Protein Representation Learning by Capturing Protein Sequence-Structure-Function Relationship

Eunji Ko[1*] **Seul Lee**[1*] **Minseon Kim**[1*] **Dongki Kim**[1] **Sung Ju Hwang**[1]
[1]KAIST, South Korea
{kosu7071,seul.lee,minseonkim,cleverki,sjhwang82}@kaist.ac.kr

Abstract

The goal of protein representation learning is to extract knowledge from protein databases that can be applied to various protein-related downstream tasks. Although protein sequence, structure, and function are the three key modalities for a comprehensive understanding of proteins, existing methods for protein representation learning have utilized only one or two of these modalities due to the difficulty of capturing the asymmetric interrelationships between them. To account for this asymmetry, we introduce our novel *asymmetric multi-modal masked autoencoder* (AMMA). AMMA adopts (1) a unified multi-modal encoder to integrate all three modalities into a unified representation space and (2) asymmetric decoders to ensure that sequence latent features reflect structural and functional information. The experiments demonstrate that the proposed AMMA is highly effective in learning protein representations that exhibit well-aligned inter-modal relationships, which in turn makes it effective for various downstream protein-related tasks.

1 Introduction

Proteins are generated in an organism in the form of a sequence, which is then folded into a three-dimensional structure, and as a three-dimensional structure, they become functional and fulfill their roles. This is the so-called protein sequence-structure-function paradigm (Liberles et al., 2012; Serçinoğlu & Ozbek, 2020). Of the three modalities—sequence, structure, and function—sequence information underlies many protein applications and is the most abundant, making it a popular choice for training neural networks. The challenge lies in developing sophisticated protein representations that utilize information across various modalities based on sequence data. However, existing methods for protein representation learning have only utilized some of the modalities, overlooking the importance of comprehensive integration of these modalities.

A significant hurdle to comprehensively considering the three modalities is the complexity of capturing the relationship between them. The correspondence between them is not straightforward, for example, even if the amino acid sequences are very similar, substrate specificity can change dramatically as the three-dimensional structure of the active site changes (Bunsupa et al., 2012). Moreover, proteins that have acquired the same function by convergent evolution, or that have accumulated sequence mutations where they do not affect protein folding, can have very little similarity in sequence. As has been described by many literatures (Illergård et al., 2009; Mahlich et al., 2018; van Kempen et al., 2022), it is a generally accepted fact that it is the structure, not the sequence, that is more conserved and directly related to the function of a protein. Figure 1 empirically supports this claim, showing that proteins with similar functional features are encoded into similar structural features, while their sequence features can differ largely. The first column of Table 1 quantitatively shows that the structure-function relationship is correlated more compared to the sequence-structure or sequence-function relationships. We refer to this relationship, where structure and function exhibit a strong alignment, while sequence and the other two show a relatively weaker correlation, as the *asymmetric relationship* between modalities.

To utilize information from multiple modalities, albeit not all three, most previous works (Zhang et al., 2022; Xu et al., 2023; Zhang et al., 2024) have leveraged contrastive learning, which learns

[*]Equal contribution

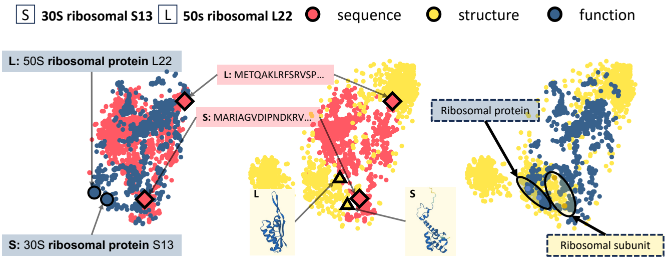


Figure 1: **t-SNE visualization of the three modalities of proteins.** Latent features for sequence (red), structure (yellow), and function (blue) extracted from ESM-1b, GearNet, and PubMedBERT-abs, respectively, are visualized. Two proteins, 30S ribosomal protein S13 (S) and 50S ribosomal protein L22 (L), are functionally similar and therefore proximal in function space (left). These proteins are encoded close together in structure space, but far apart in sequence space (middle). This trend is common across ribosomal proteins (right). Details are provided in Section B.4.

Table 1: **Cosine similarity scores** between the relation matrices which calculate relationship between the protein latents in a batch. The latents are extracted from the uni-modal encoders (i.e., ESM-1b, GearNet, and PubMedBERT-abs with additional projection layers). We report the similarity values calculated before and after applying contrastive learning (CL) and our proposed AMMA. Details are provided in Section B.5.

Alignment	Initial	CL	AMMA
Seq-Str	0.865	0.894	1.000
Seq-Func	0.855	0.907	0.999
Str-Func	0.947	0.847	0.999

the instance similarity and difference between two modalities. However, these approaches focus on improving representations of a single modality (e.g., protein sequence) by utilizing guidance from other "auxiliary" modalities. This leads to a skewed integration of modalities, as shown in the second column of Table 1, where contrastive learning shows high sequence-structure and sequence-function similarity but low structure-function similarity.

To this end, we propose *Asymmetric Multi-modal Masked Autoencoder* (AMMA), an integrated protein representation learning method that jointly embeds the three core modalities of proteins. Under the masked autoencoder framework (Bachmann et al., 2022), AMMA captures the asymmetric sequence-structure-function relationship inherent in the protein domain through (1) the unified multi-modal encoder and (2) the asymmetric design of the decoder. By having the multi-modal encoder, AMMA explicitly integrates information from all of the modalities rather than letting one modality guide the others. Furthermore, by learning to predict structure and function from sequence with the asymmetric decoders, AMMA ensures that sequence latent features faithfully reflect structure and function information. As shown in the last column of Table 1, AMMA successfully yields well-aligned multi-modal protein representations. We experimentally validate the proposed AMMA on various tasks that require accurate protein representation learning. The experimental results demonstrate that AMMA outperforms existing state-of-the-art methods, showing its superiority in learning protein representations by comprehensively and effectively considering the multi-modal aspects of proteins. We summarize our contributions as follows:

- We are the first to propose utilizing the three core modalities for protein representation learning: sequence, structure, and function.
- We point out the asymmetric relationship between sequence, structure, and function of proteins and propose AMMA, a masked autoencoder framework that adopts a unified multi-modal encoder and asymmetric decoders to account for the asymmetric relationship.
- We experimentally demonstrate that AMMA is highly effective in learning protein representations and benefits performance on a variety of downstream protein-related tasks.

2 Related works

As a means to overcome the limitations of uni-modal protein representation learning, protein representation learning using multiple modalities has gained traction. Most previous studies have adopted a contrastive learning approach to capture the relationship between modalities. Zhang et al. (2022) employed knowledge-aware negative sampling to identify negative instances, enabling contrastive learning across proteins. Xu et al. (2023) conducted contrastive learning between protein sequence and functional description. However, contrastive learning may not be an optimal for multi-modal representation learning, as it focuses on learning improved representations of a single modality using other modal information and thus cannot yield balanced multi-modal representations.

Apart from contrastive learning, there are other approaches to multi-modal protein representation learning. Su et al. (2023) proposed using structure-aware tokens from FoldSeek (van Kempen et al., 2022) to train a token-based ESM (Lin et al., 2022). However, the method of Su et al. (2023) uses 20 different structural tokens, which restricts diversity when encoding structural information.

