
UniMoT: Unified Molecule-Text Language Model with Discrete Token Representation

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 The remarkable success of Large Language Models (LLMs) across diverse tasks
2 has driven the research community to extend their capabilities to molecular appli-
3 cations, leading to the development of molecular LLMs. However, most molecular
4 LLMs employ adapter-based architectures that do not treat molecule and text
5 modalities equally and lack a supervision signal for the molecule modality. To
6 address these issues, we introduce **UniMoT**, a unified molecule-text LLM adopting
7 a tokenizer-based architecture that expands the vocabulary of LLM with molecule
8 tokens. Specifically, we introduce a Vector Quantization-driven tokenizer that
9 incorporates a Q-Former to bridge the modality gap between molecule and text.
10 This tokenizer transforms molecules into sequences of molecule tokens with causal
11 dependency, encapsulating high-level molecular and textual information. Equipped
12 with this tokenizer, UniMoT can unify molecule and text modalities under a shared
13 token representation and an autoregressive training paradigm, enabling it to in-
14 terpret molecules as a foreign language and generate them as text. Following a
15 four-stage training scheme, UniMoT emerges as a multi-modal generalist capable
16 of performing both molecule-to-text and text-to-molecule tasks. Extensive exper-
17 iments demonstrate that UniMoT achieves state-of-the-art performance across a
18 wide range of molecule comprehension and generation tasks.

19 1 Introduction

20 The incredible capabilities of Large Language Models (LLMs) [5, 44] have led to their widespread
21 use as versatile tools for completing diverse real-world tasks. This success has sparked interest in
22 Multi-modal LLMs [59, 52], which aim to enhance LLMs by enabling them to process multi-modal
23 inputs and outputs. Prior research efforts [26, 41, 12, 6, 33, 35, 25] have focused on adapting LLMs
24 to molecular tasks, resulting in the development of molecular LLMs. These molecular LLMs can
25 analyze molecule structures [35, 33, 6], address drug-related inquiries [26, 41], assist in synthesis
26 and retrosynthesis planning [12], support drug design [12], and more.

27 Prevalent molecular LLMs commonly employ adapter-based architectures, adopting either a linear
28 projection [26, 41, 6] or a Q-Former [33, 25] as an adapter to translate molecule features into the
29 semantic space of LLM, as illustrated in Figure 1a and Figure 1b. Despite demonstrating initial
30 capabilities in molecular comprehension and yielding promising results in molecule-to-text generation
31 tasks, they still lack molecule generation abilities. The critical issue within these methods is their
32 unequal treatment of molecules and text, resulting in a lack of supervision for the molecule modality.
33 This limitation significantly constrains model capacity and effectiveness. Due to limitations imposed
34 by the training paradigm, they are unable to perform text-to-molecule generation tasks.

35 Discretizing continuous molecule features into discrete molecule tokens offers a promising solution
36 for conducting both molecule-to-text and text-to-molecule generation tasks. By treating tokens from

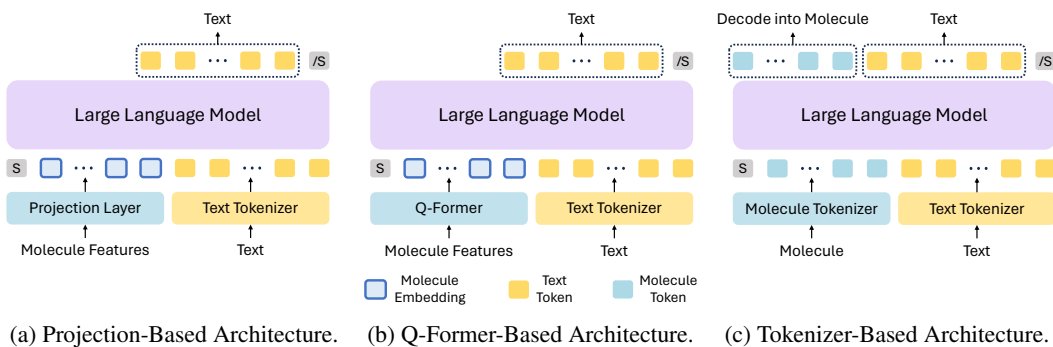


Figure 1: Comparisons among different molecular LLMs. 1a and 1b are adapter-based architectures that do not treat molecule and text modalities equally and lack a supervision signal for the molecule modality. 1c is our proposed tokenizer-based architecture, where molecules are presented in the same discrete token representation as that of text. Molecules and text can be optimized under a unified next-token-prediction objective.

37 different modalities equally, we can predict the next molecule or text token in an autoregressive
 38 manner. However, directly discretizing molecule features poses several challenges: (i) This approach
 39 results in long sequences, with lengths equivalent to the number of atoms in a batch. LLMs typically
 40 experience a quadratic increase in computational complexity with sequence length [46]. (ii) Molecule
 41 tokens derived from molecule features lack left-to-right causal dependency, which conflicts with
 42 the unidirectional attention mechanism in LLMs. (iii) Molecule features lack textual information,
 43 hindering effective molecule-text interactions and alignment.

44 To this end, we present **UniMoT**, a unified molecule-text LLM that adopts a tokenizer-based architec-
 45 ture, integrating molecule comprehension and generation, as depicted in Figure 1c. A pivotal aspect
 46 of UniMoT’s architecture is the molecule tokenizer for transforming molecules into molecule tokens.
 47 We introduce a Vector Quantization-driven [45] tokenizer, incorporating a Q-Former [23] to bridge
 48 the modality gap between molecule and text. Specifically, we incorporate causal masks for the queries,
 49 enabling the Causal Q-Former to generate a causal sequence of query embeddings compatible with
 50 the unidirectional attention in LLMs. The sequence of query embeddings is subsequently quantized
 51 into a sequence of molecule tokens using a learnable codebook. The molecule tokens encapsulate
 52 high-level molecular and textual information, which are then aligned with the latent space of a
 53 generative model via an MLP adapter, enabling the generation of desired molecules.

54 Pretrained LLMs can integrate the molecule tokenizer by treating molecule tokens as new words and
 55 constructing a molecule vocabulary through mapping the learned codebook. We adopt the unified
 56 discrete token representation for molecules and text, coupled with the unified next-token-prediction
 57 training paradigm of LLM. This unification of representation and training paradigm enhances LLMs’
 58 ability to understand molecule-text interactions and alignment. UniMoT interprets molecules akin to
 59 understanding a foreign language, and generates them as if they were text. Following a four-stage
 60 training scheme, UniMoT serves as a multi-modal generalist capable of performing both molecule
 61 comprehension and generation tasks.

62 Our contributions can be summarized as follows:

- 63 • We introduce a molecule tokenizer specifically designed for LLMs, enabling the tokenization
 64 of molecules into short sequences of molecule tokens with causal dependency. These tokens
 65 encapsulate high-level molecular and textual information and can be decoded into desired
 66 molecules during inference.
- 67 • We present UniMoT, a unified molecule-text LLM that adopts a tokenizer-based architecture
 68 instead of traditional adapter-based architectures. UniMoT unifies the modalities of molecule
 69 and text under a shared token representation and an autoregressive training paradigm.
- 70 • UniMoT exhibits remarkable capabilities in multi-modal comprehension and generation. Exten-
 71 sive experiments demonstrate that UniMoT achieves state-of-the-art performance across a wide
 72 spectrum of molecule comprehension tasks and molecule generation tasks.

73 2 Related Works

74 **Molecular Large Language Models.** The recent emergence of Vision Large Language Models
75 (VLLMs) [24, 23, 28] has catalyzed advancements in Molecular LLMs, which encompass both
76 single modality and multi-modality approaches. In the single modality domain, researchers are
77 exploring diverse molecule representations, such as 1D sequences like SMILES strings [47, 8, 17],
78 2D molecule graphs [15, 56], 3D geometric conformations [56, 32], and textual information from
79 the literature [43, 2, 21]. In the multiple modalities domain, various innovative approaches are being
80 employed. MolT5 [11], a T5-based [38] model, is designed for SMILES-to-text and text-to-SMILES
81 translations. Other works, such as MoMu [39], MoleculeSTM [31], MolFM [34], and GIT-Mol [29],
82 leverage cross-modal contrastive learning to align the representation spaces of molecules and text.
83 Additionally, some studies use multi-modal learning architectures to develop molecular LLMs,
84 which often adopt adapter-based architectures. For instance, InstructMol [6], GraphGPT [41], and
85 DrugChat [26] employ a simple projection layer to map molecule features to LLM’s input space.
86 MolCA [33] and 3D-MoLM [25] utilize a Q-Former [23] to bridge the modality gap between
87 molecules and text. However, these methods do not treat molecule and text modalities equally and
88 lack a supervision signal for the molecule modality, limiting model capacity and effectiveness.

89 **Vector Quantization.** Vector Quantization (VQ) [13] is a widely used technique in generative
90 models. VQ-VAE [45] converts an image into a set of discrete codes within a learnable discrete
91 latent space by learning to reconstruct the original image. VQ-GAN [57] enhances the generation
92 quality by leveraging adversarial and perceptual objectives. In the context of molecules, VQ has
93 been effectively applied to quantize molecule representations. For example, DGAE [4] introduces
94 a VQ model specifically for molecular graphs, where molecular graphs are encoded into discrete
95 latent codes. Mole-BERT [54] uses VQ to rethink the pre-training of GNNs for molecular tasks.
96 IMoLD [60] proposes using VQ to enhance invariant molecule representations, and VQSynergy [51]
97 demonstrates the use of VQ for drug discovery.

98 3 Method

99 Our objective is to leverage the reasoning and generation capabilities of LLMs to enhance the
100 comprehension and generation of molecule and text data. To achieve this, we focus on representing
101 these modalities uniformly within the token representation, utilizing the next-token-prediction training
102 paradigm of LLMs. As illustrated in Figure 2, we introduce a molecule tokenizer (Section 3.1)
103 designed to transform molecules into molecule tokens by learning to reconstruct the input molecule.
104 The molecule sequence can then be concatenated with the text sequence to form a multi-modal
105 sequence, which is subsequently fed into an LLM for autoregressive pretraining (Section 3.2), as
106 illustrated in Figure 3. The LLM vocabulary is expanded with molecule codes mapped from the
107 learned codebook. We introduce a four-stage training scheme for UniMoT (Section 3.3) comprising
108 Causal Q-Former pretraining, molecule tokenizer pretraining, unified molecule-text pretraining, and
109 task-specific instruction tuning. UniMoT is capable of performing both molecular comprehension
110 and generation tasks following the training scheme.

111 3.1 Molecule Tokenizer for LLMs

112 **Molecule encoder.** We represent the structural information of a molecule as a graph, denoted by
113 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of atoms and $|\mathcal{V}| = N$ is the number of atoms. The task of the
114 molecule encoder is to extract node representations that are context-aware and encompass diverse
115 local neighborhood structural information. By employing a molecule encoder, we obtain molecule
116 features $\mathbf{X} \in \mathbb{R}^{N \times F}$, where each atom representation contains context-aware structural information.

117 **Causal Q-Former.** We employ a Q-Former model introduced by BLIP-2 [23] to generate query
118 embeddings $\mathbf{Z} = \{z_i\}_{i=1}^M \in \mathbb{R}^{M \times d}$ containing high-level molecular and textual information, where
119 M represents the number of queries and d denotes the dimension of query embeddings. Specifically,
120 we incorporate causal masks into the queries, ensuring that they only interact with preceding queries.
121 This ensures the sequence of query embeddings maintains a causal dependency, aligning with the
122 requirements of LLMs operating on text sequence. Details regarding the Causal Q-Former can be
123 found in Appendix A.

