MSA Generation with Seqs2Seqs Pretraining: Advancing Protein Structure Predictions

Le Zhang[1,3]*, **Jiayang Chen**[4]* , **Tao Shen**[5], **Yu Li**[4†], **Siqi Sun**[1,2†]
[1] Fudan University [2] Shanghai Artificial Intelligence Laboratory [3] Mila, Université de Montréal
[4] The Chinese University of Hong Kong [5] Zelixir Biotech
le.zhang@mila.quebec, liyu@cse.cuhk.edu.hk, siqisun@fudan.edu.cn

Abstract

Deep learning models like AlphaFold2 (Jumper et al., 2021) have revolutionized protein structure prediction, achieving unprecedented accuracy. However, the dependence on robust multiple sequence alignments (MSAs) continues to pose a challenge, especially for proteins that lack a wealth of homologous sequences. To overcome this limitation, we introduce MSA-Generator, a self-supervised generative protein language model. Trained on a sequence-to-sequence task using an automatically constructed dataset, MSA-Generator employs protein-specific attention mechanisms to harness large-scale protein databases, generating virtual MSAs that enrich existing ones and boost prediction accuracy. Our experiments on CASP14 and CASP15 benchmarks reveal significant improvements in LDDT scores, particularly for complex and challenging sequences, enhancing the performance of both AlphaFold2 and RoseTTAFold. The code is released at https://github.com/lezhang7/MSAGen.

1 Introduction

The challenge of protein structure prediction (PSP), a central issue in structural biology, has undergone remarkable transformation thanks to advances in deep learning. Among these developments, AlphaFold2 (AF2) stands out for its exceptional performance, primarily attributed to its effective use of multiple sequence alignments (MSAs) (Jumper et al., 2021). MSAs are constructed by querying a protein sequence against extensive databases using sophisticated search algorithms, resulting in collections of homologous sequences that encapsulate evolutionary information. This information serves as the cornerstone for many PSP models. However, not all protein sequences have a wealth of homologous counterparts. In such cases, even the most advanced search algorithms may struggle to construct high-quality MSAs, thereby limiting the performance of MSA-dependent models like AF2 (Jumper et al., 2021; Wang et al., 2022a), as illustrated in fig. 1.

Inspired by the generative capabilities of language models (Raffel et al., 2020; Touvron et al., 2023; Chung et al., 2022; Chen et al., 2021), we recognize their potential to extend beyond textual data. Viewing protein sequences through this lens, we propose an innovative approach to generating virtual yet constructive MSAs, akin to generating text. These novel alignments provide supplemental evolutionary information, thereby enhancing the efficacy of protein structure predictions. Within the realm of PSP, tertiary structure prediction remains a critical challenge, occupying a central role in molecular biology by revealing intricate protein functions and interactions. While secondary structure predictions (Rost & Sander, 1993) offer valuable insights, it is the tertiary predictions that provide a comprehensive understanding of a protein's complex conformation.

[*] Works done during internship at Shanghai Artificial Intelligence Laboratory, co-first authors

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

https://doi.org/10.52202/079017-1827

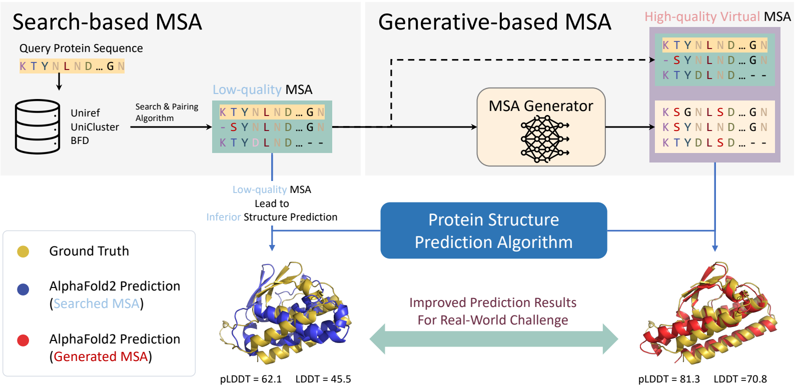


Figure 1: (**Top Left**) Certain protein sequences lack rich homologs, leading to poor MSA quality with conventional search algorithms. (**Top Right**) We propose a generative model MSA-Generator to produce informative MSA for these queries, offer a potential solution to such challenge.

For bio-tasks such as protein structure prediction, gene sequence alignment, RNA secondary structure determination, microbial community analysis, and evolutionary tree construction, where enhancing downstream performance necessitates multiple sequences, we introduce the sequences-to-sequences (seqs2seqs) generation task. Unlike the conventional sequence-to-sequence (seq2seq) tasks—e.g., machine translation, which requires a strict one-to-one correspondence between a source sequence x and a target sequence y—seqs2seqs is designed for flexibility. The task aims to generate multiple coherent sequences from a given sequences. Each generated sequence preserves patterns from the input, but there's no strict correspondence between the input and output. Instead, we prioritize maintaining interconnected patterns across them. This design endows the task with a self-supervised nature due to its intrinsic adaptability.

In the context of protein, such flexibility enables easy extraction of a portion of the MSA as the source, with the remainder acting as the target. Harnessing search algorithms, our framework adeptly extracts source and target data from comprehensive protein databases, paving the way for self-supervised pre-training. To our knowledge, this marks an initial step in tapping into self-supervised generative pretraining to bolster protein structure prediction accuracy.

We introduce MSA-Generator, a protein language model pre-trained using the seqs2seqs as its pretext task. Specialized in simultaneously generating multiple sequences, it effectively captures global structural information from input MSAs. This approach facilitates rapid, *de novo* sequence creation, improves inferior MSAs (see fig. 1), and boasts adaptability across various protein domains, adeptly navigating computational challenges.

To summarize the main contribution of this article:

- **Innovative unsupervised Seqs2Seqs Task Proposition:** We propose the unsupervised sequences-to-sequences (seqs2seqs) task, a promising approach for generating informative protein sequences, with potential applications extending to other areas like RNA. Integrated with search algorithms, this task streamlines generative pre-training by automating data retrieval from expansive protein databases.

- **Launch of MSA-Generator Model:** MSA-Generator, our state-of-the-art generative protein model, is uniquely devised to employ the seqs2seqs task on a self-curated dataset. It is optimized for multi-sequence generation, skillfully extracting global MSA insights, and has demonstrated adaptability across diverse protein domains.

- **Robust Empirical Validation:** We validate our approach's potency, showcasing significant improvements on AlphaFold2 and RoseTTAFold using CASP14 and CASP15 benchmarks. This signifies a practical leap forward in tackling the intricate challenge of protein folding. It also showcases the promise of seqs2seqs pretraining in the field of bioinformatics.

https://doi.org/10.52202/079017-1827

2 Related Work

Protein Structure Prediction Proteins, despite their immense diversity, are composed of just 20 unique amino acids. Their physical structures, which dictate their functions and characteristics, are fundamental to understanding the essence of life. While the field has witnessed significant advancements like AF2 (Jumper et al., 2021) and RoseTTAFold (Baek et al., 2021), challenges remain. The success of AF2 can be ascribed to its adept utilization of MSAs, constructed by search algorithms such as DeepMSA (Zhang et al., 2019), JackHMMER (Johnson et al., 2010) and MMseqs2 (Steinegger & Söding, 2017) across vast databases including UniRef (Suzek et al., 2007) and BFD based on a protein query sequence. Conversely, single-sequence prediction methodologies (Chowdhury et al., 2021; Lin et al., 2022; Chowdhury et al., 2022; Wu et al., 2022b) often underperform in comparison. However, for protein queries devoid of extensive family representations, obtaining quality MSAs is challenging. Consequently, the proficiency of MSA-driven techniques diminishes. In this context, our proposed method leverages generative techniques to combat the paucity of homologs in protein sequences, presenting a solution when traditional techniques falter.

