Accurate Prediction of Protein Structural Flexibility with Deep Learning

THE UNIVERSITY *of* EDINBURGH

Accurate Prediction of Protein Structural Flexibility with Deep Learning

Name: Felix Burton

Abstract:

Protein dynamics are essential for biological function yet remain challenging to measure. Here we present DeepFlex, a deep learning model trained on a large Molecular Dynamics Dataset that accurately predicts residue-level protein flexibility across varying temperatures. By integrating evolutionary information, structural descriptors, and 3D atomic context through temperature-conditioned attention, DeepFlex achieves high correlation (PCC = 0.81) with molecular dynamics simulations while being several orders of magnitude faster. DeepFlex enables rapid, temperature-aware flexibility prediction for applications ranging from thermal adaptation studies to protein engineering, providing a powerful framework for incorporating essential dynamic dimensions into computational biology workflows.

Abstract word count: 96

Report word count: 4,995

Abbreviations:

PDB = Protein Data Bank, **RMSF** = Root Mean Square Fluctuation, **MD** = Molecular Dynamics (Simulations), **DSSP** = Define Secondary Structure of Proteins, **CATH** = Class, Architecture, Topology, Homologous superfamily, **ESM-C** = Evolutionary Scale Model Cambrian, **CNN** = Convolutional Neural Network, **PLM** = Protein Language Model, **MLP** = Multilayer perceptron**, PCC** = Pearson Correlation Coefficient, **MAE** = Mean Absolute Error, **MSE** (Mean Squared Error), **R^2** = Coefficient of Determination, **SEM** (Standard Error of the Mean).

Table of Contents

1. Introduction:

Protein dynamics govern biological function[1, 2, 3, 4]. In their native environments, proteins exist not as static structures but as dynamic ensembles, continuously sampling conformational landscapes across multiple spatial and temporal scales[1, 2]. This intrinsic flexibility—ranging from picosecond side-chain rotations to millisecond domain rearrangements—orchestrates the molecular choreography underlying enzyme catalysis[3], allosteric regulation[4], and macromolecular recognition[2]. Evolution has fine-tuned this flexibility with the same precision as stability, with each protein having a distinct dynamic profile for its specific environment[1]. For example, psychrophilic enzymes loosen catalytic loops to remain active at low temperatures, whereas thermophilic counterparts reinforce hydrophobic cores to withstand extreme heat[5,6]. Increasingly, we recognize that pathogenic variants often disrupt normal cellular processes not by dramatically altering protein fold, but through subtle perturbations to this finely tuned dynamic equilibrium[7].

Despite their central importance, protein dynamics remain challenging to characterize at scale. Experimental techniques like X-ray crystallography[8], solution NMR[9,10], and cryo-EM[11] provide invaluable atomic-resolution insights, but share practical limitations for comprehensive flexibility analysis. They are inherently low-throughput, often measure dynamics under non-physiological conditions, and face technique-specific constraints—crystal contacts confounding B-factors[8], molecular size limitations for NMR[9,10], or compromised resolution in highly dynamic regions for cryo-EM[11]. These limitations restrict their application in protein design workflows or large-scale comparative studies.

Computationally, all-atom molecular dynamics (MD) simulations can reproduce experimental fluctuations with explicit environmental control by simulating the physical motions of atoms

over time[12]. MD directly yields metrics including residue-level Root Mean Square Fluctuation (RMSF), a quantitative measure of flexibility. Unfortunately, the rugged energy landscapes of proteins necessitate simulations spanning microseconds or longer to adequately sample conformational space, demanding enormous computational cost or specialized hardware[12]. Faster approximations, like Elastic Network Models (ENMs), sacrifice physical realism and sequence-specific chemical details[8,13-15]. Consequently, no single existing approach readily provides residue-level accuracy, environmental awareness, and rapid throughput for flexibility assessment.

Given the limitations of traditional methods and inspired by deep learning successes in enabling highly accurate predictions of three-dimensional protein folds[16], research leveraging Machine Learning (ML) to predict protein flexibility directly from sequence or structure is rapidly accelerating. This progress is fuelled by: (i) neural network architectures effective for static structure prediction offering starting points for dynamics modelling; (ii) the increasing availability of large-scale MD simulation datasets providing ground-truth flexibility data[17,18]; and (iii) the urgent need for computationally efficient alternatives to MD for high-throughput analysis and rational protein design[19,20]. Early computational efforts focused on shallow networks or statistical regressors to reproduce crystallographic B factors, but these values conflate intrinsic motion with crystal packing artefacts and have displayed limited predictive accuracy[8]. More recent frameworks employ sophisticated deep learning models to directly predict MD-derived values, aiming for a measure closer to solution dynamics. For example, Flexpert uses neural networks with sequence embeddings and structural features, achieving high accuracy against simulation[19]. Similarly, RMSF-net utilizes 3D convolutional networks with experimental and structural inputs[21].

While recent ML predictors show promise, accurately modelling residue-level flexibility presents distinct challenges. Current models trained at fixed temperatures (~300 K) perform

poorly outside these conditions[19]. Temperature is a fundamental thermodynamic parameter that influences conformational ensembles according to the Boltzmann distribution, primarily by shifting the delicate balance between enthalpy and entropy. This effect is critical: temperature determines protein stability and folding landscapes, directly impacts biological function and reaction rates, and drives evolutionary adaptation strategies[5, 6]. Predicting flexibility explicitly as a function of temperature is therefore essential. We hypothesized that a deep learning architecture incorporating explicit temperature conditioning through a learnable embedding space could capture the complex, nonlinear relationships between thermal energy and local conformational freedom across diverse protein folds.