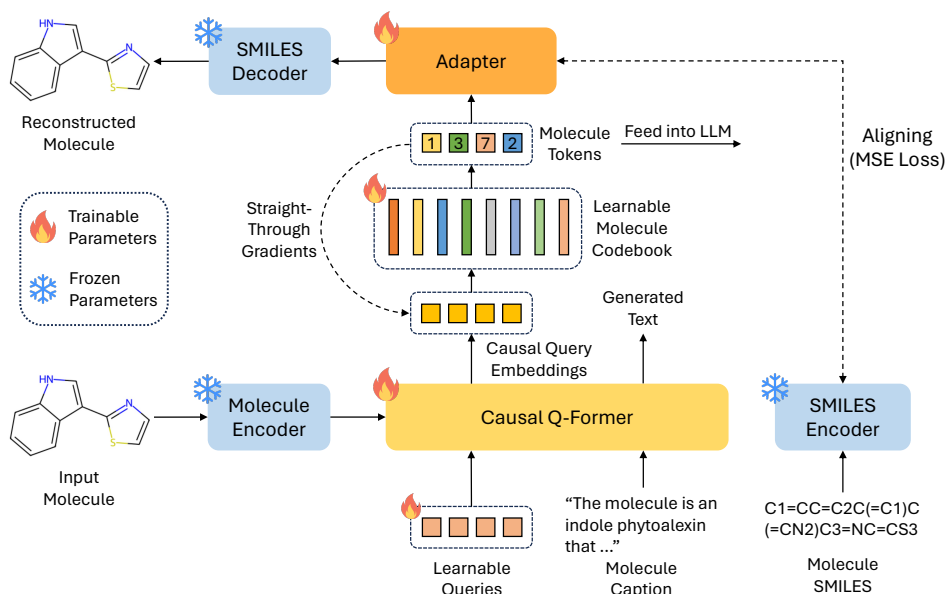


Figure 2: Illustration of our proposed molecule tokenizer. The tokenizer generates discrete molecule tokens, which can be fed into LLMs for downstream tasks. The generated molecule tokens can be decoded into molecules using the adapter and the SMILES decoder during inference.

124 **Vector Quantization.** The Causal Q-Former converts molecule and text features into a causal
 125 sequence of query embeddings. Subsequently, we aim to quantize these query embeddings into
 126 molecule tokens using a variant of VQ-VAE [45]. These discrete molecule tokens can then be
 127 integrated with text tokens to form a multi-modal sequence suitable for feeding into LLMs. The
 128 causal sequence of query embeddings $\{z_i\}_{i=1}^M$ is quantized into a causal sequence of molecule
 129 tokens $\{s_i\}_{i=1}^M$ by identifying the closest neighbor in a learnable codebook $\mathcal{C} = \{c_i\}_{i=1}^K$, where K
 130 represents the size of the codebook. The codebook is randomly initialized and optimized during
 131 pretraining. Specifically, token s_i is determined as follows:

$$s_i = \operatorname{argmin}_{j \in \{1, \dots, K\}} \|z_i - c_j\|_2, \quad \text{for } i = 1, 2, \dots, M. \quad (1)$$

132 Intuitively, the query embedding z_i is quantized to the closest neighbor c_{s_i} in the codebook. As the
 133 vector quantization process is non-differentiable, we adopt the straight-through estimator [3] to train
 134 the Causal Q-Former by copying the gradient from the molecule tokens to the query embeddings,
 135 as shown in Figure 2. The resulting embeddings of molecule tokens, denoted as $\mathbf{C} = \{c_{s_i}\}_{i=1}^M$, are
 136 subsequently utilized for reconstructing molecules.

137 **Molecule Reconstruction.** An adapter needs to be trained to align the discrete latent space of
 138 molecule tokens with the continuous latent space of a molecular generative model for molecule
 139 reconstruction. The embeddings of molecule tokens \mathbf{C} can be aligned with the latent space of
 140 the generative model via an MLP adapter ψ , represented as $\mathbf{X}_R = \psi(\mathbf{C})$, where \mathbf{X}_R denotes the
 141 embeddings for reconstruction. Subsequently, we can reconstruct the molecule from \mathbf{X}_R using the
 142 pretrained SMILES decoder. To achieve alignment, we minimize the Mean Squared Error (MSE) loss
 143 between \mathbf{X}_R and the SMILES [50] embeddings \mathbf{X}_S produced by the pretrained SMILES encoder.
 144 The training loss of the tokenizer is expressed as follows:

$$\mathcal{L}_{\text{Tokenizer}} = \|\mathbf{X}_R - \mathbf{X}_S\|_2^2 + \frac{1}{M} \sum_{i=1}^M \|\operatorname{sg}[z_i] - c_{s_i}\|_2^2 + \frac{\beta}{M} \sum_{i=1}^M \|\operatorname{sg}[c_{s_i}] - z_i\|_2^2. \quad (2)$$

145 Here, the first term represents the alignment loss, the second term is a codebook loss aimed at
 146 updating the codebook embeddings, and the third term is a commitment loss that encourages the
 147 query embedding to stay close to the chosen codebook embedding. $\operatorname{sg}[\cdot]$ denotes the stop-gradient
 148 operator, and the hyperparameter β is set to 0.25.

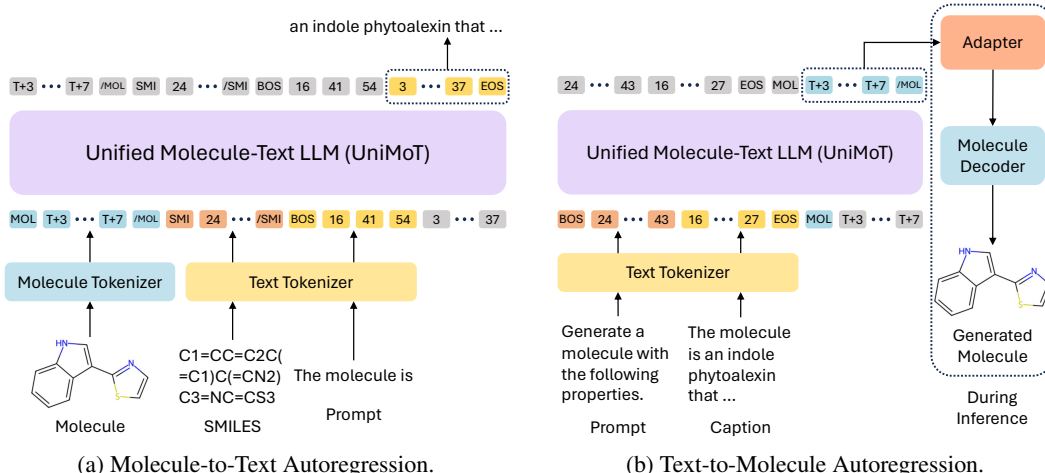


Figure 3: Illustration of the multi-modal autoregressive pretraining on molecule-text datasets. UniMoT excels in multi-modal comprehension and generation tasks, enabled by the unified LM objective. T represents the size of the text vocabulary.

149 3.2 Unified Molecule-Text Language Model

150 **Expanding Vocabulary.** Employing the molecule tokenizer, a molecule can be tokenized into a
 151 molecule sequence $\{s_i\}_{i=1}^M$ with causal dependency. The molecule sequence can be concatenated with
 152 the text sequence to form a multi-modal sequence $\{u_i\}_{i=1}^L$, where L is the length of the multi-modal
 153 sequence. To facilitate the representation of the multi-modal sequence, we construct the molecule
 154 vocabulary $\mathcal{V}^m = \{v_i^m\}_{i=1}^K$, which maintains the order of the molecule codebook $\mathcal{C} = \{c_i\}_{i=1}^K$.
 155 Additionally, \mathcal{V}^m includes several special tokens such as boundary indicators, e.g., [MOL] and
 156 [/MOL], to mark the beginning and end of the molecule sequence. Next, we merge the original text
 157 vocabulary $\mathcal{V}^t = \{v_i^t\}_{i=1}^T$ with the molecule vocabulary \mathcal{V}^m . The unified molecule-text vocabulary
 158 $\mathcal{V} = \{\mathcal{V}^m, \mathcal{V}^t\}$ facilitates joint learning from molecules and text under a unified next-token-prediction
 159 objective. As the vocabulary is expanded, the corresponding embeddings and prediction layers also
 160 need to be extended, with the newly introduced parameters initialized randomly.

161 **Unified Molecule-text Modeling.** The multi-modal sequence $\{u_i\}_{i=1}^L$ is fed into the pretrained
 162 LLM for performing multi-modal autoregression. UniMoT adopts the general Language Modeling
 163 (LM) objective to directly maximize the log-likelihood of the data distribution:

$$\mathcal{L}_{\text{LM}} = - \sum_{u \in \mathcal{D}} \sum_{i \in \mathcal{I}} \log p(u_i | u_1, \dots, u_{i-1}; \theta), \quad (3)$$

164 where \mathcal{D} represents the dataset, \mathcal{I} represents the set of indices of the generation target, and θ denotes
 165 the parameters of the LLM. The unification of representation and training paradigm for molecules and
 166 text enhances the abilities of LLMs to understand molecule-text interactions and alignment. UniMoT
 167 can interpret molecules similar to understanding a foreign language, and generate them as if they
 168 were text. We conduct autoregressive pretraining on molecule-to-text and text-to-molecule tasks to
 169 enhance the molecule comprehension and generation capabilities.

170 **Molecule-to-Text Autoregression.** While structural information is embedded in molecule features
 171 and captured by the molecule tokens through the tokenizer, we also aim to incorporate sequential
 172 information of molecules for better comprehension. Therefore, we concatenate the molecule sequence
 173 $\{s_i\}_{i=1}^M$ with the SMILES [50] sequence and a prompt to form the multi-modal input sequence
 174 $\{u_i\}_{i=1}^L$, as illustrated in Figure 3a. The text sequence of the corresponding molecule caption is used
 175 as the generation target.

176 **Text-to-Molecule Autoregression.** For molecule generation, a prompt and the molecule caption
 177 are concatenated, with a [MOL] token appended to signify the beginning of the molecule sequence,
 178 as illustrated in Figure 3b. The molecule sequence $\{s_i\}_{i=1}^M$ produced by the tokenizer is used as the

179 generation target. During inference, given a prompt and the molecule caption, the output molecule
180 sequence can be decoded into the desired molecule by the pretrained adapter and SMILES decoder.

181 3.3 Training Strategy

182 The training strategy for UniMoT is structured across four stages. Stage-1 focuses on Causal Q-
183 Former pretraining with tailored objectives. In Stage-2, the molecule tokenizer is optimized using the
184 frozen encoders and decoder. Stage-3 integrates the tokenizer with a language model for multi-modal
185 comprehension and generation. Finally, Stage-4 fine-tunes UniMoT for specific tasks, aligning it with
186 human instructions and optimizing performance for various molecular applications. More details
187 regarding the training process can be found in Appendix C.

188 **Stage-1: Causal Q-Former Pretraining.** We connect the molecule encoder and Causal Q-Former,
189 leveraging the pretrained MoleculeSTM molecule encoder [31]. The molecule encoder remains
190 frozen while only the Causal Q-Former is updated. Both queries and text inputs are used, while
191 only queries serve as input in subsequent stages. In our experiments, we utilize 16 queries. We
192 employ three tailored objectives MTC, MTM, and MTG for the pretraining of the Causal Q-Former,
193 as detailed in Appendix A.

194 **Stage-2: Molecule Tokenizer Pretraining.** We connect the Causal Q-Former with subsequent
195 blocks and use the objective defined in Equation (2). We employ the pretrained ChemFormer [17] as
196 the generative model. Specifically, we leverage the SMILES encoder and SMILES decoder provided
197 by ChemFormer. The molecule codebook size is set to $K = 2048$. As shown in Figure 2, we keep
198 the molecule encoder, SMILES encoder, and SMILES decoder frozen, while updating the Causal
199 Q-Former, codebook, and adapter.

200 **Stage-3: Unified Molecule-Text Pretraining.** We integrate the molecule tokenizer with the LLM
201 using the unified vocabulary of molecule tokens and text tokens. We employ the LM objective
202 defined in Equation (3) to pretrain the LLM. Pretraining involves molecule-to-text autoregression
203 and text-to-molecule autoregression, aimed at enhancing UniMoT’s multi-modal comprehension and
204 generation capabilities. To enhance efficiency, we train the LLM using LoRA tuning [14].

