communications biology

A Nature Portfolio journal

Article

https://doi.org/10.1038/s42003-026-09632-9

JanusDDG: a physics-informed neural network for sequence-based protein stability via two-fronts attention

Check for updates

Guido Barducci [1] ✉, Ivan Rossi [1], Francesco Codicé [1], Cesare Rollo [1], Valeria Repetto [1,2], Corrado Pancotti [3], Virginia Iannibelli [1], Tiziana Sanavia [1] & Piero Fariselli [1] ✉

Predicting how residue variations affect protein stability is crucial for rational protein design and for assessing the impact of disease-related mutations. Recent advances in protein language models have revolutionized computational protein analysis, enabling more accurate predictions of mutational effects. However, balancing predictive accuracy with the fundamental laws of thermodynamics remains a challenge for sequence-based models. Here we show JanusDDG, a physics-informed neural network that leverages embeddings from protein language models and a bidirectional cross-attention transformer architecture to predict stability changes for both single and multiple residue mutations. By adopting a physics-informed paradigm, the model is explicitly constrained to satisfy fundamental thermodynamic principles, such as antisymmetry and transitivity, while maintaining high predictive performance. Instead of conventional self-attention, JanusDDG employs a cross-interleaved attention mechanism that computes the relationship between wild-type and mutant embeddings to capture mutation-induced perturbations while preserving essential contextual information. Our results demonstrate that JanusDDG achieves state-of-the-art performance in predicting stability changes from sequence alone, matching or exceeding the accuracy of structure-based methods for both single and multiple mutations.

Protein stability is a fundamental property that characterizes the structure, function, and overall behavior of a protein in biological systems. One of the most widely used metrics to evaluate protein stability is the change in Gibbs free energy ($\Delta\Delta G$), which quantifies the difference in stability between a wild-type protein and its mutant counterpart. $\Delta\Delta G$ is calculated by comparing the free energy of folding for both proteins, providing information on whether a mutation stabilizes or destabilizes the structure.

In this study, we adopt the convention that a positive $\Delta\Delta G$ value indicates a stabilizing mutation (that is, the mutant form is more thermodynamically favorable than the wild type)[1]. In contrast, a negative $\Delta\Delta G$ suggests that the mutation is destabilizing, making the protein more prone to unfolding or degradation. Consequently, the $\Delta\Delta G$ between a wild-type (w) protein and a mutant (m) of the same protein is defined as:

$$\Delta\Delta G = \Delta G_w - \Delta G_m, \tag{1}$$

where ΔG_w and ΔG_m are given by:

$$\Delta G_w = G^f_w - G^u_w,\ \Delta G_m = G^f_m - G^u_m. \tag{2}$$

Here, G^u represents the Gibbs free energy of the unfolded state, while G^f corresponds to that of the folded state.

Analysis of protein stability is useful for assessing the impact of genetic mutations on disorders caused by protein misfolding[2,3], to guide drug discovery and development of small molecules that compensate for destabilizing mutations or exploit structural weaknesses in pathogenic proteins[4], and to design more stable or functional proteins for industrial and medical applications[5,6].

In recent years, several studies have focused on the prediction of $\Delta\Delta G$, based on energy-based force fields[7–9], machine and deep learning models[10–20], and recently on protein language models[21–23]. These methods belong to two main categories: sequence-based models and structure-based models. Although sequence-based models are more convenient to

[1]AI and Computational Biomedicine Unit, Department of Medical Sciences, University of Turin, Turin, Italy. [2]Institute of Informatics and Telematics (IIT), National Research Council of Italy (CNR), Pisa, Italy. [3]Helmholtz AI, Helmholtz Zentrum München Ingolstädter Landstraße 1 D-85764, Neuherberg, Germany.
✉e-mail: guido.barducci@unito.it; piero.fariselli@unito.it

use, since they only need the amino acid sequence of the protein and its mutations, structure-based models, which require the spatial structure of the protein, usually show better performance than sequence-based models[24–27]. The lack of experimental information on protein structure has led to the development of tools such as AlphaFold[28], RoseTTaFold[29] and OpenFold[30], which can produce high-quality protein structures but at the expense of high computational costs. So far, most of the developed systems are limited to the prediction of stability changes in single or double mutations[31,32], and only a few models are capable of predicting mutations involving more than two amino acids[7,23,33,34].

In this work, we introduce JanusDDG, a deep learning approach that takes advantage of a protein language model[35] to extract informative representations from the sequences of wild-type and mutated proteins. By relying solely on sequence information, JanusDDG avoids the need for structural input while still capturing the rich contextual and evolutionary signals encoded in the language model embeddings. JanusDDG is explicitly designed to be consistent with the physics of protein stability. Through the use of tailored physics-informed loss functions and architectural constraints, it enforces fundamental thermodynamic principles such as antisymmetry (the prediction sign changes when the mutation is reversed) and transitivity (mutational effects remain consistent across intermediate states). These constraints ensure that the model's predictions are not only accurate, but also physically grounded.

Results

JanusDDG is a sequence-based predictor designed to estimate $\Delta\Delta G$ values for both single and double mutations. It builds on the ESM2 language model[35] to extract informative embeddings from protein sequences, which are then processed by a deep neural architecture that incorporates attention mechanisms and is guided by fundamental physical principles.

At the core of the model is a bidirectional cross-attention mechanism that integrates both the difference between wild-type and mutant embeddings and the broader context of the wild-type sequence itself (Fig. 1a). This design enables JanusDDG to operate on full-length protein sequences and generalize across single and combinatorial mutations.

This architecture can be interpreted in line with contrastive learning principles: the model learns to emphasize discriminative features by contrasting the representation of the mutant-wild-type pair (which encodes structural and functional deviations) with the baseline representation of the wild type. In this sense, the $\Delta\Delta G$ prediction task benefits

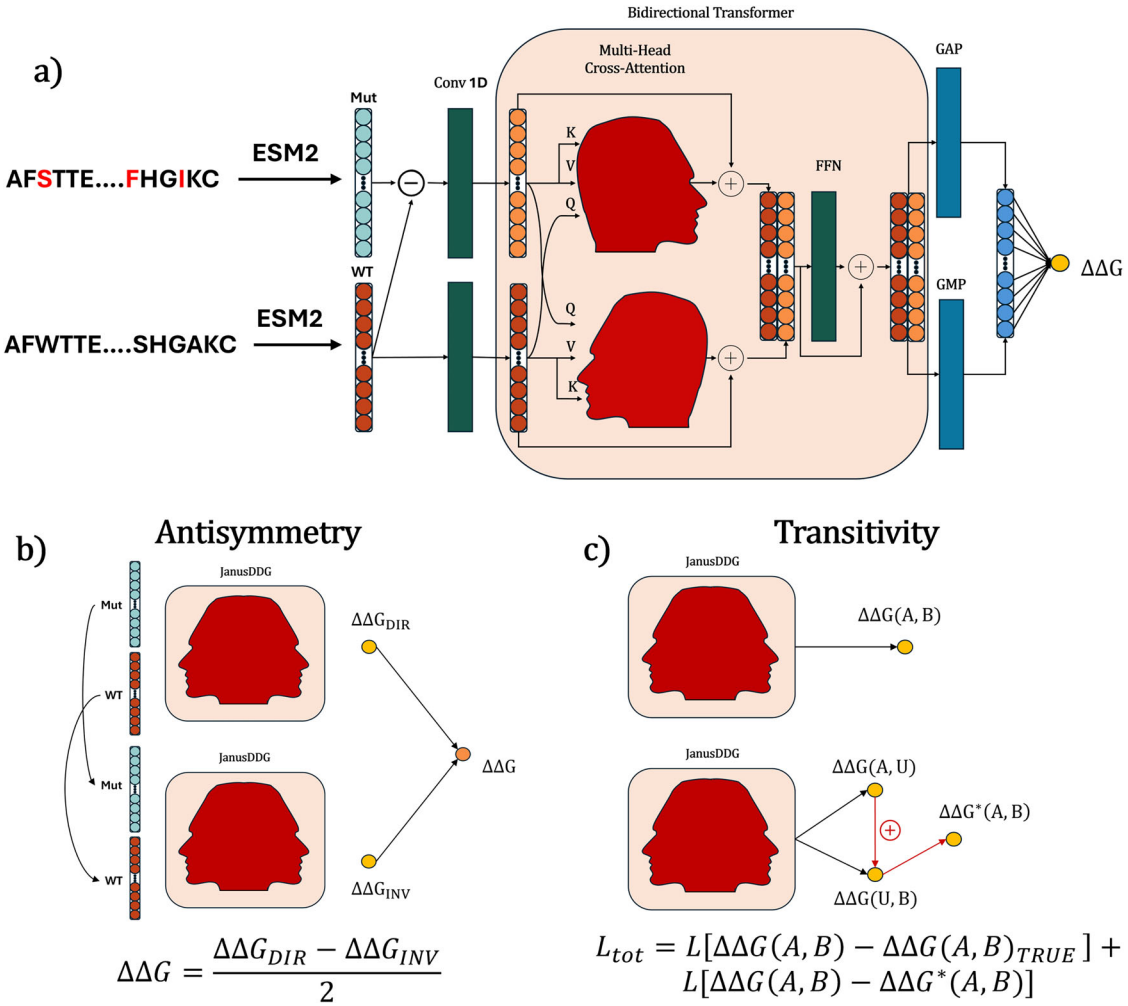

Fig. 1 | JanusDDG Overview. a JanusDDG Backbone. The model takes as input wild-type and mutant-type amino acid sequences without requiring the structural information, and provides a prediction of $\Delta\Delta G$ by leveraging the power of bidirectional cross-attention. This backbone model enables the prediction of stability changes resulting from single and multiple mutations. **b** Antisymmetry. To make the model antisymmetric by design, JanusDDG backbone is applied twice with inverted inputs. The resulting predictions are subtracted from each other and then divided by two. **c** Transitivity. To enhance the transitivity of the model, fine-tuning is applied based on the thermodynamic property that links the Gibbs free energy changes ($\Delta\Delta G$) among three mutations as indicated by the red arrows (A, U, B). Black arrows, as above, indicate JanusDDG processing. The loss function is formulated such that the model learns the following relation:
$\Delta\Delta G(A, B) = \Delta\Delta G^*(A, B) \equiv \Delta\Delta G(A, U) + \Delta\Delta G(U, B)$.

from a contrastive signal, as the model is effectively trained to highlight relevant differences while suppressing non-informative similarities between the two protein states[36].

To ensure thermodynamic consistency, JanusDDG is constructed to satisfy key physical properties of the Gibbs free energy landscape. Antisymmetry is enforced directly through the model architecture, while transitivity is promoted via a dedicated fine-tuning procedure (Fig. 1b, c).

