# Cross-Modal Learning for Chemistry Property Prediction: Large Language Models Meet Graph Machine Learning

**Anonymous Author(s)**
Affiliation
Address
`email`

## Abstract

In the field of chemistry, the objective is to create novel molecules with desired properties, facilitating accurate property predictions for applications such as material design and drug screening. However, existing graph deep learning methods face limitations that curb their expressive power. To address this, we explore the integration of vast molecular domain knowledge from Large Language Models (LLMs) with the complementary strengths of Graph Neural Networks (GNNs) to enhance performance in property prediction tasks. We introduce a Multi-Modal Fusion (MMF) framework that synergistically harnesses the analytical prowess of GNNs and the linguistic generative and predictive abilities of LLMs, thereby improving accuracy and robustness in predicting molecular properties. Our framework combines the effectiveness of GNNs in modeling graph-structured data with the zero-shot and few-shot learning capabilities of LLMs, enabling improved predictions while reducing the risk of overfitting. Furthermore, our approach effectively addresses distributional shifts, a common challenge in real-world applications, and showcases the efficacy of learning cross-modal representations, surpassing state-of-the-art baselines on benchmark datasets for property prediction tasks.

## 1 Introduction

Deep learning has great potential for efficiently predicting molecular properties compared to traditional methods, resulting in reduced computational complexity and costs. Recently, there has been a growing surge of interest in representing organic molecules as molecular graphs. Graph Neural Networks (GNNs) can learn patterns from these highly non-linear molecular topological structures, which find applications in various domains, including drug discovery[51, 54], material design[38, 45], and environmental science[63, 49]. Several state-of-the-art graph machine learning algorithms are available in the literature, including MPNN[18], SchNet[44], E(n)-GNN[43], DimeNet++[16], and SphereNet[32]. GNNs have been extensively studied for molecular property applications; however, they suffer from bottlenecks such as limited expressive power[26], over-squashing[12], and over-smoothing[41] issues. In recent years, Large Language Models (LLMs)[5, 8, 48] have revolutionized the field of natural language processing with improved performance in various tasks, generating human-like responses, facilitating complex logical reasoning, and demonstrating capabilities in multi-tasking and multi-modal learning. Zero-Shot Chain of Thought[57](for brevity, Zero-Shot CoT) and Few-shot (In-Context) Learning[5](for brevity, Few-Shot ICL) are prompt engineering strategies for utilizing LLMs in specific linguistic tasks or related problem-solving scenarios. Zero-Shot CoT relies on task-specific instructions without demonstrations, requiring the language model to generalize from the implicit knowledge embedded within its parameters, which was learned from the training data, to generate the output. Conversely, Few-Shot ICL includes a few guiding demonstrations along with the instructions in the prompts, fostering contextual understanding and task-specific adaptation. Despite the progress in the application of foundational LLMs across various scientific disciplines, integrating LLMs with GNNs for the task of molecule property prediction remains an underexplored area. However, this presents an opportunity for innovative techniques that combine LLMs and GNNs to enhance

Submitted to the Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023. Do not distribute.

property prediction applications. GNNs are effective at modeling the complex, graph-structured molecular data, capturing the structural and feature characteristics of graphs. Meanwhile, LLMs can encode molecular information implicitly within their parameters, owing to their training on extensive and diverse text corpora. LLMs can provide linguistic insights rich in domain-specific knowledge that serve as auxiliary information, bolstering property prediction tasks. The goal is to leverage the complementary strengths of both LLMs and GNNs to create a more robust and accurate predictive framework. In this study, we introduce a novel multi-modal fusion framework, `MMF`, that integrates the complementary analytical capabilities of GNNs and linguistic comprehension of LLMs within an end-to-end design to enhance the accuracy and robustness of molecular predictions. This framework achieves superior performance in chemical property prediction compared to state-of-the-art baselines, thereby reducing the risk of overfitting and potentially accelerating both the training and inference processes. Overall, this work introduces the subsequent contributions, outlined as follows:

- We propose a multi-faceted semantic fusion approach to obtain cross-modal embeddings, which combines Zero-shot LLMs prompting with graph neural networks (GNNs). We employ a five-step approach to generate cross-modal embeddings for molecular graphs: (a) First, we use custom CoT prompts, which include task-specific instructions to query LLMs in a zero-shot setting to generate technical descriptions on various aspects of the chemical SMILES representations, such as functional groups and chemical properties. (b) Next, we fine-tune small-scale language models (LMs) using the generated technical descriptions for domain-specific customization to compute context-aware token embeddings. (c) Then, we employ a softmax attention pooling mechanism to compute text-level embeddings from the contextualized token embeddings to encapsulate the rich domain-specific knowledge in the generated textual descriptions. (d) Independently and in parallel, we utilize pre-existing GNNs — specifically, Chebyshev Graph Convolution (CGC)[9, 20] — to interpret complex, graph-structured molecular data to compute molecular graph-level embeddings. (e) Finally, we use a cross-modal multi-head attention mechanism[52] to integrate the graph and text-level embeddings, offering a robust and efficient framework for generating semantically enriched cross-modal embeddings that effectively bridge structured and unstructured data.
- We propose ICL for the few-shot molecular property prediction task with LLMs. We leverage ICL to guide LLMs in predicting molecular properties without the necessity of explicit fine-tuning on labeled data. Utilizing context-augmented prompts — which comprise task-specific instructions and demonstrations (input-output mappings, in SMILES notation for molecules and their properties) — Our method queries LLMs to generate and transform predictions into a prediction embedding. It capitalizes on the implicit knowledge embedded within the pretrained parameters of LLMs, enabling it to make accurate predictions for new, unseen molecules, conditioned on the context-augmented prompt.
- We use the Mixture-of-Experts (MOE) method with a gating mechanism at the output layer for high-precision molecule property prediction. Two types of embeddings, cross-modal and prediction embeddings, are integrated into a unified embedding using a gating mechanism that dynamically allocates weights to each embedding based on their predictive performance. The framework's training objectives are twofold: first, to optimize the weight distribution of each embedding to accurately predict the ground-truth molecular properties; second, to fine-tune the embeddings based on this weight distribution. Overall, the framework aims to leverage the strengths of multiple learning strategies to achieve high-precision predictions for molecular properties.

Experiments were conducted using six publicly available molecule property prediction datasets to evaluate the performance of the `MMF` framework, with consistent results demonstrating its effectiveness in accurately predicting properties across all benchmark datasets. In summary, we present a cohesive and multifaceted framework that integrates advanced computational approaches and learning strategies to enhance precision and efficiency in molecular property predictions, potentially fostering advancements in molecular science and technology. The workflow of the proposed approach is illustrated in Figure 1.

# 2 Proposed Method

## 2.1 Task Formulation

A molecular graph $\mathcal{G}$ consists of a set of nodes(atoms) $\mathcal{V}$ and edges(bonds) $\mathcal{E}$ with node feature($\mathbf{X}^v \in \mathbb{R}^{|\mathcal{V}| \times \mathbf{d}_v}$) and edge feature matrix($\mathbf{X}^e \in \mathbb{R}^{|\mathcal{E}| \times \mathbf{d}_e}$), where $\mathbf{d}_v$, $\mathbf{d}_e$ denote the dimensions of nodes and edge features, respectively. The adjacency matrix, $\mathcal{G}_A \in [0,1]^{|\mathcal{V}| \times |\mathcal{V}|}$, describes the graph structure, where $\mathcal{G}_A[v,u] = 1$ if $(v,u) \in \mathcal{E}, u, v \in \mathcal{V}$ or else $\mathcal{G}_A[v,u] = 0$. In the graph property prediction task, let $\mathcal{D}_L = (\mathcal{G}_L, \mathcal{P}_L)$ be the labeled dataset, comprising a set of graphs denoted as $\mathcal{G}_L$, with corresponding properties $\mathcal{P}_L$. A graph encoder, denoted by $f_\gamma(\mathcal{G}) \rightarrow \mathbf{h}_g$, is trained on labeled dataset $\mathcal{D}_L$ to obtain graph-level embeddings $\mathbf{h}_g$, where $\gamma$ represents the trainable parameters of graph encoder. A pre-trained LM encoder, denoted by $f'_\theta(\mathcal{S}_e) \rightarrow \mathbf{h}_{\text{text}}$, is fine-tuned on technical descriptions($\mathcal{S}_e$) generated by zero-shot CoT prompting of LLMs on molecular graphs $\mathcal{G}$ from the labeled dataset $\mathcal{D}_L$ to compute text-level embedding $\mathbf{h}_{\text{text}}$. $\theta$ represents the trainable parameters of LM encoder. A few-shot ICL prompting of LLMs with a few input-output pairs($\mathcal{G}, p$) from the labeled dataset $\mathcal{D}_L$ to compute the predictive embedding, $\mathbf{h}_{\text{ICL}}$. The joint-optimization objective function for property prediction task is defined as minimizing the regression loss $\mathcal{L}_{\text{graph}}$ using a supervised-learning approach to predict the properties $\mathcal{P}_U$ of unlabeled graphs $\mathcal{G}_U$, described as follows,

$$\min_{\gamma,\theta,\omega} \mathcal{L}_{graph}\left(\mathcal{G}_i, \gamma, \theta, \omega\right) = \sum_{(\mathcal{G}_i, p_i) \in \mathcal{D}_L} \ell\Big(g_\omega\big(\mathbf{h}_{g_i}, h_{\text{text}_i}, h_{\text{ICL}_i}\big), p_i\Big) \tag{1}$$

where, the non-linear function $g_\omega(\cdot)$ linearly maps an input vector to a single output value, $\omega$ denote the trainable parameters of the non-linear function. $\ell(\cdot, \cdot)$ denotes the mean squared error loss.

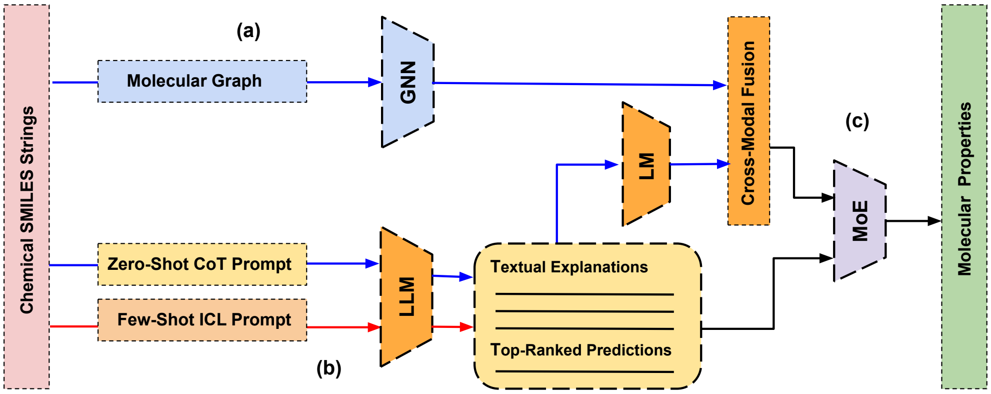

Figure 1: Overview of `MMF` framework. Our framework leverages both the generative and predictive abilities of LLMs. The proposed molecular property prediction framework is a robust, efficient, and multi-step pipeline for predicting molecular properties with high precision. (a) Firstly, it introduces a multi-faceted semantic fusion strategy that leverages Zero-shot CoT prompting of LLMs approach alongside GNNs to generate semantically-aligned cross-modal embeddings for molecules, seamlessly integrating structured and unstructured data. (b) Secondly, the framework incorporates ICL, which taps into the inherent knowledge within pre-trained parameters of LLMs to make accurate predictions on new, unseen molecules, generating prediction embeddings guided by context-augmented prompts without the necessity for explicit fine-tuning on labeled data. (c) Lastly, it employs a MOE mechanism that integrates cross-modal and prediction embeddings through a gating mechanism at the output layer and optimizes the unified embeddings for downstream supervised regression tasks to achieve high-precision predictions. Overall, the cohesive framework aims to synergize multiple learning strategies to achieve unparalleled precision and efficiency in molecular property predictions. It is important to note that we do not customize LLMs through fine-tuning for task-specific adaptation. Instead, we access LLMs through LMaaS[46] platforms via text-based API interaction. The three steps (a), (b), and (c) are illustrated with blue, red, and black arrow lines.

