Macromolecular Rapid Communications

SPECIAL ISSUE Artificial Intelligence in Polymer Science

Macromolecular Rapid Communications
www.mrc-journal.de

RESEARCH ARTICLE OPEN ACCESS

Benchmarking Large Language Models for Polymer Property Predictions

Sonakshi Gupta[1] | Akhlak Mahmood[2] | Shivank Shukla[2] | Rampi Ramprasad[2]

[1]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA | [2]School of Materials Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Correspondence: Rampi Ramprasad (rampi.ramprasad@mse.gatech.edu)

Received: 1 May 2025 | **Revised:** 20 August 2025 | **Accepted:** 24 September 2025

Funding: This work was supported by the Office of Naval Research through grants N00014-19-1-2103 and N00014-20-1-2175.

Keywords: large language model | materials informatics | polymer | property prediction

ABSTRACT

Machine learning and artificial intelligence have revolutionized polymer science by enhancing the ability to rapidly predict key polymer properties and enabling generative design. The utilization of large language models (LLMs) in polymer informatics may offers additional opportunities for advancement. Unlike traditional methods that depend on large labeled datasets, hand-crafted representations of the materials, and complex feature engineering, LLM-based methods utilize natural language inputs via a transfer learning process and eliminate the need for complex representation and fingerprinting, thus significantly simplifying the training process. In this study, we fine-tune general-purpose LLMs—open-source Llama-3-8B and commercial GPT-3.5-on a curated dataset of 11,740 entries to predict key thermal properties: glass transition (T_g), melting (T_m), and decomposition (T_d) temperatures. Using parameter-efficient fine-tuning and hyperparameter optimization, we benchmark these models against traditional fingerprinting-based approaches including Polymer Genome, polyGNN, and polyBERT, under both single-task (ST) and multi-task (MT) learning frameworks. We find that while LLM informatics techniques can come close to traditional methods, they generally underperform in terms of predictive accuracy and computational efficiency. The fine-tuned Llama-3 model consistently outperforms GPT-3.5, likely due to the flexibility and tunability of the open-source architecture. Additionally, ST learning proves more effective than MT for LLMs, which struggle to exploit cross-property correlations—a significant and known advantage of traditional methods. The analysis of molecular embeddings learned by the models provides insight into the inner workings of the LLMs, revealing fundamental limitations of general-purpose LLMs in capturing nuanced chemo-structural information compared to the handcrafted features and domain-specific embeddings utilized in the traditional methods. These findings offer insights into the interplay between molecular embeddings and natural language processing, and provide guidance for LLM model selection within the context of polymer informatics.

1 | Main

Machine learning (ML) techniques are beginning to favorably impact materials property predictions and design [1, 2]. ML-assisted efforts are contributing to the design of capacitive energy storage systems [3–5], fuel cell materials [6, 7], membranes for the separation of mixture of gases and solvents [8–10], recyclable polymers [11–13], and so on.

Traditional ML approaches in polymer informatics typically follow a two-step process: first, transforming polymer structures into numerical representations known as fingerprints-essentially,

fixed-length feature vectors that encode structural and chemical characteristics of the polymer; second, applying supervised learning to predict target properties based on these representations. Over the years, numerous studies have been developed to enhance predictive performance. For instance, the hand-crafted Polymer Genome (PG) fingerprints represent polymers at three hierarchical levels-atomic, block, and chain-capturing structural details across multiple length scales [14]. Graph-based methods like polyGNN employ molecular graphs to learn polymer embeddings, effectively balancing prediction speed and accuracy [15–18]. Transformer-based models such as polyBERT utilize the linguistic structure of SMILES (Simplified Molecular-Input Line-Entry System) strings [19], using an adaptation for polymers in which asterisks (*) mark the endpoints of repeating units, allowing for precise encoding of polymer structures [20–23].

Once the materials' structures are fingerprinted, in the next stage, these representations are used to train downstream models in a supervised fashion that learn the relationships between structure and properties. Supervised learning algorithms can range from simple linear regression models to complex deep learning algorithms, such as Gaussian process regression (GPR), artificial neural networks (NNs), and graph neural networks (GNNs), tailored to specific properties and applications to ensure target performance and predictive accuracy. Furthermore, the performance of ML methods is often enhanced by multitask learning (MT), which uses correlations between multiple properties, allowing the model to learn them simultaneously even when smaller amount of data is available [24].

Recently, LLMs have emerged as a promising tool for a range of tasks in materials science - from extracting information from literature [25], predicting polymer solubility in solvents [26], to generating crystal structures [27, 28]. LLM-generated embeddings have also demonstrated strong performance on high-dimensional regression tasks [29], providing a promising pathway to improve the predictive accuracy for molecular properties. When fine-tuned on materials-specific datasets, the models can interpret SMILES strings and predict properties directly from text, eliminating the need for handcrafted or graph-based fingerprints while offering scalability and simplicity. Recent work has shown substantial advances in molecular property prediction using LLMs [30–34].

In polymer informatics, however, the potential of LLMs for property prediction remains largely unexplored. Various challenges arise because of the large size, repeating units, and structural complexity of polymers, which differ significantly from those of small molecules. Additionally, polymer property data are scarce compared to the available molecular databases, potentially limiting the ability of models to generalize across diverse polymer chemistries. Even for widely reported thermal properties, such as glass transition temperature (T_g), melting temperature (T_m), and decomposition temperature (T_d), the available datasets are only a fraction of what is available for molecules to build reliable and generalizable predictive models. As a result, the predictive performance, strengths and limitations of LLMs compared to traditional ML approaches in polymer informatics are not fully understood.

In this work, we investigate the use of LLMs for polymer property prediction by fine-tuning Meta AI's open-source Llama-3-8B-Instruct [35] and OpenAI's closed-source GPT-3.5 and GPT-4 models to predict key thermal properties-T_g, T_m, and T_d. We show that the general purpose, pre-trained, LLMs can be fine-tuned to predict polymer properties directly from the SMILES strings, eliminating the need for handcrafted or graph-based fingerprints. We evaluate LLMs under three training strategies: single-task, multi-task, and continual learning, and perform a comprehensive comparison against traditional machine learning (ML) baselines to assess their respective strengths and limitations in the context of polymer informatics. Our results show that fine-tuned LLMs achieve predictive performance that approach (but do not surpass) classical ML methods, with the Llama-3 models generally outperforming GPT-3.5 - largely due to the flexibility of open-source fine-tuning and more effective hyperparameter optimization. Notably, we find that LLMs can jointly learn both polymer embeddings and structure–property relationships during fine-tuning, removing the need for a separate fingerprinting stage. To evaluate the quality of these learned embeddings, we compare LLM-derived representations with domain-specific alternatives, including polyBERT, polyGNN, and Polymer Genome, assessing their effectiveness in capturing polymer structure-property relationships. Through this comparative study, we provide critical insights into the feasibility, advantages, and limitations of LLMs for polymer property prediction tasks, highlighting their potential as scalable alternatives to traditional ML models.