Here, we introduce DeepFlex, a deep learning model designed to predict residue-level protein flexibility (RMSF) across varying temperatures. DeepFlex integrates evolutionary information distilled from a large protein language model (ESM-C)[22], standard structural descriptors, and 3D atomic context captured via a custom module operating on voxelized atomic neighbourhoods[23,24]. A key innovation is its temperature-responsive self-attention mechanism[25] within the core architecture, which learns how different protein regions uniquely respond to thermal changes.

The model was trained on the mdCATH dataset[17], containing extensive MD simulations (62 ms total) for 5,398 structurally diverse CATH domains [26] across five temperatures (320-450 K). Evaluated on a topology-stratified holdout set, DeepFlex demonstrates high predictive accuracy and robust generalization across the full temperature range.

By enabling rapid, temperature-aware flexibility predictions, DeepFlex bridges static structural predictions and environmentally modulated protein dynamics. For evolutionary biophysics, it facilitates analyses of thermal adaptation mechanisms across protein families. In protein engineering, where the objective is shifting from designing stable scaffolds towards controlling

functional motions[20], DeepFlex introduces a differentiable flexibility objective[19] that can be jointly optimized alongside stability, supporting the rational design of enzymes, biosensors, and therapeutic proteins whose activities rely on carefully tuned conformational dynamics. DeepFlex thus represents a significant contribution towards integrating essential physical context into protein dynamic behavior prediction.

---

2. Materials and Methods:

2.1 Dataset Origin and Processing

Source Dataset: The primary training and evaluation dataset was mdCATH[17], comprising multi-replica (5 independent runs per protein per temperature) and multi-temperature (320, 348, 379, 413, 450 K) all-atom MD simulations with a total of over 62 ms of trajectory data across 5,398 diverse protein domains. The mean per-residue Cα RMSF across the five replicates for each domain–temperature served as the ground truth flexibility measure and prediction target.

RMSF Definition: RMSF quantifies residue flexibility, calculated as the root-mean-square deviation of a Cα atom's position from its mean position over the simulation ensemble after trajectory alignment.

$$\mathrm{RMSF}_i = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{r}_i(t) - \langle \mathbf{r}_i \rangle\right)^2}$$

where RMSF($_i$) is the RMSF of residue *i, T* is the number of frames, r_i(t) is the position of the Cα of atom of residue *i* at time *t*, and r_i is the mean position of that Cα atom over the trajectory.

Sequestered Holdout Set: A holdout set (401 domains; ~7.5%) was sequestered prior to development using topology-aware stratified sampling[28]. The protocol stratified domains by CATH [26] hierarchy, ensured representative sampling, controlled homology and validated partition quality via a Representation Index > 0.9 [Appendix S1]. The holdout set remained completely excluded from all stages of model development and training.

Development Dataset Preprocessing: After excluding 514 domains due to technical issues [Appendix S2], data for 4,483 domains (3,064,132 instances) were cleaned and processed [29] [Appendix S2].

Development Set Splitting: The development set was partitioned (80% train, 10% val, 10% test) via topology-aware stratified sampling (CATH ID based)[28], preventing homology contamination.

2.2 Feature Engineering

DeepFlex integrates multi-modal features to predict protein flexibility:

- **Per-residue descriptors:** Calculated from cleaned PDB structures[27,29], including normalized residue position, protein length, secondary structure (DSSP)[29], relative solvent accessibility (RSA) [29, 31], core/surface classification, normalized backbone dihedral angles (φ/ψ), and optional Z-scored B-factors [Appendix S3].
- **3D structural context:** Provided by VoxelFlex[24], an independently trained 3D CNN (Multipath ResNet; Fig. 1) that predicts RMSF values based solely on local atomic environments. For each residue, a 3D grid (21^3 voxels, 12Å edge length) centered on its Cα atom captures surrounding atomic arrangements, generated using Aposteriori[32] [Appendix S4].
- **Environmental condition:** Raw simulation temperature (Kelvin) as direct input.

All numerical features were Min–Max scaled to [0,1] range using training set parameters applied consistently across all data splits [Appendix S3].

