

Supplementary Information for

From single-sequence structure prediction to protein fitness landscape through a composable, epistasis-aware mutation atlas

Weizhe Wang ^{1,2,3,4,†}, Zimu Yu ^{5,†}, Endi Yang ⁵, Ziyu Shi ⁵, Shize Yu ⁶, Jian Hu ^{1,2,3,4}, Yunxin Xu ^{5,*}, and Haipeng Gong ^{1,2,3,4,*}

¹MOE Key Laboratory of Bioinformatics, School of Life Sciences, Tsinghua University, Beijing 100084, China

²State Key Laboratory of Green Manufacturing, Tsinghua University, Beijing 100084, China

³Beijing Frontier Research Center for Biological Structure, Tsinghua University, Beijing 100084, China

⁴Center of Life Sciences and Artificial Intelligence, School of Life Sciences, Tsinghua University, Beijing 100084, China

⁵School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan 430073, China

⁶School of Software Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

[†]These authors contributed equally.

*Corresponding author. E-mail: hgong@tsinghua.edu.cn or xyx@zuel.edu.cn

Contents

S1	Datasets of Cerebra-Epistasis	2
S1.1	Datasets for Cerebra-Seq model training and evaluation	2
S1.2	Fitness and epistasis datasets for Cerebra-Epistasis training and evaluation	3
S2	Cerebra-Epistasis benchmark settings	5
S2.1	Structure prediction benchmark	6
S2.2	Structure inference time benchmark	6
S2.3	1M+ to 1M+ fitness prediction benchmark	6
S2.4	Low- <i>N</i> 1M/2M to 3M+ fitness prediction benchmark	7
S2.5	1M/2M to 3M+ fitness prediction benchmark	7
S2.6	cDNA proteolysis multi-mutant fitness prediction benchmark	7
S2.7	avGFP experimental fitness prediction benchmark	7
S2.8	Low- <i>N</i> 1M/2M to held-out 2M (epistasis) prediction benchmark	8
S2.9	S461 protein stability prediction benchmark	8
S3	Cerebra-Epistasis training and implementation details	9
S3.1	Cerebra-Seq pretraining	9

657	S3.2	Downstream fitness model and training setup	10
658	S3.3	Fitness and epistasis training objectives	10
659	S3.4	Benchmark-specific downstream training settings	11
660	S3.5	Specialized training strategies	11
661	S4	Baseline methods and comparison settings	13
662	S4.1	Structure prediction baselines	13
663	S4.2	Fitness and epistasis prediction baselines	13
664	S4.3	Fitness and epistasis prediction baselines	13
665	S4.4	$\Delta\Delta G$ prediction baselines	13
666	S5	Supplementary Results	14
667	S5.1	Supplementary Results for 1M+ to 1M+	14
668	S5.2	Supplementary Results for Low- N 1M/2M to 3M+	15
669	S5.3	Supplementary Results for 1M/2M to 3M+	15
670	S5.4	Supplementary Results for the cDNA proteolysis multi-mutant dataset in the 1M+ to	
671		1M+ benchmark	17
672	S5.5	Supplementary Results for the avGFP benchmark	18
673	S5.6	Supplementary Results for Low- N 1M/2M to held-out 2M (epistasis)	19
674	S5.7	Effect of end-to-end training on $\Delta\Delta G$ prediction	19
675	S6	Ablation and sensitivity analyses	20
676	S6.1	Contribution of Cerebra-Seq-derived structure-aware features to higher-order extrap-	
677		olation	21
678	S6.2	Contribution of explicit epistasis modeling to higher-order extrapolation	21
679	S6.3	Effect of maximum training mutation order on higher-order extrapolation	22
680	S6.4	Effect of double-mutant coverage on higher-order extrapolation	22
681	S6.5	Effect of end-to-end training on fitness and structure prediction	24
682	S7	Cerebra-Epistasis Inference	25
683	S7.1	Features and Notation	25
684	S7.2	Overall Cerebra-Epistasis workflow	26
685	S7.3	Cerebra-Seq structure prediction	27
686	S7.4	Downstream fitness prediction	30

687 S1 Datasets of Cerebra-Epistasis

688 S1.1 Datasets for Cerebra-Seq model training and evaluation

689 **Training datasets.** We trained Cerebra-Seq using experimental structures from the OpenFold PDB
690 training set and a separately curated AlphaFold Protein Structure Database (AFDB) distillation set. The
691 original OpenFold PDB set contained $\sim 130,000$ protein monomers. After excluding samples shorter
692 than 50 residues, $\sim 116,000$ PDB samples remained for training.

693 The AFDB distillation set was constructed from AlphaFold-predicted structures for final training [1].
694 We selected approximately 2.30 million candidate proteins from the published non-singleton structural
695 cluster representatives obtained by Foldseek clustering of AFDB50 [2]. For each candidate, we

downloaded the AlphaFold model_v4 structure in PDB format. By using DSSP secondary-structure assignments [3], we trimmed contiguous coil regions from both termini. We calculated the mean pLDDT and radius of gyration (R_g) over the retained region. We retained only structures with a mean pLDDT of at least 80, an effective trimmed length greater than 30 residues, and a full sequence length below 800 residues. We also required the trimmed structures to satisfy the length-dependent compactness criterion

$$R_g \leq 0.2653L^{0.3933} \text{ nm}, \quad (\text{S1})$$

where L denotes the effective length of the trimmed region in residues. This length-dependent empirical upper bound was derived from statistics of monomeric PDB structures. These filters yielded 583,100 protein structures for the AFDB distillation set, which was incorporated during the final fine-tuning stage.

Test datasets. Since the OpenFold dataset contains PDB structures released by December 2021, we used the annual CAMEO datasets from the two most recent complete calendar years as the primary test sets to evaluate our model: CAMEO 2024 (2024.01–2025.01) and CAMEO 2025 (2025.01–2026.01). After excluding targets shorter than 50 amino acids, we retained 642 and 573 protein targets, respectively, for evaluating the structure-prediction accuracy of Cerebra-Seq and the baseline models.

S1.2 Fitness and epistasis datasets for Cerebra-Epistasis training and evaluation

cDNA proteolysis dataset. Tsuboyama et al. developed a high-throughput cDNA display proteolysis assay to measure the thermodynamic folding stability of protein variants [4]. In this assay, proteins covalently linked to their corresponding cDNA were subjected to proteolysis, and the abundance of intact proteins was quantified by deep sequencing to infer folding stability. The resulting dataset contains large-scale stability measurements for single amino-acid variants and selected multiple mutants across hundreds of natural and designed protein domains.

For the **cDNA proteolysis dataset**, the `mut_type` field was used as the mutation annotation and the `ddG_ML` field as the mutation-effect label. Records for which `ddG_ML` could not be converted to a numeric value were removed. Records containing synonymous substitutions, insertions, or deletions were also excluded. Label values were then averaged across records with identical mutation annotations for the same protein, and records annotated as `wt` were subsequently excluded.

ArchStabMS1E10 Epistasis dataset. Faure et al. constructed large-scale combinatorial mutagenesis libraries to investigate the effects of higher-order amino-acid substitutions on protein stability and function [5]. Variant phenotypes were quantified using protein-fragment complementation assays, including AbundancePCA to measure protein abundance and, for selected libraries, BindingPCA to measure ligand binding, enabling systematic characterization of mutational effects and epistatic interactions across high-dimensional protein sequence spaces.

For the **ArchStabMS1E10 Epistasis dataset**, we retained libraries 1_lenient, 2, 3, and 4, and excluded library 1, which contained fewer records and was a subset of 1_lenient. Within each library, mutation annotations were inferred by comparing the mutant sequences with the corresponding wild-type sequence.

Antitoxin ParD3 Epistasis dataset. Lite et al. constructed a combinatorially complete library covering all $20^3 = 8,000$ amino-acid combinations at three key interaction positions in the ParD3 antitoxin and quantified the toxin-neutralization capacity of the variants using cellular growth competition and deep sequencing [6]. The processed variant-effect data used in this study were obtained from the subsequently released CoVES dataset [7].

For the **Antitoxin ParD3 Epistasis dataset**, the corresponding wild-type sequence was obtained from the original publication and added to the dataset. Records containing nonsense mutations were excluded, and label values were averaged across records with identical mutation annotations. Relative mutation effects were calculated by subtracting the wild-type label from each mutant label. The mutation annotations were then applied to the wild-type sequence to generate the corresponding mutant sequences.

TrpB Epistasis dataset. Johnston et al. constructed a combinatorially complete fitness landscape comprising all $20^4 = 160,000$ amino-acid combinations at four residues near the active site of a thermostable tryptophan synthase β -subunit (TrpB) [8]. Variant fitness was quantified using a pooled-culture enrichment assay in tryptophan-auxotrophic *Escherichia coli*, in which TrpB catalytic activity supports cellular growth, and changes in variant frequencies were measured by deep sequencing.

For the **TrpB Epistasis dataset**, the corresponding wild-type sequence was obtained from the original publication and added to the dataset. Records containing nonsense mutations were excluded. The processed mutation annotations were subsequently applied to the wild-type sequence to generate the corresponding mutant sequences.

Human Myoglobin Epistasis dataset. Küng et al. constructed a deep mutational scanning dataset for human myoglobin (hMb) using yeast surface display, fluorescence-activated cell sorting (FACS), and next-generation sequencing [9]. The variant library contained thousands of single and double mutants, and variant expression levels were quantified as expression fitness scores based on their distributions across fluorescence-sorting bins. These expression fitness scores were further shown to correlate with the thermostability of soluble hMb variants, providing a large-scale dataset for characterizing mutation effects and epistatic interactions.

For the **Human Myoglobin Epistasis dataset**, the wild-type amino-acid sequence was added to the dataset. Codon-level mutation annotations were converted to amino-acid-level mutation annotations, while records in which a mutant codon encoded a stop codon were excluded. Label values were averaged across records with identical mutation annotations, and records containing synonymous substitutions were further excluded.

CTXM Epistasis dataset. Judge et al. performed large-scale pairwise deep mutational scanning across 17 active-site residues of the CTX-M-14 β -lactamase to characterize epistatic interactions between amino acid substitutions [10]. Variant fitness was quantified by functional selection of *Escherichia coli* expressing the mutant enzymes in the presence of cefotaxime or ampicillin, followed by next-generation sequencing to measure changes in variant frequencies. The resulting fitness measurements quantify the relative antibiotic-resistance function of single and double mutants and enable systematic analysis of pairwise epistasis within the enzyme active site.

For the **CTXM Epistasis dataset**, low-confidence mutation records were excluded. Each original record was expanded into its two constituent single substitutions and the corresponding double substitution. The corresponding wild-type sequence was obtained from the original publication and added to the dataset. Mutation positions were then converted from Ambler numbering to the corresponding positions in the protein sequence. Within each antibiotic-selection condition, label values were averaged across records with identical mutation annotations.

ProteinGym DMS Substitutions dataset. Notin et al. developed ProteinGym, a large-scale benchmark that integrates deep mutational scanning datasets from diverse proteins and experimental systems for evaluating protein variant-effect prediction and protein design methods [11]. ProteinGym standardizes mutation annotations and experimental measurements across assays, with the direction of the DMS_score normalized such that higher values consistently indicate better variant fitness or performance in the measured phenotype, while the specific biological interpretation of the score remains assay dependent.

For the **ProteinGym DMS Substitutions dataset**, mutation annotations were validated and standardized. Within each assay, the wild-type sequence was inferred from the mutation annotations and corresponding mutant sequences, and the consistency of the inferred wild-type sequences was verified to ensure compatibility between the mutation annotations and mutant sequences.

For all datasets, the processed data were organized into separate directories for individual reference entries. Each directory contained three files: `data.csv`, `wt.fasta`, and `metadata.json`. The `data.csv` file contained three fields: `mutation_name`, representing the standardized mutation annotation; `mutated_sequence`, representing the corresponding mutant sequence; and `label`, representing the mutation-effect measurement. All mutation positions were standardized to 0-based indexing. The `wt.fasta` file contained the corresponding wild-type sequence, whereas the `metadata.json` file contained summary statistics for the corresponding mutation dataset. Detailed implementations of these preprocessing procedures are available in MutCleaner [12].

S2 Cerebra-Epistasis benchmark settings

We evaluated the structural accuracy and computational efficiency of Cerebra-Seq, the single-sequence structure predictor within Cerebra-Epistasis. We assessed the complete framework through benchmarks for fitness and epistasis prediction.

For fitness and epistasis evaluation, we used three sources of mutational fitness data containing multi-mutant measurements. The cDNA proteolysis dataset contributed 154 assays. After removing assays overlapping with the cDNA proteolysis dataset, ProteinGym contained 19 assays. We also curated an Epistasis dataset from published studies and associated data repositories; after removing overlaps with ProteinGym, this collection contained 10 assays. All available assays were used in the benchmarks and were assigned to individual evaluation settings according to their mutation-order composition and the requirements of each benchmark. Details of dataset collection, curation, and preprocessing are provided in Supplementary Section S1.2. Here, 1M and 2M denote single- and double-mutant variants, respectively; 3M+ denotes variants containing three or more substitutions, and 1M+ denotes variants containing one or more substitutions. For order-averaged evaluation, only

811 assay–order groups containing at least 15 test variants were included; metrics were then averaged equally
812 across eligible mutation orders within each assay and subsequently across assays. The experimental
813 settings for all benchmarks are summarized in Supplementary [Table S1](#).

814 **S2.1 Structure prediction benchmark**

815 The structure prediction benchmark was conducted on the CAMEO protein test sets described in
816 Supplementary [Section S1.1](#). Cerebra-Seq, ESMFold, OmegaFold, and SPIRED were run for four
817 forward cycles, including the initial pass. For Cerebra-Seq, five checkpoints selected based on the
818 training curves were used for prediction. Baseline checkpoints and candidate-generation settings,
819 including the ESM3 sampling procedure, are described in Supplementary [Section S4.1](#).

820 For each method and target, five candidate structures were generated and evaluated against the
821 corresponding experimental reference structure. For ESMFold and OmegaFold, candidates from the
822 two checkpoints were pooled within each method to form a single set of five structures. The highest
823 TM-score among the five candidates was used as the target-level score for all methods (best-of-5
824 evaluation).

825 **S2.2 Structure inference time benchmark**

826 Inference time was measured on a single NVIDIA A100 80GB GPU. Cerebra-Seq, ESMFold,
827 OmegaFold, and SPIRED were each run for four forward cycles, including the initial pass. ESM3 was
828 excluded from the inference-time benchmark because its iterative structure-generation protocol differed
829 from the four-cycle prediction protocol used for the other models. Timing was collected after model
830 loading and warm-up, and each measurement covered the complete inference workflow from sequence
831 input to all-atom structure output. Measurement was repeated three times for each sequence length,
832 and the mean inference time was reported.

833 For ESMFold and OmegaFold, chunking and subbatching were disabled, respectively, to exclude
834 the associated computational overhead from the measurements. For SPIRED, the timed workflow
835 included all-atom reconstruction by GDFold2 [\[13\]](#) with its default parameters.

836 **S2.3 1M+ to 1M+ fitness prediction benchmark**

837 The 1M+ to 1M+ benchmark evaluates fitness prediction for single- and multi-mutant variants within
838 the same assay. The benchmark comprised 29 assays, including 19 from ProteinGym and 10 from the
839 Epistasis dataset, each containing both single- and multi-mutant variants. All available variants from
840 each assay were included. The cDNA proteolysis assays were evaluated separately in a benchmark
841 described below.

842 For each assay, variants were divided into five folds using a position-balanced random splitting
843 strategy. Compared with a fully random split, this procedure approximately balanced the occurrence of
844 each mutation position across folds.

845 In each cross-validation round, four folds were used for model training and the remaining fold
846 was held out for testing. The procedure was repeated five times so that each variant was predicted
847 exactly once in the held-out test fold. Predictions from the five test folds were concatenated to obtain

848 out-of-fold predictions for the entire assay. Performance was evaluated both across all variants and
849 separately by mutation order. All baseline methods were evaluated using the same five-fold partitions.

850 **S2.4 Low- N 1M/2M to 3M+ fitness prediction benchmark**

851 Following the Low- N 1M/2M to 3M+ benchmarking strategy used in MULTI-evolve [14], we randomly
852 sampled single mutants and their combinatorial variants from each assay. The benchmark comprised
853 18 assays, including 10 from ProteinGym and 8 from the Epistasis dataset. Only assays containing 1M,
854 2M, and 3M+ variants were included.

855 For each assay, the sampled 1M and 2M variants formed a training set containing fewer than
856 300 labeled variants, while the corresponding 3M+ combinatorial variants were reserved for testing.
857 The 3M+ test variants represented unseen higher-order combinations composed exclusively of single
858 mutations present in the training set. All baseline methods used the same training and test variants.

859 **S2.5 1M/2M to 3M+ fitness prediction benchmark**

860 Consistent with the higher-order extrapolation setting used in MULTI-evolve [14], the full 1M/2M to
861 3M+ benchmark followed the same assay selection as the Low- N 1M/2M to 3M+ benchmark above,
862 but without Low- N sampling. For each assay, all available single mutants were retained, multi-mutant
863 variants were restricted to combinations of these single mutations, and all retained 1M and 2M variants
864 were used for training while all retained 3M+ variants were reserved for testing. All baseline methods
865 used the same training and test variants.

866 **S2.6 cDNA proteolysis multi-mutant fitness prediction benchmark**

867 Among the 398 cDNA proteolysis assays, 154 containing multi-mutant variants were included in this
868 benchmark. These assays were evaluated separately from the 1M+ to 1M+ benchmark because they
869 share a common protein-stability-related label definition across proteins, providing a more homogeneous
870 setting for cross-protein analysis.

871 For each assay, variants were divided into five folds using the same position-balanced random
872 splitting strategy as in the 1M+ to 1M+ benchmark. In each cross-validation round, Cerebra-Epistasis
873 was trained as a single shared model across all assays and evaluated on the corresponding held-out fold
874 of each assay, whereas baseline methods were trained separately for each assay using the same five-fold
875 splits. Details of the joint training procedure are provided in Supplementary [Section S3.5](#).

876 Predictions from the five held-out test folds were concatenated for each assay before calculating
877 assay-level performance. Performance was evaluated both across all variants and separately by mutation
878 order.

879 **S2.7 avGFP experimental fitness prediction benchmark**

880 The avGFP assay provided by ProteinGym [11] was used for model training, and 39 experimentally
881 measured avGFP variants reported by Mi et al. [15] were used as an independent test set.