The model is trained on the S2450 dataset introduced in Savojardo, C. et al.[23], which shares less than 25% sequence identity with the evaluation datasets (S669, S461, S96, and PTmul-NR), ensuring a robust assessment of generalization performance (see the Datasets subsection in the Methods section for details).

State-function properties of Gibbs free energy

A reliable $\Delta\Delta G$ predictor must reflect the fundamental nature of Gibbs free energy, which is a thermodynamic state function. As such, it must satisfy two key mathematical properties: (i) Antisymmetry

$$\Delta\Delta G(A,B) = -\Delta\Delta G(B,A), \tag{3}$$

and (ii) Transitivity

$$\Delta\Delta G(A,C) = \Delta\Delta G(A,B) + \Delta\Delta G(B,C). \tag{4}$$

To embed these thermodynamic constraints into the model, JanusDDG employs two specifically designed strategies that integrate physics informed reasoning into the training procedure, effectively making it a form of PINN, as outlined in the following sections[37].

Antisymmetry. The antisymmetry property of $\Delta\Delta G$ implies that, if a mutation from amino acid A to amino acid B yields a stability change of $\Delta\Delta G(A, B)$, the reverse mutation ($B \rightarrow A$) should result in the opposite change, such that $\Delta\Delta G(B, A) = -\Delta\Delta G(A, B)$. Failures in satisfying this property would imply an inconsistency in the underlying free energy landscape, violating thermodynamic constraints and potentially leading to unrealistic predictions. Thus, enforcing antisymmetry in $\Delta\Delta G$ estimates is fundamental to preserve the physical validity of the model.

Antisymmetry has already been used in the development of stability prediction, either in weak forms, by using training sets augmented by means of naive antisymmetry[38] or by thermodynamic permutation[39], or in strong forms by designing the model to be an odd function[40]. Here, we adopt a siamese neural network architecture, as illustrated in Fig. 1b. Starting from the trained model, which outputs a directional prediction $\Delta\Delta G_{\mathrm{DIR}}$, we construct a mirrored input by swapping the wild-type and mutant sequences to produce a second prediction, $\Delta\Delta G_{\mathrm{INV}}$. The final antisymmetric prediction is then obtained by averaging the two in opposite directions:

$$\Delta\Delta G = (\Delta\Delta G_{DIR} - \Delta\Delta G_{INV})/2. \tag{5}$$

To evaluate the impact of enforcing antisymmetry, we assessed JanusDDG on the S669 test dataset (see Supplementary Note 1 for details). Even without applying the antisymmetry constraint, the model already exhibited a strong inverse correlation between direct and reverse predictions, with a Pearson correlation coefficient $PCC_{d\text{-}r} = -0.95$, and mean absolute deviation from perfect antisymmetry $\langle\delta\rangle = 0.02$. However, after introducing the antisymmetry-enforcing modification, the model achieved perfect antisymmetric behavior, i.e. $PCC_{d\text{-}r} = -1.00$, $\langle\delta\rangle = 0.00$. When evaluated on the hard Ssym benchmark[24], JanusDDG maintained its antisymmetric performance, while most of the state-of-the-art methods failed to meet this criterion, as shown in Supplementary Table 4.

In addition to enhancing the antisymmetry of the model, this procedure also improves its overall performance, as shown in Supplementary Tables 1, 2 and 3.

Transitivity. Transitivity implies that, if we know $\Delta\Delta G(A, B)$ and $\Delta\Delta G(B, C)$, we can find $\Delta\Delta G(A, C)$ by summing these two quantities:

$$\begin{aligned}\Delta\Delta G(A,C) &= \Delta G(A) - \Delta G(C)\\ &= \Delta G(A) - \Delta G(B) + \Delta G(B) - \Delta G(C)\\ &= \Delta\Delta G(A,B) + \Delta\Delta G(B,C).\end{aligned} \tag{6}$$

Thus, a fine-tuning procedure was applied to encourage the model to comply with this property (Fig. 1c). Specifically, JanusDDG was further trained on additional epochs with the introduction of an extra loss function, designed to train the prediction of $\Delta\Delta G(A, B)$ between the wild-type protein and the mutant, transitioning through an additional protein state. The predicted value should match the true $\Delta\Delta G(A, B)_{TRUE}$:

$$\begin{aligned}L_{TOT} = {} & L[\Delta\Delta G(A,B) - \Delta\Delta G(A,B)_{TRUE}]\\ & + L\{\Delta\Delta G(A,B) -\\ & [\Delta\Delta G(A,X) + \Delta\Delta G(X,B)]\}.\end{aligned} \tag{7}$$

Further technical details are provided in the Methods section.

To assess the extent to which JanusDDG satisfies the transitivity property, we evaluated the model on both the S669 dataset by introducing random intermediate residues, and the $S^{transitive}$ dataset, which was specifically developed to evaluate transitivity in $\Delta\Delta G$ predictions[41]. Transitivity was quantified by computing the Pearson correlation between the direct prediction $\Delta\Delta G(A, B)$ and the corresponding transitive prediction obtained by transitioning through a variable number of random amino acids (1, 3, 5, 7, or 9) between the residues A and B. The results of this evaluation are reported in Fig. 2 (molecular graphics were generated using PyMOL[42]). Interestingly, the explicit incorporation of antisymmetry into the base model already improves transitivity, suggesting a synergistic relationship between these two fundamental thermodynamic constraints. The subsequent fine-tuning with the transitivity loss further amplifies this effect, bringing the model's behavior into even closer alignment with the expected transitive properties.

JanusDDG's performance in protein stability prediction

To evaluate the model performance on both single and multiple mutations, we used different benchmark datasets. It is well established that including proteins in the test set that are highly similar to those in the training set can lead to inflated performance estimates[10,25]. Despite this, many studies continue to report results without properly controlling for sequence identity. To avoid this issue and ensure a fair evaluation, we assessed JanusDDG on test datasets in which no proteins share more than 25% sequence identity with those in the training set. This strict filtering allows for an unbiased estimate of the model's generalization capability. Results on additional datasets with more relaxed similarity criteria are provided in the Supplementary Note 2 for completeness. The importance of removing sequence similarity is further demonstrated in Supplementary Note 3, where performance appears inflated, with a Pearson correlation of 0.87 and a Top-3 Mean Sequence Identity of 41%, compared with a Pearson correlation of 0.71 and a Top-3 Mean Sequence Identity of 11% under our strict procedure. Although this value is high, it is entirely attributable to sequence identity leakage. Unfortunately, many methods still do not check for this issue when reporting their results.

Performance on single and multiple mutations. For an evaluation that attempts to remove sequence identity, we selected three datasets sharing less than 25% sequence similarity with the JanusDDG training set: S669[25], S461[9], and S96. The first two are among the most widely used benchmarks for this task, while the third is a manually curated dataset composed of proteins distinct from the other two. As such, it provides an additional independent benchmark for assessing the model's generalization, introduced by Montanucci, L. et al.[33].

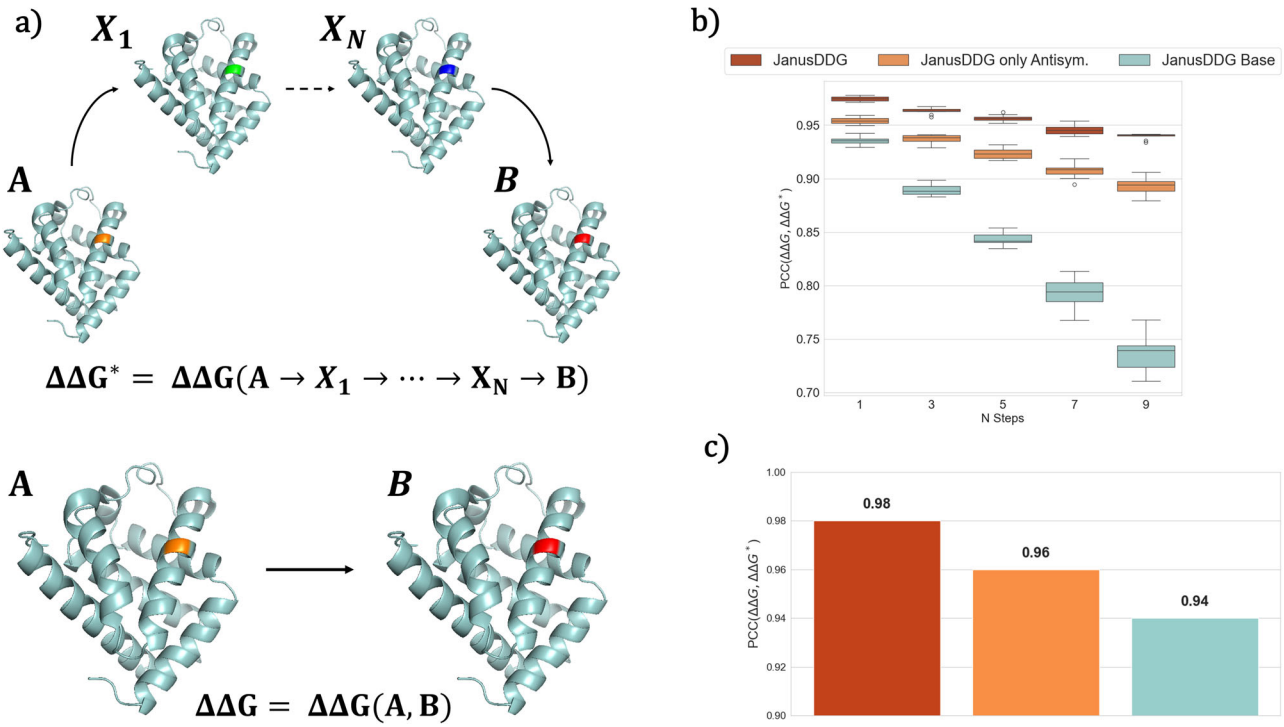

Fig. 2 | Results of the transitivity evaluation of JanusDDG. a Illustration of the transitivity property: Since $\Delta\Delta G$ depends only on the initial and final states, $\Delta\Delta G(A, B)$ should be equal to $\Delta\Delta G^*(A, B)$, where the latter is computed by summing the $\Delta\Delta G$ values of multiple intermediate mutations from step 1 to N. The protein figures have been created using PyMOL[42]. **b** Pearson correlation results between $\Delta\Delta G$ and $\Delta\Delta G^*$, calculated on S669 (n = 669 independent mutations) for different intermediate steps (1, 3, 5, 7, and 9). For each step, the Pearson correlation was computed 10 times for three different models: JanusDDG Base (the model without antisymmetry and fine-tuning), JanusDDG only Antisym. (the model with antisymmetry but without fine-tuning), and JanusDDG (the final model, incorporating both antisymmetry and fine-tuning). The boxes represent the interquartile ranges (IQR) of the Pearson correlation distributions for each step and model variant. The horizontal lines within the boxes indicate the medians. The whiskers extend to the most extreme data points within 1.5 times the IQR, showing the variability of the predictions. **c** Transitivity performance, evaluated on the external dataset $S^{\text{transitive}}$[41] (n = 1603 independent mutations), for all three models. It should be noted that the protein figures are illustrative only and are not restricted to a single mutation position.

