Exploring Molecular Pretraining Model at Scale

Xiaohong Ji[1][*], **Zhen Wang**[1][*], **Zhifeng Gao**[1][*], **Hang Zheng**[1]
Linfeng Zhang[1,2], **Guolin Ke**[1] , **Weinan E**[2,3,4]
[1]DP Technology, Beijing, 100080, China.
[2]AI for Science Institute, Beijing 100080, China.
[3]School of Mathematical Sciences, Peking University, Beijing, 100871, China.
[4]Center for Machine Learning Research, Peking University, Beijing 100084, China.
{jixh, wangz, gaozf, zhengh, zhanglf, kegl}@dp.tech
weinan@math.pku.edu.cn

Abstract

In recent years, pretraining models have made significant advancements in the fields of natural language processing (NLP), computer vision (CV), and life sciences. The significant advancements in NLP and CV are predominantly driven by the expansion of model parameters and data size, a phenomenon now recognized as the scaling laws. However, research exploring scaling law in molecular pretraining models remains unexplored. In this work, we present Uni-Mol2 , an innovative molecular pretraining model that leverages a two-track transformer to effectively integrate features at the atomic level, graph level, and geometry structure level. Along with this, we systematically investigate the scaling law within molecular pretraining models, characterizing the power-law correlations between validation loss and model size, dataset size, and computational resources. Consequently, we successfully scale Uni-Mol2 to 1.1 billion parameters through pretraining on 800 million conformations, making it the largest molecular pretraining model to date. Extensive experiments show consistent improvement in the downstream tasks as the model size grows. The Uni-Mol2 with 1.1B parameters also outperforms existing methods, achieving an average 27% improvement on the QM9 and 14% on COMPAS-1D dataset.

1 Introduction

With the exponential growth of available biological data, there arises a critical need for innovative computational methodologies to utilize this wealth of information effectively. While traditional molecular representations like fingerprint-based models [1, 2] lack the ability to capture fine-grained structural features and struggle to handle large or complex molecules effectively. Molecular Representation Learning (MRL) using molecular pretraining emerges as a promising approach, leveraging the power of machine learning to imbue algorithms with a deep understanding of molecular structures and functions. Various modalities of molecular representation by pretraining have been extensively studied in the past. The typical approach for representing molecules involves two main strategies. One strategy is to represent molecules as one-dimensional sequential strings, such as SMILES [3, 4] and InChI [5]. The representative work is SMILES-BERT[3], which learns from large-scale unlabeled data through the masked SMILES recovery task. Another strategy is to represent molecules as two-dimensional graphs [6, 7, 8]. MolCLR [8], a typical method, learns the representations from unlabeled data by contrasting positive molecule graph pairs against negative ones. Additionally, a growing trend is to leverage three-dimensional information in MRL to enable tasks like 3D geometry

---

[*]Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

prediction or generation [9, 10, 11]. The pursuit of molecular pretraining has sparked a wave of exploration and innovation across the field, marking a new era of discovery within the discipline.

While in the past few years, scaling up pre-trained language models [12, 13, 14, 15, 16, 17, 18] has been achieved remarkable progress in natural language processing (NLP) and computer vision (CV). The exponential growth in model size and the richness of training data have significantly enhanced the capabilities and performance of LLMs across various NLP and CV tasks. Despite extensive research on molecular pretraining, the majority of prior studies have been conducted on a relatively small scale, utilizing limited parameters and datasets. Learning scalable molecular representation learning is rarely explored and remains a challenging problem. The recent [19]'s work conducts a series of data-centric experiments to demonstrate scaling behaviors in various aspects. The exploration of the molecular pretraining model is limited to the GIN [20], SchNet [21], whose model scale and data scale are comparatively small.

To delve deeper into the scaling of molecular pretraining foundational models, our preliminary investigations have yielded notable insights within this domain. We summarize the contributions of this work as follows:

- We have curated and organized a dataset comprising approximately 884 million 3D conformations, which contains 73 million scaffolds for pretraining. To the best of our knowledge, this is the largest dataset of molecules with 3D conformations for molecular pretraining to date, which provides the foundation ingredient for training large-scale molecular models.

- We systematically study the scalability and flexibility of Uni-Mol2 in terms of model parameters, which range from 84M to 1.1B parameters, and characterize the relationship between validation loss and model size, dataset size, and computational resources. It is the first time to demonstrate the scaling law of molecular pretraining and Uni-Mol2 is currently the largest billion-scale molecular pretraining model to date.

- We present an in-depth analysis of scaling trends about fine-tuning on downstream tasks as the results are shown in Table4 and 5, Uni-Mol2 demonstrates consistent improvement in downstream task performance with increasing model parameters. The 1.1 billion parameters model also achieves significant improvement over the existing method.

2   Related Work

Molecular representation learning   Previous research has extensively investigated various modalities for molecular representation. A range of methods have been proposed based on different types of information utilized during pretraining. SMILES-BERT[3] uses the smiles sequence in pretraining to capture the representation. Due to SMILES representation lack of explicit encoding of molecular structural information. To address this limitation, GROVER integrates Message Passing Networks into a Transformer-style architecture and learns from unlabeled molecular data through carefully designed self-supervised tasks at different levels of molecular topology (node, edge, and graph). MGSSL [22] developed a motif-based graph self-supervised learning strategy that predicts both the topology and label of motifs as they are generated in the motif tree process. Furthermore, GEM[6] incorporates three-dimensional (3D) spatial structure information, atoms, bonds, and bond angles simultaneously to model the molecular representation. SphereNet[23] utilizes spherical coordinates and introduces a spherical message-passing approach, providing an effective framework for 3D molecular learning. The survey [24] discusses GNN shows great efficiency in processing molecular graph structures and their strong capability to capture local relationships within a molecule. However, the locally connected graph fails to adequately represent long-range interactions between atoms. In contrast, transformer-based models have shown exceptional performance in various tasks within the molecular domain, demonstrating remarkable representation capabilities.

Foundation models   Recently, there has been considerable interest in developing foundational models to consolidate and expand representations. The significant advancements in scaling up pre-trained language models[12, 13, 14] have fundamentally reshaped the field of natural language processing. [25, 15, 18, 26] also prove that the foundation model demonstrates strong performance on many NLP datasets, sometimes reaching or exceeding the human performance. Some works in CV[27, 28] demonstrate the potential for "LLM-like" scaling in vision and underscore significant improvement via model and data scaling. And Sora[29, 30], a multi-modal foundation model exhibits

2

Table 1: The different scale of Uni-Mol dataset and Uni-Mol2 dataset

Datasets	SMILES	Scaffold	Data Source
Uni-Mol Dataset	19M	4,224,621	ZINC15, ChemBL, Commercial Database[11]
Uni-Mol2 Dataset	$\sim$ 884M	73,725,454	Uni-Mol Dataset, ZINC 20[31]

the capacity to offer sophisticated understanding regarding the intricate interplay of physical and contextual dynamics within depicted scenes.