882 For Cerebra-Epistasis, 10% of the ProteinGym data were randomly held out as a validation set.
883 A small number of candidate hyperparameter configurations were empirically evaluated, and the
884 configuration with the highest validation Spearman correlation was selected. The model was then
885 retrained on the complete ProteinGym avGFP dataset using the selected configuration and evaluated on
886 the 39 experimental variants. No test-set measurements were used for hyperparameter selection.

887 **S2.8 Low- N 1M/2M to held-out 2M (epistasis) prediction benchmark**

888 To evaluate pairwise epistasis prediction for unseen mutation combinations under limited training data,
889 we constructed a Low- N 1M/2M to held-out 2M (epistasis) benchmark comprising 22 assays, including
890 15 from ProteinGym and 7 from the Epistasis dataset. For each assay, approximately 300–400 labeled
891 1M and 2M variants were sampled, with all retained double mutants composed exclusively of single
892 mutations represented in the corresponding 1M set.

893 The 2M variants were divided into five folds using a position-balanced random splitting strategy,
894 while all sampled 1M variants were retained in the training set of every cross-validation round. In each
895 round, all sampled 1M variants together with four folds of 2M variants were used for training, and the
896 remaining 2M fold was held out for testing. Thus, the constituent single mutations of each test double
897 mutant were observed during training, while the specific pairwise combination was not.

898 For each held-out double mutant, predicted pairwise epistasis was calculated as the deviation of
899 its predicted fitness from the additive expectation of its constituent single mutations, with experimental
900 epistasis calculated analogously from the measured fitness values. This procedure was repeated for
901 each of the five 2M folds so that every retained double mutant was predicted exactly once as a held-out
902 test sample. The resulting out-of-fold epistasis predictions were concatenated for each assay before
903 calculating assay-level performance. All baseline methods were evaluated using the same five-fold
904 partitions.

905 **S2.9 S461 protein stability prediction benchmark**

906 The S461 benchmark [16] is a manually corrected subset of the S669 benchmark [17] and contains 461
907 single-point mutations with experimentally measured protein stability changes ($\Delta\Delta G$). Only proteins
908 containing at least 10 experimentally measured mutations were retained for evaluation.

909 Performance was evaluated separately for each retained protein and subsequently summarized
910 across proteins.

Table S1. Overview of the Cerebra-Epistasis benchmark settings used for model evaluation.

Benchmark	Dataset	Training data	Test data
1M+ to 1M+ fitness	ProteinGym + Epistasis (29)	Four folds of 1M+	Held-out fold of 1M+
Low- <i>N</i> 1M/2M to 3M+ fitness	ProteinGym + Epistasis (18)	Sampled 1M + 2M (< 300)	3M+
1M/2M to 3M+ fitness	ProteinGym + Epistasis (18)	All 1M + 2M	3M+
cDNA proteolysis multi-mutant fitness	cDNA proteolysis (154)	Four folds of 1M+	Held-out fold of 1M+
Low- <i>N</i> 1M/2M to held-out 2M (epistasis)	ProteinGym + Epistasis (22)	1M + four folds of 2M	Held-out fold of 2M
S461 protein stability	S461 (12)	cDNA proteolysis	S461
avGFP experimental fitness	avGFP	ProteinGym avGFP	39 experimental variants

S3 Cerebra-Epistasis training and implementation details

S3.1 Cerebra-Seq pretraining

The pretraining protocol for Cerebra-Seq was adapted from Cerebra [18]. We defined two PDB subsets: a 10K dataset containing 10,000 randomly selected monomers with sequence lengths between 100 and 400 residues and resolution better than 3 Å, and a 3 Å dataset containing all eligible PDB samples with resolution better than 3 Å. The full PDB training set comprised approximately 116,000 samples after length filtering. In the fourth stage, the model was fine-tuned on a mixture of the OpenFold PDB training set and the AFDB distillation set at a ratio of 1:2. The pLDDT prediction loss was applied only to PDB samples with experimental resolution better than 3 Å.

The training strategies are described in detail in Supplementary Table S2. The training objectives were adapted from Cerebra [18] with two modifications. First, we removed the masked-MSA prediction loss, as Cerebra-Seq uses single-sequence representations rather than MSA inputs. Second, for relative-rotation supervision, we replaced the component-wise L_1 error between the predicted and reference unit quaternions with the sign-invariant discrepancy.

$$\ell_{\text{quaternion}} = 1 - (\mathbf{q}_{\text{pred}} \cdot \mathbf{q}_{\text{true}})^2. \quad (\text{S2})$$

The inner product measures their angular agreement, and squaring it accounts for the equivalence of $\mathbf{q}$ and $-\mathbf{q}$ as representations of the same rotation. The resulting errors were aggregated as in the original Cerebra loss.

Table S2. Training strategy.

Stage	Datasets	Loss function	Crop size	Side-chain prediction
1	10K PDB subset	$\mathcal{L}_{\text{Cerebra}}^{\text{init}}$	256	×
2	3 Å PDB subset	$\mathcal{L}_{\text{Cerebra}}$	256	×
3	Full PDB training set	$\mathcal{L}_{\text{Cerebra}}^{\text{fine-tuning}}$	384	✓
4	PDB + AFDB distillation data	$\mathcal{L}_{\text{Cerebra}}^{\text{fine-tuning}}$	384	✓

S3.2 Downstream fitness model and training setup

Unless otherwise specified, the pretrained Cerebra-Seq module was kept frozen and only the downstream prediction network was optimized. The SE(3)-Transformer encoder consisted of an input ConvSE3 layer, followed by one SE(3)-Transformer block comprising an equivariant multi-head attention layer and a feed-forward layer, and an output ConvSE3 layer. The attention layer used four heads with a per-head dimension of 64. The hidden fibers contained 320 degree-0 channels and 32 degree-1 channels, whereas the output fibers contained 128 degree-0 channels and 32 degree-1 channels. The interaction function f_θ was implemented as an MLP with the layer dimension $r_H \rightarrow 256 \rightarrow 128 \rightarrow 1$ and GELU activation after the first two linear layers, where r_H denotes the interaction rank.

Models were optimized using Adam without weight decay. Experimental fitness labels were standardized separately for each assay using the mean and standard deviation computed from the corresponding training partition. Models were trained to predict standardized fitness values, and predictions were transformed back to the original label scale using the corresponding training-set statistics before evaluation. The S461 protein stability benchmark was an exception, for which no label normalization was applied. Gradients were clipped to a maximum global norm of 2.0.

S3.3 Fitness and epistasis training objectives

Training was based on a primary fitness prediction objective and, where applicable, an auxiliary pairwise epistasis objective. For a training set containing N variants, the fitness prediction loss was defined as

$$\mathcal{L}_{\text{fitness}} = \frac{1}{N} \sum_{i=1}^N \text{SmoothL1}_\beta(\hat{y}_i, y_i), \quad (\text{S3})$$

where $\hat{y}_i$ and y_i denote the predicted and experimental fitness values on the training scale, respectively, and $\beta = 1.0$. For benchmarks using label normalization, the loss was evaluated in the standardized fitness space.

When both constituent single mutants of a training double mutant were available in the same training partition, an auxiliary pairwise epistasis target was constructed. Let $\tilde{y}_{AB}$ denote the standardized fitness of a double mutant and $\tilde{y}_A$ and $\tilde{y}_B$ those of its constituent single mutants. The auxiliary pairwise epistasis target was defined as

$$\varepsilon_{AB}^{(2)} = \tilde{y}_{AB} - \tilde{y}_A - \tilde{y}_B. \quad (\text{S4})$$

Consistent with the fitness assembly function, the corresponding model-predicted pairwise epistasis was defined as

$$\hat{\varepsilon}_{AB}^{(2)} = f_\theta(c_2(\mathbf{u}_A + \mathbf{u}_B)) - f_\theta(\mathbf{u}_A) - f_\theta(\mathbf{u}_B), \quad (\text{S5})$$

where $c_m = 1/\sqrt{m}$ when mutation-order scaling was enabled and $c_m = 1$ otherwise. Only double mutants for which both constituent single-mutant measurements were present in the corresponding training partition were included in the auxiliary objective. The epistasis loss was then defined as

$$\mathcal{L}_{\text{epi}} = \frac{1}{N_{\text{epi}}} \sum_{j=1}^{N_{\text{epi}}} \text{SmoothL1}_\beta(\hat{\varepsilon}_j, \varepsilon_j), \quad (\text{S6})$$

where N_{epi} is the number of eligible double mutants in the training partition. The overall training objective was

$$\mathcal{L} = \mathcal{L}_{\text{fitness}} + \lambda_{\text{epi}} \mathcal{L}_{\text{epi}}, \quad (\text{S7})$$

where λ_{epi} controls the contribution of explicit pairwise epistasis supervision. When $\lambda_{\text{epi}} = 0$, only the fitness prediction objective was optimized. Benchmark-specific values of λ_{epi} and the mutation-order scaling setting are summarized in Supplementary Table S3.

S3.4 Benchmark-specific downstream training settings

Most training settings were shared across benchmarks. A small number of settings, including the learning rate, interaction rank r_H , pairwise epistasis supervision, and mutation-order scaling, were selected separately for each benchmark. When hyperparameter selection or early stopping was required, a validation subset was held out from the training data for this purpose. The selected settings were then fixed and applied uniformly to all assays and data partitions within the same benchmark. Test data were never used for hyperparameter selection or early stopping. The final settings are summarized in Supplementary Table S3.

S3.5 Specialized training strategies

Joint training across cDNA proteolysis assays. For the cDNA proteolysis multi-mutant benchmark, Cerebra-Epistasis was trained jointly across assays using a single shared model. During training, each assay contributed its own training variants and assay-specific loss to update the shared model parameters. Label normalization was performed separately for each assay using the mean and standard deviation computed from its corresponding training partition. The corresponding training configuration is provided in Supplementary Table S3.

Table S3. Benchmark-specific training settings.

Benchmark	Epochs	Learning rate	r_H	c_m	λ_{epi}
1M+ to 1M+	Early stopping	5×10^{-4}	320	1	0
cDNA proteolysis	150	5×10^{-4}	320	1	0
Low- N 1M/2M to held-out 2M (epistasis)	150	5×10^{-4}	320	1	2
Low- N 1M/2M to 3M+	150	1×10^{-4}	64	$\frac{1}{\sqrt{m}}$	2
1M/2M to 3M+	150	1×10^{-4}	64	$\frac{1}{\sqrt{m}}$	2

End-to-end training. End-to-end training was first evaluated on the 154 cDNA proteolysis assays containing multi-mutant measurements. This protein-level split differed from the within-assay, position-balanced split described in Supplementary Section S2.6. Reference protein sequences were grouped into five clusters using NW-align-based sequence similarities, without a predefined sequence-identity threshold. Each cluster defined one cross-validation fold, with all variants of each assay kept together.

In each cross-validation round, an independent model was trained on four folds and evaluated on the remaining fold. Cerebra-Seq was initialized from its pretrained weights and then trained end to end with the downstream fitness-prediction components on the training assays. In the frozen-Cerebra-Seq baseline, Cerebra-Seq was initialized from the same pretrained weights but kept fixed, so that only the downstream components were optimized. Both settings used identical partitions. Models were trained independently across the five rounds.

The downstream components were trained with a learning rate of 5×10^{-4} . The pretrained Cerebra-Seq weights were fine-tuned at 5×10^{-6} . Experimental PDB structures were available for 139 of the 154 reference proteins. Structural supervision was applied to training proteins with experimental structures, using the end-to-end training objective

$$\mathcal{L}_{\text{E2E}} = \mathcal{L}_{\text{fitness}} + 0.001 \mathcal{L}_{\text{Cerebra}}^{\text{fine-tune}}. \quad (\text{S8})$$

Here, $\mathcal{L}_{\text{fitness}}$ is the fitness prediction loss defined above. The structural term $\mathcal{L}_{\text{Cerebra}}^{\text{fine-tune}}$ is the fine-tuning objective specified in Supplementary Table S2, incorporating the loss modifications described above. For the remaining 15 proteins without experimental structures, only the fitness prediction loss was applied. Structural metrics were evaluated on proteins with experimental structures in the held-out fold of each round.

To characterize training–test sequence similarity, we computed each held-out protein’s maximum sequence identity to any protein in the four training folds. Fold-specific nearest-neighbour identity statistics are provided in Supplementary Table S4. The maximum identity remained below 50% in every fold, with an overall maximum of 43.60%.

For the final S461 evaluation, we identified the best-performing epoch in each cDNA cross-validation round. The final training duration was determined from the arithmetic mean of these five epoch numbers. Models were then trained on all cDNA assays for the resulting number of epochs. They were evaluated on the independent S461 stability benchmark using the protocol described in Supplementary Section S2.9.

Table S4. Sequence similarity and structure-label availability across the cluster-based cDNA cross-validation folds. Nearest-neighbour (NN) identity is each held-out reference protein’s maximum sequence identity to any reference protein in the four training folds. The $\geq 40\%$ column counts held-out assays meeting this identity threshold. Structure-label counts indicate reference proteins with experimental PDB structures used for structural supervision in the training folds or structural evaluation in the held-out fold.

Test fold	Assays	Mean NN identity (%)	Maximum NN identity (%)	NN identity $\geq 40\%$ (<i>n</i>)	Training structure labels (<i>n</i>)	Test structure labels (<i>n</i>)
Fold 0	44	33.18	43.60	5	104	35
Fold 1	19	33.73	43.60	3	122	17
Fold 2	33	33.20	38.90	0	108	31
Fold 3	29	34.79	42.90	2	110	29
Fold 4	29	34.17	41.90	3	112	27

1008 **S4 Baseline methods and comparison settings**

1009 **S4.1 Structure prediction baselines**

1010 For structure prediction, Cerebra-Seq was compared with ESMFold [19], OmegaFold [20], ESM3 [21],
1011 and SPIRED [22]. ESMFold predictions used the officially released esmfold_v0 and esmfold_v1
1012 checkpoints. OmegaFold predictions used the officially released release1.pt and release2.pt
1013 checkpoints. For ESM3, each candidate structure was generated in an independent run with 20 iterative
1014 sampling steps. Unless otherwise specified, all baseline models were run using their default settings.

1015 **S4.2 Fitness and epistasis prediction baselines**

1016 For fitness and epistasis prediction, we included supervised baselines following the ProteinGym
1017 benchmark [11]. The one-hot encoding (OHE) baseline used one-hot-encoded mutant sequences as
1018 input to ridge regression. The OHE-based baselines additionally incorporated zero-shot fitness scores
1019 from ESM-1v [23], Tranception [24], or MSA Transformer [25]. The embedding-based baselines
1020 instead used mean-pooled sequence embeddings from each of these pretrained models, augmented
1021 with zero-shot fitness scores from the corresponding model, as inputs to ridge regression. We
1022 additionally compared Cerebra-Epistasis with MULTI-evolve [14], Kermut [26], and ProteinNPT [27].
1023 All supervised baseline predictors were trained on the same training partitions as Cerebra-Epistasis
1024 and evaluated on the corresponding test partitions within each benchmark. The ProteinGym-derived
1025 baselines followed their published implementations [11, 27], while MULTI-evolve and Kermut followed
1026 their respective original implementations.

1027 **S4.3 Fitness and epistasis prediction baselines**

1028 For fitness and epistasis prediction, we included supervised baselines following the ProteinGym
1029 benchmark [11]. The one-hot encoding (OHE) baseline used one-hot-encoded mutant sequences
1030 with a regularized linear prediction head. The augmented OHE baselines additionally used zero-shot
1031 fitness scores from ESM-1v [23], Tranception [24], or MSA Transformer [25] as prediction covariates.
1032 The embedding-based baselines processed residue-level embeddings from each of these pretrained
1033 models with a one-dimensional convolution followed by mean pooling, and used the corresponding
1034 zero-shot fitness score as an additional prediction covariate. We also compared Cerebra-Epistasis with
1035 MULTI-evolve [14], Kermut [26], and ProteinNPT [27].

1036 All supervised baselines were trained on the same training partitions as Cerebra-Epistasis and
1037 evaluated on the corresponding test partitions within each benchmark. The ProteinGym-derived
1038 baselines used the publicly released ProteinNPT implementation [11, 27], while MULTI-evolve and
1039 Kermut followed their respective original implementations.

1040 **S4.4 $\Delta\Delta G$ prediction baselines**

1041 For protein stability prediction on the S461 benchmark, Cerebra-Epistasis was compared with
1042 established $\Delta\Delta G$ prediction methods, including DDMut [28], ProteinEBM [29], PROSTATA [30],

1043 RaSP [31], SPIRED-Fitness [22], GeoDDG-3D and GeoDDG-Seq [32], Mutate Everything (ESM-
1044 2) [33], ThermoMPNN [34], Pythia [35], SPURS [36], and SPIRED-Stab [22]. All baseline predictions
1045 were generated using the publicly available pretrained model weights and the corresponding original
1046 inference procedures. Performance was evaluated on the S461 benchmark following the protocol
1047 described in Supplementary Section S2.9.

1048 S5 Supplementary Results

1049 S5.1 Supplementary Results for 1M+ to 1M+

1050 Results for the 1M+ to 1M+ benchmark described in Supplementary Section S2.3 are summarized in
1051 Supplementary Table S5. Cerebra-Epistasis achieved the strongest overall performance, ranking first
1052 on most evaluated correlation- and ranking-based metrics.

Table S5. Performance on the 1M+ to 1M+ benchmark under overall and order-averaged assay-level evaluation. Values are reported as mean $\pm$ SD across assays.

