eLife

Reviewed Preprint
v2 • October 18, 2024
Revised by authors

Reviewed Preprint
v1 • June 25, 2024

Computational and Systems Biology

Semantical and Geometrical Protein Encoding Toward Enhanced Bioactivity and Thermostability

Yang Tan, Bingxin Zhou, Lirong Zheng, Guisheng Fan, Liang Hong

Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai, China • School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China • Shanghai Artificial Intelligence Laboratory, Shanghai, China • Shanghai National Center for Applied Mathematics (SJTU center), Shanghai Jiao Tong University, Shanghai, China • Zhangjiang Institute for Advanced Study, Shanghai Jiao Tong University, Shanghai, China

https://en.wikipedia.org/wiki/Open_access

Copyright information

Abstract

Protein engineering is a pivotal aspect of synthetic biology, involving the modification of amino acids within existing protein sequences to achieve novel or enhanced functionalities and physical properties. Accurate prediction of protein variant effects requires a thorough understanding of protein sequence, structure, and function. Deep learning methods have demonstrated remarkable performance in guiding protein modification for improved functionality. However, existing approaches predominantly rely on protein sequences, which face challenges in efficiently encoding the geometric aspects of amino acids' local environment and often fall short in capturing crucial details related to protein folding stability, internal molecular interactions, and bio-functions. Furthermore, there lacks a fundamental evaluation for developed methods in predicting protein thermostability, although it is a key physical property that is frequently investigated in practice. To address these challenges, this paper introduces a novel pre-training framework that integrates sequential and geometric encoders for protein primary and tertiary structures. This framework guides mutation directions toward desired traits by simulating natural selection on wild-type proteins and evaluates variant effects based on their fitness to perform specific functions. We assess the proposed approach using three benchmarks comprising over 300 deep mutational scanning assays. The prediction results showcase exceptional performance across extensive experiments when compared to other zero-shot learning methods, all while maintaining a minimal cost in terms of trainable parameters. This study not only proposes an effective framework for more accurate and comprehensive predictions to facilitate efficient protein engineering, but also enhances the *in silico* assessment system for future deep learning models to better align with empirical requirements. The PyTorch implementation are available at https://github.com/tyang816/ProtSSN.

eLife Assessment

ProtSSN is a **valuable** approach that generates protein embeddings by integrating sequence and structural information, demonstrating improved prediction of mutation effects on thermostability compared to competing models. The evidence supporting the authors' claims is **solid**, with well-executed comparisons. This work will be of particular interest to researchers in bioinformatics and structural biology, especially those focused on protein function and stability.

https://doi.org/10.7554/eLife.98033.2.sa3

Introduction

The diverse roles proteins play in nature necessitate their involvement in varied bio-functions, such as catalysis, binding, scaffolding, and transport (Frauenfelder and McMahon, 1998). Advances in science and technology have positioned proteins, particularly those with desired catalytic and binding properties, to pivotal positions in medicine, biology, and scientific research. While wild-type proteins exhibit optimal bio-functionality in their native environments, industrial applications often demand adaptation to conditions such as high temperature, high pressure, strong acidity, and strong alkalinity. The constrained efficacy of proteins in meeting the stringent requirements of industrial functioning environments hinders their widespread applications (Woodley, 2013; Ismail et al., 2021; Zheng et al., 2022; Jiang et al., 2023). Consequently, there arises a need to engineer these natural proteins to enhance their functionalities, aligning them with the demands of both industrial and scientific applications.

Analyzing the relationship between protein sequence and function yields valuable insights for engineering proteins with new or enhanced functions in synthetic biology. The intrinsic goal of protein engineering is to unveil highly functional sequences by navigating the intricate, high-dimensional surface of the fitness landscape (Yang et al., 2019), which delineates the relationship between amino acid (AA) sequences and the desired functional state. Predicting the effects of AA substitutions, insertions, and deletions (Riesselman et al., 2018; Gray et al., 2018; Notin et al., 2022a) in proteins, *i.e.*, mutation, thus allows researchers to dissect how changes in the AA sequence can impact the protein's catalytic efficiency, stability, and binding affinity (Shin et al., 2021). However, the extensive, non-contiguous search space of AA combinations poses challenges for conventional methods, such as rational design (Aprile et al., 2020) or directed evolution (Wang et al., 2021), in efficiently and effectively identifying protein variants with desired properties. In response, deep learning has emerged as a promising solution that proposes favorable protein variants with high fitness.

Deep learning approaches have been instrumental in advancing scientific insights into proteins, predominantly categorized into sequence-based and structure-based methods. Autoregressive protein language models (Notin et al., 2022a; Madani et al., 2023) interpret AA sequences as raw text to generate with self-attention mechanisms (Vaswani et al., 2017). Alternatively, masked language modeling objectives develop attention patterns that correspond to the residue-residue contact map of the protein (Meier et al., 2021; Rao et al., 2021a; Rives et al., 2021; Vig et al., 2021; Brandes et al., 2022; Lin et al., 2023). Other methods start from a multiple sequence alignment, summarizing the evolutionary patterns in target proteins (Riesselman et al., 2018; Frazer et al., 2021; Rao et al., 2021b). These methods result in a strong capacity for discovering the hidden protein space. However, the many-to-one relationship of both sequence-to-structure and structureto-function projections requires excessive training input or substantial learning

resources, which raises concerns regarding the efficiency of these pathways when navigating the complexities of the vast sequence space associated with a target function. Moreover, the overlooked local environment of proteins hinders the model's ability to capture structure-sensitive properties that impact protein's thermostability, interaction with substrates, and catalytic process (Zhao et al., 2022; Koehler Leman et al., 2023). Alternatively, structure-based modeling methods serve as an effective enhancement to complement sequence-oriented inferences for proteins with their local environment (Torng and Altman, 2019; Jones et al., 2021; Zhou et al., 2023a; Yi et al., 2023). Given that core mutations often induce functional defects through subtle disruptions to structure or dynamics (Roscoe et al., 2013), incorporating protein geometry into the learning process can offer valuable insights into stabilizing protein functioning. Recent efforts have been made to encode geometric information of proteins for topology-sensitive tasks such as molecule binding (Jin et al., 2021; Myung et al., 2022; Kong et al., 2023) and protein properties prediction (Zhang et al., 2022). Nevertheless, structure encoders fall short in capturing non-local connections for AAs beyond their contact region and overlook correlations that do not conform to the 'structure-function determination' heuristic.

There is a pressing need to develop a novel framework that overcomes limitations inherent in individual implementations of sequence or structure-based investigations. To this end, we introduce ProtSSN to assimilate the semantics and topology of **Prot**eins from their **S**equence and **S**tructure with deep neural **N**etworks. ProtSSN incorporates the intermediate state of protein structures and facilitates the discovery of an efficient and effective trajectory for mapping protein sequences to functionalities. The developed model extends the generalization and robustness of self-supervised protein language models while maintaining low computational costs, thereby facilitating self-supervised training and task-specific customization. A funnel-shaped learning pipeline, as depicted in **Fig. 1**, is designed due to the limited availability of protein crystal structures compared to observed protein sequences. Initially, the linguistic embedding establishes the semantic and grammatical rules in AA chains by inspecting millions of protein sequences. The topological embedding then enhances the interactions among locally connected AAs. Since a geometry can be placed or observed by different angles and positions in space, we represent proteins' topology by graphs and enhance the model's robustness and efficiency with a rotation and translation equivariant graph representation learning scheme.

The pre-trained ProtSSN demonstrates its feasibility across a broad range of mutation effect prediction benchmarks covering catalysis, interaction, and thermostability. Mutation effects prediction serves as a common *in silico* assessment for evaluating the capability of an established deep learning framework in identifying favorable protein variants. With the continuous evolution of deep mutation scanning techniques and other high-throughput technologies, benchmark datasets have been curated to document fitness changes in mutants across diverse proteins with varying degrees of combinatorial and random modifications (Moal and Fernández-Recio, 2012; Nikam et al., 2021; Velecky et al., 2022). **ProteinGym v1** (Notin et al., 2023) comprehends fitness scoring of mutants to catalytic activity and binding affinity across over 200 well-studied proteins of varying deep mutational scanning (DMS) assays and taxa. Each protein includes tens of thousands of mutants documented from high-throughput screening techniques. While **ProteinGym v1** initiates the most comprehensive benchmark dataset for mutants toward different properties enhancement, these fitness scores are normalized and simplified into several classes without further specifications. For instance, a good portion of protein assays is scored by their stability, which, in practice, is further categorized into thermostability, photostability, pH-stability, *etc*. Moreover, these stability properties should be discussed under certain environmental conditions, such as pH and temperature. Unfortunately, these detailed attributes are excluded in **ProteinGym v1**. Since a trade-off relationship emerges between activity and thermostability (Zheng et al., 2022), protein engineering in practice extends beyond enhancing catalytic activities to maintaining or increasing thermostability for prolonged functional lifetimes and applications under elevated temperatures and chemical conditions (Liu et al., 2019). Consequently, there is a need to enrich mutation effect prediction benchmarks that evaluate the

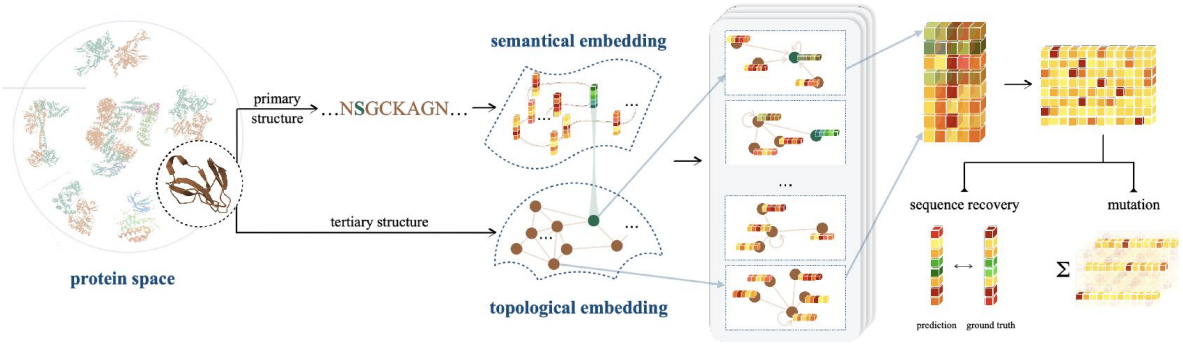


Figure 1.

An illustration of ProtSSN that extracts the semantical and geometrical characteristics of a protein from its sequentially-ordered global construction and spatially-gathered local contacts with protein language models and equivariant graph neural networks. The encoded hidden representation can be used for downstream tasks such as variants effect prediction that recognizes the impact of mutating a few sites of a protein on its functionality.

model's efficacy in capturing variant effects concerning thermostability under distinct experimental conditions. In this study, we address the mutation effect prediction tasks on thermostability with **DTm** and **DDG**, two new single-site mutation benchmarks that measure thermostability using ΔTm and ΔΔG values, respectively. Both benchmarks group experimental assays based on the protein-condition combinations. These two datasets supplement the publicly available DMS assay benchmarks and facilitate ready-to-use assessment for future deep learning methods toward protein thermostability enhancement.

This work fulfills practical necessities in the development of deep learning methods for protein engineering from three distinct perspectives. Firstly, the developed ProtSSN employs selfsupervised learning during training to obviate the necessity for additional supervision in downstream tasks. This zero-shot scenario is desirable due to the scarcity of experimental results, as well as the 'cold-start' situation common in many wet lab experiments. Secondly, the trained model furnishes robust and meaningful approximations to the joint distribution of the entire amino acid chain, thereby augmenting the *epistatic effect* (Sarkisyan et al., 2016; Khersonsky et al., 2018) in deep mutations. This augmentation stems from the model's consideration of the nonlinear combinatorial effects of amino acid sites. Additionally, we complement the existing dataset of DMS assay with additional benchmark assays related to protein-environment interactions, specifically focusing on thermostability–an integral objective in numerous protein engineering projects.