3 Pretraining

The pretraining stage of molecular involves learning from vast amounts of molecular data to acquire a comprehensive understanding of molecular representations. By pretraining on a large and diverse unlabeled dataset, the model can develop a rich understanding of molecular structures and properties, which can subsequently be fine-tuned or applied to specific downstream tasks, such as drug discovery, materials design, or chemical synthesis. The section provides details of the data curation process for pretraining, the detailed pretraining architecture, the well-designed self-supervision tasks, and the specific training procedures employed for scaling up the model.

3.1 Data

To augment the richness and diversity of the dataset, we integrated the two parts we have collected. One part consists of approximately 19 million molecules sourced from Uni-Mol [11], while the other is derived from ZINC20 [31] which includes 1.4 billion compounds. We downloaded the subset with standard reactivity, which contains 884 million compounds from website [2]. Table 1 shows the enrichment compared with Uni-Mol dataset. The overall Uni-Mol2 dataset has increased by over 40 times compared to the Uni-Mol dataset, with the number of scaffold increasing by 17 times, greatly expanding the diversity of the data. Figure 1(**Top**) shows the numeric distributions of the top 40 skeletons in Uni-Mol dataset and the number corresponding in Uni-Mol2 dataset. To prevent data leakage in evaluating pretraining performance, we randomly sampled 520k molecules from the Uni-Mol2 dataset as the validation set to evaluate the effectiveness and investigate the scaling relationship.

As illustrated in the visualization depicting the frequency distribution of the top 40 Murcko scaffolds in Uni-Mol2 dataset (refer to Figure 1 (**Bottom**)), it is observed that the molecular scaffold conforms to a distribution characterized by a long-tail pattern. To create a more balanced training dataset, we categorize the SMILES of Uni-Mol2 training set by Murcko scaffold, resulting in 73,725,454 scaffolds along with frequency distribution. Then, we utilize the temperature-based sampling method [32][33], as described in equation 1 to select molecules from Uni-Mol2 training set.

$$
\begin{aligned}
P_i &= \frac{N_{s_i}}{\sum N_{s_i}}, \\
P_{scaffold_i} &= \text{softmax}(\frac{P_i}{\tau})
\end{aligned}
\tag{1}
$$

Where N_{si} represents the number of molecules with i-th scaffold in Uni-Mol2 training set. The temperature τ modulates the smoothness of the molecular distribution across scaffolds. We use an empirical value $\tau = 0.005$ as the temperature to effectively balance the proportion of molecules with high-frequency and low-frequency scaffolds.

3.2 Architecture

As depicted in Figure 2, Uni-Mol2 essentially adheres to the model design of Uni-Mol+[34], acting as a two-track transformer that concurrently processes atom features and pair features. Consistent with

---

[2]https://zinc20.docking.org/tranches/home/

3
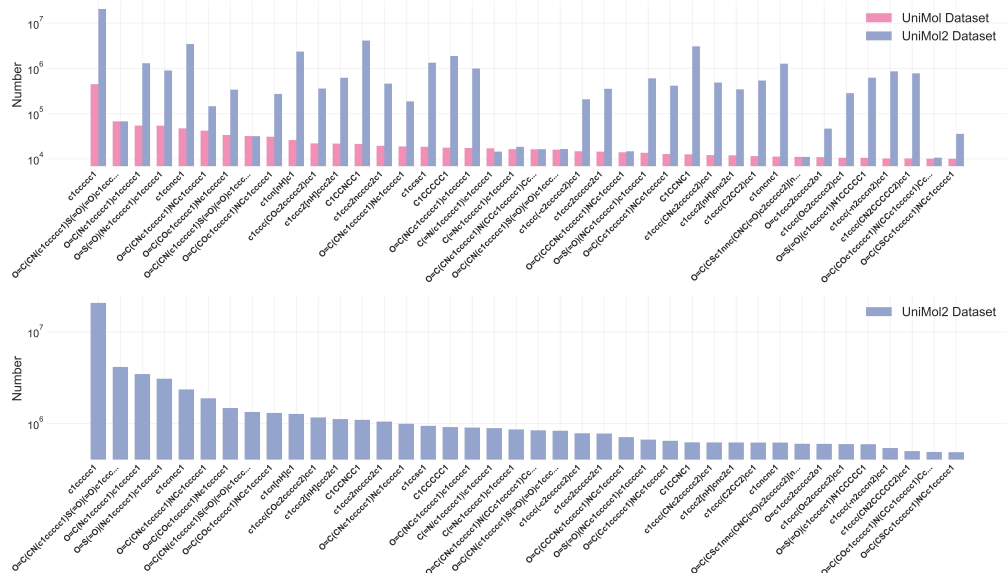
Figure 1: **Top**: Comparison of scaffold frequency between Uni-Mol and Uni-Mol2 dataset. **Bottom**: Scaffolds distribution on Uni-Mol2 dataset
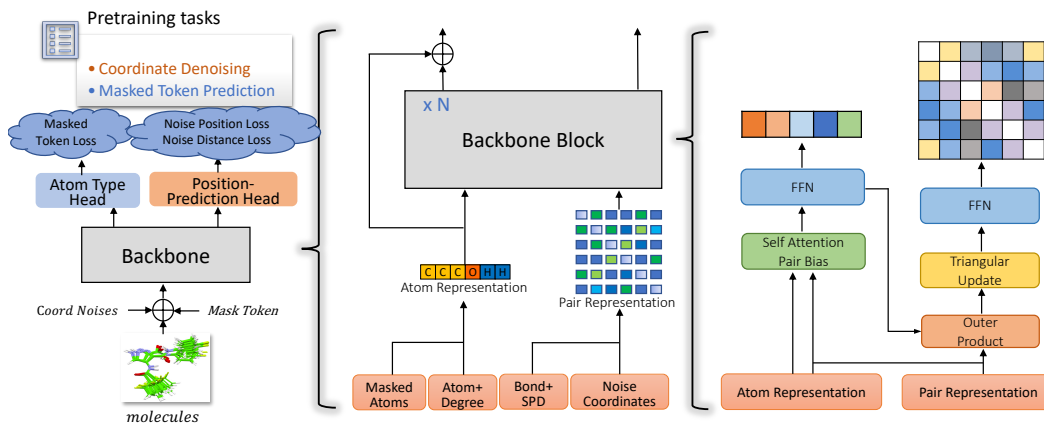
Figure 2: **Left**: The overall pretraining architecture. **Middle**: Atom and Pair representation. **Right**: The details of backbone block

Uni-Mol[11], Uni-Mol2 employs two self-supervised tasks: masked token prediction and molecule coordinate denoising. The detailed framework is presented as follows:

Feature Representation and Position Encoding Given molecular $M = (x, e, r)$, where $x \in \mathbb{R}^{n \times d_a}$ denotes atom features, $e \in \mathbb{R}^{n \times n \times d_e}$ denotes bond features and $r \in \mathbb{R}^{n \times 3}$ denotes coordinate features. Following Uni-Mol+, we employ RDKit to obtain atom token x_{token}^i, atom degree x_{degree}^i, and atomic features x_{atomic}^i for each atom. The atom embedding x_{atom}^i is then initialized as:

$$x_{\text{atom}}^i = \text{Embedding}(x_{\text{token}}^i) + \text{Embedding}(x_{\text{degree}}^i) + \text{Embedding}(x_{\text{atomic}}^i) \qquad (2)$$