## 2.2 Graph Chebyshev Convolution

Graph Convolutional Networks (GCNs) are designed to learn from graph data. They can be categorized into spatial and spectral approaches. The spatial GCN analyzes node neighborhoods. The spectral GCN, grounded in the spectral graph theory, uses the eigenvalues and eigenvectors of the graph's Laplacian matrix for convolutions. The spectral convolution[6] is computationally expensive and inherently non-local. Graph Chebyshev convolution(CGC, [9, 20]) is a scalable alternative to spectral convolution that offers locality in capturing local-graph-based features, flexibility in approximating spectral properties, and scalability through recursive Laplacian computation. CGC operator utilizes Chebyshev polynomials to approximate spectral graph convolution, enabling the application of convolutional filters on graph-structured data by approximating the graph Laplacian with Chebyshev polynomials. Chebyshev polynomials are obtained from the normalized Laplacian matrix of a graph, $\hat{L} = \hat{D}^{-1/2} \hat{\mathcal{G}}_A \hat{D}^{-1/2}$, where $\hat{\mathcal{G}}_A$ is the normalized adjacency matrix and $\hat{D}$ is the diagonal degree matrix. The Chebyshev approximation approximates the graph Laplacian using Chebyshev polynomials, designated as $T_k(\hat{L})$, calculated through a recurrence relation described as,

$$\mathrm{T}_k(\hat{L}) = \begin{cases} I, & \text{if } k = 0 \\ \hat{L}, & \text{if } k = 1 \\ 2\hat{L}\mathrm{T}_{k-1}(\hat{L}) - \mathrm{T}_{k-2}(\hat{L}), & \text{otherwise} \end{cases}$$

where k refers to the degree of the Chebyshev polynomial and $I$ is identity matrix. Given an input graph-based node and edge feature matrix, $\mathbf{X}^v$ and $\mathbf{X}^e$, respectively. The Chebyshev graph convolution operation is defined as follows,

$$\mathrm{h}_{CGC} = \sigma\left(\sum_{k=0}^{K-1} \mathrm{T}_k(\hat{L})(\mathrm{W}_0\mathbf{X}^v + \mathrm{W}_1\mathbf{X}^e)\Theta_k\right) \tag{2}$$

where $\mathrm{W}_0 \in \mathbb{R}^{d \times d_v}$, $\mathrm{W}_1 \in \mathbb{R}^{d \times d_e}$ denote the trainable weight matrices and $\sigma(\cdot)$ denotes the non-linear sigmoid activation function. $\Theta_k \in \mathbb{R}^{d \times d}$ is the weight or parameter matrix associated with the $k$th-order Chebyshev polynomial. In summary, the layerwise differentiable neural operator maps discrete graphs to a node-level embedding matrix, $\mathrm{h}_{\mathrm{GCC}} \in \mathbb{R}^{n \times d}$, that maximally captures both the topology and the feature information embedded within the graphs. We perform global-graph pooling using the Set2Set algorithm[55] to aggregate and summarize the nodes feature information to obtain a graph-level embedding($\mathbf{h}_g \in \mathbb{R}^d$) to encapsulate and preserve the overall graph characteristics.

### 2.3 Language Models

In recent years, the introduction of pre-trained large language models (LLMs), such as ChatGPT[5], Google's PaLM[8], and Meta's LLaMA[48], has had a transformative impact in the domain of language modeling, enhancing performance and capabilities across a wide range of NLP tasks and applications. The LLMs adopt a 'pre-train, prompt, and predict' approach, attaining vast linguistic understanding through pre-training and generating human-like responses with tailored prompts[57, 61]. Smaller pre-trained language models (LMs), such as BERT[11] and DeBERTa[21], lack the sophisticated logical reasoning abilities of LLMs. However, they offer advantages such as access to logits or token embeddings, which aid in explainability for downstream applications utilizing the pre-trained LM models. In addition, these small-scale LMs can be fine-tuned with labeled data in an affordable manner for domain-specific customization. However, LLMs are resource-intensive to fine-tune with labeled data for task adaptation due to their high model complexity, making them less accessible to low-budget research labs. Additionally, the black-box nature of these large language models limits interpretability in downstream applications by not providing access to latent token embeddings or logits. To address these challenges, Language Modeling as a Service (LMaaS[46]) allows text-based API access to LLMs avoiding the high computational costs of domain-specific customization through fine-tuning. While it may seem intuitive to use LLMs for interpreting chemical SMILES strings, the study of their effectiveness in predicting molecular properties is still in its early stages. LLMs have been proven to be effective at zero-shot learning and (in-context) few-shot learning across a diverse set of tasks in the field of NLP. We refer to the method of conditioning the language model as 'prompting'. Prompts with explicit conditioning based on task-specific instructions and a few demonstrations are termed as 'few-shot prompts', while those that rely solely on task-specific instructions are referred to as 'zero-shot prompts'. In our molecular property prediction task, we focus on the development and exploration of various hand-crafted prompt engineering strategies. These include the zero-shot chain-of-thought (Zero-Shot CoT) and few-shot (in-context) learning (Few-Shot ICL), which aid in conditioning the LLMs to adapt to new tasks either through instructions that describe the task (zero-shot) without prior labeled data or through task-specific instructions accompanied by demonstrations(input-output pairs, i.e., few-shot). In our framework, we utilize LLMs in both zero-shot and few-shot learning scenarios, each serving a different purpose. (a) Our approach employs customized CoT prompts to query general-purpose LLMs in a zero-shot setting, generating textual descriptions that encapsulate various aspects of organic molecules, including molecular structure, physical properties, applications, and more. We fine-tune small-scale pre-trained LMs using these technical descriptions, then compute text-level embeddings using a softmax attention pooling mechanism to encapsulate the specialized knowledge in the technical descriptions, which is critical for downstream property prediction tasks. (b) We use (in-context) few-shot prompting of LLMs for molecular property prediction by leveraging input-output mappings (chemical SMILES strings — molecular properties pairs) in the context-augmented prompts. The ICL approach exploits the inherent knowledge of LLMs, conditioning on the augmented prompt to accurately predict molecular properties for new, unseen molecules, without the need for explicit fine-tuning.

**Evaluation LLMs & LMs:** In our work, we employ three representative LLMs: text-davinci-003, ChatGPT, and BARD. Table 1 summarizes the main characteristics of the LLMs. Text-davinci-003, a GPT-3 model from OpenAI, excels in numerous natural language tasks, particularly zero-shot and few-shot instruction-following tasks. GPT-3.5-turbo, a refined variant within the GPT-3.5 model family, is widely recognized for its exceptional performance and cost-effectiveness. Meanwhile, Google's BARD[1], a recently updated chatbot featuring a new large language model (LLM) known as PaLM 2[3], stands out due to its significantly larger parameter count and expanded vocabulary size compared to the models in the GPT-3.5 family. In the Google Bard and GPT model family, two parameters — Top-p (also known as nucleus sampling or probabilistic sampling) and temperature — are crucial for controlling text generation. Specifically, Top-p governs the diversity of generated text, while the temperature parameter influences the randomness of the language models during text generation. The Top-p parameter sets a probability threshold for including a token in the generated sentence, serving as a sampling method that prevents the language model from generating tokens that are either too rare or too common. Conversely, the temperature parameter modulates the randomness of the generated text: a higher temperature leads to more random text, whereas a lower temperature results in more deterministic text. In our experiments, we set the Top-p and temperature parameters to one and zero, respectively, to retrieve factual and accurate textual outputs. Additionally, we utilized a pre-trained small-scale language model known as DeBERTa[2][21]. In a zero-shot setting, LLMs generate technical descriptions about chemical SMILES strings, while small-scale language models encode rich knowledge in textual descriptions for task-specific customization, enhancing the performance of the framework in property prediction tasks. In contrast, we employ few-shot prompting of LLMs to predict molecular properties.

Table 1: Specifications of LLMs and LMs: *Enterprise* refers to the technology organization that developed the language model; *Charges* indicates the cost associated with the use of 1K tokens; *Last Update Date* denotes that the LLM's knowledge base is limited to information available up until that date.

| Model | Enterprise | Charges | Last Update Date | Vocabulary size |
|---|---|---|---|---|
| text-davinci-003 | Open-AI | 0.02$ | Sep. 2021 | 175B |
| ChatGPT | Open-AI | 0.002$ | Jun. 2021 | 175B |
| BARD | Google | Free | Undisclosed | 1,560B |
| DeBERTa | Hugging Face | Free | N/A | 50M |

**Zero-short LLM Prompting:** We access LLMs through LMaaS platforms[46] via text-based API interactions. We utilize a standardized chain-of-thoughts (CoT) prompt template to query LLMs in a zero-shot setting, aiming to glean linguistic insights into the specialized knowledge of chemical SMILES strings, including their structure and physical properties, among others. The generated technical descriptions serve as auxiliary information for downstream applications. The prompt template guides the LLMs through a sequence of open-ended queries to acquire specific information about a given organic molecule. The custom CoT prompt format is as follows:

**Prompt 1:** What is the molecular structure of this **chemical SMILES strings**? Could you describe its atoms, bonds, functional groups, and overall arrangement? **Prompt 2:** What are the physical properties of this molecule such as its boiling point, melting point, and density? **Prompt 3:** What is the solubility behavior of this molecule? In which solvents does it dissolve and which does it not? **Prompt 4:** What is the chemical reactivity of this molecule? How does it interact with various reagents? **Prompt 5:** Are there any common reactions that this molecule is known to undergo? Could you describe them? **Prompt 6:** What is the mechanism of these reactions? Could you describe the various steps involved? **Prompt 7:** Does this molecule exhibit any unique optical, electrical, or magnetic properties? **Prompt 8:** Is this molecule chiral? If yes, how does its chirality influence its behavior or properties? **Prompt 9:** Does this molecule form part of any important biological processes or pathways? **Prompt 10:** Is this molecule synthesized industrially or in the laboratory? If yes, could you explain the process? **Prompt 11:** Is this molecule found naturally? If yes, in what sources is it most commonly found? **Prompt 12:** Are there any notable uses or applications for this molecule in medicine, industry, or other fields? **Prompt 13:** What safety measures should be taken when handling this molecule? **Prompt 14:** Are there any environmental impacts associated with the production, use, or disposal of this molecule?

Querying the LLM produces detailed technical descriptions of chemical SMILES strings and their properties.

---

[1] https://bard.google.com

[2] For more information, refer to the DeBERTa model documentation available at `https://huggingface.co/docs/transformers/index`.

(**LLMs Response**) [Textual descriptions]

In the next section, we will discuss how to integrate these textual descriptions as additional features to aid in the improvement and fine-tuning of downstream LMs and subsequent applications.

**Fine-Tuning LMs and Domain-Specific Customization:** In our approach, we utilize a small-scale pre-trained language model (LM) to encode the text outputs generated by a larger language model (LLM). During fine-tuning, the smaller LM extracts informative features from the generated descriptions for task-specific customization. The small-scale LM serves as an intermediate layer between LLMs and downstream prediction layers. We fine-tune small-scale LMs (referred to as $\text{LM}_{\text{expl}}$) to process technical descriptions generated by LLMs for the property prediction task. We input text sequences from LLMs (denoted as $\mathcal{S}_e$) into the $\text{LM}_{\text{expl}}$ model to compute context-aware token embeddings. These embeddings capture the contextual information and semantic relationships between the words or phrases described as follows,

$$h_{\text{expl}} = \text{LM}_{\text{expl}}(\mathcal{S}_e); \tag{3}$$

where the contextualized embeddings $h_{\text{expl}} \in \mathbb{R}^{m \times d}$, where $m$ represents the number of tokens in the input sequence $\mathcal{S}_e$ and $d$ is the token embedding dimension. To encode the textual explanations into a fixed-length vector, we apply a softmax attention pooling mechanism to calculate a weighted sum of the token embeddings. This results in a comprehensive representation of the entire textual descriptions, computed as follows,

$$\alpha_i = \text{softmax}(q_i); \quad q_i = \mathbf{u}^T h_{\text{expl}}^{(i)} \tag{4}$$

$$h_{\text{text}} = \sum_{i=0}^{m} \alpha_i h_{\text{expl}}^{(i)} \tag{5}$$

where $\mathbf{u}$ is a trainable parameter and $\alpha$ is the attention coefficient. The text-level embedding $h_{\text{text}} \in \mathbb{R}^d$ encapsulates the extracted domain-specific knowledge from foundational LLMs on chemical SMILES strings. $h_{\text{text}}$ enhances explainability by unpacking the black-box nature of LLMs by utilizing the generated descriptions from LLMs on chemical SMILES strings.

**Few-Shot LLM Prompting:** In-Context Learning (ICL) enables LLMs to adapt to new tasks without the need for explicit, gradient-based fine-tuning[5] on labeled data. This approach allows LLMs to learn through analogy, utilizing just a few input-output pairs specific to the downstream task. ICL leverages the implicit knowledge embedded in pre-trained LLM parameters to adapt to new tasks through task-specific demonstrations, thereby avoiding the need to repurpose LLMs with parameter updates. The context-augmented prompt provides task-specific instructions and demonstrations (input-output mappings), enabling LLMs to generate outputs conditioned on the prompt for improved generalization performance. In the case of molecular graph property prediction tasks, ICL involves constructing a context-augmented prompt using a few input-output pairs $(\mathcal{G}_i, \mathcal{P}_i)$ sampled from the training data and the task-specific instruction is related to the query SMILES representation. At inference time, on test input $\mathcal{G}_{\text{test}}$, ICL generates the output based on the conditional probability distribution, $\mathcal{P}_{\text{test}} \sim \mathbf{P}(\mathcal{P}_{\text{test}} \mid (\mathcal{G}_{\text{train}}, \mathcal{P}_{\text{train}}), \mathcal{G}_{\text{test}})$, where $\sim$ denotes the decoding strategy. To examine how the quality and quantity of ICL demonstrations impact the performance of property prediction tasks, we investigate two distinct ICL sampling strategies. We explore two distinct ICL sampling strategies: "Random" and "Scaffold". The quality of demonstrations is determined by the sampling strategies used to identify the top-$K$ chemical SMILES representations that are most similar to the query SMILES representation. To investigate the impact of the quantity of ICL demonstrations on performance, we optimize the number of ICL demonstrations ($K$) for each query SMILES representation. In the random strategy, we randomly sample $K$ input-output pairs from the training data. In contrast, the Scaffold strategy employs Tanimoto similarity [47] based on Morgan fingerprints [36] with a radius of 2, to identify the top-$K$ most similar chemical SMILES representations to a given query SMILES representation within the training data. We employ two sampling strategies to construct an augmented prompt for analyzing the effectiveness of ICL demonstrations on property prediction tasks. In summary, our goal is to task LLMs with a contextual prompt. This prompt comprises a list of input-output pairs, where the input represents an organic molecule in SMILES notation, and the output denotes its molecular properties along with task-specific instructions. The instruction in the context-augmented prompt directs LLMs to predict the multiple molecular properties of the query SMILES representation. This task will demonstrate the LLM's ability to predict these properties based on its inherent knowledge, simply by conditioning on the prompt, without any parameter updates. This approach stands in contrast to supervised learning, where parameter updates are performed to fine-tune models based on labeled data, enabling them to predict the properties of new, unseen molecules. For each query SMILES representation, the PLLMs generate a $c$-dimensional vector $h_{\text{pred}_i} \in \mathbb{R}^c$, where $c$ signifies the dimension associated with the multiple properties to be predicted. This vector is linearly encoded into a high-dimensional space to produce a prediction embedding $h_{\text{ICL}_i} \in \mathbb{R}^d$, which encapsulates the LLMs' predictions. Here, $d$ represents the embedding dimension and $c \ll d$. An example of an ICL prompt is as follows,

> Below are the input-output examples (SMILES strings-molecular properties pairs) for property prediction task. Predict the molecular properties for the query SMILES strings.