Materials and methods

This section starts by introducing the three mutational scanning benchmark datasets used in this study, including an open benchmark dataset **ProteinGym v1** and two new datasets on protein thermostability, *i.e.*, **DTm** and **DDG**. We establish the proposed ProtSSN for protein sequences and structures encoding in Section Proposed Method. The overall pipeline of model training and inference is presented in Section Model Pipeline. The experimental setup for model evaluation is provided in Section Experimental Protocol.

Dataset

To evaluate model performance in predicting mutation effects, we compare ProtSSN with SOTA baselines on **ProteinGym-v1**, the largest open benchmark for DMS assays. We also introduce two novel benchmarks, **DTm** and **DDG**, for assessing protein thermostability under consistent control environments. Details of the new datasets are provided in **Table 1**. Considering the significant influence of experimental conditions on protein temperature, we explicitly note the pH environment in each assay. Refer to the following paragraphs for more details.

ProteinGym

The assessment of ProtSSN for deep mutation effects prediction is conducted on **ProteinGym v1** (Notin et al., 2023). It is the most extensive protein substitution benchmark comprising 217 assays and more than 2.5M mutants. These DMS assays cover a broad spectrum of functional properties (*e.g.*, ligand binding, aggregation, viral replication, and drug resistance) and span various protein families (*e.g.*, kinases, ion channel proteins, transcription factors, and tumor suppressors) across different taxa (*e.g.*, humans, other eukaryotes, prokaryotes, and viruses).

DTm

The assays in the first novel thermostability benchmark are sourced from single-site mutations in **ProThermDB** (Nikam et al., 2021). Each assay is named in the format '*UniProt_ID-pH*'. For instance, '*O00095-8.0*' signifies mutations conducted and evaluated under pH 8.0 for protein

pH Range	DTm			DDG		
	# Assays	avg. # mut	avg. AA	# Assays	avg. # mut	avg.AA
Acid (pH < 7)	29	29.1	272.8	21	22.6	125.3
Neutral (pH = 7)	14	37.4	221.2	10	23.1	78.3
Alkaline (pH > 7)	23	50.1	233.3	5	24.6	101.4
sum	66	38.2	221.2	36	23.0	108.8

Table 1.

Statistical summary of DTm and DDG.

O00095. The attributes include 'mutant', 'score', and 'UniProt_ID', with at least 10 mutants to ensure a meaningful evaluation. To concentrate on single-site mutations compared to wild-type proteins, we exclude records with continuous mutations. To avoid dealing with partial or misaligned protein structures resulting from incomplete wet experimental procedures, we employ UniProt ID to pair protein sequences with folded structures predicted by AlphaFold2 (Jumper et al., 2021) from *https://alphafold.ebi.ac.uk*. In total, **DTm** consists of 66 such protein-environment pairs and 2, 520 mutations.

DDG

The second novel thermostability benchmark is sourced from **DDMut** (Zhou et al., 2023b), where mutants are paired with their PDB documents. We thus employ crystal structures for proteins in **DDG** and conduct additional preprocessing steps for environmental consistency. In particular, we group mutants of the same protein under identical experimental conditions and examine the alignment between mutation sites and sequences extracted from crystal structures. We discard mutant records that cannot be aligned with the sequence loaded from the associated PDB document. After removing data points with fewer than 10 mutations, **DDG** comprises 36 data points and 829 mutations. As before, assays are named in the format '*PDB_ID-pH-temperature*' to indicate the chemical condition (pH) and temperature (in Celsius) of the experiment on the protein. An example assay content is provided in **Fig. 2**.

Proposed Method

Labeled data are usually scarce in biomolecule research, which demands designing a general selfsupervised model for predicting variant effects on unknown proteins and protein functions. We propose ProtSSN with a pre-training scheme that recovers residues from a given protein backbone structure with noisy local environment observations.

Multi-level Protein Representation

Protein Primary Structure (Noisy)

Denote a given observation with resiude $\tilde{\boldsymbol{v}}$. We assume this arbitrary observation is undergoing random mutation toward a stable state. The training objective is to learn a revised state $\boldsymbol{v}$ that is less likely to be eliminated by natural selection due to unfavorable properties such as instability or inability to fold. The perturbed observation is defined by a multinomial distribution.

$$\boldsymbol{\pi}(\tilde{\boldsymbol{v}} \mid \boldsymbol{v}) = p\boldsymbol{\Theta}(\pi_1, \pi_2, \ldots, \pi_{20}) + (1-p)\delta(\tilde{\boldsymbol{v}} - \boldsymbol{v}), \tag{1}$$

where an AA in a protein chain has a chance of p to mutate to one of 20 AAs (including itself) following a *replacement distribution* $\boldsymbol{\Theta}(\cdot)$ and $(1-p)$ of remaining unchanged. We consider p as a tunable parameter and define $\boldsymbol{\Theta}(\cdot)$ by the frequency of residues observed in wild-type proteins.

Protein Tertiary Structure

The geometry of a protein is described by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W}_V, \boldsymbol{W}_E, \boldsymbol{X}_V)$, which is a residue graph based on the k-nearest neighbor algorithm (kNN). Each node $v_i \in \mathcal{V}$ represents an AA in the protein connected to up to k closest nodes within 30Å. Node attributes $\boldsymbol{W}_V$ are hidden semantic embeddings of residues extracted by the semantic encoder (see Section Semantic Encoding of Global AA Contacts), and edge attributes $\boldsymbol{W}_E \in \mathbb{R}^{93}$ feature relationships of connected nodes based on inter-atomic distances, local N-C positions, and sequential position encoding (Zhou et al., 2024). Additionally, $\boldsymbol{X}_V$ records 3D coordinates of AAs in the Euclidean space, which plays a crucial role in the subsequent geometric embedding stage to preserve roto-translation equivariance.

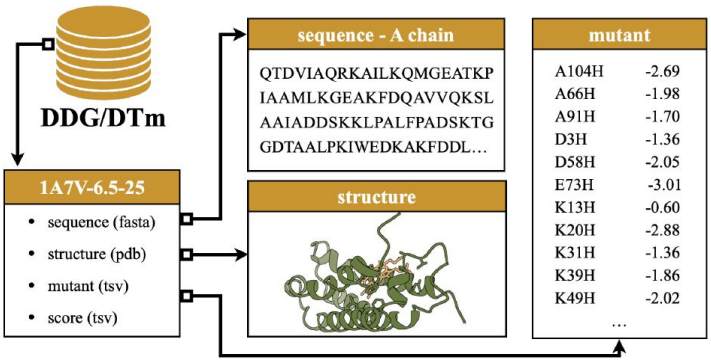


Figure 2.

An example source record of the mutation assay. The record is derived from DDG for the A chain of protein 1A7V, experimented at pH=6.5 and degree at 25°C.

Semantic Encoding of Global AA Contacts

Although it is generally believed by the community that a protein's sequence determines its biological function via the folded structure, following strictly to this singular pathway risks overlooking other unobserved yet potentially influential inter-atomic communications impacting protein fitness. Our proposed ProtSSN thus includes pairwise relationships for residues through an analysis of proteins' primary structure from $\tilde{\boldsymbol{V}}$ and embeds them into hidden representations $\boldsymbol{W}_V$ for residues. At each iteration, the input sequences are modified randomly by (3) and then embedded via an *Evolutionary Scale Modeling* method, ESM-2 (Lin et al., 2023), which employs a BERT-style masked language modeling objective. This objective predicts the identity of randomly selected AAs in a protein sequence by observing their context within the remainder of the sequence. Alternative semantic encoding strategies are discussed in Section AA Perturbation Strategies.

Geometrical Encoding of Local AA Contacts

Proteins are structured in 3D space, which requires the geometric encoder to possess roto-translation equivariance to node positions as well as permutation invariant to node attributes. This design is vital to avoid the implementation of costly data augmentation strategies. We practice *Equivariant Graph Neural Networks* (EGNN) (Satorras et al., 2021) to acquire the hidden node representation $\boldsymbol{W}_V^{l+1} = \left\{ \boldsymbol{w}_{v_1}^{l+1}, \ldots, \boldsymbol{w}_{v_n}^{l+1} \right\}$ and node coordinates $\boldsymbol{X}_{\text{pos}}^{l+1} = \left\{ \boldsymbol{x}_{v_1}^{l+1}, \ldots, \boldsymbol{x}_{v_n}^{l+1} \right\}$ at the $l+1$th layer

$$
\begin{aligned}
\boldsymbol{m}_{ij} &= \phi_e \left(\boldsymbol{w}_{v_i}^l, \boldsymbol{w}_{v_j}^l, \left\| \boldsymbol{x}_{v_i}^l - \boldsymbol{x}_{v_j}^l \right\|^2, \boldsymbol{w}_{e_{ij}} \right), \\
\boldsymbol{x}_{v_i}^{l+1} &= \mathbf{x}_{v_i}^l + \frac{1}{n} \sum_{j \neq i} \left(\mathbf{x}_{v_i}^l - \mathbf{x}_{v_j}^l \right) \phi_x \left(\mathbf{m}_{ij} \right), \\
\boldsymbol{w}_{v_i}^{l+1} &= \phi_v \left(\boldsymbol{w}_i^l, \sum_{j \neq i} \boldsymbol{m}_{ij} \right).
\end{aligned}
\tag{2}
$$

In these equations, $\boldsymbol{w}_{e_{ij}}$ represents the input edge attribute on $\mathcal{V}_{ij}$, which is not updated by the network. The propagation rules φ_e, φ_x and φ_v are defined by differentiable functions, *e.g.*multi-layer perceptrons (MLPs). The final hidden representation on nodes $\boldsymbol{w}_v^L$ embeds the microenvironment and local topology of AAs, and it will be carried on by readout layers for label predictions.

Model Pipeline

ProtSSN is designed for protein engineering with a self-supervised learning scheme. The model is capable of conducting zero-shot variant effect prediction on an unknown protein, and it can generate the joint distribution for the residue of all AAs along the protein sequence of interest. This process accounts for the epistatic effect and concurrently returns all AA sites in a sequence. Below, we detail the workflow for training the zero-shot model and scoring the effect of a specific mutant.

Training

The fundamental model architecture cascades a frozen sequence encoding module and a trainable tertiary structure encoder. The protein is represented by its sequence and structure characteristics, where the former is treated as plain text and the latter is formalized as a *k*NN graph with modelencoded node attributes, handcrafted edge attributes, and 3D positions of the

corresponding AAs. Each input sequence $\boldsymbol{V}$ are first one-hot encoded by 33 tokens, comprising both AA residues and special tokens Lin et al. (2023). The ground truth sequences, during training, will be permuted into

$$\tilde{\boldsymbol{V}} = \text{Permute}(\boldsymbol{V}). \tag{3}$$

Here $\tilde{v}$ is the perturbed AA to be encoded by a protein language model, which analyses pairwise hidden relationships among AAs from the input protein sequence and produces a vector representation $\boldsymbol{w}_{v_i} \in \boldsymbol{W}_V$ for each AA, *i.e.*,

$$\boldsymbol{W}_V = \text{LM}_{\text{frozen}}(\tilde{\boldsymbol{V}}) \tag{4}$$

The language model $\text{LM}_{\text{frozen}}(\cdot)$, ESM-2 (Lin et al., 2023) for instance, has been pre-trained on a massive protein sequence database (*e.g.*, **UniRef50** (Suzek et al., 2015)) to understand the semantic and grammatical rules of wild-type proteins with high-dimensional AA-level long-short-range interactions. The independent perturbation on AA residues follows (1) operates in every epoch, which requires constant updates on the sequence embedding. To further enhance the interaction of locally connected AAs in the node representation, a stack of L EGNN layers is implemented on the input graph $\mathcal{G}$ to yield

$$\boldsymbol{W}_V^L = \text{EGNN}(\mathcal{G}). \tag{5}$$

During the pre-training phase for protein sequence recovery, the output layer $\varphi(\cdot)$ provides the probability of tokens in one of 33 types, *i.e.*,

$$\boldsymbol{Y} = \phi(\boldsymbol{W}_V^L) \in \mathbb{R}^{n\times 33} \tag{6}$$

for a protein of n AAs. The model's learnable parameters are refined by minimizing the crossentropy of the recovered AAs with respect to the ground-truth AAs in wild-type proteins:

$$\text{loss} = -\sum_{n=1}^{L}\sum_{c=1}^{33} V_{n,c}\log\left(\text{softmax}(Y_{n,c})\right), \tag{7}$$

where L is the length of the sequence, n represents the position index within the sequence and each c corresponds to a potential type of token.