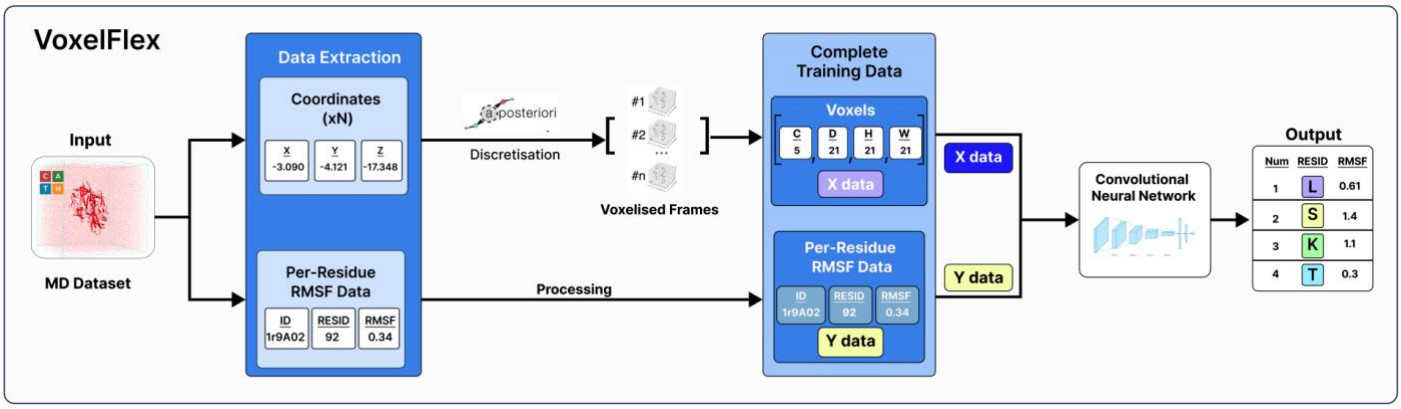

Figure 1 | VoxelFlex 3D Feature Generation Pipeline. Input structures from the mdCATH dataset are processed. Coordinates are discretized into 3D voxel grids (21×21×21, 5 channels: CNOCBCA) centered on each Cα atom using the Aposteriori[32] tool. Per residue replica averaged RMSF data from the training split serves as the ground truth target (Y data). The voxel grids (X data) and corresponding RMSF targets are used to train an independent 3D Convolutional Neural Network (VoxelFlex). The trained VoxelFlex model predicts per residue RMSF values from the voxel grids alone, which serve as a 3D structural context feature input to the main DeepFlex model.

2.3 DeepFlex Model Architecture

The DeepFlex model (Fig. 2) employs temperature-conditioned self-attention to integrate multi-modal inputs. Sequence information was encoded using frozen ESM-C embeddings (600M parameters, 1152D)[22] augmented with sinusoidal positional encodings[25]. Per-residue descriptors [Appendix S3] and the VoxelFlex-derived RMSF feature [Appendix S4] were concatenated and projected via a Fusion MLP to match the 1152D embedding dimension. Scaled input temperature was embedded into 16 dimensions using a separate MLP (Temp MLP). The core transformer block utilizes pre-Layer Normalization, residual connections, and multi-head self-attention (MHA; 8 heads, 0.1 dropout)[25]. Temperature conditioning was implemented by linearly biasing the MHA Query, Key, and Value projections using the 16D temperature embedding. A final MLP regression head predicted scalar RMSF from the processed 1152D residue representations [Appendix S5].

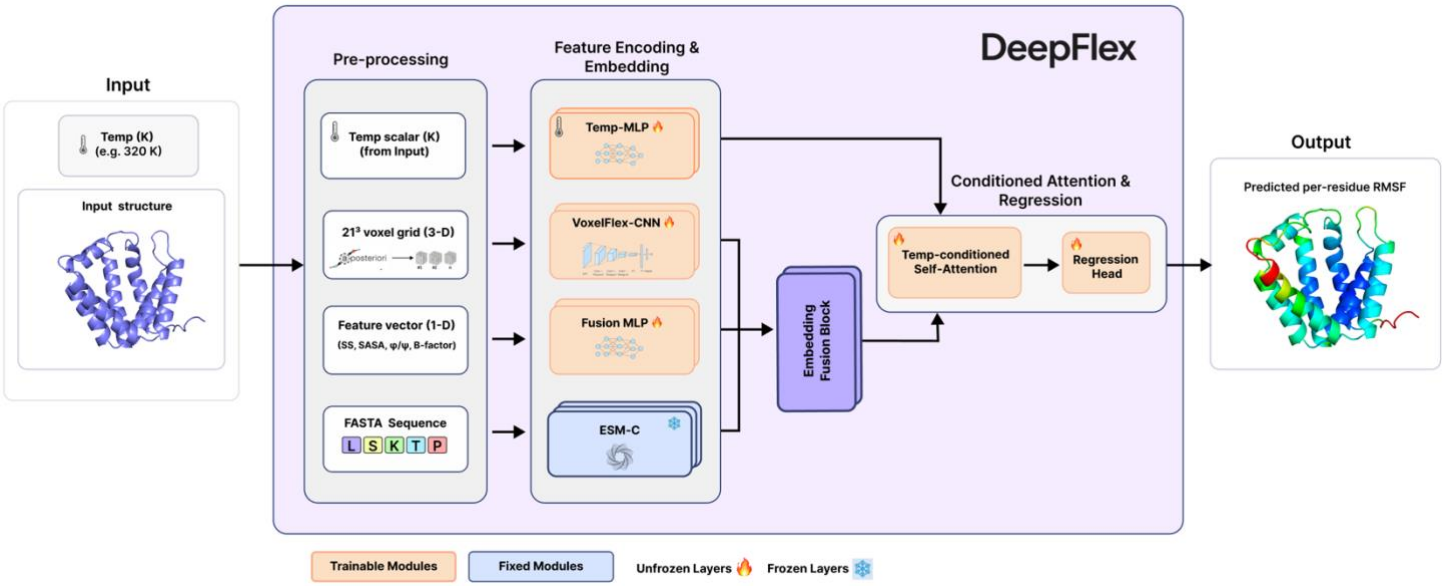

Figure 2 | DeepFlex Model Architecture. Overview of temperature-aware RMSF prediction. **Inputs:** Protein .PDB file and temperature (K**). Pre-processing:** Generates four feature streams: sequence, per-residue descriptors, 3D voxel grids, and scaled temperature. **Encoding & Embedding:** Parallel modules process these streams: frozen ESM-C (snowflake icon) for sequence, trainable VoxelFlex-CNN (fire icon) for 3D context, trainable Fusion MLP (fire icon) for descriptors, and trainable Temp MLP (fire icon) for temperature embedding. **Fusion & Attention**: Embeddings are fused before entering the Temperature-Conditioned Self-Attention block (fire icon), where the temperature embedding modulates residue interactions. **Output:** Predicted per residue RMSF, visualized on the structure (blue: low RMSF, red: high RMSF). **Legend Key:** Fire icons indicate trainable components; snowflake icons indicate frozen components.

