

JanusDDG is A Physics-Informed Neural Network for Sequence-Based Protein Stability via Two-Fronts Attention

Guido Barducci^{1,*}, Ivan Rossi¹, Francesco Codicè¹, Cesare Rollo¹, Valeria Repetto^{1,2}, Corrado Pancotti³, Virginia Iannibelli¹, Tiziana Sanavia¹, and Piero Fariselli^{1,*}

¹*AI and Computational Biomedicine Unit, Dept. 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.*

^{*}*Corresponding authors. Emails: piero.fariselli@unito.it, guido.barducci@unito.it*

Supplementary Note 1: Comparison Between JanusDDG Base, JanusDDG only Antisym., and JanusDDG

JanusDDG Base, JanusDDG only Antisym., and JanusDDG models were evaluated on the blind test sets. The assessment includes both single and multiple mutations, providing a comprehensive overview of how each model handles different mutation scenarios. JanusDDG only Antisym. is the antisymmetric model by construction, derived from the JanusDDG Base to enforce antisymmetry in its predictions. JanusDDG is the fine-tuned version of the base model.

Supplementary Table 1: Comparison Between JanusDDG Base, JanusDDG only Antisym., and JanusDDG on the S669 independent test set of single-point variations.

Method	Input	Total			Direct			Reverse			PCC _{d-r}	$\langle\delta\rangle$
		PCC	RMSE	MAE	PCC	RMSE	MAE	PCC	RMSE	MAE		
JanusDDG	SEQ	0.69	1.39	0.97	0.55	1.39	0.97	0.55	1.39	0.97	-1	0.00
JanusDDG only Antisym.	SEQ	0.69	1.37	0.96	0.55	1.37	0.96	0.55	1.37	0.96	-1	0.00
JanusDDG Base	SEQ	0.69	1.38	0.96	0.55	1.37	0.96	0.53	1.39	0.97	-0.95	0.02

Supplementary Table 2: Comparison Between JanusDDG Base, JanusDDG only Antisym., and JanusDDG on the S461 independent test set of single-point variations.

Method	Input	Total			Direct			Reverse			PCC _{d-r}	$\langle\delta\rangle$
		PCC	RMSE	MAE	PCC	RMSE	MAE	PCC	RMSE	MAE		
JanusDDG	SEQ	0.83	0.96	0.71	0.69	0.96	0.71	0.69	0.96	0.71	-1	0.00
JanusDDG only Antisym.	SEQ	0.84	0.92	0.70	0.70	0.92	0.70	0.70	0.92	0.70	-1	0.00
JanusDDG Base	SEQ	0.84	0.93	0.71	0.70	0.92	0.70	0.68	0.95	0.72	-0.95	0.00

Supplementary Table 3: Comparison Between JanusDDG Base, JanusDDG only Antisym., and JanusDDG on the PTmul-NR independent test set of multipoint variations.

Method	Input	Total			Direct			Reverse			PCC _{d-r}	$\langle\delta\rangle$
		PCC	RMSE	MAE	PCC	RMSE	MAE	PCC	RMSE	MAE		
JanusDDG	SEQ	0.61	2.06	1.57	0.61	2.06	1.57	0.61	2.06	1.57	-1	0.00
JanusDDG only Antisym.	SEQ	0.55	2.17	1.67	0.55	2.17	1.67	0.55	2.17	1.67	-1	0.00
JanusDDG Base	SEQ	0.54	2.18	1.67	0.55	2.17	1.63	0.54	2.19	1.71	-0.96	-0.09

Supplementary Note 2: Performance of JanusDDG Across Benchmark Datasets

This section reports the performance of JanusDDG alongside other state-of-the-art methods from the literature. It is important to note that, unless explicitly stated, all methods—including JanusDDG—may be affected by sequence identity bias, which can lead to overestimated performance when training and test sets share homologous proteins.

Supplementary Table 4: Performance on the Ssym benchmark. Note that all methods, including JanusDDG, likely share some sequence similarity between the test and training sets. Performance data for all models except JanusDDG are taken from Reeves, S. & Kalyaanamoorthy, S.¹.