Inferencing

For a given wild-type protein and a set of mutants, the fitness scores of the mutants are derived from the joint distribution of the altered residues relative to the wild-type template. First, the wild-type protein sequence and structure are encoded into hidden representations following (4)(5). Unlike the pre-training process, here the residue in the wild-type protein is considered as a reference state and is compared with the predicted probability of AAs at the mutated site. Next, for a mutant with mutated sites $\mathcal{T}(\mathcal{T} \geq 1)$, we define its fitness score F_x using the corresponding *log-odds-ratio, i.e.*,

$$F_x = \sum_{t\in\mathcal{T}} \log p(\boldsymbol{y}_t) - \log p(\boldsymbol{v}_t), \tag{8}$$

where y_t and $\boldsymbol{v}_t$ denote the mutated and wild-type residue of the rth AA.

Experimental Protocol

We validate the efficacy of ProtSSN on zero-shot mutation effect prediction tasks. The performance is compared with other SOTA models of varying numbers of trainable parameters. The implementations are programmed with PyTorch-Geometric (ver 2.2.0) and PyTorch (ver 1.12.1) and executed on an NVIDIA® Tesla A100 GPU with 6, 912 CUDA cores and 80GB HBM2 installed on an HPC cluster.

Training Setup

ProtSSN is pre-trained on a non-redundant subset of **CATH v4.3.0** (Orengo et al., 1997) domains, which contains 30, 948 experimental protein structures with less than 40% sequence identity. We remove proteins that exceed 2, 000 AAs in length for efficiency. For each kNN protein graph, node features are extracted and updated by a frozen ESM2-t33 prefix model (Lin et al., 2023). Protein topology is inferred by a 6-layer EGNN (Satorras et al., 2021) with the hidden dimension tuned from {512, 768, 1280}. Adam (Kingma and Ba, 2015) is used for backpropagation with the learning rate set to 0.0001. To avoid training instability or CUDA out-of-memory, we limit the maximum input to 8, 192 AA tokens per batch, constituting approximately 32 graphs.

Baseline Methods

We undertake an extensive comparison with baseline methods of self-supervised sequence or structure-based models on the fitness of mutation effects prediction (**Table 2**). Sequence models employ position embedding strategies such as autoregression (RITA (Hesslow et al., 2022), Tranception (Notin et al., 2022a), and ProGen2 (Nijkamp et al., 2023)), masked language modeling (ESM-1b (Rives et al., 2021), ESM-1v (Meier et al., 2021), and ESM2 (Lin et al., 2023)), a combination of the both (ProtTrans (Elnaggar et al., 2021)), and convolutional modeling (CARP (Yang et al., 2022)). Additionally, we also compare with ESM-if1 (Hsu et al., 2022) which incorporates masked language modeling objectives with GVP (Jing et al., 2020), as well as MIF-ST (Yang et al., 2023) and SaProt Su et al. (2023) with both sequence and structure encoding.

The majority of the performance comparison was conducted on zero-shot deep learning methods. However, for completeness, we also report a comparison with popular MSA-based methods, such as GEMME (Laine et al., 2019), Site-Independent (Hopf et al., 2017), EVmutation (Hopf et al., 2017), Wavenet (Shin et al., 2021), DeepSequence (Riesselman et al., 2018), and EVE (Frazer et al., 2021). Other deep learning methods use MSA for training or retrieval, including MSA-Transformer (Rao et al., 2021b), Tranception (Notin et al., 2022a) and TranceptEVE (>Notin et al., 2022b).

Scoring Function

Following the convention, the fitness scores of the zero-shot models are calculated by the log-odds ratio in (8) for encoder methods, such as the proposed ProtSSN and the ProtTrans series models. For autoregressive and inverse folding models (*e.g.*, Tranception and ProGen2) that predict the next token x_i based on the context of previous $x_{1:i-1}$ tokens, the fitness score F_x of a mutated sequence $\boldsymbol{y}$ is computed via the log-likelihood ratio with the wild-type sequence $\boldsymbol{v}$, *i.e.*,

$$F_x = \log \frac{P(\boldsymbol{y})}{P(\boldsymbol{v})}. \tag{9}$$

Model	Type		Description	Source Code
	seq	struct		
RITA (*Hesslow et al., 2022*)	✓		a generative protein language model with billion-level parameters	github.com/lightonai/RITA
ProGen2 (*Nijkamp et al., 2023*)	✓		a generative protein language model with billion-level parameters	github.com/salesforce/progen
ProtTrans (*Elnaggar et al., 2021*)	✓		Transformer-based models trained on large protein sequence corpus	github.com/agemagician/ProtTrans
Tranception (*Notin et al., 2022a*)	✓		an autoregressive model for variant effect prediction with retrieve machine	github.com/OATML-Markslab/Tranception
CARP (*Yang et al., 2022*)	✓		pretrained CNN protein sequence masked language models of various sizes	github.com/microsoft/protein-sequence-models
ESM-1b (*Rives et al., 2021*) ESM-1v (*Meier et al., 2021*) ESM2 (*Lin et al., 2023*)	✓ ✓ ✓		a masked language model-based pre-train method with various pre-training dataset and positional embedding strategies	github.com/facebookresearch/esm
ESM-if1 (*Hsu et al., 2022*)		✓	an inverse folding method with mask language modeling and geometric vector perception (*Jing et al., 2020*).	
MIF-ST (*Yang et al., 2023*)	✓	✓	pretrained masked inverse folding models with sequence pretraining transfer	github.com/microsoft/protein-sequence-models
SaProt (*Su et al., 2023*)	✓	✓	structure-aware vocabulary for protein language modeling with FoldSeek	github.com/westlake-repl/SaProt

Table 2.

Details of Zero-shot Baseline Models.

Benchmark Datasets

We conduct a comprehensive comparison of deep learning predictions on hundreds of DMS assays concerning different protein types and biochemical assays. We prioritize experimentally measured properties that possess a monotonic relationship with protein fitness, such as catalytic activity, binding affinity, expression, and thermostability. In particular, **ProteinGym v1** (Notin et al., 2023) groups 5 sets of protein properties from 217 assays. Two new datasets named **DTm** and **DDG** examine 102 environment-specific assays with thermostability scores. In total, 319 assays with around 2.5 million mutants are scored. To better understand the novelty of the proposed ProtSSN, we designed additional investigations on **ProteinGym v0** (***Notin et al., 2022a***), an early-release version of **ProteinGym** with 1.5M missense variants across 86 assays (excluding one assay that failed to be folded by both AlphaFold2 and ESMFold, *i.e.*, A0A140D2T1_ZIKV_Sourisseau_growth_2019).

Evaluation Metric

We evaluate the performance of pre-trained models on a diverse set of proteins and protein functions using Spearman's ρ correlation that measures the strength and direction of the monotonic relationship between two ranked sequences, *i.e.*, experimentally evaluated mutants and modelinferred mutants (Meier et al., 2021; Notin et al., 2022a; Frazer et al., 2021; Zhou et al., 2024). This non-parametric rank measure is robust to outliers and asymmetry in mutation scores, and it does not assume any specific distribution of mutation fitness scores. The scale of ρ ranges from −1 to 1, indicating the negative or positive correlation of the predicted sequence to the ground truth. The prediction is preferred if its ρ to the experimentally examined ground truth is close to 1.

Results

Variant Effect Prediction

We commence our investigation by assessing the predictive performance of ProtSSN on four benchmark datasets using state-of-the-art (SOTA) protein learning models. We deploy different versions of ProtSSN that learns kNN graphs with $k \in \{10, 20, 30\}$ and $h \in \{512, 768, 1, 280\}$ hidden neurons in each of the 6 EGNN layers. For all baselines, their performance on **DTm** and **DDG** are reproduced with the official implementation listed in **Table 2**, and the scores on **ProteinGym v1** are retrieved from *https://proteingym.org/benchmarks* by Notin et al. (2023). As visualized in **Fig. 3** and reported in **Table 3**, ProtSSN demonstrated exceptional predictive performance with a significantly smaller number of trainable parameters when predicting the function and thermostability of mutants.

Compared to protein language models (**Fig. 3**, colored circles), ProtSSN benefits from abundant structure information to more accurately capture the overall protein characteristics, resulting in enhanced thermostability and thus achieving higher correlation scores on **DTm** and **DDG**. This is attributed to the compactness of the overall architecture as well as the presence of stable local structures such as alpha helices and beta sheets, both of which are crucial to protein thermostability by providing a framework that resists unfolding at elevated temperatures (Robinson-Rechavi et al., 2006). Consequently, the other two structures-involved models, *i.e.*, ESM-if1 and MIF-ST, also exhibit higher performance on the two thermostability benchmarks.

On the other hand, although protein language models can typically increase their scale, *i.e.*, the number of trainable parameters, to capture more information from sequences, they cannot fully replace the role of the structure-based analysis. This observation aligns with the mechanism of protein functionality, where local structures (such as binding pockets and catalytic pockets) and

Table 3.

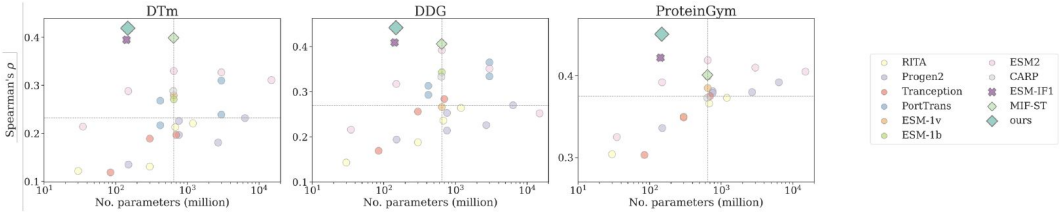
Spearman's ρ correlation of variant effect prediction by with zero-shot methods on DTm, DDG, and ProteinGym v1.

Model	Version	# Params (million)	DTm	DDG	ProteinGym v1					
					Activity	Binding	Expression	Organismal	Stability	Overall Fitness
Sequence Encoder										
RITA	small	30	0.122	0.143	0.294	0.275	0.337	0.327	0.289	0.304
	medium	300	0.131	0.188	0.352	0.274	0.406	0.371	0.348	0.350
	large	680	0.213	0.236	0.359	0.291	0.422	0.374	0.383	0.366
	xlarge	1,200	0.221	0.264	0.402	0.302	0.423	0.387	0.445	0.373
ProGen2	small	151	0.135	0.194	0.333	0.275	0.384	0.337	0.349	0.336
	medium	764	0.226	0.214	0.393	0.296	0.436	0.381	0.396	0.380
	base	764	0.197	0.253	0.396	0.294	0.444	0.379	0.383	0.379
	large	2,700	0.181	0.226	0.406	0.294	0.429	0.379	0.396	0.381
	xlarge	6,400	0.232	0.270	0.402	0.302	0.423	0.387	0.445	0.392
PortTrans	bert	420	0.268	0.313	-	-	-	-	-	-
	bert_bfd	420	0.217	0.293	-	-	-	-	-	-
	t5_xl_uniref50	3,000	0.310	0.365	-	-	-	-	-	-
	t5_xl_bfd	3,000	0.239	0.334	-	-	-	-	-	-
Tranception	small	85	0.119	0.169	0.287	0.349	0.319	0.270	0.258	0.288
	medium	300	0.189	0.256	0.349	0.285	0.409	0.362	0.342	0.349
	large	700	0.197	0.284	0.401	0.289	0.415	**0.389**	0.381	0.375
ESM-1v	-	650	0.279	0.266	0.390	0.268	0.431	0.362	0.476	0.385
ESM-1b	-	650	0.271	0.343	0.428	0.289	0.427	0.351	0.500	0.399
ESM-2	t12	35	0.214	0.216	0.314	0.292	0.364	0.218	0.439	0.325
	t30	150	0.288	0.317	0.391	0.328	0.425	0.305	0.510	0.392
	t33	650	0.330	0.392	0.391	0.328	0.425	0.305	0.510	0.419
	t36	3,000	0.327	0.351	0.417	0.322	0.425	0.379	0.509	0.410
	t48	15,000	0.311	0.252	0.405	0.318	0.425	0.388	0.488	0.405
CARP	-	640	0.288	0.333	0.395	0.274	0.419	0.364	0.414	0.373
Structure Encoder										
ESM-if1	-	142	0.395	**0.409**	0.368	**0.392**	0.403	0.324	**0.624**	0.422
Sequence + Structure Encoder										
MIF-ST	-	643	**0.400**	0.406	0.390	0.323	0.432	0.373	0.486	0.401
SaProt	masked	650	0.382	-	**0.459**	**0.382**	**0.485**	0.371	**0.583**	**0.456**
	unmasked	650	0.376	0.359	0.450	**0.376**	**0.460**	0.372	**0.577**	**0.447**
ProtSSN (ours)	k20_h512	148	**0.419**	**0.442**	**0.458**	0.371	0.436	**0.387**	0.566	0.444
	ensemble	1,467	**0.425**	**0.440**	**0.466**	0.371	**0.451**	**0.398**	0.568	**0.451**

† The top three are highlighted by **First**, **Second**, **Third**.