For pair features, we utilize RDKit to obtain bond features $x_{\text{bond}}^{i,j}$ by $\text{Embedding}(x_{\text{bond}}^{i,j})$. We adopt the method from [35, 34] to encode the shortest path distance $x_{\text{SPD}}^{i,j}$ of atom pair (i, j) in the molecular graph by $\text{Embedding}(x_{\text{SPD}}^{i,j})$. Additionally, we employ the Gaussian kernel approach with pair type, as described in [36, 11], to encode the Euclidean distance of the atom pair (i, j) by $\psi^{i,j}$. The pair embedding $x_{\text{pair}}^{i,j}$ is then initialized as:

4

$$x_{\text{pair}}^{i,j} = \text{Embedding}(x_{\text{bond}}^{i,j}) + \text{Embedding}(x_{\text{SPD}}^{i,j}) + \psi^{i,j} \tag{3}$$

Two-track Transformer Layer The backbone of Uni-Mol2 has N blocks, each block handles atom representation and pair representation concurrently. Formally, for the l-th block, Uni-Mol2 update atom representation x^l by

$$
\begin{aligned}
x^l &= \text{SelfAttentionPairBias}(\text{LN}(x^{l-1}), p^{l-1}), \\
x^l &= x^{l-1} + \text{FFN}(\text{LN}(x^l))
\end{aligned}
\tag{4}
$$

For the pair representation p^l,

$$
\begin{aligned}
p^l &= p^{l-1} + \text{OuterProduct}(\text{LN}(p^{l-1})), \\
p^l &= p^l + \text{TriangularUpdate}(\text{LN}(p^l)), \\
p^l &= p^l + \text{FFN}(\text{LN}(p^l))
\end{aligned}
\tag{5}
$$

The details of SelfAttentionPairBias, OuterProduct, and TriangularUpdate are aligned with those of Uni-Mol+. Additionally, Uni-Mol2 adopts pre-norm layer normalization at atom and pair representation, which differs from Uni-Mol+, to improve stability in the model's training dynamics. Specifically, we set atom embedding x_{atom} as atom representation x^0 and pair embedding x_{pair} as pair representation p^0 for the first block.

Pretraining Tasks To effectively model the structure of molecular conformations, we set pretraining tasks basically following Uni-Mol. In detail, for each molecule, we randomly mask 15% of the atom tokens with the placeholder token [MASK]. We then add the atom token prediction head to optimize masked atom token loss $\mathcal{L}_{\text{atom}}$ by

$$\mathcal{L}_{\text{atom}} = H(x_{\text{atom}}[\text{mask}], x_{\text{patom}}[\text{mask}]) \tag{6}$$

where H denotes the cross entropy function, $x_{\text{atom}}[\text{mask}]$ denotes the masked atom tokens and $x_{\text{patom}}[\text{mask}]$ denotes the corresponding predicted atom tokens for the masked positions.

In the coordinate denoising task, to increase the challenge of the pertaining task, we introduce Gaussian noise with a standard deviation of 0.2 for all the atom coordinates. Additionally, to enhance broader applicability across downstream applications, we mask atomic features x_{atomic}, bond features x_{bond}, and shortest path distance features x_{SPD} with a probability of 50%. Furthermore, we align the conformation of the noised molecule, denoted as $r_{\text{noised\_coor}}$, with that of the raw molecule, denoted as r_{coor}, using the Kabsch algorithm.

In contrast to Uni-Mol, Uni-Mol2 employs the position prediction head to predict the atom coordinates r_{pcoor} of molecules.

$$
\begin{aligned}
\Delta_{pos} &= \text{Dis}(r_{\text{noised\_coor}}) \\
Q_{pos} &= \text{FFN}(\text{LN}(x^N)), K_{pos} = \text{FFN}(\text{LN}(x^N)) \\
V_{pos} &= \text{FFN}(\text{LN}(x^N)), B_{pos} = \text{FFN}(\text{LN}(p^N)) \\
attn_{pos} &= \text{softmax}(Q_{pos}K_{pos}^T + B_{pos}) \circ \Delta_{pos} \\
\Delta_{vpos} &= attn_{pos}V_{pos}, \Delta_{ppos} = \text{FFN}(\Delta_{vpos}) \\
r_{\text{pcoor}} &= r_{\text{noised\_coor}} + \Delta_{ppos}
\end{aligned}
\tag{7}
$$

where Dis denotes element-wise subtraction of positions between different noised atoms $r_{\text{noised\_coor}}$. Specifically, the difference in position between atoms i and j is given by $\Delta_{pos}(i,j) = r_{\text{noised\_coor},i} - r_{\text{noised\_coor},j}$. And $\circ$ denotes Hadamard product. LN denotes layer normalization. FFN denotes a feed-forward network. In practice, we use multi-head attention; for simplicity in writing, we omitted the notation related to heads here. Once the predicted coordinates r_{pcoor} are obtained, the predicted pair-distance $r_{\text{pdistance}}$ can be derived by calculating the Euclidean distances between each pair of r_{pcoor}. We integrated coordinate prediction and pair-distance prediction with ℓ_1 loss into Uni-Mol2's optimization process for the coordinate denoising task:

5

Table 2: Architecture of Uni-Mol2 at different scale

Params	Layers	Embedding dim	Attention heads	Pair embedding dim	Pair hidden dim	FFN embedding dim	Learning rate	Batch size
42M	6	768	48	512	64	768	1e-4	1024
84M	12	768	48	512	64	768	1e-4	1024
164M	24	768	48	512	64	768	1e-4	1024
310M	32	1024	64	512	64	1024	1e-4	1024
570M	32	1536	96	512	64	1536	1e-4	1024
1.1B	64	1536	96	512	64	1536	1e-4	1024

$$\mathcal{L}_{\text{coor}} = \|r_{\text{pcoor}} - r_{\text{coor}}\|_1,$$
$$\mathcal{L}_{\text{distance}} = \|r_{\text{pdistance}} - r_{\text{distance}}\|_1 \tag{8}$$

We eliminated two stabilizing regularization terms from the Uni-Mol model, yielding the final loss of Uni-Mol2:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{atom}} + \mathcal{L}_{\text{coor}} + \mathcal{L}_{\text{distance}} \tag{9}$$

3.3 Hyperparameter and Training Details

We study the scalability of Uni-Mol with the scale from 42M to 1.1B, and all the parameters for Uni-Mol2 at different scales are listed in Table 2. And Uni-Mol2 is trained with AdamW optimzer[37, 38], with the following hyper-parameters: $\beta1 = 0.9$ and $\beta2 = 0.99$ and weight decay $1e - 4$. The gradient clip norm is set to 1.0 for training stability. The learning rate scheduler employed is a polynomial decay scheduler during pretraining. Specifically, all models reach its maximum learning rate value $1e - 4$ after 100,000 warm-up steps and decay the learning rate of each parameter group using a polynomial function with power 1.0. All the models are trained with mix-precision[39] for training efficiency.