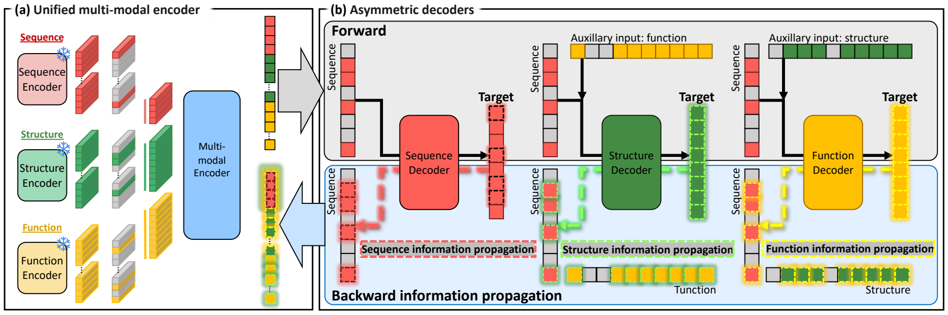

(a) Unified multi-modal encoder
Sequence
Sequence Encoder
Structure
Structure Encoder
Function
Function Encoder
Multi-modal Encoder
(b) Asymmetric decoders
Forward
Sequence
Auxillary input: function
Auxillary input: structure
Target
Sequence Decoder
Structure Decoder
Function Decoder
Sequence information propagation
Structure information propagation
Function information propagation
Tunction
Structure
Backward information propagation

Figure 2: **Multi-modal protein representation learning with AMMA.** AMMA has two key components: (a) a unified multi-modal encoder and (b) asymmetric decoders. Each modality is encoded by a frozen pretrained encoder, then integrated by a multi-modal encoder after masking. Asymmetric decoders then reconstruct original features of each modality. During decoding, the input latent features, designed to hold target-specific information, are asymmetrically passed to the decoders for target modality reconstruction. This requires AMMA to encode structural and functional information into sequence latent features, which allows AMMA to capture unique asymmetric sequence-structure-function relationships. The overall architecture is provided in Figure 5.

Moreover, these methods do not consider another important modality for proteins, the function description, and are therefore suboptimal.

3 Method

3.1 Asymmetric Multi-modal Masked Autoencoder (AMMA)

Sequence, Structure, and Function Uni-modal Encoders To integrate the protein information of sequence, structure, and function equally to construct a unified protein representation, we first propose to utilize a single multi-modal encoder. The inputs to the multi-modal encoder are the features extracted from each of the uni-modal encoders. The input data can be the sequence $X_{\texttt{seq}}$, structure $X_{\texttt{str}}$, or function $X_{\texttt{func}}$, or a combination of these modalities. We adopt pretrained feature extractors as the uni-modal encoders $\textsc{Enc}_{\texttt{seq}}$, $\textsc{Enc}_{\texttt{str}}$, and $\textsc{Enc}_{\texttt{func}}$. Specifically, we use ESM-1b (Rives et al., 2021), GearNet (Zhang et al., 2023), and PubMedBERT-abs (Gu et al., 2021) for extracting features from the sequence, structure, and function inputs, respectively. Note that our approach is model-agnostic, and any off-the-shelf uni-modal feature extractors can be used. $X_{\texttt{seq}}$, $X_{\texttt{str}}$, and $X_{\texttt{func}}$ are each passed to the corresponding uni-modal encoder and becomes $X'_{\texttt{seq}} \in \mathbb{R}^{L\times 1280}$, $X'_{\texttt{str}} \in \mathbb{R}^{L\times 3072}$, and $X'_{\texttt{func}} \in \mathbb{R}^{L'\times 768}$. L is the number of amino acids of the protein and L' is the number of the tokens of the function description. Then, we add two fully connected layers, $\textsc{Proj}_{\texttt{seq}}$, $\textsc{Proj}_{\texttt{str}}$, and $\textsc{Proj}_{\texttt{func}}$, to each of the uni-modal encoders to project the latent features of multiple modalities to the same size of projection dimension. The result becomes $Z_{\texttt{seq}} \in \mathbb{R}^{L\times D}$, $Z_{\texttt{str}} \in \mathbb{R}^{L\times D}$, and $Z_{\texttt{func}} \in \mathbb{R}^{L'\times D}$. D is the projection dimension. We set D to 512.

Mask Sampling Under the masked autoencoder framework, we mask the encoded latent features Z. Specifically, we first sample the preserving ratios between the modalities, $\lambda_{\texttt{seq}}$, $\lambda_{\texttt{str}}$, and $\lambda_{\texttt{func}}$, based on the Dirichlet distribution following Bachmann et al. (2022), where $\lambda_{\texttt{seq}}+\lambda_{\texttt{str}}+\lambda_{\texttt{func}} = 1$, $\lambda_{\texttt{seq}} \geq 0$, $\lambda_{\texttt{str}} \geq 0$, and $\lambda_{\texttt{func}} \geq 0$ as follows:

$$(\lambda_{\texttt{seq}}, \lambda_{\texttt{str}}, \lambda_{\texttt{func}}) \sim \text{Dirichlet}(\alpha_{\texttt{seq}}, \alpha_{\texttt{str}}, \alpha_{\texttt{func}}). \tag{1}$$

We set the value of $\alpha_{\texttt{seq}}$, $\alpha_{\texttt{str}}$, and $\alpha_{\texttt{func}}$ to 1, 2, and 2, respectively. We then randomly mask the latent features $Z_{\texttt{seq}}$, $Z_{\texttt{str}}$, and $Z_{\texttt{func}}$ such that the number of preserved tokens has the ratio $\lambda_{\texttt{seq}} : \lambda_{\texttt{str}} : \lambda_{\texttt{func}}$ and sums to the total number of tokens M. Subsequently, we concatenate the masked features from all three modalities to be $Z \in \mathbb{R}^{M\times D}$, which will be the input to the multi-modal encoder described in the next paragraph. We set M to 160.

Multi-modal Encoder To learn protein representations that faithfully contain information from multiple modalities uniformly in a well-aligned manner, we propose to use a unified protein multi-modal encoder $\textsc{Enc}_{\texttt{multi}}$ that integrates different modalities into a single representation space. $\textsc{Enc}_{\texttt{multi}}$ encodes the concatenated and masked multi-modal latent features as follows:

$$Z_{\texttt{multi}} = \textsc{Enc}_{\texttt{multi}}(Z) \in \mathbb{R}^{M\times D}. \tag{2}$$

We adopt an 8-layer Transformer as $\textsc{Enc}_{\texttt{multi}}$. Through the self-attention mechanism, $\textsc{Enc}_{\texttt{multi}}$ facilitates the fusion of multiple modality information.

Asymmetric Decoder To train the multi-modal encoder under the autoencoder framework, we adopt individual modality decoders that reconstruct the original latent variables $X'_{\texttt{seq}}$, $X'_{\texttt{str}}$, and $X'_{\texttt{func}}$, respectively. Unlike multi-modal representation learning in the image domain, multi-modal protein representation learning should consider that the three protein modalities exhibit a unique asymmetric relationship in which sequence-structure and sequence-function are relatively poorly aligned compared to structure-function (see Figure 1). To effectively represent proteins by incorporating information across modalities and capturing their asymmetric interrelationships, we introduce asymmetric decoders.

