Crossing New Frontiers: Knowledge-Augmented Large Language Model Prompting for Zero-Shot Text-Based De Novo Molecule Design

Sakhinana Sagar Srinivas[*], **Venkataramana Runkana**
TCS Research
sagar.sakhinana@tcs.com, venkat.runkana@tcs.com

Abstract

Molecule design is a multifaceted approach that leverages computational methods and experiments to optimize molecular properties, fast-tracking new drug discoveries, innovative material development, and more efficient chemical processes. Recently, text-based molecule design has emerged, inspired by next-generation AI tasks analogous to foundational vision-language models. Our study explores the use of knowledge-augmented prompting of large language models (LLMs) for the zero-shot text-conditional de novo molecular generation task. Our approach uses task-specific instructions and a few demonstrations to address distributional shift challenges when constructing augmented prompts for querying LLMs to generate molecules consistent with technical descriptions. Our framework proves effective, outperforming state-of-the-art (SOTA) baseline models on benchmark datasets.

1 Introduction

Molecule design is an interdisciplinary approach that involves identifying a target molecule or property to enhance, such as a drug with increased efficacy or a material with superior characteristics. Advancements in science and technology have accelerated the discovery and development of novel drugs, advanced materials, and innovative chemical processes. This iterative process begins with (a) identifying a target molecule or property to improve, followed by (b) employing computational methods to explore the vast chemical space and optimize potential candidate structure and composition. The cycle continues with (c) synthesizing and testing promising candidates in the laboratory until the desired characteristics are achieved. The transformer architecture[28] has revolutionized various fields in computer science, including language understanding[5], text generation[17, 2], image understanding[7], and multi-modal generation[19, 23]. Utilizing this architecture to scale language models has established itself as a universal approach for enhancing generalization performance. In recent times, the emergence of foundational Large Language Models (LLMs)[2, 3, 27], which are built upon transformer architectures, has significantly revolutionized performance in various natural language processing tasks by enabling enhanced linguistic comprehension and logical reasoning abilities. Different learning strategies such as Zero-Shot Chain of Thought (Zero-shot-CoT[29]) and Few-Shot (In-Context) Learning (Few-shot-ICL[22, 6]) are utilized to leverage the emerging abilities of general-purpose LLMs for a wide variety of specialized tasks across various domains. The former employs task-specific instructions without relying on downstream task-based demonstrations, utilizing the inherent knowledge that the language model acquired during training to generate outputs. In contrast, Few-shot-ICL supplements instructions with a handful of demonstrations, presented as input-output pairs, to foster contextual understanding and facilitate task-specific adaptation, thereby generating relevant output. Recently, there has been a surge in the evolution of generative AI, such as "DALL·E"[19, 20] from OpenAI — a text-to-image diffusion model that can generate realistic images from text descriptions, and "Make-A-Video"[24] from Meta AI — a text-to-video diffusion model that generates realistic, engaging, and creative videos from text, among others. Inspired by recent developments in next-generation AI, "Text-Based Molecule Design"[8] (also known as

[*]Designed and programmed research, conducted experiments, analyzed results, and drafted manuscript

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023

`text2mol`) represents a novel cross-domain task in chemistry that involves generating chemical SMILES representations from the corresponding technical descriptions of molecules expressed in natural language. Unlike traditional methods of de novo molecule generation, the `text2mol` task extracts information from technical descriptions of molecules, identifying aspects such as the specified structure, properties and functional groups, to generate chemical SMILES representations with desired characteristics. Existing models[8, 10] in the literature for the `text2mol` task face challenges in achieving optimal performance and utility, particularly in scenarios where data is scarce and unbalanced. LLMs like ChatGPT[2], while proficient in linguistic comprehension, are black-box in nature, resource-intensive, and lack interpretability. Smaller language models(LMs) like BERT[5], although flexible and interpretable, may lag in reasoning and generalization, resulting in less coherent and contextually relevant responses compared to LLMs. Navigating these challenges requires a delicate balance between performance, efficiency, and interpretability. Our study introduces a novel approach for the `text2mol` task by combining the strengths of both LLMs and small-scale LMs. LLMs predict a ranked list of chemical SMILES representations while providing explanations as justifications for these predictions, conditioned on the input prompt. These textual explanations, in conjunction with original technical descriptions of molecules, are used to fine-tune small-scale LMs to obtain context-aware token embeddings that capture the essence of both the generated explanations and original text, respectively. Concurrently, the top-ranked predictions generated by LLMs are transformed to obtain prediction embeddings. By integrating these various embeddings through a hierarchical multi-head attention mechanism, the framework inputs a unified cross-modal embedding into a transformer decoder to generate chemical SMILES representations that align with original technical descriptions. In this study, we explored the use of knowledge-augmented LLM prompting for zero-shot text-conditional molecule generation, a sequence-to-sequence cross-domain task. We present a powerful new tool, `FrontierX: LLM-MG`, where the goal is to task LLMs with a knowledge-infused prompt that consists of a few demonstrations(input-output pairs) for the text2mol task, along with task-specific instructions, where the output is chemical SMILES representations of the corresponding query technical descriptions. Our experiments on benchmark datasets provide empirical evidence supporting the framework's effectiveness in text-based molecule design tasks. The workflow of the proposed approach is illustrated in Figure 1. We discuss the proposed methodology in great detail, along with additional experimental results in the technical appendix.

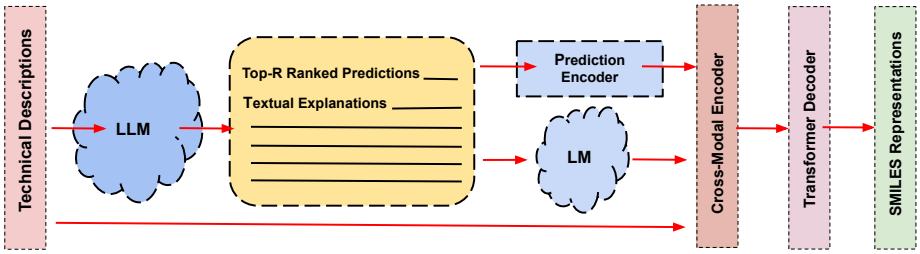

Figure 1: Overview of the `FrontierX: LLM-MG` framework. We construct knowledge-augmented prompts using task-specific instructions and a few demonstrations (input-output pairs) based on the downstream task. The augmented prompt queries LLMs to generate the top-R predictions of the SMILES representations and produces textual explanations as justifications for its predictions. We fine-tune small-scale pre-trained language models (LMs) on the generated explanations for domain-specific customization to obtain context-aware token embeddings. We utilize a weighted-sum pooling attention mechanism for task-specific adaptation to compute contextualized text-level embeddings. In parallel, we transform the LLMs' top-R predictions to compute prediction embeddings. The cross-modal encoder, modeled by a hierarchical multi-head attention mechanism, computes the unified embeddings by integrating the mono-domain text-level embeddings (both the original text and explanatory text) and prediction embeddings. Finally, the transformer decoder generates the chemical SMILES representations. We do not repurpose LLMs by fine-tuning with labeled data for domain customization. Instead, we access LLMs via LMaaS[25] using text-based API interaction.

2 Experiments & Results

2.1 Datasets & Baselines

Our study utilized the ChEBI-20 dataset[8], a bidirectional text-to-molecule translation dataset comprising 33,010 text description-molecule pairs with a predefined split ratio of 80:10:10 for training, validation, and test sets, respectively. We utilized 26,407 text description-molecule pairs from the training set for demonstrations (input-output mappings) for constructing a knowledge-augmented prompt to query LLMs. We used the MolT5 model[8], as a predominant baseline, which

is an encoder-decoder transformer architecture pretrained on a large unannotated dataset specifically for the `text2mol` translation task, building upon the foundations of the T5[18] model. We evaluated the performance of our proposed framework on the `text2mol` task, comparing it with several variants of the MolT5[8] and T5[18] models, as well as with general-purpose sequence-to-sequence models, such as the RNN-GRU and Vanilla Transformer models. In addition, various variants of few-shot (ICL) prompting of GPT-based models — reflecting the fact that this technique uses few-shot learning to prompt off-the-shelf GPT-based models to perform molecular property prediction for new, unseen molecules — referred to as baselines, are evaluated for comparison with our proposed framework. The configurations include different variants of the GPT-4 model, namely, (a) the zero-shot approach, (b) the scaffold sampling technique with K=10 or K=5, and (c) the random sampling technique with K=10 for constructing augmented prompts. In addition, we use GPT-3.5 and davinci-003 models, both employing the scaffold sampling technique with K=10 to construct knowledge-augmented prompts. For more details and information on the baselines, please refer to the earlier works[10, 8].