Using the temperature-based sampling method outlined in Equation 1, we sample 838 million conformations as training samples from the dataset. All models were subsequently trained on these 838 million samples. All these conformations were generated using the ETKGD method [40] and optimized with the Merck Molecular Force Field (MMFF) [41] in RDKit. For models containing parameters ranging from 42M to 310M, we employed 32 NVIDIA A100 GPU cards, while for models with 570M and 1.1B parameters, we utilized 64 NVIDIA A100 GPU cards. Details on pretrain time complexity across varying sizes are provided in Appendix 12.

4 Scaling Laws

Several studies[15, 42, 43] on large language models (LLMs) investigate the power-law connections between model performance, commonly assessed by validation or test loss, and factors such as the number of model parameters, dataset size, and compute budget. Here, we aim to define the power-law of validation loss $\mathcal{L}$ during the model's convergence period. In Figure 3, we present the validation loss of Uni-Mol2 models with parameter counts varying from 42 million to 1.1 billion during the training process. We mainly examine the impact factors of three aspects: data scale N, model scale M, and compute budget scale C. Given that a constant batch size B of 1024 is maintained for Uni-Mol2 across various scales, the number of training steps S is considered as a suitable proxy for D, as D can be approximated by the product BS.

We initially designed a power term for M and S separately. Additionally, we approximate the computed budget C as MS. Notably, we have neglected the intricate relationship between actual computing costs C and MS, instead subsuming it into the parameter estimation. Adhering to the design principles of [42], the loss function $\mathcal{L}(M, D)$ should exhibit scale invariance, limit consistency, and analyticity to ensure stability and consistency across varying parameters. As a result, we derived the following empirical power-law relationship:

$$\mathcal{L}(M, S, C) = \alpha_m M^{\beta_m} + \alpha_s S^{\beta_s} + \alpha_c C^{\beta_c} \tag{10}$$

6
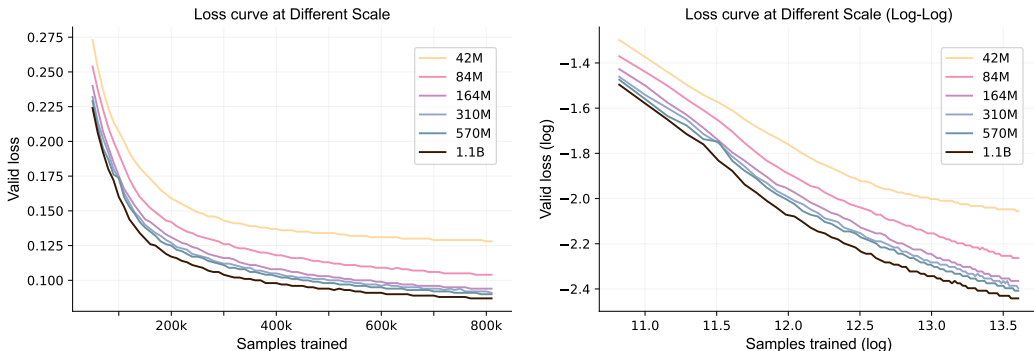
Figure 3: Validation loss curves. Training curves for Uni-Mol2 model from 42M to 1.1B parameters. Models are trained on 0.8B samples. At the convergence stage, the 84M parameters model has a loss of 0.105, and the 1.1B parameters model reaches a loss of 0.087.
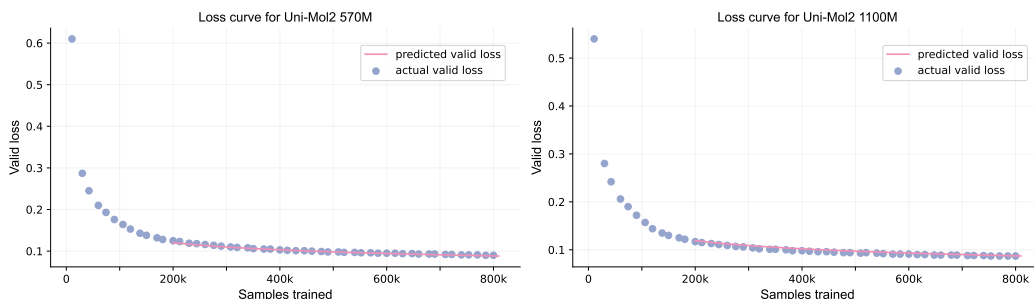
Figure 4: Graph of actual loss and prediction loss across different updates for the 570M (**left**) and 1.1B (**right**) models

We established the relationship based on the validation loss trajectory of Uni-Mol2 across different scales, as detailed in Table 2. Specifically, we utilized the validation data from Uni-Mol2 42M, 84M, 164M, and 310M, recording the validation loss every 10,000 training steps. Furthermore, to prevent the performance during the transient period from affecting the parameter estimation, we excluded the loss of information from the first 200,000 training steps. Consequently, we have:

$$\mathcal{L}(M, S, C) = 2.660M^{-1.137} + 1.848S^{-0.225} + 0.588C^{-1.479} \tag{11}$$

As shown in Fig 4, equation 11 fits the actual validation loss well for Uni-Mol2 570M and Uni-Mol2 1.1B parameters model, particularly when the model's performance reaches convergence. To assess the scaling law's effectiveness, we calculated Relative Mean Absolute Error (RMAE), Mean Square Error (MSE), R-squared, and Pearson Correlation Coefficient by comparing predicted validation loss with actual validation loss over the last 100,000 steps for Uni-Mol2 570M and Uni-Mol2 1.1B on Table 3. The high Pearson Correlation Coefficient and R-squared we computed indicate a strong linear relationship between our predicted values and the actual data. The RMAE values for Uni-Mol2 570M and Uni-Mol2 1.1B are 0.0169 and 0.0095, respectively, indicating that Equation 11 accurately models the loss curve. Specifically, for the Uni-Mol2 570M at 810,000 steps, the actual validation loss was recorded at 0.09, compared to a predicted loss of 0.088, yielding a predicted validation error of 2.22%. Meanwhile, for Uni-Mol2 1.1B at the same step, the actual validation loss stood at 0.087, slightly below the forecast of 0.0871, with a prediction error of 0.23%.

5   Downstream Experiment

Upon pretraining with extensive unlabeled datasets using the predefined task, one should acquire a highly accurate molecular representation for fine-tuning downstream tasks. In this section, we conduct experiments on the ability of scaled models on downstream tasks.