2.4 DeepFlex Model Training

DeepFlex was trained using the AdamW optimizer[33], minimizing MSE loss[34]. Training used an effective batch size of 24 (8 /GPU, 4 gradient accumulation steps), gradient clipping, a ReduceLROnPlateau scheduler and early stopping. We applied a 0.1 dropout rate in the regression head. The best model (epoch 33) was selected based on validation correlation. Training took ~12 h on NVIDIA Quadro RTX 8000 GPUs [Appendix S5].

2.5 Baseline and Comparative Models

To benchmark DeepFlex (Fig 2), alternative models were trained and evaluated on the identical data splits [Appendix S6]. These comparative models included:

- **Sequence-only (ESM-Only):** Used frozen ESM-C 600M parameter embeddings[22] combined with temperature input, fed into an MLP. Designed to isolate the contribution of the ESM-C model alone using only sequence and temperature information as input.
- **Structure-only (VoxelFlex-3D):** Used the VoxelFlex 3D CNN model [Appendix S4] to predict RMSF directly from the voxelized atomic neighbourhood. Designed to isolate the contribution of local 3D structural context alone.
- **Feature-Matched Models:** Included LightGBM[35], Random Forest (RF)[36], and a Neural Network (NN)[34]. These were trained on the exact same complete multi-modal feature set used by DeepFlex, allowing for direct comparison of the DeepFlex architecture against standard methods using identical information.
- **Ablation Model:** A Random Forest model trained without the ESM embeddings (RF (No ESM Feats)). Designed to assess the impact of removing evolutionary context from the feature-matched ensemble baseline.
- **Single-Temperature Models (DeepFlex (Single Temp)):** DeepFlex, was trained separately on data from individual temperatures during initial development. For instance, the DeepFlex 320K model was trained solely on RMSF values from 320K simulations, with separate models created for each temperature point.

2.6 Performance Evaluation Metrics

Model performance was evaluated on the holdout set using Pearson Correlation Coefficient (PCC), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and bias. Residue-level coefficient of determination (R^2) assessed inter-model similarities. Performance was also stratified by secondary structure, solvent accessibility, and sequence position to evaluate model behavior across diverse structural contexts.

2.7 Data and Code Availability

mdCATH dataset[17] is public. Code (DeepFlex, VoxelFlex, processing, sampling, analysis), weights, and data splits available at [https://github.com/wells-wood-research], ESM-C[22] source code available at [https://github.com/evolutionaryscale/esm/].

---

3. Results:

3.1 DeepFlex achieves high-accuracy prediction of protein structural flexibility

We evaluated DeepFlex's ability to predict protein dynamics using a sequestered set of 401 protein domains not used during model development. Performance was measured against the "ground truth" flexibility values (RMSF) obtained from molecular dynamics simulations using two primary metrics: Pearson Correlation Coefficient (PCC) - Measures how well our predictions match the actual flexibility patterns. Mean Absolute Error (MAE) - Quantifies the average prediction error in Angstroms.

DeepFlex achieved excellent performance with a mean PCC of 0.807 (±0.020) and MAE of only 0.139 (±0.055) Å across all temperatures, with negligible systematic bias (-0.017 ±0.024 Å; Table 1).

To understand what drives this performance, we compared DeepFlex against simpler models. Models using solely pre-trained ESM-C[22] with protein sequence information as input (ESM-Only) or only structural data (VoxelFlex-3D)[24] performed substantially worse, with correlation decreases of 0.13-0.20 and error increases of approximately 29-47%. This confirms that both sequence evolutionary information and three-dimensional structural context are necessary for accurate flexibility prediction.

Even when provided with the same input features, traditional machine learning methods (Random Forest: RF) still underperformed compared to DeepFlex (PCC lower by ~0.06), highlighting the value of our temperature-aware attention mechanism. Additionally, versions of DeepFlex trained at only a single temperature (DeepFlex (Single Temp)) performed worse than our unified multi-temperature model, demonstrating the benefits of learning flexibility patterns across diverse thermal conditions. All performance differences between DeepFlex and comparison models were statistically significant.

Table 1: Comparison of protein flexibility prediction performance averaged across temperatures. Values shown are means ± standard errors across 401 hold-out protein domains. For each domain, we first calculated performance metrics (PCC, MAE, RMSE, Bias) at each of the five temperatures (320K-450K), then averaged these five values to obtain domain-level metrics, which were then averaged across all domains to produce the values shown. Bias represents the mean signed error (positive values indicate overestimation, negative values indicate underestimation). **Bold** values indicate best performance for each metric. Statistical significance vs. DeepFlex by paired Wilcoxon test: * $p < 0.05$ (significant difference), † $p \geq 0.05$ (not significant).

Model	PCC (↑)	MAE (Å) (↓)	RMSE (Å) (↓)	Bias (Å)
DeepFlex	**0.807 ± 0.020**	**0.139 ± 0.055**	**0.179 ± 0.061**	-0.017 ± 0.024
DeepFlex (Single Temp)	0.765 ± 0.037*	0.169 ± 0.080*	0.212 ± 0.085*	0.042 ± 0.090†
Random Forest	0.747 ± 0.017*	0.158 ± 0.051*	0.200 ± 0.056*	-0.011 ± 0.005†
ESM-Only (Seq+Temp)	0.679 ± 0.028*	0.179 ± 0.058*	0.221 ± 0.062*	**0.002 ± 0.028**†
Random Forest (No ESM)	0.661 ± 0.013*	0.202 ± 0.061*	0.247 ± 0.064*	-0.035 ± 0.035†
VoxelFlex-3D	0.606 ± 0.083*	0.204 ± 0.077*	0.254 ± 0.083*	-0.079 ± 0.043*

3.2 A Single DeepFlex Model Generalizes Across Temperatures

A key advantage of DeepFlex is its ability to accurately predict protein dynamics across different temperatures with a single model. Our unified model maintained high accuracy throughout the entire 320K-450K temperature range (Table 2, Fig. 3), with correlation values remaining consistently strong (PCC never below 0.78) despite the increasing prediction challenge at higher temperatures.