Protein Language Models Language models, initially designed for language processing, have found a expanding role in bioinformatics, primarily for protein sequence representation. This improvement in performance largely stems from the adaptation of the masked language modeling (MLM) strategy, a concept inspired by BERT (Devlin et al., 2019). The ProtTrans series, which includes models such as ProtBert, ProtTXL, ProtXLNet, and ProtT5 (Elnaggar et al., 2021), along with ProteinBERT (Brandes et al., 2022) and ESM models like ESM-1b (Rives et al., 2021) and ESM-2 (Lin et al., 2022), exemplifies the transformative role of Transformer architectures in this field. Further reinforcing the connection between language models and protein structure prediction, research has demonstrated associations between protein representations learned by these models and contact maps, revealing evolutionary patterns that are crucial to the success of AlphaFold2 (AF2) (Vig et al., 2020; Rao et al., 2020). These findings have shifted research focus back towards leveraging multiple sequence alignments (MSAs) rather than relying solely on single sequences. A notable example is the MSA Transformer (Rao et al., 2021), which applies the MLM strategy specifically to MSAs, thereby capturing rich evolutionary information and advancing the field of protein structure prediction.

Protein Sequence Generation Beyond masked language modeling (MLM), a variety of generative techniques exist, each with its distinct objectives and methodologies. For instance, Potts models (Figliuzzi et al., 2018; Russ et al., 2020) are crafted specifically for individual MSA sets from which they're derived (Zhang et al., 2022). However, their shortcomings in adapting to different MSA sets (Sgarbossa et al., 2022) have spurred the development of generative language models. An exemplar, ESMPair (Chen et al., 2022), constructs MSAs of Interologs by classifying sequences based on their taxonomic lineage. In contrast, both ProGen (Madani et al., 2020) and ProGen2 (Nijkamp et al., 2022) focus on single-sequence generation, sidestepping the integral component of MSA. Another research direction involves VAE-based models (Riesselman et al., 2018; McGee et al., 2021; Sinai et al., 2017), originally developed for mutation evaluation. The challenge of efficiently sampling from distributions to create diverse and long sequences limits their application to downstream tasks. A study by Sgarbossa et al. (2022) utilized the MSA Transformer in a repetitive mask-and-reveal methodology, which unfortunately led to a compromise in sequence diversity. Of notable mention is EvoGen (Zhang et al., 2022), aiming parallelly at producing virtual MSAs for Protein Structure Prediction. However, EvoGen uniquely operates as a meta-generative model, requiring guidance from Alphafold2 to hone its MSA generation prowess.

Despite the lengthy context associated with MSA Generation, our work **connects self-supervised learning with MSA generation**. We highlight generative MSA pretraining, introducing an unsupervised sequences-to-sequences task specifically tailored for efficient MSA generation. To the best of our understanding, this marks a significant stride in the realm of protein sequence generation.

3 Sequences-to-Sequences Generative Pretraining

We present the Sequences-to-Sequences generation task and the methodology for automatic dataset construction in section 3.1. Details of the proposed MSA-Generator are provided in section 3.2.

https://doi.org/10.52202/079017-1827

In section 3.3, we delve into the ensemble approach of MSA-Generator for optimizing the Protein Structure Prediction (PSP) task.

3.1 Sequences-to-Sequences Generation For Protein Sequences

Addressing the challenge of sparse homologous matches in Multiple Sequence Alignments (MSA) within real-world databases, we introduce the Sequences-to-Sequences (seqs2seqs) Task, designed for creating virtual MSAs. This approach differs from the conventional sequence-to-sequence frameworks used in machine translation, which typically enforce a strict one-to-one correspondence between source sequence x and target sequence y. Instead, seqs2seqs allows for more flexibility, focusing on **identifying shared intrinsic patterns** between source sequences X and target sequences Y. For protein MSAs, this involves **integrating extensive evolutionary data**, both across and within the sequences, to reveal co-evolutionary relationships.

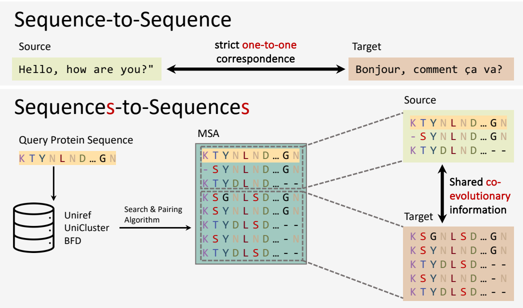


Figure 2: Difference between seq2seq and seqs2seqs and Automated Data Collection Process.

The inherent adaptability of the seqs2seqs model permits the self-supervised nature of the task and seamless gathering of substantial quantities of source and target sequences from protein sequence databases. This is achieved by deploying sequence searching algorithms like JackHMMER (Johnson et al., 2010) and MMseqs2 (Steinegger & Söding, 2017). Our process began with selecting sequences from the UniRef90 database (Suzek et al., 2007) as initial queries. Subsequently, the JackHMMER algorithm (Johnson et al., 2010) was employed iteratively to identify homologous sequences within the database, based on the query sequences following MSA dataset construction pipeline of AlphaFold2. This process was iterated until no additional sequences emerged, searching parameters are detailed in appendix C. For every batch of sequences retrieved, a random selection was made, designating query with some as the source X and the remainder as the target Y, as illustrated in fig. 2. Notably, the assurance of co-evolutionary relationships is intrinsically facilitated by the search algorithm's mechanism.

3.2 Sequences-to-Sequences Architecture

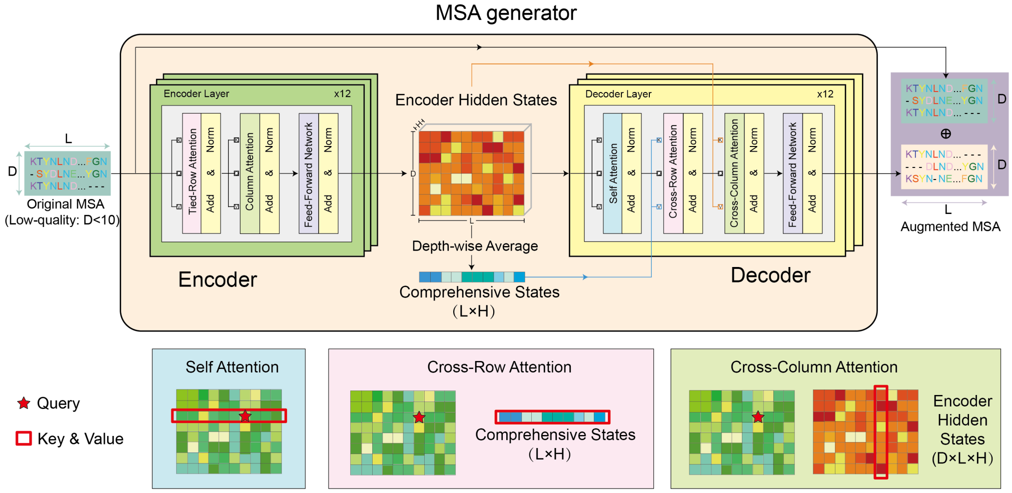


Figure 3: **MSA-Generator Overview** (Top) Overview of the architecture, processing pipeline, and module attention operations. (Bottom) Illustration of the attention mechanism. A red star represents a single query position, and the red boxes indicate keys and values utilized in attention processing and calculations.

