Multi-Scale Protein Language Model for Unified Molecular Modeling

Kangjie Zheng [*] [◇] **Siyu Long** [*] [♡] **Tianyu Lu** [†] **Junwei Yang** [◇]
Xinyu Dai [♡] **Ming Zhang** [◇] **Zaiqing Nie** [‡] **Wei-Ying Ma** [‡] **Hao Zhou** [‡]

Abstract

Protein language models have demonstrated significant potential in the field of protein engineering. However, current protein language models primarily operate at the residue scale, which limits their ability to provide information at the atom level. This limitation prevents us from fully exploiting the capabilities of protein language models for applications involving both proteins and small molecules. In this paper, we propose *ms*-ESM (multi-scale ESM), a novel approach that enables multi-scale unified molecular modeling. *ms*-ESM achieves this by pre-training on multi-scale code-switch protein sequences and utilizing a multi-scale position encoding to capture relationships among residues and atoms. Experimental results indicate that *ms*-ESM surpasses previous methods in protein-molecule tasks, demonstrating the full utilization of protein language models. Further investigations reveal that through unified molecular modeling, *ms*-ESM not only gains molecular knowledge but also retains its understanding of proteins.

1. Introduction

Protein language models (PLMs) have demonstrated significant potential in protein engineering, enabling the capture of biochemical and co-evolutionary knowledge during the pre-training of large-scale protein sequences. This has resulted in remarkable achievements across various domains, including protein structure prediction (Wu et al., 2022; Fang et al., 2022b), protein fitness prediction (Mardikoraem & Woldring, 2023; Notin et al., 2022), protein design (Zheng et al., 2023; Ferruz et al., 2022), etc. For instance, ESM (Rives et al., 2021; Lin et al., 2022b), a widely used PLM, has served as the foundation for several significant models, including ESM-Fold (Lin et al., 2023) for precise protein structure prediction and LM-Design (Verkuil et al., 2022; Hie et al., 2022) for designing proteins with given target functions.

Current PLMs primarily operate at the *protein residue* (amino acid) *scale*, which does not provide information at the *atom scale*. In such circumstances, the potential of PLMs cannot be fully exploited to benefit applications involving both macromolecules (proteins) and small molecules, both of which are vital for various downstream applications.[1] Thus, external small molecule models must be included to address these applications. However, proteins are also composed of atoms, and modeling proteins solely at the residue scale may result in low resolution, meaning that it might not capture information at the atom scale. Intuitively, extending PLMs to operate at both residue and atom scales would make them applicable to a larger range of applications.

Nevertheless, the development of multi-scale PLMs poses significant challenges. First, achieving *unified molecular modeling* that operates effectively at both the residue and atom scales is a challenging task, due to the incompatible vocabularies used at these two different scales. One potential approach to incorporate atomic information into PLMs is to represent and pre-train proteins at the atom scale instead of the original residue-scale pre-training. However, it should be noted that a typical protein can consist of thousands of residues, containing hundreds of thousands of atoms, making such an approach inefficient for modeling. Second, designing an appropriate position encoding to accurately describe the relationships among residues and atoms within the same protein is also non-trivial, which involves relationships varying from residues to residues, residues to atoms, and atoms to atoms.

[*]Equal contribution [◇]School of Computer Science, National Key Laboratory for Multimedia Information Processing, Peking University [♡]School of Artificial Intelligence, National Key Laboratory for Novel Software Technology, Nanjing University [†]Department of Computer Science, Tsinghua University [‡]Institute for AI Industry Research (AIR), Tsinghua University. This work was done during the internship of Kangjie, Siyu, Tianyu, and Junwei at AIR.

Preprint. Copyright 2024 by the author(s).

[1]These applications are widespread in the fields of chemistry and biology and are consistently pivotal for specific scientific breakthroughs. For instance, drug discovery aims to identify small molecules capable of binding to protein pockets (Anderson, 2003; Batool et al., 2019), while enzyme engineering seeks to find enzymes (a special type of protein) that can efficiently catalyze molecular reactions (Mazurenko et al., 2019; Kroll et al., 2023a).

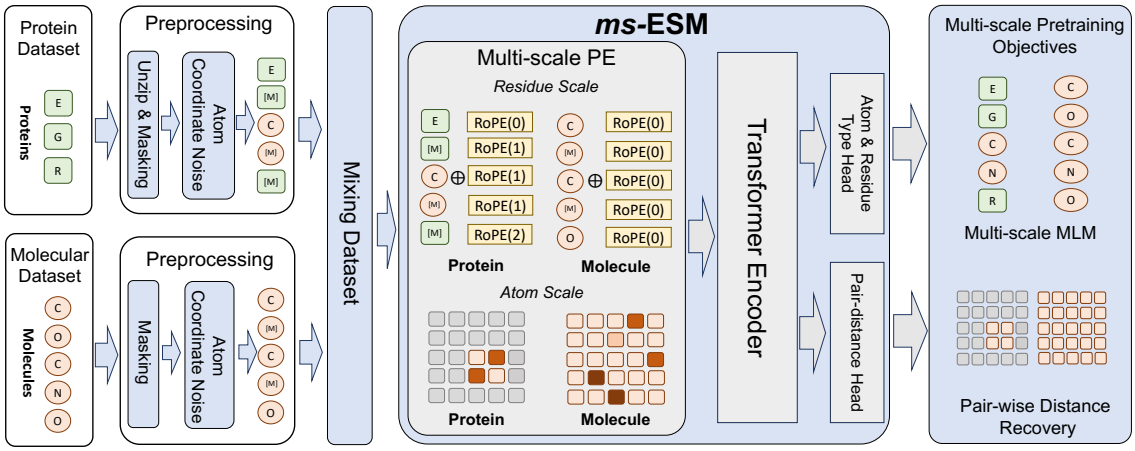


Figure 1. Overview of our multi-scale pre-training process. We mix protein datasets and molecular datasets to train *ms*-ESM. It is worth noting that the model's input is either a molecule or a protein, but not paired protein-molecule data.

To tackle the aforementioned challenges, in this paper, we propose *ms*-ESM (multi-scale ESM), which achieves multi-scale unified molecular modeling through (i) pre-training on multi-scale *code-switch protein sequences* and (ii) describing relationships among residues and atoms using a *multi-scale position encoding*.

First, drawing inspiration from the concept of multilingual code-switching in machine translation (Yang et al., 2020; Li et al., 2022a),[2] *ms*-ESM introduces the learning of multi-scale knowledge through pre-training on multi-scale code-switch protein sequences, obtained by randomly unzipping protein residues into their constituent atoms. In such a scenario, *ms*-ESM can not only capture multi-scale aligned knowledge but also efficiently handle inputs at both the residue and atom scales simultaneously.

Second, *ms*-ESM employs a multi-scale position encoding to comprehensively differentiate between residues and atoms within the code-switch protein sequence. At the residue scale, we extend the original position encoding used in ESM to align with the current best practices in handling pure residue sequences, thereby avoiding ambiguous positional information across different scales, including atom-to-atom, residue-to-residue, and residue-to-atom relationships. At the atom scale, to describe the relationships among unzipped atoms, we employ a spatial distance matrix that directly encodes their 3D positions. With this approach, we can effectively describe all relationships among the entities within the code-switch sequence.

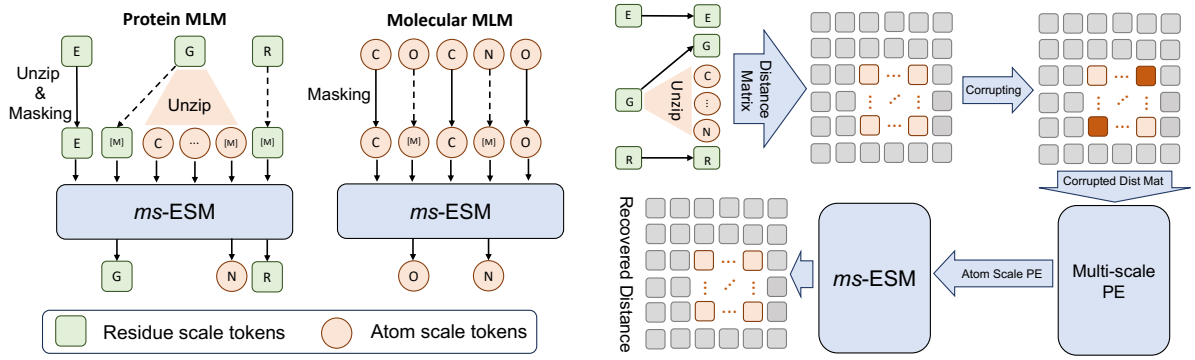
The improved experiment results demonstrate that *ms*-ESM surpasses previous methods in protein-molecule tasks, indicating the full utilization of protein language models. The solid performance in protein tasks suggests that *ms*-ESM, facilitated by the novel unified molecular modeling we first proposed, acquires molecular knowledge without sacrificing its understanding of proteins. Additionally, when applying *ms*-ESM to standard molecular benchmarks, it also outperforms several molecule-specific models. These findings clearly highlight the potential of unified molecular modeling.