7

Table 3: Metrics about Scaling Law for Uni-Mol2

Model	RMAE	MSE	R-Squared	Pearson Correlation Coefficient
Uni-Mol2 570M	0.0169	2.450e-6	0.92	0.85
Uni-Mol2 1.1B	0.0095	8.458e-5	0.87	0.75

Table 4: Mean absolute error(MAE, ↓) results on QM9 Dataset

Model	HOMO / LUMO / GAP	alpha	C_v	mu	R^2	ZPVE
$GROVER_{base}$	0.0079 (3e-04)	2.365 (0.302)	1.103 (0.339)	0.618 (0.002)	113.01 (4.206)	0.0035(3e-04)
$GROVER_{large}$	0.0083 (6e-04)	2.240 (0.385)	0.853 (0.186)	0.623 (0.006)	85.85 (6.816)	0.00381(5e-04)
GEM	0.0067(4e-05)	0.589(0.0042)	0.237(0.0137)	0.444(0.0015)	25.67(0.743)	0.0011(2e-05)
Uni-Mol	0.0043(2e-05)	0.363(0.009)	0.183(0.002)	0.155(0.0015)	4.805(0.055)	0.0011(3e-05)
Uni-Mol2 84M	0.0038(5e-05)	0.376(0.027)	0.178(0.012)	0.105(0.0009)	4.968(0.235)	0.0010(1e-04)
Uni-Mol2 164M	0.0036(1e-05)	0.325(0.004)	0.157(0.017)	0.093(0.0006)	4.935(0.189)	0.0005(1e-05)
Uni-Mol2 310M	0.0036(1e-05)	0.315(0.003)	**0.143(0.002)**	0.092(0.0013)	4.672(0.245)	0.0005(1e-05)
Uni-Mol2 570M	0.0036(2e-05)	0.315(0.004)	0.147(0.0007)	0.089(0.0015)	4.523(0.080)	0.0005(3e-05)
Uni-Mol2 1.1B	**0.0035(1e-05)**	**0.305(0.003)**	0.144(0.002)	**0.089(0.0004)**	**4.265(0.067)**	**0.0005(8e-05)**

5.1 QM9 Dataset

We employ QM9 [44, 45] datasets to evaluate the performance of the molecular pretraining model at different scales and compare Uni-Mol2 with representative existing methods. QM9 dataset provides the geometric, energetic, electronic, and thermodynamic properties of the molecule, comprising 134 thousand stable organic molecules with up to nine heavy atoms. Due to QM9 containing several quantum mechanical properties with different quantitative ranges, each property is treated as a separate task. However, the HOMO, LUMO, and HOMO-LUMO GAP, which share similar ranges, are trained together as a single task for simplicity [6].

Baselines We evaluate Uni-Mol2 against several baseline models, with a primary emphasis on pretraining baselines. Given that Uni-Mol demonstrates superior performance compared to these baselines in previous work [11], our analysis concentrates on the comparison between Uni-Mol and Uni-Mol2, specifically examining the scalability of Uni-Mol2 at various scales. It is noted that we have shifted the dataset partitioning method from scaffold-based partitioning to scaffold similarity-based partitioning, thereby increasing the task difficulty to evaluate the model's performance more comprehensively. The dataset is then divided into training, validation, and test sets in proportions of 80%, 10%, and 10%, respectively. Following previous work [6, 11], we report the mean and standard deviation by the results of 3 random seeds.

Results The results are presented comprehensively in Table 4, where the best results are marked in bold. Uni-Mol still outperforms baselines on almost all downstream datasets. Uni-Mol2 outperforms Uni-Mol in four out of the six tasks examined. But as the model parameters increase, Uni-Mol2 demonstrates significantly improved performance, surpassing Uni-Mol across all tasks at the 1.1 billion parameter level, achieving an average 27% improvement on the QM9 task for all properties. We systematically investigate the scaling of Uni-Mol2 across parameter sizes ranging from 84 million to 1.1 billion. Except for the C_v property prediction task, the results for other properties progressively improve as the model size increases, consistent with the patterns observed in the model's validation performance. This indicates that enlarging the model consistently enhances downstream performance. However, for properties such as HOMO, LUMO, HOMO-LUMO GAP, and ZPVE, the results converge as the model size increases. This convergence suggests that further increases no longer influence the performance ceiling for these tasks in model size.

5.2 COMPAS-1D Dataset

Due to the QM9 dataset only providing the conformation, some molecules failed to generate the atom and bond feature correctly. Therefore, fine-tuning Uni-Mol2 on the existing QM9 dataset to evaluate its effectiveness with bond and edge features presents a non-trivial challenge. To further validate the performance and generalization capabilities of the Uni-Mol2 pretraining model, we

8

Table 5: Mean absolute error(MAE, ↓) results on COMPAS-1D Dataset.

Model	aEA	aIP	dispersion	Dipmom Debye
Uni-Mol	**0.0099(2e-05)**	0.0083(9e-05)	0.0092(6e-04)	0.0198(2e-04)
Uni-Mol2 84M	0.0104(2e-05)	0.0081(3e-05)	0.0092(5e-04)	0.0196(1e-04)
Uni-Mol2 1.1B	0.0103(4e-04)	**0.008(1e-05)**	**0.0081(1e-04)**	**0.0186(3e-04)**
Uni-Mol2 84M ⋆	0.0104(4e-04)	0.0077(5e-05)	0.0085(1e-04)	0.0173(6e-04)
Uni-Mol2 1.1B ⋆	**0.0093(4e-05)**	**0.0074(9e-05)**	**0.0067(2e-04)**	**0.0170(2e-04)**

Table 6: Mean absolute error(MAE, ↓) about HOMO-LUMO GAP on QM9 Dataset

Model	train50	train100	train200
Uni-Mol2 84M	0.0062(8.1e-05)	0.0053(1.0e-06)	0.0046(1.0e-06)
Uni-Mol2 164M	0.0058(3.7e-05)	0.0050(1.4e-05)	0.0044(6.9e-05)
Uni-Mol2 310M	0.0056(4.7e-05)	0.0049(0.4e-06)	0.0044(4.0e-05)
Uni-Mol2 570M	0.0057(4.2e-05)	0.0048(1.8e-05)	0.0044(8.1e-06)
Uni-Mol2 1.1B	**0.0056(1.8e-05)**	**0.0048(3.5e-05)**	**0.0043(4.7e-05)**

utilized COMPAS-1D from COMPAS project [46]. COMPAS-1D offers essential computational properties crucial for comprehending the behaviour of polycyclic aromatic hydrocarbons and other organic molecules across various chemical and physical processes. Modeling the relationships of these properties has significant implications for the field of organic photoelectric materials.

We still follow the QM9 scaffold similarity-based partition and split it by a ratio of 8:1:1 into the train, validation, and test sets. Table 5 presents the predictive capabilities of Uni-Mol2 regarding photoelectric quantum properties. The model with ⋆ suffix indicates that they incorporate atom and bond features. The results indicate that Uni-Mol2 excels in all tasks except for aEA property prediction task. Additionally, consistent with findings from the QM9 dataset, Uni-Mol2 demonstrates superior performance across all tasks as the model scales up. The results also show that under the same parameter scale, models incorporating atom and bond features outperform those without these features. Uni-Mol2 1B achieves 4% improvement over Uni-Mol, while Uni-Mol2 1B with atom and bond feature achieves 14% improvement over Uni-Mol. This suggests that, in certain scenarios, these features consistently provide a significant advantage.

5.3   The Performance on Limited QM9 Dataset

In numerous fields like bio-medicine, acquiring extensive well-annotated molecular data is often expensive and time-consuming. Typically, these datasets include only a limited quantity of data[47, 48]. To evaluate the performance of Uni-Mol2 with restricted data availability, we conducted sampling on the QM9 dataset. We sampled the training set by stratifying it according to the quantile binning of the HOMO-LUMO GAP label from the QM9 test set and then created subsets named train50, train100, and train200 by sampling at 50%, 100%, and 200% of the test set size, respectively.