2 | Results and Discussion

2.1 | Overview of the Property Prediction Pipeline

In the polymer literature, thermal properties are the most frequently reported properties of homopolymers. To train and assess the performance of LLMs for property prediction, we manually curated a benchmark dataset of experimental thermal property values containing the SMILES string and property values. The curated dataset contains 5,253 Glass Transition Temperature (T_g), 2,171 Melting Temperature (T_m), and 4,316 Thermal Decomposition Temperature (T_d) values, as summarized in Table 1. Details regarding the source of the data can be found in the Data Availability section.

Polymer structures in the dataset are represented using SMILES, which provides a detailed and machine-readable format for polymer representations. However, a given polymer can have multiple syntactic variants of the SMILES string. To address this non-uniqueness, we performed canonicalization of SMILES to ensure a standardized and consistent representation for each unique polymer structure.

The dataset was subsequently transformed into an instruction-tuning format for fine-tuning LLMs. Since prompt design critically influences the performance of LLMs, a systematic prompt optimization process was employed to determine the most effective structure. The final prompt, which demonstrated superior accuracy, was structured as follows:

> **User**: `If the SMILES of a polymer is <SMILES>, what is its <property>?`
>
> **Assistant**: `smiles: <SMILES>, <property>: <value> <unit>`

TABLE 1 | Benchmark thermal property datasets used to train the ML models and finetune the LLMs.

Property	Range [K]	Number of data points
Glass transition temperature	[80.0, 873.0]	5,253
Melting temperature	[226.0, 860.0]	2,171
Thermal decomposition temperature	[291.0, 1167.0]	4,316
Total		11,740

where the <SMILES>, <property>, <value> and <unit> placeholders were replaced by the actual SMILES string, property name, value and unit from the dataset for each row. The additional variations of the different prompts evaluated are listed in the Supporting Information and the corresponding performance are plotted in Figure S1 (Supporting Information). An analysis of the token length distributions of both the SMILES dataset and the instruction-tuned inputs across all properties, using the OpenAI and LLama tokenizers, is provided in the Supporting Information. This analysis offers insight into the typical sequence lengths encountered during training and inference.

Model availability and architecture play a significant role in the performance of the LLMs. Open-source models such as Meta AI's Llama-3 provide greater flexibility for customization and control during fine-tuning, but require significant computational resources available on premise. OpenAI's GPT models, on the other end, offer ease of training and prediction with the utilization of Application Programming Interfaces (APIs), but are constrained by limited control over the hyperparameters one can tune and the black-box nature of the employed fine-tuning method. Understanding these trade-offs is essential for determining the most suitable approach for different polymer informatics tasks, balancing customization, computational efficiency, and ease of use. We therefore fine-tuned both Llama-3 and GPT-3.5 models using the instruction-formatted dataset. For Llama-3, fine-tuning was carried out on in-house servers utilizing Low-Rank Adaptation (LoRA) [36], a parameter-efficient method that approximates large pre-trained weight matrices with smaller, trainable matrices. This approach significantly reduced computational overhead, accelerated training times, and lowered memory usage while preserving model performance. We optimized a range of hyperparameters, including the rank (r), scaling factor (α), number of epochs, and softmax temperature (T) during inference to achieve the best results. In contrast, the fine-tuning of GPT-3.5 models was performed using the OpenAI API. Due to limited control over the fine-tuning process, the optimization of GPT-3.5 models was restricted only to the number of training epochs and the inference temperature, T. Additionally, the exact mechanism of the fine-tuning technique and model architecture implemented by OpenAI is not publicly available.

To comprehensively evaluate the performance of the models, we implemented three learning strategies: Single-Task (ST), Multi-Task (MT), and Multi-Task Sequential (MT-Seq). In the Single-Task framework, separate models were independently fine-tuned for each property, (T_g, T_m, and T_d), focusing exclusively on a single target at a time. The MT framework, by contrast, involved simultaneously fine-tuning a single model on all three properties, allowing the model to leverage shared parameters and capture inherent correlations across the properties. By leveraging the correlations between properties, multitask framework reduces the risk of overfitting to any single property and improve predictive accuracy across all properties. This approach is supported by prior ML-based studies with deep neural networks [24] and graph neural network models [15]. The MT-Seq approach, inspired by the continual or transfer learning approach, extended this idea by fine-tuning a single model in a stepwise manner, beginning with T_g, then T_m, and finally T_d, aiming to build on the knowledge gained from earlier tasks to improve adaptation and performance on the subsequent ones. A continual fine-tuning could enable progressive adaptation to later tasks hopefully without "forgetting" earlier ones, thus assessing the model's ability to retain and build upon previously acquired knowledge. Our overall finetuning workflow is illustrated in Figure 1.

2.2 | Optimization of Hyperparameters

2.2.1 | GPT Optimization

OpenAI's GPT-4o was initially tested by fine-tuning on the T_g dataset using the ST method. However, the performance of GPT-4o was comparable to that of GPT-3.5 (Figure S4, Supporting Information), with no notable improvement in predictive accuracy. We quantified the performance of the fine-tuned model by calculating the RMSE of the predictions made on the held-out test split of the dataset. Given this similarity and performance of GPT-4o, we opted to proceed with GPT-3.5 for subsequent experiments due to its cost-effectiveness. We optimized GPT-3.5 by testing different numbers of epochs (5 and 10) and the softmax temperatures ($T = 0.5$ and $T = 0.8$) during the inference stage. Parity plots for all fine-tuned models, corresponding to T_g, T_m, and T_d, are available in the Supporting Information. The best performance was achieved when GPT-3.5 was fine-tuned for 5 epochs and inference was conducted with $T = 0.5$. Using these optimized hyperparameters, we also trained a multitask thermal property model (GPT-MT) and a multitask sequential model (GPT-MT-Seq) on the three thermal properties. The final results of the fine-tuned GPT-3.5 models are detailed in the Performance section, where they are compared with Llama models across all tasks and training strategies.