205 **Stage-4: Task-Specific Instruction Tuning.** UniMoT is fine-tuned on seven comprehension and
206 generation tasks: molecular property prediction, molecule captioning, molecule-text retrieval, caption-
207 guided molecule generation, reagent prediction, forward reaction prediction, and retrosynthesis. We
208 also utilize LoRA tuning to improve efficiency. This stage ensures UniMoT can accurately interpret
209 and respond to human instructions, making it versatile and effective for molecular tasks.

210 4 Experiments

211 4.1 Molecule Comprehension Tasks

212 **Molecular Property Prediction Task.** The goal of molecular property prediction is to forecast
213 a molecule’s intrinsic physical and chemical properties. For the classification task, we incorporate
214 eight binary classification datasets from MoleculeNet [53]. Models are tasked with generating
215 a single prediction (“yes” or “no”). We compare UniMoT with the following baselines: KV-
216 PLM [58], AttrMask [16], InfoGraph [40], MolCLR [48], GraphMVP [30], MoleculeSTM [31],
217 and InstructMol [6]. The ROC-AUC (%) results on the MoleculeNet datasets are shown in Table 1.
218 The performance of the regression task of molecular property prediction is provided in Appendix D.
219 Compared to traditional graph learning methods and molecular LLMs like InstructMol, UniMoT
220 demonstrates consistent improvements across the eight datasets, indicating its robust molecule
221 comprehension abilities.

222 **Molecule Captioning Task.** The molecule captioning task involves generating a comprehensive
223 description of a molecule. We compare UniMoT with several baselines: MolT5 [11], MoMu [39],
224 InstructMol [6], MolCA [33], and 3D-MoLM [25]. BLEU [37], ROUGE [27], and METEOR [1] are
225 adopted as evaluation metrics. UniMoT is evaluated for molecule captioning on the PubChem and

Table 1: ROC-AUC (%) of molecular property prediction task (classification) on the MoleculeNet datasets. **Bold** indicates the best performance and underline indicates the second best performance.

Model	BBBP \uparrow	Tox21 \uparrow	ToxCast \uparrow	Sider \uparrow	ClinTox \uparrow	MUV \uparrow	HIV \uparrow	BACE \uparrow
KV-PLM [58]	<u>70.50</u>	72.12	55.03	59.83	89.17	54.63	65.40	78.50
AttrMask [16]	67.79	75.00	63.57	58.05	75.44	73.76	75.44	80.28
InfoGraph [40]	64.84	76.24	62.68	59.15	76.51	72.97	70.20	77.64
MolCLR [48]	67.79	75.55	64.58	58.66	84.22	72.76	75.88	71.14
GraphMVP [30]	68.11	77.06	<u>65.11</u>	<u>60.64</u>	84.46	74.38	<u>77.74</u>	80.48
MoleculeSTM [31]	69.98	<u>76.91</u>	<u>65.05</u>	60.96	<u>92.53</u>	73.40	76.93	80.77
InstructMol (Vicuna-7B) [6]	70.00	<u>74.67</u>	64.29	57.80	<u>91.48</u>	<u>74.62</u>	68.90	<u>82.30</u>
UniMoT (LLaMA2-7B)	71.37	76.43	65.78	59.79	92.89	75.97	78.49	83.69

Table 2: Performance (%) of molecule captioning task on the PubChem dataset. **Bold** indicates the best performance and underline indicates the second best performance.

Model	BLEU-2 \uparrow	BLEU-4 \uparrow	ROUGE-1 \uparrow	ROUGE-2 \uparrow	ROUGE-L \uparrow	METEOR \uparrow
MolT5-Small (T5-Small) [11]	22.5	15.2	30.4	13.5	20.3	24.0
MolT5-Base (T5-Base) [11]	24.5	16.6	32.2	14.0	21.4	26.1
MolT5-Large (T5-Large) [11]	25.9	17.3	34.1	16.4	23.4	28.0
MoMu-Small (T5-Small) [39]	22.9	16.0	31.0	13.7	20.8	24.4
MoMu-Base (T5-Base) [39]	24.7	16.8	32.5	14.6	22.1	27.2
MoMu-Large (T5-Large) [39]	26.3	18.0	34.8	16.9	24.8	28.7
InstructMol (Vicuna-7B) [6]	18.9	11.7	27.3	11.8	17.8	21.3
MolCA (OPT-125M) [33]	25.9	17.5	34.4	16.6	23.9	28.5
MolCA (OPT-1.3B) [33]	28.6	21.3	36.2	21.4	29.7	32.6
3D-MoLM (LLaMA2-7B) [25]	<u>30.3</u>	<u>22.5</u>	<u>36.8</u>	<u>22.3</u>	<u>31.2</u>	<u>33.1</u>
UniMoT (LLaMA2-7B)	31.3	23.8	37.5	23.7	33.6	34.8

226 CheBI-20 datasets. Performance on the PubChem dataset is shown in Table 2, while the performance
 227 on the CheBI-20 dataset and some concrete examples are presented in Appendix D.

228 From Table 2, we observe that UniMoT consistently outperforms the baselines by a significant margin.
 229 This task is more complex than classification or regression, providing a robust measure of the model’s
 230 molecule comprehension abilities. Notably, our proposed tokenizer-based architecture surpasses the
 231 projection-based architecture (such as InstructMol), Q-Former-based architecture (such as MolCA
 232 and 3D-MoLM), and models trained with contrastive learning strategies (such as MoMu). The results
 233 demonstrate that the molecule tokenizer can generate molecule tokens with high-level molecular and
 234 textual information, enhancing molecule comprehension abilities.

235 **Molecule-Text Retrieval Task.** The molecule-text retrieval task involves using a molecule to
 236 retrieve text (M2T) and using text to retrieve a molecule (T2M). We compare UniMoT with several
 237 baselines: Sci-BERT [2], KV-PLM [58], MoMu [39], MoleculeSTM [31], MolCA [33], and 3D-
 238 MoLM [25]. We report the performance of retrieval using a batch of 64 random samples and the entire
 239 test set, evaluated with the metrics of Accuracy and Recall@20. We use the checkpoint from Stage-1
 240 of pretraining. UniMoT is evaluated on the datasets of PubChem, PCdes, and MoMu. Performance
 241 on the PubChem dataset is shown in Table 3, while performance on the PCdes and MoMu datasets is
 242 presented in Appendix D. UniMoT can understand complex molecule-text interactions through the
 243 introduction of the Causal Q-Former. From Table 3, UniMoT demonstrates superior performance over
 244 the baselines on molecule-text retrieval, particularly in molecule-to-text retrieval. This underscores
 245 UniMoT’s capability in learning fine-grained alignment between molecules and text.

246 4.2 Molecule Generation Tasks

247 We employ molecule generation tasks, which encompass caption-guided molecule generation, reagent
 248 prediction, forward reaction prediction, and retrosynthesis. Caption-guided molecule generation
 249 involves generating molecular structures based on textual descriptions. Reagent prediction entails
 250 determining suitable reagents given reactants and products. Forward reaction prediction involves
 251 predicting probable products given specific reactants and reagents. Retrosynthesis involves decon-
 252 structing a target molecule into simpler starting materials. We compare UniMoT with the following

Table 3: Performance (%) of molecule-text retrieval task on the PubChem dataset. **Bold** indicates the best performance and underline indicates the second best performance.

Model	Retrieval in batch				Retrieval in test set			
	M2T (%)		T2M (%)		M2T (%)		T2M (%)	
	Acc \uparrow	R@20 \uparrow	Acc \uparrow	R@20 \uparrow	Acc \uparrow	R@20 \uparrow	Acc \uparrow	R@20 \uparrow
Sci-BERT [2]	85.3	98.7	84.2	98.4	41.7	87.3	40.2	86.8
KV-PLM [58]	86.1	98.6	85.2	98.5	42.8	88.5	41.7	87.8
MoMu (Sci-BERT) [39]	87.6	99.2	86.4	99.4	47.3	90.8	48.1	89.9
MoMu (KV-PLM) [39]	88.2	99.4	87.3	99.4	48.5	91.6	49.5	90.7
MoleculeSTM [31]	90.5	99.6	88.6	<u>99.5</u>	52.7	92.9	53.2	92.5
MolCA (OPT-1.3B) [33]	92.6	99.8	91.3	<u>99.5</u>	67.9	94.4	68.6	93.3
3D-MoLM (LLaMA2-7B) [25]	<u>93.5</u>	100.0	92.9	99.6	<u>69.1</u>	<u>95.9</u>	70.1	94.9
UniMoT (LLaMA2-7B)	93.6	100.0	<u>92.7</u>	99.4	69.5	96.3	<u>69.8</u>	<u>94.4</u>

Table 4: Performance of molecule generation tasks on the Mol-Instructions dataset, including caption-guided molecule generation, reagent prediction, forward reaction prediction, and retrosynthesis. **Bold** indicates the best performance, and underline indicates the second best performance.

Model	Exact \uparrow	BLEU \uparrow	Levenshtein \downarrow	RDK FTS \uparrow	MACCS FTS \uparrow	Morgan FTS \uparrow	Validity \uparrow
Caption-guided Molecule Generation							
LLaMA [44]	0.000	0.003	59.864	0.005	0.000	0.000	0.003
Vicuna [7]	0.000	0.006	60.356	0.006	0.001	0.000	0.001
Mol-Instructions [12]	0.002	0.345	41.367	0.231	0.412	0.147	1.000
MolT5 [11]	<u>0.112</u>	<u>0.546</u>	<u>38.276</u>	<u>0.400</u>	<u>0.538</u>	<u>0.295</u>	0.773
UniMoT	0.237	0.698	27.782	0.543	0.651	0.411	1.000
Reagent Prediction							
LLaMA [44]	0.000	0.003	28.040	0.037	0.001	0.001	0.001
Vicuna [7]	0.000	0.010	27.948	0.038	0.002	0.001	0.007
Mol-Instructions [12]	0.044	0.224	23.167	0.237	0.364	0.213	1.000
InstructMol [6]	<u>0.129</u>	<u>0.610</u>	<u>19.664</u>	<u>0.444</u>	<u>0.539</u>	<u>0.400</u>	1.000
UniMoT	0.167	0.728	14.588	0.549	0.621	0.507	1.000
Forward Reaction Prediction							
LLaMA [44]	0.000	0.020	42.002	0.001	0.002	0.001	0.039
Vicuna [7]	0.000	0.057	41.690	0.007	0.016	0.006	0.059
Mol-Instructions [12]	0.045	0.654	27.262	0.313	0.509	0.262	1.000
InstructMol [6]	<u>0.536</u>	<u>0.967</u>	<u>10.851</u>	<u>0.776</u>	<u>0.878</u>	<u>0.741</u>	1.000
UniMoT	0.611	0.980	8.297	0.836	0.911	0.807	1.000
Retrosynthesis							
LLaMA [44]	0.000	0.036	46.844	0.018	0.029	0.017	0.010
Vicuna [7]	0.000	0.057	46.877	0.025	0.030	0.021	0.017
Mol-Instructions [12]	0.009	0.705	31.227	0.283	0.487	0.230	1.000
InstructMol [6]	<u>0.407</u>	<u>0.941</u>	<u>13.967</u>	<u>0.753</u>	<u>0.852</u>	<u>0.714</u>	1.000
UniMoT	0.478	0.974	11.634	0.810	0.909	0.771	1.000

253 baselines: LLaMA [44], Vicuna [7], Mol-Instructions [12], and InstructMol [6]. The metrics used
 254 to evaluate molecule generation tasks include Exact Match, BLEU [37], Levenshtein Distance [22],
 255 RDKit Fingerprint Similarity [20], MACCS Fingerprint Similarity [10], and Morgan Fingerprint
 256 Similarity [36]. These metrics evaluate structural similarity between generated and target molecules,
 257 along with Validity [19], which assesses the proportion of chemically valid molecules generated. We
 258 utilize the Mol-Instructions dataset to evaluate the generation capabilities of UniMoT, and the results
 259 are presented in Table 4.