We enhanced Uni-Mol2 from 84M to 1.1B parameters using train50, train100, and train200 datasets to predict HOMO, LUMO, and GAP properties on the QM9 test dataset. As illustrated in Table 6, two conclusions emerge from the MAE for predicting HOMO, LUMO, and HOMO-LUMO GAP on the QM9 test set. First, the model's performance, indicated by a decreasing MAE, progressively improves as the training dataset expands. This is evident from comparing the MAE values between the train50 and train200 rows across different scales of the Uni-Mol2 models. For example, the Uni-Mol2 84M model shows a reduction in MAE from 0.0062 to 0.0046, marking a 25.8% decrease as the dataset grows from 50 to 200 instances. Secondly, in situations where training data is scarce, the larger Uni-Mol2 models demonstrate enhanced predictive capabilities. This is evidenced by the fact that the Uni-Mol2 1.1B parameters model, which has the largest parameters, consistently records the lowest MAE scores for all sizes of training sets. This is especially apparent in the train50 scenario, where it achieves an MAE of 0.0056, marking the best performance among the models discussed. These results highlight the advantages of enlarging both the training dataset and the model scale to improve predictive accuracy in downstream finetuning tasks with Uni-Mol2.

9

6 Conclusion

In this paper, to fully investigate the scaling law in the molecular pretraining field, we construct a diverse dataset of molecular structures spanning 884 million instances and present a novel molecular pretraining model Uni-Mol2. We successfully scale the model size to 1.1 billion parameters from 84 million parameters and characterize the power-law relationship between validation loss and model size, dataset size, and computational resources. By empowering the power-law relationship of Uni-Mol2, it can shed light on the performance of the larger model. Our largest 1.1B parameters model also outperforms the existing methods.

The scaling law paves the way for exploring larger models to achieve higher performance. We hope that our work can open avenues for further exploration of the foundational molecular pretraining model. While larger models yield substantial benefits, there are still several potential future directions. Firstly, beyond property prediction tasks, it is also worthwhile to explore whether the representation can be effectively utilized to enhance generative tasks. Secondly, even though the Uni-Mol2 has shown excellent results in several domains by increasing model capacity, it remains to be explored whether the advantages of scaling are beneficial for a broader range of tasks. Thirdly, the current mainstream large language models (LLMs) are predominantly based on a decode-only architecture. It is worth investigating whether there are more elegant decode-only architectures for molecular pre-training models.

References

[1] Minjian Yang et al. "Machine learning models based on molecular fingerprints and an extreme gradient boosting method lead to the discovery of JAK2 inhibitors". In: *Journal of Chemical Information and Modeling* 59.12 (2019), pp. 5002–5012.

[2] David Rogers and Mathew Hahn. "Extended-connectivity fingerprints". In: *Journal of chemical information and modeling* 50.5 (2010), pp. 742–754.

[3] Sheng Wang et al. "Smiles-bert: large scale unsupervised pre-training for molecular property prediction". In: *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*. 2019, pp. 429–436.

[4] Zheng Xu et al. "Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery". In: *Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics*. 2017, pp. 285–294.

[5] Robin Winter et al. "Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations". In: *Chemical science* 10.6 (2019), pp. 1692–1701.

[6] Xiaomin Fang et al. "Geometry-enhanced molecular representation learning for property prediction". In: *Nature Machine Intelligence* 4.2 (2022), pp. 127–134.

[7] Yu Rong et al. "Self-supervised graph transformer on large-scale molecular data". In: *Advances in neural information processing systems* 33 (2020), pp. 12559–12571.

[8] Yuyang Wang et al. "Molecular contrastive learning of representations via graph neural networks". In: *Nature Machine Intelligence* 4.3 (2022), pp. 279–287.

[9] Hannes Stärk et al. "3d infomax improves gnns for molecular property prediction". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 20479–20502.

[10] Shengchao Liu et al. "Pre-training molecular graph representation with 3d geometry". In: *arXiv preprint arXiv:2110.07728* (2021).

[11] Gengmo Zhou et al. "Uni-Mol: A Universal 3D Molecular Representation Learning Framework". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: `https://openreview.net/forum?id=6K2RM6wVqKu`.

[12] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[13] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[14] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[15] DeepSeek-AI. "DeepSeek LLM: Scaling Open-Source Language Models with Longtermism". In: *arXiv preprint arXiv:2401.02954* (2024). URL: `https://github.com/deepseek-ai/DeepSeek-LLM`.

[16] Hugo Touvron et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

[17] Haotian Liu et al. "Visual instruction tuning". In: *Advances in neural information processing systems* 36 (2024).

10

[18] Zhe Chen et al. "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks". In: *arXiv preprint arXiv:2312.14238* (2023).

[19] Dingshuo Chen et al. "Uncovering neural scaling laws in molecular representation learning". In: *Advances in Neural Information Processing Systems* 36 (2024).

[20] Keyulu Xu et al. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018).

[21] Kristof Schütt et al. "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions". In: *Advances in neural information processing systems* 30 (2017).

[22] Zaixi Zhang et al. "Motif-based graph self-supervised learning for molecular property prediction". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 15870–15882.

[23] Yi Liu et al. "Spherical message passing for 3d molecular graphs". In: *International Conference on Learning Representations (ICLR)*. 2022.

[24] Zhichun Guo et al. "Graph-based molecular representation learning". In: *arXiv preprint arXiv:2207.04869* (2022).

[25] Jinze Bai et al. "Qwen technical report". In: *arXiv preprint arXiv:2309.16609* (2023).

[26] Albert Q Jiang et al. "Mistral 7B". In: *arXiv preprint arXiv:2310.06825* (2023).

[27] Xiaohua Zhai et al. "Scaling vision transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 12104–12113.

[28] Mostafa Dehghani et al. "Scaling vision transformers to 22 billion parameters". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 7480–7512.

[29] OpenAI. *Sora: Creating video from text*. 2024. URL: https://openai.com/sora.

[30] Yixin Liu et al. "Sora: A Review on Background, Technology, Limitations, and Opportunities of Large Vision Models". In: *arXiv preprint arXiv:2402.17177* (2024).

[31] John J Irwin et al. "ZINC20—a free ultralarge-scale chemical database for ligand discovery". In: *Journal of chemical information and modeling* 60.12 (2020), pp. 6065–6073.

[32] Alexis Conneau and Guillaume Lample. "Cross-lingual language model pretraining". In: *Advances in neural information processing systems* 32 (2019).

[33] Hyung Won Chung et al. "UniMax: Fairer and More Effective Language Sampling for Large-Scale Multilingual Pretraining". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=kXwdL1cWOAi.

[34] Shuqi Lu et al. *Highly Accurate Quantum Chemical Property Prediction with Uni-Mol+*. 2023. arXiv: 2303.16982 [physics.chem-ph].

[35] Chengxuan Ying et al. "Do transformers really perform badly for graph representation?" In: *Advances in neural information processing systems* 34 (2021), pp. 28877–28888.