The multi-modal latent variable $Z_{\texttt{multi}}$ computed by the multi-modal encoder $\textsc{Enc}_{\texttt{multi}}$ is first passed to each of the three single linear layers ℓ_0, ℓ_1, and ℓ_2 and becomes $Z'_{\texttt{multi},0} \in \mathbb{R}^{M\times D}$, $Z'_{\texttt{multi},1} \in \mathbb{R}^{M\times D}$, and $Z'_{\texttt{multi},2} \in \mathbb{R}^{M\times D}$. Tokens of zero are then inserted into the masked positions of $Z'_{\texttt{multi},0}$, $Z'_{\texttt{multi},1}$, and $Z'_{\texttt{multi},2}$ to regain the original unmasked length $2L + L'$. The three latent features of size $(2L+L')\times D$ each split into three modal-specific latent features $Z'_{\texttt{seq},k} \in \mathbb{R}^{L\times D}$, $Z'_{\texttt{str},k} \in \mathbb{R}^{L\times D}$, and $Z'_{\texttt{func},k} \in \mathbb{R}^{L'\times D}$ for $k = 0, 1, 2$. Subsequently, the latent features are asymmetrically passed to modal-specific decoders $\textsc{Dec}_{\texttt{seq}}$, $\textsc{Dec}_{\texttt{str}}$, and $\textsc{Dec}_{\texttt{func}}$ depending on the modality as follows:

$$\begin{aligned}
\hat{X}_{\texttt{seq}} &= \textsc{Dec}_{\texttt{seq}}(Z'_{\texttt{seq},0}) \in \mathbb{R}^{L\times 1280}, \\
\hat{X}_{\texttt{str}} &= \textsc{Dec}_{\texttt{str}}(Z'_{\texttt{seq},1},\, Z'_{\texttt{func},1}) \in \mathbb{R}^{L\times 3072}, \\
\hat{X}_{\texttt{func}} &= \textsc{Dec}_{\texttt{func}}(Z'_{\texttt{seq},2},\, Z'_{\texttt{str},2}) \in \mathbb{R}^{L'\times 768},
\end{aligned} \tag{3}$$

where $\hat{X}_{\texttt{seq}}$, $\hat{X}_{\texttt{str}}$, and $\hat{X}_{\texttt{func}}$ are the reconstructed latent features of $X'_{\texttt{seq}}$, $X'_{\texttt{str}}$, and $X'_{\texttt{func}}$, respectively. Each decoder consists of two Transformer layers followed by a single linear layer. During the pretraining, we keep the uni-modal encoders frozen and train AMMA using the mean squared error (MSE) loss as follows:

$$\begin{gathered}
\mathcal{L}_{\texttt{seq}} = \text{MSE}(\hat{X}_{\texttt{seq}},\, X'_{\texttt{seq}}), \mathcal{L}_{\texttt{str}} = \text{MSE}(\hat{X}_{\texttt{str}},\, X'_{\texttt{str}}), \mathcal{L}_{\texttt{func}} = \text{MSE}(\hat{X}_{\texttt{func}},\, X'_{\texttt{func}}), \\
\mathcal{L} = \mathcal{L}_{\texttt{seq}} + \mathcal{L}_{\texttt{str}} + \mathcal{L}_{\texttt{func}}.
\end{gathered} \tag{4}$$

The key to the proposed AMMA is the specialized asymmetric design of each decoder. The sequence decoder takes $Z'_{\texttt{seq}}$ as input to predict the original sequence latent feature. This ensures that the sequence information remains intact during the fusion process. On the other hand, the structure decoder takes both $Z'_{\texttt{seq}}$ and $Z'_{\texttt{func}}$ as input to reconstruct the original structural latent feature. This design is crucial to ensure the sequence latent features to reflect structural information during the fusion process of the multi-modal encoder. Similarly, the function decoder takes both $Z'_{\texttt{seq}}$ and $Z'_{\texttt{str}}$ as input to predict the original function latent feature, ensuring the sequence latent features to reflect function information during the fusion process. Because structure and function are difficult to predict from sequence features alone and structure and function are relatively well aligned, structure and function can provide the auxiliary information needed to reconstruct each other. Using other modalities as auxiliary information for the structure and function decoders is more effective than using their own features as auxiliary information. This strategy prevents the decoders from overly relying on their own explicit inputs, which can weaken the modality information propagating backwards to the sequence features, preventing the integration of the modalities.

Through the proposed multi-modal encoder and asymmetric decoders, AMMA adeptly learns to encapsulate all three core protein modalities: sequence, structure, and function. This results in effective and comprehensive multi-modal protein representations that reflects the complex interdependencies of protein modalities.

4 Experiments

4.1 Protein Function Prediction

As shown in Table 2, AMMA outperforms all baselines on all tasks in terms of AUPR and on two out of four tasks in terms of F_{max}. This demonstrates that the proposed pretraining scheme with AMMA is highly effective in learning high-quality protein representations that can be universally used in downstream tasks to improve performance. Specifically, AMMA largely outperforms its sequence encoder ESM-1b and its structure encoder GearNet in terms of average score, showing that utilizing protein representations that integrate information from multiple modalities benefits

Table 2: **Performance on protein function annotation tasks.**

Method	Modality			EC		GO-MF		GO-CC		GO-BP		Avg.$_{\text{Fmax}}$	Avg.$_{\text{AUPR}}$
	Seq.	Str.	Func.	F_{max}	AUPR	F_{max}	AUPR	F_{max}	AUPR	F_{max}	AUPR		
ESM-1b (Rives et al., 2021)	✓			86.9	88.4	65.9	63.0	47.7	32.4	45.2	33.2	61.4	54.3
OntoProtein (Zhang et al., 2022)	✓		✓	84.1	85.4	63.1	60.3	44.1	30.0	43.6	28.4	58.7	51.0
GearNet (Zhang et al., 2023)		✓		87.4	89.2	65.4	59.6	48.8	33.6	**49.0**	29.2	62.7	52.9
SaProt (Su et al., 2023)	✓	✓		**88.8**	85.5	**68.8**	58.2	41.2	20.6	45.1	23.8	61.0	47.0
ProtST (Xu et al., 2023)	✓		✓	87.8	89.4	66.1	64.4	48.8	36.4	48.0	32.8	62.7	55.8
AMMA-symmetric (ours)	✓	✓	✓	71.6	74.9	52.0	52.3	48.8	35.1	35.5	24.3	52.0	46.7
AMMA-contrastive (ours)	✓	✓	✓	87.7	89.5	65.2	61.3	44.3	28.2	28.2	17.3	56.4	49.1
AMMA (ours)	✓	✓	✓	88.7	**89.8**	67.3	**65.5**	**49.8**	**36.9**	46.9	**33.6**	**63.2**	**56.5**

prediction performance. On the other hand, AMMA shows better or comparable results to ProtST, a model that use significantly more parameters and training resources than AMMA. While ProtST has 650M parameters and takes 205 hours to pretrain, AMMA has 111M parameters and takes only 120 hours to train on fewer GPUs as shown in Section B.3. This result demonstrates that AMMA learns comprehensive protein representations in an efficient and effective way to improve the performance of a wide range of downstream tasks.

4.2 Improving Performance with Unpaired Data

One of the biggest limitations of contrastive learning approach is its inability to leverage abundant unpaired data, i.e., data without all modality information. On the contrary, the pretraining strategy using AMMA can be flexibly applied to unpaired data under the masked autoencoder framework. Since there is much more data that only partially has the sequence-structure-function triplet, this is a huge advantage and can be leveraged to further improve the performance of AMMA.We construct an unpaired dataset of 50k with 25k sequence-structure pairs randomly selected from AlphaFoldDB and 25k sequence-function pairs randomly selected from ProtDescribe and further train AMMA on them. As shown in Table 3, we found that further pretraining AMMA on the unpaired data improves prediction performance, demonstrating the scalability of AMMA. The detailed experimental setting is provided in Section B.6.