2.2 Evaluation Metrics

To comprehensively evaluate the quality and similarity of the generated chemical SMILES representations compared to the ground-truth SMILES representations, we employed a range of distinct evaluation metrics, categorized into three types. These metrics include (a) chemical similarity measures, such as the FTS (Fingerprint Tanimoto Similarity) [26] and the FCD (Fréchet ChemNet Distance) [16], as well as (b) natural language processing metrics like the BLEU (Bilingual Evaluation Understudy) score, Exact Match [8], and Levenshtein distance [14]. In addition, (c) we utilized the RDKit library [13] to validate the generated molecules. We delineate the metrics as follows: (a) We employ the FTS[26] metric to gauge the chemical similarity between the ground-truth and generated chemical compounds represented as SMILES strings(notation to represent chemical structures as text) by comparing their MACCS, RDK, and Morgan fingerprints[21, 13, 4]. (b) In addition, we utilize the FCD metric[16], which leverages latent information from a pretrained model[16] to predict molecular activity[8]. The FCD is calculated by measuring the distance between the mean embeddings of two sets of chemical SMILES strings (generated and ground-truth) in the latent space of the pretrained model. A lower FCD score indicates a greater similarity between the corresponding two sets of molecules. (c) We also apply natural language processing metrics to evaluate the quality of the chemical SMILES strings generated by our framework. These metrics encompass the following: (i) BLEU — this measures the similarity between two text strings, with a higher BLEU score denoting better similarity. (ii) Exact Match[8] — this quantifies the percentage of generated chemical SMILES strings that are identical to the ground-truth strings. (iii) Levenshtein distance[14] — this calculates the minimum number of single-character edits required to modify the generated chemical SMILES strings to match the ground-truth strings, with a lower value indicating closer similarity. By utilizing these diverse metrics, we attain a nuanced understanding of the efficacy of our text-conditional de novo molecule generation framework. Higher FTS scores and lower FCD scores signify better chemical similarity and closer molecular activity resemblance, respectively. In the context of matching chemical SMILES strings from the perspective of natural language processing, higher BLEU and Exact Match scores are preferred to achieve better alignment with the ground-truth SMILES strings, while a lower Levenshtein distance indicates fewer required edits, denoting superior similarity. The RDKit library assists in verifying the validity of the generated molecules, with a higher proportion indicating successful generation[13].

2.3 Experimental Setup

We used the ChEBI-20 dataset[9] with an 80:10:10 split: 80% for training, 10% for validation, and 10% for testing. The training set was utilized to update the learnable parameters, the validation set to select optimal hyperparameters, and the test set to evaluate generalization performance of the proposed framework. Our scalable and efficient framework offers a unified solution for integrating LLMs and LMs. We configured the hyperparameters of our framework with a batch size of 32, trained it for 100 epochs, and a hidden or embedding dimension(d) of 128. Additional hyperparameters include the number of attention heads (H) set to 4, and the dimensionality of Key/Query/Value (d_h) is 32. To optimize the training process, we utilized the Adam optimizer [12], initially setting the learning rate to $1e^{-3}$. Additionally, we incorporated a learning rate decay scheduler, which reduced the learning rate by half whenever the validation loss did not improve for 10 consecutive epochs. Furthermore, we applied early stopping to prevent overfitting on the validation data. We evaluated our approach using the following LLMs: GPT-4.0, GPT-3.5-turbo, GPT-3.0-text-davinci-003, and Google Bard. In our approach, we chose not to fine-tune hyperparameters individually for each LLM, opting instead to maintain consistent settings across all language models. This strategy simplifies experimentation, ensures uniform conditions, facilitates result comparison, and promotes consistency.

Moreover, it underscores the versatility of our framework, which can be used with any off-the-shelf LLM without the need for computationally expensive hyperparameter tuning. We used the Scaffold technique with K=16 to sample demonstrations (input-output mappings) from the training data to construct augmented prompts for querying LLMs in few-shot settings. In addition, we query LLMs to generate the top-R ranked chemical SMILES strings predictions list and set the hyperparameter R as 4. To maximize computational resource utilization, we harnessed eight V100 GPUs, each equipped with 8 GB of GPU memory, for training deep learning models built upon the PyTorch framework. Considering the context length limitations imposed by LLMs, which restrict the maximum sequence length that a typical LLM can process at a time to 4096 tokens, we implemented strategies to mitigate the high computational costs associated with prompting LLMs. This approach included running each experiment twice and reporting the average results. Our approach prioritizes both resource optimization and accuracy, aiming to achieve the best possible outcomes while minimizing the computational footprint. Our evaluation incorporated several metrics, and we present the results for the test datasets and compare the performance against well-known baselines.

Table 1: The table presents a performance comparison of the proposed framework and the baselines on the `text2mol` task. The top-performing model is highlighted in bold. The baseline results are reported from previous work[10]. We leveraged the Scaffold technique, setting K to 16, to sample demonstrations and construct augmented prompts for in-context learning in all experiments involving `FrontierX: LLM-MG` with various off-the-shelf LLMs.

Method	**BLEU (↑)**	**Exact (↑)**	**Levenshtein (↓)**	**Validity (↑)**	**MACCS FTS (↑)**	**RDK FTS (↑)**	**Morgan FTS (↑)**	**FCD (↓)**
Ground-Truth	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0
GPT-4 (zero-shot)	0.490±0.017	0.046±0.009	47.418±1.668	0.758±0.015	0.733±0.020	0.514±0.021	0.432±0.014	11.913±0.97
GPT-4 (Scaffold, k=10)	0.816±0.004	0.174±0.029	21.160±0.600	0.888±0.023	0.867±0.005	0.738±0.010	0.672±0.013	6.224±0.44
GPT-4 (Scaffold, k=5)	0.815±0.011	0.164±0.018	21.862±1.768	0.874±0.030	0.865±0.015	0.741±0.023	0.670±0.028	5.843±0.515
GPT-4 (Random, k=10)	0.602±0.016	0.060±0.007	42.390±1.008	0.770±0.030	0.762±0.013	0.548±0.017	0.475±0.015	10.594±0.41
GPT-3.5 (Scaffold, k=10)	0.479±0.156	0.094±0.011	82.008±40.354	0.854±0.059	0.833±0.006	0.686±0.016	0.585±0.013	8.341±0.607
Davinci-003 (Scaffold, k=10)	0.741±0.011	0.100±0.010	25.648±2.186	0.936±0.009	0.783±0.014	0.648±0.004	0.560±0.010	8.335±0.310
FrontierX-W/GPT-4	**0.937±0.023**	**0.641 ±0.089**	**6.946 ±0.045**	**0.975 ±0.013**	**0.947 ±0.029**	**0.838±0.006**	**0.845±0.012**	**0.796±0.003**
FrontierX-W/GPT-3.5-turbo	0.893±0.051	0.567±0.112	11.431 ±0.086	0.939 ±0.027	0.914 ±0.015	0.785±0.012	0.809±0.016	0.968±0.004
FrontierX-W/GPT-3.0	0.885±0.084	0.536±0.173	13.867 ±0.137	0.914 ±0.041	0.896 ±0.034	0.774±0.028	0.823±0.034	1.025±0.007
FrontierX-W/Google Bard	0.749±0.109	0.426±0.138	16.729 ±0.145	0.855 ±0.017	0.863 ±0.018	0.692±0.037	0.727±0.045	2.345±0.012

Table 2: The table illustrates the performance of both the proposed framework and the baseline models in the `text2mol` task. We have highlighted the top-performing model using bold text. Baseline results are reported from earlier research[8]. For all `FrontierX: LLM-MG` experiments with various off-the-shelf LLMs, we employed the Scaffold technique, setting K to 16 to sample demonstrations for constructing augmented prompts for in-context learning.