We pretrain a transformer-based model (Vaswani et al., 2017), denoted as MSA-Generator, via the unsupervised Sequences-to-Sequences task. The MSA-Generator framework incorporates an encoder-decoder structure. The encoder contextualizes the input MSA data, while an auto-aggressive decoder produces sequences derived from this context (refer to section 3). To capture expansive evolutionary information from the input MSA both horizontally and vertically, the encoder integrates the tied-row and column attention mechanism (Rao et al., 2021). As the decoder concurrently generates multiple sequences—interacting with each other and the input MSA—it is enhanced with two additional modules beyond the conventional transformer. The *Cross-Row Attention* is designed to efficiently acquire a global representation by amalgamating comprehensive states. Meanwhile, to emphasize the vital conservative trait of amino acids (Dayhoff et al., 1978; Henikoff & Henikoff, 1992; Jones et al., 1992), we introduce the *Cross-Column Attention*, which, during the generation of the token at time step t, directs its attention to the t-th token of all input sequences.

Tied-Row Attention Building upon the foundation laid by the MSA Transformer (Rao et al., 2021), we incorporate a shared-attention mechanism. This is achieved by aggregating the attention map weights across each sequence from MSA $\in \mathbb{R}^{D\times L}$ prior to applying the softmax function. Notably, each sequence utilizes the same attention weight matrix. For the d-th row, the associated query, key, and value matrices are denoted as Q_d, K_d, and $V_d \in \mathbb{R}^{L\times h}$, respectively. These matrices are derived via three distinct learned projection matrices. The computation of the shared attention weight matrix is formulated as:

$$W_{TR} = \text{softmax}\left(\sum_{d=1}^{D} \frac{Q_d K_d^T}{\lambda(D,h)}\right) \in \mathbb{R}^{L\times L} \tag{1}$$

In this context, $\lambda(D,h) = \sqrt{Dh}$ serves as the square-root normalization. This normalization is crucial in mitigating potential linear scaling of attention weights with the sequences. The resultant representation for the d-th row is obtained through $W_{TR}V_d$.

It's important to highlight that, in our decoder, we deliberately bypass the tied-row attention. This decision aids in maintaining diversity in the generated sequences. Instead, we lean towards a conventional self-attention mechanism.

Cross-Row Attention Contrary to tasks like machine translation, where the target attends only to a single input during decoding, the essence of seqs2seqs lies in discerning intrisic patterns common to both source and target sequences. This necessitates a holistic comprehension of the input, implying that when generating a sequence, the decoder should attend to the entirety of the input. A naive concatenation would yield a representation with dimensions $\mathbb{R}^{D\cdot L\times h}$, rendering it computationally expensive and thus impractical.

To address this, we introduce an efficient strategy that calculates the depth-wise average pooling of encoder hidden states H_{enc}, represented as $H_c = \frac{1}{D}\sum_{d=1}^{D} H_{enc}^{d} \in \mathbb{R}^{L\times h}$. This serves as a global representation of the input and is crucial for cross-attention during decoding. Here, $K_c = H_c W_k$ and $V_c = H_c W_v$ signify the key and value matrices, while $Q = X_{dec} W_q$ stands for query matrix projected from decoder hidden states X_{dec}. The Cross-Row attention is:

$$\text{CR-Attn}(Q, K_c, V_c) = \text{softmax}(\frac{QK_c^T}{\sqrt{h}})V_c \tag{2}$$

Each sequence generation process can access comprehensive information, attending to the same keys and values, mirroring the co-evolutionary patterns of the input MSA simultaneously thus **permitting fast parallel generation of multiple sequences** (middle at bottom in fig. 3).

Self/Cross -Column Attention In MSA, each column represents residues or nucleotides at a specific position across sequences, revealing conserved regions essential for understanding biological functions, structural stability, or evolutionary significance (Dayhoff et al., 1978; Jones et al., 1992; Henikoff & Henikoff, 1992). Drawing inspiration from the vertical-direction attention proposed in Ho et al. (2019), we introduce a self-column attention in encoder for a comprehensive representation akin to Rao et al. (2021), and a cross-column attention in decoder to capture conservation characteristics.

To facilitate both attention mechanisms, the representation matrix $X \in \mathbb{R}^{D \times L \times h}$ needs to be transposed prior to the execution of self and cross attention:

$$\text{ColAtt}(Q_{col}, K_{col}, V_{col}) = \left(\text{softmax}\left(\frac{Q_{col}K_{col}^T}{\sqrt{h}}\right) V_{col}\right)^T \tag{3}$$

For the self-column attention, projections from X^T yield $Q_{col}, K_{col}, V_{col} \in \mathbb{R}^{L \times D \times h}$. In contrast, for the cross-column attention, Q_{col} is determined from decoder hidden states as $X_{dec}^T W_q$, whereas K_{col} and V_{col} are projected from encoder hidden states as H_{enc}^T (see fig. 3 bottom right).

Pre-training objective We employ the seqs2seqs task to pretrain MSA-Generator. For a given source MSA $X \in \mathbb{R}^{D \times L}$, the loss is computed with respect to the target MSA $Y \in \mathbb{R}^{D' \times L}$ as follows:

$$\mathcal{L}_{seqs2seqs} = -\frac{1}{D' \times L} \sum_{d=0}^{D'} \sum_{l=0}^{L} \log P(y_l^d | y_{<l}^d, X) \tag{4}$$

It's crucial to note that each sequence $y \in Y$ is generated referencing the entire source matrix X, and this generation occurs in parallel owing to the thoughtful design of the architecture.

Pretrained MSA-Generator adopts 12 transformer encoders/decoders with 260M parameters, 768 embedding size, and 12 heads. It's pretrained with ADAM-W at a $5e^{-5}$ rate, 0.01 linear warm-up, and square root decay for 200k steps on 8 A100 GPUs, batch size of 64, using a dataset containing 2M MSAs constructed as described in section 3.1.

3.3 Generation and Ensemble Strategy

During inference, for every query sequence x, we initially use a search algorithm to assemble a MSA denoted as X. This is subsequently inputted into MSA-Generator to produce MSA Y. The concatenated MSA $X \oplus Y$ serves as input for the subsequent task. Nucleus sampling (Holtzman et al., 2019), set with *top-p=50* and *top-k=10*, is implemented to foster unique sequences and curtail redundancy.

For the purpose of optimizing the PSP task and yielding informative sequences, we adopt pLDDT as our selection criterion, leveraging our ability for swift MSA generation. pLDDT (Kryshtafovych et al., 2019; Jumper et al., 2021) measures the accuracy of predicted inter-residue distances for each residue in a protein, serves as a confidence indicator, with elevated scores hinting at potentially more accurate predictions. Utilizing pLDDT, we enhance each MSA through multiple runs, computing corresponding pLDDT scores for each. The MSA with the premier pLDDT score is subsequently selected as the optimal ensemble result and employed to determine the prediction accuracy relative to the ground truth.

4 Empirical Validation

4.1 Setup

The tertiary structure of a protein is pivotal, directly determining its functionality. In structural biology, the tertiary structure not only reveals the overall conformation of a protein but also inherently includes insights from secondary structures (Rao et al., 2021; Jones, 1999), such as the arrangement and orientation of α-helices and β-sheets, and from contact predictions (Wang et al., 2017), denoting the spatial interactions between amino acid pairs. In essence, the tertiary structure offers a holistic view, allowing direct inference of localized structural features and interactions between amino acids. Leveraging tools like AlphaFold2 lets us directly obtain this comprehensive structural data, thereby bypassing intermediary steps like secondary structure and contact prediction. Given the centrality of tertiary structure prediction in protein functional studies, we have prioritized this task and adopted Local Distance Difference Tests [2] (LDDT) (Mariani et al., 2013) and pLDDT as our metric for evaluating prediction accuracy. We also adopt Template modeling score (TM-Score) and Global distance test (GDT-TS) to measure global structure prediction accuracy. Specifically, we assess

[2] OpenStructure is used for LDDT calculation

https://doi.org/10.52202/079017-1827

MSA-Generator by comparing preotein tertiary structures predicted from PSP algorithm, namely AlphaFold2 and RoseTTAFold, with various input MSAs. Demonstrating the usefulness of generated MSAs and the efficacy of MSA-Generator.