We then compared the performance of JanusDDG using the scores reported for the same datasets in other recent studies[23,33]. The results are displayed in Fig. 3 for S669, S461, and S96. Across all three datasets, JanusDDG achieved performance comparable to or exceeding that from existing sequence-based models and structure-informed predictors, despite relying solely on sequence information. More detailed results are reported in Supplementary Tables 5, 6 and 7. It is worth noting that in Fig. 3 several zero-shot methods yield predictions that correlate significantly with $\Delta\Delta G$ values, despite not being trained for this specific task. Nevertheless, their performance is generally lower than that of methods specifically trained for the task at hand. For completeness, Supplementary Tables 8 and 9 present the model's performance on additional datasets. However, these datasets do not meet the stringent similarity criteria required to infer generalizability.

Predicting the stability effects of multiple simultaneous mutations is notably more challenging than single-point mutations, due to potential epistatic interactions. To evaluate JanusDDG in this setting, we used the PTmul-NR dataset[23], which contains proteins with a varying number of mutations and no close homologs in the training set. Obtained results are shown in Fig. 4 and Supplementary Table 10. Although PTmul-D does not meet the required sequence similarity criteria, the results obtained on this dataset are reported in Supplementary Table 11.

In addition, Supplementary Note 1 presents a performance comparison on the S669, S461, and PTmul-NR datasets between JanusDDG, its baseline version (non-antisymmetric and without fine-tuning), and the antisymmetric-only variant (without fine-tuning). As with single mutations, JanusDDG achieves state-of-the-art performance, matching the best-performing methods and demonstrating its ability to generalize to the more complex task of multiple-mutation stability prediction, while consistently preserving physics-informed thermodynamic properties.

Distance analysis for double mutations. It has been observed that deep learning models tend to perform better when the distance between mutated residues is large, as the resulting $\Delta\Delta G$ values exhibit greater additivity[32].

In Fig. 5, we analyzed the performance of JanusDDG as a function of the Euclidean 3D distance between mutated residues and their sequence separation, respectively. The absolute error between the predicted and experimental $\Delta\Delta G$ values, measured on the PTmul-D dataset, was used as an evaluation metric. Interestingly, no significant differences in performance were observed when JanusDDG was evaluated on double mutations in terms of both spatial proximity and sequence separation, while tests on previous different models[32] reported otherwise.

Performance evaluation for stability classification. The ability to identify stabilizing mutations is relevant in protein engineering applications. However, many computational models struggle with this task: recent studies report that current sequence-based predictors correctly identify only about 20% of stabilizing mutations, with a high rate of false positives[12]. More accurate approaches, such as free energy perturbation based on molecular dynamics, achieve better classification performance (around 50% success) but are computationally expensive and not scalable for tasks like deep mutational scanning[43].

In order to evaluate the performance of JanusDDG concerning this specific capability, we considered the S461 dataset. Given that the average

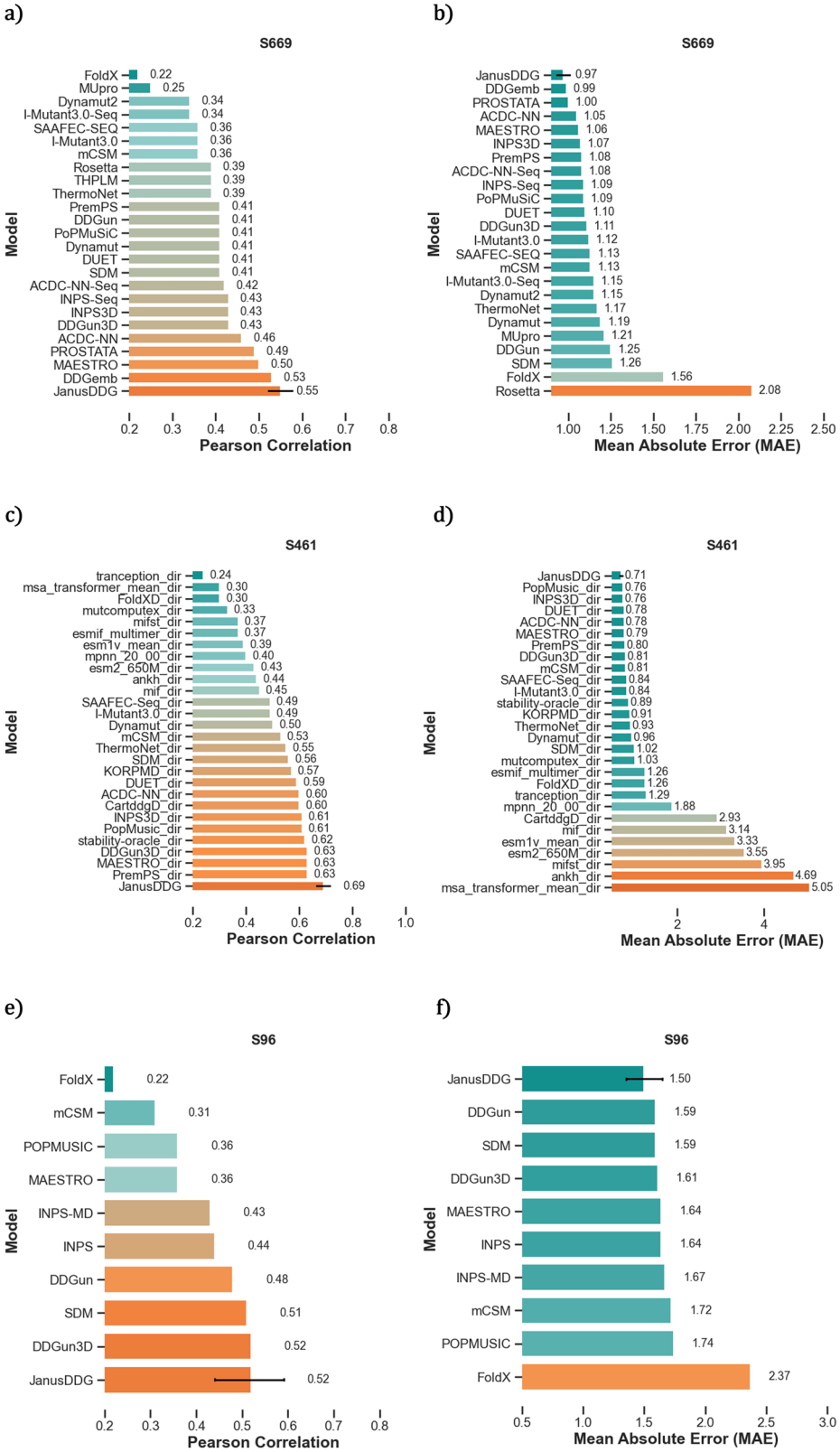

Fig. 3 | Pearson Correlation and Mean Absolute Error on Single-Point Mutation Benchmarks. Panels show the Pearson correlation coefficient (**a**, **c**, **e**) and Mean Absolute Error (MAE) (**b**, **d**, **f**). **a**, **b** Results on the S669 test set ($n = 669$ independent mutations); comparative model data (excluding JanusDDG) were taken from Savojardo, C. et al.[23]. **c**, **d** Results on the S461 test set ($n = 461$ independent mutations); comparative data taken from Reeves, S. & Kalyaanamoorthy, S.[26]. **e**, **f** Results on the S96 test set ($n = 96$ independent mutations); comparative data taken from Montanucci, L. et al.[33]. For all panels, JanusDDG error bars represent the standard error, estimated as described in Supplementary Note 6.

experimental error of $\Delta\Delta G$ is usually assumed to be around 0.5 kcal/mol, we defined stabilizing mutations as those with $\Delta\Delta G > 0.5$ kcal/mol, neutral mutations as those with -0.5 kcal/mol $< \Delta\Delta G < 0.5$ kcal/mol, and destabilizing mutations as those with $\Delta\Delta G < -0.5$ kcal/mol[38,39]. As a preliminary step, we excluded neutral mutations. We then selected the top five models with the highest Pearson correlation on S461, and then evaluated their performance for stability classification (Fig. 6). JanusDDG and DDGemb perform well across all metrics on this dataset and tend to outperform the other models, although the precision score indicates that predicting stabilizing variants remains challenging.

The comparison between $\Delta\Delta G$ regression and classification highlights fundamental challenges in protein stability prediction. On the S461 dataset, JanusDDG reaches a Pearson correlation of 0.69 in regression but only a balanced accuracy of 0.75 in binary classification. This reflects the

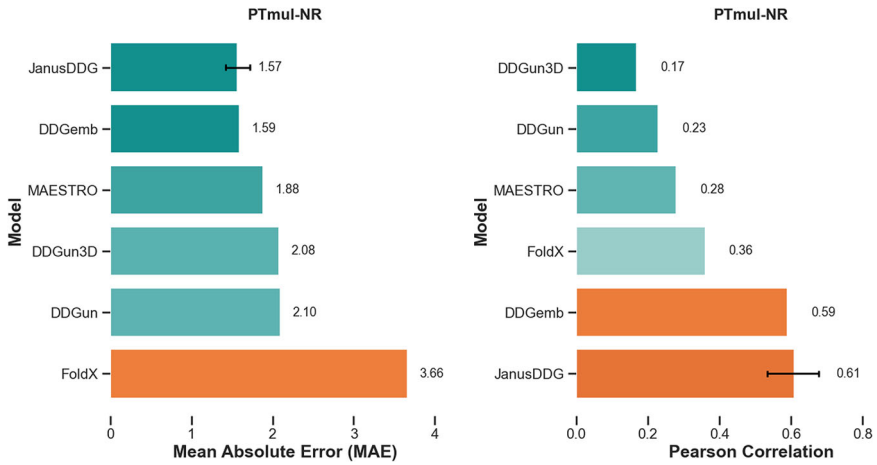

Fig. 4 | Pearson Correlation and MAE on Multi-Point Mutation Benchmark. Pearson correlation and MAE on PTmul-NR test set ($n = 82$ independent mutations). The models' performance data, excluding JanusDDG, were taken from Savojardo, C. et al.[23]. JanusDDG error bars represent the standard error, estimated as described in Supplementary Note 6.

intrinsic difficulty of classification rather than superior regression performance.

The classification task faces several inherent challenges. Sharp decision boundaries amplify the impact of small prediction errors, particularly near thresholds where biological ambiguity is highest. Additionally, the natural prevalence of destabilizing mutations creates a strong class imbalance that reduces discriminative performance, as the feature space cannot clearly separate positive and negative classes even after filtering.