Prediction error (MAE) increased across the temperature range, from 0.080 Å at 320K to 0.213 Å at 450K. In contrast, the sequence-only models (ESM-Only) performed consistently worse across all temperature points, while both structure-only models (VoxelFlex-3D) and single-temperature trained models exhibited substantially more severe performance degradation as temperatures increased.

DeepFlex's advantage over feature-matched baselines was most pronounced at physiological temperatures (18% lower error at 320K), gradually diminishing at extreme temperatures (3% at 450K) where thermal motion introduces more inherently unpredictable behaviour. Importantly, DeepFlex's predictions closely track the natural broadening of flexibility distributions seen in the simulation data (Fig. 3), demonstrating that our model captures the fundamental physical principles of how thermal energy affects protein dynamics.

Table 2: Temperature-specific performance of DeepFlex and comparison models. Values shown are means ± standard errors across 401 hold-out protein domains at each temperature. For each domain at each temperature, we calculated performance metrics, then averaged these values across all domains to produce the results shown. **Bold** values indicate best performance at each temperature. Statistical significance vs. DeepFlex by paired Wilcoxon test: * $p < 0.05$ (significant difference), † $p \geq 0.05$ (not significant).

Metric	Temp (K)	DeepFlex	DeepFlex (Single Temp)	Random Forest	ESM Only	VoxelFlex 3D
PCC (↑)	320.0	**0.823 ± 0.006**	0.798 ± 0.007*	0.761 ± 0.007*	0.670 ± 0.009*	0.677 ± 0.007*
	348.0	**0.825 ± 0.006**	0.790 ± 0.008*	0.764 ± 0.006*	0.698 ± 0.008*	0.663 ± 0.007*
	379.0	**0.812 ± 0.007**	0.784 ± 0.008*	0.751 ± 0.006*	0.704 ± 0.008*	0.637 ± 0.008*
	413.0	**0.791 ± 0.007**	0.741 ± 0.009*	0.733 ± 0.007*	0.689 ± 0.008*	0.580 ± 0.007*
	450.0	**0.781 ± 0.008**	0.711 ± 0.010*	0.726 ± 0.008*	0.635 ± 0.009*	0.474 ± 0.009*
MAE (Å) (↓)	320.0	**0.080 ± 0.003**	0.088 ± 0.004*	0.098 ± 0.003*	0.117 ± 0.003*	0.114 ± 0.005*
	348.0	**0.098 ± 0.004**	0.105 ± 0.004*	0.120 ± 0.004*	0.134 ± 0.004*	0.144 ± 0.005*
	379.0	**0.129 ± 0.005**	0.145 ± 0.005*	0.153 ± 0.005*	0.168 ± 0.004*	0.202 ± 0.008*
	413.0	**0.177 ± 0.005**	0.261 ± 0.007*	0.198 ± 0.005*	0.221 ± 0.005*	0.260 ± 0.006*
	450.0	**0.213 ± 0.006**	0.246 ± 0.007*	0.220 ± 0.005*	0.253 ± 0.005*	0.298 ± 0.006*