In the subsequent sections, we will discuss the cross-modal attention layer and an output layer.

## 2.4 Cross-modal Attention Layer

We compute the cross-modal embedding $h_i^f$ through a multi-head attention mechanism[52] that integrates the graph-level embeddings $h_{g_i}$ and text-level embeddings $h_{\text{text}_i}$. The multi-head attention(MHA) mechanism offers a robust framework for integrating cross-domain embeddings, facilitating multi-faceted analysis in the context of interpreting correlations between molecular structures and properties. (a) **Richer Representation:** The multi-head attention mechanism enables the computation of a richer, multi-faceted representation for each molecule. It allows each attention head to focus on different facets, such as chemical bonds or functional groups in the molecular structure. Meanwhile, other heads attend to corresponding textual descriptions, like reactivity or the physicochemical properties of the observed functional group. (b) **Semantic Matching:** Additionally, the multi-head mechanism facilitates semantic matching, as individual heads can specialize in aligning various semantic aspects, such as specific functional groups in the molecular graph with their corresponding textual descriptions. For example, one attention head might specialize in aligning aromatic rings in the molecular structure with textual descriptions related to aromaticity. This allows the model to effectively integrate different types of molecular information, providing a contextual and comprehensive view of both text and graph data. (c) **Enhanced Context Sensitivity:** Finally, the multi-head mechanism enhances context sensitivity. Molecules often behave differently depending on their context. For instance, a molecule's reactivity can change based on its surrounding environment, and this information might be captured in textual descriptions. Multi-head attention allows the model to be sensitive to this context by considering both the graph-level embeddings and the text-level descriptions in tandem. This could be particularly beneficial for complex tasks. In summary, MHA mechanism for computing cross-modal embeddings facilitates a rich, nuanced representation of molecules by allowing parallel focus on various facets of data, including molecular graph structures and textual descriptions. It enhances semantic matching by aligning different semantic features from both text and graph data, and improves context sensitivity, enabling the model to understand and adapt to the dynamic behaviors of molecules in different contexts. Moreover, the MHA mechanism improves computational efficiency through parallel processing across heads, and its modular nature makes it easy to adapt and extend. It also offers benefits such as increased model capacity, and better generalization to unseen data. Overall, multi-head attention provides a robust and versatile framework for seamlessly fusing information across different modalities. We compute the Query, Key, Value projections for graph-level embeddings for each head h as follows:

$$Q^h_{g_i} = h_{g_i} W^h_{Q_g}; K^h_{g_i} = h_{g_i} W^h_{K_g}; V^h_{g_i} = h_{g_i} W^h_{V_g} \tag{6}$$

Similarly, the Query, Key, Value projections for text-level embeddings for each head h:

$$Q^h_{\text{text}_i} = h_{\text{text}_i} W^h_{Q_{\text{text}}}; K^h_{\text{text}_i} = h_{\text{text}_i} W^h_{K_{\text{text}}}; V^h_{\text{text}_i} = h_{\text{text}_i} W^k_{V_{\text{text}}} \tag{7}$$

We concatenate the keys and values from both graph-level and text-level embeddings, which provides a powerful way to integrate information from both modalities into a unified, rich representation.

$$K^h_{\text{concat}_i} = [K^h_{g_i}, K^h_{\text{text}_i}]; V^h_{\text{concat}_i} = [V^h_{g_i}, V^h_{\text{text}_i}] \tag{8}$$

We perform softmax attention to integrate complementary information from the two modalities, focus on contextually relevant information, and semantically align them through attention mechanism. The Softmax function is applied over the keys for each query.

$$A^h_i = \text{Softmax}\left(\frac{(Q^h_{g_i} + Q^h_{\text{text}_i}) {K^h_{\text{concat}_i}}^T}{\sqrt{d_h}}\right) \tag{9}$$

Each head outputs a new vector representation that highlights the most relevant features in the input embeddings (both graph and text-level), according to the attention mechanism for that specific head, which is tailored to the specific aspects or relationships within the data.

$$O^h_i = A^h_i V^h_{\text{concat}_i} \tag{10}$$

Finally, all the head-specific outputs are concatenated and linearly transformed to create the final cross-modal representation as follows,

$$O_{\text{concat}_i} = [O_i^1, O_i^2, \ldots, O_i^H] \tag{11}$$

$$h_i^f = O_{\text{concat}_i} W_O \tag{12}$$

where $W^h_{Q_g}$, $W^h_{K_g}$, $W^h_{V_g}$, $Q^h_{\text{text}_i}$, $W^h_{Q_{\text{text}}}$, $W^h_{K_{\text{text}}}$, $W_O$ are the learnable weight matrices. $d_h$ is the dimensionality of the key/query/value for each head, and $H$ is the number of heads.

### 2.5 Output Layer

In this framework, we utilize the mixture-of-experts (MOE) technique with a gating mechanism for the output layer. In the context of this framework, the MOE mechanism is a competitive game where each embedding — either cross-modal embeddings obtained from the multi-head attention mechanism (integrating zero-shot LLM prompting & GNNs outputs) or prediction embeddings from few-shot LLM prompting — aims to maximize its contribution to the final prediction. Unlike a cooperative game, each embedding aims to outperform the others for a larger weight from the gating mechanism. The gating mechanism allocates weights based on individual performance, creating a competitive landscape where more accurate embeddings gain greater influence. This competition can drive the framework towards a globally optimal solution. The embeddings are combined by the gating mechanism, which allocates input-dependent weights to calculate a weighted sum of embeddings. Training in this framework aims to: a) determine the optimal weight distribution for precise predictions of ground-truth molecular properties, and b) optimize the embeddings jointly according to the weight distribution specified by the gating mechanism. The unified embeddings are obtained by merging embeddings using input-dependent weights allocated by the gating mechanism as follows,

$$g = \sigma\big(f_s(h_i^f) + f_g(h_{\text{ICL}_i})\big) \tag{13}$$

$$\mathbf{h}_u = \sigma\big(g(h_i^f) + (1-g)(h_{\text{ICL}_i})\big) \tag{14}$$

where $f_s$ and $f_g$ are linear operators and $\sigma$ is the non-linear sigmoid operation. Finally, we use a linear operator to transform $h_u$ to predict the molecular properties of each graph.

## 3 Experiments and Results

### 3.1 Datasets and Experimental Setup

The QM8[40] and QM9[39] are two large datasets of quantum chemical properties for low-tree width organic molecules, which serve as benchmarks and training data for ML models in the field of quantum chemistry property prediction task. The QM8 dataset comprises 21,786 molecules containing up to 8 heavy atoms, namely C, O, N, or F. It provides properties such as electronic excitation energies, oscillator strengths, and ionization potentials. In comparison, the QM9 dataset is larger, consisting of 133,885 molecules with up to 9 heavy atoms, including properties like atomization energy, HOMO/LUMO gap, dipole moment, and polarizability. Both datasets are split into training, validation, and test sets. The training set is used to fit the model parameters, the validation set is used to select the best hyperparameters, and the test set is used to evaluate the model's generalization performance. The quantum properties were standardized to have zero mean and unit variance. The predictions were then re-normalized to the original scale to calculate the error metric. In this work, our framework integrates both large language models (LLMs) and smaller language models (LMs). We focus on employing zero-shot CoT and few-shot ICL learning techniques to prompt LLMs for the molecule prediction task without the need for retraining or finetuning. We achieve this by using a LMaaS platform[46] to access frozen trainable parameters of LLMs through text-based API interactions. The hyperparameters of this framework were set to a batch size of 32, 50 epochs for training, and a hidden or embedding dimension of 128. We conducted our experiments using the following four large language models (LLMs): GPT-4.0, GPT-3.5-turbo, GPT-3.0-text-davinci-003, and Google Bard. We did not fine-tune the hyperparameters of our framework for each LLM. Instead, we utilized the same hyperparameters across all LLMs. This shows that our framework is general and easy to use, and that it can utilize any off-the-shelf LLMs. To optimize the use of computational resources, we utilized 8 V100 GPUs, each equipped with 8 GB of GPU memory, for the training of deep learning models built upon the PyTorch framework. The LLMs have a context length limitation with a maximum sequence length of 4096 tokens for GPT models and 4000 tokens for Google Bard. The Adam optimizer[23] was used to train the framework, starting with a learning rate of $1e^{-3}$. A learning rate decay scheduler was employed to reduce the learning rate by half if the validation loss did not improve for 7 epochs, and early stopping was implemented to prevent overfitting on the training set. For few-shot learning, we utilized the scaffold technique with a hyperparameter $K$ set to 16 for sampling demonstrations to construct context-augmented prompts. The framework's performance was evaluated using the MAE metric, and the results were presented on the test datasets. Three independent experiments were conducted, and we report the average.

### 3.2 Results

Table 2 compares the performance of the `MMF W/GPT-4` framework to baseline algorithms on the QM8 dataset. The results are reported on both the validation and test datasets. The primary objective is to predict 16 distinct properties of the electronic spectra and the energy per molecular graph. We report the average prediction error across all properties as a single value per molecular graph, maintaining consistency with earlier studies[29] for a fair comparison with the baseline algorithms. We report the baseline results from a previous study[29]. Our framework performance is compared against several baseline algorithms, namely GCN-FP [13], GGNN [27], DCNN [4], ChebyNet [9], GCN [24], MPNN [18], GraphSAGE [19], GPNN [28], and GAT [53]. Our proposed framework demonstrates a significant improvement of $25.35\%$ compared to the next-best baseline.

| Methods | Validation MAE ($\times 1.0e^{-3}$) | Test MAE ($\times 1.0e^{-3}$) |
|---|---|---|
| GCN-FP [13] | $15.06 \pm 0.04$ | $14.80 \pm 0.09$ |
| GGNN [27] | $12.94 \pm 0.05$ | $12.67 \pm 0.22$ |
| DCNN [4] | $10.14 \pm 0.05$ | $9.97 \pm 0.09$ |
| ChebyNet [9] | $10.24 \pm 0.06$ | $10.07 \pm 0.09$ |
| GCN [24] | $11.68 \pm 0.09$ | $11.41 \pm 0.10$ |
| MPNN [18] | $11.16 \pm 0.13$ | $11.08 \pm 0.11$ |
| GraphSAGE [19] | $13.19 \pm 0.04$ | $12.95 \pm 0.11$ |
| GPNN [28] | $12.81 \pm 0.80$ | $12.39 \pm 0.77$ |
| GAT [53] | $11.39 \pm 0.09$ | $11.02 \pm 0.06$ |
| LanczosNet, [29] | $9.65 \pm 0.19$ | $9.58 \pm 0.14$ |
| AdaLanczosNet [29] | $10.10 \pm 0.22$ | $9.97 \pm 0.20$ |
| **MMF W/GPT-4** | **$7.63 \pm 0.07$** | **$7.45 \pm 0.03$** |

Table 2: The table shows the experimental results of the framework's performance on the QM8 dataset in comparison to the baseline algorithms in terms of the MAE metric. We utilized the `MMF W/GPT-4` framework with the scaffold technique, setting $K$ to 16.