Dir	Direct				Inverse				Antisymmetry	
	Pearson	Spearman	RMSE	MAE	Pearson	Spearman	RMSE	MAE	PCC _{d-r}	$\langle\delta\rangle$
JanusDDG	0.84	0.84	0.85	0.55	0.84	0.84	0.85	0.55	-1	0.00
KORPM	0.57	0.61	1.28	0.92	0.49	0.51	1.38	0.98	-0.88	-0.11
mpnn_20_00	0.58	0.63	2.45	2.02	0.41	0.50	2.41	1.76	-0.58	-1.40
Cartddg	0.67	0.71	3.32	2.66	0.45	0.43	3.55	2.60	-0.41	-3.13
ACDC-NN	0.61	0.53	1.35	0.99	0.58	0.51	1.41	1.02	-0.98	-0.05
stability-oracle	0.65	0.69	1.35	0.93	0.42	0.43	1.74	1.24	-0.57	-0.50
mifst	0.47	0.47	5.52	4.55	0.31	0.30	4.13	3.23	-0.74	-2.10
msa_transformer_mean	0.35	0.32	5.44	4.52	0.35	0.32	5.44	4.52	-1.00	0.00
mif	0.56	0.54	4.72	3.71	0.35	0.38	3.76	2.78	-0.44	-3.46
esm2_650M	0.28	0.30	5.75	4.89	0.28	0.30	5.75	4.89	-1.00	0.00
ankh	0.28	0.28	6.09	5.30	0.28	0.28	6.09	5.30	-1.00	0.00
tranception	0.27	0.25	1.80	1.30	0.27	0.25	1.80	1.30	-1.00	0.00
esmif_multimer	0.55	0.51	1.79	1.30	0.14	0.23	1.82	1.32	-0.17	-0.03
DDGun3D	0.57	0.47	1.38	0.98	0.54	0.46	1.42	1.01	-0.99	-0.04
FoldX	0.56	0.67	1.92	1.14	0.37	0.39	2.14	1.45	-0.25	-1.12
Evo	0.57	0.53	1.35	1.00	0.31	0.30	1.74	1.25	-0.58	-0.36
mutcomputex	0.45	0.48	1.47	1.01	0.15	0.21	1.93	1.37	-0.19	-0.70
INPS3D	0.61	0.58	1.23	0.87	0.28	0.13	1.92	1.43	-0.51	-1.02
esm1v_mean	0.12	0.16	3.65	2.39	0.12	0.16	3.65	2.39	-1.00	0.00
ThermoNet	0.45	0.40	1.54	1.08	0.37	0.31	1.64	1.14	-0.85	-0.04
MAESTRO	0.58	0.61	1.29	0.88	0.25	0.22	2.15	1.64	-0.33	-1.25
DUET	0.63	0.62	1.22	0.87	0.16	0.11	2.27	1.73	-0.30	-1.49
I-Mutant3.0	0.65	0.69	1.18	0.75	-0.06	-0.09	2.30	1.74	0.00	-1.37
MUpro	0.81	0.80	0.91	0.50	0.04	-0.00	2.49	2.01	-0.02	-1.93
mCSM	0.61	0.57	1.23	0.91	0.13	0.05	2.41	1.90	-0.26	-1.82
SDM	0.50	0.51	1.55	1.20	0.16	0.13	2.32	1.78	-0.43	-1.09
Dynamut	0.55	0.50	1.44	1.02	0.35	0.36	1.73	1.23	-0.57	-0.25

Supplementary Table 5: Performance comparison of sequence- and structure-based methods on the S669 independent test set of single-point mutations. **No sequence identity is shared between this test set and the JanusDDG training data.** Performance values for all methods except JanusDDG are taken from Savojardo, C. et al.².