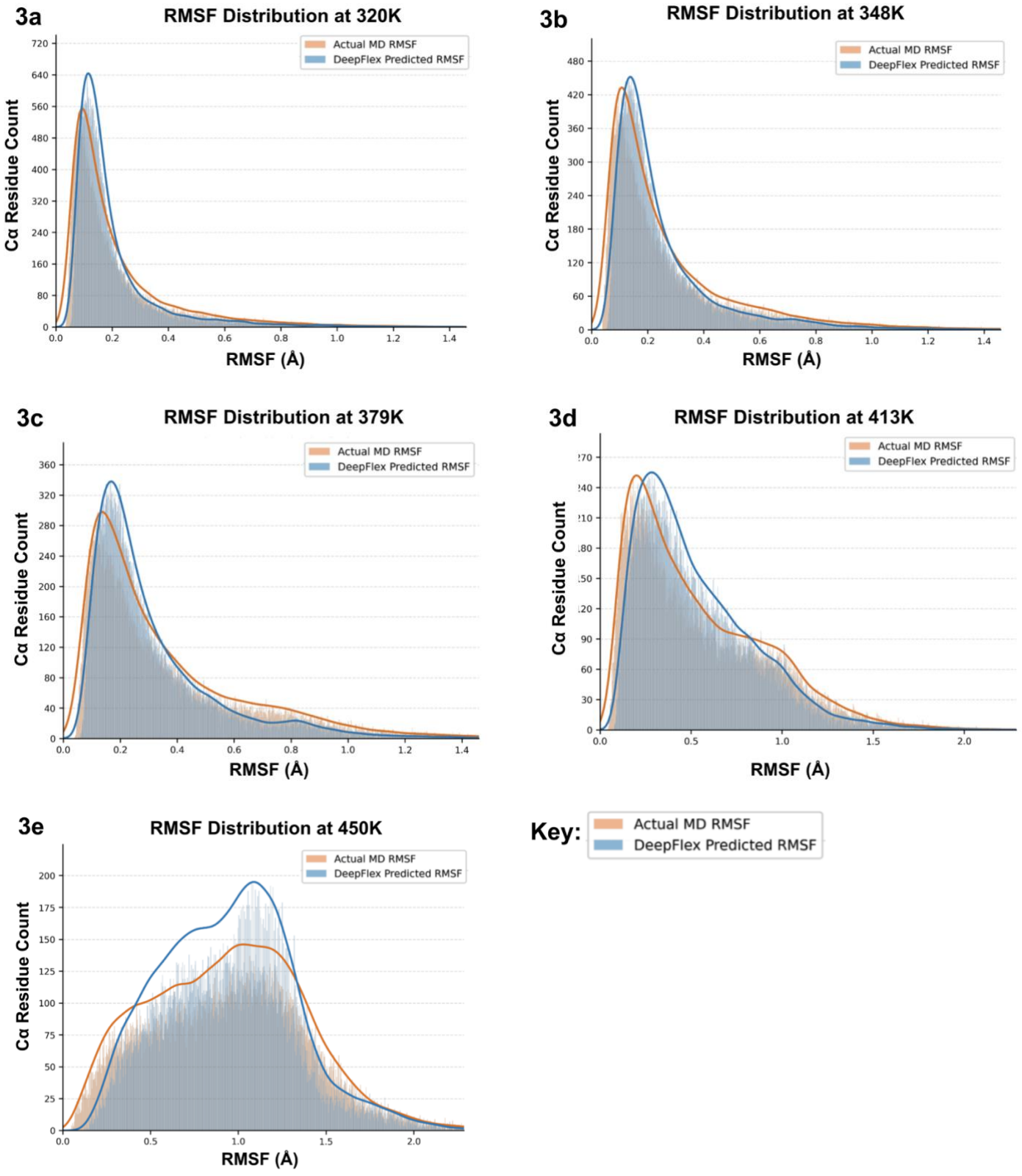

Figure 3 | DeepFlex accurately captures temperature-dependent RMSF distributions. DeepFlex accurately captures temperature-dependent RMSF distributions and domain-level trends, reproducing the expected increase in flexibility with temperature. **a – e**, predicted distributions (blue histograms) closely follow the MD ground-truth KDEs (orange) and broaden from 320 K to 450 K (≈ 57 k residues per panel). A single, fixed bin width is applied to every histogram; as the distributions widen with temperature, residue counts are spread over more bins, so peak heights decrease accordingly.

3.3 Architecture and feature integration underlie performance

To isolate the architectural benefit of DeepFlex beyond its input features, we trained three feature-matched baselines—Random Forest (RF), LightGBM (LGBM) and a standard Neural Network (NN)—on exactly the same multi-modal inputs. At the residue level, DeepFlex showed higher pooled R^2 with MD (0.761) than any baseline (0.718 – 0.724; Fig 4a). The high inter-correlation among baseline models ($R^2 \geq 0.96$) indicates they extract similar patterns from the shared features, albeit less effectively.

Error analysis revealed further architectural distinctions. While baseline models showed highly correlated error patterns ($R^2 \geq 0.83$) suggesting shared systematic limitations, DeepFlex's errors correlated weakly with these baselines ($R^2 \approx 0.43–0.46$) (Fig. 4b), indicating our temperature-conditioned attention mechanism addresses different aspects of the prediction problem.

This architectural advantage was consistent over all temperatures (Figs. 4e) and generalized broadly, with DeepFlex achieving higher median correlation (PCC = 0.830) than the best baseline (RF, PCC = 0.768) across most domains (Figs. 4c, 4d). Crucially, while models trained at single temperatures degraded progressively when predicting beyond their training conditions, our multi-temperature trained DeepFlex maintained consistent performance across the entire 320K-450K range (Fig. 4f)—confirming that temperature-conditioning enables truly generalizable flexibility prediction.