2. Proposed Method: *ms*-ESM

In this section, we will describe our multi-scale pre-training model, i.e., *ms*-ESM, in detail. Inspired by the concept of multi-lingual code-switching methods, *ms*-ESM initially generates multi-scale code-switch protein sequences by randomly unzipping partial residues. Through training on these sequences with carefully designed multi-scale position encoding, *ms*-ESM demonstrates its efficacy at both the residue and atom scales. When addressing protein-molecule tasks, i.e., tasks involving both proteins and small molecules, *ms*-ESM does not require any additional models and can fully leverage the potential of pre-training.

Specifically, in Section 2.1, we introduce the overall objective of training *ms*-ESM. Subsequently, in Section 2.2, we delve into the details of constructing a code-switch protein sequence and implementing the multi-scale pre-training approach. To describe the complicated position relationships within the code-switch sequence, we present our design of a multi-scale position encoding in Section 2.3.

[2]They create sentences that switch between two or more languages to help the model learn multilingual knowledge. Yang et al. (2020) enhance multilingual model capabilities by substituting words in the source sentence with their translations in the target language. Similarly, Li et al. (2022a) improve these abilities by replacing a source word or phrase with its counterpart in a different language and then masking the corresponding target word. Collectively, these studies demonstrate that such code-switching techniques significantly strengthen the multilingual capabilities of machine translation models.



(a) Multi-scale MLM

(b) Pair-wise Distance Recovery

Figure 2. Framework of multi-scale pre-training comprises multi-scale masked language modeling and pairwise distance recovery.

2.1. Overview

We start with an overview of our multi-scale pre-training model, i.e., *ms*-ESM (Figure 1). Briefly, the total objective of our pre-training can be expressed as the following loss function:

$$\begin{aligned}\mathcal{L}_\theta &= \sum_{X_i \in D} \mathcal{L}_{\text{MLM}}(\bar{X}_i, E_i; \theta) + \mathcal{L}_{\text{PDR}}(\bar{X}_i, E_i; \theta) \\ &= \sum_{X_i \in D} \mathcal{L}_{\text{MLM}}(\text{UNZIP}(X_i), \text{MSPE}(X_i); \theta) + \\ &\qquad \mathcal{L}_{\text{PDR}}(\text{UNZIP}(X_i), \text{MSPE}(X_i); \theta)\end{aligned}$$

For each data X_i in dataset D, we first create its code-switch sequence $\bar{X}_i$ by unzipping partial residues. Using the code-switch sequence, we employ Masked Language Modeling (MLM) and Pair-wise Distance Recovery (PDR) as pre-training tasks. We discuss the details of $\bar{X}_i$, $\mathcal{L}_{\text{MLM}}$, and $\mathcal{L}_{\text{PDR}}$ in Section 2.2. To account for the coexistence of residues and atoms in the sequence, we propose a Multi-Scale Position Encoding (MSPE) E_i to describe the complicated position relationship within $\bar{X}_i$ (Section 2.3). We show more details of *ms*-ESM, including the parameterization of θ in Section 2.4. Notably, since we utilize molecule data in pre-training, *ms*-ESM can accept either proteins or molecules as inputs.

2.2. Multi-scale Pre-training

In this section, we elaborate how to create a code-switch protein sequence $\bar{X}$ and implement the pre-training tasks, i.e., MLM and PDR, on it (Figure 2).

Code-Switch Protein Sequence Briefly, the concept of constructing a code-switch protein sequence is inspired by the multilingual code-switching technique in machine translation (Yang et al., 2020; Li et al., 2022a). This technique, which constructs sentences that switch between multiple languages, has significantly enhanced the model's capability to handle multilingual tasks. In our multi-scale unified molecular modeling, we treat residues and atoms as different "languages" and construct sequences that switch between residues and atoms, thereby augmenting the model's capability to handle downstream tasks.

Specifically, in the residue scale, a protein X can be seen as a sequence of L residues, i.e., $X = (r_1, \cdots, r_i, \cdots, r_L)$. Each residue r_i further consists of a specific set of N atoms $A_i = \{a_i^1, \cdots, a_i^N\}$. To construct a code-switch protein sequence $\bar{X}$, we randomly select a group of residues and insert their corresponding atoms into X, which is essentially an unzipping process. For each unzipped residue, we provide the model with structural information of the residue at the atomic scale, thus offering the model very diverse structural knowledge. In particular, during the unzipping process, we assign a sequential order to the unzipped atoms. Here, we take the case of unzipping a single residue as an example, whereas in actual modeling, multiple residues can be unzipped. After inserting the atom set A_i into X, i.e., unzipping the residue r_i, we obtain a code-switch sequence

$$\begin{aligned}\bar{X} &= (r_1, \cdots, r_i, \text{ORDER}(A_i), \cdots, r_L) \\ &= (r_1, \cdots, r_i, a_i^1, \cdots, a_i^N, \cdots, r_L) \\ &= (h_1, \cdots, h_i, h_{i+1}, \cdots, h_{i+N}, \cdots, h_{L+N})\end{aligned}$$

where ORDER is the order assigned to the atom set (Appendix A). h_i represents either a single residue or an individual atom in $\bar{X}$. We also denote all the atoms in $\bar{X}$ as $\bar{A}$ and all the residues as $\bar{R}$.

Notably, when we insert the atom set A_i of residue r_i, we still retain r_i. This allows the model to attend either to the corresponding residue-scale information or to the surrounding atom-scale information when predicting masked atoms and encourages the model to align residue-scale and atom-scale representations, similar to the approach in cross-lingual pre-training (Conneau & Lample, 2019). We provide an illustration of the code-switch sequence in Figure 2.

Masked Language Modeling After obtaining the code-switch sequence $\bar{X}$, we can implement MLM on it. Unlike the MLM used in ESM, which only masks residues, our approach masks both residues and atoms and requires models to predict them. Specifically, we start by randomly masking a portion of the atoms or residues in $\bar{X}$ and then ask the model to predict the original atoms or residues using the surrounding context.

$$\mathcal{L}_{\theta\text{MLM}} = -\sum_{h \in \text{MASK}(\bar{X})} \log p_\theta(h|\bar{X} \backslash \text{MASK}(\bar{X}))$$

where $\text{MASK}(\bar{X})$ represents the set of masked atoms and residues. $\bar{X} \backslash \text{MASK}(\bar{X})$ denotes the unmasked context. h is a single masked atom or residue. Figure 2a is the framework of MLM task.

Pair-wise Distance Recovery We also employ PDR as another pre-training task. Briefly, we use corrupted atoms as model input and ask model to recover the accurate Euclidean distances between these atoms. We corrupt the atoms by adding noises to their coordinates. Specifically, we replace the ground-truth coordinate with a randomly selected position that is within a certain range (Euclidean distances $< \epsilon$, Appendix A) of the true coordinate. Models are required to reconstruct the actual distances based on the corrupted coordinates. To avoid introducing residue-residue interactions that are very different from the interactions in small molecules, we only calculate PDR within residues, which can also make *ms*-ESM learn very diverse structural knowledge of residues.

$$\mathcal{L}_{\theta\text{PDR}} = \sum_{\substack{A_i = A_j \\ h_i, h_j \in \bar{A}, i \neq j \\ c_i = \text{COORD}(h_i) \\ c_j = \text{COORD}(h_j)}} \|\text{DIS}_\theta(c_i + \sigma_i, c_j + \sigma_j) - \text{DIS}(c_i, c_j)\|_2$$

where DIS_θ is the recovered distance and DIS is the ground truth. COORD extracts coordinates from atoms. σ_i, σ_j are the corresponding noises added to atom coordinates c_i, c_j. To elaborate further, these noises will affect the atom scale position encoding in Section 2.3. Figure 2b shows the framework of PDR task.