Figure 3.

Number of trainable parameters and Spearman's ρ correlation on **DTm, DDG**, and **ProteinGym v1**, with the medium value located by the dashed lines. Dot, cross, and diamond markers represent sequence-based, structure-based, and sequence-structure models, respectively.



the overall structure (such as conformational changes and allosteric effect) are both crucial for binding and catalysis (Sheng et al., 2014). Unlike the thermostability benchmarks, the discrepancies between structure-involved models and protein language models are mitigated in **ProteinGym v1** by increasing the model scale. In summary, structure-based and sequence-based methods are suitable for different types of assays, but a comprehensive framework (such as ProtSSN) demonstrates superior overall performance compared to large-scale language models. Moreover, ProtSSN demonstrates consistency in providing high-quality fitness predictions of thermostability. We randomly bootstrap 50% of the samples from **DTm** and **DDG** for ten independent runs, the results are reported In **Table 6** for both the average performance and the standard deviation. ProtSSN achieves top performance with minimal variance.

Ablation Study

The effectiveness of ProtSSN with different modular designs is examined by its performance on the early released version **ProteinGym v0** (Notin et al., 2022a) with 86 DMS assays. The complete comparison of baseline comparison on **ProteinGym v0** is detailed in Table S1 in SI. For the ablation study, the results are summarized in **Fig. 4(a)-(d)** and detailed in Table S2, where each record is subject to a combination of key arguments under investigation. For instance, in the top orange box of **Fig. 4(a)**, we report all ablation results that utilize 6 EGNN layers for graph convolution, regardless of the different scales of ESM-2 or the definitions of node attributes. For all modules investigated in this section, we separately discuss their influence on predicting mutation effects when modifying a single site or an arbitrary number of sites. The two cases are marked on the y-axis by single' and 'all', respectively.

Inclusion of Roto-Translation Equivariance

We assess the effect of incorporating rotation and translation equivariance for protein geometric and topological encoding. Three types of graph convolutions are compared, including GCN (Kipf and Welling, 2017), GAT (Veličković et al., 2018), and EGNN (Satorras et al., 2021). The first two are classic non-equivariant graph convolutional methods, and the last one, which we apply in the main algorithm, preserves roto-translation equivariance. We fix the number of EGNN layers to 6 and examine the performance of the other two methods with either 4 or 6 layers. We find that integrating equivariance when embedding protein geometry would significantly improve prediction performance.

Sequence Encoding

We next investigate the benefits of defining data-driven node attributes for protein representation learning. We compare the performance of models trained on two sets of graph inputs: the first set defines its AA node attributes through trained ESM2 (Lin et al., 2023), while the second set uses one-hot encoded AAs for each node. A clear advantage of using hidden representations by prefix models over hardcoded attributes is evident from the results presented in **Fig. 4(b)**.

Depth of EGNN

Although graph neural networks can extract topological information from geometric inputs, it is vital to select an appropriate number of layers for the module to deliver the most expressive node representation without encountering the oversmoothing problem. We investigate a wide range of choices for the number of EGNN layers among {6, 12, 18}. As reported in **Fig. 4(c)**, embedding graph topology with deeper networks does not lead to performance improvements. A moderate choice of 6 EGNN layers is sufficient for our learning task.

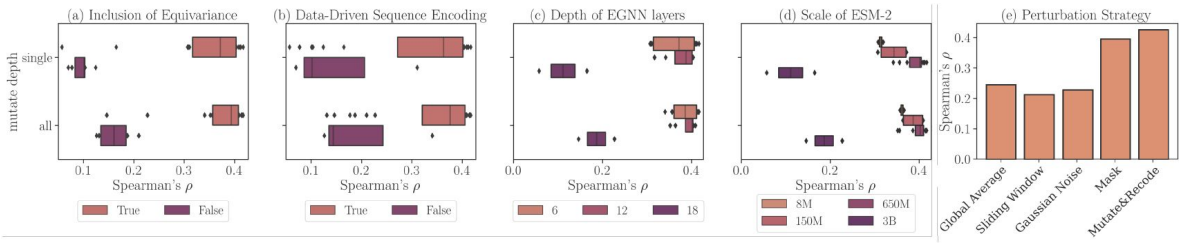


Figure 4.

Ablation Study on **ProteinGym v0** with different modular settings of ProtSSN. Each record represents the average Spearman's correlation of all assays. (a) Performance using different structure encoders: EGNN (orange) versus GCN/GAT (purple); (b) Performance using different node attributes: ESM2-embedded hidden representation (orange) versus one-hot encoding (purple); (c) Performance with varying numbers of EGNN layers; (d) Performance with different versions of ESM2 for sequence encoding; (e) Performance using different amino acid perturbation strategies during pre-training.

Scale of ESM

We also evaluate ProtSSN on different choices of language embedding dimensions to study the trade-off between computational cost and input richness. Various scales of prefix models, including {8, 150, 650, 3000} millions of parameters, have been applied to produce different sequential embeddings with {320, 640, 1280, 2560} dimensions, respectively. **Fig. 4(d)** reveals a clear preference for ESM2-t33, which employs 650 million parameters to achieve optimal model performance with the best stability. Notably, a higher dimension and richer semantic expression do not always yield better performance. A performance degradation is observed at ESM2 with 3 billion parameters.

AA Perturbation Strategies

ProtSSN utilizes noise sampling at each epoch on the node attributes to emulate random mutations in nature. This introduction of noise directly affects the node attribute input to the graphs. Alongside the 'mutate-then-recode' method we implemented in the main algorithm, we examined four additional strategies to perturb the input data during training. The construction of these additional strategies is detailed below, and their corresponding Spearman's ρ on **ProteinGym v0** is in **Fig. 4(e)**.

Global Average

Suppose the encoded sequential representation of a node is predominantly determined by its residue. In essence, the protein sequence encoder will return similar embeddings for nodes of the same residue, albeit at different positions within a protein. With this premise, the first strategy replaces the node embedding for perturbed nodes with the average representations of the same residues. For example, when perturbing an AA to glycine, an overall representation of glycine is assigned by extracting and average-pooling all glycine representations from the input sequence.

Sliding Window

The presumption in the previous method neither aligns with the algorithmic design nor biological heuristics. Self-attention discerns the interaction of the central token with its neighbors across the entire document (protein). The representation of a residue is influenced by both its neighbors' residues and locations. Thus, averaging embeddings of residues from varying positions is likely to forfeit positional information of the modified residue. For this purpose, we consider a sliding window of size 3 along the protein sequence.

Gaussian Noise

The third option regards node embeddings of AA as hidden vectors and imposes white noise on the vector values. We define the mean= 0 and variance= 0.5 on the noise, making the revised node representation $\tilde{\boldsymbol{v}} = \boldsymbol{v} + \mathcal{N}(0, 0.5)$.

Mask

Finally, we employ the masking technique prevalent in masked language modeling and substitute the perturbed residue token with a special <mask> token. The prefix language model will interpret it as a novel token and employ self-attention mechanisms to assign a new representation to it. We utilize the same hyperparameter settings as that of BERT (Devlin et al., 2018) and choose 15% of tokens per iteration as potentially influenced subjects. Specifically, 80% of these subjects are replaced with <mask>, 10% of them are replaced randomly (to other residues), and the remaining 10% stay unchanged.

Folding Methods

The analysis of protein structure through topological embeddings poses challenges due to the inherent difficulty in obtaining accurate structural observations through experimental techniques such as NMR (Wüthrich, 1990) and Cryo-EM (Elmlund et al., 2017). As a substitution, we use *in silico* folding models such as AlphaFold2 (Jumper et al., 2021) and ESMFold (Lin et al., 2023). Although these folding methods may introduce additional errors due to the inherent uncertainty or inaccuracy in structural predictions, we investigate the robustness of our model in the face of potential inaccuracies introduced by these folding strategies.

Table 4 examines the impact of different folding strategies on the performance of DMS predictions for ESM-if1, MIF-ST, and ProtSSN. Although the performance based on ESMFold-derived protein structures generally lags behind structures folded by AlphaFold2, the minimal difference between the two strategies across all four benchmarks validates the robustness of our ProtSSN in response to variations in input protein structures. We deliberately divide the assays in **ProteinGym v0** (Notin et al., 2022a) into two groups of interaction (*e.g.*, binding and stability) and catalysis (*e.g.*, activity), as the former is believed more sensitive to the structure of the protein (Robertson and Murphy, 1997).

ProtSSN emerges as the most robust method, maintaining consistent performance across different folding strategies, which underscores the importance of our model's resilience to variations in input protein structures. The reported performance difference for both ESM-if1 and MIF-ST is 3–5 times higher than that of ProtSSN. The inconsistency between the optimal results for **DDG** and the scores reported in **Table 3** comes from the utilization of crystal structures in the main results. Another noteworthy observation pertains to the performance of ESM-if1 and MIF-ST on **DDG**. In this case, predictions based on ESMFold surpass those based on AlphaFold2, even outperforming predictions derived from crystal structures. However, the superior performance of these methods with ESMFold hinges on inaccurate protein structures, rendering them less reliable.

Performance Enhancement with MSA and Ensemble

We extend the evaluation on **ProteinGym v0** to include a comparison of our zero-shot ProtSSN with few-shot learning methods that leverage MSA information of proteins (Meier et al., 2021). The details are reported in **Table 5**, where the performance of baseline methods are retrieved from *https://github.com/OATML-Markslab/ProteinGym*. For ProtSSN, we employ an ensemble of nine models with varying sizes of kNN graphs ($k \in \{10, 20, 30\}$) and EGNN hidden neurons ({512, 768, 1280}), as discussed previously. The MSA depth reflects the number of MSA sequences that can be found for the protein, which influences the quality of MSA-based methods.

ProtSSN demonstrates superior performance among non-MSA zero-shot methods in predicting both single-site and multi-site mutants, underscoring its pivotal role in guiding protein engineering towards deep mutation. Furthermore, ProtSSN outperforms MSA-based methods across taxa, except for viruses, where precise identification of conserved sites is crucial for positive mutations, especially in spike proteins (Katoh and Standley, 2021; Mercurio et al., 2021). In essence, ProtSSN provides an efficient and effective solution for variant effects prediction when compared to both non-MSA and MSA-based methods. Note that although MSA-based methods generally consume fewer trainable parameters than non-MSA methods, they incur significantly higher costs in terms of search time and memory usage. Moreover, the inference performance of MSA-based methods relies heavily on the quality of the input MSA, where the additionally involved variance makes impacts the stability of the model performance.