Method	Input	Total			Direct			Reverse			PCC _{d-r}	$\langle\delta\rangle$
		PCC	RMSE	MAE	PCC	RMSE	MAE	PCC	RMSE	MAE		
JanusDDG	SEQ	0.69	1.39	0.97	0.55	1.39	0.97	0.55	1.39	0.97	-1	0.00
DDGemb	SEQ	0.68	1.40	0.99	0.53	1.40	0.99	0.52	1.40	0.99	-0.97	0.01
PROSTATA	SEQ	0.65	1.45	1.00	0.49	1.45	1.00	0.49	1.45	0.99	-0.99	-0.01
ACDC-NN	3D	0.61	1.50	1.05	0.46	1.49	1.05	0.45	1.50	1.06	-0.98	0.02
INPS-Seq	SEQ	0.61	1.52	1.10	0.43	1.52	1.09	0.43	1.53	1.10	-1.00	0.00
PremPS	3D	0.62	1.49	1.07	0.41	1.50	1.08	0.42	1.49	1.05	-0.85	0.09
ACDC-NN-Seq	SEQ	0.59	1.53	1.08	0.42	1.53	1.08	0.42	1.53	1.08	-1.00	0.00
DDGun3D	3D	0.57	1.61	1.13	0.43	1.60	1.11	0.41	1.62	1.14	-0.97	0.05
INPS3D	3D	0.55	1.64	1.19	0.43	1.50	1.07	0.33	1.77	1.31	-0.50	0.38
THPLM	SEQ	0.53	1.63	-	0.39	1.60	-	0.35	1.66	-	-0.96	-0.01
ThermoNet	3D	0.51	1.64	1.20	0.39	1.62	1.17	0.38	1.66	1.23	-0.85	0.05
DDGun	SEQ	0.57	1.74	1.25	0.41	1.72	1.25	0.38	1.75	1.25	-0.96	0.05
MAESTRO	3D	0.44	1.80	1.30	0.50	1.44	1.06	0.20	2.10	1.66	-0.22	0.57
ThermoMPNN	SEQ	0.43	1.52	-	-	-	-	-	-	-	-	-
Dynamut	3D	0.50	1.65	1.21	0.41	1.60	1.19	0.34	1.69	1.24	-0.58	0.06
PoPMuSiC	3D	0.46	1.82	1.37	0.41	1.51	1.09	0.24	2.09	1.64	-0.32	0.69
DUET	3D	0.41	1.86	1.39	0.41	1.52	1.10	0.23	2.14	1.68	-0.12	0.67
I-Mutant3.0-Seq	SEQ	0.37	1.91	1.47	0.34	1.54	1.15	0.22	2.22	1.79	-0.48	0.76
SDM	3D	0.32	1.93	1.45	0.41	1.67	1.26	0.13	2.16	1.64	-0.40	0.40
mCSM	3D	0.37	1.96	1.49	0.36	1.54	1.13	0.22	2.30	1.86	-0.05	0.85
Dynamut2	3D	0.36	1.90	1.42	0.34	1.58	1.15	0.17	2.16	1.69	0.03	0.64
I-Mutant3.0	3D	0.32	1.96	1.49	0.36	1.52	1.12	0.15	2.32	1.87	-0.06	0.81
Rosetta	3D	0.47	2.69	2.05	0.39	2.70	2.08	0.40	2.68	2.02	-0.72	0.61
FoldX	3D	0.31	2.39	1.53	0.22	2.30	1.56	0.22	2.48	1.50	-0.20	0.34
SAAFEC-SEQ	SEQ	0.26	2.02	1.54	0.36	1.54	1.13	-0.01	2.40	1.94	-0.03	0.83
MUpro	SEQ	0.32	2.03	1.58	0.25	1.61	1.21	0.20	2.38	1.96	-0.32	0.95

Supplementary Table 6: Comparative benchmark of different sequence- and structure-based methods on the S461 independent test set of single-point variations. **No sequence identity is shared between this test set and the JanusDDG training data.** The data used to compute model performance, excluding JanusDDG, were taken from Reeves, S. & Kalyaanamoorthy, S. ¹.

Model	Pearson	Spearman	RMSE	MAE
JanusDDG	0.69	0.66	0.96	0.71
Stability Oracle	0.61	0.63	1.19	0.89
CartDDG-D	0.60	0.61	3.59	2.93
PremPS	0.63	0.60	1.03	0.80
PopMusic	0.61	0.60	1.02	0.76
MAESTRO	0.63	0.60	1.04	0.79
INPS3D	0.61	0.59	1.02	0.76
DDGun3D	0.63	0.58	1.11	0.81
DUET	0.59	0.57	1.06	0.78
ACDC-NN	0.60	0.56	1.06	0.78
KORPMD	0.57	0.54	1.21	0.91
mCSM	0.53	0.51	1.07	0.81
SDM	0.56	0.51	1.33	1.02
ThermoNet	0.55	0.48	1.24	0.93
I-Mutant3.0	0.49	0.47	1.12	0.84
SAAFEC-Seq	0.49	0.47	1.12	0.84
MIF	0.45	0.46	4.37	3.14
Ankh	0.44	0.44	5.60	4.69
ESM2-650M	0.43	0.44	4.41	3.55
Dynamut	0.50	0.43	1.27	0.96
MPNN-20-00	0.40	0.43	2.36	1.88
ESM1v Mean	0.39	0.43	4.29	3.33
ESMIF Multimer	0.37	0.41	1.64	1.26
MIFST	0.37	0.38	5.02	3.95
MutComputeX	0.33	0.36	1.39	1.03
FoldX-D	0.30	0.39	1.91	1.26
Tranception	0.24	0.27	1.68	1.29
MSA Transformer Mean	0.30	0.26	5.84	5.05

Supplementary Table 7: Performance comparison on S96. **No sequence identity is shared between this test set and the JanusDDG training data.** The data used to compute model performance, excluding JanusDDG, were taken from Montanucci, L. et al. ³.