Table 3: **EC/GO results of 15 epochs with extra unpaired data.**

Data		EC		GO-MF		Average
Paired	Unpaired	F_{max}	AUPR	F_{max}	AUPR	
120k	0k	88.1	89.7	66.4	**64.6**	77.2
120k	50k	**88.2**	**90.4**	**66.9**	**64.6**	**77.5**

4.3 Ablation Studies and Qualitative Analysis

Effect of the Asymmetric Decoders To capture the asymmetric relationship between protein sequence, structure, and function, AMMA employs asymmetric decoding, where the structure decoder predicts structural features by considering sequence and function, and the function decoder predicts functional features by considering sequence and structure. To examine the effect of the asymmetric decoding, we compare AMMA to **AMMA-symmetric**, a variant of AMMA that uses symmetric decoding. The sequence decoder, structure decoder, and function decoder of AMMA-symmetric take sequence, structure, and function inputs, respectively, and predict its own features, i.e., sequence, structure, and function features, respectively, without reference to other modalities.

As shown in Table 2, AMMA outperforms AMMA-symmetric by a large margin, demonstrating that the asymmetric design of decoders greatly benefits performance as it allows the multi-modal encoder to encode a more comprehensive protein representation by considering the asymmetric relationship between the protein modalities.

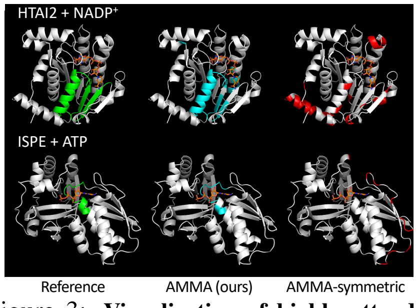
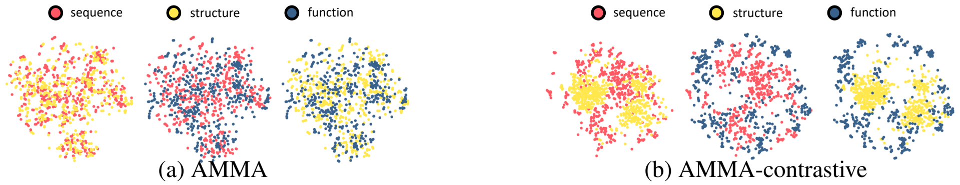


Figure 3: **Visualization of highly attended residues in a functional context.**

To further investigate the effect of the asymmetric decoders and understand the inter-modal relationship at the residue level, we visualize the residues with the highest function-to-residue attention values in AMMA and AMMA-symmetric. As shown in Figure 3, residues heavily attended by functional tokens correspond to actual protein-ligand binding regions in AMMA, while heavily attended residues in AMMA-symmetric do not correlate with the actual interaction region. This demonstrates that the asymmetric design of decoders helps AMMA understand the relationship between functional context and residual information. Furthermore, this also suggests that AMMA could be utilized to find the active region of a protein, another important research problem with various applications. We provide the experimental details in Section B.8.



(a) AMMA (b) AMMA-contrastive

Figure 4: t-SNE visualization of the three protein modalities after (a) AMMA training and (b) contrastive learning. Sequence, structure, and function features are well-aligned after training with AMMA while contrastive learning fails to align the three modalities in a balanced manner. Details are provided in Section B.4.

Comparison with Contrastive Learning We also compare AMMA to contrastive learning, a popular pretraining strategy for learning relationships between multiple modalities. Specifically, we construct **AMMA-contrastive**, a model that uses a similar architecture to AMMA but uses contrastive learning to obtain multi-modal protein representations.

As shown in Table 2, AMMA largely outperforms AMMA-contrastive, indicating that the proposed strategy to utilize a unified multi-modal encoder and asymmetric decoders is much more effective than contrastive learning in integrating the multi-modal information of proteins. Compared to its uni-modal sequence encoder ESM-1b, AMMA-contrastive shows better EC prediction performance but worse GO prediction performance, suggesting that contrastive learning is not beneficial to all tasks, while AMMA learns high-quality protein representations that are universally applicable to a variety of downstream tasks.

In addition, we provide t-SNE visualization of the uni-modal latent features obtained after training with AMMA and contrastive learning in Figure 4a and 4b, respectively. While the features of the difference modalities after training with AMMA are well aligned, those after contrastive learning are not aligned in a balanced way, because AMMA-contrastive only utilizes structure and function features to guide sequence features and does not uniformly fuse the multi-modal information. This results can also quantitatively reconfirmed in Table 1, indicating that AMMA uniformly integrates information of multiple modalities while the contrastive learning approach cannot.

Effect of the Masking Ratio We examine the effect of the α value used for Dirichlet sampling to sample the masking ratio. As shown in Table 4, the choice of α value has a large impact on the final performance of AMMA. Note that the higher the α value, the less masking is applied and the more tokens are preserved. We can observe a trend that preserving fewer sequence features favors model performance. This is due to the two reasons. First, as we saw in the previous paragraph, auxiliary structure or function features are important for successful pretraining of AMMA, so preserving them over less aligned sequence features is beneficial to performance. This can be seen by the fact that the third row performs the worst due to minimal auxiliary features, while the fourth and fifth rows perform better by utilizing more auxiliary information. Second, masking more tokens in a sequence feature make the sequence feature more robust because it must contain a wealth of information in its small subset. As a result, the best performance is achieved by setting $\alpha_{\texttt{seq}}$, $\alpha_{\texttt{str}}$, and $\alpha_{\texttt{func}}$ to 1, 2, and 2, respectively, which are the values used in the main experiment (Table 2).

Table 4: **Experimental results with different α for Dirichlet sampling.** The pretraining is conducted using a 22k dataset, a random subset of the 120k dataset.

Ratio			EC		GO-MF		Average
$\alpha_{\texttt{seq}}$	$\alpha_{\texttt{str}}$	$\alpha_{\texttt{func}}$	F_{max}	AUPR	F_{max}	AUPR	
1	1	1	84.6	87.2	66.4	64.8	75.8
1	2	2	87.7	89.8	66.4	64.2	**77.0**
2	1	1	73.0	75.9	65.1	63.2	69.3
2	1	2	86.7	89.2	65.9	64.7	76.6
2	2	1	87.9	89.5	52.4	53.6	70.9

5 Conclusion

In this paper, we proposed AMMA to address the problem of multi-modal protein representation learning that considers three core protein modalities: sequence, structure, and function. AMMA utilizes a unified multi-modal encoder and asymmetric decoders to capture the asymmetric relationship between the protein modalities, resulting in high-quality, comprehensive multi-modal protein representations. Through various experiments, AMMA demonstrated its effectiveness on protein-related downstream tasks with much less pretraining data. We believe that AMMA can be applied to improve our understanding of proteins by providing a comprehensive view of their properties, and we expect AMMA to spawn many interesting future studies.

REFERENCES

Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision*, pp. 348–367. Springer, 2022.

Somnuk Bunsupa, Kae Katayama, Emi Ikeura, Akira Oikawa, Kiminori Toyooka, Kazuki Saito, and Mami Yamazaki. Lysine decarboxylase catalyzes the first step of quinolizidine alkaloid biosynthesis and coevolved with alkaloid production in leguminosae. *The Plant Cell*, 24(3):1202–1216, 2012.

Can Chen, Jingbo Zhou, Fan Wang, Xue Liu, and Dejing Dou. Structure-aware protein self-supervised learning. *Bioinformatics*, 39(4):btad189, 2023.

UniProt Consortium. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47 (D1):D506–D515, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

Jerome Eberhardt, Diogo Santos-Martins, Andreas F Tillack, and Stefano Forli. Autodock vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 61(8):3891–3898, 2021.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1): 1–23, 2021.