260 As the baselines generate SMILES strings and then convert them to molecules, UniMoT directly
 261 leverages the generated molecule tokens and obtains their embeddings from the learned codebook.
 262 These embeddings can be decoded to desired molecules through the pretrained adapter and SMILES
 263 decoder. Regarding the results in Table 4, UniMoT exhibits the capability to generate valid molecules
 264 with a higher degree of similarity to the target molecules compared to the baselines. UniMoT can
 265 generate molecules as if they were text, demonstrating strong generation capabilities and providing a
 266 new perspective to molecule generation tasks.

Table 5: Ablation study on the projector and representation form for the molecule captioning task using the PubChem dataset.

Projector	Input to LLM	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
Projection Layer	Molecule Emb.	19.3	12.1	27.9	12.3	18.1	21.5
Q-Former	Query Emb.	28.6	21.3	36.2	21.4	29.7	32.6
Causal Q-Former	Causal Emb.	32.8	25.2	39.2	24.8	35.3	36.5
Causal Q-Former	Causal Tokens	31.3	23.8	37.5	23.7	33.6	34.8

Table 6: Ablation study on the model size and tuning strategy for the molecule captioning task using the PubChem dataset.

Model Size	Tuning	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR
LLaMA2-7B	LoRA Tuning	31.3	23.8	37.5	23.7	33.6	34.8
LLaMA2-7B	Fully Tuning	32.0	24.6	38.3	24.3	34.7	35.6
LLaMA2-13B	LoRA Tuning	31.8	24.3	38.0	24.1	34.4	35.3

267 4.3 Ablation Studies

268 **Cross-Modal Projector.** We conducted an ablation study on the cross-modal projector, with the
 269 results on the molecule captioning task shown in Table 5. The linear projection demonstrated the worst
 270 performance, indicating that the molecule features lack textual information, thus hindering effective
 271 molecule-text interactions and alignment. Additionally, we compared the performance of a Q-Former
 272 with bidirectional self-attention to a Causal Q-Former with causal self-attention. The results show
 273 that query embeddings with causal dependency outperform those with bidirectional dependency. This
 274 demonstrates that input with left-to-right causal dependency aligns with the unidirectional attention
 275 mechanism in LLMs, leading to improved performance.

276 **Discrete vs. Continuous Representation.** We compare the performance of continuous causal query
 277 embeddings and discrete tokens quantized from causal embeddings as inputs to LLMs. As shown in
 278 Table 5, continuous embeddings demonstrate better performance than discrete tokens in understanding
 279 molecules. This result is reasonable since the quantization process causes information loss in discrete
 280 tokens. However, we still use discrete token representation to facilitate the autoregressive training
 281 paradigm of LLMs, which supports the unification of comprehension and generation tasks. To achieve
 282 this unification, we unavoidably sacrifice some performance in comprehension tasks.

283 **Model Size and Tuning Strategy.** We conducted a comparison of molecule captioning performance
 284 across various model sizes and tuning strategies, as illustrated in Table 6. Our findings indicate that
 285 scaling up the LLM to 13B or adopting a fully tuning strategy yields only marginal improvements
 286 in performance compared to using LLaMA2-7B with LoRA tuning. While larger models and fully
 287 tuning strategies might offer slight gains in performance, they come at a significant cost in terms of
 288 efficiency. Considering the trade-off between achieving high performance and maintaining efficiency,
 289 we have chosen to utilize LLaMA2-7B with LoRA tuning in our experiments. This ensures that our
 290 model remains both powerful and practical.

291 5 Conclusion

292 This work introduces UniMoT, an innovation in the field of molecular-textual understanding and
 293 generation, which has successfully unified these two distinct modalities under a single, coherent
 294 framework. By integrating a Vector Quantization-driven tokenizer with a Causal Q-Former, UniMoT
 295 overcomes previous architectural limitations where molecule and text modalities were not treated
 296 equally, lacking a dedicated supervision signal for the molecular domain. This unique tokenizer
 297 transforms molecules into sequences of discrete tokens, embedding high-level molecular and textual
 298 information cohesively. Moreover, by employing a four-stage training scheme, UniMoT has emerged
 299 as a versatile multi-modal LLM, adept at handling molecule-to-text and text-to-molecule tasks.
 300 Extensive empirical evaluations demonstrate that UniMoT attains state-of-the-art performance across
 301 a diverse array of molecule comprehension and generation tasks.

References

- 302
- 303 [1] Satyanjee Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved
304 correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation*
305 *measures for machine translation and/or summarization*, pages 65–72, 2005.
- 306 [2] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv*
307 *preprint arXiv:1903.10676*, 2019.
- 308 [3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through
309 stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.
- 310 [4] Yoann Boget, Magda Gregorova, and Alexandros Kalousis. Vector-quantized graph auto-encoder. *arXiv*
311 *preprint arXiv:2306.07735*, 2023.
- 312 [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind
313 Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners.
314 *Advances in neural information processing systems*, 33:1877–1901, 2020.
- 315 [6] He Cao, Zijing Liu, Xingyu Lu, Yuan Yao, and Yu Li. Instructmol: Multi-modal integration for building a
316 versatile and reliable molecular assistant in drug discovery. *arXiv preprint arXiv:2311.16208*, 2023.
- 317 [7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan
318 Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4
319 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6, 2023.
- 320 [8] Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised
321 pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.
- 322 [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirec-
323 tional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- 324 [10] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys
325 for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.
- 326 [11] Carl Edwards, Tuan Lai, Kevin Ros, Garrett Honke, Kyunghyun Cho, and Heng Ji. Translation between
327 molecules and natural language. *arXiv preprint arXiv:2204.11817*, 2022.
- 328 [12] Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and
329 Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models.
330 *arXiv preprint arXiv:2306.08018*, 2023.
- 331 [13] Robert Gray. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29, 1984.
- 332 [14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and
333 Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*,
334 2021.
- 335 [15] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec.
336 Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.
- 337 [16] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec.
338 Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.
- 339 [17] Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained
340 transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.
- 341 [18] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A
342 Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–
343 D1380, 2023.
- 344 [19] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In
345 *International conference on machine learning*, pages 1945–1954. PMLR, 2017.
- 346 [20] Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.
- 347 [21] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo
348 Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining.
349 *Bioinformatics*, 36(4):1234–1240, 2020.

- 350 [22] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In
351 *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.
- 352 [23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training
353 with frozen image encoders and large language models. In *International conference on machine learning*,
354 pages 19730–19742. PMLR, 2023.
- 355 [24] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training
356 for unified vision-language understanding and generation. In *International conference on machine learning*,
357 pages 12888–12900. PMLR, 2022.
- 358 [25] Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and
359 Qi Tian. Towards 3d molecule-text interpretation in language models. *arXiv preprint arXiv:2401.13923*,
360 2024.
- 361 [26] Youwei Liang, Ruiyi Zhang, Li Zhang, and Pengtao Xie. Drugchat: towards enabling chatgpt-like
362 capabilities on drug molecule graphs. *arXiv preprint arXiv:2309.03907*, 2023.
- 363 [27] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches
364 out*, pages 74–81, 2004.
- 365 [28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural
366 information processing systems*, 36, 2024.
- 367 [29] Pengfei Liu, Yiming Ren, Jun Tao, and Zhixiang Ren. Git-mol: A multi-modal large language model for
368 molecular science with graph, image, and text. *Computers in Biology and Medicine*, 171:108073, 2024.
- 369 [30] Shengchao Liu, Mehmet F Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation
370 for graphs, with applications to molecules. *Advances in neural information processing systems*, 32, 2019.
- 371 [31] Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao,
372 and Animashree Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and
373 editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023.
- 374 [32] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training
375 molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.
- 376 [33] Zhiyuan Liu, Sihang Li, Yanchen Luo, Hao Fei, Yixin Cao, Kenji Kawaguchi, Xiang Wang, and Tat-Seng
377 Chua. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter.
378 *arXiv preprint arXiv:2310.12798*, 2023.
- 379 [34] Yizhen Luo, Kai Yang, Massimo Hong, Xingyi Liu, and Zaiqing Nie. Molfm: A multimodal molecular
380 foundation model. *arXiv preprint arXiv:2307.09484*, 2023.
- 381 [35] Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. Biomedgpt:
382 Open multimodal generative pre-trained transformer for biomedicine. *arXiv preprint arXiv:2308.09442*,
383 2023.
- 384 [36] Harry L Morgan. The generation of a unique machine description for chemical structures—a technique
385 developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.
- 386 [37] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation
387 of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational
388 Linguistics*, pages 311–318, 2002.
- 389 [38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou,
390 Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer.
391 *Journal of machine learning research*, 21(140):1–67, 2020.
- 392 [39] Bing Su, Dazhao Du, Zhao Yang, Yujie Zhou, Jiangmeng Li, Anyi Rao, Hao Sun, Zhiwu Lu, and Ji-Rong
393 Wen. A molecular multimodal foundation model associating molecule graphs with natural language. *arXiv
394 preprint arXiv:2209.05481*, 2022.
- 395 [40] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-
396 supervised graph-level representation learning via mutual information maximization. *arXiv preprint
397 arXiv:1908.01000*, 2019.
- 398 [41] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt:
399 Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*, 2023.

- 400 [42] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang,
401 and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.
- 402 [43] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia,
403 Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv*
404 *preprint arXiv:2211.09085*, 2022.
- 405 [44] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix,
406 Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation
407 language models (2023). *arXiv preprint arXiv:2302.13971*, 2023.
- 408 [45] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural*
409 *information processing systems*, 30, 2017.
- 410 [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz
411 Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*,
412 30, 2017.
- 413 [47] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. Smiles-bert: large scale
414 unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international*
415 *conference on bioinformatics, computational biology and health informatics*, pages 429–436, 2019.
- 416 [48] Y Wang, J Wang, Z Cao, and AB Farimani. Molclr: Molecular contrastive learning of representations via
417 graph neural networks. arxiv 2021. *arXiv preprint arXiv:2102.10056*, 2021.
- 418 [49] Zichao Wang, Weili Nie, Zhuoran Qiao, Chaowei Xiao, Richard Baraniuk, and Anima Anandkumar.
419 Retrieval-based controllable molecule generation. *arXiv preprint arXiv:2208.11126*, 2022.
- 420 [50] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology
421 and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.
- 422 [51] Jiawei Wu, Mingyuan Yan, and Dianbo Liu. Vqsynergy: Robust drug synergy prediction with vector
423 quantization mechanism. *arXiv preprint arXiv:2403.03089*, 2024.
- 424 [52] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm.
425 *arXiv preprint arXiv:2309.05519*, 2023.
- 426 [53] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu,
427 Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical*
428 *science*, 9(2):513–530, 2018.
- 429 [54] Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z Li.
430 Mole-bert: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International*
431 *Conference on Learning Representations*, 2022.
- 432 [55] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with
433 parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*, 2023.
- 434 [56] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive
435 learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.
- 436 [57] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu,
437 Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint*
438 *arXiv:2110.04627*, 2021.
- 439 [58] Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. A deep-learning system bridging molecule structure
440 and biomedical text with comprehension comparable to human professionals. *Nature communications*,
441 13(1):862, 2022.
- 442 [59] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan,
443 Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv*
444 *preprint arXiv:2402.12226*, 2024.
- 445 [60] Xiang Zhuang, Qiang Zhang, Keyan Ding, Yatao Bian, Xiao Wang, Jingsong Lv, Hongyang Chen, and
446 Huajun Chen. Learning invariant molecular representation in latent discrete space. *Advances in Neural*
447 *Information Processing Systems*, 36, 2024.