Notably, when training *ms*-ESM, we mix up a protein dataset D_p and a molecule dataset D_m as the final dataset, i.e., $D = D_p \cup D_m$. For a molecule from D_m, as it consists solely of atoms, its code-switch sequence $\bar{X}$ is the ordered set of all its atoms $\bar{A}$, and it does not have any residues, i.e., $\bar{R} = \emptyset$.

2.3. Multi-scale Position Encoding

Encoding the position relationship in the code-switch sequence is challenging. Given that both residues and atoms are present in the code-switch sequence, it is crucial for the position encoding to accurately represent the positional relationships. This includes relationships between residues, between atoms, and between residues and atoms, regardless of whether the atoms are part of the same residue. This situation is more complex than dealing with pure residue sequences. Because previous encodings in PLMs are only designed for residue sequences, they can not describe the relationships that extend from residues to atoms, and among atoms.

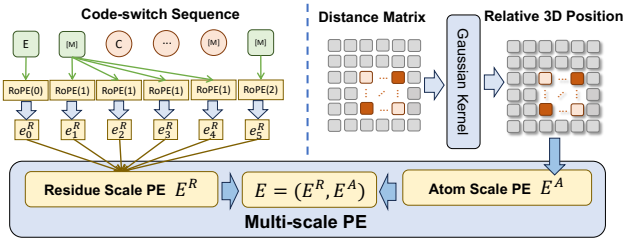


Figure 3. Framework of multi-scale position encoding.

In this section, we design a multi-scale position encoding E to encode the positional relationships within a code-switch sequence. Specifically, E contains a residue scale position encoding E^R and an atom scale position encoding E^A, i.e., $E = (E^R, E^A)$. For E^R, we carefully extend an existing encoding method, allowing it to encode relationships from residues to atoms, while maintaining consistency with the original encoding when handling pure residue sequences. For E^A, to capture the relationships among atoms, we directly encode their 3D positions using a spatial distance matrix. The multi-scale encoding approach ensures that no ambiguous positional relationships affect the pre-training, enabling *ms*-ESM to perform effectively in both scales. Figure 3 illustrates the framework of our multi-scale position encoding. We will provide detailed explanations for each of them in the following paragraphs.

Residue Scale Position Encoding We design the residue scale position encoding E^R following two principles: (i) For encoding the relationship between two residues, E^R should be consistent with the mainstream encoding method. (ii) For atoms from the same unzipped residue, E^R should not introduce any ambiguous position information.

As previous PLMs have shown the effectiveness of the mainstream encoding method in handling pure residue sequences, it is prudent for E^R to maintain consistency with it. Furthermore, when dealing with two atoms from the same residue, since we cannot define residue scale positional relationships within the residue, it is important for E^R to avoid the impact of such ill-defined information.

In particular, we use Rotary Position Embedding (RoPE) (Su et al., 2021), the original position encoding in ESM-2,

to describe the position relationship among the residues in a code-switch sequence. For assigning the position encoding to an atom in the code-switch sequence, we reuse the position encoding of the residue to which the atom belongs. In cases where the atom belongs to a small molecule, not a residue, we assign a fixed position encoding (RoPE(0) in our paper) to it. Formally, for a code-switch sequence $\bar{X}$, its residue scale position encoding $E^R = (e_1^R, \cdots, e_{L+N}^R)$ can be obtained according to the following formulation:

$$e_i^R = \begin{cases} \text{RoPE}(j) & h_i \in \bar{R}, h_i = r_j \\ \text{RoPE}(k) & h_i \in \bar{A}, \exists k, h_i \in A_k \\ \text{RoPE}(0) & \text{otherwise} \end{cases}$$

By adopting such encoding strategy, E^R satisfies the two aforementioned principles. Specifically, for pure residue sequences, E^R is equivalent to RoPE. When handling atoms from the same residue, the relative nature of RoPE ensures that no ambiguous information will impact the pre-training model. For more details about the properties of RoPE, please refer to Su et al. (2021).

Atom Scale Position Encoding Because E^R will not provide the position encoding for atoms from the same residue, we need an atom scale position encoding E^A to describe the relationship from atoms to atoms. As suggested by Zhou et al. (2023), we use Euclidean distance matrix and Gaussian kernel GAUSSIAN to encode the 3D position of atoms.

For $h_i, h_j \in \bar{X}$, their atom scale position encoding e_{ij}^A can be calculate as follows:

$$e_{ij}^A = \begin{cases} 0 & h_i \in \bar{R} \text{ or } h_j \in \bar{R} \\ \text{GAUSSIAN}(\text{DIS}(c_i, c_j)) & \text{otherwise} \end{cases}$$

where $c_i = \text{COORD}(h_i), c_j = \text{COORD}(h_j)$. We refer readers to Zhou et al. (2023) for more details of this 3D position encoding.

2.4. Integrating Multi-scale PE into Transformer

The parameterization θ of *ms*-ESM is slightly different from the original Transformer architecture proposed by Vaswani et al. (2017). To be specific, we begin by substituting the sinusoidal encoding in the Transformer with our residue scale position encoding E^R. For the atom scale position encoding E^A, we treat it as the bias term of self-attention layers (Luo et al., 2022; Zhou et al., 2023). The self-attention in *ms*-ESM can be calculated like:

$$\text{ATTENTION}(Q, K, V, E^A) = \text{SOFTMAX}(\frac{QK^T}{\sqrt{d_k}} + E^A)V$$

where Q, K, V are the query, key, and value corresponding to $\bar{X}$. We refer readers to Vaswani et al. (2017) for more details of the original Transformer. With only slight modifications to the original Transformer architecture, *ms*-ESM is capable of simultaneously processing residues and atoms, making it a versatile model for various downstream tasks. Moreover, *ms*-ESM shows great compatibility with existing pre-training model, e.g., ESM series, which allows us to bulid up a better model based on previous study more easily.

3. Experiments

To verify the effectiveness of the multi-scale unified pre-training, we evaluated the performance of *ms*-ESM in various protein-molecule tasks, i.e., tasks involving both proteins and small molecules (Section 3.1). The improved results indicate that our unified molecular modeling approach can take good advantage of the full potential of protein language models.

We also conduct ablation studies on position encoding, training data, and pre-training objectives to discuss how these elements affect the performance of our model (Section 3.2). Collectively, these studies demonstrate that each of these components plays a crucial role in *ms*-ESM.

In addition, we use multiple protein tasks to evaluate *ms*-ESM's understanding of proteins (Section 3.3). The solid performance in these protein tasks suggests that *ms*-ESM does not sacrifice its protein understanding at both local and global level.

Further applying *ms*-ESM to standard molecular benchmarks (Section 3.4), we observe that *ms*-ESM can outperform several molecule-specific methods.

The detailed pre-training configuration, including pre-training datasets and hyperparameters, e.g., the unzip ratio of residues, can be found in Appendix A.

3.1. Main Results

We use tasks involving both proteins and molecules to prove that *ms*-ESM can operate at both residue and atom scales and our unified molecular modeling approach can exploit the full potential of PLMs.

Fine-tuning For protein-molecule tasks, we follow the benchmark protocol from ProSmith (Kroll et al., 2023b) to evaluate *ms*-ESM on three tasks, including enzyme-substrate affinity regression, drug-target affinity regression, and enzyme-substrate pair classification. Specifically, each task provides the protein residue sequence and the molecule SMILES string as input and requires models to determine whether the protein-molecule pair exhibits high affinity. Since our *ms*-ESM cannot directly process SMILES strings, we initially employ RDKit (Landrum et al., 2013) to generate the corresponding molecule conformations based on the SMILES representation. Subsequently, we extract the atom sequence and atom scale position encoding for *ms*-ESM.

Table 1. Performance comparison on Enzyme-Substrate Affinity Regression (ESAR) task and Enzyme-Substrate Pair Classification (ESPC) task. *ms*-ESM outperforms other models and achieves the state-of-the-art results, which indicates that *ms*-ESM operate at both the residue and atom scales successfully and our unified modeling harness the full potential of PLMs.