Michael Heinzinger, Maria Littmann, Ian Sillitoe, Nicola Bordin, Christine Orengo, and Burkhard Rost. Contrastive learning on protein embeddings enlightens midnight zone. *NAR genomics and bioinformatics*, 4(2):lqac043, 2022.

Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. *arXiv preprint arXiv:2205.15675*, 2022.

Kristoffer Illergård, David H Ardell, and Arne Elofsson. Structure is three to ten times more conserved than sequence—a study of structural response in protein cores. *Proteins: Structure, Function, and Bioinformatics*, 77(3):499–508, 2009.

David A Liberles, Sarah A Teichmann, Ivet Bahar, Ugo Bastolla, Jesse Bloom, Erich Bornberg-Bauer, Lucy J Colwell, AP Jason De Koning, Nikolay V Dokholyan, Julian Echave, et al. The interface of protein structure, protein biophysics, and molecular evolution. *Protein Science*, 21 (6):769–785, 2012.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Yannick Mahlich, Martin Steinegger, Burkhard Rost, and Yana Bromberg. Hfsp: high speed homology-driven function annotation of proteins. *Bioinformatics*, 34(13):i304–i312, 2018.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, 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.

Onur Serçinoğlu and Pemra Ozbek. Sequence-structure-function relationships in class i mhc: A local frustration perspective. *PloS one*, 15(5):e0232849, 2020.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pp. 2023–10, 2023.

Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pp. 2022–02, 2022.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, 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.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. Protst: Multi-modality learning of protein sequences and biomedical texts. *International Conference on Machine Learning*, 2023.

Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Jiazhang Lian, Qiang Zhang, and Huajun Chen. Ontoprotein: Protein pretraining with gene ontology embedding. *International Conference on Learning Representations*, 2022.

Ruochi Zhang, Haoran Wu, Chang Liu, Huaping Li, Yuqian Wu, Kewei Li, Yifan Wang, Yifan Deng, Jiahui Chen, Fengfeng Zhou, et al. Pepharmony: A multi-view contrastive learning framework for integrated sequence and structure-based peptide encoding. *arXiv preprint arXiv:2401.11360*, 2024.

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *International Conference on Learning Representations*, 2023.

Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

Appendix

A Related Works

A.1 Uni-modal Protein Representation Learning

For learning protein representations using a single modality, the most basic and widely used is the sequence of the protein, i.e., a linear chain of amino acid residues. Many previous sequence-based protein representation learning methods have utilized language modeling techniques such as masked language modeling (MLM) (Devlin et al., 2019). ProtBERT (Elnaggar et al., 2021) used BERT (Devlin et al., 2019) to reconstruct missing amino acid residues. ESM-1b (Rives et al., 2021) conducted masked language modeling (MLM) unsupervised learning on 250M protein sequences. ESM-1v (Meier et al., 2021) focused on capturing the effect of variations in sequence on function. ESM-2 (Lin et al., 2022) proposed a language model with the larger 15B parameters than ESM-1b. Heinzinger et al. (2022) proposed to obtain optimized sequence embeddings for the CATH protein structure hierarchy (Orengo et al., 1997) by using contrastive learning to maximize the distance between sequences from different CATH classes and minimize the distance for those within the same class. Recently, the importance of structural information in learning protein representations has came to the fore and recent works have been proposed to exploit structural features in protein 3D geometry. Hermosilla & Ropinski (2022) and Zhang et al. (2023) proposed to learn geometric features through contrastive learning between substructures of a given protein. While these sequence- or structure-based methods can capture properties of proteins thanks to the vast amount of protein data available, they are limited to learning a single modality of proteins and cannot obtain comprehensive protein representations.

A.2 Multi-modal Protein Representation Learning

Contrastive Learning-based Multi-modal Learning As a means to overcome the limitations of uni-modal protein representation learning, protein representation learning using multiple modalities has gained traction. Most previous studies have adopted a contrastive learning approach to capture the relationship between modalities. Zhang et al. (2022) employed knowledge-aware negative sampling to identify negative instances, enabling contrastive learning across proteins. Xu et al. (2023) conducted contrastive learning between protein sequence and functional description. Zhang et al. (2024) utilized ESM (Rives et al., 2021) and GearNet (Zhang et al., 2023) to encode sequence and structure data, respectively, and performed contrastive learning between the encoded sequence and structure features. However, contrastive learning may not be an optimal choice for multi-modal protein representation learning, as it essentially focuses on learning improved representations of a single modality using other modal information and thus cannot yield balanced multi-modal representations.

Other Multi-modal Learning Approaches Apart from contrastive learning, there are other approaches to multi-modal protein representation learning. Chen et al. (2023) incorporated protein structural information by learning to predict residue distance or dihedral angle. This work also proposed to maximize the mutual information between the sequential representation and structural representation. Su et al. (2023) proposed using structure-aware tokens extracted from FoldSeek (van Kempen et al., 2022) to train a token-based ESM (Lin et al., 2022) backbone. These methods showed superior results as they consider the structure of proteins. However, the method of Chen et al. (2023) has the limitation that it only uses sequence information as an auxiliary to guide and enhance structural representation. The method of Su et al. (2023) uses 20 different structural tokens, which restricts diversity when encoding structural information. Moreover, these methods do not consider another important modality for proteins, the function description, and are therefore suboptimal.

B Experimental Details

B.1 Details on Pretraining

Details For pretraining, we set the batch size as 4 and the number of training epochs as 10. We used the AdamW (Loshchilov & Hutter, 2019) optimizer with a learning rate of 1×10^{-4} and a weight decay of 0.05. The pretraining scheduler was StepLR with gamma 0.5.

Dataset For pretraining, we built a dataset of 120k sequence, structure, and function triplets. We extracted common proteins from the AlphaFold v2 dataset (Varadi et al., 2022) of 440k sequence and structure pairs and the ProtDescribe (Xu et al., 2023) dataset of 553k sequence and function description pairs. We used the preprocessed sequences of AlphaFold using the torchdrug (Zhu et al., 2022) library when extracting the common proteins.

B.2 Details on Protein Function Prediction

Downstream Tasks We evaluated the performance of our model on two standard downstream tasks following Gligorijević et al. (2021): Enzyme Commission (EC) number prediction and Gene Ontology (GO) term prediction. The GO benchmark has three sub-tasks, i.e., the tasks to predict Molecular Function (GO-MF), Cellular Component (GO-CC), and Biological Process (GO-BP), respectively.

Supervised Finetuning To finetune AMMA, sequence and structural features extracted from unimodal encoders, respectively, were concatenated and passed to the multi-modal encoder without masking. Following Devlin et al. (2019), we use the latent feature corresponding to first token as the multi-modal representation. We add two fully connected (FC) layers after the multi-modal encoder and use cross entropy loss to finetune the sequence encoder, multi-modal encoder, and two FC layers together.

We set the batch size to 1 on EC, GO-MF, and GO-CC and 4 on GO-BP. We trained for 100 epochs on EC, and 50 epochs on GO. For EC and GO-BP, we used the AdamW (Loshchilov & Hutter, 2019) optimizer with a learning rate of 1.0×10^{-5} for $\textsc{Enc}_{\texttt{multi}}$, $\textsc{Enc}_{\texttt{seq}}$, and $\textsc{Proj}_{\texttt{str}}$, and 1.0×10^{-4} for the Multi-Layer Perceptron (MLP) classifier. For GO-MF and GO-CC, we used the Adam optimizer with a learning rate of 1.0×10^{-5} for $\textsc{Enc}_{\texttt{multi}}$, $\textsc{Enc}_{\texttt{seq}}$, and $\textsc{Proj}_{\texttt{str}}$, and 1.0×10^{-4} for the Multi-Layer Perceptron (MLP) classifier. We used the ExponentialLR scheduler with a gamma value of 0.95.