Model	Pearson	Spearman	RMSE	MAE
JanusDDG	0.52	0.50	2.10	1.50
DDGun	0.48	0.44	2.14	1.59
DDGun3D	0.52	0.48	2.10	1.61
INPS-MD	0.43	0.37	2.21	1.67
Maestro	0.36	0.36	2.29	1.64
mCSM	0.31	0.33	2.33	1.72
FoldX	0.22	0.38	4.18	2.37
INPS	0.44	0.41	2.20	1.64
POPMUSIC	0.36	0.33	2.29	1.74
SDM	0.51	0.47	2.12	1.59

Supplementary Table 8: Performance comparison of different models on on K2369. Note that all methods, including JanusDDG, likely share some sequence similarity between the test and training sets. The models’ performance data, excluding JanusDDG, were taken from Reeves, S. & Kalyanamoorthy, S. ¹.

Model Type	Model	MSE	Accuracy	ρ	$w\rho$	NDCG	wNDCG
unknown	$\Delta\Delta G_u$ label	0.00 ± 0.00	1	1	1	1	1
sequence	JanusDDG	1.14	0.72	0.70	0.56	0.87	0.83
ensemble	Ensemble 6 Feats*	1.52 ± 0.36	0.73	0.66	0.5	0.81	0.75
ensemble	Ensemble 5 Feats*	1.53 ± 0.36	0.73	0.65	0.51	0.82	0.74
ensemble	Ensemble 7 Feats*	1.53 ± 0.36	0.73	0.66	0.5	0.81	0.74
ensemble	Ensemble 4 Feats*	1.58 ± 0.38	0.72	0.65	0.51	0.82	0.75
transfer	Stability Oracle	1.61 ± 0.17	0.7	0.59	0.48	0.75	0.76
ensemble	Ensemble 3 Feats*	1.70 ± 0.41	0.72	0.59	0.45	0.81	0.74
potential	KORPM*	1.72 ± 0.35	0.71	0.55	0.44	0.78	0.74
ensemble	Ensemble 2 Feats*	1.96 ± 0.49	0.69	0.51	0.39	0.8	0.71
struc. PSLM	ESM-IF	2.95 ± 0.89	0.65	0.4	0.41	0.76	0.71
seq. PSLM	Tranception (reduced)	3.03 ± 0.89	0.6	0.31	0.24	0.71	0.68
seq. PSLM	Tranception	3.03 ± 0.89	0.61	0.32	0.24	0.71	0.69
unknown	Gaussian Noise	3.59 ± 0.61	0.53	-0.02	-0.02	0.65	0.59
struc. PSLM	ProteinMPNN 0.3	7.38 ± 1.42	0.66	0.47	0.42	0.81	0.73
struc. PSLM	ProteinMPNN 0.2	7.86 ± 1.54	0.67	0.47	0.42	0.78	0.73
struc. PSLM	ProteinMPNN 0.1	8.44 ± 1.49	0.66	0.47	0.4	0.79	0.73
seq. PSLM	ESM-2 150M	22.0 ± 5.61	0.62	0.24	0.27	0.76	0.69
seq. PSLM	ESM-1V mean	26.9 ± 3.87	0.63	0.26	0.28	0.76	0.68
seq. PSLM	ESM-2 650M	29.4 ± 4.54	0.63	0.32	0.3	0.75	0.7
struc. PSLM	MIF	30.7 ± 6.37	0.65	0.46	0.42	0.77	0.7
biophysical	Rosetta CartDDG	32.4 ± 3.52	0.7	0.61	0.45	0.8	0.73
seq. PSLM	MSA-T mean	32.7 ± 5.98	0.63	0.36	0.27	0.73	0.69
struc. PSLM	MIF-ST	34.6 ± 3.23	0.64	0.45	0.38	0.77	0.71
seq. PSLM	Ankh	37.7 ± 3.53	0.63	0.36	0.25	0.72	0.68
seq. PSLM	ESM-2 3B	39.7 ± 4.22	0.62	0.32	0.31	0.71	0.69
seq. PSLM	ESM-2 15B	46.0 ± 3.54	0.62	0.36	0.28	0.73	0.68
struc. PSLM	Rosetta/ProtMPNN	66.1 ± 7.09	0.69	0.65	0.53	0.83	0.75

Supplementary Table 9: Performance on Q3421 with Alternative Choices for Statistics. Note that all methods, including JanusDDG, likely share some sequence similarity between the test and training sets. The models’ performance data, excluding JanusDDG, were taken from Reeves, S. & Kalyaanamoorthy, S.¹.