Table 4.

Influence of folding strategies (AlphaFold2 and ESMFold) on prediction performance for structure-involved models.

	DTm			**DDG**			**ProteinGym**-interaction			**ProteinGym**-catalysis		
	AlphaFold2	ESMFold	**diff** ↓	AlphaFold2	ESMFold	**diff** ↓	AlphaFold2	ESMFold	**diff** ↓	AlphaFold2	ESMFold	**diff** ↓
avg. pLDDT	90.86	83.22	-	95.19	86.03	-	82.86	65.69	-	85.80	73.45	-
ESM-if1	0.395	0.371	0.024	0.457	0.488	-0.031	0.351	0.259	0.092	0.386	0.368	0.018
MIF-ST	0.400	0.378	0.022	0.438	0.423	-0.015	0.390	0.327	0.063	0.408	0.388	0.020
k30_h1280	0.384	0.370	0.014	0.396	0.390	**0.006**	0.398	0.373	**0.025**	0.443	0.439	**0.004**
k30_h768	0.359	0.356	**0.003**	0.378	0.366	0.012	0.400	0.374	**0.026**	0.442	0.436	**0.006**
k30_h512	0.413	0.399	0.014	0.408	0.394	0.014	0.400	0.372	**0.028**	0.447	0.442	**0.005**
k20_h1280	0.415	0.391	0.024	0.429	0.410	0.019	0.399	0.365	0.034	0.446	0.441	**0.005**
k20_h768	0.415	0.403	**0.012**	0.419	0.397	0.022	0.401	0.370	0.031	0.449	0.442	0.007
k20_h512	0.419	0.395	0.024	0.441	0.432	**0.009**	0.406	0.371	0.035	0.449	0.439	0.010
k10_h1280	0.406	0.391	0.015	0.426	0.411	0.015	0.396	0.365	0.031	0.440	0.434	**0.006**
k10_h768	0.400	0.391	**0.009**	0.414	0.400	0.014	0.379	0.349	0.030	0.431	0.421	0.010
k10_h512	0.383	0.364	0.019	0.424	0.414	**0.010**	0.389	0.364	**0.025**	0.440	0.432	0.008

† The top three are highlighted by **First**, **Second**, **Third**.

Table 5.

Variant Effect Prediction on ProteinGym v0 with both zero-shot and few-shot methods. Results are retrieved from (Notin et al., 2022a).

Model	**Type**	**# Params** (million)	ρ **(by mutation depth)** ↑			ρ **(by MSA depth)** ↑			ρ **(by taxon)** ↑			
			Single	Double	All	Low	Medium	High	Prokaryote	Human	Eukaryote	Virus
few-shot methods												
SiteIndep	single	-	0.378	0.322	0.378	0.431	0.375	0.342	0.343	0.375	0.401	0.406
EVmutation	single	-	0.423	0.401	0.423	0.406	0.403	0.484	0.499	0.396	0.429	0.381
Wavenet	single	-	0.399	0.344	0.400	0.286	0.404	0.489	0.492	0.373	0.442	0.321
GEMME	single	-	**0.460**	0.397	**0.463**	**0.444**	0.446	0.520	0.505	0.436	0.479	**0.451**
DeepSequence	single	-	0.411	0.358	0.416	0.386	0.391	0.505	0.497	0.396	0.461	0.332
	ensemble	-	0.433	0.394	0.435	0.386	0.411	0.534	0.522	0.405	0.480	0.361
EVE	single	-	0.451	0.406	0.452	0.417	0.434	0.525	0.518	0.411	0.469	0.436
	ensemble	-	**0.459**	**0.409**	**0.459**	0.424	**0.441**	0.532	0.526	0.419	0.481	**0.437**
MSA-Transfomer	single	100	0.405	0.358	0.426	0.372	0.415	0.500	0.506	0.387	0.468	0.379
	ensemble	500	0.440	0.374	0.440	0.387	0.428	0.513	0.517	0.398	0.467	0.406
Tranception-R	single	700	0.450	**0.427**	**0.453**	0.442	**0.438**	0.500	0.495	0.424	**0.485**	0.434
TranceptEVE	single	700	**0.481**	**0.445**	**0.481**	**0.454**	**0.465**	**0.542**	**0.539**	**0.447**	**0.498**	**0.461**
zero-shot methods												
RITA	ensemble	2,210	0.393	0.236	0.399	0.350	0.414	0.407	0.391	0.391	0.405	0.417
ESM-1v	ensemble	3,250	0.416	0.309	0.417	0.390	0.378	**0.536**	0.521	0.439	0.423	0.268
ProGen2	ensemble	10,779	0.421	0.312	0.423	0.384	0.421	0.467	0.497	0.412	0.459	0.373
ProtTrans	ensemble	6,840	0.417	0.360	0.413	0.372	0.395	0.492	0.498	0.419	0.400	0.322
ESM2	ensemble	18,843	0.415	0.316	0.413	0.391	0.381	0.509	0.508	**0.456**	0.461	0.213
SaProt	ensemble	1,285	0.447	0.368	0.450	**0.456**	0.410	**0.544**	**0.534**	**0.464**	0.460	0.334
ProtSSN	ensemble	1,476	0.433	0.381	0.433	0.406	0.402	0.532	**0.530**	0.436	**0.491**	0.290

† The top three are highlighted by **First**, **Second**, **Third**.

Model	# Params (million)	DTm	DDG
RITA	2,210	$0.195_{(0.045)}$	$0.255_{(0.061)}$
ESM-1v	3,250	$0.300_{(0.036)}$	$0.310_{(0.054)}$
Tranception	1,085	$0.202_{(0.039)}$	$0.277_{(0.062)}$
ProGen2	10,779	$0.293_{(0.042)}$	$0.282_{(0.063)}$
ProtTrans	6,840	$0.323_{(0.039)}$	$0.389_{(0.059)}$
ESM2	18,843	$0.346_{(0.035)}$	$0.383_{(0.055)}$
SaProt	1,285	$0.392_{(0.040)}$	$0.415_{(0.061)}$
ProtSSN	1,476	$\mathbf{0.425}_{(0.033)}$	$\mathbf{0.440}_{(0.057)}$

Table 6.

Average Spearman's *ρ* correlation of variant effect prediction on DTm and DDG for zero-shot methods with model ensemble. The values within () indicate the standard deviation of bootstrapping.

Discussion and conclusion

The development of modern computational methodologies for protein engineering is a crucial facet of *in silico* protein design. Effectively assessing the fitness of protein mutants not only supports cost-efficient experimental validations but also guides the modification of proteins toward enhanced or novel functions. Traditionally, computational methods rely on analyzing small sets of labeled data for specific protein assays, such as FLIP (Dallago et al., 2021), PEER (Xu et al., 2022), and PETA (Tan et al., 2024). More recent work has also focused on examining the relationship between models and supervised fitness prediction (Li et al., 2024). However, obtaining experimental data is often infeasible in real-world scenarios, particularly for positive mutants, due to the cost and complexity of protein engineering. This limitation renders supervised learning methods impractical for many applications. As a result, it is crucial to develop solutions that can efficiently suggest site mutations for wet experiments, even when starting from scratch without prior knowledge. Recent deep learning solutions employ a common strategy that involves establishing a hidden protein representation and masking potential mutation sites to recommend plausible amino acids. Previous research has primarily focused on extracting protein representations from either their sequential or structural modalities, with many treating the prediction of mutational effects merely as a secondary task following inverse folding or *de novo* protein design. These approaches often overlook the importance of comprehensive investigation on both global and local perspectives of amino acid interaction that are critical for featuring protein functions. Furthermore, these methods hardly tailored model designs for suggesting mutations, despite the significance of this type of task. In this work, we introduce ProtSSN, a denoising framework that effectively cascades protein sequence and structure embedding for predicting mutational effects. Both the protein language model and equivariant graph neural network are employed to encode the global interaction of amino acids with geometry-aware local enhancements.

On the other hand, existing benchmarks for evaluating computational solutions mostly focus on assessing model generalization on large-scale datasets (e.g., over 2 million mutants in **ProteinGym v1**). However, such high-throughput datasets often lack accurate experimental measurements (Frazer et al., 2021), leading to significant noise in the labels. Furthermore, the larger data volumes typically do not include environmental labels for mutants (e.g., temperature, pH), despite the critical importance of these conditions for biologists. In response, we propose **DTm** and **DDG**. These low-throughput datasets, which we have collected, emphasize the consistency of experimental conditions and data accuracy. They are traceable and serve as valuable complements to ProteinGym, providing a more detailed and reliable evaluation of computational models. We have extensively validated the efficacy of ProtSSN across various protein function assays and taxa, including two thermostability datasets prepared by ourselves. Our approach consistently demonstrates substantial promise for protein engineering applications, such as facilitating the design of mutation sequences with enhanced interaction and enzymatic activity.

References

Aprile FA *et al.* (2020) **Rational design of a conformation-specific antibody for the quantification of Aβ oligomers** *Proceedings of the National Academy of Sciences* **117**:13509–13518

Brandes N, Ofer D, Peleg Y, Rappoport N, Linial M. (2022) **ProteinBERT: A universal deep-learning model of protein sequence and function** *Bioinformatics* **38**:2102–2110

Dallago C, Mou J, Johnston KE, Wittmann BJ, Bhattacharya N, Goldman S, Madani A, Yang KK (2021) **FLIP: Benchmark tasks in fitness landscape inference for proteins** *bioRxiv* :2021–11

Devlin J, Chang MW, Lee K, Toutanova K. (2018) **BERT: Pre-training of deep bidirectional transformers for language understanding** *arXiv*

Elmlund D, Le SN, Elmlund H. (2017) **High-resolution Cryo-EM: the nuts and bolts** *Current Opinion in Structural Biology* **46**:1–6

Elnaggar A *et al.* (2021) **ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing** *IEEE Transactions on Pattern Analysis and Machine Intelligence*

Frauenfelder H, McMahon B. (1998) **Dynamics and function of proteins: the search for general concepts** *Proceedings of the National Academy of Sciences* **95**:4795–4797

Frazer J, Notin P, Dias M, Gomez A, Min JK, Brock K, Gal Y, Marks DS (2021) **Disease variant prediction with deep generative models of evolutionary data** *Nature* **599**:91–95

Gray VE, Hause RJ, Luebeck J, Shendure J, Fowler DM (2018) **Quantitative missense variant effect prediction using large-scale mutagenesis data** *Cell Systems* **6**:116–124

Hesslow D, Zanichelli N, Notin P, Poli I, Marks D. (2022) **RITA: a study on scaling up generative protein sequence models** *ICML Workshop on Computational Biology*

Hopf TA, Ingraham JB, Poelwijk FJ, Schärfe CP, Springer M, Sander C, Marks DS (2017) **Mutation effects predicted from sequence co-variation** *Nature Biotechnology* **35**:128–135

Hsu C, Verkuil R, Liu J, Lin Z, Hie B, Sercu T, Lerer A, Rives A. (2022) **Learning inverse folding from millions of predicted structures** *In: ICML* PMLR :8946–8970

Ismail AR, Kashtoh H, Baek KH (2021) **Temperature-resistant and solvent-tolerant lipases as industrial biocatalysts: Biotechnological approaches and applications** *International Journal of Biological Macromolecules* **187**:127–142

Jiang F, Bian J, Liu H, Li S, Bai X, Zheng L, Jin S, Liu Z, Yang GY, Hong L. (2023) **Creatinase: Using Increased Entropy to Improve the Activity and Thermostability** *The Journal of Physical Chemistry B* **127**:2671–2682

Jin W, Wohlwend J, Barzilay R, Jaakkola TS (2021) **Iterative Refinement Graph Neural Network for Antibody Sequence-Structure Co-design** *ICLR*

Jing B, Eismann S, Suriana P, Townshend RJL, Dror R. (2020) **Learning from Protein Structure with Geometric Vector Perceptrons** *ICLR*