Table 3 compares the framework's performance to baseline algorithms on the QM9 dataset. The results are reported for the test dataset, using the mean absolute error (MAE) as the evaluation metric, with lower values indicating better performance. The baseline results are taken from a previous work[16]. The baseline algorithms include SchNet ([44]), PhysNet ([50]), Provably Powerful Graph Networks (PPGN, [34]), MEGNet-simple ([7]), Cormorant (C-Net, [2]), and DimeNet ([16]). Our framework demonstrates a significant improvement compared to the next-best baseline method.

| Target | Unit | PPGN[34] | SchNet[44] | PhysNet[50] | MEGNet-s[7] | Cormorant[2] | DimeNet[16] | **MMF W/GPT-4** |
|---|---|---|---|---|---|---|---|---|
| $\mu$ | D | $4.7 \times 10^{-2}$ | $3.3 \times 10^{-2}$ | $5.29 \times 10^{-2}$ | $5 \times 10^{-2}$ | $1.3 \times 10^{-1}$ | $2.86 \times 10^{-2}$ | $1.06 \times 10^{-2}$ |
| $\alpha$ | $a_0^3$ | $1.31 \times 10^{-1}$ | $2.35 \times 10^{-1}$ | $6.15 \times 10^{-2}$ | $8.1 \times 10^{-2}$ | $9.2 \times 10^{-2}$ | $4.69 \times 10^{-2}$ | $2.19 \times 10^{-2}$ |
| $\epsilon_{\text{HOMO}}$ | meV | $4.03 \times 10^{1}$ | $4.1 \times 10^{1}$ | $3.29 \times 10^{1}$ | $4.3 \times 10^{1}$ | $3.6 \times 10^{1}$ | $2.78 \times 10^{1}$ | $1.843 \times 10^{1}$ |
| $\epsilon_{\text{LUMO}}$ | meV | $3.27 \times 10^{1}$ | $3.4 \times 10^{1}$ | $2.47 \times 10^{1}$ | $4.4 \times 10^{1}$ | $3.6 \times 10^{1}$ | $1.97 \times 10^{1}$ | 9.57 |
| $\Delta\epsilon$ | meV | $6.00 \times 10^{1}$ | $6.3 \times 10^{1}$ | $4.25 \times 10^{1}$ | $6.6 \times 10^{1}$ | $6.0 \times 10^{1}$ | $3.48 \times 10^{1}$ | $2.234 \times 10^{1}$ |
| $\langle R^2 \rangle$ | $a_0^2$ | $5.92 \times 10^{-1}$ | $7.3 \times 10^{-2}$ | $7.65 \times 10^{-1}$ | $3.02 \times 10^{-1}$ | $6.73 \times 10^{-1}$ | $3.31 \times 10^{-1}$ | $1.08 \times 10^{-1}$ |
| ZPVE | meV | 3.12 | 1.7 | 1.39 | 1.43 | 1.98 | 1.29 | $7.85 \times 10^{-1}$ |
| $U_0$ | meV | $3.68 \times 10^{1}$ | $1.4 \times 10^{1}$ | 8.15 | $1.2 \times 10^{1}$ | $2.8 \times 10^{1}$ | 8.02 | 3.55 |
| $U$ | meV | $3.68 \times 10^{1}$ | $1.9 \times 10^{1}$ | 8.34 | $1.3 \times 10^{1}$ | - | 7.89 | 2.43 |
| $H$ | meV | $3.63 \times 10^{1}$ | $1.4 \times 10^{1}$ | 8.42 | $1.2 \times 10^{1}$ | - | 8.11 | 3.09 |
| $G$ | meV | $3.64 \times 10^{1}$ | $1.4 \times 10^{1}$ | 9.40 | $1.2 \times 10^{1}$ | - | 8.98 | 4.23 |
| $c_{\text{v}}$ | /mol/K | $5.5 \times 10^{-2}$ | $3.3 \times 10^{-2}$ | $2.80 \times 10^{-2}$ | $2.9 \times 10^{-2}$ | $3.1 \times 10^{-2}$ | $2.49 \times 10^{-2}$ | $1.37 \times 10^{-2}$ |

Table 3: The table compares our method (on the right) to baselines (on the left) using MAE metric on the QM9 dataset. We utilized `MMF W/GPT-4` framework with scaffold technique, setting $K$=16.

## 4 Conclusion

In the rapidly evolving field of computational chemistry, the pressing need for methodologies with higher accuracy and robustness in predicting molecular properties is undeniable. Our pioneering efforts in this study have introduced the Multi-Modal Fusion (MMF) framework, which synergistically amalgamates LLMs and GNNs to enhance the accuracy of molecular property predictions. Our approach not only improves predictions but also reduces the likelihood of overfitting, surpassing existing benchmarks in property prediction tasks. Our results on benchmark datasets confirm our hypothesis that fusing information from text and graph-based modalities can significantly enhance performance. This breakthrough opens new avenues for scientific discovery, advancing computational chemistry applications across domains to shape the next generation of tools and insights in chemistry.

# 5 Technical appendix

## 5.1 Extended Experimental Results

Table 4 and 5 show the performance of our `MMF` framework when paired with different off-the-shelf LLMs on the QM8 and QM9 datasets, respectively. The underlying hypothesis of our framework is that GNNs can be utilized for initial explorations and generating baseline results in this regard. Zero-Shot CoT prompting of LLMs can be harnessed to enhance the preliminary outcomes of GNNs by utilizing the implicit domain-specific knowledge embedded within LLMs trainable parameters to obtain expressive cross-modal embeddings. Few-Shot ICL, on the other hand, can be utilized to further refine the framework's predictions by providing demonstrations from the training data, potentially leading to a more robust and accurate predictive framework for molecular property prediction. The experimental findings support the validity of this hypothesis, advancing drug discovery and materials science — a task where conventional deep learning methods often fall short. This glaring gap in the integration of graph-based and linguistic insights renders current architectures less comprehensive and nuanced, potentially impeding breakthroughs across various disciplines.

| Methods | Validation MAE ($\times 1.0e^{-3}$) | Test MAE ($\times 1.0e^{-3}$) |
|---|---|---|
| MMF W/GPT-4 | **7.63 ± 0.07** | **7.45± 0.03** |
| MMF W/GPT-3.5-turbo | 8.13 ± 0.09 | 8.21 ± 0.03 |
| MMF W/GPT-3.0-text-davinci-003 | 8.28 ± 0.06 | 8.37 ± 0.07 |
| MMF W/Google Bard | 9.15± 0.04 | 9.33 ± 0.06 |

Table 4: The table shows the `MMF` framework performance with different off-the-shelf LLMs on the QM8 dataset using the MAE metric. All the experiments were performed with the scaffold technique, setting $K$ to 16.

| Target | Unit | **MMF W/GPT-4** | **MMF W/GPT-3.5-turbo** | **MMF W/GPT-3.0-text-davinci-003** | **MMF W/Google Bard** |
|---|---|---|---|---|---|
| $\mu$ | D | $1.06 \times 10^{-2}$ | $1.24 \times 10^{-2}$ | $1.27 \times 10^{-2}$ | $1.89 \times 10^{-2}$ |
| $\alpha$ | $a_0^3$ | $2.19 \times 10^{-2}$ | $2.37 \times 10^{-2}$ | $2.49 \times 10^{-2}$ | $3.54 \times 10^{-2}$ |
| $\epsilon_{\text{HOMO}}$ | meV | $1.843 \times 10^{1}$ | $2.043 \times 10^{1}$ | $2.147 \times 10^{1}$ | $2.497 \times 10^{1}$ |
| $\epsilon_{\text{LUMO}}$ | meV | 9.57 | $1.002 \times 10^{1}$ | $1.185 \times 10^{1}$ | $1.565 \times 10^{1}$ |
| $\Delta\epsilon$ | meV | $2.234 \times 10^{1}$ | $2.408 \times 10^{1}$ | $2.597 \times 10^{1}$ | $2.904 \times 10^{1}$ |
| $\langle R^2 \rangle$ | $a_0^2$ | $1.08 \times 10^{-1}$ | $1.37 \times 10^{-1}$ | $1.42 \times 10^{-1}$ | $2.87 \times 10^{-1}$ |
| ZPVE | meV | $7.85 \times 10^{-2}$ | $8.46 \times 10^{-2}$ | $9.05 \times 10^{-2}$ | 1.143 |
| $U_0$ | meV | 3.55 | 4.19 | 4.52 | 6.74 |
| $U$ | meV | 2.43 | 3.16 | 3.97 | 5.89 |
| $H$ | meV | 3.09 | 3.48 | 3.72 | 5.63 |
| $G$ | meV | 4.23 | 4.71 | 4.89 | 6.67 |
| $c_{\text{v}}$ | /mol/K | $1.37 \times 10^{-2}$ | $1.44 \times 10^{-2}$ | $1.58 \times 10^{-2}$ | $2.17 \times 10^{-2}$ |

Table 5: The table shows the `MMF` framework performance using various off-the-shelf LLMs on QM9 dataset in terms of MAE metric. All experiments utilized the scaffold technique, with $K$ set to 16.

## 5.2 Additional Datasets and Experimental Results

We have additionally evaluated the performance of our multi-modal learning framework on four benchmark datasets: ESOL[10], FreeSolv[35], Lipophilicity[17], and PDBbind[56], comparing it against several popular algorithms. Each of these datasets plays a critical role in various areas of computational chemistry and molecular modeling, particularly in the field of drug discovery and development. We partitioned the data into multiple sets of 80% for training, 10% for validation, and 10% for testing. To ensure that molecules with the same scaffold did not appear in multiple splits, we utilized scaffold splitting, following the strategy implemented in [62] using the Chemprop library[60]. This approach offers a more resilient and robust evaluation of the framework's performance compared to random splitting. Table 6 presents the experimental results of the framework performance compared to the baselines. Our experimental findings support the efficacy of the `MMF` framework, which outperformed other state-of-the-art baselines on all four datasets, achieving lower prediction error scores. The baseline results are reported from a previous study[62]. Moreover, our framework can identify subtle differences in molecular structures, enhancing the expressiveness of cross-modal embeddings for improved property prediction. We aim to further explore the potential of our multi-modal learning framework in classification-based molecular property prediction tasks. Thus, we evaluated our `MMF` framework performance in comparison to the baselines using a set of additional benchmark datasets, including BBBP, HIV, BACE, Tox21, and ClinTox, which were originally introduced in an earlier study[58]. These datasets comprise chemical SMILES representations paired with binary labels that indicate specific molecular properties, such as toxicity or the ability to inhibit HIV replication. The baseline results have been reported from a previous study[62]. Performance is measured using the ROC curve metric, with higher scores indicating better results. Entries marked with "-" denote unavailable data for specific method-task combinations. The BBBP and BACE benchmark datasets are balanced, whereas the other datasets are highly imbalanced. For in-context learning (ICL), we select samples either randomly or based on the highest Tanimoto similarity computed using RDKit; however, the latter method does not guarantee a balanced class distribution. Our study employs a flexible sampling strategy, aiming to obtain a representative sample from datasets with high label imbalances. Specifically, our strategy assists in sampling demonstrations for constructing augmented prompts, utilizing a 3:2 majority-to-minority class ratio for framework evaluations. Table 7 showcases the performance of the framework in comparison to the baselines.

Table 6: Our proposed framework was evaluated against various baselines for molecular property predictions using several datasets, such as ESOL, FreeSolv, Lipophilicity, and PDBbind. The performance was measured using RMSE, with lower values indicating better prediction accuracy. All the experiments were performed with the scaffold technique, setting $K$ to 16.

| | **ESOL** | **FreeSolv** | **Lipophilicity** | **pdbbind** |
|---|---|---|---|---|
| **MMF W/GPT-4** | **0.413** | **1.283** | **0.373** | **1.028** |
| **MMF W/GPT-3.5-turbo** | 0.457 | 1.374 | 0.427 | 1.185 |
| **MMF W/GPT-3.0-text-davinci-003** | 0.513 | 1.408 | 0.459 | 1.213 |
| **MMF W/Google Bard** | 0.608 | 1.675 | 0.553 | 1.343 |
| SELFormer[62] | 0.682 | 2.797 | 0.735 | 1.488 |
| D-MPNN[60] | 1.050 | 2.082 | 0.683 | 1.397 |
| MolCLR[15] | 1.110 | 2.200 | 0.650 | - |
| Hu et al.[22] | 1.220 | 2.830 | 0.740 | - |
| MGCN[33] | 1.270 | 3.350 | 1.110 | - |
| GEM[15] | 0.798 | 1.877 | 0.660 | - |
| SchNet[44] | 1.050 | 3.220 | 0.910 | - |
| KPGT[25] | 0.803 | 2.121 | 0.600 | - |
| GraphMVP-C[31] | 1.029 | - | 0.681 | - |
| GCN[24] | 1.430 | 2.870 | 0.850 | - |
| GIN[59] | 1.450 | 2.760 | 0.850 | - |
| ChemBERTa-2[1] | - | - | 0.986 | - |

Table 7: We compared our proposed framework with various baselines for predicting molecular properties using datasets, including BACE, BBBP, HIV, Tox21, and SIDER. The performance was evaluated using the ROC curve metric, where higher scores indicate better results. All the experiments were performed with the scaffold technique, setting $K$ to 16.

| | **BACE** | **BBBP** | **HIV** | **Tox21** | **SIDER** |
|---|---|---|---|---|---|
| | ROC | ROC | ROC | ROC | ROC |
| **MMF W/GPT-4** | **0.893** | **0.937** | **0.862** | **0.898** | **0.812** |
| **MMF W/GPT-3.5-turbo** | 0.881 | 0.929 | 0.851 | 0.877 | 0.809 |
| **MMF W/GPT-3.0-text-davinci-003** | 0.877 | 0.921 | 0.849 | 0.872 | 0.792 |
| **MMF W/Google Bard** | 0.861 | 0.916 | 0.826 | 0.859 | 0.766 |
| SELFormer[62] | 0.832 | 0.902 | 0.681 | 0.653 | 0.745 |
| D-MPNN[60] | 0.809 | 0.710 | 0.771 | 0.759 | 0.570 |
| MolBERT[14] | 0.866 | 0.762 | 0.783 | - | - |
| ChemBERTa-2[1] | 0.799 | 0.728 | 0.622 | - | - |
| Hu et al.[22] | 0.859 | 0.708 | 0.802 | 0.787 | 0.652 |
| MolCLR[15] | 0.890 | 0.736 | 0.806 | 0.787 | 0.652 |
| GraphMVP-C[31] | 0.812 | 0.724 | 0.770 | 0.744 | 0.639 |
| GEM[15] | 0.856 | 0.724 | 0.806 | 0.781 | 0.672 |
| MGCN[33] | 0.734 | 0.850 | 0.738 | 0.707 | 0.552 |
| GCN[24] | 0.716 | 0.718 | 0.740 | 0.709 | 0.536 |
| GIN[59] | 0.701 | 0.658 | 0.753 | 0.740 | 0.573 |
| SchNet[44] | 0.766 | 0.848 | 0.702 | 0.772 | 0.539 |
| KPGT[25] | 0.855 | 0.908 | - | 0.848 | 0.649 |