Method	**BLEU (↑)**	**Exact (↑)**	**Levenshtein (↓)**	**MACCS FTS (↑)**	**RDK FTS (↑)**	**Morgan FTS (↑)**	**FCD (↓)**	**Validity (↑)**
Ground-Truth	1.0	1.0	0.0	1.0	1.0	1.0	0.0	1.0
RNN-GRU	0.652	0.005	38.09	0.591	0.400	0.362	4.55	0.542
Transformer	0.499	0.000	57.66	0.480	0.320	0.217	11.32	0.906
T5-Small	0.741	0.064	27.703	0.704	0.578	0.525	2.89	0.608
MolT5-Small	0.755	0.079	25.988	0.703	0.568	0.517	2.49	0.721
T5-Base	0.762	0.069	24.950	0.731	0.605	0.545	2.48	0.660
MolT5-Base	0.769	0.081	24.458	0.721	0.588	0.529	2.18	0.772
T5-Large	0.854	0.279	16.721	0.823	0.731	0.670	1.22	0.902
MolT5-Large	0.854	0.311	16.071	0.834	0.746	0.684	1.20	0.905
FrontierX-W/GPT-4	**0.937±0.023**	**0.641 ±0.089**	**6.946 ±0.045**	**0.947 ±0.029**	**0.838±0.006**	**0.845±0.012**	**0.796±0.003**	**0.975 ±0.013**

2.4 Results

The experimental results of the proposed framework and the baseline models performance on the `text2mol` task are presented in Tables 1 and 2. The results of the baseline models are reported from earlier studies [10, 8]. The results undeniably demonstrate the superior performance of the **FrontierX: LLM-MG** framework, especially when combined with the GPT-4 backbone and employing the Scaffold technique with K set to 16. This optimal combination excels in generating accurate molecular structures that closely resemble the ground truth, surpassing all baseline models across various evaluation metrics.

Ablation Studies: Our proposed framework operates through a series of interconnected stages via a progressively structured multi-step pipeline. Beginning with step (a), we create knowledge-augmented prompts using task-specific instructions and demonstrations, prompting large language models (LLMs) to (i) generate top-ranked (top-R) SMILES strings predictions along with (ii) explanatory justifications for their predictions. In step (b), these generated explanations are used to (i) fine-tune a smaller pre-trained language model (LM_{exp}) for domain customization to obtain contextualized token embeddings and utilize (ii) a weighted sum-pooling attention mechanism to compute text-level embeddings denoted as y_{exp} from the token embeddings for task-specific adaptation. Moving to step (c), the top-R predictions from the LLMs are transformed to compute prediction embeddings y_{pred}. Concurrently, in step (d), we fine-tune another small-scale language model (LM_{org}) on the original textual descriptions of molecules to compute context-aware token embeddings, and then compute the original text-level embeddings y_{org} through a weighted attention mechanism. In step (e), our proposed framework obtains a cross-modal embeddings, y_{cross}, through a hierarchical multi-head attention mechanism that integrates the original text-level embeddings y_{org}, explanatory text-level embeddings y_{exp}, and prediction embeddings y_{pred}. We conduct empirical research to understand the significance and contribution of each distinct method within the proposed framework, evaluating its learned embeddings to achieve optimal results. We perform ablation studies to assess the impact of disabling individual methods on the overall performance of our framework. To determine the contribution of each method to the framework's performance, we create various ablated variants by disabling individual methods and evaluate them using benchmark datasets for `text2mol` tasks. We choose the proposed `FrontierX: LLM-MG` framework as the reference baseline for the ablation studies. Our robust strategy not only validates the efficacy of the diverse methods but also substantiates the rationale, providing a strong basis for their design choices and justifying their inclusion within the framework. The ablated variants without the explanatory text-level embeddings, prediction embeddings, and original text-level embeddings are referred to as "w/o y_{exp}", "w/o y_{pred}", and "w/o y_{org}", respectively. The ablation study findings are summarized in Table 3. All the ablation study experiments were conducted with the **FrontierX: LLM-MG** framework using the GPT-4 backbone and Scaffold sampling technique with $K = 16$, by disabling methods as discussed earlier.

Table 3: The table shows the experimental findings on the ablation study. The experiments are conducted using the Scaffold sampling technique with K=16 and GPT-4 backbone.

Method	BLEU (↑)	Exact (↑)	Levenshtein (↓)	Validity (↑)	MACCS FTS (↑)	RDK FTS (↑)	Morgan FTS (↑)	FCD (↓)
Ground-Truth	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0
FrontierX	**0.937±0.023**	**0.641 ±0.089**	**6.946 ±0.045**	**0.975 ±0.013**	**0.947 ±0.029**	**0.838±0.006**	**0.845±0.012**	**0.796±0.003**
FrontierX — w/o y_{exp}	0.783±0.032	0.468 ±0.054	23.526 ±0.013	0.847 ±0.025	0.784 ±0.061	0.682±0.013	0.723±0.037	3.712±0.014
FrontierX — w/o y_{org}	0.741±0.032	0.455 ±0.089	27.817 ±0.067	0.819 ±0.034	0.751 ±0.037	0.646±0.024	0.695±0.016	4.685±0.024
FrontierX — w/o y_{pred}	0.828±0.044	0.543 ±0.063	13.946 ±0.045	0.911 ±0.048	0.867 ±0.058	0.749±0.019	0.763±0.027	2.736±0.033

On the ChEBI-20 dataset[9], we observe differing impacts on framework performance when certain methods are omitted. The "w/o y_{exp}" variant shows a substantial decline in performance relative to the baseline, as evidenced by a significant drop of 17.21% in MACCS FTS, 16.43% in BLEU, and 13.12% in Validity. Similarly, the "w/o y_{org}" variant performs much worse than the baseline, with a remarkable drop of 20.69% in MACCS FTS, 20.91% in BLEU, and 16.00% in Validity. In contrast, the "w/o y_{pred}" variant exhibits a marginally inferior performance compared to the baseline, with a modest drop of 8.44% in MACCS FTS, 11.63% in BLEU, and 6.56% in Validity. The significant drop in performance metrics for the ablated variants, when compared to the baseline, highlights the considerable impact of the mechanisms inherent in the methods omitted from the baseline and leads to degraded performance. Our experiments corroborate our hypothesis of joint optimization

to obtain a cross-modal embeddings, y_{cross}, through a hierarchical multi-head attention mechanism that integrates the original text-level embeddings y_{org}, explanatory text-level embeddings y_{exp}, and prediction embeddings y_{pred}, achieving state-of-the-art (SOTA) performance on the `text2mol` task.

Impact of Hierarchical Multi-Head Attention(HMHA) Mechanism: In our work, we compute the cross-modal embedding, denoted as y_{cross}, through a hierarchical multi-head attention (HMHA) mechanism that integrates the original text-level embedding (y_{org}), explanatory text-level embedding (y_{exp}), and prediction embedding (y_{pred}). To determine the impact of the HMHA mechanism on the framework performance, we conducted ablation study. We refer to the ablated variant without the HMHA mechanism as "w/o HMHA". We substitute the HMHA mechanism with dual-stage linear operators in the ablated variant to compute cross-modal embeddings. The findings of the ablation study are summarized in Table 4. We conducted the experiment using the `FrontierX: LLM-MG` framework with GPT-4 backbone, where we replaced the HMHA mechanism with linear operators, as discussed earlier. The experimental results support the inclusion of the hierarchical multi-head attention mechanism (HMHA) to generate cross-modal embeddings, aiding in the generation of more valid chemical SMILES representations in the `text2mol` task.

Table 4: The table shows the experimental findings of the study on the impact of the HMHA mechanism on the `text2mol` task. The experiments were conducted using the `FrontierX: LLM-MG` framework with a GPT-4 backbone. We utilized the Scaffold sampling technique with $K = 16$ for constructing augmented prompts.

Method	BLEU (↑)	Exact (↑)	Levenshtein (↓)	Validity (↑)	MACCS FTS (↑)	RDK FTS (↑)	Morgan FTS (↑)	FCD (↓)
Ground-Truth	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0
FrontierX - (W/GPT-4)	**0.937±0.023**	**0.641 ±0.089**	**6.946 ±0.045**	**0.975 ±0.013**	**0.947 ±0.029**	**0.838±0.006**	**0.845±0.012**	**0.796±0.003**
FrontierX - (W/GPT-4) - w/o **HMHA**	0.773±0.033	0.516 ±0.069	22.657 ±0.082	0.787 ±0.037	0.756 ±0.074	0.691±0.017	0.694±0.027	5.598±0.065

Hyperparameter Tuning: To enhance the performance of our `FrontierX: LLM-MG` framework, we embarked on meticulous hyperparameter tuning through detailed experimentation and analysis. We chose random search as a strategy to adeptly navigate the hyperparameter space, pinpointing the optimal framework configuration on the benchmark dataset, in lieu of more computationally intensive approaches such as grid search or Bayesian optimization. This strategy enabled us to obtain optimal results on the validation subset of the benchmark dataset, as evidenced by several evaluation metrics. We conducted hyperparameter optimization on the `FrontierX: LLM-MG-W/GPT-4` variant of our framework. We utilized the Scaffold sampling technique with K = 16 for constructing augmented prompts.The primary key hyperparameters within this framework include batch size ($b \in \{32, 48, 64\}$) and the embedding dimension ($d \in \{64, 128, 196, 256\}$). Table 5 presents the results of hyperparameter tuning on the representative benchmark dataset. We report the results for the near-optimal combinations of the hyperparameters.