[36] Yu Shi et al. "Benchmarking graphormer on large-scale molecular modeling datasets". In: *arXiv preprint arXiv:2203.04810* (2022).

[37] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[38] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[39] Paulius Micikevicius et al. "Mixed precision training". In: *arXiv preprint arXiv:1710.03740* (2017).

[40] Sereina Riniker and Gregory A Landrum. "Better informed distance geometry: using what we know to improve conformation generation". In: *Journal of chemical information and modeling* 55.12 (2015), pp. 2562–2574.

[41] Thomas A Halgren. "Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94". In: *Journal of computational chemistry* 17.5-6 (1996), pp. 490–519.

[42] Jared Kaplan et al. "Scaling laws for neural language models". In: *arXiv preprint arXiv:2001.08361* (2020).

[43] Hui Su et al. *Unraveling the Mystery of Scaling Laws: Part I*. 2024. arXiv: 2403.06563 [cs.LG].

[44] Raghunathan Ramakrishnan et al. "Quantum chemistry structures and properties of 134 kilo molecules". In: *Scientific data* 1.1 (2014), pp. 1–7.

[45] Zhenqin Wu et al. "MoleculeNet: a benchmark for molecular machine learning". In: *Chemical science* 9.2 (2018), pp. 513–530.

[46] Alexandra Wahab et al. "The compas project: A computational database of polycyclic aromatic systems. phase 1: cata-condensed polybenzenoid hydrocarbons". In: *Journal of Chemical Information and Modeling* 62.16 (2022), pp. 3704–3713.

[47] Keith T Butler et al. "Machine learning for molecular and materials science". In: *Nature* 559.7715 (2018), pp. 547–555.

11

[48] Han Li et al. "Improving molecular property prediction through a task similarity enhanced transfer learning strategy". In: *Iscience* 25.10 (2022).

[49] Stefan Hougardy. "The Floyd–Warshall algorithm on graphs with negative cycles". In: *Information Processing Letters* 110.8-9 (2010), pp. 279–281.

[50] Uni-Core. *Uni-Core, an efficient distributed PyTorch framework*. 2024. URL: `https://github.com/dptech-corp/Uni-Core`.

[51] Jason Ansel et al. "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. 2024, pp. 929–947.

[52] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

12

A Implementation Details

A.1 Dataset Description

QM9 Dataset   The QM9 dataset [44] is a significant resource in the field of quantum chemistry, offering a single equilibrium conformation and 12 labels that include geometric, energetic, electronic, and thermodynamic properties. For the purpose of performance evaluation, we select the following properties: HOMO, LUMO, gap, alpha, C_v, mu, R^2, and ZPVE. The details of the properties are as follows:

- **HOMO** The HOMO (Highest Occupied Molecular Orbital) is the highest energy molecular orbital that is occupied by electrons in a molecule.

- **LUMO** The LUMO (Lowest Unoccupied Molecular Orbital) is the lowest energy molecular orbital that is not occupied by electrons.

- **gap** The gap, often referred to as the HOMO-LUMO gap, is the energy difference between the HOMO and LUMO. It is a measure of the energy required to excite an electron from the HOMO to the LUMO.

- **ZPVE** ZPVE (Zero-Point Vibrational Energy) is the energy associated with the vibrational motion of atoms in a molecule at absolute zero temperature.

- α The α value represents the static polarizability of a molecule.

- C_v The C_v (Heat Capacity at Constant Volume) is the amount of heat needed to raise the temperature of a given amount of substance by one degree Celsius at constant volume.

- μ The μ (Dipole Moment) is the measure of the molecule's permanent electric dipole moment.

- R^2 The R^2 (Electronic Spatial Extent) is defined as the expectation value of the square of the electronic distance from the nucleus.

COMPAS-1D Dataset   The COMPAS-1D dataset is a part of the COMPAS Project, which is an acronym for the computational Database of Polycyclic Aromatic Systems. The dataset is specifically focused on data-condensed poly-benzenoid hydrocarbons, which are a type of polycyclic aromatic hydrocarbons (PAHs) with a unique structure where the benzene rings are connected edge-to-edge. The COMPAS-1D [46] contains 8,678 molecules and offers essential computational properties crucial for comprehending the behavior of polycyclic aromatic hydrocarbons and other organic molecules across various chemical and physical processes. The details of the properties used in the downstream tasks are as follows:

- **aEA** aEA (Adiabatic Electron Affinity) measures the tendency of a molecule to gain an electron.

- **aIp** aIP (Adiabatic Ionization Potential) measures the energy required for a molecule to lose an electron.

- **Dispersion** Dispersion describes weak inter-molecular forces important for understanding molecular interactions.

- **Dipmom Debye** Dipmom in Debye indicates the polarity of a molecule, affecting its interactions and solubility.

A.2 Atom and Bond Feature for Molecules

The molecular feature used in Uni-Mol2 contains two parts: 1) Atom and bond features, we use RDkit to generate these atom and bond features as input of Uni-Mol2. The detailed features are listed in Table 7 and Table 8. 2) Shortest path $SPD_{i,j}$. We employ the Floyd-Warshall algorithm[49] to calculate the shortest distances between each pair of connected atoms.

A.3 Hyperparameter Settings

In line with previous methods, we employ grid search to find the optimal hyper-parameters for tasks within the QM9 and COMPAS-1D datasets. The specific hyper-parameters are detailed in Table 9. In all experiments, we select the checkpoint with the lowest validation loss and report the corresponding test set results based on that checkpoint. For the COMPAS-1D dataset, experiments were conducted

13

Table 7: Atom features

features	size	description
atom type	119	type of atoms including C, N, O, etc, by atomic number
chirality	6	type of chirality like Tetrahedral chirality
degree	11	the degree of an atom in molecule
formal charge	11	integer electronic charge assigned to atom
number of H	9	number of bonded hydrogen atoms
number of radical electrons	5	number of radical electrons
hybridization	5	SP, SP2, SP3, SP3D, SP3D2
aromaticity	1	whether an atom is part of an aromatic system
in ring	1	whether an atom is within a ring structure

Table 8: Bond features

features	size	description
bond type	4	SINGLE, DOUBLE, TRIPLE, AROMATIC
stereo	6	NONE, Z, E, CIS, TRANS, ANY,
conjugated	1	whether the bond is conjugated

using a single A100 GPU, whereas for the QM9 dataset, the experiments were run on eight A100 GPUs.

Table 9: Hyper-paramters for fine-tuning on QM9 and COMPAS-1D Dataset

Hyperparameter	Value or description
Learning rate	[4e-5, 6e-5, 1e-4, 2e-4, 3e-4, 4e-4]
Batch size	[32, 64, 128]
Epochs	[40, 60, 80, 100, 200, 300]
Pooler dropout	[0.0, 0.1]
Warmup ratio	[0.0, 0.06, 0.1]

A.4 Evaluation Metrics

Diverse evaluation metrics can better help us understand and evaluate the effectiveness of our model. In this section, we introduce the evaluation metrics used in this study. Given n samples, where y_i is the actual value and $\hat{y}_i$ is the predicted value.