### 5.3 Ablation Study

Our proposed framework obtains unified embeddings by integrating knowledge from Zero-Shot

CoT and Few-Shot ICL learning methods, utilizing prompting based on LLMs (refer to Subsection 2.3) and GNNs (refer to Subsection 2.2). We conducted ablation studies to examine the impact of each method on the overall enhanced performance of our framework. By selectively disabling methods, we created multiple ablated variants of our framework and evaluated them using benchmark datasets for the property prediction task. This approach enabled us determine the contributions of the disabled methods to the overall framework performance. We chose the proposed `MMF` framework as the reference baseline for the ablation studies. This rigorous approach not only validates the effectiveness of the different methods but also provides justification for their design choices and inclusion within the framework. Our proposed framework operates through a multi-step pipeline, as follows:

- ✓ Synergistic Cross-Modal Embedding Generation (SEG): The Zero-Shot CoT prompts LLMs to generate technical descriptions of chemical SMILES representations. These descriptions are then used to fine-tune smaller LMs for domain-specific customization, facilitating the computation of context-aware token embeddings. We utilize softmax attention pooling to obtain text-level embeddings from these contextualized token embeddings. Simultaneously and in parallel, the Graph Chebyshev Convolution operator computes graph-level embeddings. These two sets of embeddings are synergistically integrated using a multi-head attention mechanism, thereby establishing a robust framework for integrating structured and unstructured data.
- ✓ Predictive Embedding Generation (PEG): The Few-Shot ICL method prompts LLMs to predict molecular properties based on a few demonstrations of the downstream task, which are then subsequently encoded to obtain prediction embeddings. ICL guides predictions without requiring explicit fine-tuning, relying solely on the implicit pre-trained knowledge encapsulated within the LLMs' parameters, and conditioned on the context-augmented prompts.
- ✓ MOE Dynamic Prediction (MOE-DP): For the output layer, we employ the MOE technique with a gating mechanism, where cross-modal and prediction embeddings are unified through the gating mechanism. The goal is to optimize predictive performance and fine-tune the embeddings, leveraging diverse learning strategies for precise molecular property predictions.

The ablated variants without the synergistic embedding generation (SEG), predictive embedding generation (PEG), and MOE dynamic prediction (MOE-DP) methods are referred to as 'w/o SEG,' 'w/o PEG,' and 'w/o MOE-DP,' respectively. In the case of 'w/o MOE-DP,' we utilize a linear operator to predict the molecular properties. The experimental findings from the ablation study are shown in Tables 8 and 9. All experiments were conducted using the scaffold technique with $K$=16. In the ablation study focusing on molecular property estimation, we evaluated the performance of the ablated variants against the baseline using the MAE error metric, allowing for a comprehensive analysis. Upon examination, it becomes evident that the synergistic embedding generation (SEG) method within the `MMF` framework holds greater significance than the predictive embedding generation (PEG) and MOE dynamic prediction (MOE-DP) methods for attaining state-of-the-art performance on the benchmark datasets. For the QM8 dataset, the 'w/o SEG' variant shows a substantial decline in performance relative to the baseline, as evidenced by a marked increase of $46.44\%$ in MAE. In contrast, the 'w/o PEG' variant exhibits marginally inferior performance compared to the baseline, with a modest increase of $14.49\%$ in MAE. Similarly, the 'w/o MOE-DP' variant performs much worse than the baseline, with an increase of $20.41\%$ in MAE. This increase in error might be attributed to the substitution with an oversimplified linear operator in the output layer. Similar trends were observed in the QM9 dataset. The higher increase in the performance metrics of the ablated variants, when compared to the baseline, underscores the relative significance of the mechanisms underpinning the omitted methods of the baseline. The experimental findings suggest that integrating knowledge from both text and graph modalities is a promising approach for enhancing the overall performance of the framework. Specifically, our innovative framework leverages a fusion of deep learning techniques, including GNNs and both larger and smaller language models, to predict molecular properties. The zero-shot CoT prompting of LLMs generates technical descriptions of molecules, which are then encoded by smaller language models to generate text-level embeddings. These text-level embeddings are seamlessly integrated with graph embeddings, resulting in improved cross-modal embeddings. We jointly optimize the cross-modal embeddings and the prediction embeddings generated by the few-shot ICL prompting of LLMs through a MOE technique with gating mechanism and then combine them to enhance the performance of our framework. In conclusion, this holistic approach not only demonstrates the benefits of combining diverse knowledge sources for optimized molecular property predictions but also provides a comprehensive understanding of molecular representations, setting a benchmark in the domain.

| Methods | Validation MAE ($\times 1.0e^{-3}$) | Test MAE ($\times 1.0e^{-3}$) |
|---|---|---|
| **MMF W/GPT-4** | **7.63 ± 0.07** | **7.45± 0.03** |
| w/o SEG | 10.86 ± 0.02 | 10.91 ± 0.05 |
| w/o PEG | 8.47± 0.03 | 8.53± 0.02 |
| w/o MOE-DP | 8.89± 0.04 | 8.97 ± 0.03 |

Table 8: The table presents the ablation study results on the QM8 dataset.

| Target | Unit | **MMF W/GPT-4** | w/o SEG | w/o PEG | w/o MOE-DP |
|---|---|---|---|---|---|
| $\mu$ | D | $1.06 \times 10^{-2}$ | $2.51 \times 10^{-2}$ | $1.24 \times 10^{-2}$ | $1.79 \times 10^{-2}$ |
| $\alpha$ | $a_0^3$ | $2.19 \times 10^{-2}$ | $4.56 \times 10^{-2}$ | $2.98 \times 10^{-2}$ | $3.65 \times 10^{-2}$ |
| $\epsilon_{\text{HOMO}}$ | meV | $1.843 \times 10^{1}$ | $2.936 \times 10^{1}$ | $2.105 \times 10^{1}$ | $2.598 \times 10^{1}$ |
| $\epsilon_{\text{LUMO}}$ | meV | $9.57$ | $1.991 \times 10^{1}$ | $1.243 \times 10^{1}$ | $1.577 \times 10^{1}$ |
| $\Delta\epsilon$ | meV | $2.234 \times 10^{1}$ | $3.291 \times 10^{1}$ | $2.457 \times 10^{1}$ | $2.709 \times 10^{1}$ |
| $\langle R^2 \rangle$ | $a_0^2$ | $1.08 \times 10^{-1}$ | $2.78 \times 10^{-1}$ | $1.43 \times 10^{-1}$ | $2.21 \times 10^{-1}$ |
| ZPVE | meV | $7.85 \times 10^{-2}$ | 1.35 | $9.6 \times 10^{-2}$ | 1.14 |
| $U_0$ | meV | 3.55 | 7.81 | 4.48 | 6.29 |
| $U$ | meV | 2.43 | 7.76 | 3.14 | 5.97 |
| $H$ | meV | 3.09 | 8.24 | 4.15 | 5.04 |
| $G$ | meV | 4.23 | 9.32 | 5.17 | 6.83 |
| $c_{\text{v}}$ | /mol/K | $1.37 \times 10^{-2}$ | $3.11 \times 10^{-2}$ | $1.62 \times 10^{-2}$ | $2.53 \times 10^{-2}$ |

Table 9: The table presents the ablation study results on the QM9 dataset.

## 5.4 Additional In-Depth Studies

In this section, we will analyze more into the synergistic cross-modal embedding generation(SEG) and predictive embedding generation(PEG) methods.

### 5.4.1 Impact of Synergistic Cross-Modal Embedding Generation(SEG) method

The synergistic embedding generation(SEG) method consists of three main components:

- ✓ Text-Level Embeddings (TL-Emb): Computed using softmax attention pooling of contextual token embeddings. These context-aware embeddings are obtained from fine-tuning smaller LMs on the technical descriptions generated by the zero-shot CoT prompting of LLMs on chemical SMILES representations.
- ✓ Graph-Level Embeddings (GL-Emb): Computed through Graph Chebyshev Convolution technique on the corresponding non-linear molecular graph representations obtained from the chemical SMILES representations.
- ✓ Cross-Modal Embeddings (CM-Emb): Computed using a multi-head attention mechanism to facilitate the robust integration and analysis of text-level and graph-level embeddings.

We conducted ablation studies to examine the impact of different embeddings within the Synergistic Embedding Generation (SEG) method on the overall improved performance of our framework. The ablated variants without the text-level embeddings (TL-Emb), graph-level embeddings (GL-Emb), and cross-modal embeddings (CM-Emb) methods are referred to as 'w/o TL-Emb', 'w/o GL-Emb', and 'w/o CM-Emb', respectively. In the case of 'w/o CM-Emb', we concatenate the embeddings and utilize a linear operator to predict the cross-modal embeddings. The experimental findings from the ablation study are presented in Tables 10 and 11.

| Methods | Validation MAE ($\times 1.0e^{-3}$) | Test MAE ($\times 1.0e^{-3}$) |
|---|---|---|
| **MMF W/GPT-4** | **7.63 ± 0.07** | **7.45± 0.03** |
| w/o TL-Emb | 9.43± 0.06 | 9.51± 0.02 |
| w/o GL-Emb | 8.11± 0.04 | 8.16± 0.07 |
| w/o CM-Emb | 10.15± 0.03 | 10.27± 0.05 |

Table 10: The table presents the impact of various embeddings within the synergistic embedding generation (SEG) method on the QM8 dataset. All experiments were conducted using the scaffold technique with $K$=16.

Upon closer examination, it becomes evident that text-level embeddings (TL-Emb) are more significant than graph-level embeddings (GL-Emb) in achieving state-of-the-art performance on the benchmark datasets. For the QM8 dataset, the ‘w/o TL-Emb’ variant demonstrates a substantial decline in performance compared to the baseline, as evidenced by a significant increase of $27.65\%$ in MAE. In contrast, the ‘w/o GL-Emb’ variant exhibits only a marginally inferior performance compared to the baseline, with a modest increase of $9.53\%$ in MAE. The ‘w/o CM-Emb’ variant shows the poorest performance relative to the baseline with an increase of $37.85\%$ in MAE. This increase in error may be attributed to the substitution of an oversimplified concatenation of text-level and graph-level embeddings, along with the utilization of a linear operator to predict the cross-modal embeddings. Similar trends were observed in the QM9 dataset. The greater increase in performance metrics for the ablated variants compared to the baseline underscores the relative significance of the omitted methods

| Target | Unit | **MMF W/GPT-4** | w/o TL-Emb | w/o GL-Emb | w/o CM-Emb |
|---|---|---|---|---|---|
| $\mu$ | D | $1.06 \times 10^{-2}$ | $1.58 \times 10^{-2}$ | $1.39 \times 10^{-1}$ | $2.57 \times 10^{-1}$ |
| $\alpha$ | $a_0^3$ | $2.19 \times 10^{-2}$ | $3.37 \times 10^{-2}$ | $3.08 \times 10^{-2}$ | $4.13 \times 10^{-2}$ |
| $\epsilon_{\text{HOMO}}$ | meV | $1.843 \times 10^1$ | $2.419 \times 10^1$ | $2.331 \times 10^1$ | $2.697 \times 10^1$ |
| $\epsilon_{\text{LUMO}}$ | meV | $9.57$ | $1.396 \times 10^1$ | $1.228 \times 10^1$ | $1.896 \times 10^1$ |
| $\Delta\epsilon$ | meV | $2.234 \times 10^1$ | $2.765 \times 10^1$ | $2.689 \times 10^1$ | $3.107 \times 10^1$ |
| $\langle R^2 \rangle$ | $a_0^2$ | $1.08 \times 10^{-1}$ | $1.46 \times 10^{-1}$ | $1.53 \times 10^{-1}$ | $2.92 \times 10^{-1}$ |
| ZPVE | meV | $7.85 \times 10^{-2}$ | $9.8 \times 10^{-2}$ | $8.7 \times 10^{-2}$ | $1.21$ |
| $U_0$ | meV | $3.55$ | $5.17$ | $4.92$ | $7.88$ |
| $U$ | meV | $2.43$ | $3.19$ | $3.37$ | $7.71$ |
| $H$ | meV | $3.09$ | $4.16$ | $3.97$ | $7.75$ |
| $G$ | meV | $4.23$ | $5.05$ | $4.92$ | $7.89$ |
| $c_{\text{v}}$ | $/\text{mol}/\text{K}$ | $1.37 \times 10^{-2}$ | $1.87 \times 10^{-2}$ | $1.72 \times 10^{-2}$ | $2.23 \times 10^{-2}$ |

Table 11: The table showcases the influence of diverse embeddings in the synergistic embedding generation (SEG) method on the QM9 dataset. All experiments were conducted using the scaffold technique with $K$=16.