The regression approach benefits from physics-informed constraints that enforce thermodynamic consistency, including antisymmetry and transitivity across continuous $\Delta\Delta G$ predictions. These constraints help the model learn smooth, physically coherent energy landscapes. In contrast, classification only applies these constraints indirectly, and binarization during thresholding leads to information loss that undermines thermodynamic consistency.

Discussion

In this work, we introduced JanusDDG, a sequence-based deep learning model for predicting protein stability changes upon mutation. JanusDDG effectively captures both contextual and differential information between wild-type and mutant sequences by integrating protein language model embeddings with a bidirectional cross-attention architecture. The model is also thermodynamically compliant by design, enforcing antisymmetry and promoting transitivity through targeted fine-tuning.

Our benchmarking on datasets with low sequence identity to the training set demonstrates that JanusDDG consistently matches or outperforms existing sequence-based predictors, as well as several structure-informed models. Since JanusDDG operates exclusively on sequence data, these results highlight its broad applicability to proteins lacking reliable structural information. Furthermore, JanusDDG generalizes effectively to the more challenging task of predicting the effects of multiple mutations, which is an area where many current models still exhibit limited performance.

Our model consistently shows strong performance on established test datasets. However, the scarcity and noisiness of publicly available datasets make it difficult to draw statistically robust conclusions about the absolute superiority of one model over another. Consequently, most comparative studies refrain from reporting error bars to avoid overstating uncertainty, suggesting that current methods perform similarly on average.

A key enabler of JanusDDG performance is the use of embeddings from the ESM-2[35] language model, which provide rich contextual information derived from large-scale evolutionary data. However, this also represents a critical dependency: the quality and resolution of JanusDDG predictions are inherently linked to the expressive power of the underlying ESM-2 embeddings. As such, improvements in protein language models could directly enhance the predictive capabilities of JanusDDG, while any biases or limitations in the embeddings may propagate through the system and affect downstream predictions.

An important final consideration in benchmarking protein stability predictors is the degree of sequence similarity between training and test sets. High sequence identity can lead to inflated performance estimates driven by memorization rather than true generalization. In this study, JanusDDG was rigorously evaluated on benchmarks filtered to exclude sequences sharing more than 25% identity with the training data. In contrast, many existing methods, including some recent ones, do not explicitly control for this bias. As a result, their reported performance may not accurately reflect real-world applicability, particularly when predicting stability effects for novel or evolutionarily distant proteins. Addressing this issue is critical for developing models that are both robust and broadly generalizable across diverse protein families.

Methods

Datasets

In this subsection, we report the datasets used for training, validation, and testing our model for both single and multiple mutations.

Training and validation datasets. The datasets underpinning the training and validation of JanusDDG are detailed below. The S2450[23] dataset served as the foundational resource for training JanusDDG, being used both during the initial training phase and the subsequent fine-tuning phase to enhance its predictive performance. In contrast, the M28[33] dataset was used exclusively as an independent validation set during fine-tuning. Specifically, it was employed to monitor performance across epochs and select the optimal number of fine-tuning epochs based on validation correlation, thereby serving as the stopping criterion.

S2450. The S2450 dataset, introduced by Savojardo, C.[23], is a refined version of the established S2648 dataset[8] and originates from a collection of 2648 single amino acid substitutions across 131 distinct proteins. These mutations have experimentally determined $\Delta\Delta G$ values obtained from the ProTherm database[44]. While S2648 was created to have low similarity with S669 using sequences from the PDB[25], the sequence identity of S2450 was re-evaluated using full-length UniProt sequences. Any protein in S2648 exhibiting more than 25% sequence identity with a protein in S669 was excluded. This rigorous filtering process resulted in the removal of 18 proteins, encompassing 198 individual mutations, ultimately yielding the S2450 dataset utilized as training set. To balance this dataset between stabilizing and destabilizing mutations, we used the antisymmetry property: $\Delta\Delta G(B, A) = -\Delta\Delta G(A, B)$ to double the dataset and make it less imbalanced in terms of mutation stability.

M28. The M28 dataset, a collection of multiple-site variants, was constructed using the 2021 version of ProTherm[45]. Its selection criteria

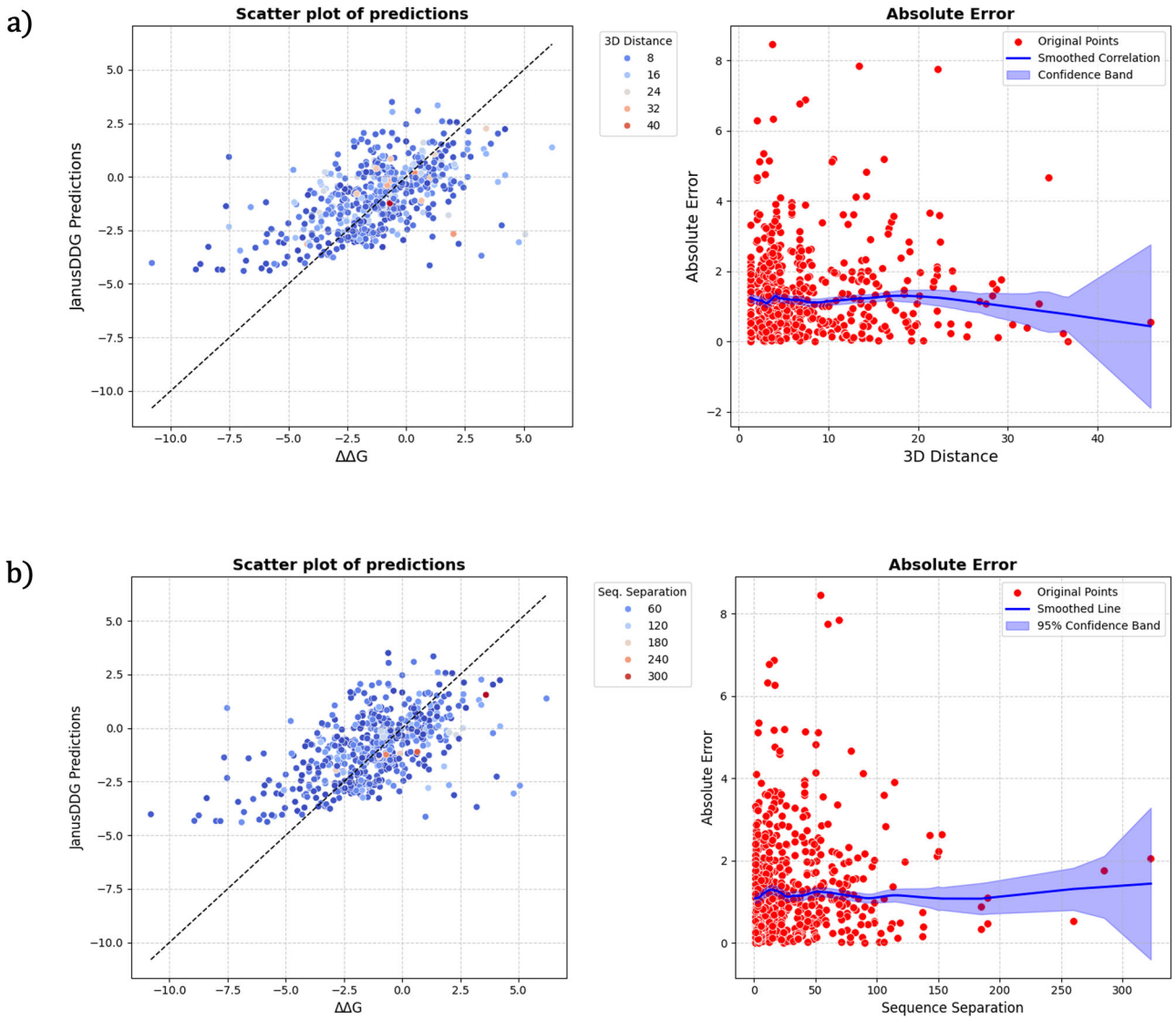

Fig. 5 | Distance Analysis for Double Mutations. The analysis considers the distance between mutated residues defined as (**a**) 3D spatial Euclidean distance and (**b**) sequence separation. In both sub-figures, the scatter plot of the predictions shows the correlation between predicted and observed $\Delta\Delta G$ values, colored by the respective distance metric. The Absolute Error panel displays the absolute error for each double mutation (red dots) along with a LOWESS-smoothed fitted curve (blue line). Shaded regions represent 95% confidence intervals computed using a local statistical approach based on Student's *t*-distribution ($t_{\alpha/2,n-1} \times \sigma_{local}/\sqrt{n}$). Bands are narrower in dense regions (low uncertainty) and wider in sparse regions; for windows with fewer than 3 points, a conservative error of $2 \times \sigma_{global}$ was applied. The dataset used for this analysis is PTmul-D (n = 583 independent mutations).

specifically targeted variants with experimental $\Delta\Delta G$ or $\Delta\Delta G_{H_2O}$ values reported after 2013. In this dataset, when multiple experimental $\Delta\Delta G$ values were reported for the same variant, the average was considered.

Test datasets. The following details the blind datasets employed to evaluate JanusDDG and to provide a comparative analysis of its performance against other models. These datasets are specifically composed of proteins exhibiting low sequence similarity (less than 25%) with the training set, since, as shown in Supplementary Table 12, high sequence similarity would bias the results regarding model generalization.

S669. The S669 dataset[25] is widely recognized as a benchmark for scoring protein stability predictors. The strength of this dataset lies in its construction: it exhibits low sequence redundancy (below 25% identity) compared to common training datasets like S2648[8] and VariBench[46], thus facilitating unbiased comparisons. Including 1338 single-site mutations, both direct and reverse, across 95 protein chains, S669 provides experimentally determined $\Delta\Delta G$ values, which were retrieved from ThermoMutDB[47] and manually verified.

S461. The S461 dataset[9] is another widely used dataset to measure the performance on protein stability by predictors. This curated dataset addressed some inaccuracies present in the original S669 and excluded mutations potentially involved in natural protein function, such as those at oligomer interfaces. The S461 dataset encompasses experimental structures for 48 wild-type proteins, with a range of 1 to 68 mutations per protein, for a total of 461 mutations, each with a single experimental $\Delta\Delta G$ measurement.