Table 5: The table shows the experimental results of the hyperparameter tuning experiments.

Method	BLEU (↑)	Exact (↑)	Levenshtein (↓)	Validity (↑)	MACCS FTS (↑)	RDK FTS (↑)	Morgan FTS (↑)	FCD (↓)
Ground-Truth	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0
FrontierX: LLM-MG - (b=32, d=128)	**0.937±0.023**	**0.641 ±0.089**	**6.946 ±0.045**	**0.975 ±0.013**	**0.947 ±0.029**	**0.838±0.006**	**0.845±0.012**	**0.796±0.003**
FrontierX: LLM-MG - (b=48, d=196)	0.887±0.017	0.604 ±0.094	8.581 ±0.084	0.942 ±0.019	0.895 ±0.094	0.797±0.012	0.817±0.018	0.827±0.015
FrontierX: LLM-MG - (b=48, d=128)	0.893±0.023	0.627 ±0.081	8.024 ±0.053	0.955 ±0.022	0.915 ±0.044	0.807±0.067	0.829±0.025	0.814±0.033
FrontierX: LLM-MG - (b=64, d=256)	0.903±0.011	0.633 ±0.072	7.813 ±0.047	0.915 ±0.032	0.929 ±0.036	0.817±0.069	0.833±0.037	0.821±0.037

3 Conclusion

In this study, we pioneered the `text2mol` approach, inaugurating a transformative paradigm where chemistry meets language models, expediting scientific advancements. Through the creation of `FrontierX: LLM-MG`, we demonstrated the efficacy of using large language models for seamless and efficient translation between textual descriptions and chemical SMILES representations. Acknowledging the limitations of current methods, our research highlights a promising horizon in molecule design, potentially ushering in an era of accelerated innovation and interdisciplinary collaboration. Our study illustrates the transformative impact of integrating molecular design with language models, offering an innovative approach to molecule generation that can catalyze groundbreaking developments in science and technology.

4 Technical Appendix

4.1 Study of Knowledge-Augmented Prompting

In our study, we employ knowledge-augmented prompting with LLMs for zero-shot text-to-molecule(`text2mol`) translation task by leveraging the pre-existing knowledge embedded within the language model parameters. LLMs are capable of generating chemical SMILES representations from textual descriptions through entity recognition, grammar understanding, symbol mapping, and structure validation, which marks significant progress in molecule generation via language models. This knowledge-augmentation prompting technique allows LLMs to adapt to new, unseen molecule textual descriptions using a few task-specific demonstrations, thereby eliminating the need for fine-tuning with labeled data for task-specific adaptation. The approach involves creating knowledge-augmented prompts that combine task-specific instructions with demonstrations (input-output pairs) sampled from training data relevant to the target molecule textual descriptions determined using off-the-shelf semantic similarity techniques. In this context, each pair consists of a textual description of a molecule (input) and its corresponding SMILES representation (output), where the task-specific instruction is to convert the target molecule textual descriptions into the standardized chemical SMILES notation. This approach aligns the LLM's capabilities with the `text2mol` task by crafting knowledge-augmented prompts that blend specific instructions with relevant demonstrations, selected based on semantic similarity. This strategic alignment facilitates accurate chemical SMILES strings generation without necessitating language model parameter updates. We have employed two sampling strategies — random and scaffold — to evaluate the impact of both the quality and quantity of demonstrations in the knowledge-augmented prompt, which is utilized for querying LLMs during text-based de novo molecule generation. The scaffold strategy utilizes a semantic similarity method to sample the top-K relevant text-molecule pairs, using OpenAI's `text-embedding-ada-002 technique`[2]. The random technique involves the arbitrary selection of K text-molecule pairs without any prior knowledge in a non-deterministic manner. The study compares the effectiveness of both strategies in enhancing the language-conditioned molecule generation task using off-the-shelf pre-trained LLMs, including Google Bard and other GPT model family variants. We conducted experiments to compare and contrast the performance of the "Random" and "Scaffold" sampling strategies, and to identify the optimal number of demonstrations.

Table 6: The table shows the results of the experimental study examining the impact of both quantity and quality of demonstrations on knowledge-augmented prompting strategies in the `text2mol` task.

Method	BLEU (↑)	Exact (↑)	Levenshtein (↓)	Validity (↑)	MACCS FTS (↑)	RDK FTS (↑)	Morgan FTS (↑)	FCD (↓)
Ground-Truth	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0
FrontierX: LLM-MG - (Baseline - Scaffold, k=16, W/GPT-4)	**0.937±0.023**	**0.641 ±0.089**	**6.946 ±0.045**	**0.975 ±0.013**	**0.947 ±0.029**	**0.838±0.006**	**0.845±0.012**	**0.796±0.003**
FrontierX: LLM-MG - (Scaffold, k=12, W/GPT-3.5-turbo)	0.837±0.067	0.527±0.121	14.117 ±0.072	0.885 ±0.067	0.868 ±0.052	0.773±0.026	0.763±0.037	2.099±0.045
FrontierX: LLM-MG - (Scaffold, k=12, W/GPT-3.0-text-davinci-003)	0.843±0.032	0.487±0.135	16.789 ±0.096	0.867 ±0.059	0.856 ±0.102	0.714±0.042	0.776±0.064	3.242±0.014
FrontierX: LLM-MG - (Scaffold, k=12, W/Google Bard)	0.703±0.128	0.466±0.104	19.211 ±0.145	0.803 ±0.074	0.819 ±0.082	0.614±0.075	0.687±0.067	3.105±0.054
FrontierX: LLM-MG - (random, k=12, W/GPT-3.5-turbo)	0.735±0.033	0.417±0.102	21.324 ±0.089	0.741 ±0.097	0.767 ±0.085	0.692±0.063	0.667±0.059	8.125±0.075
FrontierX: LLM-MG - (random, k=12, W/GPT-3.0-text-davinci-003)	0.716±0.045	0.327±0.087	23.196 ±0.054	0.753 ±0.109	0.766 ±0.113	0.609±0.012	0.684±0.019	9.309±0.014
FrontierX: LLM-MG - (random, k=12, W/Google Bard)	0.623±0.113	0.403±0.064	15.103 ±0.165	0.717 ±0.126	0.689 ±0.037	0.504±0.085	0.597±0.073	15.135±0.057
FrontierX: LLM-MG - (Scaffold, k=4, W/GPT-3.5-turbo)	0.605±0.077	0.346±0.061	26.106 ±0.081	0.651 ±0.079	0.697 ±0.035	0.612±0.085	0.577±0.031	17.321±0.056
FrontierX: LLM-MG - (Scaffold, k=4, W/GPT-3.0-text-davinci-003)	0.623±0.042	0.259±0.038	26.096 ±0.037	0.673 ±0.068	0.686 ±0.093	0.517±0.027	0.603±0.061	19.219±0.038
FrontierX: LLM-MG - (Scaffold, k=4, W/Google Bard)	0.514±0.049	0.317±0.047	22.033 ±0.187	0.636 ±0.137	0.587 ±0.056	0.446±0.032	0.508±0.049	24.101±0.095

Results: Table 6 presents the results of the experimental study that examined the effects of both the quantity and quality of demonstrations on the performance of knowledge-augmented prompting strategies in the `text2mol` task. Our study compared the performance of various GPT models with that of Google Bard on the ChEBI-20 dataset[9]. The results indicated that the GPT models consistently outperformed Google Bard across all evaluation metrics when provided with the same number of task-specific demonstrations in the augmented prompt. Notably, GPT-4 demonstrated the highest performance among the tested models, generating a greater number of valid chemical SMILES

[2]https://platform.openai.com/docs/guides/embeddings

representations. Furthermore, the study indicates that enhancing the knowledge-augmented prompt with more task-specific demonstrations directly improves the predictive accuracy of the language models. This highlights a positive correlation between the number of task-specific demonstrations and the performance of LLMs on the `text2mol` task. The study found that scaffold sampling consistently outperforms random sampling on the `text2mol` task when using any off-the-shelf LLMs. One possible reason for this superior performance is the strong textual similarities between the text-molecule pairs sampled using the scaffold technique and the target molecule descriptions. Therefore, using scaffold sampling instead of random sampling may lead GPT models to generate more accurate chemical SMILES representations. LLMs continue to face challenges in precisely interpreting molecular representations in chemical SMILES notations, resulting in poor performance on `text2mol` tasks. SMILES representations can possess multiple valid forms and implicit hydrogen atoms, causing ambiguity and presenting difficulties for LLMs. Improved LLMs capable of handling molecular structures and seamlessly integrating with tools like RDKit are necessary