Model	Spearman ρ	Pearson r	Kendall τ	NDCG (top 10%)	Top recall (top 10%)
Overall assay-level evaluation					
OHE	0.74 $\pm$ 0.18	0.73 $\pm$ 0.17	0.56 $\pm$ 0.17	0.88 $\pm$ 0.09	0.50 $\pm$ 0.17
OHE + ESM-1v	0.75 $\pm$ 0.18	0.75 $\pm$ 0.16	0.58 $\pm$ 0.16	0.89 $\pm$ 0.08	0.51 $\pm$ 0.18
Embeddings + ESM-1v	0.70 $\pm$ 0.20	0.72 $\pm$ 0.18	0.53 $\pm$ 0.19	0.86 $\pm$ 0.11	0.48 $\pm$ 0.20
OHE + Tranception	0.77 $\pm$ 0.18	0.76 $\pm$ 0.16	0.60 $\pm$ 0.17	0.90 $\pm$ 0.07	0.54 $\pm$ 0.17
Embeddings + Tranception	0.77 $\pm$ 0.19	0.80 $\pm$ 0.14	0.60 $\pm$ 0.18	0.90 $\pm$ 0.08	0.55 $\pm$ 0.18
OHE + MSA Transformer	0.78 $\pm$ 0.19	0.77 $\pm$ 0.16	0.60 $\pm$ 0.17	0.91 $\pm$ 0.07	0.55 $\pm$ 0.18
Embeddings + MSA Transformer	0.76 $\pm$ 0.18	0.79 $\pm$ 0.16	0.59 $\pm$ 0.18	0.90 $\pm$ 0.08	0.54 $\pm$ 0.18
ProteinNPT	0.84 $\pm$ 0.17	0.91 $\pm$ 0.12	0.69 $\pm$ 0.18	0.94 $\pm$ 0.07	0.67 $\pm$ 0.20
Cerebra-Epistasis	0.85 $\pm$ 0.17	0.91 $\pm$ 0.12	0.70 $\pm$ 0.17	0.95 $\pm$ 0.06	0.68 $\pm$ 0.19
Order-averaged assay-level evaluation					
OHE	0.68 $\pm$ 0.17	0.68 $\pm$ 0.16	0.52 $\pm$ 0.15	0.86 $\pm$ 0.10	0.47 $\pm$ 0.14
OHE + ESM-1v	0.69 $\pm$ 0.16	0.69 $\pm$ 0.15	0.52 $\pm$ 0.14	0.86 $\pm$ 0.09	0.45 $\pm$ 0.14
Embeddings + ESM-1v	0.63 $\pm$ 0.21	0.65 $\pm$ 0.20	0.47 $\pm$ 0.18	0.81 $\pm$ 0.16	0.41 $\pm$ 0.16
OHE + Tranception	0.70 $\pm$ 0.16	0.70 $\pm$ 0.15	0.54 $\pm$ 0.14	0.87 $\pm$ 0.09	0.48 $\pm$ 0.14
Embeddings + Tranception	0.70 $\pm$ 0.17	0.73 $\pm$ 0.16	0.54 $\pm$ 0.16	0.86 $\pm$ 0.11	0.47 $\pm$ 0.14
OHE + MSA Transformer	0.71 $\pm$ 0.17	0.71 $\pm$ 0.16	0.55 $\pm$ 0.15	0.87 $\pm$ 0.09	0.49 $\pm$ 0.14
Embeddings + MSA Transformer	0.69 $\pm$ 0.17	0.71 $\pm$ 0.16	0.53 $\pm$ 0.15	0.85 $\pm$ 0.11	0.45 $\pm$ 0.15
ProteinNPT	0.78 $\pm$ 0.19	0.85 $\pm$ 0.16	0.63 $\pm$ 0.18	0.91 $\pm$ 0.09	0.60 $\pm$ 0.19
Cerebra-Epistasis	0.78 $\pm$ 0.17	0.85 $\pm$ 0.15	0.63 $\pm$ 0.17	0.92 $\pm$ 0.09	0.62 $\pm$ 0.18

S5.2 Supplementary Results for Low- N 1M/2M to 3M+

Results for the Low- N 1M/2M to 3M+ benchmark described in Supplementary Section S2.4 are summarized in Supplementary Table S6. Cerebra-Epistasis achieved the strongest overall performance and retained its advantage under order-averaged evaluation.

Table S6. Performance on the Low- N 1M/2M to 3M+ benchmark under overall and order-averaged assay-level evaluation. Values are reported as mean $\pm$ SD across assays.

Model	Spearman ρ	Pearson r	Kendall τ	NDCG (top 10%)	Top recall (top 10%)
Overall assay-level evaluation					
OHE	0.60 $\pm$ 0.25	0.61 $\pm$ 0.25	0.45 $\pm$ 0.21	0.84 $\pm$ 0.11	0.46 $\pm$ 0.23
OHE + ESM-1v	0.63 $\pm$ 0.23	0.64 $\pm$ 0.22	0.47 $\pm$ 0.19	0.86 $\pm$ 0.10	0.48 $\pm$ 0.22
Embeddings + ESM-1v	0.63 $\pm$ 0.23	0.63 $\pm$ 0.23	0.47 $\pm$ 0.19	0.84 $\pm$ 0.12	0.47 $\pm$ 0.21
OHE + Tranception	0.63 $\pm$ 0.22	0.64 $\pm$ 0.21	0.47 $\pm$ 0.19	0.86 $\pm$ 0.10	0.48 $\pm$ 0.22
Embeddings + Tranception	0.62 $\pm$ 0.26	0.66 $\pm$ 0.24	0.47 $\pm$ 0.21	0.85 $\pm$ 0.11	0.48 $\pm$ 0.23
OHE + MSA Transformer	0.64 $\pm$ 0.23	0.64 $\pm$ 0.22	0.48 $\pm$ 0.19	0.87 $\pm$ 0.11	0.48 $\pm$ 0.22
Embeddings + MSA Transformer	0.60 $\pm$ 0.29	0.61 $\pm$ 0.28	0.45 $\pm$ 0.24	0.80 $\pm$ 0.16	0.46 $\pm$ 0.22
MULTI-evolve	0.61 $\pm$ 0.26	0.65 $\pm$ 0.25	0.47 $\pm$ 0.22	0.85 $\pm$ 0.12	0.48 $\pm$ 0.22
Kermut	0.62 $\pm$ 0.25	0.63 $\pm$ 0.23	0.47 $\pm$ 0.21	0.86 $\pm$ 0.11	0.50 $\pm$ 0.21
ProteinNPT	0.56 $\pm$ 0.31	0.62 $\pm$ 0.30	0.42 $\pm$ 0.25	0.84 $\pm$ 0.13	0.48 $\pm$ 0.23
Cerebra-Epistasis	0.65 $\pm$ 0.24	0.66 $\pm$ 0.21	0.49 $\pm$ 0.21	0.86 $\pm$ 0.10	0.51 $\pm$ 0.19
Order-averaged assay-level evaluation					
OHE	0.62 $\pm$ 0.25	0.62 $\pm$ 0.25	0.47 $\pm$ 0.21	0.85 $\pm$ 0.11	0.45 $\pm$ 0.22
OHE + ESM-1v	0.63 $\pm$ 0.24	0.62 $\pm$ 0.24	0.47 $\pm$ 0.20	0.85 $\pm$ 0.10	0.45 $\pm$ 0.21
Embeddings + ESM-1v	0.62 $\pm$ 0.23	0.62 $\pm$ 0.24	0.47 $\pm$ 0.20	0.83 $\pm$ 0.12	0.45 $\pm$ 0.21
OHE + Tranception	0.62 $\pm$ 0.24	0.62 $\pm$ 0.23	0.47 $\pm$ 0.21	0.85 $\pm$ 0.10	0.46 $\pm$ 0.22
Embeddings + Tranception	0.61 $\pm$ 0.26	0.64 $\pm$ 0.24	0.47 $\pm$ 0.22	0.85 $\pm$ 0.11	0.45 $\pm$ 0.21
OHE + MSA Transformer	0.62 $\pm$ 0.24	0.62 $\pm$ 0.24	0.47 $\pm$ 0.21	0.85 $\pm$ 0.10	0.46 $\pm$ 0.22
Embeddings + MSA Transformer	0.59 $\pm$ 0.30	0.60 $\pm$ 0.28	0.46 $\pm$ 0.25	0.82 $\pm$ 0.15	0.43 $\pm$ 0.23
MULTI-evolve	0.63 $\pm$ 0.26	0.65 $\pm$ 0.25	0.48 $\pm$ 0.22	0.85 $\pm$ 0.12	0.47 $\pm$ 0.23
Kermut	0.62 $\pm$ 0.25	0.62 $\pm$ 0.24	0.48 $\pm$ 0.22	0.85 $\pm$ 0.13	0.44 $\pm$ 0.22
ProteinNPT	0.57 $\pm$ 0.29	0.59 $\pm$ 0.30	0.43 $\pm$ 0.24	0.83 $\pm$ 0.12	0.42 $\pm$ 0.24
Cerebra-Epistasis	0.65 $\pm$ 0.25	0.65 $\pm$ 0.22	0.50 $\pm$ 0.21	0.87 $\pm$ 0.10	0.50 $\pm$ 0.19

S5.3 Supplementary Results for 1M/2M to 3M+

Following the 1M/2M to 3M+ benchmark setting described in Supplementary Section S2.5, we further evaluated higher-order extrapolation across the full collection of eligible assays.

Overall and order-averaged assay-level performance is summarized in Supplementary Table S7, where Cerebra-Epistasis achieved the best performance on most aggregate metrics. Performance at individual mutation orders is shown in Supplementary Fig S1. Although the performance of all methods

generally decreased with increasing mutation order, Cerebra-Epistasis maintained the strongest overall performance across mutation orders.

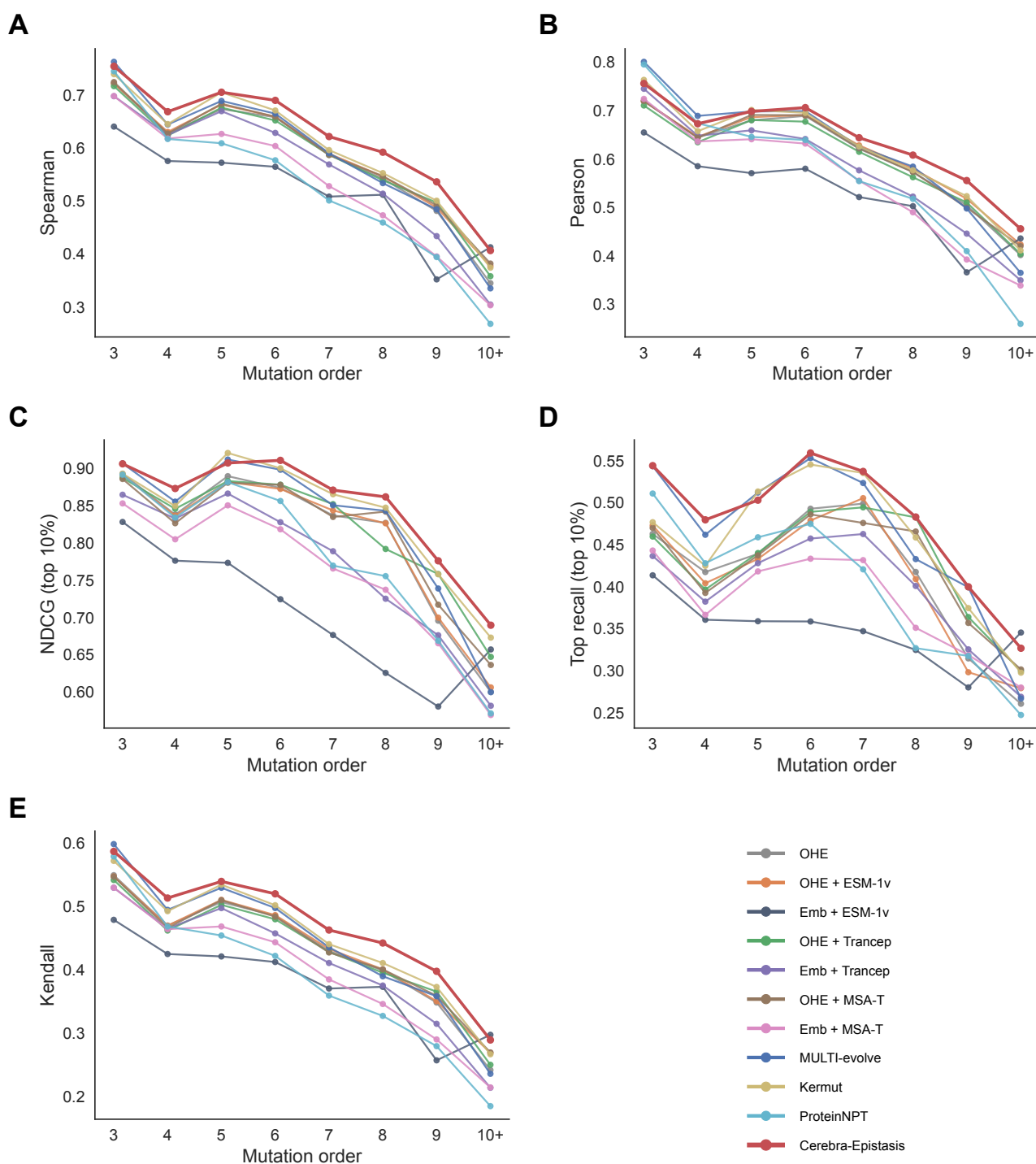


Figure S1. Mutation-order-resolved performance on the 1M/2M to 3M+ benchmark. Models were trained on single and double mutants and evaluated on held-out variants with three or more mutations. Curves show assay-averaged Spearman, Pearson, NDCG (top 10%), top recall (top 10%), and Kendall at each mutation order.

Table S7. Performance on the 1M/2M to 3M+ benchmark under overall and order-averaged assay-level evaluation. Values are reported as mean $\pm$ SD across assays.

Model	Spearman ρ	Pearson r	Kendall τ	NDCG (top 10%)	Top recall (top 10%)
Overall assay-level evaluation					
OHE	0.57 $\pm$ 0.22	0.58 $\pm$ 0.19	0.42 $\pm$ 0.19	0.80 $\pm$ 0.13	0.39 $\pm$ 0.23
OHE + ESM-1v	0.62 $\pm$ 0.20	0.63 $\pm$ 0.18	0.46 $\pm$ 0.17	0.83 $\pm$ 0.14	0.45 $\pm$ 0.23
Embeddings + ESM-1v	0.59 $\pm$ 0.21	0.59 $\pm$ 0.17	0.43 $\pm$ 0.18	0.79 $\pm$ 0.16	0.40 $\pm$ 0.22
OHE + Tranception	0.65 $\pm$ 0.20	0.64 $\pm$ 0.16	0.48 $\pm$ 0.17	0.86 $\pm$ 0.12	0.46 $\pm$ 0.21
Embeddings + Tranception	0.63 $\pm$ 0.21	0.65 $\pm$ 0.14	0.46 $\pm$ 0.18	0.84 $\pm$ 0.12	0.43 $\pm$ 0.20
OHE + MSA Transformer	0.66 $\pm$ 0.20	0.65 $\pm$ 0.16	0.49 $\pm$ 0.16	0.87 $\pm$ 0.12	0.46 $\pm$ 0.19
Embeddings + MSA Transformer	0.61 $\pm$ 0.24	0.61 $\pm$ 0.20	0.46 $\pm$ 0.21	0.79 $\pm$ 0.17	0.41 $\pm$ 0.22
MULTI-evolve	0.62 $\pm$ 0.23	0.68 $\pm$ 0.18	0.47 $\pm$ 0.20	0.84 $\pm$ 0.13	0.47 $\pm$ 0.22
Kermut	0.67 $\pm$ 0.23	0.67 $\pm$ 0.19	0.51 $\pm$ 0.20	0.88 $\pm$ 0.14	0.50 $\pm$ 0.19
ProteinNPT	0.62 $\pm$ 0.24	0.69 $\pm$ 0.21	0.47 $\pm$ 0.22	0.86 $\pm$ 0.11	0.50 $\pm$ 0.20
Cerebra-Epistasis	0.69 $\pm$ 0.21	0.68 $\pm$ 0.17	0.53 $\pm$ 0.19	0.88 $\pm$ 0.11	0.52 $\pm$ 0.18
Order-averaged assay-level evaluation					
OHE	0.59 $\pm$ 0.21	0.60 $\pm$ 0.18	0.44 $\pm$ 0.17	0.82 $\pm$ 0.13	0.43 $\pm$ 0.20
OHE + ESM-1v	0.59 $\pm$ 0.20	0.60 $\pm$ 0.18	0.44 $\pm$ 0.17	0.82 $\pm$ 0.13	0.43 $\pm$ 0.19
Embeddings + ESM-1v	0.55 $\pm$ 0.23	0.55 $\pm$ 0.20	0.41 $\pm$ 0.19	0.74 $\pm$ 0.20	0.40 $\pm$ 0.18
OHE + Tranception	0.59 $\pm$ 0.20	0.59 $\pm$ 0.18	0.44 $\pm$ 0.17	0.82 $\pm$ 0.12	0.43 $\pm$ 0.20
Embeddings + Tranception	0.57 $\pm$ 0.21	0.59 $\pm$ 0.16	0.42 $\pm$ 0.18	0.79 $\pm$ 0.13	0.41 $\pm$ 0.18
OHE + MSA Transformer	0.59 $\pm$ 0.20	0.60 $\pm$ 0.17	0.44 $\pm$ 0.16	0.81 $\pm$ 0.13	0.43 $\pm$ 0.19
Embeddings + MSA Transformer	0.56 $\pm$ 0.23	0.57 $\pm$ 0.19	0.42 $\pm$ 0.20	0.77 $\pm$ 0.15	0.41 $\pm$ 0.19
MULTI-evolve	0.60 $\pm$ 0.22	0.64 $\pm$ 0.18	0.46 $\pm$ 0.19	0.83 $\pm$ 0.13	0.47 $\pm$ 0.21
Kermut	0.60 $\pm$ 0.22	0.61 $\pm$ 0.19	0.45 $\pm$ 0.19	0.83 $\pm$ 0.14	0.44 $\pm$ 0.20
ProteinNPT	0.56 $\pm$ 0.21	0.62 $\pm$ 0.21	0.42 $\pm$ 0.19	0.80 $\pm$ 0.14	0.43 $\pm$ 0.22
Cerebra-Epistasis	0.63 $\pm$ 0.21	0.63 $\pm$ 0.18	0.48 $\pm$ 0.18	0.85 $\pm$ 0.12	0.50 $\pm$ 0.17

S5.4 Supplementary Results for the cDNA proteolysis multi-mutant dataset in the 1M+ to 1M+ benchmark

Results for the cDNA proteolysis multi-mutant benchmark described in Supplementary [Section S2.6](#) are summarized in Supplementary [Table S8](#). Cerebra-Epistasis achieved the best performance on most metrics.

Table S8. Performance on the cDNA proteolysis multi-mutant dataset of the 1M+ to 1M+ benchmark. Results are reported using overall assay-level evaluation and order-averaged assay-level evaluation as mean $\pm$ standard deviation.