S96. Comprising 96 single-site variants across 14 distinct proteins, the S96 dataset[33] was assembled using the 2021 version of ProTherm[45] as its source. Each variant within this dataset was subjected to a rigorous manual checking and correction process, informed by the experimental data presented in the corresponding research articles. Furthermore, to

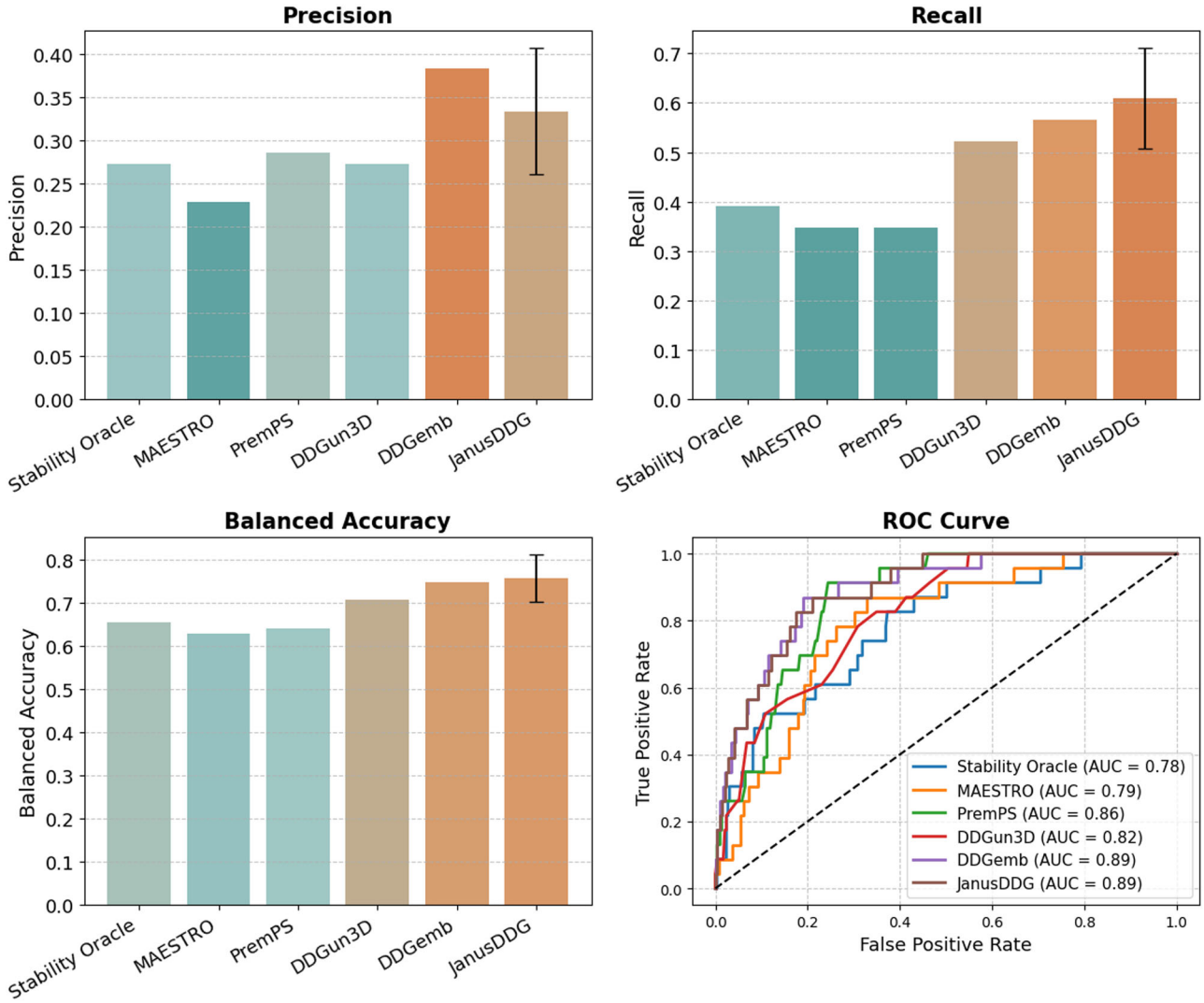

Fig. 6 | Performance Evaluation for Stability Classification. Performance of the top 4 models—based on Pearson correlation on the S461 dataset (n = 461 independent mutations)—in predicting the stability of mutated proteins, with the addition of DDGemb and JanusDDG. JanusDDG error bars represent the standard error, estimated as described in Supplementary Note 6. The evaluation includes recall, precision, balanced accuracy, and AUC, as illustrated by the ROC curves. Performance data for the models, excluding JanusDDG and DDGemb, were taken from Reeves, S. & Kalyaanamoorthy[26]. The DDGemb predictions were obtained from its official website: https://ddgemb.biocomp.unibo.it.

ensure independence from commonly used training data, only those variants whose parent proteins showed less than 25% sequence identity to any protein in the S2648 and VariBench datasets were included in S96. In this dataset, when multiple experimental $\Delta\Delta G$ values were reported for the same variant, the average was considered.

PTmul-NR. The PTmul-NR[23] dataset is a carefully curated subset derived from the original PTmul[48], specifically designed to assess model performance in predicting $\Delta\Delta G$ for multipoint mutations, particularly under conditions of low sequence similarity with the S2450 dataset. The original PTmul dataset, which includes 914 multipoint variations across 91 proteins, exhibited substantial sequence overlap with our S2450 training data. As a result of a rigorous removal procedure, the PTmul-NR dataset was created, consisting of 82 multipoint variants across 14 proteins. While this reduction significantly decreased the number of variants, it was crucial to ensure a robust comparison across different methods.

Other datasets. We evaluated JanusDDG on additional datasets to facilitate a comparative analysis of its performance against other methods documented in the literature. Unlike the strictly blind test sets previously discussed, these datasets may include proteins with sequence similarity exceeding 25% to our training data. This potential overlap could influence the observed performance, possibly leading to an overestimation of the model's true capability on unseen data. Consequently, the results obtained from these datasets, reported in Supplementary Note 2, have to be interpreted with respect to the findings observed from the previous test sets. For details on these datasets, please refer to the cited papers. The datasets are as follows: (i) PTmul-D, a dataset derived from PTmul, filtered to include only double mutations; (ii) K2369, a dataset containing high sequence identity with S2450, as defined in Reeves, S. & Kalyaanamoorthy, S.[26]; (iii) Q3421, another dataset with high sequence identity to S2450, as defined in Reeves, S. & Kalyaanamoorthy, S.[26]; (iv) Ssym, a dataset generated to test antisymmetry based on protein structure[24]; (v) $S^{\text{transitive}}$, a dataset designed to evaluate transitivity of the prediction methods[41].

For further information about the datasets, please refer to Supplementary Table 15.

Performance metrics

To assess the model's regression performance, we used the following metrics, which are among the most commonly used for this purpose.

Pearson correlation coefficient. It measures the linear relationship between two variables X and Y and is defined as:

$$r_p = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}},$$

where $\bar{X}$ and $\bar{Y}$ are the means of X and Y, respectively.

Spearman correlation coefficient. It evaluates the monotonic relationship between two variables X and Y using ranked values. The Spearman rank correlation coefficient r_s is defined as the Pearson correlation coefficient between the ranked variables:

$$r_s = \frac{\sum_{i=1}^{n}(R_{X_i} - \bar{R}_X)(R_{Y_i} - \bar{R}_Y)}{\sqrt{\sum_{i=1}^{n}(R_{X_i} - \bar{R}_X)^2}\sqrt{\sum_{i=1}^{n}(R_{Y_i} - \bar{R}_Y)^2}},$$

where R_{X_i} and R_{Y_i} are the ranks of the values X_i and Y_i, and $\bar{R}_X$, $\bar{R}_Y$ are the mean ranks.

Root mean square error (RMSE). It quantifies the average squared difference between predicted values $\hat{y}_i$ and observed values y_i as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}.$$

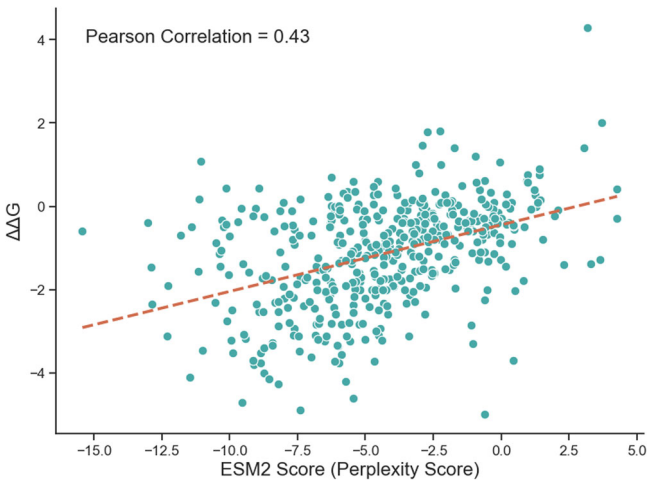

Fig. 7 | Relationship between perplexity values and protein stability. Scatter plot illustrating the relationship between perplexity values and protein stability. The x-axis represents the perplexity score derived from the language model, while the y-axis displays the experimentally measured $\Delta\Delta G$ (kcal/mol). The regression line highlights the correlation between the model's prediction uncertainty and the thermodynamic stability of the mutations. The dataset used for this analysis is S461 (n = 461 independent mutations).

Mean absolute error (MAE). It measures the average of absolute differences between predictions and observations:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|.$$

Furthermore, we adopted two additional metrics to assess anti-symmetry properties[24].

Pearson correlation coefficient between direct and reverse predictions. *It measures the Pearson correlation coefficient between* p_{dir} and p_{rev}, referred to as PCC_{d-r}, and is defined as:

$$\mathrm{PCC}_{d-r} = \mathrm{PCC}(p_{\mathrm{dir}}, p_{\mathrm{rev}}),$$

where p_{dir} and p_{rev} are the predicted values in the direct and reverse directions, respectively.

Anti-symmetry bias. The bias ($\langle\delta\rangle$), which quantifies the average deviation from perfect antisymmetry between p_{dir} and p_{rev}, is computed as:

$$\langle\delta\rangle = \frac{\sum_{i=1}^{N}(p_{i,\mathrm{dir}} + p_{i,\mathrm{rev}})}{N},$$

where N is the total number of observations.

Lastly, to evaluate classification performance in identifying stabilizing mutations, we used the following metrics.

Recall (or Sensitivity). It measures the proportion of actual positive cases that are correctly identified by the model. It is defined as:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$

where TP represents true positives and FN false negatives.

Precision. It quantifies the proportion of positive predictions that are actually correct. It is calculated as:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$

where FP denotes false positives.

Balanced accuracy. It accounts for class imbalance by averaging the recall of each class:

$$\mathrm{Balanced\ Accuracy} = \frac{\mathrm{Sensitivity} + \mathrm{Specificity}}{2},$$

where specificity is given by:

$$\mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}.$$

ROC curve and AUC. The Receiver Operating Characteristic (ROC) curve plots the true positive rate (sensitivity) against the false positive rate at various threshold settings. The Area Under the Curve (AUC) quantifies the overall performance, with a value of 1 indicating a perfect classifier and 0.5 representing a random model.