2.2.2 | Llama-3 Optimization

The initial fine-tuning of the Llama-3-8B-Instruct model involved optimizing multiple additional hyperparameters, namely, alpha

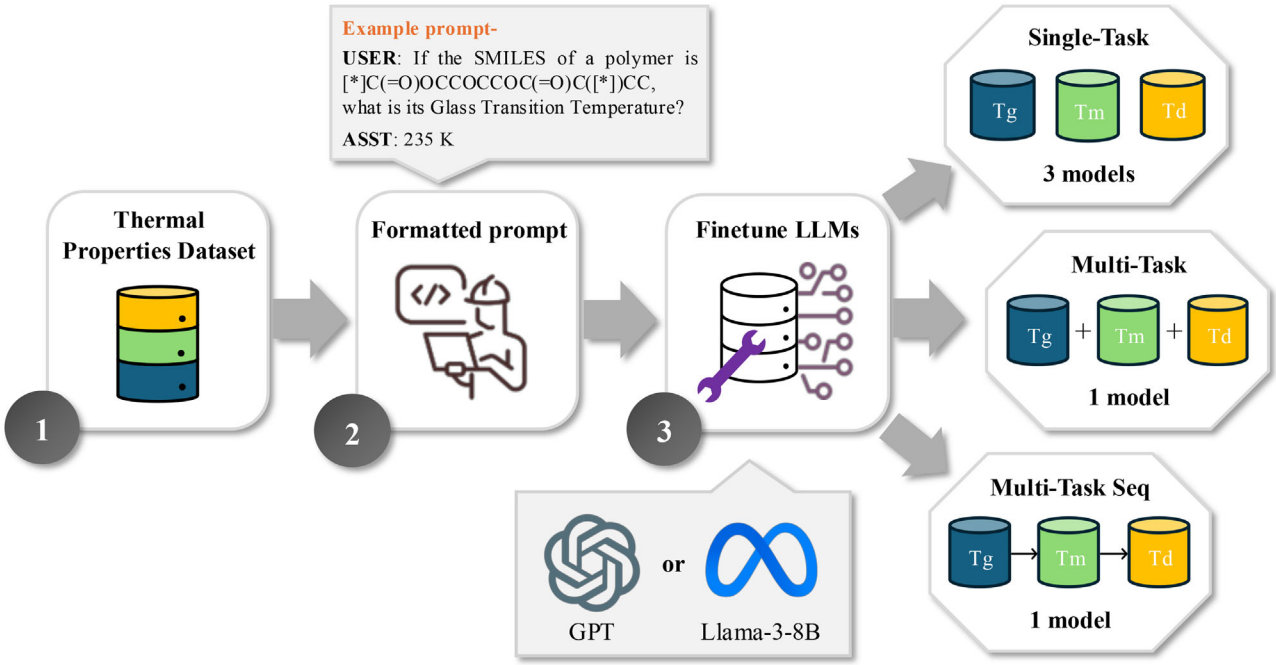

FIGURE 1 | Overall workflow for adapting LLMs to predict thermal properties of polymers. (1) The thermal properties dataset, including Glass Transition Temperature (T_g), Melting Temperature (T_m), and Thermal Decomposition Temperature (T_d), is curated and processed into a standardized format using SMILES. (2) The dataset is transformed into an instruction-tuning format with an optimized prompt for property prediction tasks. (3) Large language models (LLama-3 and GPT-3.5) are fine-tuned using three strategies: Single-Task (ST), where separate models are trained independently for each property; Multi-Task (MT), where a single model is simultaneously trained on all three properties; and Multi-Task Sequential (MT-Seq), where properties are learned in a stepwise manner ($T_g \rightarrow T_m \rightarrow T_d$).

(a) and rank (r), in addition to evaluating the softmax temperatures (T) of 0.5 and 0.8 during inference. When T = 0.8, the combination of $a = 16$ and $r = 16$ yielded the best performance for the T_g dataset (cf. Figure S8, Supporting Information). Increasing both alpha and rank progressively to 128 resulted in a performance decline. This configuration ($a = 16, r = 16, T = 0.8$) also performed consistently well across the T_m and T_d datasets. For T_m, similar analysis revealed two optimum configurations, (16, 16) and (32, 32), that achieved comparable performance. Similarly, for T_d, multiple configurations, including (16, 16), (32, 16), and (64, 16), delivered similar results. It is important to note that stochastic variations across multiple inference trials can lead to slight fluctuations in these values, as elaborated in the later sections. Despite this variability, the results indicate a consistent pattern, with (16, 16) emerging as the most effective hyperparameter combination for these datasets. Building on our initial hyperparameters tuning, we explored the effect of the number of epochs, increasing them from 5 to 30 in increments of 5 while keeping $a = 16$ and $r = 16$. As shown in Figure S7 (Supporting Information), RMSE initially decreased with increasing epochs, reaching an optimal value at 25 epochs. At this point, the RMSE for the T_g, T_m, and T_d datasets at $T = 0.8$ was 39.48, 56.89, and 75.79 K, respectively, with inference temperature $T = 0.8$ consistently outperforming $T = 0.5$. Interestingly, a similar RMSE minimum was observed at 5 epochs; however, upon closer inspection of the parity plots (cf. Figure S5, Supporting Information), significant clustering of the predicted values became evident, suggesting poor generalization and learning by the model despite the lower RMSE values.

This clustering phenomenon highlighted the need for an additional metric to better evaluate model performance. From our experiments, similar clustering effects were observed for both Llama-3 and GPT-3.5 models, usually prominent at lower softmax temperatures. We introduced the Bin Height Dispersion Index (BHDI), described in Methods section, to measure clustering in predictions. As shown in Figure S7 (Supporting Information), BHDI decreased with increasing epochs suggesting the absence of clustering, reaching its lowest at 25 epochs. For T_g, BHDI dropped from 4.67 at 5 epochs to 0.98 at 25 epochs. Similarly, for T_m, it decreased from 7.52 to 1.48, and for T_d, from 7.03 to 2.02. The reduction in BHDI demonstrates that 25 epochs not only minimize clustering but also promote a well-distributed range of predictions, reflecting improved model performance. These results also underscored the importance of moving beyond traditional metrics like RMSE to evaluate the distribution, variability, and generalization of predictions.

2.3 | Single-Task vs. Multi-Task Performance

The best performing fine-tuned LLM models were benchmarked against state-of-the-art (SOTA) approaches, including Polymer Genome (PG), polyGNN, and polyBERT, which primarily function as fingerprinting techniques to encode polymer structures for machine learning models. The training details for the SOTA approaches are provided in the Supporting Information.