Model Type	Model	MSE	Accuracy	Spearman’s ρ	w ρ	NDCG	wNDCG
unknown	$\Delta\Delta$ Gu label	0.00 ± 0.00	1	1	1	1	
struc.	JanusDDG	2.10	0.86	0.78	0.63	0.72	0.80
transfer	Stability Oracle	2.98 ± 0.48	0.77	0.58	0.46	0.61	0.71
ensemble	Ensemble 5 Feats	3.09 ± 0.42	0.75	0.59	0.48	0.6	0.68
ensemble	Ensemble 6 Feats	3.09 ± 0.43	0.75	0.59	0.48	0.6	0.68
ensemble	Ensemble 7 Feats	3.10 ± 0.42	0.75	0.59	0.48	0.6	0.69
ensemble	Ensemble 4 Feats	3.20 ± 0.43	0.75	0.57	0.48	0.61	0.68
ensemble	Ensemble 3 Feats	3.43 ± 0.45	0.72	0.5	0.4	0.59	0.66
potential	KORPM	3.54 ± 0.48	0.73	0.47	0.34	0.59	0.66
ensemble	Ensemble 2 Feats	3.68 ± 0.45	0.73	0.43	0.34	0.59	0.64
struc.	PSLM MutComputeX	4.04 ± 0.44	0.78	0.36	0.28	0.56	0.64
unknown	Gaussian Noise	4.96 ± 0.39	0.7	0	0	0.51	0.56
struc.	PSLM ESM-IF	5.04 ± 0.49	0.77	0.44	0.41	0.6	0.66
seq.	PSLM Tranception (reduced)	5.09 ± 0.50	0.77	0.25	0.22	0.53	0.58
seq.	PSLM Tranception	5.09 ± 0.50	0.78	0.26	0.24	0.54	0.58
struc.	PSLM ProteinMPNN 0.3	7.26 ± 0.62	0.78	0.48	0.41	0.61	0.69
struc.	PSLM ProteinMPNN 0.2	7.57 ± 0.60	0.78	0.49	0.41	0.6	0.68
struc.	PSLM ProteinMPNN 0.1	8.33 ± 0.64	0.78	0.48	0.4	0.6	0.68
seq.	PSLM ESM-2 150M	16.2 ± 2.14	0.72	0.22	0.24	0.57	0.62
struc.	PSLM MIF	23.8 ± 1.89	0.77	0.47	0.4	0.6	0.68
seq.	PSLM ESM-1V mean	24.2 ± 3.80	0.74	0.22	0.25	0.56	0.6
biophysical	Rosetta CartDDG	26.7 ± 1.98	0.78	0.56	0.43	0.61	0.69
seq.	PSLM ESM-2 650M	27.8 ± 2.64	0.76	0.29	0.3	0.57	0.63
struc.	PSLM MIF-ST	34.8 ± 2.44	0.77	0.4	0.32	0.59	0.63
seq.	PSLM Ankh	37.6 ± 2.74	0.77	0.31	0.28	0.55	0.62
seq.	PSLM MSA-T mean	37.7 ± 3.01	0.77	0.28	0.24	0.56	0.61
seq.	PSLM ESM-2 3B	42.8 ± 4.93	0.77	0.26	0.26	0.54	0.62
seq.	PSLM ESM-2 15B	52.5 ± 5.34	0.77	0.25	0.25	0.55	0.61
struc.	PSLM Rosetta/ProtMPNN	59.2 ± 3.80	0.8	0.62	0.49	0.62	0.71

Supplementary Table 10: Performance comparison of different models on PTmul-NR test set. **No sequence identity is shared between this test set and the JanusDDG training data.** The models’ performance data, excluding JanusDDG, were taken from Savojardo, C. et al.².

Model	Pearson	RMSE	MAE
JanusDDG	0.61	2.06	1.57
DDGemb	0.59	2.16	1.59
FoldX	0.36	5.51	3.66
Maestro	0.28	2.55	1.88
DDGun	0.23	2.55	2.10
DDGun3D	0.17	2.57	2.08

Supplementary Table 11: Performance comparison of different methods on Ptmul-D. Note that all methods, including JanusDDG, likely share some sequence similarity between the test and training sets. The performance of the ThermoMPNN-D model was taken from Dieckhaus, H. & Kuhlman, B.⁴.