Protein embedding

In this subsection, we show some characteristics of the protein embedding that we used as model input.

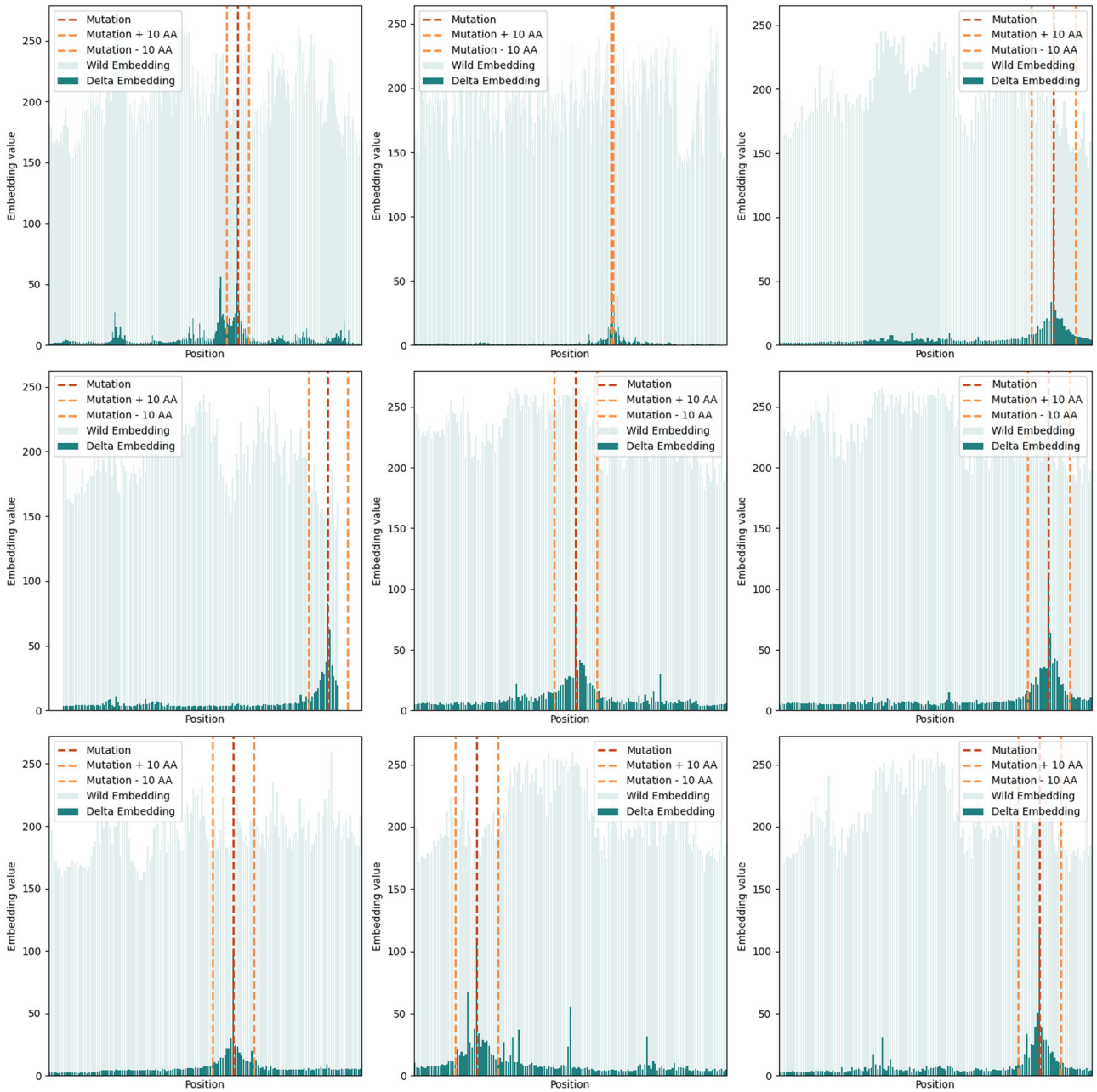

Fig. 8 | Analysis of embedding perturbations induced by mutations. The graphs illustrate nine examples (from the S2450 dataset) of the element-wise absolute difference between the wild-type and mutated protein embeddings, compared against the absolute sum of the wild-type embedding elements. As observed, the deviation between the mutated and wild-type representations is strictly localized around the mutation site, indicating that the model captures the local effect of the variation without altering the global context.

The embedding used to describe the proteins was obtained from ESM2 with 650M parameters[35]. Beyond their representational capabilities, large language models offer additional metrics that can inform stability prediction. Perplexity, which quantifies a model's uncertainty over sequence predictions, emerges as a particularly relevant measure for protein stability assessment. This metric reflects the model's confidence in predicting the next amino acid given the preceding context, with high perplexity values potentially indicating structurally atypical or unstable regions, while low perplexity corresponds to evolutionarily conserved, stable motifs. The connection between perplexity and structural stability provides a principled framework for model selection in protein analysis tasks. Accordingly, we employed ESM2-650M, as it has been shown to be among the top-performing sequence-based PLMs for $\Delta\Delta G$ prediction in multiple studies, yielding competitive zero-shot performance across several benchmark datasets[26]. To illustrate the relationship between perplexity and protein stability, Fig. 7 presents a scatter plot correlating the perplexity values with the experimental $\Delta\Delta G$ measurements. This approach establishes perplexity as a complementary feature that enhances the interpretability of LLM-derived embeddings and strengthens their application to stability estimation problems.

We analyzed these representations to better understand the differences between the embeddings of the wild-type and mutated proteins. The Fig. 8 presents graphs comparing the element-wise absolute difference between the wild-type and mutated protein embeddings with the absolute sum of the elements in the wild-type embedding. The difference between the mutated and wild-type protein is almost entirely localized around the mutation site.

To test whether the embedding used is suitable for the prediction task, we tried using the difference between the two embeddings (this time not in

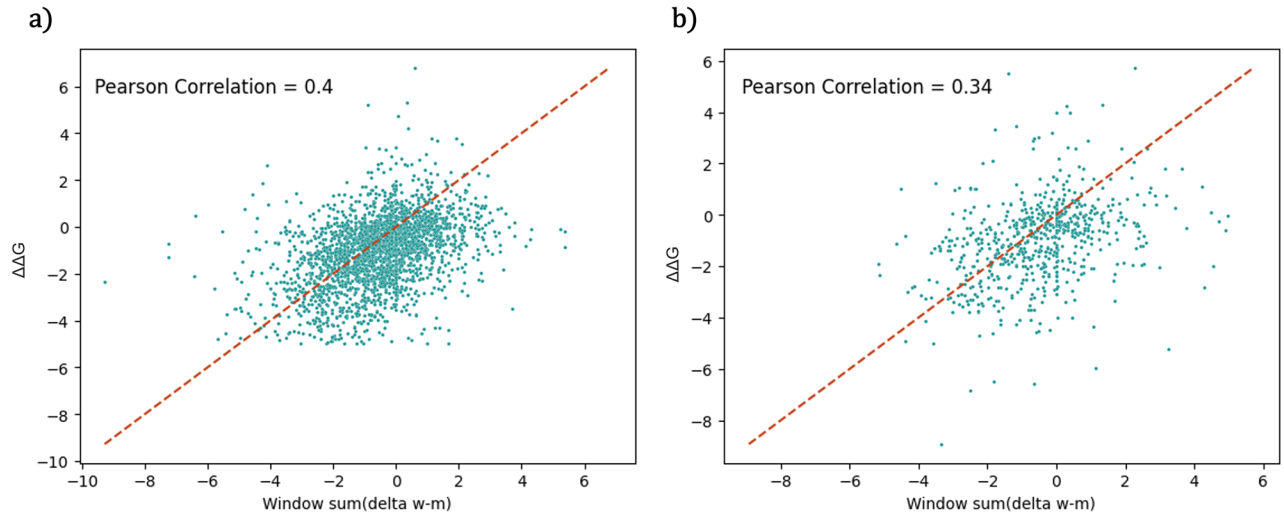

Fig. 9 | Predictive Signal of Simple Embedding Differences. Pearson correlation between the sum of the element-wise differences in ESM2 embeddings of the wild-type and mutant proteins and the ΔΔG values. **a** Results calculated using the S2450 dataset (n = 2450 independent mutations). **b** Results calculated using the S669 dataset (n = 669 independent mutations).

absolute value) to predict the ΔΔG. This operation is quite naive; therefore, we used only a window of size 20 around the mutation instead of the entire sequence, as this window contains most of the information about the difference between the wild-type sequence and the mutated sequence. The results are shown in Fig. 9. The Pearson correlation achieved on the training and test sets is high, suggesting that ESM2 can be considered a valid model to extract the input to be processed by the network. Nonetheless, since our model is capable of processing information in a much more complex way than a simple mathematical operation, the difference between the embeddings across the entire sequence has been used in the final version of the model in order to preserve information and make it more generalizable to a variable number of mutations.

Model architecture: JanusDDG

The architecture of our model, shown in Fig. 1a, is based on the integration of protein language models with the cross-attention mechanism[49]. The strength of this model is mainly due to two factors: it uniquely relies on the protein sequence (allowing it to predict the stability of proteins whose structure is unknown) and can be applied regardless of their number of mutations. This section explains the various building blocks that constitute it.

Input. As input to the model, we used the embedding of the two sequences (wild-type and mutated) obtained from ESM2 650M parameters. These embeddings were also subtracted to derive a third embedding that represents the difference between the two. The model takes as input both the wild-type embedding and the difference embedding.

Conv1D. Once the input is fed into the model, two 1D convolutions (one for each input) are applied to identify patterns within the sequences while simultaneously reducing their dimensionality. The default Torch parameters were used for the convolution, except for the kernel size, which was set to 20 based on the embedding trend shown in Fig. 8.

Bidirectional cross attention transformer. The core of the model is the Bidirectional Cross-Attention Transformer[49]. The classic Transformer block consists of the following components: a Multihead Self-Attention block, residual connections, normalization layers and a position-wise feed-forward network (FFN). Our proposed model retains all these components except for the first one.

We substituted the standard Multihead Self-Attention block with two Cross-Attention blocks, whose configuration is bidirectional: one block utilizes the difference embedding as Queries (Q) and Values (V) while the wild-type embedding serves as Keys (K); the second block operates in the reverse direction, swapping these roles.

The Attention layer is unable to capture the position of each element in the sequence; therefore, a sinusoidal positional embedding is added to the input before applying the Bidirectional Multihead Cross-Attention blocks[50]. After applying the two Bidirectional Multihead Cross-Attention blocks, each output is summed with the input of its respective block. Then, the two outputs are concatenated along the feature dimension before being passed into a position-wise FFN.