Baselines We compared AMMA with five protein representation learning baselines. **ESM-1b** (Rives et al., 2021) used masked language modeling (MLM) to learn protein representations from a large sequence database. **OntoProtein** (Zhang et al., 2022) incorporated knowledge graphs (KGs) to enhance protein sequence embeddings with biological knowledge facts. **GearNet** (Zhang et al., 2023) leveraged multiview contrastive learning to train the structural encoder. **SaProt** (Su et al., 2023) used structure-aware tokens to incorporate structure information to ESM (Lin et al., 2022). **ProtST** (Xu et al., 2023) used contrastive learning to align sequence information from a protein language model with functional information from a biomedical language model.

Finetuning Dataset We tested on the Enzyme Commission (EC) and Gene Ontology (GO) in downstream tasks. We used a 95% sequence identity cutoff for both EC and GO, following GearNet (Zhang et al., 2023). The dataset size of each dataset is shown in Table 5.

Table 5: **The size of the finetuning datasets.**

	# Train	# Valid	# Test
Enzyme Commission (EC)	15,035	1,665	1,840
Gene Ontology (GO)	27,581	3,061	2,991

Evaluation Metrics To quantify the effectiveness of protein representation learning methods, we used two commonly used metrics: the protein-centric maximum F-score ($\mathrm{F}_{\mathrm{max}}$) and pair-centric area under precision-recall curve (AUPR) that are implemented in torchdrug (Zhu et al., 2022).

The $\mathrm{F}_{\mathrm{max}}$ is the protein-centric maximum F-score. $t \in [0, 1]$ denotes a decision threshold for a target protein i, and we calculated precision and recall as follows:

$$\text{precision}_i(t) = \frac{\sum_f \mathbb{1}[f \in P_i(t) \cap T_i]}{\sum_f \mathbb{1}[f \in P_i(t)]}, \tag{5}$$

$$\text{recall}_i(t) = \frac{\sum_f \mathbb{1}[f \in P_i(t) \cap T_i]}{\sum_f \mathbb{1}[f \in T_i]}, \tag{6}$$

where f indicates a functional term in EC or GO ontology. For protein i, T_i denotes the set comprising all experimentally validated functional terms for the protein. $P_i(t)$ is the set of predicted functional terms for protein i, each with a score at least threshold t. $\mathbb{1}[\cdot]$ denotes an indicator function.

The average precision and recall with threshold t for all proteins are defined as follows:

$$\text{precision}(t) = \frac{1}{M(t)} \sum_i \text{precision}_i(t), \tag{7}$$

$$\text{recall}(t) = \frac{1}{N} \sum_i \text{recall}_i(t), \tag{8}$$

where N is the number of proteins, and $M(t)$ represents the count of proteins that have at least one predicted function exceeding the threshold t.

F_{max} is calculated as follows:

$$\text{F}_{\text{max}} = \max_t \left\{ \frac{2 \cdot \text{precision}(t) \cdot \text{recall}(t)}{\text{precision}(t) + \text{recall}(t)} \right\}. \tag{9}$$

The second metric, AUPR is pair-centric area under precision-recall curve which calculate the average precision score over all protein-function pairs.

B.3 Number of Pretraining Data, Parameters and Training Time

We compare the number of pretraining data, number of parameter, and training time required for pretraining ProtST (Xu et al., 2023) and AMMA. ProtST was pretrained on 553k dataset, while AMMA was pretrained on 120k dataset. AMMA has 111M parameters, while ProtST has 675M parameters. ProtST takes 205 hours of pretraining using 4 Tesla V100 GPUs, while AMMA takes 120 hours using 2 NVIDIA RTX 3090 GPUs. AMMA requires less amount of pretraining data, less number of parameters, and shorter training time, while achieving better performance. This shows that AMMA is a very efficient and powerful model with much less pretraining data, parameters, and training time.

B.4 Details on the t-SNE Visualization

In Figure 1, Figure 4a, and Figure 4b, we visualized representation space of encoders using t-SNE. In this section, we describe in more detail how we performed t-SNE.

To visualize t-SNE in Figure 1, a subset of 1,000 paired data points was randomly selected from the larger 120k pretraining dataset for visualization purposes. Prior to t-SNE visualization, Principal Component Analysis (PCA) was applied to reduce the dimension of the outputs from each modality encoder, namely ESM-1b, GearNet, and PubMedBERT, to 100, and the resulting latent representations were then visualized in 2D using t-SNE. The t-SNE algorithm was executed for 2,500 iterations with a perplexity setting of 200. To incorporate real data insights, two specific instances, representing the proteins 30S ribosomal S13 and 50S ribosomal L22, were included. Furthermore, the 3D structures were visualized using the PDB data available on AlphaFold[1].

In Figure 4a, 500 paired data points were randomly selected for visualization from the 120k pretraining dataset. These paired data were forwarded to our multi-modal encoder without masking, resulting in $Z'_{\texttt{seq},0} \in \mathbb{R}^{L\times 512}$, $Z'_{\texttt{str},0} \in \mathbb{R}^{L\times 512}$, $Z'_{\texttt{func},0} \in \mathbb{R}^{L'\times 512}$ similar to Eq. equation 3. We then average theses vectors over the length dimension to make $z''_{\texttt{seq},0} \in \mathbb{R}^{512}$, $z''_{\texttt{str},0} \in \mathbb{R}^{512}$, $z''_{\texttt{func},0} \in \mathbb{R}^{512}$. Before performing principle component analysis (PCA), we scaled the latent of each modality based on its minimum and maximum values for normalization, and then performed PCA to reduce the dimensionality to 500 components for each z''. Finally, we concatenated the latents and applied t-SNE to visualize into 2D space.

For Figure 4b, we used $z_{\texttt{seq}}$, $z_{\texttt{str}}$, $z_{\texttt{func}}$ from equation 14 instead of $z''_{\texttt{seq}}$, $z''_{\texttt{str}}$, $z''_{\texttt{func}}$ in the above paragraph.

[1] `https://alphafold.ebi.ac.uk`

B.5 Details on the Cosine Similarity Calculation

To compute the cosine similarity in Table 1, we first normalized each latent to the L2 norm. We will denote sequence latent, structure latent, function latent used for the cosine similarity calculation process as $z_{\texttt{seq}}$, $z_{\texttt{str}}$, $z_{\texttt{func}}$. When computing the first column of Table 1, $X'_{\texttt{seq}}$, $X'_{\texttt{str}}$, $X'_{\texttt{func}}$ from 3.1 worked as $z_{\texttt{seq}}$, $z_{\texttt{str}}$, $z_{\texttt{func}}$. When computing the second column, $z_{\texttt{seq}}$, $z_{\texttt{str}}$, $z_{\texttt{func}}$ from B.9 worked as $z_{\texttt{seq}}$, $z_{\texttt{str}}$, $z_{\texttt{func}}$. When computing the third column, $Z'_{\texttt{seq},0}$, $Z'_{\texttt{str},0}$, $Z'_{\texttt{func},0}$ from Eq. equation 3 worked as $z_{\texttt{seq}}$, $z_{\texttt{str}}$, $z_{\texttt{func}}$. We then computed the relation matrix (Relation) for each latent by performing matrix multiplication with its transposed counterparts as follows:

$$\begin{aligned}\text{Relation}_{\texttt{seq}} &= z_{\texttt{seq}} \cdot z_{\texttt{seq}}^T,\\ \text{Relation}_{\texttt{str}} &= z_{\texttt{str}} \cdot z_{\texttt{str}}^T,\\ \text{Relation}_{\texttt{func}} &= z_{\texttt{func}} \cdot z_{\texttt{func}}^T.\end{aligned} \tag{10}$$