448 A Details of Causal Q-Former

449 The Q-Former operates as a query-based transformer that utilizes learnable query vectors to interact
 450 with molecule features extracted by a frozen encoder. These queries are essential for extracting rele-
 451 vant information from the molecule features. The Q-Former comprises both a molecule transformer
 452 and a text transformer, sharing self-attention layers. The text transformer architecture is based on
 453 BERT [9], while the molecule transformer incorporates cross-attention layers between self-attention
 454 and feed-forward layers. Q-Former employs a cross-attention mechanism where the query vectors
 455 selectively attend to different aspects of the molecule features, allowing the model to capture critical
 456 details necessary for understanding and generating textual descriptions of molecular properties.

457 Specifically, we incorporate causal masks into the queries, ensuring that they only interact with
 458 preceding queries. This ensures the sequence of query embeddings maintains a causal dependency,
 459 aligning with the requirements of LLMs operating on text sequence. The Causal Q-Former is
 460 illustrated in Figure 4. We employ the Causal Q-Former to generate causal query embeddings
 461 $\mathbf{Z} = \{z_i\}_{i=1}^M \in \mathbb{R}^{M \times d}$ containing high-level molecular and textual information, where M represents
 462 the number of queries and d denotes the dimension of query embeddings. Next, we introduce three
 463 tailored objectives MTC, MTM, and MTG for the pretraining of the Causal Q-Former.

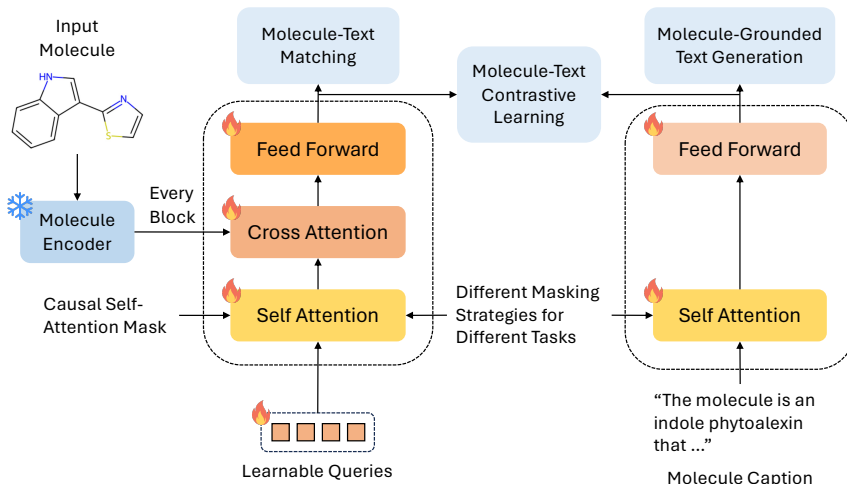


Figure 4: Illustration of our proposed Causal Q-Former. The Causal Q-Former provides causal query embeddings for subsequent blocks.

464 **Molecule-Text Contrastive Learning (MTC)** aims to align molecule and text features by maximizing
 465 their mutual information. This is achieved by maximizing the molecule-text similarity of positive
 466 pairs against that of negative pairs. We utilize the last query embedding z_M of the query sequence
 467 $\{z_i\}_{i=1}^M$ as the query representation, since the output query sequence is causal and the last embedding
 468 contains global information from the queries. For text representation, we use the output embedding
 469 of the [CLS] token, denoted as \mathbf{y} . The contrastive learning loss is expressed as follows:

$$\mathcal{L}_{\text{MTC}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp((z_M^i)^T \mathbf{y}^i / \tau)}{\sum_{j=1}^B \exp((z_M^i)^T \mathbf{y}^j / \tau)} - \frac{1}{B} \sum_{i=1}^B \log \frac{\exp((\mathbf{y}^i)^T z_M^i / \tau)}{\sum_{j=1}^B \exp((\mathbf{y}^i)^T z_M^j / \tau)}, \quad (4)$$

470 where B denotes the batch size, and τ represents the temperature parameter. Here, z_M^i and \mathbf{y}^i refer
 471 to the i -th query and text representations in a batch, respectively.

472 **Molecule-Text Matching (MTM)** focuses on learning fine-grained alignment between molecule and
 473 text features. As query embeddings $\mathbf{Z} = \{z_i\}_{i=1}^M$ capture both molecular and textual information
 474 through cross-attention and self-attention layers respectively, we utilize the last query embedding z_M
 475 as input to a binary classifier. This classifier predicts whether a given molecule-text pair is matched
 476 or unmatched. The corresponding loss function is formulated as follows:

$$\mathcal{L}_{\text{MTM}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\phi(z_M | \mathbf{X}^i, \mathbf{t}^i))}{\sum_{j=1}^B \exp(\phi(z_M | \mathbf{X}^i, \mathbf{t}^j)) + \sum_{j=1}^B \exp(\phi(z_M | \mathbf{X}^j, \mathbf{t}^i))}, \quad (5)$$

477 where ϕ represents a binary classifier, and \mathbf{X}^i and t^i denote the i -th input molecule features and input
478 text in a batch, respectively.

479 **Molecule-grounded Text Generation (MTG)** focuses on generating textual descriptions given
480 a molecule input. In this task, causal masks for queries are not applied since only textual output
481 is required. However, causal masks are applied for text, allowing each text token to attend to its
482 preceding text tokens and all queries, but not subsequent tokens. The Language Modeling (LM)
483 loss function is applied to model the generation of text t^i conditioned on the molecule input \mathbf{X}^i ,
484 formulated as:

$$\mathcal{L}_{\text{MTG}} = -\frac{1}{B} \sum_{i=1}^B \sum_{j=1}^L \log p(t_j^i | t_1^i, \dots, t_{j-1}^i, \mathbf{X}^i), \quad (6)$$

485 where t_j^i represents the j -th token in the text sequence t^i . Here, \mathbf{X}^i and t^i denote the i -th input
486 molecule features and generated text in a batch, respectively.

487 The total loss for training the Q-Former encompasses the three aforementioned objectives:

$$\mathcal{L}_{\text{Q-Former}} = \mathcal{L}_{\text{MTC}} + \mathcal{L}_{\text{MTM}} + \mathcal{L}_{\text{MTG}}. \quad (7)$$

488 B Details of Datasets

489 This section provides detailed information about the datasets used in evaluating the performance of
490 UniMoT across various tasks. The datasets are utilized for molecular property prediction, molecule
491 captioning, molecule-text retrieval, and molecule generation tasks. Each dataset serves a unique
492 purpose in assessing different capabilities of the model.

493 We present the details of the Molecular Property Prediction Datasets below:

- 494 • **BBBP** [53]: The Blood-Brain Barrier Penetration dataset predicts the ability of molecules to
495 penetrate the blood-brain barrier.
- 496 • **Tox21** [53]: This dataset is part of the Toxicology in the 21st Century initiative, used for toxicity
497 prediction.
- 498 • **ToxCast** [53]: Another toxicity prediction dataset with a broader range of biological assays.
- 499 • **Sider** [53]: Side Effect Resource database, used for predicting drug side effects.
- 500 • **ClinTox** [53]: Clinical Toxicity dataset for predicting clinical trial toxicity outcomes.
- 501 • **MUV** [53]: Maximum Unbiased Validation dataset for virtual screening.
- 502 • **HIV** [53]: Human Immunodeficiency Virus dataset for predicting anti-HIV activities.
- 503 • **BACE** [53]: Beta-Secretase 1 dataset for predicting inhibitors of the BACE-1 enzyme, relevant
504 for Alzheimer’s research.
- 505 • **QM9** [12]: The quantum mechanics properties dataset, where the objective is to predict key
506 quantum mechanics properties of a given molecule, such as HUMO, LUMO, and the HUMO-
507 LUMO gap.

508 We present the details of the Molecule Captioning Datasets below:

- 509 • **PubChem** [18]: A large dataset of chemical molecules used for generating textual descriptions
510 of molecular structures.
- 511 • **ChEBI-20** [11]: A subset of the Chemical Entities of Biological Interest database, provides
512 structured and detailed descriptions of molecules, enhancing the model’s ability to generate
513 accurate captions.

514 We present the details of the Molecule-Text Retrieval Datasets below:

- 515 • **PubChem** [18]: Used for both molecule-to-text (M2T) and text-to-molecule (T2M) retrieval
516 tasks.
- 517 • **PCdes** [58]: Another dataset for evaluating M2T and T2M retrieval accuracy.
- 518 • **MoMu** [39]: Dataset specifically designed for molecule-text interactions and retrieval tasks.

Table 7: Summary of datasets, their types, tasks, descriptions, URLs, and licenses used for evaluating UniMoT.

Dataset	Type	Tasks	Description	URL	License
BBBP	Classification	Molecular Property Prediction	Predicts blood-brain barrier penetration ability.	BBBP URL	CC-BY 4.0
Tox21	Classification	Molecular Property Prediction	Toxicity prediction using the Tox21 initiative data.	Tox21 URL	Public Domain
ToxCast	Classification	Molecular Property Prediction	Broad toxicity prediction with various biological assays.	ToxCast URL	Public Domain
Sider	Classification	Molecular Property Prediction	Predicts drug side effects.	Sider URL	CC-BY 4.0
ClinTox	Classification	Molecular Property Prediction	Clinical trial toxicity prediction.	ClinTox URL	Public Domain
MUV	Classification	Molecular Property Prediction	Virtual screening for unbiased validation.	MUV URL	CC-BY 4.0
HIV	Classification	Molecular Property Prediction	Predicts anti-HIV activity of molecules.	HIV URL	Public Domain
BACE	Classification	Molecular Property Prediction	Predicts inhibitors of the BACE-1 enzyme.	BACE URL	Public Domain
QM9	Regression	Molecular Property Prediction	Predicts various molecular properties such as atomization energy, dipole moment, etc.	QM9 URL	CC-BY 4.0
PubChem	Captioning, Retrieval	Molecule Captioning, Molecule-Text Retrieval	Generates descriptions and retrieves text/molecules based on input molecules/text.	PubChem URL	Public Domain
ChEBI-20	Captioning	Molecule Captioning	Generates detailed descriptions of molecular structures.	ChEBI-20 URL	CC-BY 4.0
PCdes	Retrieval	Molecule-Text Retrieval	Used for evaluating accuracy in molecule-text retrieval tasks.	PCdes URL	CC-BY 4.0
MoMu	Retrieval	Molecule-Text Retrieval	Dataset for molecule-text interaction and retrieval evaluation.	MoMu URL	CC-BY 4.0
Mol-Instructions	Generation	Molecule Generation	Includes tasks such as molecule generation from descriptions, reagent prediction, etc.	Mol-Instructions URL	CC-BY 4.0

519 We present the details of the Molecule Generation Datasets below:

- 520 • **Mol-Instructions** [12]: This dataset includes tasks such as caption-guided molecule generation,
521 reagent prediction, forward reaction prediction, and retrosynthesis. It is used to evaluate the
522 model’s ability to generate molecular structures based on textual descriptions and other related
523 tasks.

524 We summarize the datasets used for evaluating UniMoT in Table 7. It encompasses various types
525 of datasets, including those for classification, regression, captioning, retrieval, and generation tasks.
526 Each dataset is described in terms of its type, tasks it supports, a brief description of its content, its
527 URL for access, and the license under which it is distributed. The licenses vary, with some datasets
528 being in the public domain and others under CC-BY 4.0 license.

529 C Details of Training

530 **Stage-1: Causal Q-Former Pretraining.** During Stage-1, we only connect the molecule encoder
531 and the Causal Q-Former, leaving out other blocks. We leverage the pretrained molecule encoder from
532 MoleculeSTM [31], which has undergone extensive contrastive learning with molecule-text pairs.
533 We utilize the PubChem dataset [18] for pretraining, keeping the molecule encoder frozen while
534 updating only the Causal Q-Former. Both queries and text serve as input to the Causal Q-Former,
535 while only queries serve as input in subsequent stages. Inspired by BLIP-2 [23], we employ three
536 tailored objectives – Molecule-Text Contrastive Learning (MTC), Molecule-Text Matching (MTM),
537 and Molecule-grounded Text Generation (MTG) – for the pretraining of the Causal Q-Former, as
538 detailed in Appendix A.