Method	PCC	SCC	RMSE
JanusDDG	0.55	0.55	1.91
ThermoMPNN-D	0.57	0.59	1.95

Supplementary Note 3: Impact of Sequence Similarity Between Training and Test Sets

When training models for predicting the change in protein stability upon mutation ($\Delta\Delta G$), it is crucial to ensure that the training and test sets contain structurally and sequentially diverse proteins. If the training set includes proteins that are highly similar to those in the test set, the resulting performance may be overly optimistic and not representative of real-world generalization. This issue is illustrated in Supplementary Table 12, which compares the performance of the same model under two different data splitting strategies.

In the first case, we report the mean and standard deviation obtained from a 5-fold cross-validation where the folds were constructed such that no two folds share proteins with more than 25% sequence identity. This setup enforces a realistic generalization scenario. In the second case, we train the model on the original folds but remove 200 examples from each fold to form a validation set. While the folds were originally created to be non-redundant with respect to sequence similarity, this modified validation setup introduces overlap between training and validation sets, as training now includes proteins similar to those used for validation. As shown in Supplementary Table 12, the performance measured under this relaxed setting is significantly higher, highlighting the importance of carefully controlling for sequence similarity when evaluating model performance.

Supplementary Table 12: Comparison of model performance under different sequence similarity constraints between training and validation sets. Sequence identity coefficient represents the Top-3 Mean: the average sequence identity of the three most similar training set examples for each test protein, calculated using global alignment.

Validation Strategy	Pearson mean (standard deviation σ)
Strict split (sequence identity < 25%) (Sequence identity coefficient = 11%)	0.71 ($\sigma = 0.01$)
Validation with similar sequences (Sequence identity coefficient = 42%)	0.87 ($\sigma = 0.01$)

Supplementary Note 4: Theoretical Rationale for Using a Shared Zero-Embedding Reference in Transitive Training

Ideally, $\Delta\Delta G(A, B)$ is the difference of a state function,

$$\Delta\Delta G(A, B) = \Delta G(A) - \Delta G(B), \quad (1)$$

where $\Delta G(\cdot)$ is the folding free energy defined up to an additive constant (gauge). In practice, a neural predictor trained on finite data can exhibit non-conservative (path-dependent) residuals, so transitivity may not hold exactly. A direct way to reduce such residuals is to enforce multi-step consistency via an auxiliary *transitivity loss* by sampling chains of intermediates $A \rightarrow x_1 \rightarrow \dots \rightarrow x_n \rightarrow B$, i.e.

$$\widehat{\Delta\Delta G}(A, B) \approx \sum_{k=1}^n \widehat{\Delta\Delta G}(x_{k-1}, x_k), \quad (x_0 = A, x_n = B). \quad (2)$$

Alternatively, a loss can regularize by averaging over a large set of candidate intermediates $a \in \mathcal{A}$:

$$\widehat{\Delta\Delta G}(A, B) \approx \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left(\widehat{\Delta\Delta G}(A, a) + \widehat{\Delta\Delta G}(a, B) \right). \quad (3)$$

(where $\mathcal{A}$ could correspond to exhaustive enumeration over the 20 amino acids at the mutated position(s), or a sampled subset). However, both approaches are computationally expensive and, when approximated by limited sampling, introduce Monte-Carlo noise because each pair is constrained through different stochastic intermediates.

Instead, we introduce a single shared reference state U defined by the zero vector ($\vec{0}$) in the ESM-2 latent space. We do not interpret U as a realizable protein sequence; rather, it serves as a canonical thermodynamic gauge origin (reference state). Compared to using a biological intermediate, $\vec{0}$ provides a particularly neutral and stable gauge choice for encouraging the model to learn a path-independent, potential-like structure. This choice is motivated by three complementary aspects of a good gauge origin: it should be (i) representation-neutral, (ii) architecturally uninformative (so it does not inject sequence-specific structure), and (iii) shared across all pairs so that the predictor is encouraged to behave like a difference of a potential (as true state functions do). In more detail:

- **Representation Neutrality (Algebraic):** Research on Protein Language Models⁵ indicates that fundamental physicochemical properties are often linearly decodable from the embedding space, acting as direction vectors $\vec{v}_{feat}$. Any biological reference sequence would possess a non-zero magnitude, inevitably projecting onto these axes ($\vec{x}_{ref} \cdot \vec{v}_{feat} \neq 0$) and introducing inductive bias. Using the zero vector minimizes reference-dependent projections along linear feature directions in the embedding space, reducing the inductive bias that a biologically specific reference sequence would introduce.
- **Architectural Uninformativeness (Signal Isolation):** In our bidirectional cross-attention architecture, the transition $A \rightarrow \vec{0}$ implies a difference embedding $\Delta = A - \vec{0} = A$. Consequently, the forward cross-attention becomes dominated by information coming from A (and, in the limit of an uninformative reference branch, approaches a self-referential operation), encouraging the model to extract features of A in isolation. Conversely, in the inverse path where keys/values are derived from $\vec{0}$, the reference branch carries minimal content (up to learned bias terms), limiting the ability of attention to encode intermediate-specific structure. Overall, this encourages measuring the stability potential of A relative to a structurally uninformative background.
- **Shared reference state (consistent zero of energy):** While deep networks can exhibit non-zero activations for null inputs due to internal bias terms, this baseline response C_{sys} is deterministic and constant. By enforcing transitivity through $U = \vec{0}$, the model learns to decompose any mutation effect through the same reference state:

$$\widehat{\Delta\Delta G}(A, B) \approx \widehat{\Delta\Delta G}(A, U) + \widehat{\Delta\Delta G}(U, B). \quad (4)$$

If the predictor is close to a state-function form, i.e., $\widehat{\Delta\Delta G}(X, Y) \approx \widehat{\Delta G}(X) - \widehat{\Delta G}(Y)$, then this becomes

$$\widehat{\Delta\Delta G}(A, B) \approx (\widehat{\Delta G}(A) - \widehat{\Delta G}(U)) - (\widehat{\Delta G}(B) - \widehat{\Delta G}(U)), \quad (5)$$

i.e., the difference between two free-energy-like scores *expressed relative to the same reference state*. Thus, we obtain

$$\widehat{\Delta\Delta G}(A, B) \approx \widehat{\Delta G}(A) - \widehat{\Delta G}(B), \quad (6)$$

where $\widehat{\Delta G}$ represents the learned absolute stability potential. The systematic network bias $\widehat{\Delta G}(U = \vec{0}) = C_{sys}$ effectively acts as the thermodynamic gauge constant, which cancels out in the difference.

Using a fixed zero reference is thus computationally efficient, avoids stochastic sampling noise, and provides a residue-agnostic anchor that encourages the model to learn a consistent, path-independent energy landscape. By distilling a more state-function-like predictor (i.e., one that is closer to a difference of potentials), the resulting model is expected to satisfy the transitive property at lower computational cost.

Supplementary Note 5: Hyperparameter Selection and Ablation Study on Pooling Mechanisms

Due to the long training times, hyperparameter optimization focused on the transformer component. We tested three configurations (see Supplementary Table 13) and selected the most parameter-efficient model that maintained comparable accuracy: 8 attention heads, a feed-forward network size of 512, and an input dimension of 128. Training utilized the Mean Squared Error (MSE) loss and the Adam optimizer. We employed a 5-fold cross-validation strategy on the training set², using MMSeqs2⁶ to cluster proteins with $> 25\%$ sequence identity into distinct folds. Consequently, the final model was trained for 300 epochs, corresponding to the average convergence point observed across the folds.

Supplementary Table 13: Comparison of transformer configurations. The performance is reported as mean $\pm$ standard deviation. The hyperparameters selected for the final model are highlighted in bold.

Heads	FFN Size	Input Dim	Performance (mean $\pm$ std)
4	512	128	0.70 $\pm$ 0.02
8	512	128	0.71 $\pm$ 0.01
8	1024	128	0.71 $\pm$ 0.01

The JanusDDG architecture employs a dual pooling strategy that concatenates the outputs of Global Average Pooling (GAP) and Global Max Pooling (GMP) before the final classification/regression layer. To quantify the specific contribution of each mechanism and justify their combined usage, we performed an ablation study by training three variants of the model:

- GAP-only: Using only Global Average Pooling.

- GMP-only: Using only Global Max Pooling.
- JanusDDG (Concatenated): Using the concatenation of both (default architecture).

From a modeling perspective, global average pooling (GAP) and global max pooling (GMP) summarize the sequence embeddings in complementary ways: GAP aggregates overall feature trends, while GMP emphasizes the most strongly activated features. We evaluated variants of JanusDDG using only GAP, only GMP, or their concatenation on the validation set. As reported in Supplementary Table 14, models relying solely on GAP or GMP perform slightly worse than when the two are used together. This suggests that combining GAP and GMP allows the network to exploit complementary information, leading to improved predictive performance.

Supplementary Table 14: Performance comparison of pooling strategies on the validation dataset.

Pooling Strategy	Pearson mean (standard deviation σ)
GAP only	0.69 ($\sigma = 0.01$)
GMP only	0.70 ($\sigma = 0.02$)
GMP + GAP (JanusDDG)	0.71 ($\sigma = 0.01$)

Supplementary Note 6: Estimation of Prediction Error Bars in JanusDDG

There are two complementary approaches can be used to estimate such error bars: an empirical method and a statistical one.