4.2 Proposed Method

The Large Language Models (LLMs), such as ChatGPT[2], Meta's LLaMA[27] — that have been pre-trained on large text corpora and operate based on a "prompt and predict" approach (utilizing natural language prompts to generate the subsequent contextual word or phrase, aligning with human-like responses) — have revolutionized language modeling with their proficiency in linguistic comprehension and advanced logical reasoning abilities, providing improved performance on general-purpose NLP tasks. While LLMs are inherently black box in nature, they possess remarkable capabilities. However, their widespread adoption for applications in various downstream tasks is hindered by the unavailability of logits or token embeddings, which limits explainability. Additionally, they require significant computational resources for fine-tuning on labeled data for task-specific adaptation or for repurposing for domain-customization. In contrast, the small-scale language models (LMs), such as BERT[5] and DeBERTa[11], following a "pre-train, fine-tune" approach, offer more affordable flexibility for fine-tuning with minimal labeled data and provide access to logits or token embeddings, aiding interpretability. While smaller LMs can learn complex patterns, they often fall short in reasoning and generalization abilities compared to LLMs, which generate more coherent and contextually relevant responses. To alleviate resource constraints, Language Modeling as a Service (LMaaS[25]) offers access to LLMs through text-based API interactions, while remaining scalable and cost-effective. However, the potential of LLMs for text-conditional de novo molecular generation tasks remains largely underexplored. Our proposed approach for the `text2mol` task leverages LLMs by utilizing: (a) their predictive ability to provide a top-R ranked list of chemical SMILES representations; and (b) their generative ability to offer auxiliary explanations as justifications for their predictions by conditioning on the augmented prompt. Furthermore, we fine-tune two different small-scale LMs using (a) generated explanations from LLMs and (b) input technical descriptions of molecules to compute their respective contextualized token embeddings — which capture semantic coherence and contextual relevance for text-to-molecule generation tasks. We utilize weighted attention mechanism to compute both original and explanatory text-level embeddings from their respective context-aware token embeddings. In addition, we transform the LLMs' top-R predictions of chemical SMILES representations into predictive embeddings. We use a hierarchical multi-head attention mechanism to integrate various embeddings into unified cross-modal embedding for input into a transformer decoder, generating the chemical SMILES representation.

Table 7: Technical details of LLMs and LMs. *Enterprise* refers to the organization that developed the language models. *Cost* denotes the expenses associated with using 1K tokens. *Last Update Date* indicates that the LLM's knowledge base is limited to information available up to that specific date.

Model	Enterprise	Cost	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

Evaluation LLMs & LMs: In this work, we evaluated three popular LLMs: text-davinci-003[3], ChatGPT[4], and Google BARD[5], in order to thoroughly compare their distinct strengths. text-davinci-003 was the earliest LLM released by OpenAI and was tailored for a broad spectrum of linguistic tasks. GPT-3.5-turbo is a substantial improvement over the GPT-3 base models, demonstrating remarkable

[3]https://platform.openai.com/docs/models/gpt-3-5

[4]https://chat.openai.com/chat

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

performance on a wide range of linguistic tasks while also being cost-effective. Google BARD[1] stands out due to its extraordinary scale, complexity, and an impressively extensive vocabulary compared to the GPT-3.5 models. In addition to these, our study also incorporates a pre-trained smaller LM, DeBERTa[6], which is an improved version of the BERT[5] architecture. Table 7 presents a comprehensive summary of the technical specifications of these language models.

Knowledge-Augmented Prompts: In our work, we offer essential context and task-specific instructions by using input natural language descriptions of the target molecule to prompt LLMs in a zero-shot setting to generate corresponding chemical SMILES representations. In this scenario, the primary task-specific instructions involve the translation of these descriptions into chemical SMILES representations. We create an augmented prompt that incorporates both the task-specific instructions and a few demonstrations. These demonstrations, which establish the context, are grounded in the downstream `text2mol` task and comprise input-output pairs (i.e., technical descriptions and their corresponding chemical SMILES representations). This approach facilitates knowledge-augmented prompting of the LLMs for zero-shot text-to-molecule generation tasks. The construction of an augmented prompt involves sampling text-molecule pairs from the training data that are relevant to the target molecule descriptions. We then prepend these pairs to the task-specific instructions to form an augmented prompt, which is used to query the LLMs in a zero-shot setting for the generation of chemical SMILES representations. To evaluate the impact of the quality and quantity of sampled text-molecule pairs on the performance of text-conditional de novo molecule generation tasks, we employ two different sampling strategies. The quality of these pairs is determined by the sampling methods used to identify pairs similar to the target molecule descriptions. We navigate through the training dataset using two semantic search-retrieval methodologies — `random` and `scaffold` — to sample text-molecule pairs relevant to the target molecule descriptions. The random approach involves arbitrarily sampling K text-molecule pairs from the training dataset. In contrast, the scaffold technique employs semantic similarity methods, specifically `text-embedding-ada-002` from OpenAI[7], to evaluate the similarity between the molecular textual descriptions in the training dataset and the target molecule descriptions. It then selects the top-K most relevant text-molecule pairs, where the hyperparameter K is set using a random search technique. We employ the different sampling strategies to analyze the effectiveness of augmenting prompts with relevant text-molecule pairs in language-conditioned molecule generation tasks. In short, unlike traditional supervised learning, LLMs (a) predict the chemical SMILES representations and (b) generate textual explanations for their predictions, utilizing the inherent knowledge embedded within the language model's parameters, all conditioned on the augmented prompt, without needing any parameter updates.

Querying LLMs: We access LLMs with LMaaS[25] platforms via text-based API interaction, necessitating solely text-based input and output. We create a customized zero-shot prompt template to query LLMs to translate textual descriptions into chemical SMILES representations. The LLMs' response serves the dual purpose of (a) providing detailed textual explanations for the underlying rationale, (reasoning or logic), behind the predictions and (b) generating a list of the top-R ranked chemical SMILES representations. Subsequently, we fine-tune smaller downstream LMs using the generated auxiliary explanations. The custom augmented prompt format is as follows:

> Below are the textual descriptions – chemical SMILES representation pairs. Generate the chemical SMILES representation for the textual description provided below.

Querying LLMs (a) predicts the top-R ranked chemical SMILES representations and (b) provides auxiliary explanations as logical justifications for its predictions.

> (**LLMs Response**) [top-R ranked predictions — Auxiliary Explanations]

In the next section, we will discuss the use of auxiliary explanations and original textual descriptions for fine-tuning various downstream smaller LMs for domain customization. Later, we will transform the LLMs top-R predictions of chemical SMILES representations into predictive embeddings.

Fine-tuning LMs for Domain-Specific Customization: Our novel approach leverages the integration of a smaller language model (LM) to extract relevant information from the original molecular textual descriptions and auxiliary explanations generated by LLMs, thereby aiding downstream tasks. The intermediary LM serves as a bridge between the LLM and the downstream layers that generate chemical SMILES representations. To elucidate further, we fine-tune pre-trained LMs, denoted as LM_{exp} and LM_{org}, to compute context-aware token embeddings by passing the text sequences

[6]For more information on DeBERTa, please refer to `https://huggingface.co/docs/transformers/index` [11]

[7]https://platform.openai.com/docs/guides/embeddings

generated by LLMs (referred to as $\mathcal{S}_{\text{exp}}$) and original textual descriptions (referred to as $\mathcal{S}_{\text{org}}$) through the LM_{exp} and LM_{org} models, respectively, as described below:

$$h_{\text{exp}} = \text{LM}_{\text{exp}}(\mathcal{S}_{\text{exp}}) \in \mathbb{R}^{(m \times d)}; \quad h_{\text{org}} = \text{LM}_{\text{org}}(\mathcal{S}_{\text{org}}) \in \mathbb{R}^{(n \times d)} \tag{1}$$

where both contextualized embeddings h_{exp} and h_{org} capture not only the contextual information of the tokens but also encapsulate the semantic relationships among tokens within their respective textual content. Here, m and n represent the number of tokens in $\mathcal{S}_{\text{exp}}$ and $\mathcal{S}_{\text{org}}$, while d represents the token embedding dimension. We employ a softmax attention mechanism to compute a weighted sum of the contextualized token embeddings, encoding the auxiliary explanations and original textual descriptions into single fixed-length vectors or embeddings denoted as y_{exp} and y_{org} and computed as follows,

$$\alpha_i = \text{softmax}(q_i); \quad q_i = \mathbf{u}^T h_{\text{exp}}^{(i)} \;\; || \;\; \beta_i = \text{softmax}(r_i); \quad r_i = \mathbf{v}^T h_{\text{org}}^{(i)} \tag{2}$$

$$y_{\text{exp}} = \sum_{i=0}^{\text{m}} \alpha_i h_{\text{exp}}^{(i)} \in \mathbb{R}^{(d)}; \quad y_{\text{org}} = \sum_{i=0}^{\text{n}} \beta_i h_{\text{org}}^{(i)} \in \mathbb{R}^{(d)} \tag{3}$$

where $\mathbf{u}$ and $\mathbf{v}$ are differentiable vectors. The explanatory text-level embedding, represented as y_{exp}, encapsulates domain-specific knowledge retrieved from foundational LLMs to support its predictions. The original text-level embedding, denoted as y_{org}, captures the overall context and semantics within the original textual descriptions by extracting the most pertinent and task-relevant information.