Method	Protein Pre-training	Molecule Pre-training	ESAR			ESPC		
			MSE ↓	R^2 ↑	Pearson ↑	ACC ↑	MCC ↑	ROC-AUC ↑
Gollub et al. (2023)	/	/	/	0.463	0.680	/	/	/
Kroll et al. (2021)	/	/	0.653	0.527	0.728	/	/	/
Baseline $_{\text{XGBoost}}$	ESM-2 35M	Uni-Mol 48M	0.652	0.528	0.727	89.9%	0.729	0.941
Baseline $_{\text{ProSmith}}$	ESM-2 35M	Uni-Mol 48M	0.642	0.536	0.733	90.8%	0.754	0.943
Ours $_{\text{XGBoost}}$	*ms*-ESM 35M	*ms*-ESM 35M	0.623	0.548	0.742	90.6%	0.750	0.943
Ours $_{\text{ProSmith}}$	*ms*-ESM 35M	*ms*-ESM 35M	**0.599**	**0.566**	**0.753**	**91.8%**	**0.781**	**0.954**

Table 2. Performance comparison on drug-target affinity regression task. *ms*-ESM achieves the state-of-the-art results on most metrics.

Method	Protein Pre-training	Molecule Pre-training	MSE ↓	CI ↑	r_m^2 ↑
Öztürk et al. (2018)	/	/	0.261	0.878	0.630
Shin et al. (2019)	/	Molecule Transformer	0.245	0.887	0.665
Nguyen et al. (2021a)	/	/	0.229	0.893	0.685
Nguyen et al. (2021b)	TAPE 38M	/	0.228	0.893	/
Qiu et al. (2021)	ProtBert 420M	/	0.205	0.896	0.709
Kao et al. (2021)	/	/	0.202	0.907	/
Yuan et al. (2022)	ESM-1b 650M	/	0.208	**0.913**	0.743
Yang et al. (2022)	/	/	0.207	0.900	0.710
Baseline $_{\text{XGBoost}}$	ESM-2 35M	Uni-Mol 48M	0.261	0.885	0.652
Baseline $_{\text{ProSmith}}$	ESM-2 35M	Uni-Mol 48M	0.219	0.899	0.711
Ours $_{\text{XGBoost}}$	*ms*-ESM 35M	*ms*-ESM 35M	0.248	0.889	0.668
Ours $_{\text{ProSmith}}$	*ms*-ESM 35M	*ms*-ESM 35M	**0.191**	0.906	**0.759**

For additional fine-tuning details, please refer to Appendix B.1.

Baselines We compare *ms*-ESM with multiple baselines on each tasks, including both supervised and pre-training baselines. For each baseline, we list their protein pre-training model and molecule pre-training model in corresponding tables. More details of each baseline can be seen in corresponding papers. Since ProSmith (Kroll et al., 2023b) is the only framework that offers a method to integrate protein pre-training models and molecule pre-training models, we adopt their framework and replace both the protein and molecule models with *ms*-ESM to ensure a fair comparison. We also provide an XGBoost (Chen & Guestrin, 2016) variant of ProSmith, which takes the concatenation of protein and molecule representation as features and can directly assess the compatibility of the two representations

Results Table 1 and Table 2 display the experimental results of *ms*-ESM and baselines for the three tasks. Based on the results, we can summarize our findings as follows: (i) *ms*-ESM outperforms other models and achieves the state-of-the-art results on most metrics. (ii) Fine-tuning strategies such as ProSmith and XGBoost, when built upon our *ms*-ESM, consistently outperform versions that combine two separate pre-training models (as shown in the last four rows of both Table 1 and Table 2). (iii) *ms*-ESM can even beat methods that are based on much larger pre-training models (comparing the 5th and 7th rows to the last row in Table 2).

These findings clearly indicate that *ms*-ESM operate at both the residue and atom scales successfully and **pre-training proteins and molecules in a single model can harness the full potential of pre-training techniques for protein-molecule tasks**. Fusing two separate pre-training models can be suboptimal for such tasks, and the issue cannot be resolved by using larger pre-training models.

3.2. Ablation Study

We have conducted comprehensive ablation studies focusing on various aspects such as position encoding, pre-training objectives, and training data. These studies demonstrate that each of these components plays a crucial role in the efficacy of our method.

Ablation on Multi-scale Position Encoding To validate the effectiveness of multi-scale position encoding, we conduct ablation tests under two conditions: one without using Atom Scale Position Encoding (ASPE) and another without using Residue Scale Position Encoding (RSPE). The

Table 3. Performance comparison on the Contact Prediction (CP) task and Secondary Structure Prediction (SSP) task. *ms*-ESM has similar performance to ESM-2 indicating it preserves the local and global understanding of proteins originally presented in ESM-2.

Method	CP (P@L/5)↑			SSP (ACC)↑	
	Short Range	Medium Range	Long Range	SS3-casp12	SS8-casp12
TAPE 38M	0.46	0.33	0.25	0.71	0.59
ResNet 38M	0.46	0.35	0.17	0.72	0.58
ESM-2 35M	0.46	**0.45**	**0.49**	**0.74**	**0.61**
ms-ESM 35M	**0.48**	**0.45**	0.48	**0.74**	0.60

Table 4. Experimental results on ablation study. The results show that each component contributes to our method.

Method	ESAR	
	MSE ↓	R^2 ↑
w/o ASPE	0.639(+0.012)	0.537(-0.009)
w/o RSPE	0.676(+0.049)	0.511(-0.035)
w/o MLM Loss	0.642(+0.015)	0.535(-0.011)
w/o PDR Loss	0.645(+0.018)	0.533(-0.013)
w/o Molecule Data	0.648(+0.021)	0.531(-0.015)
w/o Protein Data	0.708(+0.081)	0.487(-0.059)
w/o Unzip Operation	0.638(+0.011)	0.538(-0.008)
ms-ESM	**0.627**	**0.546**

employed task is enzyme-substrate affinity regression. As shown in Table 4, when atom scale position encoding or residue scale position encoding is omitted, the model's performance suffers significantly. This is due to the model's inability to capture positional information of atoms and residues in the absence of position encoding. These results prove the effectiveness of our multi-scale position encoding.

Ablation on Pre-training Objectives We observed a substantial decrease in model performance when we omitted either the masked atom type prediction loss or the pairwise distance recovery loss, as demonstrated in Table 4. Notably, the omission of the pairwise distance recovery loss leads to a more substantial performance deterioration compared to the omission of the masked atom type prediction loss. This is likely because, without the pairwise distance recovery loss, *ms*-ESM cannot learn structural information at the atom scale. These results suggest that, while both atom type and structural information are crucial for atom-scale details, structural information is of greater significance.

Ablation on Pre-training Data We observed a significant decrease in model performance when excluding either molecular or protein data, as depicted in Table 4. It is interesting to note that removing protein data results in a more significant performance decline compared to omitting molecule data. This suggests that when the model is not trained with protein data, it rapidly loses protein-related knowledge, leading to a notable drop in overall performance. However, the model can still acquire atomic scale information through unzip operations even without molecule data. Hence, the model performs better without molecule data compared to the scenario without protein data. Furthermore, the model's performance significantly deteriorates when the unzip operation is omitted. These results confirm the effectiveness of the unzip operation.

3.3. *ms*-ESM Preserves the Strong Ability of Protein Understanding

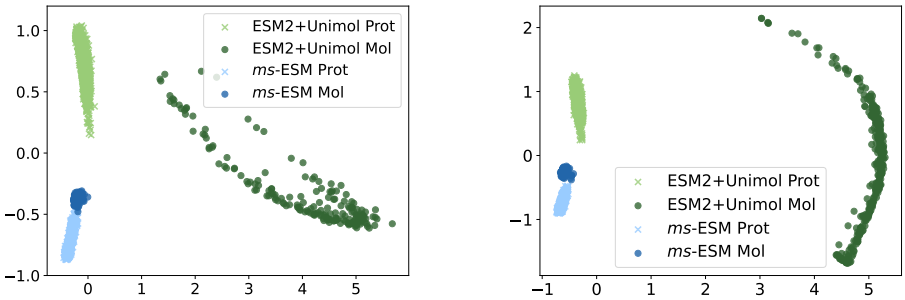
Because *ms*-ESM is developed based on existing PLMs, we would like to determine whether it still preserves a thorough understanding of proteins. Specifically, we follow TAPE (Rao et al., 2019) and use the tasks secondary structure prediction and contact prediction to test the ability of protein pre-training models in protein structure understanding. For secondary structure prediction, models must grasp the local protein structure, such as helices and strands. For contact prediction, models need a comprehensive understanding of proteins at a global level. Notably, both *ms*-ESM and baseline methods have exactly the same input (pure residue sequence) for these two tasks. For more details of the fine-tuning and baselines, readers can find them in Appendix B.2.