539 The dimension of molecule features is set to 300. We use 16 queries, each with a dimension of 768.
540 The size of \mathbf{Z} (16×768) is much smaller than the size of molecule features \mathbf{X} (e.g., 150×300). The
541 Q-former is pretrained for 50 epochs. We adopt the AdamW optimizer with a weight decay of 0.05,
542 and a cosine decay learning rate scheduler, with a minimal learning rate of $1e-5$. The batch size is set
543 to 64. The computational overhead for this pretraining is 20 GPU hours on 4 NVIDIA A100 GPUs.

544 **Stage-2: Molecule Tokenizer Pretraining.** We connect the Causal Q-Former with the subsequent
545 blocks and train the molecule tokenizer using the objective defined in Equation (2). Following
546 the approach of RetMol [49], we utilize SMILES strings [50] to represent molecules, and employ the
547 pretrained ChemFormer [17] as the generative model. Specifically, we leverage the SMILES
548 encoder and SMILES decoder components provided by ChemFormer. We utilize PubChem [18]
549 and CheBI-20 [11] datasets, keeping the molecule encoder, SMILES encoder, and SMILES decoder
550 frozen, while updating the Causal Q-Former, codebook, and adapter. Once optimized, the molecule
551 tokenizer remains unchanged throughout the subsequent stages.

552 The molecule codebook size is set to $K = 2048$, and the dimension of codebook embedding is 768.
553 The tokenizer is pretrained for 50 epochs. We adopt the AdamW optimizer with a weight decay of
554 0.05, and a cosine decay learning rate scheduler, with a minimal learning rate of $1e-5$. The batch size
555 is set to 64. The computational overhead for this pretraining is 40 GPU hours on 4 NVIDIA A100
556 GPUs.

557 **Stage-3: Unified Molecule-Text Pretraining.** We connect the molecule tokenizer with the LLM
558 and employ the LM objective defined in Equation (3) to pretrain the LLM. We utilize LLaMA [44] as
559 the default LLM. To construct the unified molecule-text vocabulary, we merge 2048 molecule codes
560 with the original text vocabulary. Pretraining the LLM involves molecule-to-text autoregression
561 and text-to-molecule autoregression, aimed at enhancing UniMoT’s multi-modal comprehension
562 and generation capabilities. We utilize datasets PubChem [18], CheBI-20 [11], PCdes [58], and
563 MoMu [39] for this purpose. To enhance efficiency, we train the LLM using LoRA tuning [14].

564 The multi-modal LLM is pretrained for 10 epochs. We adopt the AdamW optimizer with a weight
565 decay of 0.05, and a cosine decay learning rate scheduler, with a minimal learning rate of $1e-5$. The
566 batch size is set to 32. The computational overhead for this pretraining is 50 GPU hours on 4 NVIDIA
567 A100 GPUs. To reduce CUDA memory usage, we integrate LoRA with the parameters set to $r = 8$,
568 $\alpha = 32$, and dropout = 0.1. This integration is applied to the `k_proj`, `v_proj`, `q_proj`, and `o_proj`
569 modules.

Table 8: Instruction samples for comprehension and generation tasks: molecular property prediction, molecule captioning, molecule-text retrieval, caption-guided molecule generation, reagent prediction, forward reaction prediction, and retrosynthesis.

Task	Instruction
Molecular Property Prediction (Regression)	Instruction: <i>Could you give me the LUMO energy value of this molecule?</i> (Optional: The SMILES sequence is: SMILES) Output: <i>0.0576.</i>
Molecular Property Prediction (Classification)	Instruction: <i>Evaluate whether the given molecule is able to enter the blood-brain barrier.</i> (Optional: The SMILES sequence is: SMILES) Output: <i>Yes.</i>
Molecule Captioning	Instruction: <i>Could you give me a brief overview of this molecule?</i> (Optional: The SMILES sequence is: SMILES) Output: <i>The molecule is an indole phytoalexin that ...</i>
Molecule-Text Retrieval	Instruction: <i>Retrieve relevant text for the given molecule.</i> (Optional: The SMILES sequence is: SMILES) Output: <i>The molecule is associated with ...</i>
Caption-Guided Molecule Generation	Instruction: <i>Create a molecule with the structure as described: The molecule is a primary arylamine that ...</i> Output: <i>SMILES of the molecule.</i>
Reagent Prediction	Instruction: <i>Please provide possible reagents based on the following chemical reaction.</i> <REACTANT A> <REACTANT B> ... > <PRODUCTs> Output: <i>SMILES of the reagents.</i>
Forward Reaction Prediction	Instruction: <i>With the provided reactants and reagents, propose a potential product:</i> <REACTANT A> <REACTANT B> ... <REAGENT A> <REAGENT B> ... Output: <i>SMILES of the products.</i>
Retrosynthesis	Instruction: <i>Please suggest potential reactants used in the synthesis of the product:</i> <PRODUCTs> Output: <i>SMILES of the reactants and reagents.</i>

570 **Stage-4: Task-Specific Instruction Tuning.** We perform instruction tuning to align UniMoT with
 571 human instructions through supervised fine-tuning on seven tasks: molecular property prediction,
 572 molecule captioning, molecule-text retrieval, caption-guided molecule generation, reagent prediction,
 573 forward reaction prediction, and retrosynthesis. For the molecular property prediction task, we
 574 utilize the quantum mechanics properties dataset [12] for regression prediction and the MoleculeNet
 575 datasets [53] for property classification. For the molecule captioning and molecule-text retrieval
 576 tasks, we employ datasets PubChem [18], CheBI-20 [11], PCdes [58], and MoMu [39]. For the
 577 remaining tasks, we utilize the Mol-Instructions dataset [12] to conduct instruction tuning. We
 578 fine-tune UniMoT for 10 epochs on each task using the same optimizer, learning rate scheduler, and
 579 LoRA configurations as in Stage-3 pretraining. Instruction samples for comprehension and generation
 580 tasks are shown in Table 8.

581 We have summarized the detailed training hyperparameters of UniMoT in Table 9.

582 D Details and More Results of Experiments

583 **Molecular Property Prediction Task.** Property prediction aims to anticipate a molecule’s intrinsic
 584 physical and chemical properties based on its structural or sequential characteristics. In the regression
 585 task, we conduct experiments on the quantum mechanics properties dataset QM9 [12], where the
 586 objective is to predict key quantum mechanics properties of a given molecule, such as HUMO, LUMO,
 587 and the HUMO-LUMO gap. We compare UniMoT against several baselines, including Alpaca [42],
 588 Baize [55], LLaMA2-7B [44], Vicuna-13B [7], Mol-Instructions [12], and InstructMol [6]. Mean
 589 Absolute Error (MAE) serves as our evaluation metric. The performance of the regression task on the
 590 QM9 dataset is presented in Table 10. Compared to previous single-modal instruction-tuned LLMs
 591 and molecular LLMs, UniMoT exhibits further improvement on the regression task, showcasing its
 592 fundamental comprehension abilities in molecular contexts.

593 **Molecule Captioning Task.** The molecule captioning task involves generating a comprehensive
 594 description of a molecule. For this task, we compare UniMoT with several baselines: MolT5 [11],

Table 9: The detailed training hyperparameters of UniMoT.

Configuration	Q-Former Pretraining	Tokenizer Pretraining	LLM Pretraining
Molecule Encoder	MoleculeSTM	MoleculeSTM	MoleculeSTM
SMILES Encoder	-	ChemFormer	ChemFormer
SMILES Decoder	-	ChemFormer	ChemFormer
LLM Base	-	-	LLaMA2-7B
Epoch	50	50	10
Optimizer	AdamW	AdamW	AdamW
Codebook Size	2048	2048	2048
Number of Queries	16	16	16
Query Embedding Dim.	768	768	768
Molecule Embedding Dim.	300	300	300
Batch Size	64	64	32
Minimal Learning Rate	1e-5	1e-5	1e-5
Learning Rate Scheduler	Cosine	Cosine	Cosine
Warm-up Steps	1000	1000	1000
Weight Decay	0.05	0.05	0.05
LoRA Config	-	-	$r = 8, \alpha = 32, \text{dropout} = 0.1$
Precision	bfloat16	bfloat16	bfloat16
GPU Usage	4 NVIDIA A100	4 NVIDIA A100	4 NVIDIA A100
Training Time	20 GPU hours	40 GPU hours	50 GPU hours

Table 10: Mean Absolute Error (MAE) of molecular property prediction task (regression) on the QM9 dataset. **Bold** indicates the best performance and underline indicates the second best performance. $\Delta\epsilon$ is the HOMO-LUMO energy gap.

Model	HOMO↓	LUMO↓	$\Delta\epsilon$ ↓	AVG↓
Alpaca (LLaMA-7B) [42]	-	-	-	322.109
Baize (LLaMA-7B) [55]	-	-	-	261.343
LLaMA2-7B [44]	0.7367	0.8641	0.5152	0.7510
Vicuna-13B [7]	0.7135	3.6807	1.5407	1.9783
Mol-Instructions (LLaMA-7B) [12]	0.0210	0.0210	0.0203	0.0210
InstructMol (Vicuna-7B) [6]	<u>0.0048</u>	<u>0.0050</u>	<u>0.0061</u>	<u>0.0050</u>
UniMoT (LLaMA2-7B)	0.0042	0.0047	0.0055	0.0049

595 MoMu [39], InstructMol [6], MolCA [33], and 3D-MoLM [25]. We adopt BLEU [37], ROUGE [27],
596 and METEOR [1] as the evaluation metrics. The performance of UniMoT in the molecule captioning
597 task on the CheBI-20 dataset is presented in Table 11. Some concrete examples of molecule captioning
598 task are presented in Table 12. From the results, it is evident that UniMoT consistently outperforms
599 the baselines by a significant margin. These results underscore the effectiveness of the molecule
600 tokenizer in providing molecule tokens with high-level molecular and textual information, thus
601 enhancing molecule comprehension.

602 **Molecule-Text Retrieval Task.** The molecule-text retrieval task involves using a molecule to
603 retrieve text (M2T) and using text to retrieve a molecule (T2M). We compare UniMoT with several
604 baselines: Sci-BERT [2], KV-PLM [58], MoMu [39], MoleculeSTM [31], MolCA [33], and 3D-
605 MoLM [25]. We report the performance of retrieval using a batch of 64 random samples and the
606 entire test set, evaluated with the metrics of Accuracy and Recall@20. We use the checkpoint
607 from Stage-1 of pretraining. Performance on the PCdes and MoMu datasets is shown in Table 13.
608 UniMoT demonstrates superior performance over the baselines on molecule-text retrieval, particularly
609 in molecule-to-text retrieval. This demonstrates that UniMoT has learned fine-grained alignment
610 between molecules and text, and it can understand molecule-text interactions through the introduction
611 of the Causal Q-Former.

612 **Molecule Generation Tasks.** Molecule generation tasks include caption-guided molecule genera-
613 tion, reagent prediction, forward reaction prediction, and retrosynthesis.

Table 11: Performance (%) of molecule captioning task on the CheBI-20 dataset. **Bold** indicates the best performance and underline indicates the second best performance.