Among the traditional informatics methods, the PG-fingerprinting approach achieved the best performance,

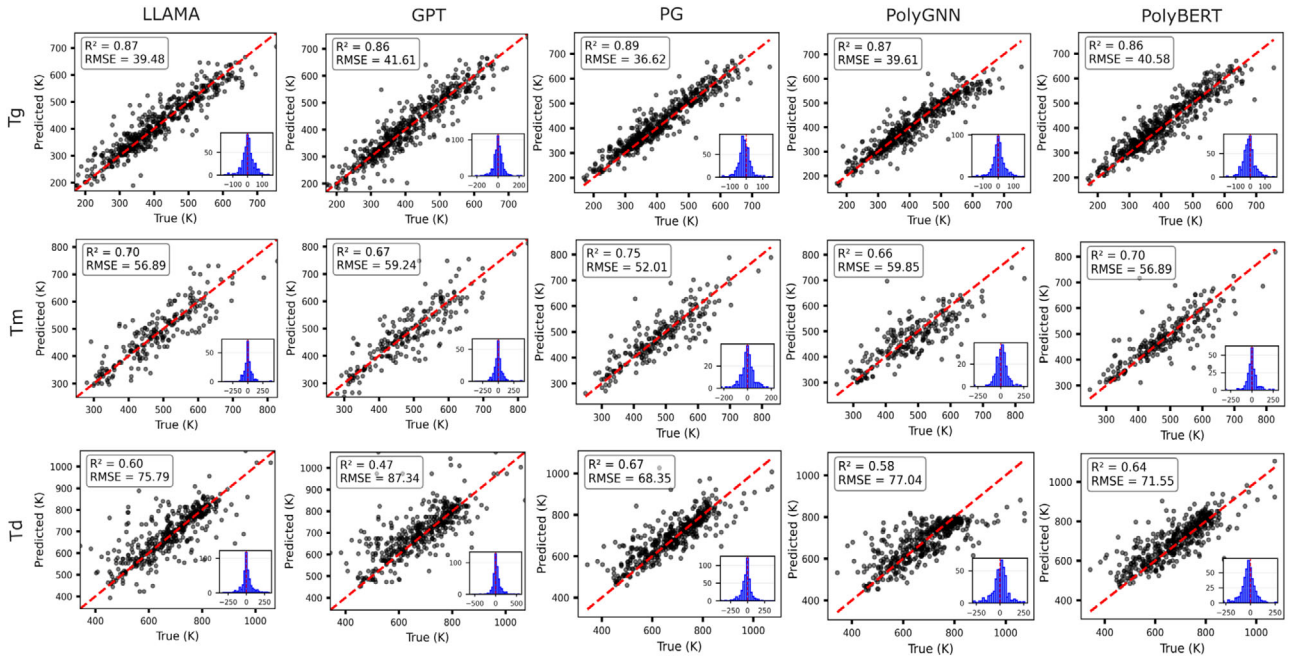

FIGURE 2 | Parity plots for T_g, T_m, T_d under ST learning using the fine-tuned LLaMa-3 model ($\alpha = 16$, $r = 16$, epochs = 25, $T = 0.8$), fine-tuned GPT-3.5 model (epochs = 5, $T = 0.5$), and traditional fingerprinting-based models: PG, polyGNN and polyBERT.

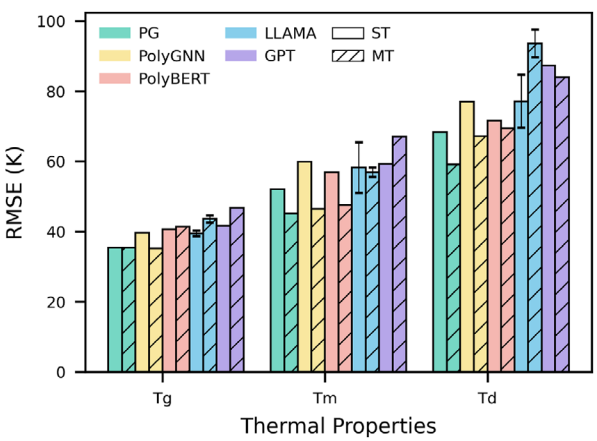

FIGURE 3 | Comparison of T_g, T_m, and T_d predictions for PG, PolyGNN, PolyBERT, and fine-tuned LLaMA-3-8B-Instruct (LoRA: $a = 16, r = 16, T = 0.8$, epochs = 25) and GPT-3.5 models (epochs = 10, $T = 0.5$) under the single-task (ST) and multi-task (MT) learning framework. Error bars for LLaMA-3 models were calculated based on three separate inference runs for each configuration. GPT-3.5 models do not include error bars, as only a single run was performed for each configuration to reduce costs.

followed closely by polyGNN. Fine-tuned Llama-3 performed on par with polyBERT, a domain-specific chemical language model, demonstrating that a general-purpose LLM can achieve competitive accuracy without requiring extensive polymer-specific pretraining. However, LLMs did not surpass traditional ML models, reinforcing the advantage of domain-specific fingerprinting and feature engineering during polymer property prediction. Figure 2 presents the corresponding parity plots.

Figure 3 presents the comparative model performance under Single-Task (ST) and Multi-Task (MT) configurations. With optimal hyperparameter settings, fine-tuned Llama-3 (a = 16, r = 16, epochs = 25, T = 0.8) consistently outperformed GPT-3.5 (epochs = 5, T = 0.5) across all datasets. The ST models for Llama-3 achieved lower RMSE values in predicting T_g, T_m, and T_d, recording 39.48, 58.23, and 77.11 K, respectively compared to 47.2, 63.8, and 80.5 K for GPT-3.5. Error bars for Llama-3 models represent variability across three independent inference runs for each configuration. GPT-3.5 models do not include error bars, as only a single run was performed for each configuration to reduce API cost. These results indicate the efficiency of the smaller, fine-tuned Llama-3 model over GPT-3.5 under single-task learning for thermal property prediction.

In addition to general-purpose models, ChemLLM [37], a domain-specific LLM, instruction-tuned on materials science corpora, was also evaluated. The model was fine-tuned under a single-task setup using the same optimized hyperparameter configuration as Llama-3 ($\alpha = 16$, $r = 16$, epochs = 25, $T = 0.8$). As seen from the parity plots in Figure **S13**, ChemLLM captured the overall distributional trends reasonably well but did not outperform either Llama-3 or GPT-3.5, yielding higher RMSE values across all three thermal properties. This highlights that further fine-tuning of a domain-specialized model does not necessarily translate to improved regression performance when compared to well-optimized general-purpose LLMs.

Under the Multi-Task (MT) configuration, neither Llama-3 nor GPT-3.5 demonstrated substantial improvements over their ST counterparts. For both T_g and T_m, the RMSE values were similar to those of the ST models, though performance for T_d showed a degradation in accuracy. As shown in Figure 3, PG, PolyGNN, and PolyBERT benefited from MT learning, while Llama-3 and GPT-3.5 models failed to achieve similar improvements, suggesting that LLMs struggle to efficiently share learned representations across multiple polymer properties. This is a notable divergence

of the LLMs from the trends observed in traditional ML-based methods, where multi-task learning typically enhances predictive performance by leveraging cross-property correlations [15, 20].

Interestingly, the fine-tuned Llama-3 models consistently achieved lower RMSE values than fine-tuned GPT-3.5 models, highlighting its greater adaptability. While the GPT-3.5 models required fewer epochs and incurred lower on-premise computational costs, its black-box nature and limited hyperparameter tuning options restricted its optimization potential. In contrast, Llama's open-source flexibility enabled more effective fine-tuning, which potentially led to a better predictive accuracy. Ultimately, the advantage of LLMs lies in their scalability and ease of fine-tuning compared to traditional ML models, which require extensive domain-specific customization and computationally expensive pretraining. However, domain-specific approaches still hold an edge in predictive accuracy, emphasizing the need for future research to integrate the strengths of LLMs with specialized polymer informatics models.