Model	Spearman ρ	Pearson r	Kendall τ	NDCG (top 10%)	Top recall (top 10%)
Overall assay-level evaluation					
OHE	0.79 $\pm$ 0.06	0.75 $\pm$ 0.07	0.61 $\pm$ 0.07	0.89 $\pm$ 0.05	0.44 $\pm$ 0.12
OHE + ESM-1v	0.79 $\pm$ 0.08	0.79 $\pm$ 0.08	0.60 $\pm$ 0.08	0.89 $\pm$ 0.04	0.41 $\pm$ 0.11
Embeddings + ESM-1v	0.89 $\pm$ 0.05	0.89 $\pm$ 0.05	0.71 $\pm$ 0.06	0.94 $\pm$ 0.02	0.55 $\pm$ 0.10
OHE + Tranception	0.81 $\pm$ 0.07	0.81 $\pm$ 0.07	0.63 $\pm$ 0.07	0.90 $\pm$ 0.04	0.41 $\pm$ 0.11
Embeddings + Tranception	0.90 $\pm$ 0.05	0.90 $\pm$ 0.05	0.73 $\pm$ 0.06	0.94 $\pm$ 0.02	0.59 $\pm$ 0.09
OHE + MSA Transformer	0.82 $\pm$ 0.07	0.82 $\pm$ 0.07	0.64 $\pm$ 0.07	0.90 $\pm$ 0.04	0.42 $\pm$ 0.12
Embeddings + MSA Transformer	0.92 $\pm$ 0.04	0.92 $\pm$ 0.04	0.76 $\pm$ 0.05	0.95 $\pm$ 0.02	0.62 $\pm$ 0.10
Kermut	0.94 $\pm$ 0.03	0.94 $\pm$ 0.03	0.79 $\pm$ 0.05	0.96 $\pm$ 0.01	0.68 $\pm$ 0.09
ProteinNPT	0.94 $\pm$ 0.03	0.94 $\pm$ 0.03	0.81 $\pm$ 0.05	0.97 $\pm$ 0.02	0.71 $\pm$ 0.08
Cerebra-Epistasis	0.94 $\pm$ 0.03	0.94 $\pm$ 0.04	0.81 $\pm$ 0.05	0.97 $\pm$ 0.01	0.70 $\pm$ 0.08
Order-averaged assay-level evaluation					
OHE	0.76 $\pm$ 0.09	0.73 $\pm$ 0.09	0.58 $\pm$ 0.09	0.89 $\pm$ 0.06	0.49 $\pm$ 0.09
OHE + ESM-1v	0.73 $\pm$ 0.09	0.75 $\pm$ 0.09	0.56 $\pm$ 0.08	0.89 $\pm$ 0.05	0.47 $\pm$ 0.08
Embeddings + ESM-1v	0.83 $\pm$ 0.08	0.85 $\pm$ 0.08	0.65 $\pm$ 0.08	0.93 $\pm$ 0.03	0.56 $\pm$ 0.09
OHE + Tranception	0.76 $\pm$ 0.08	0.78 $\pm$ 0.09	0.59 $\pm$ 0.08	0.90 $\pm$ 0.05	0.49 $\pm$ 0.08
Embeddings + Tranception	0.85 $\pm$ 0.08	0.86 $\pm$ 0.08	0.67 $\pm$ 0.08	0.94 $\pm$ 0.03	0.60 $\pm$ 0.08
OHE + MSA Transformer	0.78 $\pm$ 0.08	0.79 $\pm$ 0.09	0.60 $\pm$ 0.08	0.90 $\pm$ 0.05	0.50 $\pm$ 0.08
Embeddings + MSA Transformer	0.87 $\pm$ 0.07	0.88 $\pm$ 0.07	0.70 $\pm$ 0.08	0.94 $\pm$ 0.03	0.62 $\pm$ 0.08
Kermut	0.89 $\pm$ 0.07	0.91 $\pm$ 0.07	0.74 $\pm$ 0.08	0.96 $\pm$ 0.03	0.68 $\pm$ 0.08
ProteinNPT	0.90 $\pm$ 0.07	0.91 $\pm$ 0.07	0.76 $\pm$ 0.08	0.96 $\pm$ 0.03	0.71 $\pm$ 0.08
Cerebra-Epistasis	0.91 $\pm$ 0.07	0.92 $\pm$ 0.07	0.76 $\pm$ 0.08	0.96 $\pm$ 0.03	0.69 $\pm$ 0.08

S5.5 Supplementary Results for the avGFP benchmark

Results for the avGFP experimental fitness prediction benchmark described in Supplementary [Section S2.7](#) are summarized in Supplementary [Table S9](#). Cerebra-Epistasis achieved the highest Spearman correlation (0.83) and Kendall correlation (0.67) among the compared methods, while attaining competitive performance on the remaining metrics, including an NDCG (top 5) of 0.67 and a TopRecall (top 5) of 0.40.

Table S9. Performance on the avGFP experimental fitness prediction benchmark. The best result for each metric is shown in bold.

Model	Spearman ρ	Pearson r	Kendall τ	NDCG (top 5)	Top recall (top 5)
OHE	0.47	0.40	0.33	0.48	0.20
OHE + ESM-1v	0.56	0.44	0.38	0.37	0.20
Embeddings + ESM-1v	0.51	0.40	0.35	0.42	0.20
OHE + Tranception	0.73	0.52	0.56	0.73	0.60
Embeddings + Tranception	0.66	0.52	0.48	0.43	0.00
OHE + MSA Transformer	0.74	0.57	0.57	0.62	0.40
Embeddings + MSA Transformer	0.70	0.53	0.54	0.50	0.60
MULTI-evolve	0.54	0.22	0.38	0.64	0.40
ProteinNPT	0.73	0.51	0.58	0.62	0.20
Cerebra-Epistasis	0.83	0.55	0.67	0.67	0.40

S5.6 Supplementary Results for Low- N 1M/2M to held-out 2M (epistasis)

Results for the Low- N 1M/2M to held-out 2M (epistasis) benchmark described in Supplementary Section S2.8 are summarized in Supplementary Table S10. Cerebra-Epistasis showed strong performance across both correlation- and ranking-based metrics for predicting epistatic effects of unseen double-mutant combinations.

Table S10. Assay-level performance on the Low- N 1M/2M to held-out 2M (epistasis) benchmark. Values are reported as mean $\pm$ SD across assays. The best result for each metric is shown in bold.

Model	Spearman ρ	Pearson r	Kendall τ	NDCG (top 10%)	Top recall (top 10%)
OHE	0.00 $\pm$ 0.10	0.02 $\pm$ 0.12	0.00 $\pm$ 0.07	0.62 $\pm$ 0.12	0.12 $\pm$ 0.07
OHE + ESM-1v	-0.07 $\pm$ 0.11	-0.06 $\pm$ 0.12	-0.04 $\pm$ 0.07	0.59 $\pm$ 0.14	0.08 $\pm$ 0.06
Embeddings + ESM-1v	0.19 $\pm$ 0.21	0.23 $\pm$ 0.22	0.13 $\pm$ 0.15	0.68 $\pm$ 0.14	0.23 $\pm$ 0.15
OHE + Tranception	-0.05 $\pm$ 0.13	-0.07 $\pm$ 0.15	-0.04 $\pm$ 0.09	0.62 $\pm$ 0.13	0.11 $\pm$ 0.07
Embeddings + Tranception	0.37 $\pm$ 0.26	0.43 $\pm$ 0.28	0.26 $\pm$ 0.19	0.77 $\pm$ 0.13	0.34 $\pm$ 0.22
OHE + MSA Transformer	-0.06 $\pm$ 0.09	-0.09 $\pm$ 0.10	-0.04 $\pm$ 0.07	0.59 $\pm$ 0.13	0.09 $\pm$ 0.06
Embeddings + MSA Transformer	0.23 $\pm$ 0.28	0.24 $\pm$ 0.28	0.16 $\pm$ 0.20	0.68 $\pm$ 0.15	0.20 $\pm$ 0.16
MULTI-evolve	0.19 $\pm$ 0.20	0.29 $\pm$ 0.29	0.13 $\pm$ 0.14	0.78 $\pm$ 0.12	0.33 $\pm$ 0.27
Kermut	0.35 $\pm$ 0.26	0.39 $\pm$ 0.28	0.25 $\pm$ 0.19	0.79 $\pm$ 0.12	0.34 $\pm$ 0.23
Cerebra-Epistasis	0.55 $\pm$ 0.25	0.60 $\pm$ 0.26	0.41 $\pm$ 0.20	0.84 $\pm$ 0.11	0.47 $\pm$ 0.24

S5.7 Effect of end-to-end training on $\Delta\Delta G$ prediction

To assess whether end-to-end training improves downstream fitness prediction, we focused this comparison on protein stability prediction, because large-scale cDNA proteolysis data support training across diverse proteins, while established $\Delta\Delta G$ benchmarks provide standardized external evaluation.

We compared a downstream-only setting, in which the pretrained Cerebra-Seq module was frozen, with an end-to-end setting, in which the Cerebra-Seq module and downstream prediction network were trained end to end. The two model variants were evaluated alongside established $\Delta\Delta G$ prediction methods on the S461 benchmark using the evaluation protocol described in Supplementary [Section S2.9](#). The results are shown in Supplementary [Table S11](#).

Table S11. Comparison of $\Delta\Delta G$ prediction performance on the S461 benchmark. Cerebra-Epistasis* denotes end-to-end training of Cerebra-Seq and the downstream predictor on the complete cDNA proteolysis dataset; Cerebra-Epistasis denotes the corresponding setting in which Cerebra-Seq was frozen and only the downstream predictor was optimized. Values are the mean $\pm$ standard deviation across proteins. Bold values indicate the best mean performance for each metric, including ties.

Model	Spearman ρ	Kendall τ	Pearson r	Top-5 Precision	Top-10 Precision
DDMut	0.48 $\pm$ 0.40	0.37 $\pm$ 0.31	0.52 $\pm$ 0.39	0.45 $\pm$ 0.26	0.61 $\pm$ 0.18
ProteinEBM	0.51 $\pm$ 0.22	0.38 $\pm$ 0.16	0.53 $\pm$ 0.26	0.47 $\pm$ 0.23	0.67 $\pm$ 0.19
PROSTATA	0.53 $\pm$ 0.33	0.40 $\pm$ 0.25	0.55 $\pm$ 0.34	0.53 $\pm$ 0.25	0.67 $\pm$ 0.23
RaSP	0.56 $\pm$ 0.30	0.43 $\pm$ 0.23	0.54 $\pm$ 0.28	0.47 $\pm$ 0.25	0.64 $\pm$ 0.22
SPIRED-Fitness (zero-shot)	0.58 $\pm$ 0.33	0.45 $\pm$ 0.25	0.59 $\pm$ 0.30	0.58 $\pm$ 0.16	0.70 $\pm$ 0.17
GeoDDG-3D	0.59 $\pm$ 0.23	0.44 $\pm$ 0.20	0.59 $\pm$ 0.23	0.63 $\pm$ 0.17	0.72 $\pm$ 0.15
GeoDDG-Seq	0.60 $\pm$ 0.23	0.46 $\pm$ 0.18	0.63 $\pm$ 0.20	0.55 $\pm$ 0.19	0.72 $\pm$ 0.17
Mutate Everything (ESM-2)	0.63 $\pm$ 0.19	0.46 $\pm$ 0.16	0.62 $\pm$ 0.18	0.45 $\pm$ 0.24	0.69 $\pm$ 0.18
ThermoMPNN	0.63 $\pm$ 0.32	0.48 $\pm$ 0.26	0.60 $\pm$ 0.31	0.48 $\pm$ 0.23	0.67 $\pm$ 0.21
Pythia	0.64 $\pm$ 0.19	0.48 $\pm$ 0.16	0.63 $\pm$ 0.18	0.57 $\pm$ 0.22	0.68 $\pm$ 0.17
SPURS	0.69 $\pm$ 0.25	0.55 $\pm$ 0.23	0.70 $\pm$ 0.23	0.62 $\pm$ 0.26	0.73 $\pm$ 0.14
SPIRED-Stab	0.71 $\pm$ 0.24	0.57 $\pm$ 0.21	0.69 $\pm$ 0.21	0.62 $\pm$ 0.22	0.68 $\pm$ 0.24
Cerebra-Epistasis	0.61 $\pm$ 0.32	0.48 $\pm$ 0.28	0.66 $\pm$ 0.30	0.52 $\pm$ 0.26	0.67 $\pm$ 0.21
Cerebra-Epistasis*	0.71 $\pm$ 0.24	0.57 $\pm$ 0.20	0.71 $\pm$ 0.22	0.57 $\pm$ 0.14	0.74 $\pm$ 0.17

S6 Ablation and sensitivity analyses

We performed a series of ablation and sensitivity analyses to examine two key design principles underlying Cerebra-Epistasis: structure-aware representation learning and explicit modeling of epistatic interactions across mutation orders. We first assessed how Cerebra-Seq-derived structure-aware features contribute to higher-order extrapolation and whether explicit geometric information helps recover the spatial organization of epistasis. We then examined the contribution of explicit epistasis modeling and investigated how the mutation orders and pairwise interaction coverage represented during training affect generalization to unseen higher-order combinations. Finally, we evaluated whether end-to-end optimization of the structure and fitness prediction modules further improves mutation-effect prediction.

S6.1 Contribution of Cerebra-Seq-derived structure-aware features to higher-order extrapolation

To assess whether structure-aware information improves higher-order extrapolation beyond sequence representations alone, we progressively augmented the ESM-2-derived features with three-dimensional coordinates, Cerebra residue-level node representations, and pairwise edge representations. This progression introduces increasingly rich geometric and structural context, from explicit residue coordinates to learned residue- and pair-level representations derived by Cerebra. On the 1M/2M to 3M+ benchmark, progressively incorporating these features generally improved both overall performance and performance across mutation orders (Supplementary Fig S2). These results indicate that Cerebra-Seq-derived structural information provides complementary signals beyond sequence embeddings for predicting unseen higher-order mutant combinations.

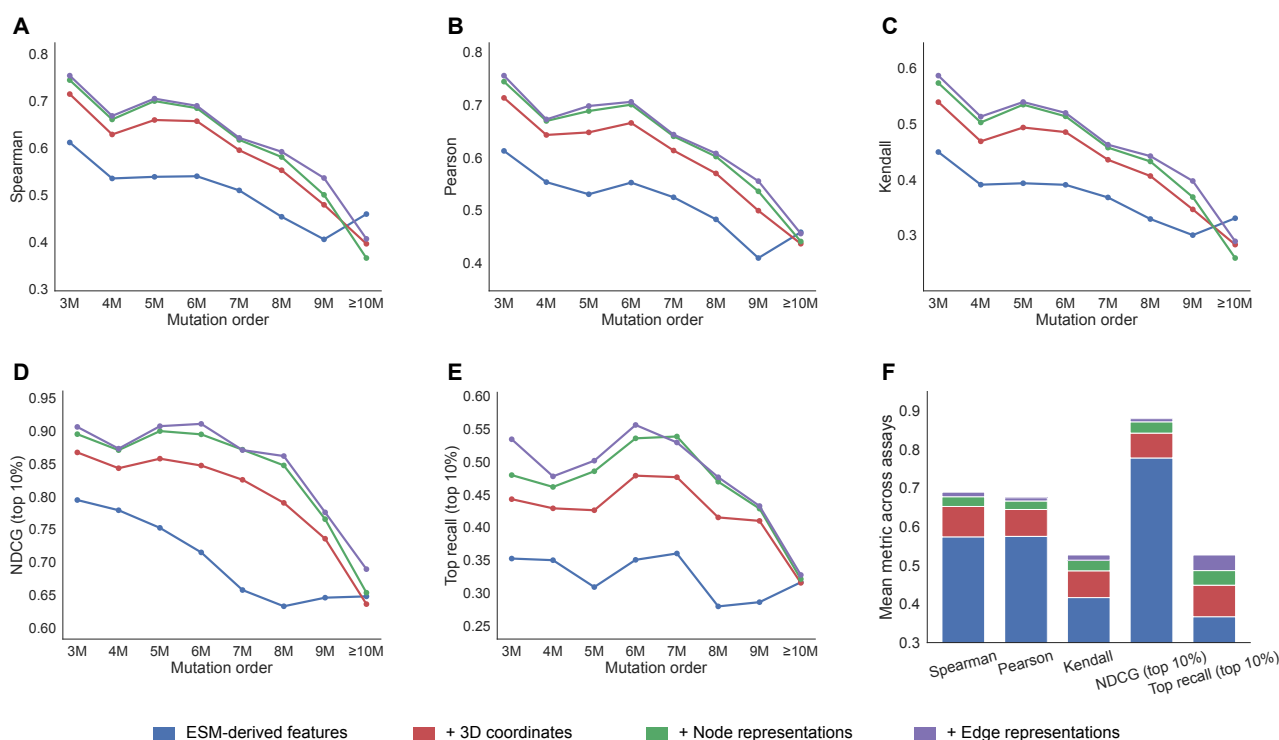


Figure S2. Contribution of Cerebra-Seq-derived structure-aware features to higher-order extrapolation. Models were trained on single- and double-mutant variants and evaluated on held-out variants containing three or more mutations. Starting from ESM-2-derived sequence features, three-dimensional coordinates, Cerebra residue-level node representations, and pairwise edge representations were progressively incorporated. **A–E**, Spearman, Pearson, Kendall, NDCG (top 10%), and Top recall (top 10%) across mutation orders. **F**, Assay-averaged performance across the five evaluation metrics, summarizing the contribution of progressively added structure-aware features.

S6.2 Contribution of explicit epistasis modeling to higher-order extrapolation

To determine whether higher-order extrapolation in the 1M/2M to 3M+ benchmark benefits from explicit epistasis modeling, we compared the complete Cerebra-Epistasis model with an additive-only model. Both models were trained on the same single- and double-mutant data and evaluated on the same held-out 3M+ variants. The additive-only model predicted multi-mutant fitness as the sum of its constituent single-mutation contributions, whereas the complete model combined these additive contributions with an explicit epistatic term. The complete model outperformed the additive-only model in overall performance and generally maintained stronger performance across mutation orders

(Supplementary Fig S3), demonstrating that modeling non-additive interactions explicitly is important for transferring epistatic information from lower-order measurements to unseen higher-order mutant combinations.