Model	BLEU-2 \uparrow	BLEU-4 \uparrow	ROUGE-1 \uparrow	ROUGE-2 \uparrow	ROUGE-L \uparrow	METEOR \uparrow
T5-Small [38]	50.1	41.5	60.2	44.6	54.5	53.2
T5-Base [38]	51.1	42.3	60.7	45.1	55.0	53.9
T5-Large [38]	55.8	46.7	63.0	47.8	56.9	58.6
MolT5-Small (T5-Small) [11]	51.9	43.6	62.0	46.9	56.3	55.1
MolT5-Base (T5-Base) [11]	54.0	45.7	63.4	48.5	57.8	56.9
MolT5-Large (T5-Large) [11]	59.4	50.8	65.4	51.0	59.4	61.4
MoMu-Small (T5-Small) [39]	53.2	44.5	-	-	56.4	55.7
MoMu-Base (T5-Base) [39]	54.9	46.2	-	-	57.5	57.6
MoMu-Large (T5-Large) [39]	59.9	51.5	-	-	59.3	59.7
InstructMol (Vicuna-7B) [6]	47.5	37.1	56.6	39.4	50.2	50.9
MolCA (OPT-125M) [33]	61.6	52.9	67.4	53.3	61.5	63.9
MolCA (OPT-1.3B) [33]	<u>63.9</u>	<u>55.5</u>	<u>69.7</u>	<u>55.8</u>	<u>63.6</u>	<u>66.9</u>
UniMoT (LLaMA2-7B)	66.4	58.3	72.2	58.4	66.4	70.3

- 614 • Caption-guided molecule generation involves creating molecular structures from textual descriptions, leveraging NLP and cheminformatics to interpret and translate descriptions into chemical structures.
- 615
- 616
- 617 • Reagent prediction focuses on identifying suitable reagents for given reactants and desired products, optimizing synthetic routes.
- 618
- 619 • Forward reaction prediction forecasts probable products from specific reactants and reagents, using knowledge of chemical reactivity.
- 620
- 621 • Retrosynthesis deconstructs target molecules into simpler starting materials.

622 In molecule generation tasks, evaluating the quality of generated molecules involves several metrics that measure different aspects of similarity and validity.

- 624 • Exact Match checks if the generated molecule is identical to the target molecule, offering a stringent criterion for precise replication but potentially overlooking chemically similar variants.
- 625
- 626 • The BLEU score [37], adapted from machine translation, measures the overlap of n-grams (short sequences of atoms or bonds) between generated and target molecules, thus assessing partial similarities.
- 627
- 628
- 629 • Levenshtein Distance [22] evaluates the minimum number of edits needed to transform the generated molecule into the target, providing insight into structural changes required.
- 630
- 631 • RDKit [20], MACCS [10], and Morgan [36] Fingerprint Similarities compare the generated and target molecules based on various molecular fingerprinting methods, which capture different aspects of molecular structure and properties.
- 632
- 633
- 634 • The Validity [19] metric assesses the proportion of chemically valid molecules generated, ensuring that the output consists of plausible chemical structures.
- 635

636 Together, these metrics offer a comprehensive evaluation framework, balancing exact matches with structural and chemical validity.

638 E Limitations

639 While UniMoT demonstrates considerable advancements in unifying molecule and text modalities for comprehensive understanding and generation tasks, several limitations must be acknowledged. 640 Although UniMoT exhibits strong performance in molecule-to-text and text-to-molecule tasks, it has 641 not been extensively tested on more complex molecule generation tasks such as molecule editing, 642 which require precise modifications to molecular structures. Future work could explore extending 643 UniMoT’s capabilities to handle such sophisticated molecular manipulations. 644

645 Due to the scarcity of annotated data in the molecular field, the training of UniMoT is less extensive 646 compared to fields like computer vision. This limitation restricts the model’s ability to fully learn and 647 generalize from diverse molecular structures and properties. In contrast, the visual domain benefits

Table 12: Examples of molecule captioning task on the ChEBI-20 dataset. We highlight in blue the text that accurately describes the molecule structures in the generated caption, ensuring alignment with the ground truth.

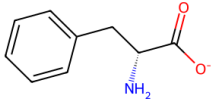
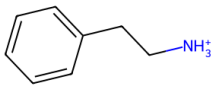
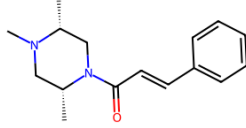
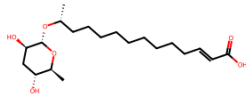
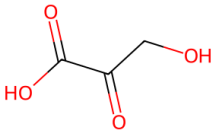
Molecule	Generated Molecule Caption	Ground Truth
	The molecule is an optically active form of phenylalaninate having D-configuration. It is a conjugate base of a D-phenylalanine . It is an enantiomer of a L-phenylalaninate .	The molecule is the D-enantiomer of phenylalaninate. It is a conjugate base of a D-phenylalanine . It is an enantiomer of a L-phenylalaninate .
	The molecule is an ammonium ion that is the conjugate acid of 2-phenylpropylamine arising from protonation of the primary amino function; major species at pH 7.3. It has a role as a human metabolite, an Escherichia coli metabolite and a mouse metabolite. It is a conjugate acid of a 2-phenylpropylamine .	The molecule is the cation obtained by protonation of the amino group of 2-phenylethylamine . It has a role as a human metabolite and an Escherichia coli metabolite. It is a conjugate acid of a 2-phenylethylamine .
	The molecule is an enamide obtained by the carboxy group of trans-cinnamic acid with the secondary amino group of (2S,5R)-1,2,5-trimethylpiperazine. It has a role as an Aspergillus metabolite . It is an alkaloid, a N-acylpiperazine, an enamide and a tertiary carboxamide. It derives from a trans-cinnamic acid .	The molecule is an enamide obtained by formal condensation of the carboxy group of trans-cinnamic acid with the secondary amino group of (2R,5R)-1,2,5-trimethylpiperazine. It has a role as an Aspergillus metabolite . It is a N-acylpiperazine, a N-alkylpiperazine, an alkaloid, an enamide and a tertiary carboxamide. It derives from a trans-cinnamic acid .
	The molecule is an (omega-1)- hydroxy fatty acid ascaroside obtained by formal condensation of the alcoholic hydroxy group of (10R)- 10-hydroxyundecanoic acid with ascaropyranose (the alpha anomer). It is a metabolite of the nematode Caenorhabditis elegans. It has a role as a Caenorhabditis elegans metabolite. It is a monocarboxylic acid and an (omega-1)- hydroxy fatty acid ascaroside . It derives from an (11R)-11-hydroxyundecanoic acid. It is a conjugate acid of an ascr18(1-).	The molecule is an (omega-1)- hydroxy fatty acid ascaroside obtained by formal condensation of the alcoholic hydroxy group of (10R)- 10-hydroxyundecanoic acid with ascaropyranose (the alpha anomer). It is a metabolite of the nematode Caenorhabditis elegans. It is a monocarboxylic acid and an (omega-1)- hydroxy fatty acid ascaroside . It derives from a (10R)-10-hydroxyundecanoic acid. It is a conjugate acid of an ascr18(1-).
	The molecule is a 2-oxo monocarboxylic acid that is pyruvic acid in which one of the methyl hydrogens is substituted by a 4-vinylcyclohex-2-en-1-yl group. It has a role as a plant metabolite. It derives from a pyruvic acid . It is a conjugate acid of a 4-[(1E)-4-vinylcyclohex-2-en-1-yl]pyruvate.	The molecule is a 2-oxo monocarboxylic acid that is pyruvic acid in which one of the methyl hydrogens has been replaced by a methylenecyclopropyl group. It has a role as a rat metabolite and a xenobiotic metabolite. It is a 2-oxo monocarboxylic acid, a member of cyclopropanes and an olefinic compound. It derives from a pyruvic acid .

Table 13: Accuracy (%) of molecule-text retrieval task on the PCdes and MoMu datasets. **Bold** indicates the best performance and underline indicates the second best performance. We report the performance of retrieval using a batch of 64 random samples and the entire test set.

(a) Accuracy (%) of molecule-text retrieval task on the PCdes dataset.

Model	Retrieval in batch		Retrieval in test set	
	M2T (%)	T2M (%)	M2T (%)	T2M (%)
Sci-BERT [2]	62.6	61.8	60.7	60.8
KV-PLM [58]	77.9	65.0	75.9	64.3
MoMu (Sci-BERT) [39]	80.6	77.0	79.1	75.5
MoMu (KV-PLM) [39]	81.1	80.2	80.2	79.0
MoleculeSTM [31]	86.2	83.9	84.6	85.1
MolCA (OPT-1.3B) [33]	91.4	88.4	90.5	87.6
3D-MoLM (LLaMA2-7B) [25]	<u>92.3</u>	89.6	<u>91.2</u>	88.5
UniMoT (LLaMA2-7B)	92.6	<u>89.4</u>	91.6	<u>88.3</u>

(b) Accuracy (%) of molecule-text retrieval task on the MoMu dataset.

Model	Retrieval in batch		Retrieval in test set	
	M2T (%)	T2M (%)	M2T (%)	T2M (%)
Sci-BERT [2]	1.4	1.6	0.3	0.3
KV-PLM [58]	1.5	1.3	0.5	0.3
MoMu (Sci-BERT) [39]	45.7	40.0	43.3	43.4
MoMu (KV-PLM) [39]	46.2	38.5	43.7	43.5
MoleculeSTM [31]	81.8	81.9	75.8	74.5
MolCA (OPT-1.3B) [33]	83.7	84.3	88.6	87.3
3D-MoLM (LLaMA2-7B) [25]	<u>84.9</u>	<u>85.4</u>	<u>89.9</u>	<u>88.7</u>
UniMoT (LLaMA2-7B)	85.4	85.6	90.3	89.0

648 from abundant labeled datasets, allowing for more comprehensive training and better performance.
 649 Addressing this data scarcity in the molecular domain is crucial for improving UniMoT’s training
 650 effectiveness and overall capabilities.

651 The current empirical evaluations, though extensive, are primarily conducted on standard datasets
 652 and benchmarks; expanding the evaluation to a broader array of datasets and real-world scenarios
 653 will provide a more comprehensive understanding of the model’s robustness and generalizability.

654 F Broader Impacts

655 The development of UniMoT, a unified model for molecule and text modalities, has significant
 656 potential to positively impact various fields. UniMoT can streamline the drug discovery process by
 657 enabling efficient molecule generation and optimization based on textual descriptions. In material
 658 science, it can aid in discovering new materials with desirable properties. Additionally, UniMoT
 659 can enhance research collaboration between chemists, biologists, and data scientists by integrating
 660 molecular and textual data, leading to comprehensive research insights and innovative solutions.

661 This paper does not pose any ethical concerns. The study does not involve human subjects and follows
 662 proper procedures for data set releases. There are no potentially harmful insights, methodologies, or
 663 applications. Additionally, there are no conflicts of interest or sponsorship concerns. Discrimination,
 664 bias, and fairness issues are not applicable. Privacy and security matters have been appropriately
 665 addressed, legal compliance has been maintained, and research integrity has been upheld.

666 **NeurIPS Paper Checklist**

667 **1. Claims**

668 Question: Do the main claims made in the abstract and introduction accurately reflect the
669 paper's contributions and scope?

670 Answer: [\[Yes\]](#)

671 Justification: We conduct extensive experiments to verify our claims.

672 Guidelines:

- 673 • The answer NA means that the abstract and introduction do not include the claims
674 made in the paper.
- 675 • The abstract and/or introduction should clearly state the claims made, including the
676 contributions made in the paper and important assumptions and limitations. A No or
677 NA answer to this question will not be perceived well by the reviewers.
- 678 • The claims made should match theoretical and experimental results, and reflect how
679 much the results can be expected to generalize to other settings.
- 680 • It is fine to include aspirational goals as motivation as long as it is clear that these goals
681 are not attained by the paper.

682 **2. Limitations**

683 Question: Does the paper discuss the limitations of the work performed by the authors?

684 Answer: [\[Yes\]](#)

685 Justification: We discuss the limitations in Appendix E.

686 Guidelines:

- 687 • The answer NA means that the paper has no limitation while the answer No means that
688 the paper has limitations, but those are not discussed in the paper.
- 689 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 690 • The paper should point out any strong assumptions and how robust the results are to
691 violations of these assumptions (e.g., independence assumptions, noiseless settings,
692 model well-specification, asymptotic approximations only holding locally). The authors
693 should reflect on how these assumptions might be violated in practice and what the
694 implications would be.
- 695 • The authors should reflect on the scope of the claims made, e.g., if the approach was
696 only tested on a few datasets or with a few runs. In general, empirical results often
697 depend on implicit assumptions, which should be articulated.
- 698 • The authors should reflect on the factors that influence the performance of the approach.
699 For example, a facial recognition algorithm may perform poorly when image resolution
700 is low or images are taken in low lighting. Or a speech-to-text system might not be
701 used reliably to provide closed captions for online lectures because it fails to handle
702 technical jargon.
- 703 • The authors should discuss the computational efficiency of the proposed algorithms
704 and how they scale with dataset size.
- 705 • If applicable, the authors should discuss possible limitations of their approach to
706 address problems of privacy and fairness.
- 707 • While the authors might fear that complete honesty about limitations might be used by
708 reviewers as grounds for rejection, a worse outcome might be that reviewers discover
709 limitations that aren't acknowledged in the paper. The authors should use their best
710 judgment and recognize that individual actions in favor of transparency play an impor-
711 tant role in developing norms that preserve the integrity of the community. Reviewers
712 will be specifically instructed to not penalize honesty concerning limitations.