Jones D, Kim H, Zhang X, Zemla A, Stevenson G, Bennett WD, Kirshner D, Wong SE, Lightstone FC, Allen JE (2021) **Improved protein–ligand binding affinity prediction with structure-based deep fusion inference** *Journal of Chemical Information and Modeling* **61**:1583–1592

Jumper J *et al.* (2021) **Highly accurate protein structure prediction with AlphaFold** *Nature* **596**:583–589

Katoh K, Standley DM (2021) **Emerging SARS-CoV-2 variants follow a historical pattern recorded in outgroups infecting non-human hosts** *Communications Biology* **4**

Khersonsky O *et al.* (2018) **Automated design of efficient and functionally diverse enzyme repertoires** *Molecular Cell* **72**:178–186

Kingma DP, Ba J. (2015) **ADAM: A method for stochastic optimization** *International Conference on Learning Representation*

Kipf TN, Welling M. (2017) **Semi-supervised classification with graph convolutional networks** *ICLR*

Koehler Leman J *et al.* (2023) **Sequence-structure-function relationships in the microbial protein universe** *Nature Communications* **14**

Kong X, Huang W, Liu Y. (2023) **Conditional Antibody Design as 3D Equivariant Graph Translation**

Laine E, Karami Y, Carbone A. (2019) **GEMME: a simple and fast global epistatic model predicting mutational effects** *Molecular biology and evolution* **36**:2604–2619

Li FZ, Amini AP, Yue Y, Yang KK, Lu AX (2024) **Feature reuse and scaling: Understanding transfer learning with protein language models** *bioRxiv* :2024–2

Lin Z *et al.* (2023) **Evolutionary-scale prediction of atomic-level protein structure with a language model** *Science* **379**:1123–1130

Liu Q, Xun G, Feng Y. (2019) **The state-of-the-art strategies of protein engineering for enzyme stabilization** *Biotechnology Advances* **37**:530–537

Madani A *et al.* (2023) **Large language models generate functional protein sequences across diverse families** *Nature Biotechnology* **41**:1099–1106

Meier J, Rao R, Verkuil R, Liu J, Sercu T, Rives A. (2021) **Language models enable zero-shot prediction of the effects of mutations on protein function** *In: NeurIPS* **34**:29287–29303

Mercurio I, Tragni V, Busto F, De Grassi A, Pierri CL (2021) **Protein structure analysis of the interactions between SARS-CoV-2 spike protein and the human ACE2 receptor: from conformational changes to novel neutralizing antibodies** *Cellular and Molecular Life Sciences* **78**:1501–1522

Moal IH, Fernández-Recio J. (2012) **SKEMPI: A structural kinetic and energetic database of mutant protein interactions and its use in empirical models** *Bioinformatics* **28**:2600–2607

Myung Y, Pires DE, Ascher DB (2022) **CSM-AB: graph-based antibody–antigen binding affinity prediction and docking scoring function** *Bioinformatics* **38**:1141–1143

Nijkamp E, Ruffolo JA, Weinstein EN, Naik N, Madani A. (2023) **ProGen2: exploring the boundaries of protein language models** *Cell Systems* **14**:968–978

Nikam R, Kulandaisamy A, Harini K, Sharma D, Gromiha MM (2021) **ProThermDB: thermodynamic database for proteins and mutants revisited after 15 years** *Nucleic Acids Research* **49**:D420–D424

Notin P, Dias M, Frazer J, Hurtado JM, Gomez AN, Marks D, Gal Y. (2022) **Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval** *ICML* :16990–17017

Notin P *et al.* (2023) **ProteinGym: Large-scale benchmarks for protein fitness prediction and design** *NeurIPS*

Notin P, Van Niekerk L, Kollasch AW, Ritter D, Gal Y, Marks DS (2022) **TranceptEVE: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction** *bioRxiv* :2022–12

Orengo C, Michie A, Jones S, Jones D, Swindells M, Thornton J. (1997) **CATH – a hierarchic classification of protein domain structures** *Structure* **5**:1093–1109 https://doi.org/10.1016/S0969-2126(97)00260-8

Rao R, Meier J, Sercu T, Ovchinnikov S, Rives A. (2021) **Transformer protein language models are unsupervised structure learners** *ICLR*

Rao RM, Liu J, Verkuil R, Meier J, Canny J, Abbeel P, Sercu T, Rives A. (2021) **MSA transformer** *ICML* :8844–8856

Riesselman AJ, Ingraham JB, Marks DS (2018) **Deep generative models of genetic variation capture the effects of mutations** *Nature Methods* **15**:816–822

Rives A *et al.* (2021) **Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences** *Proceedings of the National Academy of Sciences* **118**

Robertson AD, Murphy KP (1997) **Protein structure and the energetics of protein stability** *Chemical Reviews* **97**:1251–1268

Robinson-Rechavi M, Alibés A, Godzik A. (2006) **Contribution of electrostatic interactions, compactness and quaternary structure to protein thermostability: lessons from structural genomics of Thermotoga maritima** *Journal of Molecular Biology* **356**:547–557

Roscoe BP, Thayer KM, Zeldovich KB, Fushman D, Bolon DN (2013) **Analyses of the effects of all ubiquitin point mutants on yeast growth rate** *Journal of Molecular Biology* **425**:1363–1377

Sarkisyan KS *et al.* (2016) **Local fitness landscape of the green fluorescent protein** *Nature* **533**:397–401

Satorras VG, Hoogeboom E, Welling M. (2021) **E(n) equivariant graph neural networks** *ICML* :9323–9332

Sheng G, Zhao H, Wang J, Rao Y, Tian W, Swarts DC, van der Oost J, Patel DJ, Wang Y. (2014) **Structure-based cleavage mechanism of Thermus thermophilus Argonaute DNA guide strand-mediated DNA target cleavage** *Proceedings of the National Academy of Sciences* **111**:652–657

Shin JE, Riesselman AJ, Kollasch AW, McMahon C, Simon E, Sander C, Manglik A, Kruse AC, Marks DS (2021) **Protein design and variant prediction using autoregressive generative models** *Nature Communications* **12**

Su J, Han C, Zhou Y, Shan J, Zhou X, Yuan F. (2023) **Saprot: Protein language modeling with structure-aware vocabulary** *bioRxiv* :2023–10

Suzek BE, Wang Y, Huang H, McGarvey PB, Wu CH, Consortium U. (2015) **UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches** *Bioinformatics* **31**:926–932

Tan Y, Li M, Zhou Z, Tan P, Yu H, Fan G, Hong L. (2024) **PETA: evaluating the impact of protein transfer learning with sub-word tokenization on downstream applications** *Journal of Cheminformatics* **16**

Torng W, Altman RB (2019) **High precision protein functional site detection using 3D convolutional neural networks** *Bioinformatics* **35**:1503–1512

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. (2017) **Attention Is All You Need** *NeurIPS*

Velecky J, Hamsikova M, Stourac J, Musil M, Damborsky J, Bednar D, Mazurenko S. (2022) **SoluProtMutDB: A manually curated database of protein solubility changes upon mutations** *Computational and Structural Biotechnology Journal* **20**:6339–6347

Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. (2018) **Graph attention networks** *ICLR*

Vig J, Madani A, Varshney LR, Xiong C, Rajani N, et al. (2021) **BERTology Meets Biology: Interpreting Attention in Protein Language Models** *ICLR*

Wang Y, Xue P, Cao M, Yu T, Lane ST, Zhao H. (2021) **Directed evolution: methodologies and applications** *Chemical Reviews* **121**:12384–12444

Woodley JM (2013) **Protein engineering of enzymes for process applications** *Current Opinion in Chemical Biology* **17**:310–316

Wüthrich K. (1990) **Protein structure determination in solution by NMR spectroscopy** *Journal of Biological Chemistry* **265**:22059–22062

Xu M, Zhang Z, Lu J, Zhu Z, Zhang Y, Chang M, Liu R, Tang J. (2022) **Peer: A comprehensive and multi-task benchmark for protein sequence understanding** *NeurIPS* **35**:35156–35173

Yang KK, Lu AX, Fusi N. (2022) **Convolutions are competitive with transformers for protein sequence pretraining** *ICLR Machine Learning for Drug Discovery*

Yang KK, Wu Z, Arnold FH (2019) **Machine-learning-guided directed evolution for protein engineering** *Nature Methods* **16**:687–694

Yang KK, Zanichelli N, Yeh H. (2023) **Masked inverse folding with sequence transfer for protein representation learning** *Protein Engineering, Design and Selection* **36**

Yi K, Zhou B, Shen Y, Liò P, Wang YG (2023) **Graph denoising diffusion for inverse protein folding** *NeurIPS*

Zhang N, Bi Z, Liang X, Cheng S, Hong H, Deng S, Lian J, Zhang Q, Chen H. (2022) **Ontoprotein: Protein pretraining with gene ontology embedding** *arXiv*

Zhao W, Zhong B, Zheng L, Tan P, Wang Y, Leng H, de Souza N, Liu Z, Hong L, Xiao X. (2022) **Proteome-wide 3D structure prediction provides insights into the ancestral metabolism of ancient archaea and bacteria** *Nature Communications* **13**

Zheng L *et al.* (2022) **Loosely-packed dynamical structures with partially-melted surface being the key for thermophilic argonaute proteins achieving high DNA-cleavage activity** *Nucleic Acids Research* **50**:7529–7544

Zhou B, Zheng L, Wu B, Tan Y, Lv O, Yi K, Fan G, Hong L. (2024) **Protein engineering with lightweight graph denoising neural networks** *Journal of Chemical Information and Modeling* **64**:3650–3661

Zhou B, Zheng L, Wu B, Yi K, Zhong B, Lio P, Hong L. (2023) **Conditional Protein Denoising Diffusion Generates Programmable Endonucleases** *bioRxiv* :2023–8

Zhou Y, Pan Q, Pires DE, Rodrigues CH, Ascher DB (2023) **DDMut: predicting effects of mutations on protein stability using deep learning** *Nucleic Acids Research*

Editors

Reviewing Editor
Peter Koo
Cold Spring Harbor Laboratory, Cold Spring Harbor, United States of America

Senior Editor
Qiang Cui
Boston University, Boston, United States of America

Reviewer #1 (Public review):

After revisions:

My concerns have been addressed.

Prior to revisions:

Summary:
The authors introduce a denoising-style model that incorporates both structure and primary-sequence embeddings to generate richer embeddings of peptides. My understanding is that the authors use ESM for the primary sequence embeddings, take resolved structures (or use structural predictions from AlphaFold when they're not available), then develop an architecture to combine these two with a loss that seems reminiscent of diffusion models or masked language model approaches. The embeddings can be viewed as ensemble-style embedding of the two levels of sequence information, or with AlphaFold, an ensemble of two

methods (ESM+AlphaFold). The authors also gather external datasets to evaluate their approach and compare it to previous approaches. The approach seems promising, and appears to out-compete previous methods at several tasks. Nonetheless, I have strong concerns about a lack of verbosity as well as exclusion of relevant methods and references.

Advances:
I appreciate the breadth of the analysis and comparisons to other methods. The authors separate tasks, models, and sizes of models in an intuitive, easy-to-read fashion that I find valuable for selecting a method for embedding peptides. Moreover, the authors gather two datasets for evaluating embeddings' utility for predicting thermostability. Overall, the work should be helpful for the field as more groups choose methods/pretraining strategies amenable to their goals, and can do so in an evidence-guided manner.

Considerations:
Primarily, a majority of the results and conclusions (e.g., Table 3) are reached using data and methods from ProteinGym, yet the best-performing methods on ProteinGym are excluded from the paper (e.g., EVE-based models and GEMME). In the ProteinGym database, these methods outperform ProtSSN models. Moreover, these models were published over a year---or even 4 years in the case of GEMME---before ProtSSN, and I do not see justification for their exclusion in the text.