The resulting relation matrices contains the interrelationships between the various protein features within the modalities. We then calculated cosine similarity (CosineSim) to capture the relationships between the modalities as follows:

$$\begin{aligned}\text{CosineSim}_{\texttt{seq-str}} &= \frac{\text{Relation}_{\texttt{seq}} \cdot \text{Relation}_{\texttt{str}}}{\max(\|\text{Relation}_{\texttt{seq}}\|_2 \cdot \|\text{Relation}_{\texttt{str}}\|_2, \epsilon)},\\ \text{CosineSim}_{\texttt{seq-func}} &= \frac{\text{Relation}_{\texttt{seq}} \cdot \text{Relation}_{\texttt{func}}}{\max(\|\text{Relation}_{\texttt{seq}}\|_2 \cdot \|\text{Relation}_{\texttt{func}}\|_2, \epsilon)},\\ \text{CosineSim}_{\texttt{str-func}} &= \frac{\text{Relation}_{\texttt{str}} \cdot \text{Relation}_{\texttt{func}}}{\max(\|\text{Relation}_{\texttt{str}}\|_2 \cdot \|\text{Relation}_{\texttt{func}}\|_2, \epsilon)}.\end{aligned} \tag{11}$$

ϵ is a sufficiently small number (10^{-8}) to avoid division by zero. This cosine similarity measure is averaged over the training dataset to get the final value reported in Table 1.

B.6 Details on Experimental Results with Unpaired Data

We further trained the 120k-pretrained AMMA with the 50k dataset consists of 25k sequence-structure pairs and 25k sequence-function pairs. At each step, the batch of sequence-structure pairs and the batch of sequence-function pairs were updated simultaneously. Let us denote sequence and structure data of the sequence-structure pairs as $X'_{\texttt{seq1}}$, $X'_{\texttt{str1}}$ and the sequence and function data of the sequence-function pairs as $X'_{\texttt{seq2}}$, $X'_{\texttt{func2}}$. AMMA was trained using the following objective function:

$$\begin{aligned}\mathcal{L}_{\texttt{seq1}} &= \text{MSE}(\hat{X}_{\texttt{seq1}}, X'_{\texttt{seq1}}),\\ \mathcal{L}_{\texttt{str1}} &= \text{MSE}(\hat{X}_{\texttt{str1}}, X'_{\texttt{str1}}),\\ \mathcal{L}_{\texttt{seq2}} &= \text{MSE}(\hat{X}_{\texttt{seq2}}, X'_{\texttt{seq2}}),\\ \mathcal{L}_{\texttt{func2}} &= \text{MSE}(\hat{X}_{\texttt{func2}}, X'_{\texttt{func2}}),\\ \mathcal{L} &= \mathcal{L}_{\texttt{seq1}} + \mathcal{L}_{\texttt{str1}} + \mathcal{L}_{\texttt{seq2}} + \mathcal{L}_{\texttt{func2}}.\end{aligned} \tag{12}$$

B.7 Details on Symmetric Learning

AMMA-symmetric used the following decoding strategy instead of equation 3:

$$\begin{aligned}\hat{X}_{\texttt{seq}} &= \text{DEC}_{\texttt{seq}}(Z'_{\texttt{seq},0}),\\ \hat{X}_{\texttt{str}} &= \text{DEC}_{\texttt{str}}(Z'_{\texttt{str},1}),\\ \hat{X}_{\texttt{func}} &= \text{DEC}_{\texttt{func}}(Z'_{\texttt{func},2}).\end{aligned} \tag{13}$$

We provide the overall architecture of AMMA-symmetric in Section D.2.

B.8 Details on Attention Visualization

To understand the inter-modal relationship, we concatenated the latent features of sequence, structure, and protein name without masking. After forwarding these features through the multi-modal encoder, we extracted the function-to-residue attention from the whole self-attention map of the last layer and averaged the attention values of each residue token with respect to the function tokens. Then, we selected residues with the highest attention values in the same number as the actual interacting residues annotated from UniProtKB (Consortium, 2019). For visualization, the conformations for the protein-ligand interaction were simulated by AutoDockVina Eberhardt et al. (2021).

B.9 Details on Contrastive Learning

AMMA-contrastive first took a length-wise average over the uni-modal features $X'_{\texttt{seq}} \in \mathbb{R}^{L\times 1280}$, $X'_{\texttt{str}} \in \mathbb{R}^{L\times 3072}$, and $X'_{\texttt{func}} \in \mathbb{R}^{L'\times 768}$ obtained in 3.1 to yield $X''_{\texttt{seq}} \in \mathbb{R}^{1280}$, $X''_{\texttt{str}} \in \mathbb{R}^{3072}$, and $X''_{\texttt{func}} \in \mathbb{R}^{768}$. The features were then processed as follows:

$$\begin{aligned}
z'_{\texttt{seq}} &= \textsc{Proj}_{\texttt{seq}}(X''_{\texttt{seq}}) \in \mathbb{R}^{D},\\
z_{\texttt{seq}} &= \textsc{Enc}_{\texttt{con}}(z'_{\texttt{seq}}) \in \mathbb{R}^{D},\\
z_{\texttt{str}} &= \textsc{Proj}_{\texttt{str}}(X''_{\texttt{str}}) \in \mathbb{R}^{D},\\
z_{\texttt{func}} &= \textsc{Proj}_{\texttt{func}}(X''_{\texttt{func}}) \in \mathbb{R}^{D}.
\end{aligned} \tag{14}$$

Here, $\textsc{Enc}_{\texttt{con}}$ is a sequence encoder for contrastive learning. We adopted the same 8-layer Transformer as $\textsc{Enc}_{\texttt{multi}}$ of AMMA as $\textsc{Enc}_{\texttt{con}}$. By utilizing the following contrastive loss that aligns sequence-structure and sequence-function, AMMA-contrastive was trained to guide sequence features with structure and function information:

$$\begin{aligned}
\mathcal{L}_{\texttt{seq2str}} &= \mathcal{L}_{\text{con}}(z_{\texttt{seq}},\, z_{\texttt{str}}),\\
\mathcal{L}_{\texttt{seq2func}} &= \mathcal{L}_{\text{con}}(z_{\texttt{seq}},\, z_{\texttt{func}}),\\
\mathcal{L}_{\text{reg}} &= \mathcal{L}_{\text{con}}(z_{\texttt{seq}},\, z'_{\texttt{seq}}),\\
\mathcal{L} &= \mathcal{L}_{\texttt{seq2str}} + \mathcal{L}_{\texttt{seq2func}} + \mathcal{L}_{\text{reg}},
\end{aligned} \tag{15}$$

where $\mathcal{L}_{\text{reg}}$ was for regularization and contrastive loss $\mathcal{L}_{\text{con}}$ was defined as follows:

$$\begin{aligned}
\mathcal{L}_{\text{con}}(z_P, z_Q) = -\frac{1}{2N}\sum_{i=1}^{N}\Big(&\log\frac{\exp(z_{P,i}\cdot z_{Q,i}/\tau)}{\sum_{j=1}^{N}\exp(z_{P,i}\cdot z_{Q,j}/\tau)}\\
&+\log\frac{\exp(z_{P,i}\cdot z_{Q,i}/\tau)}{\sum_{j=1}^{N}\exp(z_{P,j}\cdot z_{Q,i}/\tau)}\Big).
\end{aligned} \tag{16}$$

We provide the overall architecture of AMMA-contrastive in Section D.3.