2.4 | Continual Learning and Forgetfulness of LLMs

To explore the feasibility of continual learning, referred to in this paper as MT-Seq, we tested a method where models were fine-tuned iteratively to study a hypothetical scenario where additional data or data for new properties become available at a later time. The goal was to evaluate whether a model could retain previously learned knowledge while integrating new information. Two tests were conducted to investigate this.

In the first test, sequential training was applied to the T_g dataset only. The dataset was split into 90% training and 10% testing data. The training set was further divided into two halves, with the model being fine-tuned iteratively: the first iteration involved fine-tuning on the first half of the training data, and the second iteration fine-tuned the model using the remaining half. The same testing dataset was used to evaluate performance across iterations. As shown in Figure 4a, the fine-tuned Llama-3 model achieved RMSE of 60.76 K after the first iteration and 56.81 K after the second iteration, demonstrating improved performance as additional data was introduced. Similarly, for GPT-3.5, a similar improvement in performance was noted, where the RMSE slightly decreases from 43.73 K in the first iteration to 42.22 K in the second.

The second test extended this sequential training approach to the entire thermal properties dataset, where LLMs were first fine-tuned on the (T_g dataset, next on the T_m dataset, and finally on the T_d) dataset. Both GPT-3.5 and Llama-3 showed degraded performance compared to the previously discussed ST and MT training configurations. For the GPT-MT-Seq configuration, RMSE values for T_g and T_m significantly increased during the later stages of training, indicating that the model failed to retain previously learned information. However, for T_d, GPT's performance remained comparable to the MT and ST configurations, potentially due to overfitting on the T_d dataset. Interestingly, GPT-3.5 exhibited an intuitive trend where the oldest dataset (T_g) suffered the greatest forgetting, while information about more recent dataset, (T_m) were retained to a greater degree. In contrast, Llama-MT-Seq showed consistent performance degradation across all three datasets, without any clear retention of recent data, further highlighting its susceptibility to the phenomenon known in computer science as 'catastrophic forgetting'.

These results underscore the limitations of continual learning in LLMs, with both models exhibiting "catastrophic forgetting" when fine-tuned iteratively on datasets. These findings align with existing literature, where continual fine-tuning has shown to result in similar forgetting behavior [38]. This phenomenon, also known as the palimpsest effect, occurs when newly learned information overwrites previously acquired knowledge [39].

2.5 | Effects of Polymer Representations via Embeddings

Leveraging molecular embeddings as input feature vectors is essential to achieve accurate predictions of chemical properties using traditional machine learning [40]. For embeddings to be effective, they must not only capture the intricate chemo-structural attributes of materials but also remain robust to transformations that leave the material's physical state unchanged. Over the years, a variety of approaches, such as PG, polyGNN and polyBERT fingerprinting, have been developed to generate such robust embeddings [15, 20, 41].

To assess the chemical understanding captured by traditional fingerprinting methods and general-purpose LLM embeddings, we compared fingerprints generated using PG, polyGNN, polyBERT with those produced by OpenAI's text-embedding-3-small model, and Llama-3-8B embeddings. For Llama-3, the embeddings were extracted from the output of the model's final layer. The quality of these embeddings was assessed based on their ability to differentiate between distinct polymer structural features, including functional groups, cyclic vs non-cyclic structures, and aromatic vs non-aromatic compounds. Figure 5 showcases scatter plots of 2D t-SNE projections of embeddings for three comparisons: a) amide vs. ester functional groups, b) cyclic vs. non-cyclic structures, and c) aromatic vs. non-aromatic compounds. Polymers with amide and ester functional groups were selected for analysis due to their substantial representation in the dataset, with 1920 and 1457 data points, respectively, ensuring a fair comparison across methods. Similarly, cyclic and aromatic classes were chosen for their relevance in structural and chemical diversity, providing additional insight into the embedding quality. Centroids for each cluster, calculated as the mean of all data points within a cluster, are also included for clarity.

Figure 5a illustrates the embeddings derived for polymers with amide and ester functional groups. The PG fingerprints exhibit distinct and well-separated clusters, demonstrating the effectiveness of handcrafted features in capturing the chemo-structural attributes of these polymer classes. PolyGNN and PolyBERT embeddings also show clustering for the two classes, though the separation is less pronounced compared to PG. Llama-3 embeddings, while not as refined as the domain-specific methods, still reveal evident clustering, showcasing its capability to capture underlying chemical and structural differences from SMILES representations. In contrast, embeddings generated by

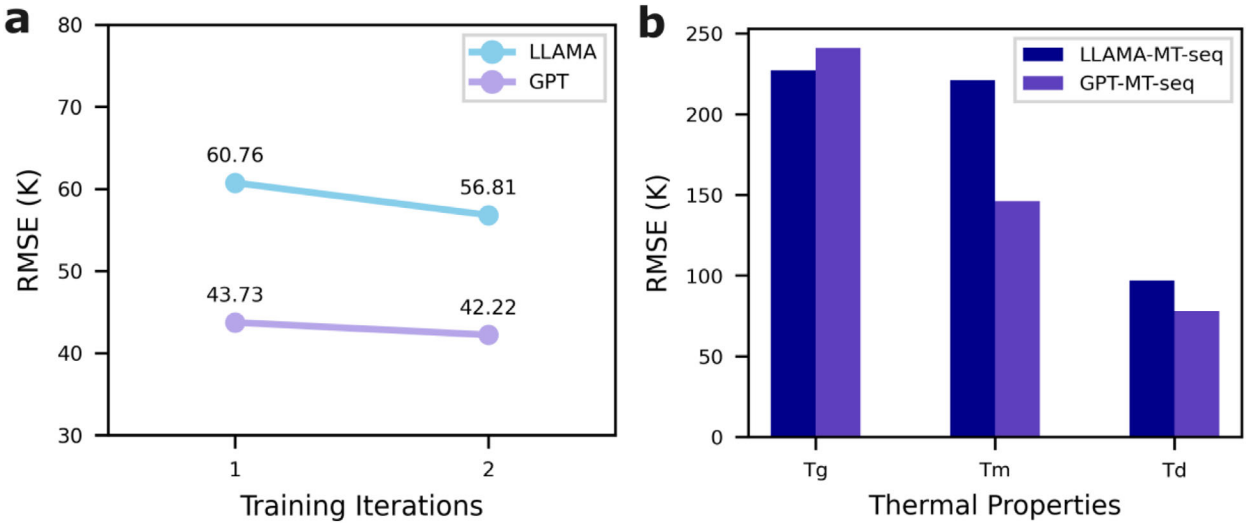

FIGURE 4 | Performance evaluation of continual fine-tuning experiments. (a) RMSE values for the T_g dataset across two iterations of fine-tuning using the LLaMA-3 and GPT-3.5 models. (b) RMSE values for continual fine-tuning on T_g, T_m, and T_d datasets under the Multi-Task-Sequential (MT-Seq) configuration, comparing LLaMA-3 and GPT-3.5 models.