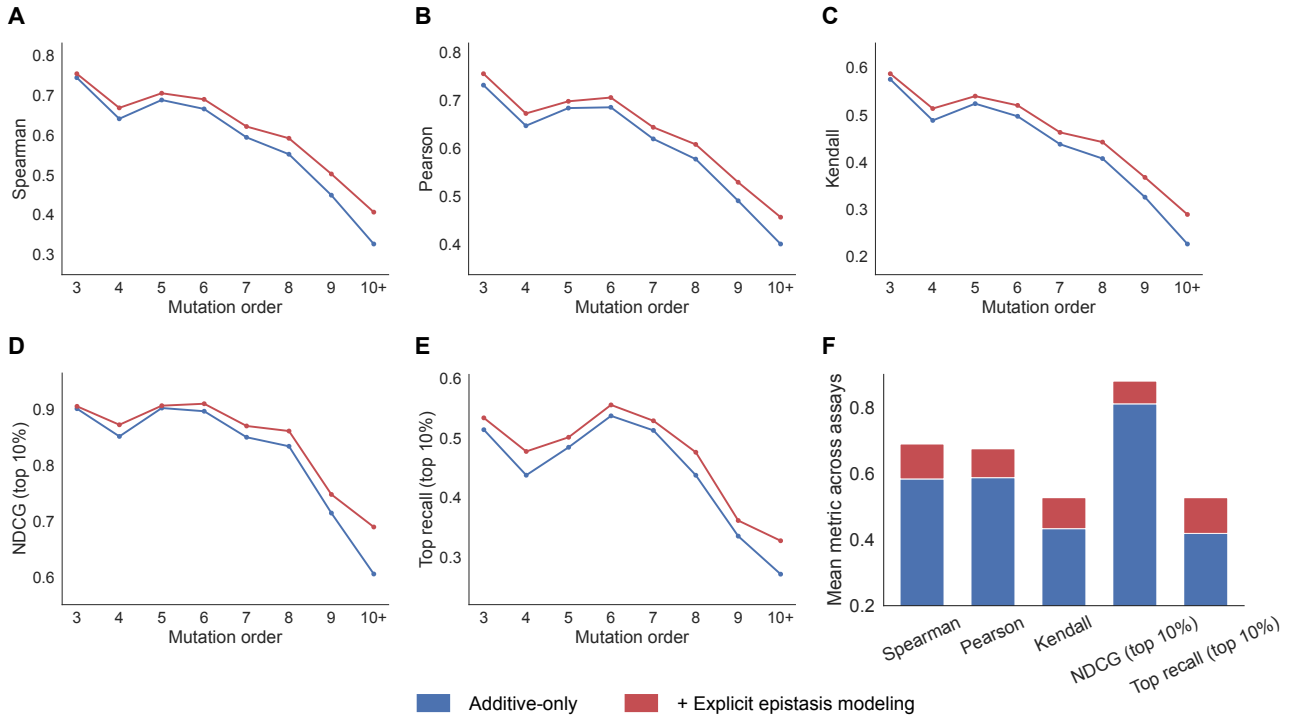


Figure S3. Effect of explicit epistasis modeling on higher-order fitness extrapolation. Comparison of Cerebra-Epistasis and the additive-only model on the 1M/2M to 3M+ benchmark. **A–E**, Assay-averaged Spearman, Pearson, Kendall, NDCG (top 10%), and top recall (top 10%) at each mutation order. **F**, Overall assay-level performance across held-out higher-order variants, with the incremental gain over the additive-only model indicating the contribution of explicit epistasis modeling.

S6.3 Effect of maximum training mutation order on higher-order extrapolation

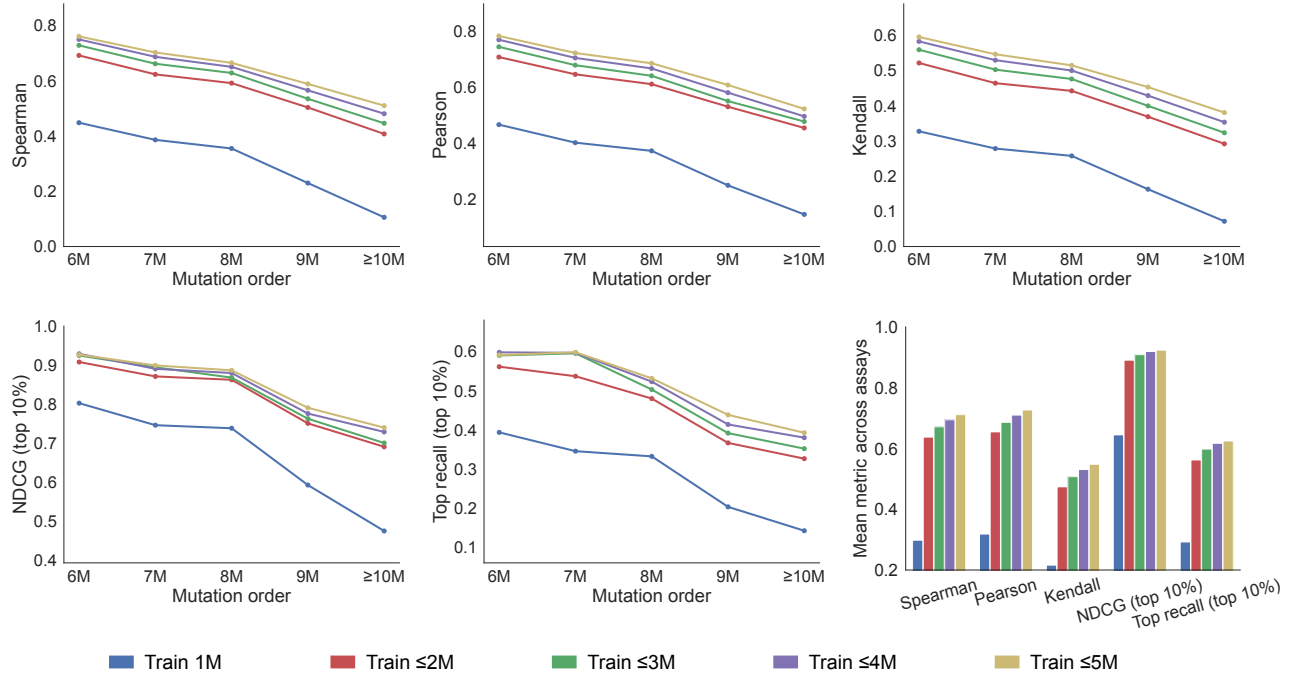
To examine how the maximum mutation order included in the training data affects extrapolation to more complex variants, models were trained on progressively expanded mutation-order ranges and evaluated on the same higher-order test variants (Supplementary Fig S4A). Adding double mutants to the single-mutant training set produced a marked improvement in higher-order prediction, indicating that pairwise measurements provide important information about non-additive interactions that is difficult to infer from single-mutant effects alone. Including variants with progressively higher mutation orders yielded further improvements, although these gains were generally smaller than the improvement obtained from introducing double mutants. These results suggest that pairwise epistasis provides much of the interaction information needed for higher-order extrapolation, whereas higher-order training examples contribute additional information about more complex mutational dependencies.

S6.4 Effect of double-mutant coverage on higher-order extrapolation

To investigate how extensively pairwise interactions must be sampled before reliable higher-order extrapolation becomes possible, the single-mutant training set and higher-order test set were held fixed, while the proportion of available double mutants used for training was varied (Supplementary Fig S4B). Prediction performance improved as double-mutant coverage increased, confirming that direct observations of pairwise epistasis provide transferable information for predicting unseen higher-order

combinations. Notably, substantial gains were already obtained from partial double-mutant coverage, suggesting that the model can infer useful interaction patterns without exhaustively measuring the complete pairwise landscape. The progressively smaller improvements at higher coverage further indicate diminishing returns, with additional double-mutant measurements mainly refining an interaction representation that is already partially established from a limited subset of pairwise observations.

A Maximum mutation order included in training



B Double-mutant coverage in training

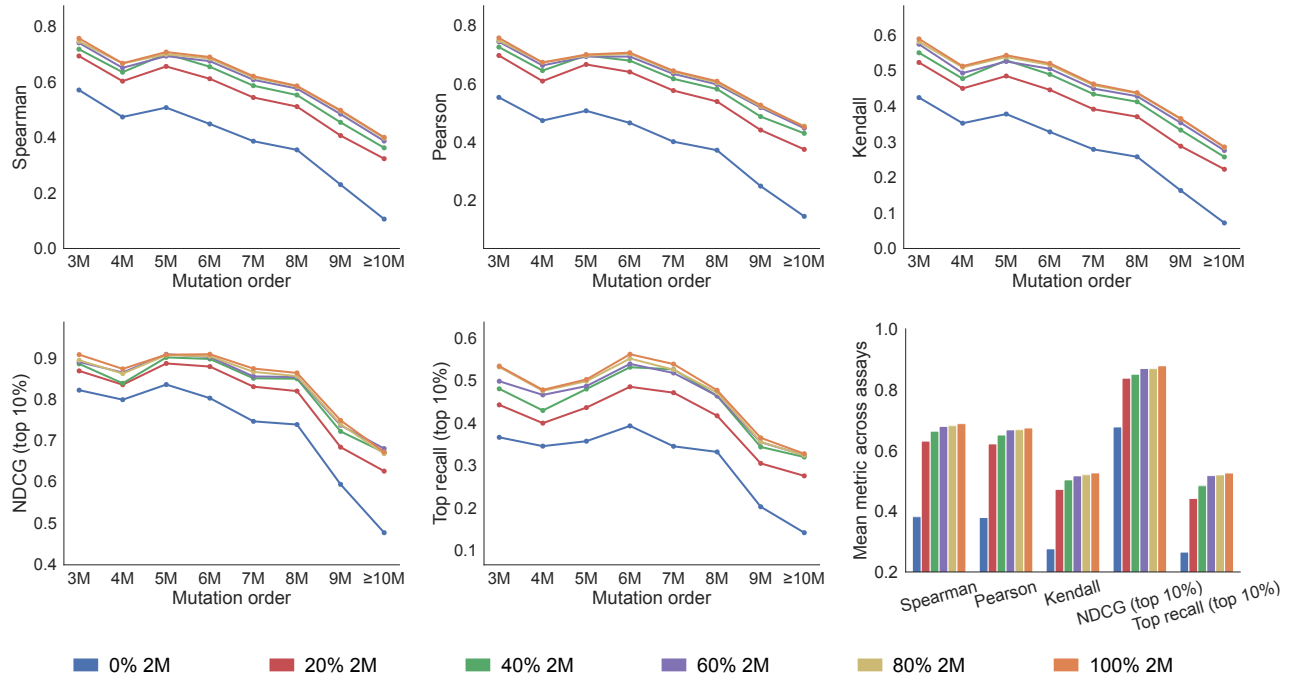


Figure S4. Effects of training mutation order and double-mutant coverage on higher-order extrapolation. **A**, Performance of models trained with progressively increasing maximum mutation orders. **B**, Performance of models trained with different proportions of the available double mutants. For each ablation, the five mutation-order plots show assay-averaged Spearman, Pearson, Kendall, NDCG (top 10%), and top recall (top 10%), while the final plot shows overall assay-level performance across held-out 3M+ variants.

1143 **S6.5 Effect of end-to-end training on fitness and structure prediction**

1144 To distinguish the contributions of fitness gradients and structural supervision, we compared four
1145 settings on identical sequence-grouped five-fold partitions of the cDNA proteolysis multi-mutant
1146 dataset. Fitness gradients updated the downstream components in every setting, with Cerebra-Seq
1147 kept fixed in the frozen baseline. In structure-only training, only structural gradients updated Cerebra-
1148 Seq, while fitness gradients were restricted to the downstream components. Fitness-only training
1149 propagated fitness gradients through both modules without structural supervision. End-to-end training
1150 (Structure + Fitness) allowed both fitness and structural gradients to update Cerebra-Seq. These
1151 comparisons also tested whether adding structural supervision improves fitness prediction when fitness
1152 gradients already update the upstream model. Across epochs 1–100, all three update settings improved
1153 trajectory-averaged Spearman correlations over frozen Cerebra-Seq. End-to-end training achieved the
1154 highest correlation, exceeding structure-only training and modestly improving on fitness-only training
1155 (Supplementary Fig S5A,B). Fitness-only training progressively increased side-chain χ torsion-angle
1156 loss and reduced C_α IDDT. End-to-end training instead reduced torsion-angle loss below the pretrained
1157 baseline and largely preserved IDDT (Supplementary Fig S5C,D). Its trajectory-averaged IDDT
1158 remained slightly below that of structure-only training. These results indicate that fitness gradients
1159 improve prediction beyond structural supervision alone, while additional structural supervision further
1160 improves prediction and helps preserve structural accuracy.

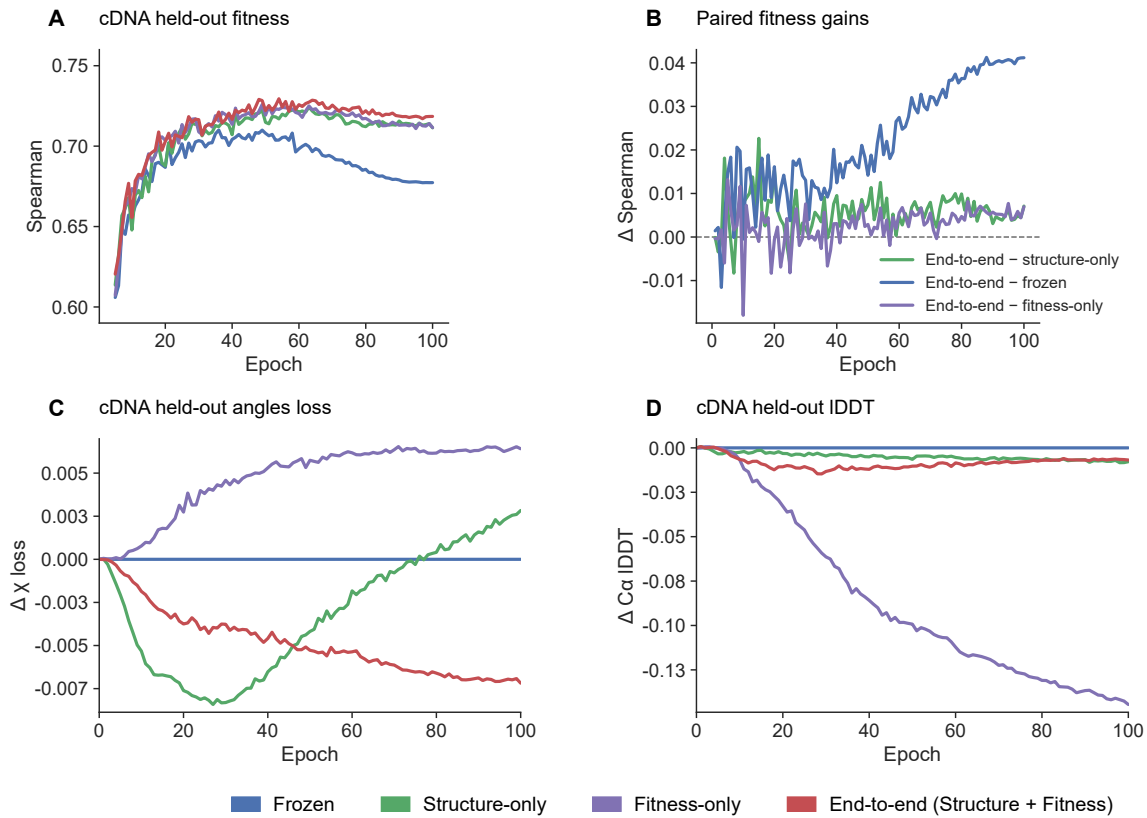


Figure S5. Contributions of fitness gradients and structural supervision to end-to-end training. Models used identical sequence-grouped five-fold partitions of the cDNA proteolysis multi-mutant dataset, with fitness gradients updating the downstream components in every setting. In **A**, **C** and **D**, blue denotes frozen Cerebra-Seq; green, structure-only training; purple, fitness-only training; and red, end-to-end training (Structure + Fitness). Only structural gradients updated Cerebra-Seq in structure-only training, only fitness gradients in fitness-only training, and both in end-to-end training. **A**, Spearman correlations for held-out proteins. **B**, Paired gains in Spearman correlation from end-to-end training over structure-only training (green), frozen Cerebra-Seq (blue), and fitness-only training (purple). Differences were calculated within the same fold and epoch before averaging. **C**, Changes in side-chain χ torsion-angle loss from the pretrained baseline; negative values indicate improvement. **D**, Changes in C_{α} IDDT from the pretrained baseline; positive values indicate improvement. Zero denotes the fold-specific pretrained baseline in **C** and **D**. Lines show equal-weight means across five folds.

S7 Cerebra-Epistasis Inference

This section presents pseudocode for the complete inference workflow of Cerebra-Epistasis and its main components. Supplementary **Algorithm S1** first provides a high-level overview of the framework, encompassing one-time wild-type encoding by Cerebra-Seq, structure-aware mutation-atlas construction, and fitness and epistasis prediction for arbitrary-order mutants. The subsequent algorithms describe the Cerebra-Seq structure predictor, the downstream fitness prediction modules, and the core SE(3)-equivariant operations in detail.

S7.1 Features and Notation

Supplementary **Table S12** summarizes the principal features and notation shared across the algorithms. Shapes of short-lived intermediate variables are given locally in the rightmost column of each algorithm.

Table S12. Features and notation used in the pseudocode algorithms.

Feature	Description	Shape
L	Number of residues in the wild-type protein	
K	Number of anchor-conditioned structure predictions	
B	Number of candidate mutants evaluated together	
m	Number of substitutions in an individual mutant	
r_H	Dimension of the learned epistasis representation	
wt_seq	Wild-type amino-acid sequence	$[L]$
$\mathbf{X}^{\text{ESM-2}}$	Per-residue ESM-2 representation	$[L, 1280]$
$\mathbf{X}^{\text{ESMC}}$	Per-residue ESMC representation	$[L, 1152]$
$\mathbf{X}^{\text{ESM3}}$	Per-residue ESM3 representation	$[L, 1536]$
m	Cerebra-Seq single representation arranged over ten streams	$[10, L, 256]$
z	Cerebra-Seq residue-pair representation	$[L, L, 128]$
T	Anchor-conditioned residue translations	$[K, L, 3]$
Q	Anchor-conditioned residue orientations represented as quaternions	$[K, L, 4]$
N	Cerebra-Seq residue-level structural representation	$[L, 256]$
E	Cerebra-Seq residue-pair structural representation	$[L, L, 128]$
$\mathcal{F}$	Collection of degree-wise SE(3) features, $\{\mathbf{F}^{(0)}, \mathbf{F}^{(1)}\}$	$[L, C_0, 1]; [L, C_1, 3]$
$\tilde{\mathbf{E}}$	Pair representation projected to the SE(3) edge channels	$[L, L, 32]$
Z	Degree-0 invariant output representation of the SE(3) encoder	$[L, 128]$
V	Degree-1 equivariant output representation of the SE(3) encoder	$[L, 32, 3]$
D^{stat}	Mean and standard deviation of the selected neighborhood distances	$[L, 2]$
S	WT-centered single-mutation score tensor	$[L, 20]$
U	WT-centered epistasis representation tensor	$[L, 20, r_H]$
$\mathcal{M}$	Individual mutant represented as $\{(i_k, a_k)\}_{k=1}^m$	$[m, 2]$
$\mathcal{Q}$	Collection of candidate mutants, $\{\mathcal{M}_b\}_{b=1}^B$	$[B]$
$\hat{y}_{\mathcal{M}}$	Predicted fitness of mutant $\mathcal{M}$	
$\hat{\epsilon}_{\mathcal{M}}$	Predicted total epistatic contribution of mutant $\mathcal{M}$	

S7.2 Overall Cerebra-Epistasis workflow

Given a wild-type sequence and a collection of candidate mutants, Cerebra-Epistasis first runs Cerebra-Seq once to obtain atom14 coordinates together with residue-level node and residue-pair edge representations. The downstream encoder then converts these shared wild-type features into the single-mutation score tensor **S** and epistasis representation tensor **U**. Each candidate mutant is subsequently evaluated by lightweight indexing and permutation-invariant assembly, without repeating sequence or structure encoding.