We report the results of contact prediction and secondary structure prediction in Table 3. While *ms*-ESM may not achieve the best performance among the compared methods, the tables demonstrate that it performs similarly to ESM-2 in both secondary structure prediction and contact prediction. This indicates that ***ms*-ESM does not sacrifice its understanding of proteins**. Promisingly, *ms*-ESM can achieve improved protein understanding by initializing its parameters with a larger ESM-2. More results on these two protein tasks are shown in Appendix C, specifically in Table 7 and Table 8.

3.4. *ms*-ESM Performs Well on Molecular Benchmarks

We employ molecular benchmarks to evaluate the integrated molecular knowledge within *ms*-ESM. Following Uni-Mol (Zhou et al., 2023), we utilize the standard molecular benchmarks, MoleculeNet (Wu et al., 2018), in this paper. For additional details on fine-tuning and baseline information, please refer to Appendix B.3.

Table 9 in Appendix C shows the experiment results of



(a) Enzyme-substrate Pair Classification (b) Drug-target Affinity Regression

Figure 4. Visualization of representations learned by *ms*-ESM and ESM-2+Uni-Mol.

both molecular property classification and regression tasks. *ms*-ESM is comparable to the Uni-Mol in most tasks and outperforms several molecule-specific models in many instances, which makes it a strong method for molecular tasks.

3.5. Visualization

To provide a more intuitive illustration of the higher quality of protein and small molecule representations learned by *ms*-ESM, we conducted a visual comparison of the representations extracted from *ms*-ESM and ESM-2+Uni-Mol in the tasks of enzyme-substrate pair classification and drug-target affinity regression. Specifically, we use the fine-tuned models, i.e., Baseline $_{\text{ProSmith}}$ and Ours $_{\text{ProSmith}}$ in both Table 1 and Table 2, to extract the representations of proteins and molecules. Subsequently, we employ Principal Component Analysis (PCA) to visualize these representations.

As illustrated in Figure 4, the representations of proteins and molecules learned by the *ms*-ESM model are more closely aligned. This suggests that the *ms*-ESM model is capable of creating a more cohesive semantic representation encompassing both proteins and molecular data, which makes *ms*-ESM outperform two separate pre-trained models.

4. Related Work

Protein Pre-training Pre-training has been proved to be an efficient technique in many domains, like natural language processing and protein engineering. Existing work studies protein pre-training mainly in two ways: (i) Sequence-based methods learn protein primary sequences to capture the biochemical and co-evolutionary knowledge. ESM series models (Rives et al., 2021; Lin et al., 2022b; 2023) use vanilla masked language modeling to learn protein representations on evolutionary scale. Aiming at the specific contact prediction task, Rao et al. (2021) further extends the masked language modeling to multiple sequence alignment (MSA) data. Inspired by the large language model (LLM), ProtGPT2 (Ferruz et al., 2022), ProGen(Madani et al., 2023), and ProGen2 (Nijkamp et al., 2022) scale up the model size of protein language model and show promising results in protein generation tasks. (ii) Structure-based methods directly learn protein structure in different levels. Gligorijević et al. (2021); Zhang et al. (2022); Xu et al. (2022) learn residues from a local part of protein structures. Jing et al. (2020); Zhang et al. (2023) try to capture atomic structure knowledge in proteins. We develop *ms*-ESM based on ESM. Differently, *ms*-ESM is a mixture of sequence and structure-based methods, which gives it the ability to process information from different scales and makes it a versatile model.

Unified Molecular Modeling Because of the huge scale difference between proteins and small molecules, it is challenging to model both of them in a unified style. As far as we know, Uni-Mol (Zhou et al., 2023) is the only method that tries to process proteins and molecules uniformly. Uni-Mol realizes the uniformity by directly modeling proteins and molecules at atom scale. However, because an entire protein contains hundreds of thousands of atoms, Uni-Mol can only model a local structure of proteins, i.e., protein pocket. Unlike Uni-Mol, as *ms*-ESM only unzips partial residues into their corresponding atoms, it can handle an entire protein efficiently. We also provide some discussion of general molecular modeling in Appendix D.

5. Conclusions

In this study, we propose a multi-scale protein language model *ms*-ESM, which realizes multi-scale unified molecular modeling by pre-training on multi-scale code-switch protein sequence and describing relationships among residues and atoms with a multi-scale position encoding. Experiment results show that *ms*-ESM outperforms previous methods in protein-molecule tasks and effectively integrates molecular knowledge into the protein language model without sacrificing the understanding of proteins.

Broader Impact

PLMs have been applied to a wide range of applications, including protein structure prediction, protein fitness prediction, and protein design. Our unified molecular modeling extends the capabilities of PLMs to effectively operate at both the residue and atom scales, thereby enhancing their applicability to these tasks. For instance, our method can serve as the foundation for constructing more accurate protein structure prediction and design models at the atomic level. In addition, our unified molecular modeling has also opened up new avenues for research in the field of protein-small molecule interactions. Novel binding and drug design models can benefit from our method. We also admit that our method inherits the potential negative influence of PLMs. For example, it could be used to design and manufacture proteins and molecules with biological harm.

Acknowledgements

We would like to thank Qiying Yu and Hanlin Wu from AIR for their insightful discussions on the project. We also thank other members from AIR for their valuable feedback given during the internal seminar.

References

AlQuraishi, M. Proteinnet: a standardized data set for machine learning of protein structure. *BMC bioinformatics*, 20(1):1–10, 2019.

Anderson, A. C. The process of structure-based drug design. *Chemistry & biology*, 10(9):787–797, 2003.

Batool, M., Ahmad, B., and Choi, S. A structure-based drug discovery paradigm. *International journal of molecular sciences*, 20(11):2783, 2019.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Chithrananda, S., Grand, G., and Ramsundar, B. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Conneau, A. and Lample, G. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32, 2019.

Cuff, J. A. and Barton, G. J. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519, 1999.

Davis, M. I., Hunt, J. P., Herrgard, S., Ciceri, P., Wodicka, L. M., Pallares, G., Hocker, M., Treiber, D. K., and Zarrinkar, P. P. Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29(11):1046–1051, 2011.

Fang, X., Liu, L., Lei, J., He, D., Zhang, S., Zhou, J., Wang, F., Wu, H., and Wang, H. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022a.

Fang, X., Wang, F., Liu, L., He, J., Lin, D., Xiang, Y., Zhang, X., Wu, H., Li, H., and Song, L. Helixfold-single: Msa-free protein structure prediction by using protein language model as an alternative. *arXiv preprint arXiv:2207.13921*, 2022b.

Fang, Y., Zhang, Q., Yang, H., Zhuang, X., Deng, S., Zhang, W., Qin, M., Chen, Z., Fan, X., and Chen, H. Molecular contrastive learning with chemical element knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3968–3976, 2022c.

Ferruz, N., Schmidt, S., and Höcker, B. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

Gligorijević, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B. C., Fisk, I. M., Vlamakis, H., et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

Gollub, M. G., Backes, T., Kaltenbach, H.-M., and Stelling, J. Enkie: A package for predicting enzyme kinetic parameter values and their uncertainties. *bioRxiv*, pp. 2023–03, 2023.

Guo, Z., Sharma, P., Martinez, A., Du, L., and Abraham, R. Multilingual molecular representation learning via contrastive pre-training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3441–3453, 2022.

Halgren, T. A. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.

Hie, B., Candido, S., Lin, Z., Kabeli, O., Rao, R., Smetanin, N., Sercu, T., and Rives, A. A high-level programming language for generative protein design. *bioRxiv*, pp. 2022–12, 2022.

Honda, S., Shi, S., and Ueda, H. R. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv preprint arXiv:1911.04738*, 2019.