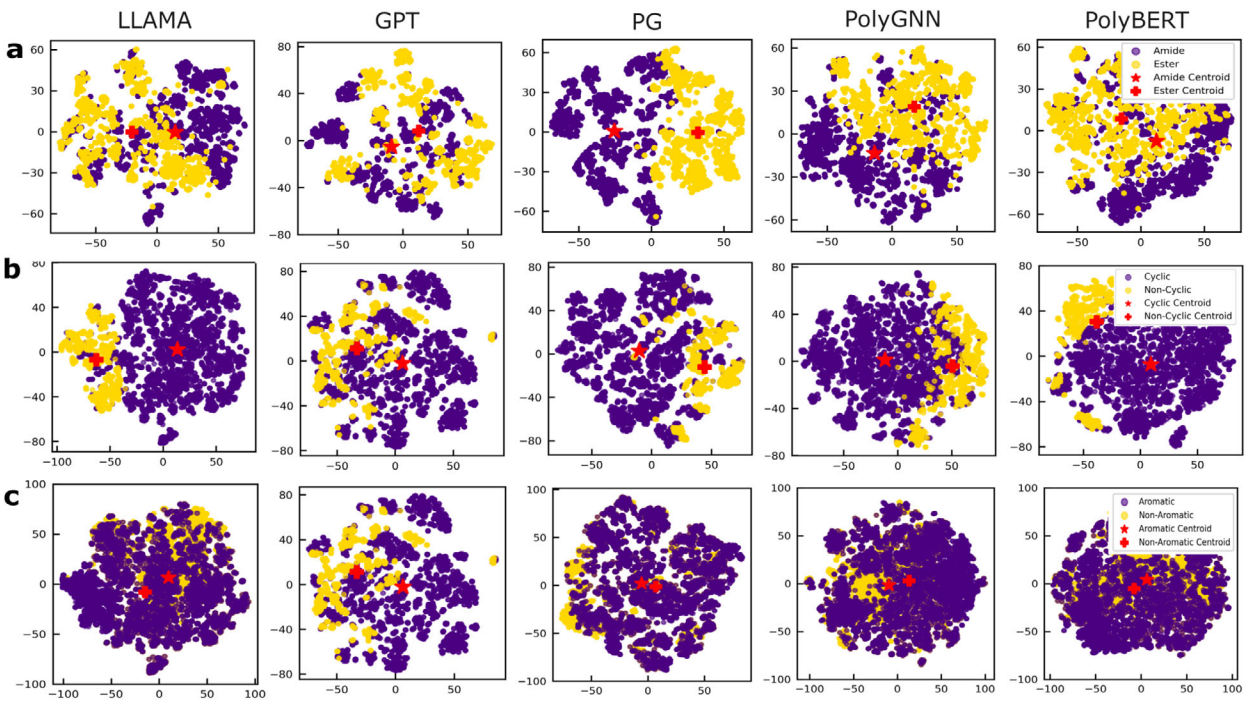

FIGURE 5 | 2D t-SNE projections of polymer embeddings, showcasing: (a) amide vs. ester functional groups, (b) cyclic vs. non-cyclic polymers, and (c) aromatic vs. non-aromatic polymers.

GPT display minimal separation between the two functional groups, indicating limited clustering. This suggests that GPT struggles to distinguish between these chemical classes, elucidating a deficiency in its understanding of chemical relationships when compared to both handcrafted fingerprints and domain-specific embeddings.

Figure 5b showcases embeddings for cyclic vs. non-cyclic polymers, with the PG fingerprinting taken as the reference due to its superior performance across tests. Both PG and Llama-3 embeddings demonstrate a clear separation between the two classes, effectively capturing the structural distinctions. In contrast, GPT embeddings show significant overlap and interspersed clusters, highlighting its potentially limited ability to differentiate between cyclic and non-cyclic polymers. Similarly, Figure 5c compares aromatic and non-aromatic compounds. All models exhibit overlapping clusters in this case, reflecting the inherent difficulty of distinguishing between these two classes in a 2D representation. While Llama-3 embeddings provide slightly better separation than GPT, the overlap remains significant. Distance between the centroids of the clusters further emphasizes these overlaps, illustrating the challenges and limitations of these embeddings in effectively capturing chemical distinctions in this context. These observations are consistent with prior studies,

where Llama-2 demonstrated superior performance in generating molecular embeddings from SMILES strings compared to GPT's text-small-3-embedding's embedding model [42].

Overall, these results emphasize Llama-3's ability to generate embeddings which better capture chemical and structural distinctions compared to GPT-3.5, particularly for distinct classifications like cyclic vs. non-cyclic polymers. However, the findings also highlight the importance of chemically informed embeddings, such as those generated by PG and domain-specific ML models, in achieving superior representation of polymers.

3 | Conclusion

In this study, we investigated the capabilities of LLMs in polymer informatics, benchmarking their performance against traditional domain-specific approaches. The findings demonstrate that while LLMs are effective for property predictions of polymers, their performance is significantly shaped by task-specific requirements, model configurations and the quality of the learned embeddings. Notably, while LLMs can reach the performance of the SOTA ML models, they are not able to surpass them, highlighting the ongoing need for optimization and hybrid approaches. Notably, ST learning methods consistently outperformed MT approaches using LLMs, a deviation from the trends seen in the traditional ML models. This suggests that LLMs benefit from specialized tuning for individual properties rather than relying on shared parameter spaces across multiple properties. Another limitation of LLM finetuning is catastrophic forgetting- a significant challenge when integrating new data without compromising previously learned knowledge. These findings highlight the need for customizing training strategies to align with both the model architecture and dataset characteristics.

A comparison between GPT-3.5 and Llama-3 revealed that the smaller LLama-3 model achieved superior performance, showcasing both its efficiency and adaptability. However, analysis of RMSE parity plots exposed limitations such as clustering and reduced variability in predictions. These findings emphasize the inadequacy of relying solely on traditional metrics like RMSE to evaluate performance. Advanced metrics, such as the Bin Height Dispersion Index (BHDI) introduced here, enabled a deeper and more nuanced assessment of model predictions.

Embeddings emerged as a critical factor in model performance. Handcrafted fingerprints and domain-specific embeddings excelled at capturing chemo-structural details, leading to more accurate property predictions. In contrast, GPT embeddings exhibited insufficient chemical understanding, leading to poor differentiation and clustering of polymer classes. Llama-3's embeddings, on the other hand, demonstrated greater consistency and effectively captured critical aspects of polymer chemistry. This advantage, coupled with its flexibility in hyperparameter tuning, likely contributed to Llama-3's superior performance over GPT-3.5.