Bidirectional cross-attention. The standard self-attention mechanism[50] enables each element in a sequence to attend to all others within the same sequence. In contrast, bidirectional cross-attention extends standard cross-attention by enabling mutual information flow between two input sequences, rather than a one-directional mapping. This mechanism enhances the model's ability to capture deep interdependencies between two entities.

More specifically, cross-attention is a variant of the attention mechanism where the query $\mathbf{Q}$ comes from one sequence, while the key $\mathbf{K}$ and value $\mathbf{V}$ come from another sequence. Given two input sequences $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathbb{R}^{m \times d}$, their corresponding projections are:

$$\mathbf{Q}_X = \mathbf{X}W_Q,\ \mathbf{K}_Y = \mathbf{Y}W_K,\ \mathbf{V}_Y = \mathbf{Y}W_V; \tag{8}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$ are learnable weight matrices.

The attention scores are computed using the scaled dot-product:

$$\mathrm{Attn}\left(\mathbf{Q}_X, \mathbf{K}_Y, \mathbf{V}_Y\right) = \mathrm{softmax}\left(\frac{\mathbf{Q}_X\mathbf{K}_Y^{\top}}{\sqrt{d_k}}\right)\mathbf{V}_Y. \tag{9}$$

This allows sequence $\mathbf{X}$ to attend to sequence $\mathbf{Y}$. However, this formulation is inherently asymmetric, meaning that sequence $\mathbf{Y}$ does not simultaneously attend to $\mathbf{X}$.

To capture mutual dependencies, we introduce bidirectional cross-attention, where both sequences $\mathbf{X}$ and $\mathbf{Y}$ attend to each other. This is

achieved by computing attention in both directions:

$$\mathbf{H}_X = \text{softmax}\left(\frac{\mathbf{Q}_X\mathbf{K}_Y^\top}{\sqrt{d_k}}\right)\mathbf{V}_Y, \tag{10}$$

$$\mathbf{H}_Y = \text{softmax}\left(\frac{\mathbf{Q}_Y\mathbf{K}_X^\top}{\sqrt{d_k}}\right)\mathbf{V}_X, \tag{11}$$

where: $\mathbf{H}_X$ represents the updated representation of $\mathbf{X}$ attending to $\mathbf{Y}$, and $\mathbf{H}_Y$ represents the updated representation of $\mathbf{Y}$ attending to $\mathbf{X}$.

The final representations are then combined through concatenation:

$$\mathbf{Z} = \text{Cat}(\mathbf{H}_X, \mathbf{H}_Y). \tag{12}$$

Pooling layers: GAP and GMP. The output from the previous layer is processed using two different pooling operations: Global Average Pooling and Global Max Pooling. Global Average Pooling computes the average value of each feature map, reducing the spatial dimensions while preserving the overall feature distribution. On the other hand, Global Max Pooling selects the maximum value from each feature map, capturing the most prominent activations. These two operations help to refine the most relevant information before passing the representation to the subsequent layers. The outputs of these two layers are concatenated and then passed to the final layer.

Linear layer. To obtain the final $\Delta\Delta G$ value, a linear layer with a single output neuron is applied.

Model capacity and regularization strategies

A critical aspect of JanusDDG architecture is the balance between parameter count and dataset size ($N = 2450 \times 2 = 4900$). While the complete model contains approximately 7 million trainable parameters, a structural analysis reveals that the effective capacity available for complex reasoning (and potentially overfitting) is tightly constrained.

Dimensionality management. The vast majority of the parameter count (> 90%) is concentrated in the initial feature projection stage. Specifically, processing the high-dimensional embeddings from ESM2 ($d = 1280$) requires two independent Conv1D layers (one for the wild-type and one for the difference embedding). With a kernel size of 20 and a reduction to 128 channels, these projection layers alone account for approximately 6.55×10^6 parameters ($2 \times 1280 \times 128 \times 20$). This computational cost is strictly necessary to project the rich semantic space of the protein language model into a compact feature space ($d = 128$) manageable by the attention mechanism.

Lightweight reasoning core. In contrast, the core reasoning module, i.e. the Bidirectional Cross-Attention Transformer, is intentionally parsimonious. By selecting a configuration with 8 heads, an input dimension of 128, and a feed-forward network size of 512, the transformer blocks comprise fewer than 0.5×10^6 parameters. This design ensures that the non-linear logic of the model remains low-capacity, significantly reducing the risk of memorizing the training data noise.

Physics-informed regularization. To further mitigate overfitting, JanusDDG employs structural regularization via Physics-Informed Neural Network constraints. The Siamese architecture enforces exact antisymmetry ($\Delta\Delta G_{dir} = -\Delta\Delta G_{inv}$), effectively halving the degrees of freedom for the output function. Additionally, the transitivity fine-tuning restricts the solution space to physically consistent energy landscapes. These inductive biases act as powerful regularizers that are arguably more effective than standard weight penalties in small-data regimes.

Validation protocol. Finally, generalizability is enforced through a strict training protocol. We employed a fold-specific early stopping mechanism, monitored on validation sets with low sequence identity (< 25%) relative to the training folds. This ensures that training is halted at the point of maximum generalization rather than minimum training error.

Training

JanusDDG was trained in two distinct phases: a main training phase followed by a fine-tuning phase. Throughout both phases, the model parameters were optimized using the Adam optimizer with Mean Squared Error (MSE) serving as the loss function and a batch size of 6.

Main training phase. During the main training phase of JanusDDG, the model was trained on the S2450 dataset augmented with its inverses. The number of training epochs was set to 300, determined via 5-fold cross-validation. In this procedure, we identified the optimal epoch for each fold as the one yielding the maximum Pearson correlation coefficient on the respective validation set. The final count of 300 epochs represents the average of these optimal epoch numbers across the five folds

$$E^* = \frac{1}{5}\sum_{i=1}^{5} \arg\max_E PCC(E).$$

An alternative approach would be to select the final epoch as

$$E^* = \arg\max_E \sum_{i=1}^{5} PCC(E).$$

However, this strategy precludes the use of early stopping. By contrast, our approach employs a fold-specific early stopping mechanism, ensuring that each model is trained up to its point of maximum generalization. This provides a more reliable and robust estimate of performance across the entire dataset.

The model training was conducted on a single NVIDIA GeForce RTX 3090 graphics processing unit (GPU), equipped with 24 GB of dedicated video memory. Leveraging the computational capabilities of this hardware, the training process achieved an average execution time of approximately 2 minutes per epoch. Switching the workload to CPU-only processing (Intel Core i9-12900F) led to a drastic increase in training time, reaching approximately 25 minutes per epoch. This comparison demonstrates a speedup factor of roughly 12.5x when utilizing the GPU compared to the CPU baseline.

Fine-tuning procedure. After the main training phase, we performed a fine-tuning procedure to maximize the performance of JanusDDG with respect to the following two tasks: predicting multiple mutations and compliance with the above-mentioned transitivity property. The transitivity property is one of the two fundamental properties of $\Delta\Delta G$. It states that:

$$\Delta\Delta G(A, B) = \Delta\Delta G(A, C) + \Delta\Delta G(C, B). \tag{13}$$

To enforce transitivity during fine-tuning, we introduce a dedicated two-step loss function (see Fig. 1c) that incorporates a null intermediate state. The second term of the loss, $\text{MSE}(\Delta\Delta G(A, B) - (\Delta\Delta G(A, U) + \Delta\Delta G(U, B)))$, imposes thermodynamic transitivity by introducing a null intermediate state U, represented by a zero vector ($\vec{0}$). While a zero vector does not correspond to a physical protein sequence, it serves as the optimal dataset-independent reference (the origin of the ESM2 latent space). Unlike using an average sequence embedding, which would inherently carry biases related to the specific training distribution, the zero vector acts as a featureless pivot. From an architectural perspective, calculating the transition $A \to \vec{0}$ forces

the network to process the embedding of A against a null difference, effectively training the model to estimate the absolute stability potential of state A in isolation. Consequently, the transitivity constraint encourages the model to decompose the relative $\Delta\Delta G(A, B)$ into consistent state functions relative to this absolute origin, acting as a strict information bottleneck that prevents the learning of spurious sequence correlations (see Supplementary Note 4 for details).

As training dataset, we used S2450 and its inverse, as in the previous training phase. To choose the number of epochs for fine-tuning, we tracked the Pearson correlation on the validation set for each epoch over a short fine-tuning period corresponding to 10% of the main training phase (30 epochs). The fine-tuning step is intentionally limited to enforce transitivity while preserving the model's primary predictive capabilities. Based on this validation tracking, the final selected epoch was 28.

Hyperparameter selection

Given the long training time of the model and the large number of hyperparameters to be tuned, we focused on selecting those related to the transformer component. Specifically, we tested three different configurations with increasing complexity, and selected the one that, for comparable performance and error, required the fewest parameters: 8 transformer heads, a position-wise feed-forward network size of 512, and an input dimension of 128. The results of the three configurations are reported in Supplementary Table 13. The mean squared error was the loss function with the standard Adam optimizer for the learning rate. To determine the optimal number of training epochs, we performed a 5-fold cross-validation on the training set. We adopted the five folds defined by Savojardo, C.[23], where MMSeqs2[51] was used to partition the training data into five subsets, each consisting of proteins with similar sequences. This procedure ensured that proteins sharing more than 25% sequence identity were grouped into the same subset. The final number of training epochs was set to 300, corresponding to the average of the best-performing epoch across the five folds.

Furthermore, in Supplementary Note 5 and in Supplementary Table 14 the results of an ablation study relating to the final part of the model are reported, which demonstrates the usefulness of concatenating both poolings used.

Statistics and reproducibility

The model was trained and validated using the S2450 (n = 2450) and M28 (n = 28) datasets, respectively, while blind testing was performed on independent benchmarks including S669 (n = 669), S461 (n = 461), S96 (n = 96), and PTmul-NR (n = 82). Sample sizes were determined based on data availability in the literature rather than pre-determined statistical power calculations. To ensure robust generalization and prevent data leakage, proteins sharing more than 25% sequence identity between training and test sets were strictly excluded. Consequently, the training data was partitioned into five independent folds using MMSeqs2 to cluster homologous sequences, ensuring that no cluster spanned across folds. We employed a 5-fold cross-validation protocol to optimize hyperparameters and determine the final training duration of 300 epochs, calculated as the average optimal epoch across folds. Statistical performance was evaluated using Pearson correlation, Root Mean Squared Error, and Mean Absolute Error. Unless otherwise stated, error bars for regression metrics represent the 95% Confidence Interval estimated via Fisher's z-transformation. Conversely, uncertainty for classification metrics was estimated using a non-parametric bootstrap procedure with 1000 resamples. Finally, to guarantee full reproducibility, the source code, trained model weights, and datasets have been made publicly available.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The datasets generated and/or analysed during the current study are available in the JanusDDG GitHub repository. This includes the training dataset, the validation set, and the independent test sets. Source data are available at: https://github.com/compbiomed-unito/JanusDDGand archived on Zenodo[52] (https://doi.org/10.5281/zenodo.18170934.