Algorithm S1 Complete Cerebra-Epistasis workflow

def Cerebra_Epistasis_Inference($\mathbf{wt_seq} \in \mathcal{A}^L$, $\mathbf{target_feat}$, $\mathbf{residue_index}$, AnchorList, $\mathcal{Q} = \{\mathcal{M}_b\}_{b=1}^B$):

Encode the wild-type protein sequence with ESM models once

1: $\mathbf{X}^{\text{ESM-2}} \leftarrow (\text{ESM-2}(\mathbf{wt_seq}))$ $\mathbf{X}^{\text{ESM-2}} \in \mathbb{R}^{L \times 1280}$
2: $\mathbf{X}^{\text{ESMC}} \leftarrow (\text{ESMC}(\mathbf{wt_seq}))$ $\mathbf{X}^{\text{ESMC}} \in \mathbb{R}^{L \times 1152}$
3: $\mathbf{X}^{\text{ESM3}} \leftarrow (\text{ESM3}(\mathbf{wt_seq}))$ $\mathbf{X}^{\text{ESM3}} \in \mathbb{R}^{L \times 1536}$

Predict the wild-type structure and extract structural representations once with Cerebra-Seq

4: $\mathbf{atom_coords}, \mathbf{N}, \mathbf{E} \leftarrow \text{Cerebra_Seq}(\mathbf{X}^{\text{ESMC}}, \mathbf{X}^{\text{ESM3}}, \mathbf{target_feat}, \mathbf{residue_index}, \text{AnchorList})$ $\mathbf{atom_coords} \in \mathbb{R}^{L \times 14 \times 3}$
 $\mathbf{N} \in \mathbb{R}^{L \times 256}$; $\mathbf{E} \in \mathbb{R}^{L \times L \times 128}$

Construct the shared structure-aware mutation atlas once

5: $\mathbf{Z}, \mathbf{V}, \mathbf{D}^{\text{stat}} \leftarrow \text{SE3_Transformer_Encoder}(\mathbf{N}, \mathbf{E}, \mathbf{atom_coords})$ $\mathbf{Z} \in \mathbb{R}^{L \times 128}$; $\mathbf{V} \in \mathbb{R}^{L \times 32 \times 3}$
 $\mathbf{D}^{\text{stat}} \in \mathbb{R}^{L \times 2}$
6: $\mathbf{S}, \mathbf{U} \leftarrow \text{Mutation_Landscape_Head}(\mathbf{Z}, \mathbf{V}, \mathbf{D}^{\text{stat}}, \mathbf{X}^{\text{ESM-2}}, \mathbf{wt_seq})$ $\mathbf{S} \in \mathbb{R}^{L \times 20}$; $\mathbf{U} \in \mathbb{R}^{L \times 20 \times r_H}$

Reuse the mutation atlas to predict all candidate mutants

7: **for** $b = 1, \dots, B$ **do**
8: $\hat{\mathbf{y}}_b, \hat{\mathbf{e}}_b \leftarrow \text{Assembly_Module}(\mathbf{S}, \mathbf{U}, \mathcal{M}_b; f_\theta)$ $\hat{\mathbf{y}}_b, \hat{\mathbf{e}}_b \in \mathbb{R}$
9: **end for**
10: **return** $\mathbf{S}, \mathbf{U}, \{\hat{\mathbf{y}}_b, \hat{\mathbf{e}}_b\}_{b=1}^B$

S7.3 Cerebra-Seq structure prediction

Supplementary **Algorithm S2** and Supplementary **Algorithm S3** summarize the Cerebra-Seq forward pass using the module-oriented notation of the original Cerebra description [18]. Supplementary **Algorithm S2** presents inference for one Cerebra-Seq checkpoint. Frozen ESMC and ESM3 representations are extracted once and reused across four prediction cycles, consisting of an initial pass and three recycling updates with shared network weights. In each cycle, the single and pair representations are initialized together with the recycled states, refined by the 32-block Evoformer Module, and passed through four sequential Cerebra StructureGenerationMotifs. Each motif retains the original Cerebra ordering of the 1D encoder, structure encoder, 1D decoder, structure decoder, and PSA operations. After the final cycle, the anchor-conditioned residue frames are aligned and combined, and the resulting frames and seven torsion-angle representations are converted to atom14 coordinates. The next algorithm isolates the ESMC–ESM3 fusion and mHC-inspired mixing used in the input pathway. Unless otherwise specified, modules not detailed here retain the architectures described in Cerebra [18].

Algorithm S2 Cerebra-Seq model inference

```
def Cerebra_Seq( $\mathbf{X}^{\text{ESMC}} \in \mathbb{R}^{L \times 1152}$ ,  $\mathbf{X}^{\text{ESM3}} \in \mathbb{R}^{L \times 1536}$ ,  $\text{target\_feat} \in \{1, \dots, 22\}^L$ ,  $\text{residue\_index} \in \mathbb{Z}^L$ ,  
AnchorList  $\in \{1, \dots, L\}^K$ ,  $N_{\text{cycle}} = 4$ ):  
# Initialize the recycling state  
1:  $\mathbf{m}^{\text{prev}}, \mathbf{z}^{\text{prev}}, \mathbf{D}^{\text{prev}} \leftarrow \emptyset, \emptyset, \emptyset$   
# Initial prediction followed by recycling updates; weights are shared across cycles  
2: for  $c = 1, \dots, N_{\text{cycle}}$ :  
3:    $\mathbf{m}, \mathbf{z} \leftarrow \text{MHC\_Head}(\mathbf{X}^{\text{ESMC}}, \mathbf{X}^{\text{ESM3}}, \text{target\_feat}, \text{residue\_index},$   $\mathbf{m} \in \mathbb{R}^{10 \times L \times 256}; \mathbf{z} \in \mathbb{R}^{L \times L \times 128}$   
       $(\mathbf{m}^{\text{prev}}, \mathbf{z}^{\text{prev}}, \mathbf{D}^{\text{prev}}))$   
4:    $\mathbf{m}, \mathbf{z} \leftarrow \text{EvoformerStack}(\mathbf{m}, \mathbf{z})$   
5:    $\mathbf{N}_{\text{raw}}, \mathbf{E}, \{\mathbf{T}, \mathbf{Q}\} \leftarrow \text{CerebraStructureStack}(\mathbf{m}, \mathbf{z}, \text{AnchorList})$   $\mathbf{N}_{\text{raw}} \in \mathbb{R}^{10 \times L \times 256}; \mathbf{E} \in \mathbb{R}^{L \times L \times 128}$   
6:    $\mathbf{N} \leftarrow \text{Linear}(\mathbf{N}_{\text{raw}})$   $\mathbf{N} \in \mathbb{R}^{L \times 256}$   
7:    $\alpha \leftarrow \text{AngleResNet}(\mathbf{N})$   $\alpha \in \mathbb{R}^{L \times 7 \times 2}$   
8:    $\mathbf{P}^{\text{pLDDT}} \leftarrow \text{Plddt\_Head}(\mathbf{N}, \mathbf{E})$   $\mathbf{P}^{\text{pLDDT}} \in \mathbb{R}^{K \times L}$   
9:    $D_{kij} \leftarrow \|\mathbf{T}_{ki} - \mathbf{T}_{kj}\|_2$   $\mathbf{D} \in \mathbb{R}^{K \times L \times L}$   
10:   $\mathbf{m}^{\text{prev}}, \mathbf{z}^{\text{prev}}, \mathbf{D}^{\text{prev}} \leftarrow \mathbf{m}, \mathbf{z}, \mathbf{D}$   
11: end for  
# Combine anchor-conditioned frames and construct the side-chain-resolved structure  
12:  $\bar{\mathbf{T}}, \bar{\mathbf{Q}} \leftarrow \text{AnchorFrameConsensus}(\mathbf{T}, \mathbf{Q}, \mathbf{P}^{\text{pLDDT}}, \text{AnchorList})$   $\bar{\mathbf{T}} \in \mathbb{R}^{L \times 3}; \bar{\mathbf{Q}} \in \mathbb{R}^{L \times 4}$   
13:  $\text{atom\_coords} \leftarrow$   $\text{atom\_coords} \in \mathbb{R}^{L \times 14 \times 3}$   
    $\text{computeAllAtomCoordinates}(\bar{\mathbf{T}}, \bar{\mathbf{Q}}, \alpha, \text{target\_feat})$   
14: return  $\text{atom\_coords}, \mathbf{N}, \mathbf{E}$ 
```

1191 Supplementary **Algorithm S3** details the component that distinguishes the Cerebra-Seq input
1192 pathway from the MSA-based Cerebra Embedding Module. ESMC and ESM3 representations are
1193 concatenated residue-wise, projected to 2,560 channels, and partitioned into ten streams of width
1194 256. A shared learnable logit matrix is converted in log space to a doubly stochastic matrix by 20
1195 Sinkhorn iterations and then mixes the ten streams. The mixed language-model representation is
1196 combined with the target-residue embedding, while the 128-channel pair representation is initialized
1197 from target-residue and relative-position embeddings. From the second prediction cycle onward, the
1198 normalized single and pair states and the binned structural distances from the preceding cycle are added
1199 to these initial representations.

Algorithm S3 MHCMixingEmbedder Head

```
def MHC_Head( $\mathbf{X}^{\text{ESMC}} \in \mathbb{R}^{L \times 1152}$ ,  $\mathbf{X}^{\text{ESM3}} \in \mathbb{R}^{L \times 1536}$ , target_feat, residue_index, prevs = ( $\mathbf{m}^{\text{prev}}$ ,  $\mathbf{z}^{\text{prev}}$ ,  $\mathbf{D}^{\text{prev}}$ )):
    # Fuse the ESMC and ESM3 representations
    1:  $\mathbf{m} \leftarrow \text{Cat}(\mathbf{X}^{\text{ESMC}}, \mathbf{X}^{\text{ESM3}})$   $\mathbf{m} \in \mathbb{R}^{L \times 2688}$ 
    2:  $\mathbf{m} \leftarrow \text{Reshape}_{L \times 10 \times 256}(\text{LayerNorm}(\text{Linear}(\mathbf{m})))$   $\mathbf{m} \in \mathbb{R}^{L \times 10 \times 256}$ 
    # Construct an mHC-constrained operator for stable exchange across streams
    3: Construct the learnable cross-stream residual mapping  $\tilde{\mathbf{H}}$   $\tilde{\mathbf{H}} \in \mathbb{R}^{10 \times 10}$ 
    4:  $\mathbf{H} \leftarrow \text{SinkhornKnopp}(\tilde{\mathbf{H}})$   $\mathbf{H} \in \mathbb{R}^{10 \times 10}$ ; doubly stochastic
    5:  $\tilde{\mathbf{m}}_l \leftarrow \mathbf{H} \mathbf{m}_l$ ,  $l = 1, \dots, L$   $\tilde{\mathbf{m}} \in \mathbb{R}^{L \times 10 \times 256}$ 
    # Initialize single and pair representations
    6:  $\mathbf{m} \leftarrow \text{Permute}(\tilde{\mathbf{m}}) + \text{Reshape}_{10 \times L \times 256}(\text{Linear}(\mathbf{target\_feat}))$   $\mathbf{m} \in \mathbb{R}^{10 \times L \times 256}$ 
    7:  $\mathbf{z}_{ij} \leftarrow \text{RelPos}(\mathbf{residue\_index}_i - \mathbf{residue\_index}_j) +$   $\mathbf{z} \in \mathbb{R}^{L \times L \times 128}$ 
        $\text{Linear}(\mathbf{target\_feat}_i) + \text{Linear}(\mathbf{target\_feat}_j)$ 
    # Add recycled representations when available
    8: if prevs  $\neq (\emptyset, \emptyset, \emptyset)$ :
    9:    $\mathbf{m} \leftarrow \mathbf{m} + \text{LayerNorm}(\mathbf{m}^{\text{prev}})$ 
    10:   $\mathbf{z} \leftarrow \mathbf{z} + \text{LayerNorm}(\mathbf{z}^{\text{prev}}) + \text{Linear}(\text{Distgram}(\mathbf{D}^{\text{prev}}))$ 
    11: return  $\mathbf{m}, \mathbf{z}$ 
```

1200 Supplementary **Algorithm S4** summarizes how the anchor-conditioned predictions are combined.
1201 The prediction associated with the middle anchor is used as the reference. Each remaining prediction is
1202 rotated into this reference coordinate system. For every residue, the five aligned predictions with the
1203 highest pLDDT scores are then averaged to obtain one complete set of residue frames.

Algorithm S4 Consensus of anchor-conditioned residue frames

def AnchorFrameConsensus($\mathbf{T}, \mathbf{Q}, \mathbf{P}^{\text{LDDT}}, \text{AnchorList}$):

Use the middle anchor-conditioned prediction as the reference

1: $K \leftarrow |\text{AnchorList}|, \quad r \leftarrow \lfloor K/2 \rfloor + 1 \quad r \in \{1, \dots, K\}$
2: $\mathbf{T}_r^c \leftarrow \text{Center}(\mathbf{T}_r) \quad \mathbf{T}_r^c \in \mathbb{R}^{L \times 3}$

Place all anchor-conditioned residue frames in the reference coordinate system

3: **for** $k = 1, \dots, K$ **do**
4: $\mathbf{T}_k^c \leftarrow \text{Center}(\mathbf{T}_k)$
5: $\mathbf{R}_k \leftarrow \text{Kabsch}(\mathbf{T}_k^c, \mathbf{T}_r^c)$
6: $\mathbf{q}_k^{\text{align}} \leftarrow \text{RotationToQuaternion}(\mathbf{R}_k) \quad \mathbf{q}_k^{\text{align}} \in \mathbb{R}^4$
7: $\hat{\mathbf{T}}_{ki} \leftarrow \mathbf{T}_{ki}^c \mathbf{R}_k, \quad \hat{\mathbf{Q}}_{ki} \leftarrow \mathbf{q}_k^{\text{align}} \otimes \mathbf{Q}_{ki}, \quad i = 1, \dots, L$
8: **end for**

Select the five most confident predictions and obtain one complete residue-frame set

9: **for** $i = 1, \dots, L$ **do**
10: $\mathcal{I}_i \leftarrow \text{TopFive}_k(P_{ki}^{\text{LDDT}}) \quad |\mathcal{I}_i| = 5$
11: $\bar{\mathbf{T}}_i \leftarrow \frac{1}{5} \sum_{k \in \mathcal{I}_i} \hat{\mathbf{T}}_{ki} \quad \bar{\mathbf{T}} \in \mathbb{R}^{L \times 3}$
12: $\bar{\mathbf{Q}}_i \leftarrow \text{Normalize}\left(\frac{1}{5} \sum_{k \in \mathcal{I}_i} \hat{\mathbf{Q}}_{ki}\right) \quad \bar{\mathbf{Q}} \in \mathbb{R}^{L \times 4}$
13: **end for**
14: **return** $\bar{\mathbf{T}}, \bar{\mathbf{Q}}$

S7.4 Downstream fitness prediction

The following algorithms expand the downstream part of Supplementary [Algorithm S1](#). They describe the one-time encoding of the Cerebra-Seq representations, construction of the mutation atlas, and assembly of arbitrary-order mutant fitness and epistasis predictions.

S7.4.1 Downstream model workflow

Following the complete workflow in Supplementary [Algorithm S1](#), the downstream model integrates the Cerebra-Seq residue, pairwise, and coordinate features of the wild-type protein using the SE(3)-Transformer encoder. The encoder integrates sequence and structural information into a degree-0 invariant representation $\mathbf{Z}$ and a degree-1 equivariant representation $\mathbf{V}$, while also summarizing the selected neighborhood distances as local distance statistics $\mathbf{D}^{\text{stat}}$. These outputs, together with the ESM-2 650M embedding, are used to construct the single-mutation score tensor $\mathbf{S} \in \mathbb{R}^{L \times 20}$ and the epistasis representation tensor $\mathbf{U} \in \mathbb{R}^{L \times 20 \times r_H}$. The entries corresponding to an arbitrary-order mutant are then indexed and assembled to predict its fitness and epistatic component.

S7.4.2 SE(3)-Transformer encoder

Supplementary [Algorithm S5](#) describes the SE(3)-Transformer encoder used to integrate the Cerebra-Seq-derived residue, pairwise, and coordinate features of the wild-type protein. The residue features are

1220 projected into degree-0 invariant features, the atom coordinates initialize degree-1 equivariant features,
 1221 and the pairwise features are projected into edge features. Geometric and pairwise-representation
 1222 neighborhoods are then constructed to define the residue graph. The resulting features are processed
 1223 by an input equivariant convolution, a single SE(3)-Transformer block, and an output equivariant
 1224 convolution, yielding the degree-0 invariant representation $\mathbf{Z} \in \mathbb{R}^{L \times 128}$ and the degree-1 equivariant
 1225 representation $\mathbf{V} \in \mathbb{R}^{L \times 32 \times 3}$. The distances within the selected neighborhoods are additionally
 1226 summarized by their residue-wise mean and standard deviation, producing the local distance statistics
 1227 $\mathbf{D}^{\text{stat}} \in \mathbb{R}^{L \times 2}$.