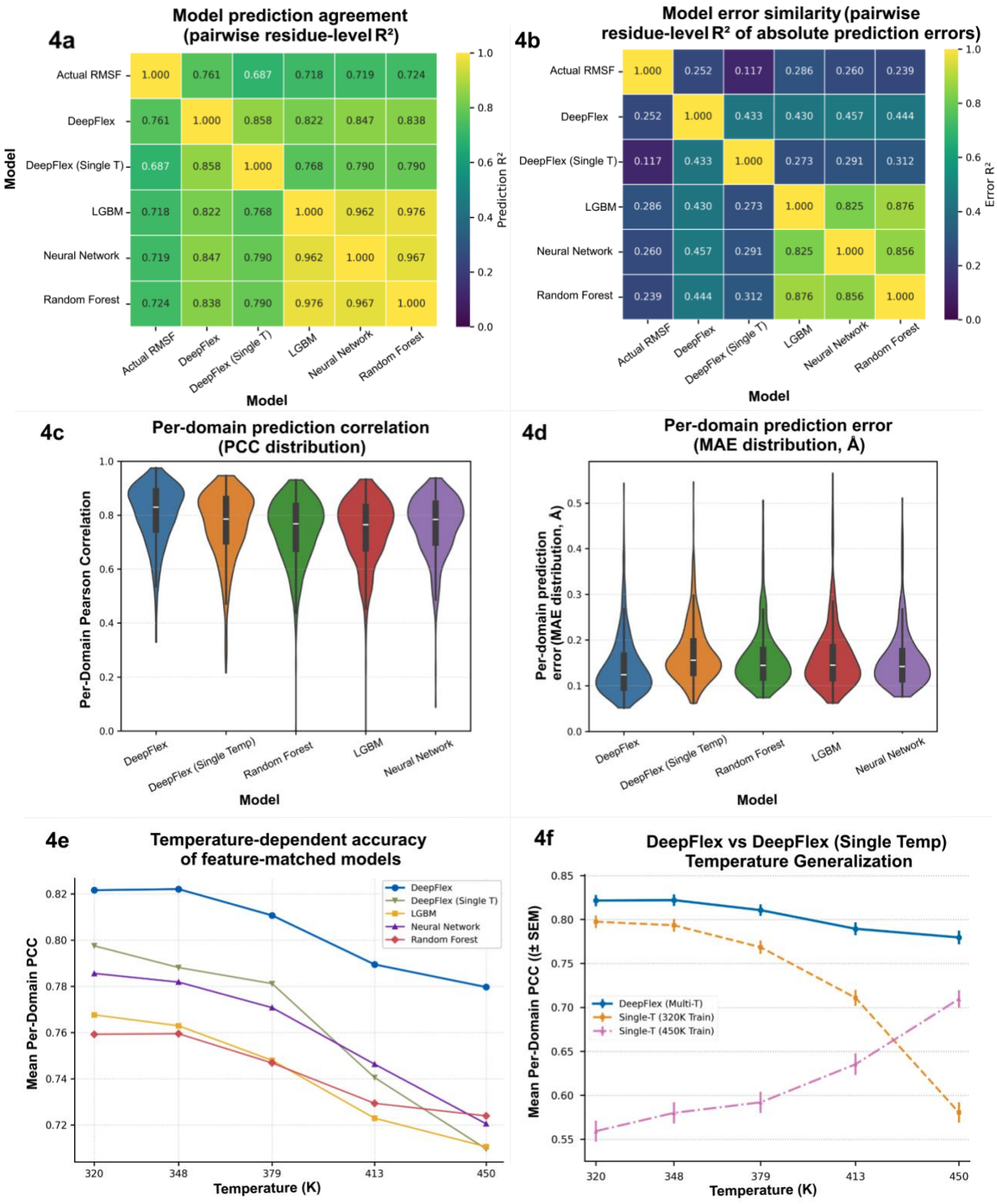

Figure 4 | Architecture, features, and temperature robustness of DeepFlex.
a, Residue-level agreement (pair-wise R^2) between every model's RMSF predictions and the MD ground truth, calculated over ≈ 57 000 Cα residues pooled across all proteins and five temperatures. **b,** Residue-level similarity (pair-wise R^2) of the *absolute* prediction errors for the same residues; high off-diagonal values indicate models that tend to mis-predict the same positions. **c, d,** Violin plots summarising, for each model, the distribution of per-domain Pearson correlation (PCC) c and mean absolute error (MAE, Å) d after averaging over temperature; boxes show median ± IQR, whiskers 1.5 × IQR (n = 128 protein domains). **e,** mean per-domain PCC versus temperature for DeepFlex and three feature-matched baselines trained on the identical input set; DeepFlex maintains the highest accuracy across the full 320 – 450 K range. **f,** Temperature-generalisation test: multi-temperature DeepFlex (blue) compared with two single-temperature variants trained at 320 K (orange) or 450 K (lilac). Points are mean per-domain PCC ± SEM; the 320 K-only model falls from PCC ≈ 0.80 at its training temperature to ≈ 0.58 at 450 K, whereas multi-temperature DeepFlex remains ≥ 0.78 throughout.

3.4 Robust prediction across diverse protein features

DeepFlex demonstrated robust performance across various intrinsic residue properties and structural contexts (Fig. 5). Prediction accuracy varied with local structure as expected: MAE was lowest for residues in β-sheets (0.109 Å) and α-helices (0.128 Å), and higher in loop/other regions (0.151 Å; Fig. 5a). Buried core residues (RSA ≤ 0.2) were predicted with lower error (MAE=0.113 Å) than solvent-exposed exterior residues (MAE=0.146 Å; Fig. 5b, d). Errors were elevated at sequence termini relative to central regions (Fig. 5c).

Performance also varied moderately by amino acid type (Fig. 5e): smaller hydrophobic residues showed the lowest errors, while charged and certain small/polar/unique residues showed slightly higher errors. Across all structural contexts, DeepFlex consistently outperformed individual baseline models (Fig. 5a-d).

Comparing baseline performance across structural contexts revealed complementary strengths. ESM-Only performed better in loops, termini, and surface regions. Conversely, VoxelFlex-3D performed better in β-sheets (MAE 0.148 Å vs. ESM-Only's 0.155 Å) and showed competitive performance in hydrophobic cores. DeepFlex consistently outperformed both specialized models across all structural elements (MAE ranging from 0.109 Å in β-sheets to 0.151 Å in loops), demonstrating the advantage of its integrated approach.