Benchmark & Dataset We employ CASP14/15 as our test set, a prestigious dataset that encompasses proteins from a broad spectrum of biological families. The creation of a vast protein structure prediction dataset is prohibitively expensive, and given that AF2 has already trained on all previously available structures, this dataset emerges as the best evaluation benchmark. It's important to highlight that **sequences from CASP14/15 aren't part of our pretraining dataset**, and our evaluations precede the AF2 version updated with CASP14/15 information.

Our primary interest lies in challenging protein sequences devoid of homologues, rendering traditional search algorithms ineffective. For every query in our test dataset, we use JackHMMER to search within UniRef90, which contains 70 million sequences, in order to gather related homologues. We define two scenarios: (1) *artificial challenging MSAs* , where we purposefully pick the top 15 homologues for each test set query as an *artificial gold standard*. From these, 5 homologues are further sampled as the *artificial baseline*, offering a synthetic challenge. (2) *real-world challenging MSAs* , which includes 20 sequences from test set, each with homologues less than 20, significantly challenge PSP algorithms. All assessments are executed in a zero-shot setting.

4.2 Are Virtual MSAs as good as Real MSAs ?

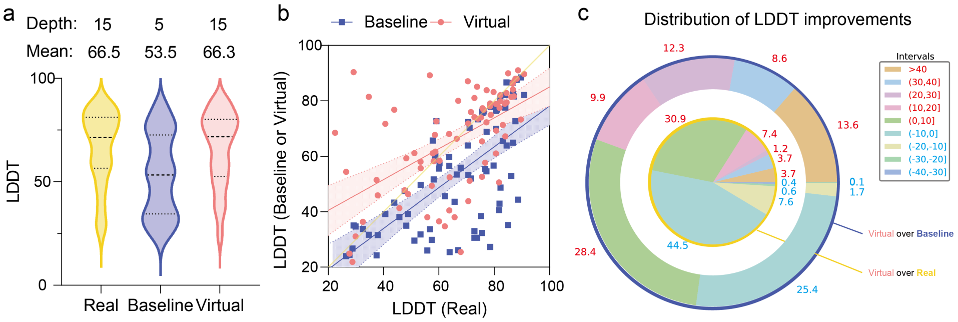


Figure 4: **Artificial Challenging Cases Results from AlphaFold2** (a) Violin plots of LDDT distribution. (b) x-axis represents LDDT of *artificial gold standard*, and the y-axis represents LDDT of *artificial baseline* and *artificial augmentation*. Dashed-line represents 95% confidence intervals (c) Pie chart of LDDT improvements in intervals. The inner circle represents a comparison with the baseline, while the outer circle represents a comparison with the real.

We employ *Artificial challenging MSAs* to thoroughly compare our generated virtual MSAs with conventional searched real MSAs to demonstrate that virtual MSAs can closely approximate real ones in downstream tasks. Specifically, for every *Baseline* MSA, we deploy MSA-Generator to produce a *Virtual* MSA that poses the same depth of the *Real* MSA. For each MSA, we employ an ensemble of three runs using the strategy outlined in section 3.3. Refer to fig. 4 for the results.

The fig. 4 (a) shows that the LDDT distribution of the *baseline* suffers a sharp decline when reduced from 15 to 5 sequences. This underscores the importance of MSA quality for cutting-edge PSP algorithms. Yet, when supplemented by MSA-Generator's generative virtual MSA, the gap to the *Real* narrows considerably, reflecting a LDDT enhancement of 12.8. This underscores the effectiveness of our generated MSAs.

A more granular observation in fig. 4 (b) illustrates that the majority of *baseline* data points are below the diagonal, while most *Virtual* points sit above it, some even considerably outpacing the *Real*. fig. 4 (c) illustrates the statistics on improvements in intervals. It's evident that our generative virtual MSAs effectively improve results for 72.8% of protein sequences (over Baseline). Remarkably, nearly half of the generated virtual MSAs even outperform the real searched MSAs. This emphasizes the importance of generative MSAs and the potential of the seqs2seqs task in uncovering co-evolutionary patterns within bio-sequences. Among them, there are even 6 generative virtual MSA that surpass real MSA by more than 30 LDDT, including most notable T1032-D1 (**+46.02** LDDT), T1054-D1 (**+46.8** LDDT), and T1070-D2 (**+61.22** LDDT), suggesting that without generative virtual MSA,

https://doi.org/10.52202/079017-1827

current PSA algorithm may fail on these queries. Comprehensive results for individual MSAs can be found in the Appendix.

4.3 Real-World MSA Challenges

PSP Algorithm	CASP14 (avg. Det. = 6.1)				CASP15 (avg. Det. = 7.4)			
	pLDDT↑	LDDT↑	TM-Score ↑	GDT-TS ↑	pLDDT↑	LDDT↑	TM-Score↑	GDT-TS ↑
single-sequence-based								
ESMFold	43.3	41.9	0.56	0.47	46.0	53.4	0.47	0.41
OmegaFold	-	43.1	0.51	0.44	-	49.6	0.44	0.39
MSA-based								
RoseTTAFold	63.5	51.3	0.56	0.51	62.6	52.1	0.53	0.46
RoseTTAFold+Potts Generation	63.2	48.9	0.56	0.50	94.4	51.0	0.52	0.46
RoseTTAFold+Iterative Unmasking	63.9	52.2	0.57	0.53	65.3	55.3	0.55	0.48
RoseTTAFold+MSA-Generator	68.9$_{+5.4}$	56.3$_{+5.0}$	0.61$_{+5\%}$	0.56$_{+5\%}$	69.0$_{+6.4}$	58.4$_{+6.3}$	0.58$_{+5\%}$	0.50$_{+4\%}$
AlphaFold2	65.1	53.2	0.59	0.54	65.1	55.6	0.55	0.49
AlphaFold2+Potts Generation	63.8	50.9	0.60	0.55	64.5	52.6	0.56	0.48
AlphaFold2+Iterative Unmasking	65.2	54.6	0.61	0.57	69.5	57.3	0.57	0.51
AlphaFold2+MSA-Generator	71.6$_{+6.4}$	57.5$_{+4.3}$	0.65$_{+6\%}$	0.60$_{+6\%}$	73.7$_{+8.6}$	63.7$_{+8.1}$	0.61$_{+6\%}$	0.53$_{+4\%}$

Table 1: **Real-World MSA Challenges** average pLDDT, LDDT and RMSD enhancement scores, averaged over 3 runs; avg. Det. represents average depth of MSA.

Our ultimate goal is to produce high-quality MSAs for protein sequences with few homologues. Current search algorithms often fail to construct quality MSAs for these, making PSP algorithms similarly struggle with accurate predictions. The *Real-world challenging MSAs* evaluation is devised to test the efficacy of the seqs2seqs generative pretraining approach in addressing this challenge. For this, we curate sequences with fewer than 20 homologues using the search method detailed in section 4.1. For every identified MSA, we employ MSA-Generator to generate an MSA of identical depth across three independent runs. Subsequently, we measure the ensemble LDDT by inputting them to the PSP algorithm following section 3.3. For comparison, our benchmarks include the strong single-sequence folding technique, ESMFold (Lin et al., 2022) and OmegaFold (Wu et al., 2022a); we apply the same evaluation set up for the iterative unmasking strategy highlighted in (Sgarbossa et al., 2022); and generation with Potts models (Figliuzzi et al., 2018).

