. doi: 10.1021/ci00053a005.

[37] Shingo Otsuka, Isao Kuwajima, Junko Hosoya, Yibin Xu, and Masayoshi Yamazaki. Poly-info: Polymer database for polymeric materials design. In 2011 International Conference on Emerging Intelligent Data and Web Technologies, pages 22–29. IEEE, 2011.

[38] Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2311–2320, 2020.

[39] Paul N. Patrone, Andrew Dienstfrey, Andrea R. Browning, and Samuel Tuckerand Stephen Christensen. Uncertainty quanti cation in molecular dynamics studies of the glass transition temperature. Polymer, 87, 2016.

[40] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. Journal of computational physics, 117(1):1–19, 1995.

[41] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features, 2019. URL `https://arxiv.org/abs/1706.09516`.

[42] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. Journal of chemical information and modeling, 50(5):742–754, 2010.

[43] Jean-Paul Ryckaert, Giovanni Ciccotti, and Herman JC Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. Journal of computational physics, 23(3):327–341, 1977.

[44] David S Sholl and Ryan P Lively. Seven chemical separations to change the world. Nature, 532 (7600):435–437, 2016.

[45] Huai Sun, Jon R. Maple Stephen J. Mumby, and Arnold T. Hagler. An ab initio cff93 all-atom force field for polycarbonates. Journal of the American Chemical Society, 116(7):2978–2987, 1994.

[46] Lei Tao, Vikas Varshney, and Ying Li. Benchmarking machine learning models for polymer informatics: an example of glass transition temperature. Journal of Chemical Information and Modeling, 61(11):5395–5413, 2021.

[47] Lei Tao, Jinlong He, Tom Arbaugh, Jeffrey R McCutcheon, and Ying Li. Machine learning prediction on the fractional free volume of polymer membranes. Journal of Membrane Science, 665:121131, 2023.

[48] Dario Vassetti, Marco Pagliai, and Piero Procacci. Assessment of gaff2 and opls-aa general force fields in combination with the water models tip3p, spce, and opc3 for the solvation free energy of druglike organic molecules. Journal of chemical theory and computation, 15(3): 1983–1995, 2019.

[49] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL `https://arxiv.org/abs/2412.13663`.

[50] 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.

[51] Tzu-Tsung Wong. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern recognition, 48(9):2839–2846, 2015.

[52] Yihan Zhu, Gang Liu, Eric Inae, Tengfei Luo, and Meng Jiang. Learning repetition-invariant representations for polymer informatics, 2025. URL `https://arxiv.org/abs/2505.10726`.