Secondly, related to comparison of other models, there is no section in the methods about how other models were used, or how their scores were computed. When comparing these models, I think it's crucial that there are explicit derivations or explanations for the exact task used for scoring each method. In other words, if the pre-training is indeed the important advance of the paper, the paper needs to show this more explicitly by explaining exactly which components of the model (and previous models) are used for evaluation. Are the authors extracting the final hidden layer representations of the model, treating these as features, then using these features in a regression task to predict fitness/thermostability/DDG etc.? How are the model embeddings of other methods being used, since, for example, many of these methods output a k-dimensional embedding of a given sequence, rather than one single score that can be correlated with some fitness/functional metric. Summarily, I think the text is lacking an explicit mention of how these embeddings are being summarized or used, as well as how this compares to the model presented.

I think the above issues can mainly be addressed by considering and incorporating points from Li et al. 2024[1] and potentially Tang & Koo 2024[2]. Li et al.[1] make extremely explicit the use of pretraining for downstream prediction tasks. Moreover, they benchmark pretraining strategies explicitly on thermostability (one of the main considerations in the submitted manuscript), yet there is no mention of this work nor the dataset used (FLIP (Dallago et al., 2021)) in this current work. I think a reference and discussion of [1] is critical, and I would also like to see comparisons in line with [1], as [1] is very clear about what features from pretraining are used, and how. If the comparisons with previous methods were done in this fashion, this level of detail needs to be included in the text.

To conclude, I think the manuscript would benefit substantially from a more thorough comparison of previous methods. Maybe one way of doing this is following [1] or [2], and using the final embeddings of each method for a variety of regression tasks---to really make clear where these methods are performing relative to one another. I think a more thorough methods section detailing how previous methods did their scoring is also important. Lastly, TranceptEVE (or a model comparable to it) and GEMME should also be mentioned in these results, or at the bare minimum, be given justification for their absence.

[1] Feature Reuse and Scaling: Understanding Transfer Learning with Protein Language Models Francesca-Zhoufan Li, Ava P. Amini, Yisong Yue, Kevin K. Yang, Alex X. Lu bioRxiv 2024.02.05.578959; doi: https://doi.org/10.1101/2024.02.05.578959

[2] Evaluating the representational power of pre-trained DNA language models for regulatory genomics Ziqi Tang, Peter K Koo bioRxiv 2024.02.29.582810; doi: https://doi.org/10.1101/2024.02.29.582810

https://doi.org/10.7554/eLife.98033.2.sa2

Reviewer #2 (Public review):

Summary:

To design proteins and predict disease, we want to predict the effects of mutations on the function of a protein. To make these predictions, biologists have long turned to statistical models that learn patterns that are conserved across evolution. There is potential to improve our predictions however by incorporating structure. In this paper the authors build a denoising auto-encoder model that incorporates sequence and structure to predict mutation effects. The model is trained to predict the sequence of a protein given its perturbed sequence and structure. The authors demonstrate that this model is able to predict the effects of mutations better than sequence-only models.

As well, the authors curate a set of assays measuring the effect of mutations on thermostability. They demonstrate their model also predicts the effects of these mutations better than previous models and make this benchmark available for the community.

Strengths:

The authors describe a method that makes accurate mutation effect predictions by informing its predictions with structure.

Weaknesses:

In the review period, the authors included a previous method, SaProt, that similarly uses protein structure to predict the effects of mutations, in their evaluations.
They see that SaProt performs similarly to their method.

Readers should note that methods labelled as "few-shot" in comparisons do not make use of experimental labels, but rather use sequences inferred as homologous; these sequences are also often available even if the protein has never been experimentally tested.

ProteinGym is largely made of deep mutational scans, which measure the effect of every mutation on a protein. These new benchmarks contain on average measurements of less than a percent of all possible point mutations of their respective proteins. It is unclear what sorts of protein regions these mutations are more likely to lie in; therefore it is challenging to make conclusions about what a model has necessarily learned based on its score on this benchmark. For example, several assays in this new benchmark seem to be similar to each other, such as four assays on ubiquitin performed in pH 2.25 to pH 3.0.

The authors state that their new benchmarks are potentially more useful than those of ProteinGym, citing Frazer 2021; readers should be aware that the mutations from the later source are actually mutations whose impact on human health has been determined through multiple sources, including population genetics, clinical evidence and some experiment.

https://doi.org/10.7554/eLife.98033.2.sa1

Author response:

The following is the authors' response to the original reviews.

> ***Response to Reviewer 1***
>
> *Summary:*
>
> *The authors introduce a denoising-style model that incorporates both structure and primary-sequence embeddings to generate richer embeddings of peptides. My understanding is that the authors use ESM for the primary sequence embeddings, take resolved structures (or use structural predictions from AlphaFold when they're not available), and then develop an architecture to combine these two with a loss that seems reminiscent of diffusion models or masked language model approaches. The embeddings can be viewed as ensemble-style embedding of the two levels of sequence information, or with AlphaFold, an ensemble of two methods (ESM+AlphaFold). The authors also gather external datasets to evaluate their approach and compare it to previous approaches. The approach seems promising and appears to out-compete previous methods at several tasks. Nonetheless, I have strong concerns about a lack of verbosity as well as the exclusion of relevant methods and references.*

Thank you for the comprehensive summary. Regarding the concerns listed in the review below, we have made point-to-point response. We also modified our manuscript in accordance.

> *Advances:*
>
> *I appreciate the breadth of the analysis and comparisons to other methods. The authors separate tasks, models, and sizes of models in an intuitive, easy-to-read fashion that I find valuable for selecting a method for embedding peptides. Moreover, the authors gather two datasets for evaluating embeddings' utility for predicting thermostability. Overall, the work should be helpful for the field as more groups choose methods/pretraining strategies amenable to their goals, and can do so in an evidence-guided manner.*

Thank you for recognizing the strength of our work in terms of the notable contributions, the solid analysis, and the clear presentation.

> *Considerations:*
>
> *(1) Primarily, a majority of the results and conclusions (e.g., Table 3) are reached using data and methods from ProteinGym, yet the best-performing methods on ProteinGym are excluded from the paper (e.g., EVEbased models and GEMME). In the ProteinGym database, these methods outperform ProtSSN models. Moreover, these models were published over a year---or even 4 years in the case of GEMME---before ProtSSN, and I do not see justification for their exclusion in the text.*

We decided to exclude the listed methods from the primary table as they are all MSA-based methods, which are considered few-shot methods in deep learning (Rao et al., ICML, 2021). In contrast, the proposed ProtSSN is a zero-shot method that makes inferences based on less information than few-shot methods. Moreover, it is possible for MSA-based methods to query aligned sequences based on predictions. For instance, Tranception (Notin et al., ICML, 2022) selects the model with the optimal proportions of logits and retrieval results according to the average correlation score on ProteinGym (Table 10, Notin et al., 2022).

With this in mind, we only included zero-shot deep learning methods in Table 3, which require no more than the sequence and structure of the underlying wild-type protein when scoring the mutants. In the revision, we have added the performance of SaProt to Table 3, and the performance of GEMME, TranceptEVE, and SaProt to Table 5. Furthermore, we have

released the model's performance on the public leaderboard of ProteinGym v1 at proteingym.org.

> *(2) Secondly, related to the comparison of other models, there is no section in the methods about how other models were used, or how their scores were computed. When comparing these models, I think it's crucial that there are explicit derivations or explanations for the exact task used for scoring each method. In other words, if the pre-training is indeed an important advance of the paper, the paper needs to show this more explicitly by explaining exactly which components of the model (and previous models) are used for evaluation. Are the authors extracting the final hidden layer representations of the model, treating these as features, and then using these features in a regression task to predict fitness/thermostability/DDG etc.? How are the model embeddings of other methods being used, since, for example, many of these methods output a k-dimensional embedding of a given sequence, rather than one single score that can be correlated with some fitness/functional metric? Summarily, I think the text lacks an explicit mention of how these embeddings are being summarized or used, as well as how this compares to the model presented.*

Thank you for the suggestion. Below we address the questions in three points.

(1) The task and the scoring for each method. We followed your suggestion and added a new paragraph titled “Scoring Function” on page 9 to provide a detailed explanation of the scoring functions used by other deep learning zero-shot methods.

(2) The importance of individual pre-training modules. The complete architecture of the proposed ProtSSN model has been introduced on page 7-8. Empirically, the influence of each pre-training module on the overall performance has been examined through ablation studies on page 12. In summary, the optimal performance is achieved by combining all the individual modules and designs.

(3) The input of fitness scoring. For a zero-shot prediction task, the final score for a mutant will be calculated by wildly-used functions named log-odds ratio (for encoder models, including ours) or loglikelihood (for autoregressive models or inverse folding models. In the revision, we explicitly define these functions in sections “Inferencing” (page 7) and “Scoring Function” (page 9).

> *(3) I think the above issues can mainly be addressed by considering and incorporating points from Li et al. 2024[1] and potentially Tang & Koo 2024[2]. Li et al.[1] make extremely explicit the use of pretraining for downstream prediction tasks. Moreover, they benchmark pretraining strategies explicitly on thermostability (one of the main considerations in the submitted manuscript), yet there is no mention of this work nor the dataset used (FLIP (Dallago et al., 2021)) in this current work. I think a reference and discussion of [1] is critical, and I would also like to see comparisons in line with [1], as [1] is very clear about what features from pretraining are used, and how. If the comparisons with previous methods were done in this fashion, this level of detail needs to be included in the text.*

The initial version did not include an explicit comparison with the mentioned reference due to the difference in the learning task. In particular, [1] formulates a supervised learning task on predicting the continuous scores of mutants of specific proteins. In comparison, we make zero-shot predictions, where the model is trained in a self-supervised learning manner that requires no labels from experiments. In the revision, we added discussions in “Discussion and Conclusion” (lines 476-484):

> ***Recommendations For The Authors:***
>
> *Comment 1*
>
> *I found the methods lacking in the sense that there is never a simple, explicit statement about what is the exact input and output of the model. What are the components of the input that are required by the user (to generate) or supply to the model? Are these inputs different at training vs inference time? The loss function seems like it's trying to de-noise a modified sequence, can you make this more explicit, i.e. exactly what values/objects are being compared in the loss?*

We have added a more detailed description in the "Model Pipeline" section (page 7), which explains the distinct input requirements for training and inference, as well as the formulation of the employed loss function. To summarize:

(1) Both sequence and structure information are used in training and inference. Specifically, structure information is represented as a 3D graph with coordinates, while sequence information consists of AA-wise hidden representations encoded by ESM2-650M. During inference, instead of encoding each mutant individually, the model encodes the WT protein and uses the output probability scores relevant to the mutant to calculate the fitness score. This is a standard operation in many zero-shot fitness prediction models, commonly referred to as the log-odds-ratio.

(2) The loss function compares the differences between the noisy input sequence and the output (recovered) AA sequence. Noise is added to the input sequences, and the model is trained to denoise them (see “Ablation Study” for the different types of noise we tested). This approach is similar to a one-step diffusion process or BERT-style token permutation. The model learns to recover the probability of each node (AA) being one of 33 tokens. A cross-entropy loss is then applied to compare this distribution with the ground-truth (unpermuted) AA sequence, aiming to minimize the difference.

To better present the workflow, we revised the manuscript accordingly.

> *Comment 2*
>
> *Related to the above, I'm not exactly sure where the structural/tertiary structure information comes from. In the methods, they don't state exactly whether the 3D coordinates are given in the CATH repository or where exactly they come from. In the results section they mention using AlphaFold to obtain coordinates for a specific task---is the use of AlphaFold limited only to these tasks/this is to show robustness whether using AlphaFold or realized coordinates?*

The 3D coordinates of all proteins in the training set are derived from the crystal structures in CATH v4.3.0 to ensure a high-quality input dataset (see "Training Setup," Page 8). However, during the inference phase, we used predicted structures from AlphaFold2 and ESMFold as substitutes. This approach enhances the generalizability of our method, as in real-world scenarios, the crystal structure of the template protein to be engineered is not always available. The associated descriptions can be found in “Training Setup” (lines 271-272) and “Folding Methods” (lines 429-435).

> *Comment 3*
>
> *Lines 142+144 missing reference "Section establishes", "provided in Section ."*
>
> *199 "see Section " missing reference*

> *214 missing "Section"*

Thank you for pointing this out. We have fixed all missing references in the revision.