Table 1 presents the pLDDT, LDDT, TM-Score and GDT-TS improvements achieved through various MSA generation techniques across different models. The single-sequence-based models lags behind MSA-based strategies. AF2 consistently outperforms RoseTTAFold in both metrics. Potts models, intriguingly, don't demonstrate a pronounced ability to produce effective MSAs. This is evidenced by their marginally reduced average performance in both metrics, echoing findings from (Sgarbossa et al., 2022; Rao et al., 2020). While iterative unmasking with MSA Transformer (Sgarbossa et al., 2022) can generate usable MSAs in certain scenarios, the MSAs it produces are often less diverse due to the inherent unmasking process, thus limiting its enhancement potential.

In contrast, the method we propose demonstrates significant enhancements across all metrics in both models. This suggests that the MSAs produced by our approach are predominantly informative. In particular, 75% of the MSAs experienced substantial improvement, leading to an average increase in LDDT scores of 4.3 for CASP14 and 8.1 for CASP15. Our observations reveal a consistent 6% improvement in both TM-Score and GDT-TS for CASP14, alongside a 6% increase in TM-Score and a 4% rise in GDT-TS for CASP15. These results underscore our method's efficacy in enhancing both local and global structure predictions.

Remarkable LDDT enhancements were observed in T1093-D1 (with 3 homologous), moving from 45.5 to 70.77, and in T1113 (with 10 homologous), rising from 32.6 to 80.6. However, in specific MSAs, like T1094-D2 (7 homologous, pLDDT +2.8, LDDT -0.1), T1099-D1 (8 homologous, pLDDT +1.11, LDDT -0.5), and T1178 (15 homologous, pLDDT +2.2, LDDT -12.3), an elevation in pLDDT was offset by a decline in LDDT. This suggests that pLDDT might not always be a consistent indicator for selection strategy outlined in section 3.3. Detailed

Orphan25	**pLDDT**	**LDDT**	**TM-Score**	**GDT-TS**
AlphaFold2	77.2	61.6	0.61	0.62
AlphaFold2+Potts Generation	68.9	49.3	0.49	0.43
AlphaFold2+Iterative Unmasking	78.9	62.5	0.64	0.63
AlphaFold2+MSA-Generator	81.8	66.4	0.69	0.67

Table 2: Comparison of AlphaFold2 variants on the Orphan25 dataset.

results for each individual MSA are available in the appendix E A more challenging aspect of our study involves enhancing the prediction of structures for orphan protein sequences. To evaluate our method in this context, we included results from an orphan protein family, Orphan25 Wang et al. (2022b). We employed MMseqs2 to search against the UniRef30 and ColabFoldDB databases Mirdita et al. (2022), with ColabFoldDB being an extension of BFD/MGnify, enriched with metagenomic sequences from diverse environments. By selecting sequences without homologues, we identified a test set comprising 10 proteins (6WKY, 6WL0, 6XA1, 6XN9, 6XYI, 7A5P, 7AL0, 7JJV). The outcomes, summarized in table 2, indicate that our approach demonstrates substantial benefits for orphan protein sequences. This suggests that our method is particularly advantageous for challenging inputs.

4.4 Re-Evaluating pLDDT as a Selection Metric

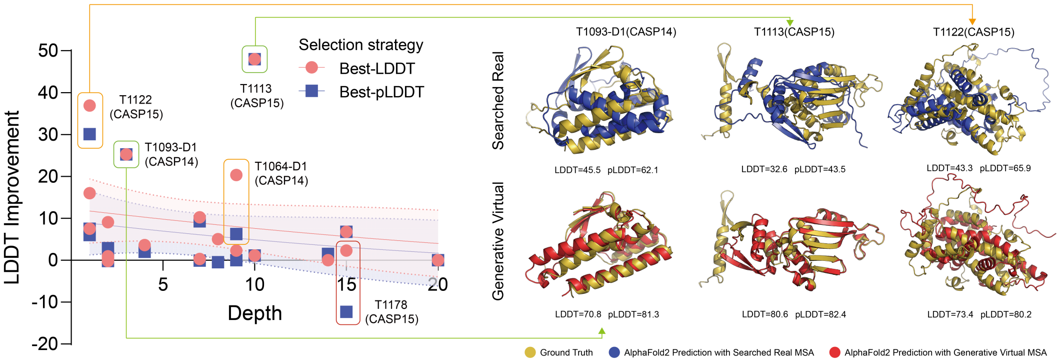
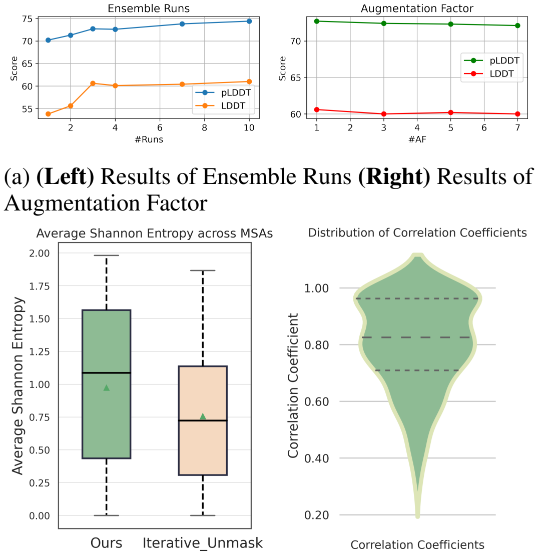


Figure 5: (Left) LDDT improvement selected by different criteria; (Right) Protein structure visualization with pLDDT and LDDT selected by Best-pLDDT.

We sought to examine the effectiveness of pLDDT as a criterion. As previously highlighted, for specific proteins, improvements in pLDDT do not necessarily correlate with increases in LDDT. To delve deeper, we conducted an experiment where LDDT was directly calculated for each enhanced MSA in section 4.3. We then selected the highest LDDT as the output, bypassing the use of pLDDT as an intermediary metric.

The disparity between pLDDT-based and LDDT-based predictions shown in fig. 5 suggests that pLDDT may not always be the best criterion. A noticeable gap exists between the two criteria (evident in the blue and red regions). For example, proteins T1064-D1 (depth=9) and T1122 (depth=1) exhibit significant gaps between scores chosen by LDDT versus pLDDT, highlighted in red boxes. Interestingly, for T1178 (depth=15), the highest pLDDT selection scores -12.3 against the baseline, while LDDT selection results in a +2.7. This implies that some generated MSAs, even with lower pLDDT scores, can enhance the LDDT. This indicates that our approach has untapped potential that could benefit from more nuanced selection criteria. The ideal situation would be for both predictions to produce the same significant improvement, as emphasized by the green boxes and visualized in fig. 5 (Right). Notably, MSA-Generator shows notable improvements for protein sequences with few homologs, emphasizing its utility in real-world protein folding challenges.



(a) **(Left)** Results of Ensemble Runs **(Right)** Results of Augmentation Factor

(b) **(Left)** Box Plot of Averaged Shannon Entropy **(Right)** Violin Plot of Pearson correlation coefficient between Searched Real PSSM and Generated Virtual PSSM

Figure 6: Ablation and MSA feature.

https://doi.org/10.52202/079017-1827

4.5 MSA Diversity & Conservation

We directly evaluated the generated MSAs based on two key characteristics: **diversity** and **conservation**. All experiments setup are consist with section 4.3. To measure diversity, we analyzed the average Shannon Entropy across MSA columns. Compared to the Iterative Unmask method (Sgarbossa et al., 2022), our technique, as illustrated in fig. 6b (left), consistently yields higher Entropy, suggesting increased diversity. For conservation, we examined the Position-Specific Scoring Matrix (PSSM) (Altschul et al., 1997) of both original and MSA-Generator-generated MSAs. The PSSM gauges conservation of specific amino acids. The **Pearson correlation coefficient** between the searched real and generated virtual PSSMs, shown in fig. 6b (right), highlights the retention of amino acid conservation in generative virtual MSAs (see fig. 7 for visualization of conversation). The outcomes highlight MSA-Generator's effectiveness as an MSA generator and demonstrate the broader capabilities of the seqs2seqs framework.