Jiao, R., Han, J., Huang, W., Rong, Y., and Liu, Y. Energy-motivated equivariant pretraining for 3d molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8096–8104, 2023.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J., and Dror, R. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kao, P.-Y., Kao, S.-M., Huang, N.-L., and Lin, Y.-C. Toward drug-target interaction prediction via ensemble modeling and transfer learning. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2384–2391. IEEE, 2021.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Klausen, M. S., Jespersen, M. C., Nielsen, H., Jensen, K. K., Jurtz, V. I., Soenderby, C. K., Sommer, M. O. A., Winther, O., Nielsen, M., Petersen, B., et al. Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, 2019.

Kroll, A., Engqvist, M. K., Heckmann, D., and Lercher, M. J. Deep learning allows genome-scale prediction of michaelis constants from structural features. *PLoS biology*, 19(10):e3001402, 2021.

Kroll, A., Ranjan, S., Engqvist, M. K., and Lercher, M. J. A general model to predict small molecule substrates of enzymes based on machine and deep learning. *Nature Communications*, 14(1):2787, 2023a.

Kroll, A., Ranjan, S., and Lercher, M. J. A multimodal transformer network for protein-small molecule interactions enhances drug-target affinity and enzyme-substrate predictions. *bioRxiv*, pp. 2023–08, 2023b.

Landrum, G. et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8:31, 2013.

Li, P., Wang, J., Qiao, Y., Chen, H., Yu, Y., Yao, X., Gao, P., Xie, G., and Song, S. Learn molecular representations from large-scale unlabeled molecules for drug discovery. *arXiv preprint arXiv:2012.11175*, 2020.

Li, P., Wang, J., Qiao, Y., Chen, H., Yu, Y., Yao, X., Gao, P., Xie, G., and Song, S. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. *Briefings in Bioinformatics*, 22 (6):bbab109, 2021.

Li, P., Li, L., Zhang, M., Wu, M., and Liu, Q. Universal conditional masked language pre-training for neural machine translation. *arXiv preprint arXiv:2203.09210*, 2022a.

Li, S., Zhou, J., Xu, T., Dou, D., and Xiong, H. Geomgcl: Geometric graph contrastive learning for molecular property prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4541–4549, 2022b.

Lin, X., Xu, C., Xiong, Z., Zhang, X., Ni, N., Ni, B., Chang, J., Pan, R., Wang, Z., Yu, F., et al. Pangu drug model: learn a molecule like a human. *bioRxiv*, pp. 2022–03, 2022a.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022b.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023.

Liu, S., Wang, H., Liu, W., Lasenby, J., Guo, H., and Tang, J. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.

Liu, S., Guo, H., and Tang, J. Molecular geometry pretraining with se (3)-invariant denoising distance matching. *arXiv preprint arXiv:2206.13602*, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Luo, S., Chen, T., Xu, Y., Zheng, S., Liu, T.-Y., Wang, L., and He, D. One transformer can understand both 2d & 3d molecular data. *arXiv preprint arXiv:2210.01765*, 2022.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pp. 1–8, 2023.

Mardikoraem, M. and Woldring, D. Protein fitness prediction is impacted by the interplay of language models,

ensemble learning, and sampling methods. *Pharmaceutics*, 15(5):1337, 2023.

Mazurenko, S., Prokop, Z., and Damborsky, J. Machine learning in enzyme engineering. *ACS Catalysis*, 10(2): 1210–1223, 2019.

Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. Critical assessment of methods of protein structure prediction (casp)—round xii. *Proteins: Structure, Function, and Bioinformatics*, 86:7–15, 2018.

Nguyen, T., Le, H., Quinn, T. P., Nguyen, T., Le, T. D., and Venkatesh, S. Graphdta: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37 (8):1140–1147, 2021a.

Nguyen, T. M., Nguyen, T., Le, T. M., and Tran, T. Gefa: early fusion approach in drug-target affinity prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 19(2):718–728, 2021b.

Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N., and Madani, A. Progen2: exploring the boundaries of protein language models. *arXiv preprint arXiv:2206.13517*, 2022.

Notin, P., Dias, M., Frazer, J., Hurtado, J. M., Gomez, A. N., Marks, D., and Gal, Y. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pp. 16990–17017. PMLR, 2022.

Öztürk, H., Özgür, A., and Ozkirimli, E. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

Qiu, Z., Jiao, Q., Wang, Y., Chen, C., Zhu, D., and Cui, X. rzmlp-dta: gmlp network with rezero for sequence-based drug-target affinity prediction. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 308–313. IEEE, 2021.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021.

Riniker, S. and Landrum, G. A. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12):2562–2574, 2015.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

Shin, B., Park, S., Kang, K., and Ho, J. C. Self-attention based molecule representation for predicting drug-target interaction. In *Machine Learning for Healthcare Conference*, pp. 230–248. PMLR, 2019.

Stärk, H., Beaini, D., Corso, G., Tossou, P., Dallago, C., Günnemann, S., and Liò, P. 3d infomax improves gnns for molecular property prediction. In *International Conference on Machine Learning*, pp. 20479–20502. PMLR, 2022.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Verkuil, R., Kabeli, O., Du, Y., Wicky, B. I., Milles, L. F., Dauparas, J., Baker, D., Ovchinnikov, S., Sercu, T., and Rives, A. Language models generalize beyond natural proteins. *bioRxiv*, pp. 2022–12, 2022.

Wang, S., Guo, Y., Wang, Y., Sun, H., and Huang, J. Smilesbert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, pp. 429–436, 2019.

Wang, Y., Magar, R., Liang, C., and Barati Farimani, A. Improving molecular contrastive learning via faulty negative mitigation and decomposed fragment contrast. *Journal of Chemical Information and Modeling*, 62(11):2713–2725, 2022.

Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B., et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pp. 2022–07, 2022.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9:513–530, 2018.

Xu, M., Guo, Y., Xu, Y., Tang, J., Chen, X., and Tian, Y. Eurnet: Efficient multi-range relational modeling of spatial multi-relational data. *arXiv preprint arXiv:2211.12941*, 2022.

Xue, D., Zhang, H., Xiao, D., Gong, Y., Chuai, G., Sun, Y., Tian, H., Wu, H., Li, Y., and Liu, Q. X-mol: large-scale pre-training for molecular understanding and diverse molecular analysis. *bioRxiv*, pp. 2020–12, 2020.

Yang, Y., Gao, J., Wang, J., Heffernan, R., Hanson, J., Paliwal, K., and Zhou, Y. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in bioinformatics*, 19(3):482–494, 2018.

Yang, Z., Hu, B., Han, A., Huang, S., and Ju, Q. Csp: code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2624–2636, 2020.

Yang, Z., Zhong, W., Zhao, L., and Chen, C. Y.-C. Mgraphdta: deep multiscale graph neural network for explainable drug–target binding affinity prediction. *Chemical science*, 13(3):816–833, 2022.

Yu, F., Koltun, V., and Funkhouser, T. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 472–480, 2017.

Yuan, W., Chen, G., and Chen, C. Y.-C. Fusiondta: attention-based feature polymerizer and knowledge distillation for drug-target binding affinity prediction. *Briefings in Bioinformatics*, 23(1):bbab506, 2022.

Zaidi, S., Schaarschmidt, M., Martens, J., Kim, H., Teh, Y. W., Sanchez-Gonzalez, A., Battaglia, P., Pascanu, R., and Godwin, J. Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*, 2022.

Zhang, X.-C., Wu, C.-K., Yang, Z.-J., Wu, Z.-X., Yi, J.-C., Hsieh, C.-Y., Hou, T.-J., and Cao, D.-S. Mg-bert: leveraging unsupervised atomic representation learning for molecular property prediction. *Briefings in bioinformatics*, 22(6):bbab152, 2021a.

Zhang, Z., Liu, Q., Wang, H., Lu, C., and Lee, C.-K. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 34:15870–15882, 2021b.

Zhang, Z., Xu, M., Jamasb, A., Chenthamarakshan, V., Lozano, A., Das, P., and Tang, J. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

Zhang, Z., Xu, M., Lozano, A., Chenthamarakshan, V., Das, P., and Tang, J. Physics-inspired protein encoder pretraining via siamese sequence-structure diffusion trajectory prediction. *arXiv preprint arXiv:2301.12068*, 2023.