Code availability

The source code for JanusDDG, including the trained model weights and instructions for use, is available on GitHub at: https://github.com/compbiomed-unito/JanusDDGand archived on Zenodo[52] (https://doi.org/10.5281/zenodo.18170934.

Received: 4 August 2025; Accepted: 22 January 2026;
Published online: 03 February 2026

References

1. Capriotti, E., Fariselli, P. & Casadio, R. A neural-network-based method for predicting protein stability changes upon single point mutations. *Bioinformatics* **20**, i63–i68 (2004).
2. Mehra, R. & Kepp, K. P. Computational analysis of alzheimer-causing mutations in amyloid precursor protein and presenilin 1. *Arch. Biochem. Biophysics* **678**, 108168 (2019).
3. Bahia, M. S. et al. Stability prediction for mutations in the cytosolic domains of cystic fibrosis transmembrane conductance regulator. *J. Chem. Inf. Modeling* **61**, 1762–1777 (2021).
4. Meghwanshi, G. K. et al. Enzymes for pharmaceutical and therapeutic applications. *Biotechnol. Appl. Biochem.* **67**, 586–601 (2020).
5. Gebauer, M. & Skerra, A. Engineered protein scaffolds as next-generation therapeutics. *Annu. Rev. Pharmacol. Toxicol.* **60**, 391–415 (2020).
6. Shi, J., Yuan, B., Yang, H. & Sun, Z. Recent advances on protein engineering for improved stability. *BioDesign Res.* 100005 (2025).
7. Delgado, J. et al. FoldX force field revisited, an improved version. *Bioinforma. (Oxf., Engl.)* **41**, btaf064 (2025).
8. Dehouck, Y., Kwasigroch, J. M., Gilis, D. & Rooman, M. Popmusic 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. *BMC Bioinforma.* **12**, 1–12 (2011).
9. Hernández, I. M., Dehouck, Y., Bastolla, U., López-Blanco, J. R. & Chacón, P. Predicting protein stability changes upon mutation using a simple orientational potential. *Bioinformatics* **39**, btad011 (2023).
10. Sanavia, T. et al. Limitations and challenges in protein stability prediction upon genome variations: towards future applications in precision medicine. *Comput. Struct. Biotechnol. J.* **18**, 1968–1979 (2020).
11. Pancotti, C. et al. A Deep-Learning Sequence-Based Method to Predict Protein Stability Changes Upon Genetic Variations. *Genes* **12**, 911 (2021).
12. Benevenuta, S., Birolo, G., Sanavia, T., Capriotti, E. & Fariselli, P. Challenges in predicting stabilizing variations: An exploration. *Front. Mol. Biosci.* **9**, 1075570 (2023).
13. Umerenkov, D. et al. PROSTATA: a framework for protein stability assessment using transformers. *Bioinforma. (Oxf., Engl.)* **39**, btad671 (2023).
14. Yang, K. K., Zanichelli, N. & Yeh, H. Masked inverse folding with sequence transfer for protein representation learning. *Protein Eng., Des. selection: PEDS* **36**, gzad015 (2023).
15. Jiang, F. et al. A general temperature-guided language model to design proteins of enhanced stability and activity. *Sci. Adv.* **10**, eadr2641 (2024).
16. Li, G., Yao, S. & Fan, L. ProSTAGE: Predicting Effects of Mutations on Protein Stability by Using Protein Embeddings and Graph Convolutional Networks. *J. Chem. Inf. Modeling* **64**, 340–347 (2024).

17. Cuturello, F., Celoria, M., Ansuini, A. & Cazzaniga, A. Enhancing predictions of protein stability changes induced by single mutations using MSA-based language models. *Bioinformatics* **40**, btae447 (2024).
18. Dieckhaus, H., Brocidiacono, M., Randolph, N. Z. & Kuhlman, B. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proc. Natl. Acad. Sci.* **121**, e2314853121 (2024).
19. Chu, S. K. S., Narang, K. & Siegel, J. B. Protein stability prediction by fine-tuning a protein language model on a mega-scale dataset. *PLoS Comput. Biol.* **20**, e1012248 (2024).
20. Sun, J., Zhu, T., Cui, Y. & Wu, B. Structure-based self-supervised learning enables ultrafast protein stability prediction upon mutation. *Innov. (Camb. (Mass.))* **6**, 100750 (2025).
21. Chen, J.-Y. et al. A Comprehensive Review of Protein Language Models arXiv:2502.06881 [q-bio] version: 1. (2025).
22. Ruffolo, J. A. & Madani, A. Designing proteins with language models. *Nat. Biotechnol.* **42**, 200–202 (2024).
23. Savojardo, C., Manfredi, M., Martelli, P. L. & Casadio, R. Ddgemb: predicting protein stability change upon single-and multi-point variations with embeddings and deep learning. *Bioinformatics* **41**, btaf019 (2025).
24. Pucci, F., Bernaerts, K. V., Kwasigroch, J. M. & Rooman, M. Quantification of biases in predictions of protein stability changes upon mutations. *Bioinformatics* **34**, 3659–3665 (2018).
25. Pancotti, C. et al. Predicting protein stability changes upon single-point mutation: a thorough comparison of the available tools on a new dataset. *Brief. Bioinforma.* **23**, bbab555 (2022).
26. Reeves, S. & Kalyaanamoorthy, S. Zero-shot transfer of protein sequence likelihood models to thermostability prediction. *Nat. Mach. Intell.* **6**, 1063–1076 (2024).
27. Rollo, C. et al. Influence of model structures on predictors of protein stability changes from single-point mutations. *Genes* **14**, 2228 (2023).
28. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
29. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
30. Ahdritz, G. et al. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *Nat. Methods* **21**, 1514–1524 (2024).
31. Ouyang-Zhang, J., Diaz, D. J., Klivans, A. R. & Krähenbühl, P. Predicting a Protein's Stability under a Million Mutations arXiv:2310.12979 [q-bio]. (2023).
32. Dieckhaus, H. & Kuhlman, B. Protein stability models fail to capture epistatic interactions of double point mutations. *Protein Sci.* **34**, e70003 (2025).
33. Montanucci, L. et al. Ddgun: an untrained predictor of protein stability changes upon amino acid variants. *Nucleic Acids Res.* **50**, W222–W227 (2022).
34. Chen, Y., Xu, Y., Liu, D., Xing, Y. & Gong, H. An end-to-end framework for the prediction of protein structure and fitness from single sequence. *Nat. Commun.* **15**, 7400 (2024).
35. Lin, Z. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
36. Yu, T. et al. Enzyme function prediction using contrastive learning. *Science* **379**, 1358–1363 (2023).
37. Cuomo, S. et al. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *J. Sci. Comput.* **92**, 88 (2022).
38. Capriotti, E., Fariselli, P., Rossi, I. & Casadio, R. A three-state prediction of single point mutations on protein stability changes. *BMC Bioinforma.* **9**, 1–9 (2008).
39. Diaz, D. J. et al. Stability Oracle: a structure-based graph-transformer framework for identifying stabilizing mutations. *Nat. Commun.* **15**, 6170 (2024).
40. Benevenuta, S., Pancotti, C., Fariselli, P., Birolo, G. & Sanavia, T. An antisymmetric neural network to predict free energy changes in protein variants. *J. Phys. D: Appl. Phys.* **54**, 245403 (2021).
41. Samaga, Y. B. L., Raghunathan, S. & Priyakumar, U. D. SCONES: Self-Consistent Neural Network for Protein Stability Prediction Upon Mutation. *J. Phys. Chem. B* **125**, 10657–10671 (2021).
42. Schrödinger, L.L.C. The PyMOL molecular graphics system, version 1.8 (2015).
43. Broom, A., Trainor, K., Jacobi, Z. & Meiering, E. M. Computational modeling of protein stability: quantitative analysis reveals solutions to pervasive problems. *Structure* **28**, 717–726 (2020).
44. Bava, K. A., Gromiha, M. M., Uedaira, H., Kitajima, K. & Sarai, A. Protherm, version 4.0: thermodynamic database for proteins and mutants. *Nucleic Acids Res.* **32**, D120–D121 (2004).
45. Nikam, R., Kulandaisamy, A., Harini, K., Sharma, D. & Gromiha, M. M. Prothermdb: thermodynamic database for proteins and mutants revisited after 15 years. *Nucleic Acids Res.* **49**, D420–D424 (2021).
46. Nair, P. S. & Vihinen, M. V ari b ench: A benchmark database for variations. *Hum. Mutat.* **34**, 42–49 (2013).
47. Xavier, J. S. et al. Thermomutdb: a thermodynamic database for missense mutations. *Nucleic Acids Res.* **49**, D475–D479 (2021).
48. Montanucci, L., Capriotti, E., Frank, Y., Ben-Tal, N. & Fariselli, P. Ddgun: an untrained method for the prediction of protein stability changes upon single and multiple point variations. *BMC Bioinforma.* **20**, 1–10 (2019).
49. Bishop, C. M. & Bishop, H. Deep learning - foundations and concepts https://doi.org/10.1007/978-3-031-45468-4 (2024).
50. Vaswani, A. et al. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6000–6010 (Curran Associates Inc., Red Hook, NY, USA, 2017).
51. Steinegger, M. & Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017).
52. Barducci, G. Janusddg https://doi.org/10.5281/zenodo.18170934 (2026).

Acknowledgements

We thank the Fondazione Compagnia di San Paolo for supporting the ON-AIR project, of which this work is a part. The authors also acknowledge support from the Italian Ministry of University and Research through the "Grant for Internationalization" program. T.S. acknowledges support from the PRIN project "Investigating the role of NF-YA isoform/lncRNA axis in mesoderm specification" (Grant ID: 20224TWKNJ).

Author contributions

G.B. and P.F. conceived the original idea. G.B. designed the study, developed the JanusDDG architecture and software, performed the experiments, analyzed the data, and wrote the original draft. P.F. supervised the project and contributed to the interpretation of the results. F.C. assisted with the Python code implementation and management. V.I. assisted in the identification and curation of the datasets. I.R. contributed to the theoretical framework and assisted in reviewing and editing the manuscript. C.R., V.R., C.P., and T.S. provided critical advice on the theoretical implementation. V.R. also contributed to the manuscript revision. All authors reviewed and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42003-026-09632-9.

Correspondence and requests for materials should be addressed to Guido Barducci or Piero Fariselli.

Peer review information *Communications Biology* thanks Robert Schmirler, David Wagemann, Yunxin Xu alongside their co-reviewer Shize Yu, and the other, anonymous, reviewer for their contribution to the peer review of this work. Primary Handling Editor: George Inglis. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2026