4.6 Ablation Study

Following section 4.3 setup, we investigated the impact of Ensemble Runs and Sequence Augmentation Factor on PSP outcomes, as shown in fig. 6a. While additional ensemble runs lead to improved pLDDT scores, gains in LDDT plateau after three runs, even with higher computational expenses. Hence, we chose 3 runs to optimize performance and efficiency. Furthermore, a higher augmentation factor does not always yield better results; the identical input MSA might lack new insights, and extra sequences risk introducing noise.

5 Conclusion

We present an unsupervised seqs2seqs task, accompanied by an automated dataset construction pipeline, designed to pre-train MSA-Generator for simultaneous multi-sequence generation. Rigorous experimentation underscore the effectiveness, diversity, and conservation feature of generated virtual MSAs, amplifying the prowess of stalwarts like AlphaFold2 in scenarios where conventional methods come up short. Furthermore, our approach demonstrates generalization across a wide array of protein sequence families in a zero-shot fashion. Our findings highlight the immense promise of the unsupervised seqs2seqs task, pointing towards its prospective utility in a broader spectrum of bio-sequences, thereby amplifying its benefits.

Limitation

While MSA-Generator demonstrates promising improvements in protein structure prediction, there are several limitations to consider.

First, we did not conduct scale-up experiments to explore the effects of increasing model size and expanding the pre-training dataset. It is possible that larger models and more comprehensive training data could further enhance the generative capabilities for sequence prediction, potentially improving structure prediction accuracy. Second, our current approach relies on selecting the maximum pLDDT score from multiple runs, which may not be the most optimal method and introduces inefficiency due to the need for repeated predictions. Developing a more refined metric to assess MSA quality or better predict the reliability of the final structure could significantly improve efficiency and consistency in future work. Lastly, although our model effectively handles proteins with limited homologous sequences, the generated MSAs typically do not achieve substantial depth. Increasing the depth of these generated MSAs would require higher computational resources, as the complexity scales with MSA depth. Addressing this challenge will be crucial for extending the applicability of our approach to a broader range of protein sequences.

Acknowledgments

The technical support extended by the Mila IDT team in managing the computational infrastructure is greatly appreciated. This project was partially supported by Shanghai Artificial Intelligence Laboratory. S.S. is supported by funds from Netmind.AI and Protagolabs Inc.

References

Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R. Glassman, Andy DeGiovanni, Jose H. Pereira, Andria V. Rodrigues, Alberdina A. van Dijk, Ana C. Ebrecht, Diederik J. Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Udit Dalwadi, Calvin K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021. doi: 10.1126/science.abj8754. URL `https://www.science.org/doi/abs/10.1126/science.abj8754`.

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Bo Chen, Ziwei Xie, Jiezhong Qiu, Zhaofeng Ye, Jinbo Xu, and Jie Tang. Improved the protein complex prediction with protein language models. *bioRxiv*, pp. 2022–09, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Charlotte Rochereau, George M Church, Peter Karl Sorger, and Mohammed N AlQuraishi. Single-sequence protein structure prediction using language models from deep learning. *bioRxiv*, 2021.

Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Christina Floristean, Anant Kharkar, Koushik Roy, Charlotte Rochereau, Gustaf Ahdritz, Joanna Zhang, George M Church, et al. Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology*, 40 (11):1617–1623, 2022.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

M Dayhoff, R Schwartz, and B Orcutt. 22 a model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5:345–352, 1978.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing, 2021.

Matteo Figliuzzi, Pierre Barrat-Charlaix, and Martin Weigt. How pairwise coevolutionary models capture the collective residue variability in proteins? *Molecular biology and evolution*, 35(4): 1018–1027, 2018.

Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2019.

L Steven Johnson, Sean R Eddy, and Elon Portugaly. Hidden markov model speed heuristic and iterative hmm search procedure. *BMC bioinformatics*, 11(1):1–8, 2010.

David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202, 1999.

David T Jones, William R Taylor, and Janet M Thornton. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, 8(3):275–282, 1992.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (casp)—round xiii. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1011–1020, 2019.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, 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.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

Francisco McGee, Sandro Hauri, Quentin Novinger, Slobodan Vucetic, Ronald M Levy, Vincenzo Carnevale, and Allan Haldane. The generative capacity of probabilistic protein sequence models. *Nature communications*, 12(1):6302, 2021.

Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682, 2022.

Erik Nijkamp, Jeffrey Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: Exploring the boundaries of protein language models." arxiv, 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, 2020.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8844–8856. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/rao21a.html`.

Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

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.

Burkhard Rost and Chris Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of molecular biology*, 232(2):584–599, 1993.

William P Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, et al. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, 2020.

Damiano Sgarbossa, Umberto Lupo, and Anne-Florence Bitbol. Generative power of a protein language model trained on multiple sequence alignments. *bioRxiv*, 2022.

Sam Sinai, Eric Kelsic, George M Church, and Martin A Nowak. Variational auto-encoding of protein sequences. *arXiv preprint arXiv:1712.03346*, 2017.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Baris E Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. Bertology meets biology: interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.

Qin Wang, Jiayang Chen, Yuzhe Zhou, Yu Li, Liangzhen Zheng, Sheng Wang, Zhen Li, and Shuguang Cui. Contact-distil: Boosting low homologous protein contact map prediction by self-supervised distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4620–4627, 2022a.

Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Computational Biology*, 13(1):e1005324, jan 2017. doi: 10.1371/journal.pcbi.1005324. URL `https://doi.org/10.1371%2Fjournal.pcbi.1005324`.

Wenkai Wang, Zhenling Peng, and Jianyi Yang. Single-sequence protein structure prediction using supervised transformer protein language models. *Nature Computational Science*, 2(12):804–814, 2022b.

Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pp. 2022–07, 2022a.

Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, 2022b.

Chengxin Zhang, Wei Zheng, S. M. Mortuza, Yang Li, Yang Li, and Yang Zhang. Deepmsa: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics*, 2019. URL `https://api.semanticscholar.org/CorpusID:208142755`.

Jun Zhang, Sirui Liu, Mengyun Chen, Haotian Chu, Min Wang, Zidong Wang, Jialiang Yu, Ningxi Ni, Fan Yu, Diqing Chen, et al. Few-shot learning of accurate folding landscape for protein structure prediction. *arXiv preprint arXiv:2208.09652*, 2022.

A Pretraining Details

We have compiled a pretraining dataset containing 2M MSAs from four databases: Uniref90 v.2022_04 using pipeline discussed in section 3.1. Specifically, for each MSA we randomly select 10-30 sequences and query as source X and another 10-30 sequences as target Y .

B MSA Visualization

Our goal is to explore the variations in MSA sequences using MSA-Generator. Accordingly, we depict the MSA's colored distribution in Fig 7 using Jalview[3]. Observing the columnar distribution, it's evident that the produced MSA bears resemblances to the original sequences but introduces unique variations that encapsulate the external insights derived from MSA-Generator. This show the conservative feature reserved by the MSA and enhanced diversity as well.