713 **3. Theory Assumptions and Proofs**

714 Question: For each theoretical result, does the paper provide the full set of assumptions and
715 a complete (and correct) proof?

716 Answer: [\[NA\]](#)

717 Justification: The paper does not include theoretical results.

718 Guidelines:

- 719 • The answer NA means that the paper does not include theoretical results.
- 720 • All the theorems, formulas, and proofs in the paper should be numbered and cross-
- 721 referenced.
- 722 • All assumptions should be clearly stated or referenced in the statement of any theorems.
- 723 • The proofs can either appear in the main paper or the supplemental material, but if
- 724 they appear in the supplemental material, the authors are encouraged to provide a short
- 725 proof sketch to provide intuition.
- 726 • Inversely, any informal proof provided in the core of the paper should be complemented
- 727 by formal proofs provided in appendix or supplemental material.
- 728 • Theorems and Lemmas that the proof relies upon should be properly referenced.

729 4. Experimental Result Reproducibility

730 Question: Does the paper fully disclose all the information needed to reproduce the main ex-

731 perimental results of the paper to the extent that it affects the main claims and/or conclusions

732 of the paper (regardless of whether the code and data are provided or not)?

733 Answer: [Yes]

734 Justification: We disclose all the information to reproduce the experimental results in

735 Section 4, Appendix C, and Appendix D.

736 Guidelines:

- 737 • The answer NA means that the paper does not include experiments.
- 738 • If the paper includes experiments, a No answer to this question will not be perceived
- 739 well by the reviewers: Making the paper reproducible is important, regardless of
- 740 whether the code and data are provided or not.
- 741 • If the contribution is a dataset and/or model, the authors should describe the steps taken
- 742 to make their results reproducible or verifiable.
- 743 • Depending on the contribution, reproducibility can be accomplished in various ways.
- 744 For example, if the contribution is a novel architecture, describing the architecture fully
- 745 might suffice, or if the contribution is a specific model and empirical evaluation, it may
- 746 be necessary to either make it possible for others to replicate the model with the same
- 747 dataset, or provide access to the model. In general, releasing code and data is often
- 748 one good way to accomplish this, but reproducibility can also be provided via detailed
- 749 instructions for how to replicate the results, access to a hosted model (e.g., in the case
- 750 of a large language model), releasing of a model checkpoint, or other means that are
- 751 appropriate to the research performed.
- 752 • While NeurIPS does not require releasing code, the conference does require all submis-
- 753 sions to provide some reasonable avenue for reproducibility, which may depend on the
- 754 nature of the contribution. For example
- 755 (a) If the contribution is primarily a new algorithm, the paper should make it clear how
- 756 to reproduce that algorithm.
- 757 (b) If the contribution is primarily a new model architecture, the paper should describe
- 758 the architecture clearly and fully.
- 759 (c) If the contribution is a new model (e.g., a large language model), then there should
- 760 either be a way to access this model for reproducing the results or a way to reproduce
- 761 the model (e.g., with an open-source dataset or instructions for how to construct
- 762 the dataset).
- 763 (d) We recognize that reproducibility may be tricky in some cases, in which case
- 764 authors are welcome to describe the particular way they provide for reproducibility.
- 765 In the case of closed-source models, it may be that access to the model is limited in
- 766 some way (e.g., to registered users), but it should be possible for other researchers
- 767 to have some path to reproducing or verifying the results.

768 5. Open access to data and code

769 Question: Does the paper provide open access to the data and code, with sufficient instruc-

770 tions to faithfully reproduce the main experimental results, as described in supplemental

771 material?

772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822

Answer: [No]

Justification: Once our paper is accepted, we will make the code openly accessible.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We disclose all the details of our experiments in Appendix C and Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Given the considerable computational resources required for experiments with LLMs, we adhere to the common practice in the community.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- 823
- 824
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
 - 825 • It is OK to report 1-sigma error bars, but one should state it. The authors should
826 preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis
827 of Normality of errors is not verified.
 - 828 • For asymmetric distributions, the authors should be careful not to show in tables or
829 figures symmetric error bars that would yield results that are out of range (e.g. negative
830 error rates).
 - 831 • If error bars are reported in tables or plots, The authors should explain in the text how
832 they were calculated and reference the corresponding figures or tables in the text.

833 8. Experiments Compute Resources

834 Question: For each experiment, does the paper provide sufficient information on the com-
835 puter resources (type of compute workers, memory, time of execution) needed to reproduce
836 the experiments?

837 Answer: [Yes]

838 Justification: The information regarding compute resources is provided in Appendix C.

839 Guidelines:

- 840 • The answer NA means that the paper does not include experiments.
- 841 • The paper should indicate the type of compute workers CPU or GPU, internal cluster,
842 or cloud provider, including relevant memory and storage.
- 843 • The paper should provide the amount of compute required for each of the individual
844 experimental runs as well as estimate the total compute.
- 845 • The paper should disclose whether the full research project required more compute
846 than the experiments reported in the paper (e.g., preliminary or failed experiments that
847 didn't make it into the paper).

848 9. Code Of Ethics

849 Question: Does the research conducted in the paper conform, in every respect, with the
850 NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

851 Answer: [Yes]

852 Justification: We have carefully reviewed the code of ethics to ensure strict adherence to the
853 guidelines.

854 Guidelines:

- 855 • The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- 856 • If the authors answer No, they should explain the special circumstances that require a
857 deviation from the Code of Ethics.
- 858 • The authors should make sure to preserve anonymity (e.g., if there is a special consid-
859 eration due to laws or regulations in their jurisdiction).

860 10. Broader Impacts

861 Question: Does the paper discuss both potential positive societal impacts and negative
862 societal impacts of the work performed?

863 Answer: [Yes]

864 Justification: We discuss the broader impacts in Appendix F.

865 Guidelines:

- 866 • The answer NA means that there is no societal impact of the work performed.
- 867 • If the authors answer NA or No, they should explain why their work has no societal
868 impact or why the paper does not address societal impact.
- 869 • Examples of negative societal impacts include potential malicious or unintended uses
870 (e.g., disinformation, generating fake profiles, surveillance), fairness considerations
871 (e.g., deployment of technologies that could make decisions that unfairly impact specific
872 groups), privacy considerations, and security considerations.

- 873 • The conference expects that many papers will be foundational research and not tied
874 to particular applications, let alone deployments. However, if there is a direct path to
875 any negative applications, the authors should point it out. For example, it is legitimate
876 to point out that an improvement in the quality of generative models could be used to
877 generate deepfakes for disinformation. On the other hand, it is not needed to point out
878 that a generic algorithm for optimizing neural networks could enable people to train
879 models that generate Deepfakes faster.
- 880 • The authors should consider possible harms that could arise when the technology is
881 being used as intended and functioning correctly, harms that could arise when the
882 technology is being used as intended but gives incorrect results, and harms following
883 from (intentional or unintentional) misuse of the technology.
- 884 • If there are negative societal impacts, the authors could also discuss possible mitigation
885 strategies (e.g., gated release of models, providing defenses in addition to attacks,
886 mechanisms for monitoring misuse, mechanisms to monitor how a system learns from
887 feedback over time, improving the efficiency and accessibility of ML).

888 11. Safeguards

889 Question: Does the paper describe safeguards that have been put in place for responsible
890 release of data or models that have a high risk for misuse (e.g., pretrained language models,
891 image generators, or scraped datasets)?

892 Answer: [NA]

893 Justification: The paper poses no such risks.

894 Guidelines:

- 895 • The answer NA means that the paper poses no such risks.
- 896 • Released models that have a high risk for misuse or dual-use should be released with
897 necessary safeguards to allow for controlled use of the model, for example by requiring
898 that users adhere to usage guidelines or restrictions to access the model or implementing
899 safety filters.
- 900 • Datasets that have been scraped from the Internet could pose safety risks. The authors
901 should describe how they avoided releasing unsafe images.
- 902 • We recognize that providing effective safeguards is challenging, and many papers do
903 not require this, but we encourage authors to take this into account and make a best
904 faith effort.

905 12. Licenses for existing assets

906 Question: Are the creators or original owners of assets (e.g., code, data, models), used in
907 the paper, properly credited and are the license and terms of use explicitly mentioned and
908 properly respected?

909 Answer: [Yes]

910 Justification: The licenses for existing assets are provided in Appendix B.

911 Guidelines:

- 912 • The answer NA means that the paper does not use existing assets.
- 913 • The authors should cite the original paper that produced the code package or dataset.
- 914 • The authors should state which version of the asset is used and, if possible, include a
915 URL.
- 916 • The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- 917 • For scraped data from a particular source (e.g., website), the copyright and terms of
918 service of that source should be provided.
- 919 • If assets are released, the license, copyright information, and terms of use in the
920 package should be provided. For popular datasets, paperswithcode.com/datasets
921 has curated licenses for some datasets. Their licensing guide can help determine the
922 license of a dataset.
- 923 • For existing datasets that are re-packaged, both the original license and the license of
924 the derived asset (if it has changed) should be provided.

925 • If this information is not available online, the authors are encouraged to reach out to
926 the asset’s creators.

927 **13. New Assets**

928 Question: Are new assets introduced in the paper well documented and is the documentation
929 provided alongside the assets?

930 Answer: [NA]

931 Justification: The paper does not release new assets.

932 Guidelines:

- 933 • The answer NA means that the paper does not release new assets.
- 934 • Researchers should communicate the details of the dataset/code/model as part of their
935 submissions via structured templates. This includes details about training, license,
936 limitations, etc.
- 937 • The paper should discuss whether and how consent was obtained from people whose
938 asset is used.
- 939 • At submission time, remember to anonymize your assets (if applicable). You can either
940 create an anonymized URL or include an anonymized zip file.

941 **14. Crowdsourcing and Research with Human Subjects**

942 Question: For crowdsourcing experiments and research with human subjects, does the paper
943 include the full text of instructions given to participants and screenshots, if applicable, as
944 well as details about compensation (if any)?

945 Answer: [NA]

946 Justification: The paper does not involve crowdsourcing nor research with human subjects.

947 Guidelines:

- 948 • The answer NA means that the paper does not involve crowdsourcing nor research with
949 human subjects.
- 950 • Including this information in the supplemental material is fine, but if the main contribu-
951 tion of the paper involves human subjects, then as much detail as possible should be
952 included in the main paper.
- 953 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation,
954 or other labor should be paid at least the minimum wage in the country of the data
955 collector.

956 **15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human
957 Subjects**

958 Question: Does the paper describe potential risks incurred by study participants, whether
959 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)
960 approvals (or an equivalent approval/review based on the requirements of your country or
961 institution) were obtained?

962 Answer: [NA]

963 Justification: The paper does not involve crowdsourcing nor research with human subjects.

964 Guidelines:

- 965 • The answer NA means that the paper does not involve crowdsourcing nor research with
966 human subjects.
- 967 • Depending on the country in which research is conducted, IRB approval (or equivalent)
968 may be required for any human subjects research. If you obtained IRB approval, you
969 should clearly state this in the paper.
- 970 • We recognize that the procedures for this may vary significantly between institutions
971 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
972 guidelines for their institution.
- 973 • For initial submissions, do not include any information that would break anonymity (if
974 applicable), such as the institution conducting the review.