### 5.4.2 Impact of Predictive Embedding Generation(PEG) method

In-context learning (ICL) or few-shot prompting enables pretrained foundational large language models (LLMs) to adapt to new tasks with only a few task-specific demonstrations, thus eliminating the need for parameter updates. This approach represents a departure from traditional supervised learning methods. In our study, we employ few-shot prompting with LLMs to predict molecular properties of new, unseen molecules. We construct context-augmented prompts, comprising task-specific instructions and demonstrations (inputs in the form of chemical SMILES strings and outputs as molecular property pairs) sampled from the training data to direct general-purpose LLMs in predicting molecular properties of the target chemical SMILES strings. In essence, the instruction serves to contextualize the task of predicting molecular properties for the target chemical SMILES strings, while the demonstrations are crucial in guiding the language model to produce relevant and accurate responses for the specific task at hand. In this scenario, the augmented prompt guides the LLMs to tap into the pre-existing knowledge embedded within their parameters, acquired during training on vast and diverse corpora, to predict the molecular graph properties. We explore two sampling strategies, "Random" and "Scaffold", to construct augmented prompts, and we evaluate the predictive abilities of LLMs that rely solely on these prompts. Our experiments focus on both the quality (how helpful are the demonstrations in reducing prediction error) and quantity (number of demonstrations) of these sampling methods, aiming to enhance property prediction accuracy.

**Results:** Tables 12 and 13 present the MAE scores, showcasing the results of property prediction performance in our study on the impact of both quality and quantity of demonstrations during few-shot prompting of LLMs. Our analysis revealed that the `MMF W/GPT` models outperformed the `MMF W/Google Bard` in terms of MAE on both datasets under examination. Notably, among the evaluated `MMF W/GPT` models, `MMF W/GPT-4` exhibited superior performance compared to `MMF W/davinci-003` and `MMF W/GPT-3.5 Turbo` in predicting molecular properties. Our study supports the notion that increasing training examples in few-shot prompting(in-context learning) can significantly enhance framework performance. It underscores a direct correlation between the volume of ICL demonstrations and the predictive accuracy of the LLMs. Furthermore, our investigation provides compelling evidence that scaffold sampling consistently outperforms random sampling across distinct datasets (QM8 and QM9), further bolstering the empirical validity of our research. A possible reason for this might be the structural similarities between the molecules sampled from the scaffold strategy and the query molecule. This could potentially tilt the `MMF GPT` models towards making more precise decisions. LLMs still face a significant limitation in understanding molecular representations in SMILES strings, resulting in inaccurate results in property prediction tasks.

SMILES notation is a widely used textual representation for chemical structures, but LLMs struggle to interpret it accurately due to issues like implicit hydrogen atoms, multiple valid representations for a single molecule, which leads to ambiguity, and treating SMILES strings as mere sequences of characters. This hampers the performance of LLMs in tasks such as property prediction and affects downstream cheminformatics tasks. Therefore, LLMs with improved capabilities in handling molecular structures and coupling with existing tools such as RDKit will be necessary.

| Methods | Validation MAE ($\times 1.0e^{-3}$) | Test MAE ($\times 1.0e^{-3}$) |
|---|---|---|
| MMF W/ GPT-4 (Scaffold, $K$=4) | 9.83 ± 0.05 | 9.89± 0.07 |
| MMF W/ GPT-4 (Scaffold, $K$=12) | 8.24 ± 0.04 | 8.36± 0.02 |
| MMF W/ GPT-4 (random, $K$=12) | 10.67 ± 0.03 | 10.72± 0.06 |
| MMF W/ GPT-3.5 (Scaffold, $K$=4) | 10.35 ± 0.04 | 10.41 ± 0.06 |
| MMF W/ GPT-3.5 (Scaffold, $K$=12) | 8.93 ± 0.08 | 8.98 ± 0.05 |
| MMF W/ GPT-3.5 (random, $K$=12) | 11.03 ± 0.07 | 11.17 ± 0.03 |
| MMF W/ davinci-003 (Scaffold, $K$=4) | 10.13 ± 0.03 | 10.09 ± 0.06 |
| MMF W/ davinci-003 (Scaffold, $K$=12) | 8.87 ± 0.07 | 8.96 ± 0.09 |
| MMF W/ davinci-003 (random, $K$=12) | 10.98 ± 0.05 | 11.02 ± 0.04 |
| MMF W/ Google Bard (Scaffold, $K$=12) | 9.53± 0.04 | 9.67 ± 0.03 |
| MMF W/ Google Bard (random, $K$=12) | 11.45± 0.05 | 11.63 ± 0.07 |

Table 12: The table shows the MAE scores of the `MMF W/GPT` models for predicting molecular properties on the QM8 dataset. Here, $K$ represents the number of training examples used in few-shot prompting. All experiments were conducted using the scaffold technique with $K$=16.

| Target | Unit | MMF W/ GPT-4 (Scaffold, $K$=16) | MMF W/ GPT-4 (Scaffold, $K$=4) | MMF W/GPT-davinci (Scaffold, $K$=16) | MMF W/GPT-davinci (Scaffold, $K$=4) |
|---|---|---|---|---|---|
| $\mu$ | D | $1.06 \times 10^{-2}$ | $1.32 \times 10^{-2}$ | $1.27 \times 10^{-2}$ | $1.37 \times 10^{-2}$ |
| $\alpha$ | $a_0^3$ | $2.19 \times 10^{-2}$ | $4.43 \times 10^{-2}$ | $2.49 \times 10^{-2}$ | $4.83 \times 10^{-2}$ |
| $\epsilon_{\text{HOMO}}$ | meV | $1.843 \times 10^{1}$ | $2.602 \times 10^{1}$ | $2.147 \times 10^{1}$ | $3.239 \times 10^{1}$ |
| $\epsilon_{\text{LUMO}}$ | meV | 9.57 | $1.758 \times 10^{1}$ | $1.185 \times 10^{1}$ | $1.942 \times 10^{1}$ |
| $\Delta\epsilon$ | meV | $2.234 \times 10^{1}$ | $3.043 \times 10^{1}$ | $2.597 \times 10^{1}$ | $3.251 \times 10^{1}$ |
| $\langle R^2 \rangle$ | $a_0^2$ | $1.08 \times 10^{-1}$ | $2.16 \times 10^{-1}$ | $1.42 \times 10^{-1}$ | $2.53 \times 10^{-1}$ |
| ZPVE | meV | $7.85 \times 10^{-2}$ | $9.5 \times 10^{-2}$ | $9.05 \times 10^{-2}$ | 1.073 |
| $U_0$ | meV | 3.55 | 5.34 | 4.52 | 6.05 |
| $U$ | meV | 2.43 | 4.92 | 3.97 | 6.89 |
| $H$ | meV | 3.09 | 5.01 | 3.72 | 5.85 |
| $G$ | meV | 4.23 | 5.23 | 4.89 | 5.97 |
| $c_{\text{v}}$ | /mol/K | $1.37 \times 10^{-2}$ | $2.03 \times 10^{-2}$ | $1.58 \times 10^{-2}$ | $1.77 \times 10^{-2}$ |

Table 13: The table presents the MAE scores achieved by the `MMF W/GPT` models when predicting molecular properties on the QM9 dataset. All experiments were performed with the scaffold technique, using a value of $K$=16. In this context, $K$ denotes the number of training samples utilized in few-shot prompting.

### 5.5 Hyperparameter Tuning

Hyperparameters are parameters that are not learned from data but are set prior to the training process. They have a direct impact on the performance of the framework. To optimize the performance of our `MMF` framework, we conducted in-depth hyperparameter tuning through careful experimentation and analysis. We chose to use random search as an efficient method for exploring hyperparameters and identifying the best framework configuration on benchmark datasets, rather than employing computationally intensive methods like grid search or Bayesian optimization. This approach allowed us to achieve optimal performance on the test dataset across various benchmark datasets, as measured by the MAE metric. We identified a set of hyperparameters that significantly improved the `MMF` framework's performance. We did not fine-tune large language models (LLMs) for the downstream property prediction task; instead, we accessed LLMs through text-based API interactions. We fine-tuned small-scale LMs using LLM-generated textual descriptions for the molecular property prediction task, while minimizing supervised regression loss. Hyperparameter optimization was performed on the `MMF-W/GPT-4` variant of our framework. The key prioritized hyperparameters for this framework are batch size ($b \in 32, 48, 64$) and hidden or embedding dimension ($d \in 64, 128, 196, 256$). Tables 14 and 15 present the results of hyperparameter tuning on representative benchmark datasets. We report the results for the near-optimal combinations of hyperparameters. All experiments were conducted using the GPT-4 framework with scaffold technique, setting $K$=16. In summary, based on the experimental results presented in tables 14 and 15, the best hyperparameters for the `MMF` framework—particularly for the QM-8 dataset and many targets in the QM-9 dataset—appear to be a batch size ($b$) of 32 and an embedding dimension ($d$) of 128.

| Methods | Validation MAE ($\times 1.0e^{-3}$) | Test MAE ($\times 1.0e^{-3}$) |
|---|---|---|
| MMF ($b$=32, $d$=128) | **7.63 ± 0.07** | **7.45± 0.03** |
| MMF ($b$=32, $d$=256) | 8.34 ± 0.02 | 8.41± 0.05 |
| MMF ($b$=48, $d$=128) | 8.13 ± 0.06 | 8.25± 0.03 |
| MMF ($b$=64, $d$=256) | 8.57 ± 0.09 | 8.67± 0.04 |

Table 14: The table presents the hyperparameter study results on the QM-8 dataset.

| Target | Unit | MMF($b$=32, $d$=128) | MMF($b$=32, $d$=256) | MMF($b$=48, $d$=128) | MMF($b$=64, $d$=256) |
|---|---|---|---|---|---|
| $\mu$ | D | $1.06 \times 10^{-2}$ | $1.27 \times 10^{-2}$ | $1.19 \times 10^{-2}$ | $1.34 \times 10^{-2}$ |
| $\alpha$ | $a_0^3$ | $2.19 \times 10^{-2}$ | $2.77 \times 10^{-2}$ | $2.24 \times 10^{-2}$ | $3.11 \times 10^{-2}$ |
| $\epsilon_{\text{HOMO}}$ | meV | $1.843 \times 10^{1}$ | $2.236 \times 10^{1}$ | $2.053 \times 10^{1}$ | $2.368 \times 10^{1}$ |
| $\epsilon_{\text{LUMO}}$ | meV | 9.57 | $1.078 \times 10^{1}$ | $1.041 \times 10^{1}$ | $1.173 \times 10^{1}$ |
| $\Delta\epsilon$ | meV | $2.234 \times 10^{1}$ | $2.662 \times 10^{1}$ | $2.405 \times 10^{1}$ | $2.815 \times 10^{1}$ |
| $\langle R^2 \rangle$ | $a_0^2$ | $1.08 \times 10^{-2}$ | $1.21 \times 10^{-2}$ | $1.14 \times 10^{-2}$ | $1.39 \times 10^{-2}$ |
| ZPVE | meV | $7.85 \times 10^{-1}$ | $9.17 \times 10^{-1}$ | $8.42 \times 10^{-1}$ | $9.89 \times 10^{-1}$ |
| $U_0$ | meV | 3.55 | 3.83 | 3.67 | 4.07 |
| $U$ | meV | 2.43 | 2.55 | 2.51 | 2.66 |
| $H$ | meV | 3.09 | 3.31 | 3.14 | 3.44 |
| $G$ | meV | 4.23 | 4.68 | 4.41 | 4.97 |
| $c_{\text{v}}$ | /mol/K | $3.3 \times 10^{-2}$ | $8.1 \times 10^{-2}$ | $4.7 \times 10^{-2}$ | $9.9 \times 10^{-2}$ |

Table 15: The table presents the hyperparameter study results on the QM-9 dataset.

### 5.6 Large Language Models and Prompting

Large Language Models (LLMs) have brought about a paradigm shift in natural language processing for task adaptation with the 'pre-train, prompt, and predict' approach. This approach allows for more generalized adaptation to various tasks by using natural language instructions as prompts, without the need for extensive fine-tuning. It is flexible and efficient, enabling LLMs to perform a wide range of NLP tasks with minimal task-specific adaptation. This is achieved through demonstrations that allow LLMs to learn from analogy. Consequently, it has replaced the previously established 'pre-train, fine-tune' approach, which involves customizing the LLMs for each task through fine-tuning using task-specific labeled data. In this new paradigm, the LLM undergoes an initial phase of pre-training on a vast collection of unannotated text corpora without explicit human supervision. This process facilitates the learning of grammar, syntax, semantics, and even some level of common-sense reasoning, allowing the LLM to achieve improved linguistic comprehension and generate human-like responses. Instead of fine-tuning the language model with task-specific labeled data, as is customary in conventional approaches, this approach prompts the LLM with a natural language query that explicitly outlines the task and context. Based on this prompt, the LLM directly generates the desired output, using its pre-existing knowledge[30] embedded within its trainable parameters acquired during pretraining. A prompt is a textual input provided to an instruction-based or prompt-based language model to guide its behavior and generate desired outputs. It can take the form of a phrase, question, or sentence and may involve supplementary information or constraints. The prompt sets the context, provides instructions, and helps shape the LLM's responses to align with the desired outcome. LLMs possess the ability to generate responses based on contextual information and prior training, allowing them to provide context-aware and coherent responses in conversations. Moreover, LLMs can be further customized and enhanced through fine-tuning, which involves using reinforcement learning techniques[37, 42] with human feedback to optimize their performance for specialized tasks or domains. This adaptability and versatility make LLMs powerful tools for a wide range of natural language processing applications. Consider an LLM, which accepts an input sequence of tokens $x = (x_1, x_2, ..., x_n)$ and outputs a token sequence $y = (y_1, y_2, ..., y_m)$. The LLM model is generally trained to optimize a conditional probability distribution $p(y|x)$, which assigns the probability to each possible output sequence $y$ given $x$. A prompt, denoted as $p$, can be integrated with the input sequence $x$ by concatenating the elements of $p$ to the beginning of the input sequence, resulting in a new sequence $\hat{x} = (p, x_1, x_2, ..., x_n)$. We utilize $\hat{x}$ to calculate the conditional probability distribution $p(y|\hat{x})$. Formally, the probability of the output sequence $y$ given $\hat{x}$ can be represented as:

$$p(y|\hat{x}) = \prod_{i=1}^{m} p(y_i|y_{<i}, \hat{x})$$

where $y_{<i}$ denotes the prefix of the sequence $y$ up to position $i-1$, and $p(y_i|y_{<i}, \hat{x})$ represents the probability of generating token $y_i$ given $y_{<i}$ and $\hat{x}$. The model's attention mechanism can identify the most relevant parts of the input prompt during each output token generation, facilitating the computation of $p(y_i|y_{<i}, \hat{x})$. In summary, the prompt plays a crucial role in shaping the conditional probability distribution, as it provides relevant information and context to guide the generation of the desired output. Zero-shot prompting is a process in NLP in which large language models (LLMs), such as GPT-3.5 or Google Bard, are equipped to perform specific tasks without undergoing task-specific training. This ability is derived from their initial training on extensive and diverse text corpora, enabling them to access a broad spectrum of general knowledge. This method utilizes detailed, context-rich prompts to outline the task at hand, guiding the LLM to generate suitable responses based on its foundational knowledge, thereby demonstrating the language model's zero-shot learning capabilities. Few-shot prompting is a technique in NLP that allows LLMs to perform a task with a few task-specific demonstrations. This is accomplished by first pre-training the LLM on a massive volume of text corpora, which allows it to learn a vast range of general knowledge. Then, the LLM is given a few task-specific input-output pairs and is trained to generate similar outputs. The LLM uses its pre-existing knowledge to generalize from the few demonstrations and learn to perform the task. In our work, we harness the zero-shot reasoning of LLMs, acquired through pre-training on vast text corpora. This empowers LLMs to enhance molecular graph-level embeddings from GNNs with richer textual information about molecular properties, reactivity, and more. This capability enables LLMs to tackle complex problems without requiring dedicated task-specific training. Table 16 shows the LLM(GPT-3.5)-retrieved text for a natural language query about an organic molecule in SMILES notation “CC(=O)C". Additionally, we leverage LLMs' few-shot learning abilities to predict molecular properties of unseen molecules using demonstrations from training data that consists of chemical SMILES strings and property pairs.

## References

[1] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.

[2] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems*, 32, 2019.

[3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[4] James Atwood and Don Towsley. Diffusion-convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[7] Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9):3564–3572, 2019.

[8] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[10] John S Delaney. Esol: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences*, 44(3):1000–1005, 2004.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *International Conference on Machine Learning*, pages 7865–7885. PMLR, 2023.

[13] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

[14] Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.

[15] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.

[16] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

[17] Anna Gaulton, Anne Hersey, Michał Nowotka, A Patricia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J Bellis, Elena Cibrián-Uhalte, et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

[18] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[19] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[20] Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited. *Advances in Neural Information Processing Systems*, 35:7264–7276, 2022.

[21] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[22] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[25] Han Li, Dan Zhao, and Jianyang Zeng. Kpgt: knowledge-guided pre-training of graph transformer for molecular property prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 857–867, 2022.

[26] Pan Li and Jure Leskovec. The expressive power of graph neural networks. *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 63–98, 2022.

[27] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[28] Renjie Liao, Marc Brockschmidt, Daniel Tarlow, Alexander L Gaunt, Raquel Urtasun, and Richard Zemel. Graph partition neural networks for semi-supervised classification. *arXiv preprint arXiv:1803.06272*, 2018.

[29] Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard S Zemel. Lanczosnet: Multi-scale deep graph convolutional networks. *arXiv preprint arXiv:1901.01484*, 2019.

[30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[31] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.

[32] Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks. *arXiv preprint arXiv:2102.05013*, 2021.

[33] Chengqiang Lu, Qi Liu, Chao Wang, Zhenya Huang, Peize Lin, and Lixin He. Molecular property prediction: A multilevel quantum interactions modeling perspective. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1052–1060, 2019.

[34] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *Advances in neural information processing systems*, 32, 2019.

[35] David L Mobley and J Peter Guthrie. Freesolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design*, 28:711–720, 2014.

[36] Harry L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

[37] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[38] Elton Pan, Christopher Karpovich, and Elsa Olivetti. Deep reinforcement learning for inverse inorganic materials design. *arXiv preprint arXiv:2210.11931*, 2022.

[39] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[40] Raghunathan Ramakrishnan, Mia Hartmann, Enrico Tapavicza, and O Anatole Von Lilienfeld. Electronic spectra from tddft and machine learning in chemical space. *The Journal of chemical physics*, 143(8), 2015.

[41] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

[42] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[43] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[44] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

[45] Saketh Sridhara, Aaditya Chandrasekhar, and Krishnan Suresh. A generalized framework for microstructural optimization using neural networks. *Materials & Design*, 223:111213, 2022.

[46] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[47] Taffee T Tanimoto. Elementary mathematical theory of classification and prediction. 1958.

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[49] Devis Tuia, Konrad Schindler, Begüm Demir, Gustau Camps-Valls, Xiao Xiang Zhu, Mrinalini Kochupillai, Sašo Džeroski, Jan N van Rijn, Holger H Hoos, Fabio Del Frate, et al. Artificial intelligence to advance earth observation: a perspective. *arXiv preprint arXiv:2305.08413*, 2023.

[50] Oliver T Unke and Markus Meuwly. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation*, 15(6):3678–3693, 2019.

[51] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477, 2019.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[53] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[54] Jean-Philippe Vert. How will generative ai disrupt data science in drug discovery? *Nature Biotechnology*, pages 1–2, 2023.

[55] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[56] Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. The pdbbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry*, 47(12):2977–2980, 2004.

[57] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[58] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[59] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[60] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[61] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

[62] Atakan Yüksel, Erva Ulusoy, Atabey Ünlü, and Tunca Doğan. Selformer: Molecular representation learning via selfies language models. *Machine Learning: Science and Technology*, 2023.

[63] Shifa Zhong, Kai Zhang, Majid Bagheri, Joel G Burken, April Gu, Baikun Li, Xingmao Ma, Babetta L Marrone, Zhiyong Jason Ren, Joshua Schrier, et al. Machine learning: new ideas and tools in environmental science and engineering. *Environmental Science & Technology*, 55(19):12741–12754, 2021.