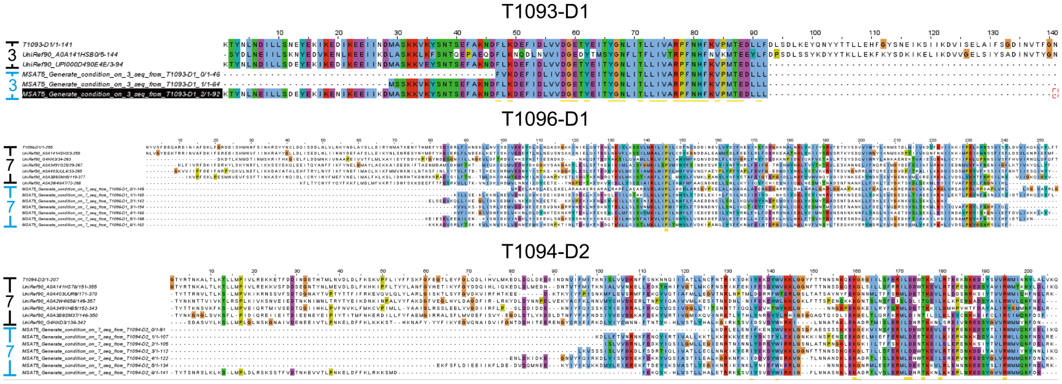


Figure 7: colored-distribution MSA, different colors represent different amino acids in protein sequence, from top to bottom is T1060s2-D1, T1093-D1, T1096-D1. since MSA-Generator augment one times more sequences, the top half of each diagram represents original MSA, and bottom half represent generated MSA

C Search Parameter

We use JackHMMER to build pretraining MSA dataset. We adopt default parameters, including: -E = 0.001, -N=5, -Z=1000, –incE=0.01.

D Relation between MSA depth, quality, and downstream performance

The relationship between Multiple Sequence Alignment (MSA) depth, its quality, and the consequent impact on downstream performance is a critical factor in our analysis. For an in-depth discussion on how MSA depth influences quality and downstream tasks, we refer the reader to Figure 5 (a) in the AlphaFold2 paper by Jumper et al. Jumper et al. (2021), where the authors illustrate that a reduction in MSA depth can lead to decreased quality and hinder performance in downstream applications.

In support of this, our Figure 4, which presents a violin plot, shows that a baseline MSA depth of 5 tends to underperform when compared to real and virtual depths of 15. This observation is particularly evident from the distribution of scores, where the baseline depth of 5, represented in blue, is more concentrated towards lower scores, unlike the distributions for real and virtual depths of 15, depicted in yellow and red respectively.

Additionally, an intuitive approach would be introduction of random protein sequences to the MSA, our implementation of 'Iterative Unmasking' baseline already employed a similar concept which

[3] This figure is intended solely for the visualization of the generated Multiple Sequence Alignments, and does not contribute to the analytical evaluations of our study.

Method	CSAP14-plddt	CSAP14-lddt	CSAP15-plddt	CSAP15-lddt
AlphaFold2	65.1	53.2	65.1	55.6
AlphaFold2+5% random	64.3	50.6	62.3	50.9
AlphaFold2+10% random	60.5	49.5	59.6	43.2
AlphaFold2+15% random	55.6	44.3	53.8	39.6
AlphaFold2+Ours	71.6	57.5	73.7	63.7

Table 3: Comparison of different methods on CSAP14 and CSAP15 datasets.

randomly unmasked tokens using the MSA Transformer. Despite this strategy, as shown in our Table 1, the improvement in results was not significant. To address further concerns, we explored the introduction of random sequences by substituting 5%, 10%, and 15% of the tokens within the MSA, thereby artificially increasing its depth as documented in Table 3. This modification, however, led to a decline in downstream performance, thereby affirming the effectiveness of our original method.

E Detailed Results

Real-Word Difficult and Challenging Results Detailed results for Section 4.3 and 4.4 are presented in Table 4.

ID	Depth	pLDDT				LDDT			
		Org	Run1	Run2	Run3	Org	Run1	Run2	Run3
T1037-D1	4	36.04	40.68	36.04	36.04	24.09	26.08	24.09	24.09
T1042-D1	2	41.99	42.25	45.45	47.86	32.42	32.19	32.43	32.84
T1064-D1	9	60.34	66.93	63.1	65	31.25	37.48	31.34	51.36
T1074-D1	9	85.96	85.96	85.96	85.96	81.14	81.14	81.14	81.14
T1082-D1	10	92.31	92.79	92.31	92.31	87.37	88.49	87.37	87.37
T1093-D1	3	62.1	81.26	62.1	62.1	45.5	70.77	45.5	45.5
T1094-D2	7	90.54	90.54	93.34	91.47	77.12	77.12	77.02	76.47
T1096-D1	7	70.28	73.06	86.25	83.28	61.92	62.83	71.19	69.23
T1096-D2	2	45.55	50.01	54.55	50.24	34.07	36.56	36.98	34.66
T1099-D1	8	89.27	89.99	89.27	90.38	75.12	74.27	75.12	74.62
T1100-D2	2	42.16	47.27	42.16	42.16	34.93	35.87	34.93	34.93
T1113	10	43.5	82.4	76.4	77.0	32.6	80.6	78.9	78.0
T1119	2	93.4	94.5	92.4	93.6	91.1	90.8	90.4	90.0
1122	1	65.9	80.2	78.6	79.8	43.3	73.4	70.2	83.2
1125	15	49.6	54.4	54.8	53.2	38.2	45.0	45.0	43.2
1130	1	50.6	60.2	60.0	58.9	39.4	46.0	46.9	45.8
1131	1	40.6	46.0	45.4	46.2	32.6	48.6	38.2	38.6
1178	15	79.2	80.6	81.4	80.3	58.2	60.9	45.9	58.9
1194	14	93.4	94.1	93.4	93.4	90.2	91.7	90.5	91.7

Table 4: pLDDT (left) and LDDT (right) improvement over difficult MSA (depth$\leq$20) of 3 runs.

Artificial Extreme Challenging Results Results for section 4.2 are shown in table 5 and table 6.

https://doi.org/10.52202/079017-1827

MSA-ID	Gold	Original	Aug1	Aug2	Aug3	ENs	over original	over gold
T1024-D1	87.10	81.56	77.25	81.62	81.62	81.62	0.06	-5.48
T1024-D2	84.44	86.66	79.67	82.84	82.84	82.84	-3.82	-1.60
T1025-D1	83.81	82.04	78.92	76.61	76.61	78.92	-3.12	-4.89
T1026-D1	82.27	48.72	22.59	79.82	79.82	79.82	31.10	-2.45
T1027-D1	48.81	39.15	37.31	46.38	46.38	46.38	7.23	-2.43
T1028-D1	75.57	32.35	52.62	52.09	52.09	52.62	20.27	-22.95
T1029-D1	47.62	47.43	47.76	47.44	47.44	47.76	0.33	0.14
T1030-D1	86.85	64.92	86.68	86.56	86.56	86.68	21.76	-0.17
T1030-D2	82.25	80.99	81.76	82.06	82.06	82.06	1.07	-0.19
T1031-D1	65.65	43.58	30.75	37.99	37.99	37.99	-5.59	-27.66
T1032-D1	22.48	19.90	68.50	62.11	62.11	68.50	48.60	46.02
T1034-D1	86.75	83.15	83.65	84.05	84.05	84.05	0.90	-2.70
T1035-D1	32.52	33.78	34.56	36.32	36.32	36.32	2.54	3.80
T1036s1-D1	67.02	26.11	79.99	79.13	79.13	79.99	53.88	12.97
T1038-D1	55.80	26.03	38.04	27.61	27.61	38.04	12.01	-17.76
T1038-D2	80.60	76.37	61.02	58.57	53.99	61.02	-15.35	-19.58
T1041-D1	62.13	45.60	32.14	0.00	34.74	34.74	-10.86	-27.39
T1045s1-D1	56.74	33.67	89.22	83.99	88.47	89.22	55.55	32.48
T1045s2-D1	27.65	24.44	22.81	24.89	21.61	24.89	0.45	-2.76
T1046s1-D1	87.67	72.65	87.33	0.00	87.98	87.98	15.33	0.31
T1046s2-D1	63.88	25.42	51.58	52.44	36.05	52.44	27.02	-11.44
T1047s1-D1	58.76	60.01	59.75	61.65	61.23	61.65	1.64	2.89
T1047s2-D1	40.65	39.12	69.07	70.64	71.69	71.69	32.57	31.04
T1047s2-D2	58.59	59.58	64.60	64.30	67.72	67.72	8.14	9.13
T1047s2-D3	58.12	51.61	57.63	58.18	57.53	58.18	6.57	0.06
T1048-D1	70.60	72.34	71.81	71.75	71.82	71.82	-0.52	1.22
T1049-D1	72.85	33.01	80.28	81.03	72.12	81.03	48.02	8.18
T1050-D1	89.66	88.46	73.27	72.51	72.97	73.27	-15.19	-16.39
T1050-D2	81.05	76.84	76.81	78.93	0.00	78.93	2.09	-2.12
T1050-D3	71.11	70.25	48.28	51.18	49.74	51.18	-19.07	-19.93
T1052-D1	77.26	68.84	82.18	83.82	83.08	83.82	14.98	6.56
T1052-D2	48.22	34.95	57.37	59.27	30.91	59.27	24.32	11.05
T1052-D3	90.81	82.12	89.62	85.22	0.00	89.62	7.50	-1.19
T1053-D1	72.86	30.58	73.75	74.09	65.32	74.09	43.51	1.23
T1053-D2	75.75	75.61	77.31	76.25	0.00	77.31	1.70	1.56
T1054-D1	34.64	31.68	70.68	81.44	0.00	81.44	49.76	46.80
T1055-D1	67.90	55.88	19.03	25.46	21.62	25.46	-30.42	-42.44