Zheng, Z., Deng, Y., Xue, D., Zhou, Y., Ye, F., and Gu, Q. Structure-informed language models are protein designers. *bioRxiv*, pp. 2023–02, 2023.

Zhou, G., Gao, Z., Ding, Q., Zheng, H., Xu, H., Wei, Z., Zhang, L., and Ke, G. 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.

Zhu, J., Xia, Y., Wu, L., Xie, S., Qin, T., Zhou, W., Li, H., and Liu, T.-Y. Unified 2d and 3d pre-training of molecular representations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2626–2636, 2022.

A. Pre-training Configuration

Pre-training Datasets We use a combined dataset consisting of both protein and molecule data for pre-training. Since Euclidean distance is necessary for atom scale position encoding and pair-wise distance recovery, we utilize datasets that come with structural information, i.e., atom coordinates. For the protein dataset, we use AlphaFold DB (Varadi et al., 2022) dataset, which contains 8M protein sequences and structures predicted by AlphaFold2 (Jumper et al., 2021) with high confidence. For the molecule dataset, we use the dataset provided by Zhou et al. (2023), which contains 19M molecules and 209M conformations generated by ETKGD (Riniker & Landrum, 2015) and Merck Molecular Force Field (Halgren, 1996). Unlike Zhou et al. (2023), we do not train two models using two datasets respectively, instead we mix these two datasets and only train one *ms*-ESM.

ORDER Procedure For ORDER procedure, we use the default order in PDB (protein) and SDF (molecule) files as the order assigned to the atom set. To elaborate, PDB and SDF serve as standard formats for describing atomic structures of proteins and small molecules, respectively. In both formats, atoms follow specific sorting principles. In our study, we directly utilize the sorted atoms for ease of implementation. It is important to note that, given our atom scale position encoding employs Euclidean distance to describe positional relationships, the permutation of atom order does not impact our pre-training model.

Hyperparameters We implement *ms*-ESM using 12 stacked Transformer layers, each with 20 attention heads, as discussed in Section 2.4. The model dimension and feedforward dimension of each Transformer layer are 480 and 1920. The total number of *ms*-ESM's parameters is 35M. We use Adam (Kingma & Ba, 2014) and polynomial learning rate scheduler to train *ms*-ESM and set the learning rate 4e-4, weight decay 1e-2, warmup step 5000. The total training step is 300K and each batch has 256K tokens at maximum. We train *ms*-ESM on 16 NVIDIA A100 GPU cards for 3 days. *ms*-ESM is compatible with ESM series, so we load a ESM-2 35M checkpoint as the initialization of *ms*-ESM. When pre-training, 1.0% of residues are unzipped as the main experimental setting, which makes the unzipped protein sequence 1.08 times longer than before on average. Thus we make an adjustment to the maximum sequence length permissible for *ms*-ESM, transitioning from ESM-2's 1024 to 2048. For more pre-training hyperparameters, please refer to Table 5.

Table 5. ms-ESM hyperparameters for pre-training.

hyperparameters	Value
Learning rate	4e-4
LR scheduler	polynomial_decay
End learning rate	4e-5
Warmup updates	5000
Max update	300000
Max tokens	262144
Distance loss function and its weight	Smooth L1, 10.0
MLM loss function and its weight	Cross entropy, 4.0
Dropout	0.0
Attention dropout	0.0
Activation dropout	0.0
Num of encoder layers	12
Num of encoder attention heads	20
Encoder embedding dim	480
Encoder feedForward dim	1920
Adam (β_1, β_2)	(0.9,0.98)
Mask ratio	0.15
Unzip ratio	0.01
Distance noise ϵ	1 Å

B. Fine-tuning Details

Here, we offer additional implementation details for fine-tuning in downstream tasks. We also include the statistics of each fine-tuning dataset in Table 6.

Table 6. The statistics of downstream datasets in one table. ESAR: Enzyme-Substrate Affinity Regression, DTAR: Drug-Target Affinity Regression, ESPC: Enzyme-Substrate Pair Classification, SSP: Secondary Structure Prediction, CP: Contact Prediction, MPR: Molecular Property Regression, MPC: Molecular Property Classification.

	Protein-Molecule Task			Protein Task		Molecule Task									
Task	ESAR	DTAR	ESPC	SSP	CP	MPR			MPC						
Dataset	KM	Davis	ESP	NetSurfP-2.0, CB513 CASP12, TS115	ProteinNet	QM7	QM8	QM9	HIV	MUV	BACE	BBBP	TOX21	PCBA	SIDER
Train	8407	24045	49876	8678	20	5464	17428	107108	32901	74469	1210	1631	6,264	350343	1141
Valid	934	3006	5540	2170	24	685	2179	13388	4113	9309	151	204	783	43793	143
Test	2335	3005	13336	513/21/115	13945	681	2179	13389	4113	9309	151	204	783	43793	143
Total	11676	30056	68754	11497	13989	6830	21786	133885	41127	93087	1512	2039	7830	437929	1427

B.1. Fine-tuning Details of Protein-Molecule Tasks

Fine-tuning Datasets Following ProSmith (Kroll et al., 2023b), we fine-tune *ms*-ESM and all baseline models on dataset KM (Kroll et al., 2021), Davis (Davis et al., 2011), and ESP (Kroll et al., 2023a) for enzyme-substrate affinity regression, drug-target affinity regression, and enzyme-substrate pair classification respectively. The KM dataset contains experimental affinity constants of 11676 enzyme-substrate pairs. The Davis dataset provides 30056 binding affinities for pairs of 72 drugs and 442 proteins. The ESP dataset consists of 68754 positive or negative enzyme-substrate pairs with experimental evidence. We use the standard data split provided by ProSmith in fine-tuning.

Fine-tuning Framework As mentioned in Section 3.1, we use ProSmith's framework for a fair comparison. Specifically, the framework contains three main modules, i.e., molecule encoder, protein encoder, and fusion block. Two encoders extract features from proteins and molecules severally. The fusion block is a Transformer model, which is responsible for fusing protein and molecule features. The fused features are further used to regress the affinity values or predict binary affinity. We apply our model to ProSmith's framework by replacing both protein and molecule encoders with *ms*-ESM. We also provide the results of an XGBoost (Chen & Guestrin, 2016) variant of ProSmith, which removes the fusion block and uses simple concatenation for feature fusing. Note that we freeze both encoders in the experiments as suggested by ProSmith. We turn off the unzip operation when performing fine-tuning.

Fine-tuning Hyperparameters We directly use the hyperparameters provided by ProSmith. Specifically, the fusion block for three tasks has 6 layers of Transformer whose hidden size is 768. The epoch number is 100 and the learning rate is 1e-5. The batch sizes of the three tasks are 12, 12, and 24. We use Adam (Kingma & Ba, 2014) as the optimizer for ProSmith and GBDT (Ke et al., 2017) with 500 iterations as the predictors for XGBoost.

B.2. Fine-tuning Details of Protein Tasks

Fine-tuning Datasets Following TAPE's protocol (Rao et al., 2019), we evaluate *ms*-ESM on secondary structure prediction and contact prediction tasks. Specifically, for secondary structure prediction, we use data from Klausen et al. (2019) as training and validation sets and use CB513 (Cuff & Barton, 1999), CASP12 (Moult et al., 2018), and TS115 (Yang et al., 2018) as test sets. The training and validation sets are filtered at the 25% sequence identity threshold with these test tests. The final training, validation and three test sets have 8678, 2170, 513, 21, 115 protein sequences, respectively. For contact prediction tasks, we use training, validation, and test sets from ProteinNet (AlQuraishi, 2019) with training and validation sets filtered at the 30% sequence identity threshold. For a fair comparison, we also remove the test data that appears in the pre-training data, and the proportion of this part of the data is less than 4‰. The final training, validation, and test sets have 20, 24, 13945 protein sequences.

Fine-tuning Framework As suggested by TAPE, for both protein-only tasks, we use *ms*-ESM as the protein encoder. When doing secondary structure prediction, we use a linear output layer to predict the secondary structure which each residue belongs to. When handling the contact prediction task, we use the attention from the last layer as features and then use a linear layer to predict whether these two residues have contact or not. Notably, both input of these two tasks is only protein sequences without structural information. Therefore, when using *ms*-ESM to handle these two tasks, we turn off the unzip.