Overall, while LLMs have demonstrated substantial promise in polymer informatics, their current performance suggests they provide ease rather than accuracy relative to SoTA traditional models, atleast for the type of tests considered here. Future research should focus on integrating domain knowledge with LLM-driven approaches, refining training methodologies, and improving embedding strategies to enhance their generalizability and predictive power in polymer property prediction.

4 | Methods

Prompt Optimization To evaluate the impact of prompt design on model performance, we conducted prompt optimization experiments using GPT-3.5 on a dataset for T_g. Given the cost constraints associated with using GPT-3.5, a 10/90 data split was employed, with only 10% of the dataset used for training and 90% for testing. Inference was performed with a temperature (T) of 0.5. These considerations were made to maximize insights while minimizing computational expenses. Five different prompt designs were tested, each varying in structure and the specificity of the query-response format. These prompts were carefully crafted to assess their influence on the model's ability to generate accurate predictions, as measured by Root Mean Squared Error (RMSE).

The results, as shown in Figure S5 (Supporting Information), indicate significant variability in RMSE across the prompts. Prompt 01, which employed a straightforward and well-structured question-response format, achieved the lowest RMSE, highlighting its effectiveness in eliciting accurate predictions. In contrast, Prompt 04, designed to handle uncertainty, resulted in the highest RMSE, likely due to the added complexity and potential for generating ambiguous responses. Similarly, Prompt 03 showed a relatively high RMSE, which may be attributed to its overly simplistic format that asked the model to predict only a numeric value. These findings emphasize the importance of prompt design in optimizing the predictive capabilities of LLMs. A well-balanced prompt structure, as demonstrated by Prompt 01, structured as:

> **User**: `If the SMILES of a polymer is <SMILES>, what is its <property>?`
>
> **Assistant**: `{smiles: <SMILES>, <property>: <value> <unit>}`

enables the model to leverage its contextual understanding effectively while maintaining accuracy.

Llama Model The Meta Llama-3-8B-Instruct model was employed for fine-tuning, leveraging pretrained weights available on the Hugging Face hub via the Transformers library. To enhance computational and memory efficiency parameter efficient fine-tuning was performed with Low-Rank Adaptation (LoRA) [36] method on two NVIDIA L40S GPUs (46 GB, 350 W) hosted on our in-house servers. Hyperparameter optimization included variations in alpha and rank values (16, 32, 64, and 128), with the number of epochs tested incrementally from 5 to 30. Inference temperatures of 0.5 and 0.8 were also explored to achieve optimal performance.

GPT Model The GPT-3.5-turbo-0125 model was fine-tuned through the OpenAI API, accessed via the OpenAI Python package. Although GPT-4o-2024-08-06 was initially tested, its performance was comparable to GPT-3.5, leading to the selection

of the latter due to its cost efficiency. Performance comparisons between the two models are detailed in the Supporting Information. The use of an API-based approach streamlines deployment but limits flexibility in hyperparameter control. Optimization focused on the number of epochs (varied between 5 and 10) and inference temperatures (evaluated at 0.5 and 0.8).

Evaluation Metrics Model performance was evaluated using the coefficient of determination (R^2) and the root mean square error ($RMSE$), standard metrics in materials science for assessing precision and the fit between predictions and ground truth. However, these metrics fail to capture the distributional characteristics and clustering of the predictions. To address the gap, we introduce Bin Height Dispersion Index (BHDI), a complementary metric designed to assess the uniformity and variability of model predictions. BHDI quantifies the clustering of predicted values, often observed as horizontal lines in parity plots or sharp spikes in Predicted vs. Ground Truth distribution plots, offering deeper insights into prediction diversity and enhancing the overall evaluation of model performance.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

Bin Height Dispersion Index Bin Height Dispersion Index (BHDI) quantifies clustering by measuring the deviation of bin heights in the histogram h_{pred} of the predicted values from those in the histogram h_{true} of the ground truth values. We define BHDI as:

$$BHDI = \frac{\sum_{i=1}^{N}(h_{pred}^{i} - h_{true}^{i})^2}{\sum_{i=1}^{N} h_{true}^{i}}$$

where, N represents the total number of bins in the histograms, $h_{\text{true},i}$ is the height (number of times a property value is predicted) of the i-th bin in the ground truth histogram, and $h_{\text{pred},i}$ is the corresponding height in the predicted histogram. The denominator, $\sum_{i=1}^{N} h_{\text{true},i}$, represents the total frequency count of the ground truth histogram, ensuring normalization. A lower BHDI value reflects a well-distributed range of predictions, indicating good variability and robust generalization across the data space. In contrast, a higher BHDI value highlights significant clustering, where predictions are overly concentrated in specific bins, signaling poor generalization and limited variability. By addressing the limitations of traditional evaluation metrics, BHDI offers a more comprehensive assessment of model performance, especially in tasks like property prediction where balanced and diverse predictions are critical.

Acknowledgements

This work was supported by the Office of Naval Research through grants N00014-19-1-2103 and N00014-20-1-2175.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

1. R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, "Machine Learning in Materials Informatics: Recent Applications and Prospects," *npj Computational Materials* 3, no. 1 (2017): 54.

2. H. Tran, R. Gurnani, C. Kim, et al., "Design of Functional and Sustainable Polymers Assisted by Artificial Intelligence," *Nature Reviews Materials* 9, no. 12 (Dec. 2024): 866–886, publisher: Nature Publishing Group.

3. R. Gurnani, S. Shukla, D. Kamal, et al., "AI-Assisted Discovery of High-Temperature Dielectrics for Energy Storage," *Nature Communications* 15, no. 1 (July 2024): 6107, publisher: Nature Publishing Group.

4. J. Kern, L. Chen, C. Kim, and R. Ramprasad, "Design of Polymers for Energy Storage Capacitors Using Machine Learning and Evolutionary Algorithms," *Journal of Materials Science* 56, no. 35 (Dec. 2021): 19623–19635.

5. R. Gurnani, D. Kamal, H. Tran, et al., "polyG2G: A Novel Machine Learning Algorithm Applied to the Generative Design of Polymer Dielectrics," *Chemistry of Materials* 33, no. 17 (Sept. 2021): 7008–7016, publisher: American Chemical Society.

6. W. Schertzer, S. Shukla, A. Sose, et al., "Ai-Driven Design of Fluorine-Free Polymers for Sustainable and High-Performance Anion Exchange Membranes," *Journal of Materials Informatics* 5, no. 1 (2025): N–A.

7. Y. D. Wang, Q. Meyer, K. Tang, et al., "Large-Scale Physically Accurate Modelling of Real Proton Exchange Membrane Fuel Cell With Deep Learning," *Nature communications* 14, no. 1 (2023): 745.

8. Y. J. Lee, L. Chen, J. Nistane, et al., "Data-Driven Predictions of Complex Organic Mixture Permeation in Polymer Membranes," *Nature Communications* 14, no. 1 (2023): 4931.