LLMs Prediction Embeddings: As mentioned earlier, the LLMs not only provide the auxiliary textual explanations but also predict the top-R ranked chemical SMILES representations list, which can be informative. For each target molecule in the `text2mol` task, the top-R predictions are converted into one-hot encoded vectors $p_{i,1}, \ldots, p_{i,R} \in \mathbb{R}^C$, where C represents the total number of elements in the SMILES vocabulary, encompassing a wide range of characters and symbols used to represent chemical structures. These vectors are subsequently concatenated into a single RC-dimensional vector, and finally, they undergo linear encoding into a fixed-length prediction embedding $y_{\text{pred}} \in \mathbb{R}^d$, encapsulating the top-$R$ predictions from the LLMs.

Cross-modal Attention Layer: We compute the cross-modal embedding, denoted as y_{cross}, using a hierarchical multi-head attention mechanism that integrates the original text-level embedding y_{org}, the explanatory text-level embedding y_{exp}, and the prediction embedding y_{pred}. This mechanism provides a robust framework for integrating diverse information encapsulated from different modalities, addressing several key aspects critical to the performance of cross-modal learning tasks. It involves hierarchical implementation of multi-head attention mechanisms. We employ two layers, each focusing on different aspects of the input embeddings, enabling more complex interactions and potentially leading to more scalable and efficient models. In the initial layer, we apply a multi-head attention mechanism to the mono-domain embeddings, specifically the original text-level embedding y_{org} and the explanatory text-level embedding y_{exp}, to obtain unified mono-domain embeddings denoted as y_{uni}. In the subsequent layer, we utilize the multi-head attention mechanism on the cross-domain embeddings, comprising the prediction embedding y_{pred} and the unified mono-domain embeddings y_{uni}, to compute the cross-modal embeddings denoted as y_{cross}. For the first layer, we compute the Query, Key, Value projections for the original text-level embedding y_{org} for each head h as follows:

$$Q_{\text{org}}^h = y_{\text{org}} W_{Q_{\text{org}}}^h; K_{\text{org}}^h = y_{\text{org}} W_{K_{\text{org}}}^h; V_{\text{org}}^h = y_{\text{org}} W_{V_{\text{org}}}^h \tag{4}$$

Similarly, the Query, Key, Value projections for explanation text-level embedding y_{exp} for each head h as follows:

$$Q_{\text{exp}}^h = y_{\text{exp}} W_{Q_{\text{exp}}}^h; K_{\text{exp}}^h = y_{\text{exp}} W_{K_{\text{exp}}}^h; V_{\text{exp}}^h = y_{\text{exp}} W_{V_{\text{exp}}}^h \tag{5}$$

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

$$K_{\text{concat}}^h = [K_{\text{org}}^h, K_{\text{exp}}^h]; V_{\text{concat}}^h = [V_{\text{org}}^h, V_{\text{exp}}^h] \tag{6}$$

We use softmax attention to integrate complementary information from the mono-domain embeddings, focus on contextually relevant information, and semantically align them through an attention mechanism. The softmax function is applied to the keys for each query.

$$A_{\text{uni}}^h = \text{Softmax}\left(\frac{(Q_{\text{org}}^h + Q_{\text{exp}}^h) {K_{\text{concat}}^h}^T}{\sqrt{d_h}}\right) \tag{7}$$

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

$$O_{\text{uni}}^h = A_{\text{uni}}^h V_{\text{concat}}^h \tag{8}$$

Finally, all the head-specific outputs are concatenated and linearly transformed to create the unified mono-domain embedding as follows,

$$O_{\text{concat}} = [O^1_{\text{uni}}, O^2_{\text{uni}}, \ldots, O^H_{\text{uni}}] \tag{9}$$

$$y_{\text{uni}} = O_{\text{concat}} W_{O_{\text{uni}}} \tag{10}$$

where $W^h_{Q_{\text{org}}}$, $W^h_{K_{\text{org}}}$, $W^h_{V_{\text{org}}}$, $W^h_{V_{\text{exp}}}$, $W^h_{Q_{\text{exp}}}$, $W^h_{K_{\text{exp}}}$, $W_{O_{\text{uni}}}$ are the learnable weight matrices. Here, d_h represents the dimensionality of the key/query/value for each head, and H is the number of heads. y_{uni} denotes the unified mono-domain embeddings. The unified embeddings can learn and integrate complementary, diverse information present in both the y_{org} and y_{exp} embeddings. These unified embeddings facilitate semantic alignment among similar features across different embeddings and enable the identification of contextual relevance between distinct yet related y_{org} and y_{exp} mono-domain embeddings. The next step involves computing the cross-modal embedding y_{cross} using a second layer of a multihead attention mechanism that integrates both y_{pred} and y_{uni}. We compute the Query, Key, and Value projections for the prediction embedding y_{pred} for each head h as follows:

$$Q^h_{\text{pred}} = y_{\text{pred}} W^h_{Q_{\text{pred}}}; K^h_{\text{pred}} = y_{\text{pred}} W^h_{K_{\text{pred}}}; V^h_{\text{pred}} = y_{\text{pred}} W^h_{V_{\text{pred}}} \tag{11}$$

Similarly, we compute the Query, Key, Value projections for the unified embedding y_{uni} for each head h as follows:

$$Q^h_{\text{uni}} = y_{\text{uni}} W^h_{Q_{\text{uni}}}; K^h_{\text{uni}} = y_{\text{uni}} W^h_{K_{\text{uni}}}; V^h_{\text{uni}} = y_{\text{uni}} W^h_{V_{\text{uni}}} \tag{12}$$

We concatenate the keys and values from both the prediction and unified embeddings, thereby facilitating a robust integration of insights from the cross-domain embeddings into a synergized and enriched representation.

$$K^h_{\text{cross}} = [K^h_{\text{uni}}, K^h_{\text{pred}}]; V^h_{\text{cross}} = [V^h_{\text{uni}}, V^h_{\text{pred}}] \tag{13}$$

We utilize a softmax attention mechanism to merge and align information from different domains, thereby prioritizing contextually relevant information and ensuring semantic alignment. The softmax function is applied to the keys for each query, described as follows:

$$A^h_{\text{cross}} = \text{Softmax}\left(\frac{(Q^h_{\text{uni}} + Q^h_{\text{pred}}){K^h_{\text{cross}}}^T}{\sqrt{d_h}}\right) \tag{14}$$

In the multi-head attention mechanism, each head processes both embeddings(unified and predictive embeddings) to highlight important patterns, focusing on specific relationships or aspects within the data, enhancing performance in cross-modal learning tasks.

$$O^h_{\text{cross}} = A^h_{\text{cross}} V^h_{\text{cross}} \tag{15}$$

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

$$O_{\text{cross}} = [O^1_{\text{cross}}, O^2_{\text{cross}}, \ldots, O^H_{\text{cross}}] \tag{16}$$

$$y_{\text{cross}} = O_{\text{cross}} W_{O_{\text{cross}}} \tag{17}$$

where $W^h_{Q_{\text{uni}}}$, $W^h_{K_{\text{uni}}}$, $W^h_{V_{\text{uni}}}$, $W^h_{V_{\text{pred}}}$, $W^h_{Q_{\text{pred}}}$, $W^h_{K_{\text{pred}}}$, $W_{O_{\text{cross}}}$ are the learnable weight matrices. y_{cross} denotes the cross-domain embeddings. Implementing the hierarchical attention mechanism facilitates the structured integration of information from different modalities. This mechanism employs multi-head attention method, using multiple sets of learned weight matrices to emphasize various aspects or relationships within the data. Consequently, this approach has the potential to foster robust and enriched embeddings capable of capturing complex patterns. Additionally, it aids in focusing on contextually pertinent information and achieving semantic alignment across different embeddings, thereby enhancing the capacity to identify and utilize crucial features in the input data.