Table 7. Performance comparison on the contact prediction task.

Method	Short Range ↑			Medium Range ↑			Long Range ↑		
	P@L	P@L/2	P@L/5	P@L	P@L/2	P@L/5	P@L	P@L/2	P@L/5
TAPE 38M	**0.28**	**0.35**	0.46	0.19	0.25	0.33	0.17	0.20	0.25
ResNet 38M	0.25	0.34	0.46	0.18	0.25	0.35	0.10	0.13	0.17
ESM-2 35M	0.20	0.29	0.46	0.22	**0.32**	**0.45**	**0.30**	**0.39**	**0.49**
ms-ESM 35M	0.21	0.31	**0.48**	**0.23**	**0.32**	**0.45**	0.29	0.38	0.48

Fine-tuning Hyperparameters We set up all the hyperparameters aligned to TAPE. For secondary structure prediction, the epoch is 5000, batch size is 10, and learning rate is 0.001. For contact prediction, the epoch is 5, batch size 64, and learning rate is 3e-5. We use AdamW (Loshchilov & Hutter, 2017) as the optimizer in secondary structure prediction and Adam (Kingma & Ba, 2014) in contact prediction.

Baselines For protein tasks, we chose several popular protein pre-training models as our baselines. TAPE (Rao et al., 2019) and ResNet (Rao et al., 2019) employ a Transformer (Vaswani et al., 2017) and a dilated residual network (Yu et al., 2017), respectively, as the backbone network for training a masked language model (MLM). Because *ms*-ESM initializes its parameters by loading a checkpoint from ESM-2, we also include the ESM-2 model (Lin et al., 2023) in our comparison.

B.3. Fine-tuning Details of Molecule Tasks

Fine-tuning Datasets We use the fine-tuning data of Uni-Mol (Zhou et al., 2023) to evaluate the molecule understanding ability of *ms*-ESM. Specifically, we use QM7, QM8, and QM9 datasets for molecular property regression and HIV, MUV, BACE, BBBP, TOX21, PCBA, and SIDER datasets for molecular property classification, which have 6830, 21786, 133885, 41127, 93087, 1512, 2039, 7830, 437929, and 1427 molecules, respectively. The data split is also provided by Uni-Mol.

Fine-tuning Framework Following Uni-Mol, a special token, i.e., `[CLS]`, also exists in *ms*-ESM. Similar to NLP/CV, we simply use the representation of `[CLS]` to represent the whole molecule, and then use a linear head for fine-tuning on downstream tasks. For each molecule, we use the 3D conformation provided by Zhou et al. (2023) as the input of *ms*-ESM. In the fine-tuning stage, we do not add noises to atom coordinates.

Fine-tuning Hyperparameters For a fair comparison, we did not search the best hyperparameters. Instead, we set up all the hyperparameters aligned to Uni-Mol. Specifically, the batch sizes for these tasks are 32, 32, 128, 256, 128, 64, 128, 128, 128, and 32. The learning rates are 3e-4, 1e-4, 1e-4, 5e-5, 2e-5, 1e-4, 4e-4, 1e-4, 1e-4, and 5e-4. The training epochs are 100, 40, 40, 5, 40, 60, 40, 80, 20, and 80. We use Adam optimizer for all these tasks.

Baselines Following Uni-Mol, we use multiple supervised and pre-training methods as our baselines. The details of each baseline model can be found in the Uni-Mol paper (Zhou et al., 2023). For a fair comparison, we evaluate the performance of the official Uni-Mol checkpoint, which uses the same molecule training data as *ms*-ESM (remove all hydrogen atoms during training).

C. More Experiment Results

Table 7 and Table 8 display the completed experimental results for the contact prediction and secondary structure prediction tasks. Table 9 shows the experiment results of both molecular property classification and regression tasks.

D. More Related Work

Molecular Modeling Regarding the modality of molecules, studies on molecular modeling can be categorized into three groups. (i) 1D-based methods: These represent molecules with SMILES strings and employ language modeling techniques, such as masking and contrastive self-supervision, to enhance molecular representation (Wang et al., 2019; Honda et al., 2019; Chithrananda et al., 2020; Zhang et al., 2021a; Xue et al., 2020; Guo et al., 2022). (ii) 2D-based methods: These represent molecules with molecular graphs, sharing common ideas with general graph modeling. Some methods (Rong

Table 8. Performance comparison on secondary structure prediction task.

Method	SS3(ACC) ↑			SS8(ACC) ↑		
	cb513	ts115	casp12	cb513	ts115	casp12
TAPE 38M	0.73	0.77	0.71	0.59	0.64	0.59
ResNet 38M	0.75	0.78	0.72	0.58	0.64	0.58
ESM-2 35M	**0.80**	**0.82**	**0.74**	**0.65**	**0.70**	**0.61**
ms-ESM 35M	0.79	0.81	**0.74**	0.63	0.69	0.60

Table 9. Experimental results on molecular tasks. Compared with the vast majority of baseline models, *ms*-ESM performs well, which demonstrates that through the unified modeling approach we enable PLMs to perform well on pure molecule tasks as well.

Method	Reg. (MAE) ↓			Cls. (AUC,%) ↑						
	QM7	QM8	QM9	BACE	BBBP	TOX21	PCBA	SIDER	HIV	MUV
D-MPNN	103.5	0.0190	0.00814	80.9	71.0	75.9	86.2	57.0	77.1	78.6
Attentive FP	72.0	0.0179	0.00812	78.4	64.3	76.1	80.1	60.6	75.7	76.6
N-Gram$_{RF}$	92.8	0.0236	0.01037	77.9	69.7	74.3	-	66.8	77.2	76.9
N-Gram$_{XBG}$	81.9	0.0215	0.00964	79.1	69.1	75.8	-	65.5	78.7	74.8
GROVER$_{base}$	94.5	0.0218	0.00984	82.6	70.0	74.3	76.5	64.8	62.5	67.3
GROVER$_{large}$	92.0	0.0224	0.00986	81.0	69.5	73.5	83.0	65.4	68.2	67.3
PretrainGNN	113.2	0.0200	0.00922	84.5	68.7	78.1	86.0	62.7	79.9	81.3
GraphMVP	-	-	-	81.2	72.4	75.9	-	63.9	77.0	77.7
MolCLR	66.8	0.0178	-	82.4	72.2	75.0	-	58.9	78.1	79.6
Uni-Mol	58.9	0.0160	0.00540	83.2	71.5	78.9	88.1	57.7	78.3	72.0
ms-ESM	60.9	0.0171	0.00590	83.5	70.2	75.4	87.3	63.6	77.3	76.2

et al., 2020; Li et al., 2020; Zhang et al., 2021b; Li et al., 2021) mask key substructures of molecular graphs, like motifs and functional groups, and task models with reconstructing the masked parts. Others (Wang et al., 2022; Fang et al., 2022c; Lin et al., 2022a) align views from positive pairs (corrupt versions of the same graph) and simultaneously contrast views from negative pairs (different graphs). (iii) 3D-based methods: These directly utilize the 3D structure of molecules, aligning closely with our work. Earlier studies incorporated 3D information as an auxiliary input for 2D-based methods (Liu et al., 2021; Li et al., 2022b; Zhu et al., 2022; Stärk et al., 2022). More recent methods focus on molecular modeling with pure 3D inputs (Fang et al., 2022a; Zhou et al., 2023; Luo et al., 2022; Zaidi et al., 2022; Liu et al., 2022; Jiao et al., 2023). Three self-supervised techniques have been designed: geometry masking, geometry predicting, and denoising. For masking, Fang et al. (2022a) mask bond information, while Zhou et al. (2023) mask atom types, requiring models to predict masked information based on remaining context. For predicting, Fang et al. (2022a) proposes an atomic prediction task with bond information to capture global structure from local information. For denoising, models reconstruct 3D structures by adjusting corrupted structures. When corrupting structures, Zhou et al. (2023); Luo et al. (2022); Zaidi et al. (2022) add Gaussian noise to each atom of the input molecule. Several methods further introduce E(3)- and SE(3)-invariance inductive bias to the denoising technique (Zhou et al., 2023; Liu et al., 2022; Jiao et al., 2023).