9. J. Wang, K. Tian, D. Li, et al., "Machine Learning in gas Separation Membrane Developing: Ready for Prime Time," *Separation and Purification Technology* 313 (2023): 123493.

10. B. K. Phan, K.-H. Shen, R. Gurnani, H. Tran, R. Lively, and R. Ramprasad, "Gas Permeability, Diffusivity, and Solubility in Polymers: Simulation-Experiment Data Fusion and Multi-Task Machine Learning," *npj Computational Materials* 10, no. 1 (2024): 186.

11. J. Kern, Y. Su, W. Gutekunst, and R. Ramprasad, "An Informatics Framework for the Design of Sustainable, Chemically Recyclable, Synthetically-Accessible and Durable Polymers," (2024).

12. C. Atasi, J. Kern, and R. Ramprasad, "Design of Recyclable Plastics With Machine Learning and Genetic Algorithm," *Journal of Chemical Information and Modeling* 64, no. 24 (2024): 9249–9259.

13. C. Guarda, J. Caseiro, and A. Pires, "Machine Learning to Enhance Sustainable Plastics: A Review," *Journal of Cleaner Production* (2024): 143602.

14. C. Kim, A. Chandrasekaran, T. D. Huan, D. Das, and R. Ramprasad, "Polymer Genome: A Data-Powered Polymer Informatics Platform for Property Predictions," *The Journal of Physical Chemistry C* 122, no. 31 (2018): 17575–17585.

15. R. Gurnani, C. Kuenneth, A. Toland, and R. Ramprasad, "Polymer Informatics at Scale With Multitask Graph Neural Networks," *Chemistry of Materials* 35, no. 4 (Feb. 2023): 1560–1567, publisher: American Chemical Society.

16. X. Zang, X. Zhao, and B. Tang, "Hierarchical Molecular Graph Self-Supervised Learning for Property Prediction," *Communications Chemistry* 6, no. 1 (2023): 34.

17. J. Xia, Y. Zhu, Y. Du, and S. Z. Li, "Pre-Training Graph Neural Networks for Molecular Representations: Retrospect and Prospect," in *ICML 2022 2nd AI for Science Workshop* (2022).

18. Y. Wang, J. Wang, Z. Cao, and A. Barati Farimani, "Molecular Contrastive Learning of Representations via Graph Neural Networks," *Nature Machine Intelligence* 4, no. 3 (2022): 279–287.

19. D. Weininger, "Smiles, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules," *Journal of Chemical Information and Computer Sciences* 28, no. 1 (1988): 31–36.

20. C. Kuenneth and R. Ramprasad, "polyBERT: A Chemical Language Model to Enable Fully Machine-Driven Ultrafast Polymer Informatics," *Nature Communications* 14, no. 1 (July 2023): 4099, publisher: Nature Publishing Group.

21. S. Wang, Y. Guo, Y. Wang, H. Sun, and J. Huang, "Smiles-Bert: Large Scale Unsupervised Pre-Training for Molecular Property Prediction," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* (2019), 429–436.

22. J. Ross, B. Belgodere, V. Chenthamarakshan, I. Padhi, Y. Mroueh, and P. Das, "Large-Scale Chemical Language Representations Capture Molecular Structure and Properties," *Nature Machine Intelligence* 4, no. 12 (2022): 1256–1264.

23. P. Li, J. Wang, Y. Qiao, et al., "Learn Molecular Representations From Large-Scale Unlabeled Molecules for Drug Discovery," *arXiv preprint arXiv:2012.11175* (2020).

24. C. Kuenneth, A. C. Rajan, H. Tran, L. Chen, C. Kim, and R. Ramprasad, "Polymer Informatics With Multi-Task Learning," *Patterns* 2, no. 4 (Apr. 2021): 100238.

25. S. Gupta, A. Mahmood, P. Shetty, A. Adeboye, and R. Ramprasad, "Data Extraction From Polymer Literature Using Large Language Models," *Communications Materials* 5, no. 1 (2024): 269.

26. S. Agarwal, A. Mahmood, and R. Ramprasad, "Polymer Solubility Prediction Using Large Language Models," *ACS Materials Letters* 7 (2025): 2017–2023.

27. J. Gan, P. Zhong, Y. Du, et al., "Large Language Models are Innate Crystal Structure Generators," *arXiv preprint arXiv:2502.20933* (2025).

28. L. M. Antunes, K. T. Butler, and R. Grau-Crespo, "Crystal Structure Generation With Autoregressive Large Language Modeling," *Nature Communications* 15, no. 1 (2024): 1–16.

29. E. Tang, B. Yang, and X. Song, "Understanding LLM Embeddings for Regression," (Nov. 2024), arXiv:2411.14708.

30. Y. Liu, S. Ding, S. Zhou, W. Fan, and Q. Tan, "MolecularGPT: Open Large Language Model (LLM) for Few-Shot Molecular Property Prediction," (2024).

31. Z. Liu, W. Zhang, Y. Xia, et al., "Molxpt: Wrapping Molecules With Text for Generative Pre-Training," 2023.

32. C. Edwards, T. Lai, K. Ros, G. Honke, K. Cho, and H. Ji, "Translation Between Molecules and Natural Language," (2022).

33. Q. Pei, L. Wu, K. Gao, et al., "Biot5+: Towards Generalized Biological Understanding With Iupac Integration and Multi-Task Tuning," (2024).

34. Y. Liu, S. Ding, S. Zhou, W. Fan, and Q. Tan, "Moleculargpt: Open Large Language Model (llm) for few-shot Molecular Property Prediction," (2024).

35. A. E. A. Grattafiori, "The Llama 3 Herd of Models," (2024).

36. E. J. Hu, Y. Shen, P. Wallis, et al., "Lora: Low-Rank Adaptation of Large Language Models," (2021).

37. D. Zhang, W. Liu, Q. Tan, et al., "Chemllm: A Chemical Large Language Model," (2024).

38. Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang, "An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-Tuning," (2024).

39. F. Zenke and A. Laborieux, "Theories of Synaptic Memory Consolidation and Intelligent Plasticity for Continual Learning," (2024).

40. G. B. Goh, N. O. Hodas, C. Siegel, and A. Vishnu, "Smiles2vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties," *arXiv preprint arXiv:1712.02034* (2017).

41. H. Doan Tran, C. Kim, L. Chen, et al., "Machine-Learning Predictions of Polymer Properties With Polymer Genome," *Journal of Applied Physics* 128, no. 17 (Nov. 2020): 171104.

42. S. Sadeghi, A. Bui, A. Forooghi, J. Lu, and A. Ngom, "Can Large Language Models Understand Molecules?" *BMC Bioinformatics* 25, no. 1 (2024): 225.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.

Supporting File: marc70087-sup-0001-SuppMat.pdf.