Output Layer: We then utilize a transformer decoder[28] to generate chemical SMILES representations character by character, using cross-modal embeddings (y_{cross}) that incorporate global context through the hierarchical multi-head self-attention mechanism. We implement a softmax layer to transform the decoder's output, creating a probability distribution over potential elements for each position in the SMILES strings. For our sequence generation tasks, we minimize the categorical cross-entropy loss to penalize the proposed framework based on the negative log-likelihood of the ground-truth chemical SMILES strings under the predicted probability distribution, thus facilitating the generation of valid molecules. In summary, by integrating multi-modal embeddings, namely y_{org}, y_{exp}, and y_{pred}, our approach enables the concurrent capture of complementary information, ultimately enhancing the overall performance of the framework.

4.3 Molecule captioning

Molecule captioning is a crucial task in the field of computational chemistry, serving as a bridge between complex chemical data and human comprehension. It involves generating detailed and correct textual descriptions that accurately describe a chemical SMILES representation in the `mol2text` task. This stands in contrast to the `text2mol` task, which entails generating chemical SMILES

representations from detailed and factual textual descriptions. Meanwhile, the `mol2text` task helps to translate complex chemical structures into understandable language, enhancing our understanding of molecules with potential applications spanning multiple fields, including drug discovery, materials science, and chemical synthesis. To evaluate the quality of the generated text in the `mol2text` task, we employ traditional metrics commonly used in natural language processing and machine translation, including BLEU, ROUGE, and METEOR, as described below:

- **BLEU-2** and **BLEU-4** are part of the BiLingual Evaluation Understudy (BLEU) metric family. BLEU is typically computed for different n-gram levels, where 'n' represents the number of contiguous words or tokens considered. BLEU-2 evaluates the accuracy of two-word phrases (bigrams) in generated text, while BLEU-4 extends this analysis to four-word sequences (4-grams). These metrics offer insights into the alignment between machine-generated and human reference texts. The BLEU metric variants help quantify the performance of language generation-based NLP models.
- **ROUGE-1** and **ROUGE-2** are part of the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric family. ROUGE-1, also known as ROUGE unigram or ROUGE-N1, evaluates the overlap between generated and reference text at the unigram (single-word) level, assessing specific word choices. In contrast, ROUGE-2 (also known as ROUGE bigram or ROUGE-N2) extends this to evaluate consecutive word pairs (bigrams), offering comprehensive insights into content matching. These metrics measure alignment with reference texts at both the word and bigram levels, providing precision and recall evaluations of textual elements.
- **ROUGE-L**, or Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence, measures the quality of machine-generated text by considering the longest common subsequence between the generated text and a reference text. This subsequence represents a sequence of words that appear in the same order in both the generated and reference texts, allowing flexibility in word order. ROUGE-L assesses content overlap and structural similarity, capturing the core content and organization of generated text concerning the reference text, even when there are variations in wording or word order.
- **METEOR** (Metric for the Evaluation of Translation with Explicit Ordering) considers precision, recall, stemming, synonymy, and word order, offering a well-rounded evaluation of text quality by analyzing matching words and their order, providing detailed assessments for translation and captioning models.

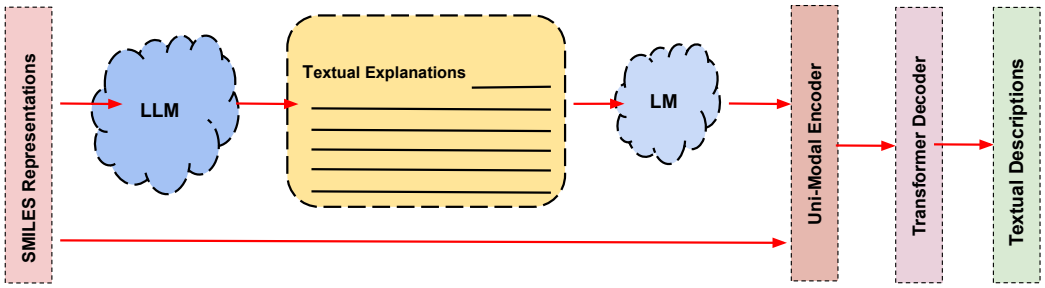

Figure 2: Overview of **FrontierX: LLM-MG** framework for `mol2text` task.

4.3.1 FrontierX: LLM-MG - `mol2text` task

We have modified the FrontierX: LLM-MG pipeline for the `text2mol` task to adapt it for the `mol2text` task. The workflow of the proposed approach is illustrated in Figure 2. Given a chemical SMILES representation, it can generate the technical descriptions of the molecule. We construct a knowledge-infused prompt using task-specific instructions and a few demonstrations (input-output mappings) for the downstream `mol2text` task. The task-specific instructions involve translating chemical SMILES representations into their corresponding technical descriptions. The primary objective of the prompt engineering method is to enhance the context-awareness of language models and improve their ability to provide relevant and accurate responses. This enhancement is achieved through learning from demonstrations, rather than relying on conventional supervised learning methods of fine-tuning with labeled data for the `mol2text` task. The knowledge-infused prompts guide the language models to generate technical descriptions based on the provided instructions. Next, we fine-tune small-scale, pre-trained language models (LMs) using the generated explanations,

which facilitates domain customization and yields context-aware token embeddings. To create a text-level embedding that encapsulates the generated technical descriptions, we utilize a weighted sum-pooling attention mechanism on the contextualized embeddings. Additionally, the unimodal encoder, which is implemented with a multi-head attention mechanism, integrates the mono-domain chemical SMILES representations with explanatory text-level embeddings to compute unimodal embeddings. Finally, the transformer decoder generates the technical descriptions that correspond to the input chemical SMILES representations[28]. Instead of repurposing large language models (LLMs) by either retraining them from scratch or fine-tuning them with labeled data for domain customization, we employ LMaaS[25] to engage with LLMs through text-based API interactions.

Table 8: The table shows the experimental findings of the framework performance on the `mol2text` task in comparison with the baselines.

Method	BLEU-2 (↑)	BLEU-4 (↑)	ROUGE-1 (↑)	ROUGE-2 (↑)	ROUGE-L (↑)	METEOR (↑)
GPT-4 (zero-shot)	0.062±0.001	0.013±0.001	0.192±0.002	0.040±0.002	0.125±0.002	0.209±0.002
GPT-4 (Scaffold, k=10)	0.464±0.008	0.365±0.008	0.545±0.003	0.362±0.003	0.459±0.007	0.519±0.005
GPT-4 (Scaffold, k=5)	0.456±0.003	0.357±0.004	0.540±0.005	0.355±0.007	0.455±0.005	0.505±0.005
GPT-4 (Random, k=10)	0.260±0.007	0.140±0.007	0.393±0.004	0.180±0.006	0.309±0.004	0.320±0.007
GPT-3.5 (Scaffold, k=10)	0.468±0.010	0.368±0.010	0.534±0.005	0.355±0.007	0.457±0.006	0.497±0.005
Davinci-003 (Scaffold, k=10)	0.488±0.011	0.391±0.012	0.532±0.008	0.359±0.010	0.465±0.008	0.478±0.011
FrontierX: LLM-MG (Scaffold, k=16, W/GPT-4)	**0.743 ±0.081**	**0.656 ±0.097**	**0.818 ±0.034**	**0.727 ±0.013**	**0.783 ±0.047**	**0.812 ±0.051**

Table 9: The table presents the framework performance and the baselines on the experimental study on the `mol2text` task.