> *Comment 4*
>
> *Table 2 - seems inconsistent to mention the number of parameters in the first 2 methods, then not in the others (though I see in Table 3 this is included, so maybe should just be omitted in Table 2).*

In Table 2, we present the zero-shot methods used as baselines. Since many methods have different versions due to varying hyperparameter settings, we decided to list the number of parameters in the following tables.

We have double-checked both Table 3 and Table 5 and confirm that there is no inconsistency in the reported number of parameters. One potential explanation for the observed difference in the comment could be due to the differences in the number of parameters between single and ensemble methods. The ensemble method averages the predictions of multiple models, and we sum the total number of parameters across all models involved. For example, RITA-ensemble has 2210M parameters, derived from the sum of four individual models with 30M, 300M, 680M, and 1200M parameters.

> *Comment 5*
>
> *In general, I found using the word "type" instead of "residue" a bit unnatural. As far as I can tell, the norm in the field is to say "amino acid" or "residue" rather than "type". This somewhat confused me when trying to understand the methods section, especially when talking about injecting noise (I figured "type" may refer to evolutionarily-close, or physicochemically-close residues). Maybe it's not necessary to change this in every instance, but something to consider in terms of ease of reading.*

Thank you for your suggestion. The term "type" we used is a common expression similar to "class" in the NLP field. To avoid further confusion to the biologists, we have revised the manuscript accordingly.

> *Comment 6*
>
> *197 should this read "based on the kNN "algorithm"" (word missing) or maybe "based on "its" kNN"?*

We have corrected the typo accordingly. It now reads “the k-nearest neighbor algorithm (kNN)” (line 198).

> *Comment 7*
>
> *200 weights of dimension 93, where does this number come from?*

The edge features are derived by Zhou et al., 2024. We have updated the reference in the manuscript for clarity (lines 201-202).

> *Comment 8*
>
> *210-212 "representations of the noisy AA sequence are encoded from the noisy input" what is the "noisy AA sequence?" might be helpful to exactly defined what is "noisy input"*

> *or "noisy AA sequence". This sentence could potentially be worded to make it clearer, e.g. "we take the modified input sequence and embed it using [xyz]."*

We have revised the text accordingly. In the revised see lines 211-212:

> *Comment 9*
>
> *In Table 3*
>
> *Formatting, DTm (million), (million) should be under "# Params" likely?*
>
> *Also for DDG this is reported on only a few hundred mutations, it might be worth plotting the confidence intervals over the Spearman correlation (e.g. by bootstrapping the correlation coefficient).*

We followed the suggestion and added “million” under the "# Params". We have added the bootstrapped results for DDG and DTm to Table 6. For each dataset, we randomly sampled 50% of the data for ten independent runs. ProtSSN achieves the top performance with a considerably small variance.

> *Comment 10*
>
> *The paragraph in lines 319 to lines 328 I feel may lack sufficient evidence.*
>
> *"While sequence-based analysis cannot entirely replace the role of structure-based analysis, compared to a fully structure-based deep learning method, a protein language model is more likely to capture sufficient information from sequences by increasing the model scale, i.e., the number of trainable parameters."*
>
> *This claim is made without a citation, such as [1]. Increasing the scale of the model doesn't always align with improving out-of-sample/generalization performance. I don't feel fully convinced by the claim that worse prediction is ameliorated by increasing the number of parameters. In Table 3 the performance is not monotonic with (nor scales with) the number of parameters, even within a model. See ProGen2 Expression scores, or ESM-2 Stability scores, as a function of their model sizes. In [1], the authors discuss whether pretraining strategies are aligned with specific tasks. I think rewording this paragraph and mentioning this paper is important. Figure 3 shows that maybe there's some evidence for this but I don't feel entirely convinced by the plot.*

We agree that increasing the number of learnable parameters does not always result in better performance in downstream tasks. However, what we intended to convey is that language models typically need to scale up in size to capture the interactions among residues, while structure-based models can achieve this more efficiently with lower computational costs. We have rephrased this paragraph in the paper to clarify our point in lines 340-342.

> *Comment 11*
>
> *Line 327 related to my major comment, " a comprehensive framework, such as ProtSSN, exhibits the best performance." Refers to performance on ProteinGym, yet the best-performing methods on ProteinGym are excluded from the comparison.*

The primary comparisons were conducted using zero-shot models for fairness, meaning that the baseline models were not trained on MSA and did not use test performance to tune their hyperparameters. It's also worth noting that SaProt (the current SOTA model) had not been updated on the leaderboard at the time of submitting this paper. In the revised manuscript, we have included GEMME and TranceptEVE in Table 5 and SaProt in Tables 3, 5, and 6. While

ProtSSN does not achieve SOTA performance in every individual task, our key argument in the analysis is to highlight the overall advantage of hybrid encoders compared to single sequence-based or structure-based models. We made clearer statement in the revised manuscript (line 349):

> *Comment 12*
>
> *Line 347, line abruptly ends "equivariance when embedding protein geometry significantly." (?).*

We have fixed the typo, (lines 372-373):

> *Comment 13*
>
> *Figure 3 I think can be made clearer. Instead of using True/false maybe be more explicit. For example in 3b, say something like "One-hot encoded" or "ESM-2 embedded".*

The labels were set to True/False with the title of the subfigures so that they can be colored consistently.

Following the suggestion, we have updated the captions in the revised manuscript for clarity.

> *Comment 14*
>
> *Lines 381-382 "average sequential embedding of all other Glycines" is to say that the score is taken as the average score in which Glycine is substituted at every other position in the peptide? Somewhat confused by the language "average sequential embedding" and think rephrasing could be done to make things clearer.*

We have revised the related text accordingly a for clearer presentation (lines 406-413).

> *Comment 15*
>
> *Table 5, and in mentions to VEP, if ProtSSN is leveraging AlphaFold for its structural information, I disagree that ProtSSN is not an MSA method, and I find it unfair to place ProtSSN in the "non-MSA" categories. If this isn't the case, then maybe making clearer the inputs etc. in the Methods will help.*

Your response is well-articulated and clear, but here is a slight revision for improved clarity and flow:

We respectfully disagree with classifying a protein encoding method based solely on its input structure. While AF2 leverages MSA sequences to predict protein structures, this information is not used in our model, and our model is not exclusive to AF2-predicted structures. When applicable, the model can encode structures derived from experimental data or other folding methods. For example, in the manuscript, we compared the performance of ProtSSN using proteins folded by both AF2 and ESMFold.

However, we would like to emphasize that comparing the sensitivity of an encoding method across different structures or conformations is not the primary focus of our work. In contrast, some methods explicitly use MSA during model training. For instance, MSA-Transformer encodes MSA information directly into the protein embedding, and Tranception-retrieval utilizes different sets of MSA hyperparameters depending on the validation set's performance.

To avoid further confusion, we have revised the terms "MSA methods" and "non-MSA methods" in the manuscript to "zero-shot methods" and "few-shot methods."

> *Comment 16*
>
> *Table 3 they're highlighted as the best, yet on ProteinGym there's several EVE models that do better as well as GEMMA, which are not referenced.*

The comparison in Table 3 focuses on zero-shot methods, whereas GEMME and EVE are few-shot models. Since these methods have different input requirements, directly comparing them could lead to

unfair conclusions. For this reason, we reserved the comparisons with these few-shot models for Table 5, where we aim to provide a more comprehensive evaluation of all available methods.

> ***Response to Reviewer 2***
>
> *Summary:*
>
> *To design proteins and predict disease, we want to predict the effects of mutations on the function of a protein. To make these predictions, biologists have long turned to statistical models that learn patterns that are conserved across evolution. There is potential to improve our predictions however by incorporating structure. In this paper, the authors build a denoising auto-encoder model that incorporates sequence and structure to predict mutation effects. The model is trained to predict the sequence of a protein given its perturbed sequence and structure. The authors demonstrate that this model is able to predict the effects of mutations better than sequence-only models.*

Thank you for your thorough review and clear summary of our work. Below, we provide a detailed, pointby-point response to each of your questions and concerns.

> *Strengths:*
>
> *The authors describe a method that makes accurate mutation effect predictions by informing its predictions with structure.*

Thank you for your clear summary of our highlights.

> *Weaknesses:*
>
> *Comment 1*
>
> *It is unclear how this model compares to other methods of incorporating structure into models of biological sequences, most notably SaProt.*
>
> *(https://www.biorxiv.org/content/10.1101/2023.10.01.560349v1.full.pdf).*

In the revision, we have updated the performance of SaProt single models (with both masked and unmasked versions with the pLDDT score) and ensemble models in the Tables 3, 5, and 6.

In the revised manuscript, we have updated the performance results for SaProt's single models (both masked and unmasked versions with the pLDDT score) as well as the ensemble models. These updates are reflected in Tables 3, 5, and 6.

> *Comment 2*
>
> *ProteinGym is largely made of deep mutational scans, which measure the effect of every mutation on a protein. These new benchmarks contain on average measurements of less*

> *than a percent of all possible point mutations of their respective proteins. It is unclear what sorts of protein regions these mutations are more likely to lie in; therefore it is challenging to make conclusions about what a model has necessarily learned based on its score on this benchmark. For example, several assays in this new benchmark seem to be similar to each other, such as four assays on ubiquitin performed at pH 2.25 to pH 3.0.*

We agree that both DTm and DDG are smaller datasets, making them less comprehensive than ProteinGym. However, we believe DTm and DDG provide valuable supplementary insights for the following reasons:

(1) These two datasets are low-throughput and manually curated. Compared to datasets from highthroughput experiments like ProteinGym, they contain fewer errors from experimental sources and data processing, offering cleaner and more reliable data.

(2) Environmental factors are crucial for the function and properties of enzymes, which is a significant concern for many biologists when discussing enzymatic functions. Existing benchmarks like ProteinGym tend to simplify these factors and focus more on global protein characteristics (e.g., AA sequence), overlooking the influence of environmental conditions.

(3) While low-throughput datasets like DTm and DDG do not cover all AA positions or perform extensive saturation mutagenesis, these experiments often target mutations at sites with higher potential for positive outcomes, guided by prior knowledge. As a result, the positive-to-negative ratio is more meaningful than random mutagenesis datasets, making these benchmarks more relevant for evaluating model performance.

We would like to emphasize that DTm and DDG are designed to complement existing benchmarks rather than replace ProteinGym. They address different scales and levels of detail in fitness prediction, and their inclusion allows for a more comprehensive evaluation of deep learning models.

> ***Recommendations For The Authors:***
>
> *Comment 1*
>
> *I recommend including SaProt in your benchmarks.*

In the revision, we added comparisons with SaProt in all the Tables (3, 5 and 6).

> *Comment 2*
>
> *I also recommend investigating and giving a description of the bias in these new datasets.*

The bias of the new benchmarks could be found in Table 1, where the mutants are distributed evenly at different level of pH values.

In the revision, we added a discussion regarding the new datasets in “Discussion and Conclusion” (lines 496-504 of the revised version).

> *Comment 3*
>
> *I also recommend reporting the model's ability to predict disease using ClinVar -- this experiment is conspicuously absent.*

Following the suggestion, we retrieved 2,525 samples from the ClinVar dataset available on ProteinGym’s website. Since the official source did not provide corresponding structure files,

we performed the following three steps:

(1) We retrieved the UniProt IDs for the sequences from the UniProt website and downloaded the corresponding AlphaFold2 structures for 2,302 samples.

(2) For the remaining proteins, we used ColabFold 1.5.5 to perform structure prediction.

(3) Among these, 12 proteins were too long to be folded by ColabFold, for which we used the AlphaFold3 server for prediction.

All processed structural data can be found at https://huggingface.co/datasets/tyang816/ClinVar_PDB. Our test results are provided in the following table. ProtSSN achieves the top performance over baseline methods.

Author response table 1.

Model	AUC
ESM2	0.867
ESM-IF	0.735
MIF	0.767
MIF-ST	0.856
SaProt	0.878
ProtSSN	0.878

https://doi.org/10.7554/eLife.98033.2.sa0