Table 5: *artificial challenging MSAs* results (1/2).

MSA-ID	Gold	Original	Aug1	Aug2	Aug3	ENs	over original	over gold
T1056-D1	75.61	76.11	64.62	65.39	62.93	65.39	-10.72	-10.22
T1057-D1	86.18	52.96	81.38	80.30	81.04	81.38	28.42	-4.80
T1058-D1	45.72	38.10	37.47	41.31	39.76	41.31	3.21	-4.41
T1058-D2	51.18	29.59	61.13	60.41	60.73	61.13	31.54	9.95
T1060s2-D1	75.25	38.44	72.84	74.01	0.00	74.01	35.57	-1.24
T1060s3-D1	80.69	69.93	78.72	78.31	78.89	78.89	8.96	-1.80
T1061-D0	60.24	57.51	54.74	0.00	58.36	58.36	0.85	0.85
T1061-D1	37.50	24.28	51.45	46.94	50.60	51.45	27.17	13.95
T1061-D2	53.34	31.82	58.05	50.32	55.92	58.05	26.23	4.71
T1061-D3	81.92	42.92	77.44	77.34	76.29	77.44	34.52	-4.48
T1062-D1	59.10	72.66	72.61	59.58	0.00	72.61	-0.05	13.51
T1065s1-D1	88.59	44.32	90.31	91.00	0.00	91.00	46.68	2.41
T1065s2-D1	70.46	63.90	80.77	87.09	83.49	87.09	23.19	16.63
T1067-D1	78.55	27.33	79.05	75.29	77.91	79.05	51.72	0.50
T1068-D1	85.05	35.20	89.55	88.93	89.92	89.92	54.72	4.87
T1070-D1	53.44	49.68	55.69	52.55	52.73	55.69	6.01	2.25
T1070-D2	29.16	35.04	87.50	86.84	90.38	90.38	55.34	61.22
T1070-D3	74.12	73.32	73.56	73.50	0.00	73.56	0.24	-0.56
T1070-D4	73.83	30.62	74.39	84.71	85.07	85.07	54.45	11.24
T1073-D1	76.17	76.19	76.79	0.00	76.11	76.79	0.60	0.62
T1076-D1	83.10	65.52	71.31	68.61	68.35	71.31	5.79	-11.79
T1078-D1	77.22	54.56	0.00	64.61	72.25	72.25	17.69	-4.97
T1079-D1	84.51	63.88	79.44	69.34	68.08	79.44	15.56	-5.07
T1080-D1	63.68	59.76	67.91	67.67	68.08	68.08	8.32	4.40
T1083-D1	78.49	78.26	78.02	77.48	77.72	78.02	-0.24	-0.47
T1084-D1	86.64	86.47	49.09	86.03	86.25	86.25	-0.22	-0.39
T1087-D1	80.76	77.32	37.14	39.73	69.27	69.27	-8.05	-11.49
T1088-D1	26.48	24.00	33.92	34.10	54.40	54.40	30.40	27.92
T1089-D1	83.99	77.55	64.70	64.30	67.15	67.15	-10.40	-16.84
T1090-D1	76.73	55.34	57.65	0.00	56.06	57.65	2.31	-19.08
T1091-D1	43.89	43.26	70.35	71.45	76.76	76.76	33.50	32.87
T1091-D2	81.76	40.56	50.57	0.00	47.30	50.57	10.01	-31.19
T1091-D3	71.58	52.00	75.93	0.00	75.98	75.98	23.98	4.40
T1091-D4	79.76	56.64	83.20	0.00	78.01	83.20	26.56	3.44
T1092-D1	63.77	56.56	33.97	38.95	40.48	40.48	-16.08	-23.29
T1092-D2	70.61	53.20	43.76	26.54	43.80	43.80	-9.40	-26.81
T1093-D2	29.91	34.69	28.20	28.12	31.08	31.08	-3.61	1.17
T1094-D1	28.74	26.73	21.89	21.63	20.72	21.89	-4.84	-6.85
T1095-D1	60.06	55.57	35.95	36.50	32.78	36.50	-19.07	-23.56
T1098-D1	57.45	29.82	49.36	45.94	36.76	49.36	19.54	-8.09
T1098-D2	37.88	31.58	45.70	45.89	45.36	45.89	14.31	8.01
T1100-D1	57.78	65.70	60.39	60.01	61.10	61.10	-4.60	3.32
T1101-D1	88.14	87.57	87.95	87.64	86.52	87.95	0.38	-0.19
T1101-D2	78.74	77.81	43.27	42.09	74.91	74.91	-2.90	-3.83
Average	66.47	53.45	61.77	57.14	56.43	66.32	12.87	-0.11

Table 6: *artificial challenging MSAs* results (2/2).

NeurIPS Paper Checklist

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

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

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

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

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

IMPORTANT, please:

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

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, the abstract is summary of the content of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the last paragraph of Conclusion section.

https://doi.org/10.52202/079017-1827

Guidelines:

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

3. **Theory Assumptions and Proofs**

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

Answer: [NA]

Justification: No theoretical results.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

Answer: [Yes]

Justification: We provide details of training and inference in section 3 and 4.

Guidelines:

- The answer NA means that the paper does not include experiments.

https://doi.org/10.52202/079017-1827

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

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

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

   Answer: [Yes]

   Justification: We released the code.

   Guidelines:

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

https://doi.org/10.52202/079017-1827

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: yes, provided in section 4.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: We didn't provide statical results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer ”Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [No]

   Justification: We perform the task with one gpu, and we provide training details for pretraining.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

https://doi.org/10.52202/079017-1827

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: Yes, we do.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: No relevant.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: No relevant.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

https://doi.org/10.52202/079017-1827

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We follow protocol.

    Guidelines:

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

13. **New Assets**

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

    Answer: [Yes]

    Justification: We provide codebase.

    Guidelines:

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

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

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

    Answer: [No]

    Justification: Not relevant.

https://doi.org/10.52202/079017-1827

Guidelines:

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

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

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

    Answer: [NA]

    Justification: Not relevant.

    Guidelines:

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

https://doi.org/10.52202/079017-1827