Method	BLEU-2 (↑)	BLEU-4 (↑)	ROUGE-1 (↑)	ROUGE-2 (↑)	ROUGE-L (↑)	METEOR (↑)
RNN	0.251	0.176	0.450	0.278	0.394	0.363
Transformer	0.061	0.027	0.204	0.087	0.186	0.114
T5-Small	0.501	0.415	0.602	0.446	0.545	0.532
MolT5-Small	0.519	0.436	0.620	0.469	0.563	0.551
T5-Base	0.511	0.423	0.607	0.451	0.550	0.539
MolT5-Base	0.540	0.457	0.634	0.485	0.578	0.569
T5-Large	0.558	0.467	0.630	0.478	0.569	0.586
MolT5-Large	0.594	0.508	0.654	0.510	0.594	0.614
FrontierX: LLM-MG (Scaffold, k=16, W/GPT-4)	**0.743 ±0.081**	**0.656 ±0.097**	**0.818 ±0.034**	**0.727 ±0.013**	**0.783 ±0.047**	**0.812 ±0.051**

4.3.2 LLM Prompting

Knowledge-infused LLM prompting, a method of prompt engineering, involves crafting effective prompts or input queries to elicit desired responses from language models. This technique enhances language models by combining their natural language understanding and generation capabilities with access to external factual information(demonstrations), making them more versatile for task-specific applications. Consequently, it enables the LLM to generate responses enriched with accurate, contextually relevant information. Knowledge-infused prompting enables LLMs to adapt to new tasks without the need for explicit, gradient-based fine-tuning with gold-standard annotated data for task-specific adaptation in downstream tasks [2]). This approach allows LLMs to acquire knowledge through analogies, relying on a limited set of input-output mappings (demonstrations) tailored to the specific downstream task. Knowledge-infused prompting harnesses the implicit knowledge embedded in pretrained LLM parameters to facilitate adaptation to new tasks via task-specific demonstrations, all without necessitating parameter updates. The Knowledge-Infused prompt provides task-specific instructions and demonstrations, allowing LLMs to generate outputs conditioned on the prompt for improved generalization performance. In the case of `mol2text` tasks, we construct a knowledge-infused prompt using a few demonstrations sampled from the training data. To examine how the

quality and quantity of task-specific demonstrations impact performance on `mol2text` tasks, we investigate two different sampling strategies. The quality of examples is determined by the retrieval techniques employed to select the top-K demonstrations (chemical SMILES strings-text data pairs) from the training set that match the query chemical SMILES representations. We explore two distinct sampling strategies: 'Random' and 'Scaffold'. To study the impact of the quantity of demonstrations on the framework's performance on the `mol2text` task, we optimize the number of demonstrations (K) used to construct the augmented prompt for each query chemical SMILES representation. In the 'Random' strategy, we randomly sample K demonstrations from the training data. In contrast, the 'Scaffold' strategy uses Tanimoto similarity [26] based on Morgan fingerprints [15] with a radius of 2 to identify the top-K most similar chemical SMILES representations from the training data for query chemical SMILES representations. We explore the different sampling strategies to analyze the impact of the quality of demonstrations on the `mol2text` task with a hypothesis that the 'Scaffold' sampling technique outperforms the 'Random' technique for the same number of demonstrations. In summary, our goal is to task LLMs with a knowledge-infused prompt that consists of a few demonstrations for the `mol2text` task, along with task-specific instructions, where the output is technical descriptions of the query chemical SMILES representation. The task-specific instruction in the augmented prompt guides LLMs to generate technical descriptions. This task showcases the LLM's capacity to generate textual descriptions via prompt conditioning, relying on its inherent knowledge, without requiring parameter updates, in contrast to supervised learning, which relies on labeled data for parameter updates. Tables 8 and 9 present the experimental findings on the ChEBI-20 benchmark dataset[9]. We report the baseline results from earlier studies [10, 8]. The best performing model is in bold font.

4.3.3 Ablation Studies

Our proposed framework operates in a structured, multi-step pipeline. In step (a), we create knowledge-augmented prompts using task-specific instructions and demonstrations, prompting large language models (LLMs) to generate textual descriptions. In step (b), we use these generated explanations to fine-tune a smaller, pre-trained language model (LM_{exp}) for domain-specific customization, resulting in context-sensitive token embeddings. We employ a weighted sum-pooling attention mechanism for task-specific adaptation to compute text-level embeddings, denoted as y_{exp}, from the contextualized token embeddings. In parallel, in step (c), we fine-tune another small-scale language model (LM_{org}) on query chemical SMILES representations, computing an entire chemical SMILES string embeddings y_{org}. In step (d), our framework obtains a unimodal embedding, y_{uni}, through a multi-head attention mechanism that integrates the original text-level embeddings y_{org} and descriptive text-level embeddings y_{exp}. In the final step, the transformer decoder generates the textual descriptions of the query chemical SMILES string from the unimodal embedding, y_{uni}. Our empirical research aims to elucidate the significance and unique contributions of each method within our proposed framework, particularly in assessing the effectiveness of their learned embeddings for achieving optimal results on the `mol2text` task. Ablation studies have been conducted to investigate the impact of disabling individual methods on our framework's overall performance in the `mol2text` task. To precisely measure the impact of each method on the framework's performance, we have generated various ablated variants by disabling individual methods and assessed their performance using a benchmark dataset across multiple evaluation metrics for the `mol2text` task. We choose the `FrontierX: LLM-MG` framework as the reference baseline for ablation studies in the context of the `mol2text` task. Our comprehensive approach not only confirms the effectiveness of various methods but also provides substantial support for their design choices, reinforcing their rationale and justifying their inclusion in the framework. The ablated variants without the descriptive text-level embeddings and the original text-level embeddings are denoted as 'w/o y_{exp}' and 'w/o y_{org}', respectively. The findings of the ablation study are summarized in Table 10. All ablation experiments were conducted with the `FrontierX: LLM-MG W/GPT-4` framework using the Scaffold sampling technique with a value of $K = 16$, involving the deliberate exclusion of specific methods, as previously described. On the ChEBI-20 dataset[9], the 'w/o y_{exp}' variant exhibits a significant decline in performance relative to the baseline, evidenced by a 10.63% drop in BLEU-2, a 14.04% drop in ROUGE-L, and a 20.93% drop in METEOR. Similarly, the 'w/o y_{org}' variant performs much worse than the baseline, with a 20.18% drop in BLEU-2, a 25.06% drop in ROUGE-L, and a 29.06% drop in METEOR. These substantial performance drops across all evaluation metrics when comparing the ablated variants to the baseline consistently highlighting the significant impact of the mechanisms disabled from the baseline. Our experiments validate our hypothesis on joint optimization to obtain a unimodal embedding, denoted as y_{uni}, through a multi-head attention mechanism that combines the original text-level embeddings y_{org} and explanatory text-level embeddings y_{exp}. This approach results in achieving state-of-the-art (SOTA) performance on the `mol2text` task.

Table 10: The table shows the experimental findings on the ablation study on the `mol2text` task.

Method	BLEU-2 (↑)	BLEU-4 (↑)	ROUGE-1 (↑)	ROUGE-2 (↑)	ROUGE-L (↑)	METEOR (↑)
FrontierX: LLM-MG (Scaffold, k=16, W/GPT-4)	**0.743 ±0.081**	**0.656 ±0.097**	**0.818 ±0.034**	**0.727 ±0.013**	**0.783 ±0.047**	**0.812 ±0.051**
FrontierX: LLM-MG — w/o y_{exp}	0.664 ±0.058	0.537 ±0.071	0.675 ±0.042	0.543 ±0.035	0.673 ±0.021	0.642 ±0.075
FrontierX: LLM-MG — w/o y_{org}	0.593 ±0.027	0.496 ±0.068	0.612 ±0.086	0.482 ±0.037	0.575 ±0.076	0.576 ±0.059

References

[1] 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.

[2] 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.

[3] 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.

[4] Debadutta Dash, Rahul Thapa, Juan M Banda, Akshay Swaminathan, Morgan Cheatham, Mehr Kashyap, Nikesh Kotecha, Jonathan H Chen, Saurabh Gombar, Lance Downing, et al. Evaluation of gpt-3.5 and gpt-4 for supporting real-world information needs in healthcare delivery. *arXiv preprint arXiv:2304.13714*, 2023.

[5] 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.

[6] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.

[9] Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, 2021.

[10] Taicheng Guo, Kehan Guo, Zhengwen Liang, Zhichun Guo, Nitesh V Chawla, Olaf Wiest, Xiangliang Zhang, et al. What indeed can gpt models do in chemistry? a comprehensive benchmark on eight tasks. *arXiv preprint arXiv:2305.18365*, 2023.

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

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

[13] G. A. Landrum. Rdkit: Open-source cheminformatics software. http://www.rdkit.org, 2020.

[14] Frederic P Miller, Agnes F Vandome, and John McBrewster. Levenshtein distance: Information theory, computer science, string (computer science), string metric, damerau? levenshtein distance, spell checker, hamming distance, 2009.

[15] 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.

[16] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter Klambauer. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741, 2018.

[17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[19] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[20] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[21] David Ratcliff, John W.; Metzener. Pattern matching: The gestalt approach, 1988.

[22] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021.

[23] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[24] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[25] 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.

[26] Taffee T Tanimoto. Elementary mathematical theory of classification and prediction. *Journal of Biomedical Science and Engineering*, 1958.

[27] 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.

[28] 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.

[29] 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.