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| Query prompt | LLM-retrieved text |
|---|---|
| What is the molecular structure of this organic molecule in SMILES notation "CC(=O)C". Could you describe its atoms, bonds, functional groups, and overall arrangement? | The organic molecule in SMILES notation "CC(=O)C" represents the compound acetone, which is a simple ketone. Let's break down the SMILES notation and describe its molecular structure:<br>SMILES notation: "CC(=O)C"<br>Atoms:<br>C: Represents carbon atoms. O: Represents an oxygen atom.<br>Bonds:<br>Single bonds (represented by just the atom symbols): There are three single bonds between the carbon atoms (C-C). Double bond (represented by "="): There is one double bond between the carbon and oxygen atoms (C=O).<br>Functional groups:<br>Carbonyl group: The functional group "C=O" is a carbonyl group, which is a carbon atom double-bonded to an oxygen atom.<br>Overall arrangement: Acetone (CC(=O)C) is a simple three-carbon compound. It consists of three carbon atoms and one oxygen atom. The carbon atoms are connected in a chain, and one of the carbons is double-bonded to an oxygen atom, forming the carbonyl group. The molecule forms a linear structure with the carbonyl group in the middle. The "CH3" groups on both sides represent methyl (CH3) groups, which are single-bonded to the adjacent carbon atoms. The central carbon is double-bonded to the oxygen atom, forming the carbonyl functional group (C=O). Acetone is a common organic solvent and is found in some household products like nail polish removers and paint thinners. It is also used in chemical and industrial processes. |
| What are the physical properties of this molecule such as its boiling point, melting point, and density? | The physical properties of acetone (CC(=O)C) are as follows:<br>Boiling Point: The boiling point of acetone is approximately 56.5°C (approximately 133.7°F). This relatively low boiling point makes it a volatile liquid, which means it evaporates readily at room temperature.<br>Melting Point: The melting point of acetone is around -94.9°C (approximately -138.8°F). As with its boiling point, the low melting point is characteristic of many small organic molecules.<br>Density: The density of acetone is about 0.79 g/cm³ (at 20°C or 68°F). It is less dense than water, which has a density of 1 g/cm³ at the same temperature. This lower density is why acetone floats on water.<br>Acetone is a colorless, flammable liquid with a characteristic odor. It is highly miscible with water and many other organic solvents, making it a versatile solvent in various industries, laboratories, and domestic applications. It is important to note that these physical properties can vary slightly depending on the specific experimental conditions and purity of the acetone sample. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| Query prompt | LLM-retrieved text |
|---|---|
| What is the solubility behavior of this molecule? In which solvents does it dissolve and which does it not? | Acetone (CC(=O)C) is a highly versatile solvent due to its unique solubility behavior. It is classified as a polar aprotic solvent, which means it has a significant dipole moment and can dissolve a wide range of polar and nonpolar substances. Here's a summary of its solubility behavior:<br>Solubility of Acetone:<br>Water (H2O): Acetone is highly miscible with water. It can dissolve in water in all proportions, forming a homogeneous mixture. This property makes acetone an excellent solvent for substances that are water-soluble or partially water-soluble.<br>Polar Solvents: Acetone is miscible with many polar solvents, including alcohols, ethers, ketones, and some chlorinated solvents. It can dissolve various organic compounds that contain polar functional groups, such as hydroxyl (OH) and carbonyl (C=O) groups.<br>Nonpolar Solvents: Acetone can also dissolve in nonpolar solvents to some extent. However, its solubility in nonpolar solvents is not as high as in polar solvents. Acetone's ability to dissolve nonpolar substances is due to its relatively small size and nonpolar "tail" (methyl groups) in its molecular structure.<br>Examples of Solvents in which Acetone Dissolves:<br>Ethanol, Diethyl ether, Ethyl acetate, Chloroform, Carbon tetrachloride, Hexane.<br>Solvents in which Acetone Does Not Dissolve: Acetone is not suitable as a solvent for some highly nonpolar substances or compounds that lack any polar functional groups. These may include certain hydrocarbons and nonpolar solvents like benzene and toluene.<br>As a general rule, acetone is an effective solvent for a wide range of organic compounds, but its solubility behavior depends on the specific properties of the substances being dissolved. Additionally, in some cases, the solubility of acetone may be limited by competing chemical reactions or phase separations, especially at high concentrations or with certain reactants. Always perform solubility tests and consult reference sources when choosing a solvent for a specific application. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| **Query prompt** | **LLM-retrieved text** |
|---|---|
| What is the chemical reactivity of this molecule? How does it interact with various reagents? | Acetone (CC(=O)C) is a chemically reactive molecule due to the presence of the carbonyl group (C=O) in its structure. Its chemical reactivity allows it to participate in various types of reactions with different reagents. Here are some common reactions that acetone can undergo:<br>Nucleophilic Addition Reactions: The carbonyl carbon in acetone is electrophilic, making it susceptible to attack by nucleophiles. In nucleophilic addition reactions, the C=O bond is broken, and a nucleophile adds to the carbonyl carbon. This leads to the formation of new compounds.<br>Example: Acetone + Hydrogen Cyanide → Cyanohydrin<br>Aldol Condensation: Acetone can undergo aldol condensation reactions, where two molecules of acetone react in the presence of a base to form a $\beta$-hydroxy ketone. This reaction is important in organic synthesis to build larger molecules.<br>Example: Acetone → Diacetone Alcohol<br>Reduction: Acetone can be reduced to produce different products, depending on the reagent used. For example, with sodium borohydride (NaBH4), acetone is reduced to yield isopropanol (2-propanol), while with lithium aluminum hydride (LiAlH4), it can be reduced to isopropyl alcohol (propan-2-ol).<br>Example: Acetone + NaBH4 → Isopropanol<br>Oxidation: Acetone can be oxidized to form various products, including carboxylic acids or esters, depending on the oxidizing agent and reaction conditions.<br>Example: Acetone + Acidified Potassium Dichromate → Acetic Acid<br>Halogenation: Acetone can undergo halogenation reactions, where halogens (e.g., chlorine or bromine) add to the carbonyl carbon, resulting in halogenated derivatives.<br>Example: Acetone + Bromine → 2,2-Dibromo-2-propanol<br>Polymerization: Acetone can polymerize under certain conditions, especially when exposed to strong acids or bases. This can lead to the formation of polyacetals.<br>Acetone's reactivity is crucial in various industrial processes and organic synthesis. However, its chemical reactivity also means that it should be handled with care and caution, especially when using it as a solvent or in chemical reactions, to prevent undesired reactions and ensure safety. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| **Query prompt** | **LLM-retrieved text** |
|---|---|
| Are there any common reactions that this molecule is known to undergo? Could you describe them? | Acetone (CC(=O)C) is known to undergo several common reactions due to the presence of its carbonyl group (C=O). Some of the most important reactions involving acetone are:<br>Nucleophilic Addition Reactions: Acetone can undergo nucleophilic addition reactions, where a nucleophile attacks the electrophilic carbon of the carbonyl group. This results in the breaking of the C=O bond and the formation of a new compound. One common nucleophilic addition reaction involving acetone is the formation of cyanohydrins: Example: Acetone + Hydrogen Cyanide → Cyanohydrin<br>Aldol Condensation: Acetone can undergo aldol condensation, a reaction where two molecules of acetone or an acetone molecule and another aldehyde or ketone react in the presence of a base. This reaction leads to the formation of a $\beta$-hydroxy ketone or aldol compound. Example: Acetone → Diacetone Alcohol<br>Haloform Reaction: Acetone can undergo the haloform reaction in the presence of halogens (e.g., chlorine or bromine) and hydroxide ions. This reaction results in the formation of a halogenated methyl ketone and a halide ion. Example: Acetone + Chlorine + Hydroxide → Chloroform + Acetate Ion<br>Cannizzaro Reaction: Under strong basic conditions, acetone can undergo the Cannizzaro reaction, which involves the disproportionation of the molecule into an alcohol and a carboxylic acid. Example: Acetone + Hydroxide → Isopropanol + Acetate Ion<br>Dehydration: Acetone can undergo dehydration in the presence of acid catalysts, such as sulfuric acid. In this reaction, water is eliminated from acetone to form an unsaturated compound. Example: Acetone → 2-Butanone (Methyl Ethyl Ketone)<br>Polymerization: Acetone can undergo polymerization under certain conditions, leading to the formation of polyacetals. These are just a few examples of the common reactions involving acetone. As a versatile solvent and reagent, acetone participates in various chemical processes in both industrial and laboratory settings. Its reactivity and ability to form a wide range of products make it an essential component in organic synthesis and many chemical applications. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| **Query prompt** | **LLM-retrieved text** |
|---|---|
| Does this molecule exhibit any unique optical, electrical, or magnetic properties? | Acetone (CC(=O)C) is a simple organic molecule that does not exhibit any unique optical, electrical, or magnetic properties under standard conditions. It is a colorless liquid at room temperature and does not display any inherent color, making it optically inactive. Acetone is also a non-conductive material, meaning it is not an electrical conductor, and it does not possess any special electrical properties.<br>Regarding magnetism, acetone is a diamagnetic substance. Diamagnetic materials are those that generate a weak magnetic field in opposition to an externally applied magnetic field. However, the diamagnetic effect is very weak in acetone and most organic compounds. As a result, the magnetic properties of acetone are generally not significant for most practical purposes.<br>It's worth noting that the lack of unique optical, electrical, or magnetic properties does not diminish the importance and versatility of acetone as a solvent and reactant in various chemical processes and applications. Its value lies in its role as a solvent, its ability to participate in various chemical reactions, and its use as an intermediate in the synthesis of various compounds in organic chemistry and industry. |
| Is this molecule chiral? If yes, how does its chirality influence its behavior or properties? | The acetone molecule (CC(=O)C) is not chiral. Chirality is a property of molecules that have non-superimposable mirror images, known as enantiomers. Chiral molecules possess an asymmetric carbon atom (a carbon atom bonded to four different groups), and this asymmetry gives rise to the existence of two enantiomeric forms.<br>In acetone, all three carbon atoms are bonded to two hydrogen atoms and one methyl (CH3) group each. Since all the carbon atoms have the same groups bonded to them, there is no chiral center, and acetone does not have enantiomers.The absence of chirality in acetone means that it behaves as an achiral molecule. It exhibits typical chemical and physical properties of a small ketone, such as its ability to act as a solvent, participate in various chemical reactions, and display its characteristic odor and volatility.<br>Chirality can play a significant role in the behavior and properties of some molecules, especially in biological systems. Chiral molecules often interact differently with other chiral molecules or biomolecules, such as enzymes or receptors, leading to different biological activities. The two enantiomers of a chiral molecule can have different pharmacological effects, toxicities, and physiological behaviors, which is important in drug development and medicinal chemistry.<br>However, since acetone is achiral, it does not exhibit such differences between enantiomers. Its chemical and physical properties are consistent, and both enantiomeric forms would be identical if they were to exist, which, in this case, they do not. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| **Query prompt** | **LLM-retrieved text** |
|---|---|
| Is this molecule synthesized industrially or in the laboratory? If yes, could you explain the process? | Acetone (CC(=O)C) is synthesized both industrially and in the laboratory. It is produced on a large scale industrially due to its wide range of applications as a solvent and chemical intermediate. The primary industrial method for acetone production is through the cumene process, which involves the following steps:<br>Industrial Synthesis of Acetone via Cumene Process:<br>Step 1: Propylene and Benzene Alkylation: Propylene and benzene are reacted in the presence of an acid catalyst (typically aluminum chloride, AlCl3) to produce cumene (isopropylbenzene) through alkylation.<br>CH3CH=CH2 + C6H6 → (CH3)2C6H5 (cumene)<br>Step 2: Cumene Oxidation: Cumene is then oxidized using air (oxygen) in the presence of a metal catalyst (usually cobalt or manganese acetate) to form cumene hydroperoxide.<br>(CH3)2C6H5 + O2 → (CH3)2C6H5OOH (cumene hydroperoxide)<br>Step 3: Cleavage of Cumene Hydroperoxide: Cumene hydroperoxide is further cleaved into phenol and acetone using an acid catalyst (sulfuric acid).<br>(CH3)2C6H5OOH → C6H5OH (phenol) + (CH3)2CO (acetone)<br>The acetone obtained from the cumene process is typically purified through various distillation steps to obtain a high-purity product suitable for various industrial applications.<br>Laboratory Synthesis of Acetone:<br>In the laboratory, acetone can be synthesized through various methods, including the hydration of propyne (also known as methylacetylene or propargylene) or the dehydrogenation of isopropanol (2-propanol). However, these methods are less common than the industrial cumene process due to cost and efficiency considerations.<br>For example, the hydration of propyne involves reacting propyne with water in the presence of a strong acid catalyst to form acetone:<br>HC≡C-CH3 + H2O → (CH3)2CO<br>The dehydrogenation of isopropanol can also produce acetone:<br>(CH3)2CHOH → (CH3)2CO + H2<br>In the laboratory, acetone is often synthesized as needed for specific experiments or applications rather than produced on an industrial scale. The choice of the synthesis method depends on factors such as availability, cost, and the desired purity of the acetone product. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| **Query prompt** | **LLM-retrieved text** |
|---|---|
| Are there any notable uses or applications for this molecule in medicine, industry, or other fields? | Acetone (CC(=O)C) has numerous notable uses and applications in various fields, including medicine, industry, and other areas. Some of the key applications are as follows:<br>Solvent: Acetone is a versatile and widely used solvent in various industries, including pharmaceuticals, paints, coatings, adhesives, and plastics. It is an effective solvent for a wide range of organic compounds and is often used for cleaning and degreasing purposes.<br>Chemical Intermediate: Acetone serves as an important chemical intermediate in the production of various compounds. It is a precursor for the synthesis of chemicals such as methyl methacrylate (used in the production of acrylic plastics), bisphenol-A (used in the production of polycarbonate plastics), and mesityl oxide (used as a solvent and in perfumes).<br>Cleaning Silicon Wafers: Acetone is employed in the electronics industry to clean and remove contaminants from silicon wafers used in semiconductor manufacturing.<br>Pharmaceuticals: In the pharmaceutical industry, acetone is used in the synthesis and formulation of various drugs and medications. It is employed to dissolve and stabilize active pharmaceutical ingredients and excipients.<br>Laboratory Reagent: Acetone is frequently used in laboratory settings as a reagent for various chemical reactions and as a solvent for cleaning glassware and equipment.<br>Cosmetic and Personal Care Products: Acetone is used in cosmetic and personal care products such as lotions, creams, and astringents, where it acts as a solvent and helps deliver active ingredients into the skin.<br>Extractive Distillation: In certain chemical processes, acetone is used as an extractive distillation agent to separate mixtures and purify various compounds.<br>Fuel Additive: Acetone has been studied as a potential fuel additive that may enhance the combustion efficiency of certain fuels, although its practical use in this regard is limited.<br>It is essential to handle acetone with care, as it is a flammable liquid and can pose health risks if not used or stored properly. Its versatility, as well as its physical and chemical properties, make acetone a valuable and widely employed compound in various industries and applications. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| **Query prompt** | **LLM-retrieved text** |
|---|---|
| Is this molecule found naturally? If yes, in what sources is it most commonly found? | Acetone (CC(=O)C) can be found naturally in trace amounts in certain sources. However, its natural occurrence is generally limited, and it is not a prominent component in the natural environment. Acetone is primarily produced through biological processes, and its concentration in natural sources is typically quite low. Some of the common natural sources of acetone include:<br>Human and Animal Metabolism: Small amounts of acetone are produced as a byproduct of metabolism in humans and animals. It is formed as a result of the breakdown of fatty acids, amino acids, and carbohydrates in the body.<br>Plants: Certain plants are capable of producing acetone as part of their metabolism. Acetone can be found in trace amounts in some plant tissues.<br>Volcanic Emissions: Acetone has been detected in volcanic emissions and geothermal areas, likely as a result of microbial activity in the volcanic environment.<br>Forest Fires: Acetone can be present in the smoke and emissions from forest fires due to the combustion of organic materials.<br>While acetone occurs naturally in these sources, its concentrations are generally very low, and it is typically overshadowed by other more abundant volatile organic compounds (VOCs). In most cases, the levels of naturally occurring acetone are not significant enough to have a major impact on the environment or human health. The primary sources of acetone in the environment are industrial processes, which release much higher quantities of acetone into the air and water. The industrial production and use of acetone for various applications contribute significantly to its presence in the environment. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| Query prompt | LLM-retrieved text |
|---|---|
| What safety measures should be taken when handling this molecule? | Handling acetone (CC(=O)C) requires some safety precautions due to its flammable and volatile nature. Here are important safety measures to consider when working with acetone:<br>Flammability: Acetone is highly flammable and can form explosive vapor-air mixtures. Keep acetone away from open flames, sparks, and sources of heat. Use it only in well-ventilated areas, and avoid using it near ignition sources.<br>Proper Storage: Store acetone in tightly sealed, non-reactive containers away from direct sunlight, heat, and ignition sources. Ensure that the storage area is well-ventilated and separate from incompatible chemicals.<br>Personal Protective Equipment (PPE): When handling acetone, wear appropriate personal protective equipment, including chemical-resistant gloves, safety goggles or a face shield, and a lab coat or protective clothing to prevent contact with skin and eyes.<br>Ventilation: Work with acetone in a well-ventilated area, such as a fume hood, to reduce the risk of inhalation exposure to vapors.<br>Avoid Skin Contact: Acetone can cause skin dryness and irritation. Minimize skin contact, and promptly wash off any spills on the skin with soap and water. If skin irritation occurs, seek medical attention.<br>Avoid Inhalation: Acetone vapors can be irritating to the respiratory tract. Use acetone in a well-ventilated area or wear a suitable respirator if exposure levels are high.<br>No Smoking Policy: Do not smoke while handling acetone, and enforce a strict no-smoking policy in areas where acetone is used or stored.<br>Fire Safety: Keep fire extinguishing equipment, such as fire extinguishers, nearby in case of accidental fires. Know the appropriate fire suppression methods for flammable liquid fires.<br>Spill Response: In the event of a spill, contain and absorb the acetone with suitable absorbent materials, and follow proper disposal procedures for hazardous waste.<br>Labeling and Storage Compatibility: Label containers containing acetone with appropriate hazard warnings and identification. Store acetone away from incompatible chemicals to prevent accidental reactions.<br>Disposal: Dispose of acetone waste in accordance with local, state, and federal regulations. Do not dispose of acetone down the drain or in regular trash.<br>It is important to review and follow the safety data sheet (SDS) provided by the manufacturer for specific handling instructions and safety precautions. When using acetone in an industrial or laboratory setting, ensure that personnel are trained in proper handling procedures and emergency response protocols. |

Table 16: LLM(GPT-3.5)-retrieved text for a natural language query for an organic molecule in SMILES notation "CC(=O)C".

| **Query prompt** | **LLM-retrieved text** |
|---|---|
| Are there any environmental impacts associated with the production, use, or disposal of this molecule? | There are potential environmental impacts associated with the production, use, and disposal of acetone (CC(=O)C). While acetone is a naturally occurring compound in small amounts, the significant quantities produced and used in various industries can contribute to environmental concerns. Some of the environmental impacts include:<br>Air Pollution: Acetone is a volatile organic compound (VOC) and is emitted into the atmosphere during its production, use as a solvent, and various industrial processes. VOCs can contribute to the formation of ground-level ozone and smog, which can have negative effects on air quality and human health.<br>Water Contamination: Improper disposal of acetone or industrial wastewater containing acetone can lead to its release into water bodies. Acetone is relatively soluble in water, and its presence in water sources can have adverse effects on aquatic ecosystems and wildlife.<br>Soil Contamination: Accidental spills or improper disposal of acetone-containing waste can lead to soil contamination. Acetone is capable of migrating through soil and potentially contaminating groundwater.<br>Non-Target Species: Acetone's volatility and widespread use may result in its presence in various environmental compartments. Although acetone is not considered highly toxic to humans, it can be harmful to certain aquatic organisms and other non-target species if present in elevated concentrations.<br>Ozone Depletion: While acetone itself is not an ozone-depleting substance, its industrial production and use may involve processes that generate other ozone-depleting substances or greenhouse gases. The release of these substances can contribute to stratospheric ozone depletion and climate change.<br>To mitigate these environmental impacts, it is essential to adopt proper waste management practices, control emissions during production and industrial processes, and promote the responsible use and handling of acetone. Implementing technologies for VOC capture and recovery can help reduce air emissions. Recycling and reusing acetone waste can also be beneficial in reducing its environmental impact. Additionally, regulatory measures and best management practices can be implemented to ensure the safe and sustainable use of acetone in various industries. |