Mean Absolute Error (MAE) calculates the average of the absolute differences between predicted and actual values in regression tasks, treating errors of different scales equally.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{12}$$

Relative Mean Absolute Error (RMAE) measures the average absolute prediction error relative to the actual values, providing a dimensionless indication of model accuracy. By normalizing with the actual values, it removes the effect of the data scale, making it possible to compare data with different scales.

$$\text{RMAE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{|y_i|} \tag{13}$$

Mean Square Error (MSE) calculates the average of the squared differences between predicted and actual values, heavily penalizing larger errors.

$$\text{MSE} = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n} \tag{14}$$

14

R-squared measures the proportion of variance in the dependent variable that can be predicted by the independent variables, highlighting the goodness of fit for a regression model. A higher R-squared indicates that the independent variables explain a significant portion of the variance in the dependent variable, while a lower R-squared indicates that the model explains less.

$$\text{R-squared} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{15}$$

The Pearson Correlation Coefficient (r) measures the linear correlation between two variables, ranging from -1 to 1. Larger absolute values signify a stronger linear relationship between the two variables, while values near 0 indicate a weak or non-existent linear relationship.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{16}$$

B Extra QM9 Property and ADME Experiment

B.1 Additional QM9 Property Experiment Results

We have undertaken additional experiments utilizing the QM9 dataset to assess molecular properties 10. Our findings indicate that the model's performance consistently improves with an increase in model size. This scalability suggests that employing larger models will further enhance the accuracy and reliability of molecular property predictions, thereby increasing the utility of our approach for practical applications in the field.

Table 10: Mean absolute error(MAE, ↓) results on QM9 Dataset

Model	HOMO / LUMO / GAP	alpha	C_v	mu	R^2	ZPVE	U_0	U	G	H
Uni-Mol2 310M	0.0036(1e-05)	0.315(0.003)	**0.143(0.002)**	0.092(0.0013)	4.672(0.245)	0.0005(1e-05)	6.149(0.161)	6.009(0.217)	5.564(0.124)	6.3547(0.1048)
Uni-Mol2 570M	0.0036(2e-05)	0.315(0.004)	0.147(0.0007)	0.089(0.0015)	4.523(0.080)	0.0005(3e-05)	5.421(0.432)	5.396(0.384)	5.178(0.1842)	5.9046(0.23469)
Uni-Mol2 1.1B	**0.0035(1e-05)**	**0.305(0.003)**	0.144(0.002)	**0.089(0.0004)**	**4.265(0.067)**	**0.0005(8e-05)**	**4.512(0.188)**	**5.774(0.165)**	**3.921(0.076)**	**5.880(0.293)**
Improvment	2.78%	3.17%	-0.6%	3.26%	8.71%	0%	26.6%	3.9%	29.5%	7.5%

B.2 Biogen ADME Dataset

The Biogen ADME dataset focuses on the evaluation of drug metabolism and pharmacokinetics (DMPK) properties, specifically assessing absorption, distribution, metabolism, and excretion (ADME) characteristics of potential drug candidates. Table 11 illustrates the predictive performance of the Uni-Mol2 model regarding these ADME properties.

Table 11: Mean absolute error(MAE, ↓) results on Biogen ADME Dataset

Model	HCLint-1	PERM-1	SOLU-1
Uni-Mol	0.3085(0.003)	0.2886(0.008)	0.3167(0.010)
Uni-Mol2 84M	0.3117(0.007)	0.2853(0.006)	0.325(0.006)
Uni-Mol2 1.1B	**0.3045(0.003)**	**0.2795(0.00291))**	**0.3062(0.005)**

C Infrastructures

We utilize an efficient distributed PyTorch framework called Uni-Core [50], specifically designed for swiftly developing high-performance PyTorch models [51], particularly those based on Transformer architectures[52]. Given the variability in molecule lengths, padding inputs to match the maximum molecular length is necessary during training. Consequently, the batch size for model training is influenced by the longest molecule in each batch. However, since molecule lengths follow a long-tail distribution (with the majority falling within a specific range), we employ dynamic batching techniques to enhance GPU utilization. By adjusting batch sizes according to the maximum lengths of different batches, we can significantly boost GPU utilization with minimal effort.

15

The time consumption of reading data from distributed storage is often overlooked. We employ a singular, dedicated process on each computational node to asynchronously replicate the training dataset of each epoch onto the host machine. This strategy effectively mitigates time overheads, thereby obscuring the duration spent on data reading from distributed storage. To resume the corruption due to the infra and other factors effectively, we save model weight and optimizer state for every 1k step asynchronously. This means we will lose 1k step training resources in the worst case of hardware instability or loss spike during training. Meanwhile, any checkpoints exceeding the most recent ten files will be deleted to avoid consuming too much storage space.

D   Pretrain Time Complexity and GPU Resources

We utilized a computational cluster comprising 64 NVIDIA A100 GPUs, each equipped with 80GB of HBM2 memory. The GPUs were interconnected via a high-speed Nvidia Infiniband fabric, offering 400 Gbps bandwidth for inter-GPU communication. The details of each model size are listed in table 12.

Table 12: Training Time of Uni-Mol2 at different scale

Params	Compute Resouce (GPUs)	Training Time (GPU hours)
84M	32	2585.6
164M	32	5120
310M	32	7680
570M	64	13824
1.1B	64	30720

E   Limitations

The major limitation of our study pertains to the absence of an exploration of the optimal batch size and learning rate. Our investigation primarily focuses on analyzing and delineating the power-law relationships among validation loss, model size, dataset size, and computational resources. The predictive accuracy of performance aligns well with the scaling curve, indicating that the current optimal learning rate and batch size approximate the near-optimal values. However, existing research suggests a progressive increase in the optimal batch size with augmented computing resources, while the optimal learning rate tends to decrease gradually. It is necessary to note that as we further increase the model's parameters, the final optimal values for learning rate and batch size may fall outside the currently identified range. Consequently, investigating the scaling law for optimal batch size and learning rate is also paramount.

16

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims articulated in the abstract and introduction accurately mirror the paper's contributions and scope.

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper outlines the limitations of the research and discusses the future direction to explore the limitations.

17

Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

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

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides comprehensive details necessary for reproducing the main experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

18

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code, model, and data are made publicly available upon acceptance.

   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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

19

Justification: The paper meticulously details all aspects of the training and testing process. It specifies the data splits used for training, validation, and testing, along with a thorough description of the hyperparameters and their selection criteria.

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: [Yes]

Justification: he paper appropriately reports error bars. All the results reported with the mean and standard deviation by the results of 3 random seeds.

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

8. **Experiments Compute Resources**

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

Answer: [Yes]

Justification: The paper provided the detailed information of compute resources.

Guidelines:

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

9. **Code Of Ethics**

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

20

Answer: [Yes]

Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

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

10. **Broader Impacts**

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

Answer: [NA]

Justification: There is no societal impact of the work performed in this paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

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

12. **Licenses for existing assets**

21

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

Answer: [NA]

Justification: The paper does not use existing assets.

Guidelines:
- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

Answer: [NA]

Justification: The paper does not release new assets.

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

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

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

Answer: [NA]

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

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

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

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

22

Answer: [NA]

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

Guidelines:

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

23