C Additional Experiments

C.1 Effect of the Auxiliary Modalities in Decoders

In addition to sequence features, the AMMA structure decoder takes function features as inputs and the AMMA function decoder takes structure features as inputs. We examined the effect of these auxiliary inputs in Table 6. **AMMA-w/o auxiliary** is the AMMA variant that only takes sequence features as inputs, i.e., that uses the following decoding strategy instead of equation 3:

$$\begin{aligned}
\hat{X}_{\texttt{seq}} &= \textsc{Dec}_{\texttt{seq}}(Z'_{\texttt{seq},0}),\\
\hat{X}_{\texttt{str}} &= \textsc{Dec}_{\texttt{str}}(Z'_{\texttt{seq},1}),\\
\hat{X}_{\texttt{func}} &= \textsc{Dec}_{\texttt{func}}(Z'_{\texttt{seq},2}).
\end{aligned} \tag{17}$$

Table 6: **Experimental results without auxiliary structure or function inputs in the AMMA decoders.** The pretraining is conducted using a 22k dataset, a random subset of the 120k dataset.

Method	EC		GO-MF		Average
	F_{max}	AUPR	F_{max}	AUPR	
AMMA	87.7	89.8	66.4	64.2	**77.0**
AMMA-w/o auxiliary	87.7	90.1	65.0	63.6	76.6

As shown in the table, the auxiliary inputs to the decoders were beneficial to the performance. This is because structure and function are difficult to predict based on sequence alone, making pretraining of AMMA-w/o auxiliary too challenging. As structure and function are relatively well aligned and therefore easy to predict from each other, the auxiliary inputs aid the structure and function decoders.

D Architecture

In this section, we illustrate the architecture of our model, AMMA, and two variants of AMMA: AMMA-symmetric and AMMA-contrastive.

D.1 AMMA

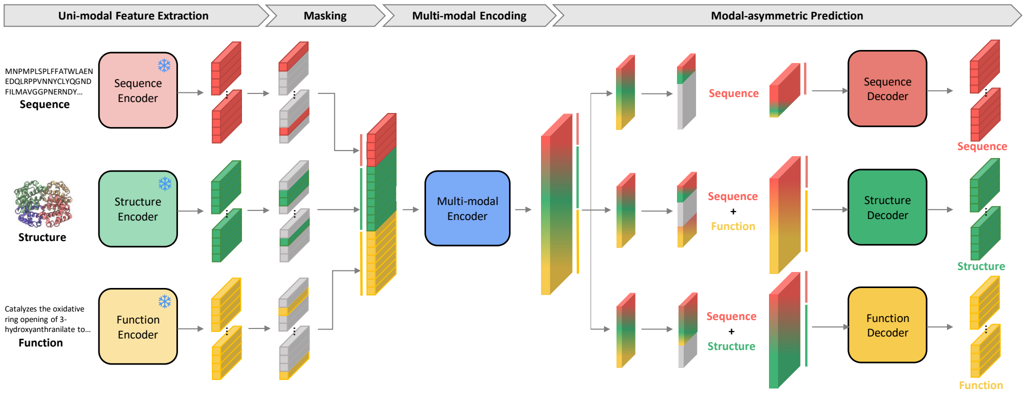


Figure 5: **The overall architecture of AMMA.**

Architecture of AMMA consists of four procedures: uni-modal feature extraction, masking, multi-modal encoding, and model-asymmetric decoding as shown in Figure 5. First, we extracted the features of each modality using a feature encoder: ESM-1b (Rives et al., 2021), GearNet (Zhang et al., 2023), PubMedBERT-abs (Gu et al., 2021). To be specific, the sequence encoder, ESM-1b (Rives et al., 2021) is a model with 33 layers of Transformers with ∼650M parameters. The structure encoder, GearNet (Zhang et al., 2023) is composed of 6 GearNet layers, using hidden dimension of 512. The function encoder, PubMedBERT-abs (Gu et al., 2021) is a model based on 12 layers of BERT (Devlin et al., 2019). Then, we masked the latents, leaving only 160 latents in total according to the masking sampling proposed in 3.1. Afterwards, we have a multi-modal encoder to fuse all modalities into a unified representation space. Finally, we performed asymmetric decoding, which reconstructs structure latent vector from sequence and function information and function latent vector from sequence and structure information, allowing AMMA to capture asymmetric sequence-structure-function relationships. The multi-modal encoder is an 8-layer transformer and the three decoders are 2-layer transformers each.

D.2 AMMA-symmetric

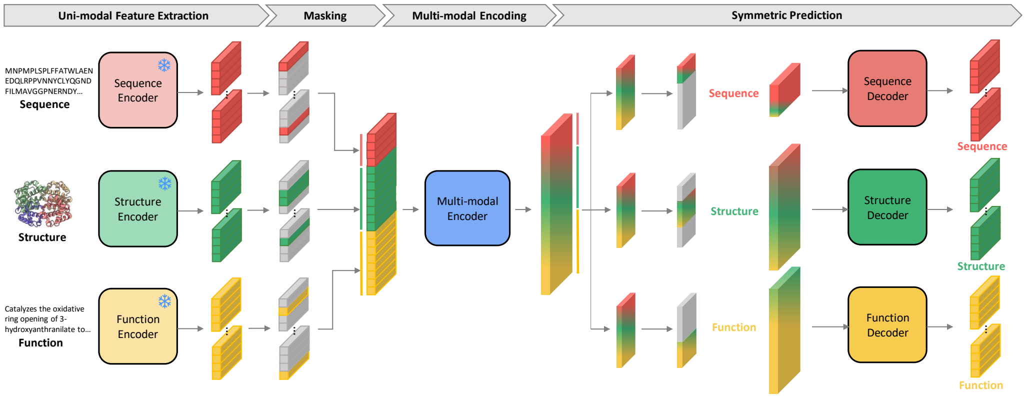


Figure 6: **The overall architecture of AMMA-symmetric.**

In AMMA-symmetric, the only part that differs from the original AMMA lies in the design of decoders. Each decoder processes and predicts its corresponding modality from corresponding inputs as shown in Figure 6. Specifically, structure latent vector is employed to structure decoder to predict the original structure latent vector. Also, function latent vector is employed to function decoder to predict the original function latent vector.

D.3 AMMA-CONTRASTIVE

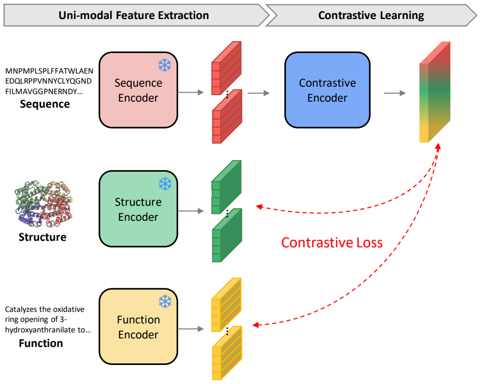


Figure 7: **The overall architecture of AMMA-contrastive.**

For AMMA-contrastive, we introduced contrastive encoder with 8-layers of transformers as shown in Figure 7. To train the AMMA-contrastive, we utilized contrastive loss between the output of contrastive encoder and structure encoder, and between the output of contrastive encoder and function encoder. We also took advantage of the regularization loss between the input and output of the contrastive encoder to regularize the output of the sequence feature extractor and the ouput of the contrastive encoder.