(i) *Empirical approach.* This method consists of bootstrapping the model predictions and computing dispersion measures, such as the standard deviation across the resampled outcomes.

(ii) *Fisher’s z -transformation.* To estimate the confidence interval of the Pearson correlation coefficient r , Fisher’s z -transformation can be applied. This transformation approximates the distribution of r as normal for sufficiently large sample sizes and is defined as

$$z = \frac{1}{2} \ln \frac{1+r}{1-r}.$$

The standard error of z is given by

$$SE_z = \frac{1}{\sqrt{n-3}},$$

where n denotes the number of observations. The confidence interval in the z -domain is then calculated as

$$z \pm z_{\alpha/2} \cdot SE_z,$$

where $z_{\alpha/2}$ represents the $(1 - \alpha/2)$ quantile of the standard normal distribution. Using $z_{\alpha/2} = 1$ corresponds to one standard error. The confidence interval expressed in terms of r is subsequently obtained via the inverse transformation

$$r = \frac{e^{2z} - 1}{e^{2z} + 1}.$$

For the S669 dataset, we obtain $r = 0.55$ with $n = 669$ samples. Employing one standard error results in an interval of $r \in [0.52, 0.58]$. The error bars are substantially wider for the S461 and PTmul-NR datasets, which contain fewer samples. It is also important to note that predictions from S669 cannot be combined with those from S461, as the latter constitutes a subsample of the former. Consistent estimates are obtained using the empirical method with 1000 bootstrap replicates, yielding a standard error interval of $r \in [0.52, 0.58]$. In the bar plots shown in Figures 3 and 4, Fisher’s z -transformation was applied to compute the error bars.

In order to estimate the uncertainty associated with the classification metrics (Figure 6), we adopted two different strategies. For precision and recall of the JanusDDG model, the error was derived using the analytical standard error of a binomial proportion:

$$SE(p) = \sqrt{\frac{p(1-p)}{n}},$$

where p denotes the observed metric (precision or recall) and n corresponds to the number of relevant instances ($n = TP + FP$ for precision and $n = TP + FN$ for recall). This formula relies on the binomial approximation of the sampling distribution.

For the balanced accuracy, no closed-form analytical standard error exists, as it is an average of sensitivity and specificity. Therefore, we estimated its uncertainty using a non-parametric bootstrap: the test set was resampled with replacement $B = 1000$ times, and the metric was recomputed for each resample. The empirical standard deviation of the bootstrap distribution was then used as an estimate of the error:

$$\widehat{SE}_{BA} = \sqrt{\frac{1}{B-1} \sum_{b=1}^B \left(BA^{(b)} - \overline{BA} \right)^2}.$$

Supplementary Note 7: Dataset

In this section we provide a summary of the datasets used for training, validation and test.

Supplementary Table 15: Summary of datasets used, including the number of mutations, proteins, and $\Delta\Delta G$ statistics.

Dataset	Number of Mutations	Number of Proteins	$\Delta\Delta G$ Mean	$\Delta\Delta G$ Std	$\Delta\Delta G$ Min	$\Delta\Delta G$ Max
S2450	2450	115	-1.01	1.48	-5.0	6.80
M28	28	12	-1.76	2.45	-8.03	3.42
S669	669	96	-0.98	1.62	-8.93	5.73
S461	461	48	-1.15	1.25	-5.00	4.30
S96	96	14	-0.89	2.45	-6.38	9.73
PTmul-NR	82	13	-0.15	2.61	-7.53	6.20
Q3421	3421	148	-1.13	1.97	-17.40	13.70
K2369	2369	181	-0.69	1.60	-6.90	4.40
Ssym	684	357	0.00	1.83	-7.5	7.5
PTmul-D	583	83	-1.26	2.24	-10.80	6.20

Supplementary References

- [1] Reeves, S. & Kalyaanamoorthy, S. Zero-shot transfer of protein sequence likelihood models to thermostability prediction. *Nature Machine Intelligence* **6**, 1063–1076 (2024).
- [2] 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* btaf019 (2025).
- [3] Montanucci, L. *et al.* Ddgun: an untrained predictor of protein stability changes upon amino acid variants. *Nucleic Acids Research* **50**, W222–W227 (2022).
- [4] Dieckhaus, H. & Kuhlman, B. Protein stability models fail to capture epistatic interactions of double point mutations. *Protein Science* **34**, e70003 (2025). URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.70003>.
- [5] Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **118**, e2016239118 (2021).
- [6] Steinegger, M. & Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology* **35**, 1026–1028 (2017).