Algorithm S5 SE(3)-Transformer encoder

```

def SE3_Transformer_Encoder( $\mathbf{N} \in \mathbb{R}^{L \times 256}$ ,  $\mathbf{E} \in \mathbb{R}^{L \times L \times 128}$ ,  $\mathbf{atom\_coords} \in \mathbb{R}^{L \times 14 \times 3}$ ):
  # Initialize degree-0, degree-1, and edge features
  1:  $\mathbf{F}^{(0)} \leftarrow \text{Node\_Projection}(\mathbf{N})[\dots, \text{None}]$   $\mathbf{F}^{(0)} \in \mathbb{R}^{L \times 320 \times 1}$ 
  2:  $\mathbf{F}^{(1)} \leftarrow \mathbf{atom\_coords}$   $\mathbf{F}^{(1)} \in \mathbb{R}^{L \times 14 \times 3}$ 
  3:  $\tilde{\mathbf{E}} \leftarrow \text{Edge\_Projection}(\mathbf{E})$   $\tilde{\mathbf{E}} \in \mathbb{R}^{L \times L \times 32}$ 
  4:  $\mathbf{C} \leftarrow \mathbf{atom\_coords}_{\mathbf{C}_\alpha}$   $\mathbf{C} \in \mathbb{R}^{L \times 3}$ 

  # Construct geometric and pairwise-representation neighborhoods
  5:  $D_{ij} \leftarrow \|\mathbf{C}_i - \mathbf{C}_j\|_2$ ,  $s_{ij} \leftarrow \|\mathbf{E}_{ij}\|_2$   $\mathbf{D}, \mathbf{s} \in \mathbb{R}^{L \times L}$ 
  6:  $k_{\text{geo}} \leftarrow \text{round}(L^{1/2})$ ,  $k_{\text{pair}} \leftarrow \text{round}(L^{1/3})$ 
  7:  $\mathcal{N}_{\text{geo}}(i) \leftarrow \text{TopKIndices}_{\min}(\{D_{ij} : j \neq i\}, k_{\text{geo}})$ 
  8:  $\mathcal{N}_{\text{pair}}(i) \leftarrow \text{TopKIndices}_{\max}(\{s_{ij} : j \notin \mathcal{N}_{\text{geo}}(i) \cup \{i\}\}, k_{\text{pair}})$ 
  9:  $\mathcal{N}(i) \leftarrow \mathcal{N}_{\text{geo}}(i) \cup \mathcal{N}_{\text{pair}}(i)$ ,  $K \leftarrow |\mathcal{N}(i)|$ 

  # Compute relative geometry, local distance statistics, and equivariant bases
  10:  $\Delta \mathbf{x}_{ij} \leftarrow \mathbf{C}_i - \mathbf{C}_j$ ,  $d_{ij} \leftarrow \|\Delta \mathbf{x}_{ij}\|_2$ ,  $j \in \mathcal{N}(i)$   $\mathbf{d} \in \mathbb{R}^{L \times K}$ 
  11:  $\mathbf{D}_i^{\text{stat}} \leftarrow [\text{Mean}_{j \in \mathcal{N}(i)}(d_{ij}), \text{Std}_{j \in \mathcal{N}(i)}(d_{ij})]$   $\mathbf{D}^{\text{stat}} \in \mathbb{R}^{L \times 2}$ 
  12:  $\Phi_{ij} \leftarrow \text{SE3\_Equivariant\_Basis}(\Delta \mathbf{x}_{ij})$ 

  # Apply the input convolution with  $\mathbf{C}_\alpha$ -centered atomic-coordinate features
  13:  $\mathcal{F} \leftarrow \text{ConvSE3}_{\text{in}}(\mathbf{F}^{(0)}, \mathbf{F}^{(1)}, \tilde{\mathbf{E}}, \mathcal{N}, \mathbf{d}, \Phi;$   $\mathcal{F} = \{\mathbf{F}^{(0)}, \mathbf{F}^{(1)}\}$ 
    pool = True, self_interaction = True, center_deg1_by_ca = True)

  # Apply one SE(3)-Transformer block including the central residue in attention
  14:  $\mathcal{F} \leftarrow \text{SE3\_Transformer\_Block}(\mathcal{F}, \tilde{\mathbf{E}}, \mathcal{N}, \mathbf{d}, \Phi)$ 

  # Apply the output convolution without coordinate centering
  15:  $\mathcal{F} \leftarrow \text{ConvSE3}_{\text{out}}(\mathcal{F}, \tilde{\mathbf{E}}, \mathcal{N}, \mathbf{d}, \Phi;$ 
    pool = True, self_interaction = True, center_deg1_by_ca = False)

  # Extract encoded representations
  16:  $\mathbf{Z} \leftarrow \text{Squeeze}(\mathcal{F}^{(0)})$ ,  $\mathbf{V} \leftarrow \mathcal{F}^{(1)}$   $\mathbf{Z} \in \mathbb{R}^{L \times 128}$ ,  $\mathbf{V} \in \mathbb{R}^{L \times 32 \times 3}$ 
  17: return  $\mathbf{Z}, \mathbf{V}, \mathbf{D}^{\text{stat}}$ 

```

1228 **S7.4.3 Construction of the single-mutation score tensor and the epistasis representation tensor**

1229 Supplementary **Algorithm S6** defines the mutation landscape head, which coordinates the single-

1230 mutation score branch and the epistasis representation branch detailed in Supplementary **Algorithm S7**

1231 and Supplementary **Algorithm S8**, respectively. Given $\mathbf{Z}$, $\mathbf{V}$, $\mathbf{D}^{\text{stat}}$, $\mathbf{X}^{\text{ESM-2}}$, and $\mathbf{wt_seq}$, the two

1232 branches construct the single-mutation score tensor $\mathbf{S} \in \mathbb{R}^{L \times 20}$ and the epistasis representation tensor

1233 $\mathbf{U} \in \mathbb{R}^{L \times 20 \times r_H}$.

Algorithm S6 Mutation landscape head

```

def Mutation_Landscape_Head( $\mathbf{Z} \in \mathbb{R}^{L \times 128}$ ,  $\mathbf{V} \in \mathbb{R}^{L \times 32 \times 3}$ ,  $\mathbf{D}^{\text{stat}} \in \mathbb{R}^{L \times 2}$ ,  $\mathbf{X}^{\text{ESM-2}} \in \mathbb{R}^{L \times 1280}$ ,  $\mathbf{wt\_seq} \in \mathcal{A}^L$ ):
    # Convert the wild-type sequence to amino-acid indices
    1:  $\mathbf{w} \leftarrow \text{AAIndex}(\mathbf{wt\_seq})$   $\mathbf{w} \in \{1, \dots, 20\}^L$ 
    # Construct the single-mutation score tensor
    2:  $\mathbf{S} \leftarrow \text{Single\_Mutation\_Score\_Branch}(\mathbf{Z}, \mathbf{X}^{\text{ESM-2}}, \mathbf{w})$   $\mathbf{S} \in \mathbb{R}^{L \times 20}$ 
    # Construct the epistasis representation tensor
    3:  $\mathbf{U} \leftarrow \text{Epistasis\_Representation\_Branch}(\mathbf{Z}, \mathbf{V}, \mathbf{D}^{\text{stat}}, \mathbf{w})$   $\mathbf{U} \in \mathbb{R}^{L \times 20 \times r_H}$ 
    4: return  $\mathbf{S}, \mathbf{U}$ 

```

Algorithm S7 Single-mutation score branch

```

def Single_Mutation_Score_Branch( $\mathbf{Z} \in \mathbb{R}^{L \times 128}$ ,  $\mathbf{X}^{\text{ESM-2}} \in \mathbb{R}^{L \times 1280}$ ,  $\mathbf{w} \in \{1, \dots, 20\}^L$ ):
    # Map the invariant structural representation to mutation scores
    1:  $\mathbf{H} \leftarrow \text{LayerNorm}(\mathbf{Z})$ 
    2:  $\mathbf{H} \leftarrow \text{LeakyReLU}(\text{Linear}(\mathbf{H}))$ 
    3:  $\mathbf{S} \leftarrow \text{Linear}(\mathbf{H})$   $\mathbf{S} \in \mathbb{R}^{L \times 20}$ 
    # Map the ESM-2 representation to a sequence-derived correction
    4:  $\mathbf{H} \leftarrow \text{LayerNorm}(\mathbf{X}^{\text{ESM-2}})$ 
    5:  $\mathbf{H} \leftarrow \text{LeakyReLU}(\text{Linear}(\mathbf{H}))$ 
    6:  $\mathbf{H} \leftarrow \text{LeakyReLU}(\text{Linear}(\mathbf{H}))$ 
    7:  $\Delta \mathbf{S} \leftarrow \text{Linear}(\mathbf{H})$   $\Delta \mathbf{S} \in \mathbb{R}^{L \times 20}$ 
    # Stochastically gate the ESM-2 correction, where  $p_{\text{drop}}^{\text{ESM-2}}$  is the dropout probability and  $\beta$  is a learnable scalar
    8:  $p_{\text{keep}} \leftarrow 1 - p_{\text{drop}}^{\text{ESM-2}}$ ,  $\delta \sim \text{Bernoulli}(p_{\text{keep}})$   $\delta \in \{0, 1\}$ 
    9:  $\gamma \leftarrow \delta / p_{\text{keep}}$  if training, otherwise 1
    10:  $\mathbf{S} \leftarrow \mathbf{S} + \beta \gamma \Delta \mathbf{S}$ 
    # Center scores relative to the wild-type amino acid
    11:  $S_{i,a} \leftarrow S_{i,a} - S_{i,w_i}$ ,  $i = 1, \dots, L$ ,  $a = 1, \dots, 20$   $\mathbf{S} \in \mathbb{R}^{L \times 20}$ 
    12: return  $\mathbf{S}$ 

```

Algorithm S8 Epistasis representation branch

def Epistasis_Representation_Branch($\mathbf{Z} \in \mathbb{R}^{L \times 128}$, $\mathbf{V} \in \mathbb{R}^{L \times 32 \times 3}$, $\mathbf{D}^{\text{stat}} \in \mathbb{R}^{L \times 2}$, $\mathbf{w} \in \{1, \dots, 20\}^L$):

Learnable parameters: amino-acid embeddings $\mathbf{E}^{\text{AA}} \in \mathbb{R}^{20 \times r_H}$ and a bias-free linear map $\mathbf{W}_U \in \mathbb{R}^{r_H \times r_H}$

Construct an invariant residue-level interaction context

1: $\mathbf{V}^{\text{mag}} \leftarrow \|\mathbf{V}\|_2$

$\mathbf{V}^{\text{mag}} \in \mathbb{R}^{L \times 32}$

2: $\mathbf{H} \leftarrow \text{Concat}[\mathbf{Z}, \mathbf{V}^{\text{mag}}, \mathbf{D}^{\text{stat}}]$

3: $\mathbf{H} \leftarrow \text{GELU}(\text{Linear}(\text{LayerNorm}(\mathbf{H})))$

4: $\mathbf{H} \leftarrow \text{LayerNorm}(\mathbf{H})$

5: $\mathbf{H} \leftarrow \text{GELU}(\text{Linear}(\mathbf{H}))$

6: $\mathbf{H} \leftarrow \text{GELU}(\text{Linear}(\mathbf{H}))$

7: $\mathbf{H} \leftarrow \text{Linear}(\mathbf{H})$

$\mathbf{H} \in \mathbb{R}^{L \times r_H}$

Condition residue contexts on substituted amino-acid identities

8: $\tilde{\mathbf{U}}_{i,a} \leftarrow \mathbf{W}_U(\mathbf{H}_i \odot \mathbf{E}_a^{\text{AA}})$

$\tilde{\mathbf{U}} \in \mathbb{R}^{L \times 20 \times r_H}$

Center representations relative to the wild-type amino acid

9: $\mathbf{U}_{i,a} \leftarrow \tilde{\mathbf{U}}_{i,a} - \tilde{\mathbf{U}}_{i,w_i}, \quad i = 1, \dots, L, a = 1, \dots, 20$

$\mathbf{U} \in \mathbb{R}^{L \times 20 \times r_H}$

10: **return** $\mathbf{U}$

S7.4.4 Assembly and prediction of arbitrary-order mutants

Given the precomputed single-mutation score tensor $\mathbf{S}$ and epistasis representation tensor $\mathbf{U}$, the assembly module predicts arbitrary-order mutants without re-encoding their sequences or structures. For notational simplicity, consider an individual mutant $\mathcal{M} = \{(i_k, a_k)\}_{k=1}^m$, where the mutation-specific score $s_k = S_{i_k, a_k}$ and epistasis representation $\mathbf{u}_k = \mathbf{U}_{i_k, a_k, :}$ are directly indexed from $\mathbf{S}$ and $\mathbf{U}$ for each substitution. The mutant fitness is predicted as

$$\hat{y}_{\mathcal{M}} = \underbrace{\sum_{k=1}^m [s_k + f_{\theta}(\mathbf{u}_k)]}_{\text{single-mutation contributions}} + \underbrace{\left[f_{\theta} \left(c_m \sum_{k=1}^m \mathbf{u}_k \right) - \sum_{k=1}^m f_{\theta}(\mathbf{u}_k) \right]}_{\hat{\epsilon}_{\mathcal{M}}}. \quad (\text{S9})$$

The first term represents the sum of the individual mutation contributions, whereas the second term defines the total epistatic contribution $\hat{\epsilon}_{\mathcal{M}}$. Because the assembly depends only on summation over substitutions, both $\hat{y}_{\mathcal{M}}$ and $\hat{\epsilon}_{\mathcal{M}}$ are invariant to mutation ordering. Supplementary [Algorithm S9](#) summarizes the corresponding indexing and assembly rule for an individual mutant. For complete-landscape inference, the same rule is vectorized over batches of mutants in the implementation.

Algorithm S9 Assembly and prediction of an arbitrary-order mutant

def Assembly_Module($S \in \mathbb{R}^{L \times 20}$, $U \in \mathbb{R}^{L \times 20 \times r_H}$, $\mathcal{M} = \{(i_k, a_k)\}_{k=1}^m, f_\theta$):

Extract mutation-specific components from the precomputed tensors

1: $s_k \leftarrow S_{i_k, a_k}, \quad \mathbf{u}_k \leftarrow U_{i_k, a_k}, \quad k = 1, \dots, m$

$s_k \in \mathbb{R}, \mathbf{u}_k \in \mathbb{R}^{r_H}$

Assemble the mutant fitness and epistatic component

2: $\hat{y}_{\mathcal{M}} \leftarrow \sum_{k=1}^m s_k + f_\theta \left(c_m \sum_{k=1}^m \mathbf{u}_k \right)$

$\hat{y}_{\mathcal{M}} \in \mathbb{R}$

3: $\hat{\epsilon}_{\mathcal{M}} \leftarrow f_\theta \left(c_m \sum_{k=1}^m \mathbf{u}_k \right) - \sum_{k=1}^m f_\theta(\mathbf{u}_k)$

$\hat{\epsilon}_{\mathcal{M}} \in \mathbb{R}$

4: **return** $\hat{y}_{\mathcal{M}}, \hat{\epsilon}_{\mathcal{M}}$

1245 The shared nonlinear interaction function f_θ used in the assembly module maps either an individual
1246 epistasis representation or the normalized aggregate of multiple epistasis representations to a scalar
1247 response. Supplementary [Algorithm S10](#) details its architecture.

Algorithm S10 Learnable interaction function

def $f_\theta(\mathbf{h} \in \mathbb{R}^{r_H})$:

Map an epistasis representation to a scalar response

1: $\mathbf{z}_1 \leftarrow \text{Dropout}(\text{GELU}(\text{LayerNorm}(\mathbf{W}_1 \mathbf{h} + \mathbf{b}_1)))$

$\mathbf{z}_1 \in \mathbb{R}^{256}$

2: $\mathbf{z}_2 \leftarrow \text{Dropout}(\text{GELU}(\mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2))$

$\mathbf{z}_2 \in \mathbb{R}^{128}$

3: $y \leftarrow \mathbf{W}_3 \mathbf{z}_2 + b_3$

$y \in \mathbb{R}$

4: **return** y

1248 **S7.4.5 Core operations of the SE(3)-Transformer encoder**

1249 Supplementary [Algorithm S11](#) and Supplementary [Algorithm S12](#) describe the input projections applied
1250 to the Cerebra residue and pair representations before equivariant processing. The node projection maps
1251 the normalized 256-channel residue representation to 320 degree-0 input channels through two linear
1252 layers with a LeakyReLU activation and final normalization. The edge projection maps the 128-channel
1253 pair representation through a GELU-activated hidden representation to the 32 edge channels used
1254 by the equivariant kernels. Supplementary [Algorithm S14](#) and Supplementary [Algorithm S13](#) then
1255 summarize the two core equivariant modules used within the encoder. ConvSE3 gathers degree-wise
1256 features from the selected neighbors and transforms them using SE(3)-equivariant kernels conditioned
1257 on Fourier-encoded inter-residue distances and projected Cerebra pairwise features, together with
1258 spherical-harmonic equivariant bases. In ConvSE3_{in}, the degree-1 atomic-coordinate features of each
1259 neighboring residue are expressed relative to the C_α coordinate of the central residue, with atom-validity
1260 masking handled internally and omitted from the pseudocode for clarity. The SE(3)-Transformer block
1261 subsequently updates the degree-wise representations through equivariant attention and feed-forward
1262 transformations. The remaining mathematical formulations follow the standard SE(3)-Transformer
1263 architecture [37].

Algorithm S11 Node feature projection

def Node_Projection($\mathbf{N} \in \mathbb{R}^{L \times 256}$):

Project the Cerebra residue representation to the degree-0 input channels

1: $\mathbf{H} \leftarrow \text{LayerNorm}(\mathbf{N})$

2: $\mathbf{H} \leftarrow \text{LeakyReLU}(\text{Linear}(\mathbf{H}))$

$\mathbf{H} \in \mathbb{R}^{L \times 320}$

3: $\mathbf{F}^N \leftarrow \text{LayerNorm}(\text{Linear}(\mathbf{H}))$

$\mathbf{F}^N \in \mathbb{R}^{L \times 320}$

4: **return** $\mathbf{F}^N$

Algorithm S12 Edge feature projection

def Edge_Projection($\mathbf{E} \in \mathbb{R}^{L \times L \times 128}$):

Project the Cerebra pair representation to the SE(3) edge channels

1: $\mathbf{H} \leftarrow \text{GELU}(\text{Linear}(\mathbf{E}))$

$\mathbf{H} \in \mathbb{R}^{L \times L \times 200}$

2: $\tilde{\mathbf{E}} \leftarrow \text{Linear}(\mathbf{H})$

$\tilde{\mathbf{E}} \in \mathbb{R}^{L \times L \times 32}$

3: **return** $\tilde{\mathbf{E}}$

Algorithm S13 SE(3)-Transformer block

def SE3_Transformer_Block($\mathcal{F}, \tilde{\mathbf{E}}, \mathcal{N}, \mathbf{d}, \Phi$):

Generate equivariant queries, keys, and values

- 1: $\mathcal{H}_{\text{attn}} \leftarrow \text{NormSE3}(\mathcal{F}, \text{GELU}, \text{use_LN} = \text{True})$
- 2: $\mathcal{Q} \leftarrow \text{LinearSE3}_Q(\mathcal{H}_{\text{attn}})$
- 3: $\mathcal{K} \leftarrow \text{ConvSE3}_K(\mathcal{H}_{\text{attn}}, \tilde{\mathbf{E}}, \mathcal{N}, \mathbf{d}, \Phi;$
 $\text{pool} = \text{False}, \text{self_interaction} = \text{False}, \text{center_deg1_by_ca} = \text{False})$
- 4: $\mathcal{V} \leftarrow \text{ConvSE3}_V(\mathcal{H}_{\text{attn}}, \tilde{\mathbf{E}}, \mathcal{N}, \mathbf{d}, \Phi;$
 $\text{pool} = \text{False}, \text{self_interaction} = \text{False}, \text{center_deg1_by_ca} = \text{False})$

Include the central residue in attention

- 5: $\mathcal{K}_{\text{self}} \leftarrow \text{LinearSE3}_{K, \text{self}}(\mathcal{H}_{\text{attn}}), \quad \mathcal{V}_{\text{self}} \leftarrow \text{LinearSE3}_{V, \text{self}}(\mathcal{H}_{\text{attn}})$
- 6: $\mathcal{K} \leftarrow \text{Prepend}(\mathcal{K}_{\text{self}}, \mathcal{K}), \quad \mathcal{V} \leftarrow \text{Prepend}(\mathcal{V}_{\text{self}}, \mathcal{V})$

Compute invariant attention weights and aggregate equivariant values

- 7: **for** $\ell \in \{0, 1\}$ **do**
- 8: $\mathbf{Q}^{(\ell)}, \mathbf{K}^{(\ell)}, \mathbf{V}^{(\ell)} \leftarrow \text{SplitHeads}(\mathbf{Q}^{(\ell)}, \mathbf{K}^{(\ell)}, \mathbf{V}^{(\ell)})$
- 9: $a_{ijh}^{(\ell)} \leftarrow \frac{\langle \mathbf{q}_{ih}^{(\ell)}, \mathbf{k}_{ijh}^{(\ell)} \rangle_{\text{F}}}{\sqrt{d_h}}$
- 10: $\alpha_{ijh}^{(\ell)} \leftarrow \text{Softmax}_j(\text{Mask}(a_{ijh}^{(\ell)}))$
- 11: $\mathbf{O}^{(\ell)} \leftarrow \text{ConcatHeads}\left(\alpha_{iih}^{(\ell)} \mathbf{v}_{iih}^{(\ell)} + \sum_{j \in \mathcal{N}(i)} \alpha_{ijh}^{(\ell)} \mathbf{v}_{ijh}^{(\ell)}\right)$
- 12: **end for**
- 13: $\mathcal{O}_{\text{attn}} \leftarrow \text{LinearSE3}_O(\{\mathbf{O}^{(0)}, \mathbf{O}^{(1)}\})$
- 14: $\tilde{\mathcal{F}} \leftarrow \text{ResidualSE3}(\mathcal{O}_{\text{attn}}, \mathcal{F})$

Apply a degree-wise feed-forward network with a residual connection

- 15: $\mathcal{H}_{\text{ffn}} \leftarrow \text{NormSE3}(\tilde{\mathcal{F}}, \text{Identity}, \text{use_LN} = \text{True})$
 - 16: $\mathcal{P} \leftarrow \text{FeedForwardSE3}(\mathcal{H}_{\text{ffn}})$
 - 17: $\mathcal{F}_{\text{out}} \leftarrow \text{ResidualSE3}(\mathcal{P}, \tilde{\mathcal{F}})$
 - 18: **return** $\mathcal{F}_{\text{out}}$
-

Algorithm S14 SE(3)-equivariant convolution

def ConvSE3($\mathcal{F}, \tilde{\mathbf{E}}, \mathcal{N}, \mathbf{d}, \Phi, \text{pool}, \text{self_interaction}, \text{center_deg1_by_ca}$):

Construct distance- and edge-conditioned equivariant kernels

1: $\rho_{ij} \leftarrow \text{Concat}\left[\text{FourierEncode}(d_{ij}), \tilde{\mathbf{E}}_{ij}\right], \quad j \in \mathcal{N}(i)$

$$\rho \in \mathbb{R}^{L \times K \times d_\rho}$$

Transform features between degree-0 and degree-1 representations

2: **for** $\ell_{\text{out}} \in \{0, 1\}$ **do**

3: $\mathbf{Y}_{ij}^{(\ell_{\text{out}})} \leftarrow 0$

4: **for** $\ell_{\text{in}} \in \{0, 1\}$ **do**

5: $\mathbf{X}_{ij}^{(\ell_{\text{in}})} \leftarrow \text{Gather}\left(\mathbf{F}_j^{(\ell_{\text{in}})}, j \in \mathcal{N}(i)\right)$

6: **if** center_deg1_by_ca and $\ell_{\text{in}} = 1$ **then**

7: $\mathbf{X}_{ij,q}^{(1)} \leftarrow \mathbf{X}_{ij,q}^{(1)} - \mathbf{F}_{i,C_\alpha,:}^{(1)}$

8: **end if**

9: $\mathbf{K}_{ij}^{\ell_{\text{in}} \rightarrow \ell_{\text{out}}} \leftarrow \text{PairwiseKernel}\left(\rho_{ij}, \Phi_{ij}^{\ell_{\text{in}}, \ell_{\text{out}}}\right)$

10: $\mathbf{Y}_{ij}^{(\ell_{\text{out}})} \leftarrow \mathbf{Y}_{ij}^{(\ell_{\text{out}})} + \mathbf{K}_{ij}^{\ell_{\text{in}} \rightarrow \ell_{\text{out}}} \text{Vec}\left(\mathbf{X}_{ij}^{(\ell_{\text{in}})}\right)$

11: **end for**

Optionally aggregate neighboring messages

12: **if** pool **then**

13: $\mathbf{G}_i^{(\ell_{\text{out}})} \leftarrow \text{MaskedMean}_{j \in \mathcal{N}(i)} \mathbf{Y}_{ij}^{(\ell_{\text{out}})}$

14: **else**

15: $\mathbf{G}_{ij}^{(\ell_{\text{out}})} \leftarrow \mathbf{Y}_{ij}^{(\ell_{\text{out}})}$

16: **end if**

17: **end for**

Optionally add a degree-preserving self interaction

18: **if** self_interaction **then**

19: $\mathcal{F}_{\text{self}} \leftarrow \mathcal{F}$

20: **if** center_deg1_by_ca **then**

21: $\mathbf{F}_{\text{self},i,q}^{(1)} \leftarrow \mathbf{F}_{i,q}^{(1)} - \mathbf{F}_{i,C_\alpha,:}^{(1)}$

22: **end if**

23: $\mathcal{G} \leftarrow \text{ResidualSE3}(\mathcal{G}, \text{LinearSE3}(\mathcal{F}_{\text{self}}))$

$$\mathcal{G} = \{\mathbf{G}^{(0)}, \mathbf{G}^{(1)}\}$$

24: **end if**

25: **return** $\mathcal{G}$

Supplementary References

- [1] Mihaly Varadi et al. “AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models”. In: *Nucleic Acids Research* 50.D1 (2022), pp. D439–D444. DOI: [10.1093/nar/gkab1061](https://doi.org/10.1093/nar/gkab1061).
- [2] Inigo Barrio-Hernandez et al. “Clustering predicted structures at the scale of the known protein universe”. In: *Nature* 622.7983 (2023), pp. 637–645. DOI: [10.1038/s41586-023-06510-w](https://doi.org/10.1038/s41586-023-06510-w).

- [3] Wolfgang Kabsch and Christian Sander. “Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features”. In: *Biopolymers* 22.12 (1983), pp. 2577–2637. DOI: [10.1002/bip.360221211](https://doi.org/10.1002/bip.360221211).
- [4] Kotaro Tsuboyama et al. “Mega-scale experimental analysis of protein folding stability in biology and design”. In: *Nature* 620.7973 (July 2023), pp. 434–444. ISSN: 1476-4687. DOI: [10.1038/s41586-023-06328-6](https://doi.org/10.1038/s41586-023-06328-6). URL: <http://dx.doi.org/10.1038/s41586-023-06328-6>.
- [5] Andre J. Faure et al. “The genetic architecture of protein stability”. In: *Nature* 634.8035 (Sept. 2024), pp. 995–1003. ISSN: 1476-4687. DOI: [10.1038/s41586-024-07966-0](https://doi.org/10.1038/s41586-024-07966-0). URL: <http://dx.doi.org/10.1038/s41586-024-07966-0>.
- [6] Thuy-Lan V Lite et al. “Uncovering the basis of protein-protein interaction specificity with a combinatorially complete library”. In: *eLife* 9 (Oct. 2020). ISSN: 2050-084X. DOI: [10.7554/eLife.60924](https://doi.org/10.7554/eLife.60924). URL: <http://dx.doi.org/10.7554/eLife.60924>.
- [7] David Ding et al. “Protein design using structure-based residue preferences”. In: *Nature Communications* 15.1 (Feb. 2024). ISSN: 2041-1723. DOI: [10.1038/s41467-024-45621-4](https://doi.org/10.1038/s41467-024-45621-4). URL: <http://dx.doi.org/10.1038/s41467-024-45621-4>.
- [8] Kadina E. Johnston et al. “A combinatorially complete epistatic fitness landscape in an enzyme active site”. In: *Proceedings of the National Academy of Sciences* 121.32 (July 2024). ISSN: 1091-6490. DOI: [10.1073/pnas.2400439121](https://doi.org/10.1073/pnas.2400439121). URL: <http://dx.doi.org/10.1073/pnas.2400439121>.
- [9] Christoph Küng et al. “Deep mutational scanning reveals a de novo disulfide bond and combinatorial mutations for engineering thermostable myoglobin”. In: *Protein Science* 34.5 (Apr. 2025). ISSN: 1469-896X. DOI: [10.1002/pro.70112](https://doi.org/10.1002/pro.70112). URL: <http://dx.doi.org/10.1002/pro.70112>.
- [10] Allison Judge et al. “Network of epistatic interactions in an enzyme active site revealed by large-scale deep mutational scanning”. In: *Proceedings of the National Academy of Sciences* 121.12 (Mar. 2024). ISSN: 1091-6490. DOI: [10.1073/pnas.2313513121](https://doi.org/10.1073/pnas.2313513121). URL: <http://dx.doi.org/10.1073/pnas.2313513121>.
- [11] Pascal Notin et al. “ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design”. In: *Advances in Neural Information Processing Systems* 36. NeurIPS 2023. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023, pp. 64331–64379. DOI: [10.52202/075280-2810](https://doi.org/10.52202/075280-2810). URL: <http://dx.doi.org/10.52202/075280-2810>.
- [12] Ziyu Shi et al. “MutCleaner: Cleaning and Standardizing Biological Mutation Datasets for Variant Effect Prediction”. In: *bioRxiv* (2026). DOI: [10.64898/2026.09.06.749687](https://doi.org/10.64898/2026.09.06.749687). URL: <https://doi.org/10.64898/2026.09.06.749687>.
- [13] Tianyu Mi, Nan Xiao, and Haipeng Gong. “GDFold2: A fast and parallelizable protein folding environment with freely defined objective functions”. In: *Protein Science* 34.2 (2025), e70041. DOI: [10.1002/pro.70041](https://doi.org/10.1002/pro.70041). URL: <https://doi.org/10.1002/pro.70041>.
- [14] Vincent Q. Tran et al. “Rapid directed evolution guided by protein language models and epistatic interactions”. In: *Science* 392.6798 (May 2026). ISSN: 1095-9203. DOI: [10.1126/science.aea1820](https://doi.org/10.1126/science.aea1820). URL: <http://dx.doi.org/10.1126/science.aea1820>.

- [15] Tianyu Mi et al. “Accelerating Virtual Directed Evolution of Proteins via Reinforcement Learning”. In: (June 2025). DOI: [10.1101/2025.06.25.661516](https://doi.org/10.1101/2025.06.25.661516). URL: <http://dx.doi.org/10.1101/2025.06.25.661516>.
- [16] Iván Martín Hernández et al. “Predicting protein stability changes upon mutation using a simple orientational potential”. In: *Bioinformatics* 39.1 (Jan. 2023). Ed. by Alfonso Valencia. ISSN: 1367-4811. DOI: [10.1093/bioinformatics/btad011](https://doi.org/10.1093/bioinformatics/btad011). URL: <http://dx.doi.org/10.1093/bioinformatics/btad011>.
- [17] Corrado Pancotti et al. “Predicting protein stability changes upon single-point mutation: a thorough comparison of the available tools on a new dataset”. In: *Briefings in Bioinformatics* 23.2 (Jan. 2022). ISSN: 1477-4054. DOI: [10.1093/bib/bbab555](https://doi.org/10.1093/bib/bbab555). URL: <http://dx.doi.org/10.1093/bib/bbab555>.
- [18] Jian Hu, Weizhe Wang, and Haipeng Gong. “Cerebra: a computationally efficient framework for accurate protein structure prediction”. In: (Feb. 2024). DOI: [10.1101/2024.02.02.578551](https://doi.org/10.1101/2024.02.02.578551). URL: <http://dx.doi.org/10.1101/2024.02.02.578551>.
- [19] Zeming Lin et al. “Evolutionary-scale prediction of atomic-level protein structure with a language model”. In: *Science* 379.6637 (Mar. 2023), pp. 1123–1130. ISSN: 1095-9203. DOI: [10.1126/science.ade2574](https://doi.org/10.1126/science.ade2574). URL: <http://dx.doi.org/10.1126/science.ade2574>.
- [20] Ruidong Wu et al. “High-resolution de novo structure prediction from primary sequence”. In: (July 2022). DOI: [10.1101/2022.07.21.500999](https://doi.org/10.1101/2022.07.21.500999). URL: <http://dx.doi.org/10.1101/2022.07.21.500999>.
- [21] Thomas Hayes et al. “Simulating 500 million years of evolution with a language model”. In: *Science* 387.6736 (Feb. 2025), pp. 850–858. ISSN: 1095-9203. DOI: [10.1126/science.ads0018](https://doi.org/10.1126/science.ads0018). URL: <http://dx.doi.org/10.1126/science.ads0018>.
- [22] Yinghui Chen et al. “An end-to-end framework for the prediction of protein structure and fitness from single sequence”. In: *Nature Communications* 15.1 (Aug. 2024). ISSN: 2041-1723. DOI: [10.1038/s41467-024-51776-x](https://doi.org/10.1038/s41467-024-51776-x). URL: <http://dx.doi.org/10.1038/s41467-024-51776-x>.
- [23] Joshua Meier et al. “Language models enable zero-shot prediction of the effects of mutations on protein function”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 29287–29303. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/f51338d736f95dd42427296047067694-Paper.pdf.
- [24] Pascal Notin et al. “Tranception: Protein Fitness Prediction with Autoregressive Transformers and Inference-time Retrieval”. In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. PMLR, 2022, pp. 16990–17017.
- [25] Roshan M. Rao et al. “MSA Transformer”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. PMLR, 2021, pp. 8844–8856.
- [26] Peter Groth et al. “Kermut: Composite kernel regression for protein variant effects”. In: *Advances in Neural Information Processing Systems* 37. NeurIPS 2024. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024, pp. 29514–29565. DOI: [10.52202/079017-0929](https://doi.org/10.52202/079017-0929). URL: <http://dx.doi.org/10.52202/079017-0929>.

- [27] Pascal Notin et al. “ProteinNPT: Improving Protein Property Prediction and Design with Non-Parametric Transformers”. In: *Advances in Neural Information Processing Systems 36*. NeurIPS 2023. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023, pp. 33529–33563. DOI: [10.52202/075280-1458](https://doi.org/10.52202/075280-1458). URL: <http://dx.doi.org/10.52202/075280-1458>.
- [28] Yunzhuo Zhou et al. “DDMut: predicting effects of mutations on protein stability using deep learning”. In: *Nucleic Acids Research* 51.W1 (2023), W122–W128. DOI: [10.1093/nar/gkad472](https://doi.org/10.1093/nar/gkad472).
- [29] James P. Roney, Chenxi Ou, and Sergey Ovchinnikov. “Protein Diffusion Models as Statistical Potentials”. In: *bioRxiv* (2025). DOI: [10.64898/2025.12.09.693073](https://doi.org/10.64898/2025.12.09.693073).
- [30] Dmitriy Umerenkov et al. “PROSTATA: a framework for protein stability assessment using transformers”. In: *Bioinformatics* 39.11 (2023), btad671. DOI: [10.1093/bioinformatics/btad671](https://doi.org/10.1093/bioinformatics/btad671).
- [31] Lasse M. Blaabjerg et al. “Rapid protein stability prediction using deep learning representations”. In: *eLife* 12 (2023), e82593. DOI: [10.7554/eLife.82593](https://doi.org/10.7554/eLife.82593).
- [32] Yunxin Xu, Di Liu, and Haipeng Gong. “Improving the prediction of protein stability changes upon mutations by geometric learning and a pre-training strategy”. In: *Nature Computational Science* 4 (2024), pp. 840–850. DOI: [10.1038/s43588-024-00716-2](https://doi.org/10.1038/s43588-024-00716-2).
- [33] Jeffrey Ouyang-Zhang et al. “Predicting a Protein’s Stability under a Million Mutations”. In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 76229–76247. DOI: [10.52202/075280-3332](https://doi.org/10.52202/075280-3332).
- [34] Henry Dieckhaus et al. “Transfer learning to leverage larger datasets for improved prediction of protein stability changes”. In: *Proceedings of the National Academy of Sciences* 121.6 (2024), e2314853121. DOI: [10.1073/pnas.2314853121](https://doi.org/10.1073/pnas.2314853121).
- [35] Jinyuan Sun et al. “Structure-based self-supervised learning enables ultrafast protein stability prediction upon mutation”. In: *The Innovation* 6.1 (2025), p. 100750. DOI: [10.1016/j.xinn.2024.100750](https://doi.org/10.1016/j.xinn.2024.100750).
- [36] Ziang Li and Yunan Luo. “Generalizable and scalable protein stability prediction with rewired protein generative models”. In: *Nature Communications* 17 (2026), p. 891. DOI: [10.1038/s41467-025-67609-4](https://doi.org/10.1038/s41467-025-67609-4).
- [37] Fabian Fuchs et al. “SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1970–1981. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/15231a7ce4ba789d13b722cc5c955834-Paper.pdf.