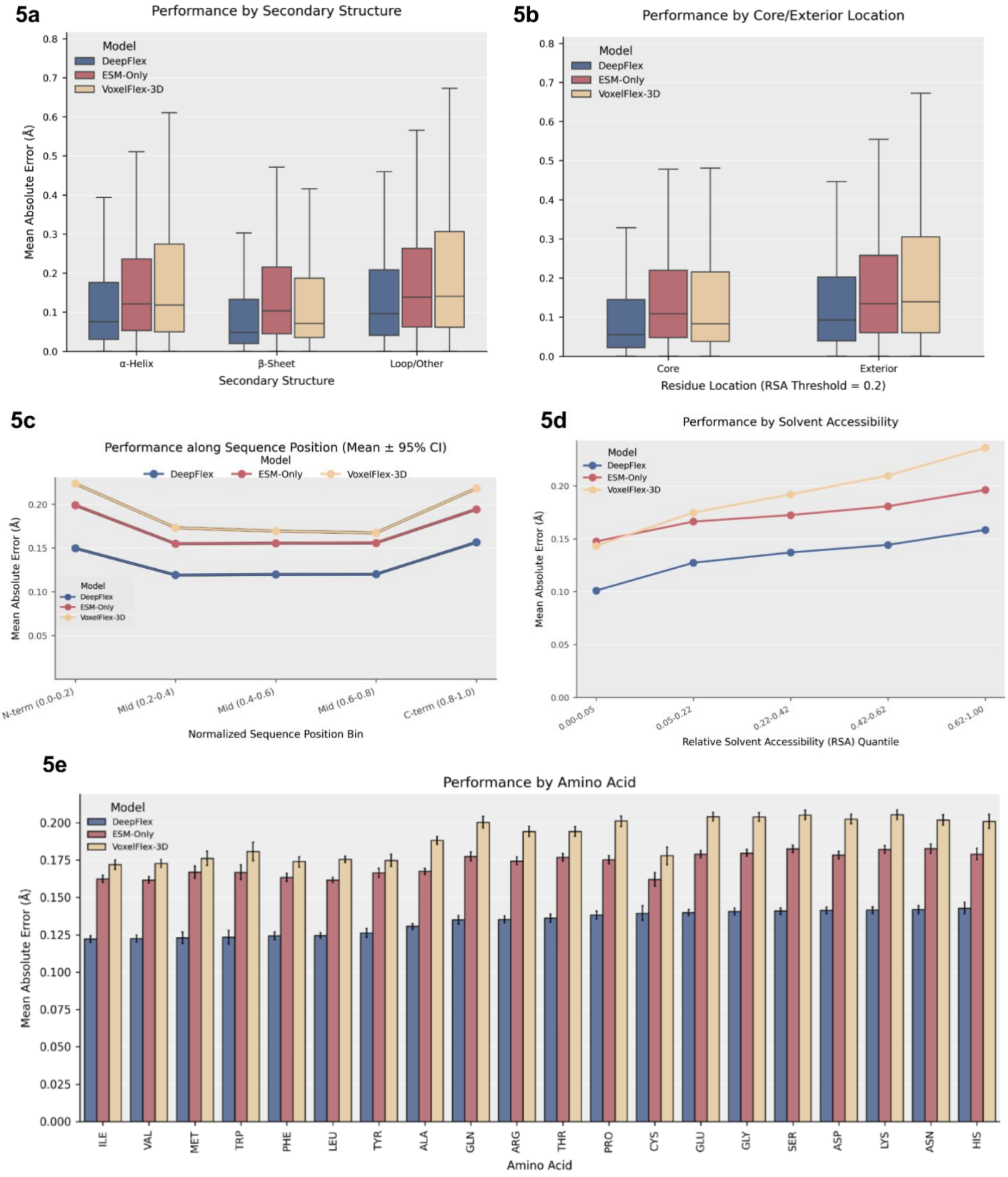

Figure 5 | Performance of DeepFlex and baselines across diverse residue contexts. Mean Absolute Error (MAE, Å) on the holdout set for DeepFlex (blue), ESM-Only (red/orange), and VoxelFlex-3D (yellow/gold), stratified by residue features. **(a)** MAE distributions by secondary structure (DSSP classification). **(b)** MAE distributions by core (RSA ≤ 0.2) versus exterior location. Box plots in (a, b) show median (line), interquartile range (box), and 1.5×IQR whiskers. **(c)** Mean MAE (points) ± 95% confidence intervals versus normalized sequence position (5 bins). **(d)** Mean MAE versus relative solvent accessibility (RSA) quantile (5 bins). **(e)** Mean MAE ± 95% confidence intervals (error bars) versus amino acid type, sorted by increasing DeepFlex MAE.

3.5 Case studies illustrate prediction characteristics

To assess performance in real structural contexts, we examined five representative domains at their physiological temperature spanning ordered scaffolds, disulfide stabilised loops and temperature adapted folds (Fig. 6).

For well packed proteins, DeepFlex reproduced experimental trends with near atomic fidelity. In the C terminal jelly roll of the peridinin–chlorophyll protein (PCP; PDB 1PPR, chain M), the model preserved the rigid pigment binding core (residues 244–258, RMSF < 0.25 Å) while correctly elevating fluctuations at the N terminus (RMSF ≈ 1.2 Å at residue 156) and at the extreme C helix (291–308), (Fig. 6a). Similarly, in the mixed α/β core of DNA topoisomerase I (Core III; 1K4T), DeepFlex sharply discriminated helical/strand scaffolds from surface loops (e.g. 476–480), yielding MAE < 0.07 Å and PCC > 0.92 at 320 K (Fig. 6a).

Prediction errors emerged when dynamics depended on local covalent constraints or ligand specific motions. In the disulfide rich soybean protease inhibitor (SMPI; 1BHU), the model underestimated flexibility of the inhibitory loop (61–69) by ~40%, raising the domain averaged MAE to 0.18 Å (Fig. 6b). A similar pattern appeared in inhibitor bound influenza B neuraminidase (1B9V): the β propeller core was well captured, but surface loops—including the 150 loop critical for catalysis—were mis-ranked, depressing PCC to 0.51, (Fig. 6b)

Finally, the thermophilic TyrRS C-terminal domain (1H3E) demonstrates DeepFlex's temperature-dependent prediction capabilities (Fig. 6c). At 320K, predictions closely matched MD (MAE = 0.042Å, PCC = 0.97), while increasing temperature accurately recapitulated differential mobility changes—pronounced softening in the N-terminal linker (346-351) with minimal helical core changes. While DeepFlex tends to predict slightly more rigid structures than MD simulations, the colour differences in protein visualizations have been intentionally

enhanced to highlight prediction patterns. The 1D RMSF traces (Fig. 6c, bottom) reveal the true relationship, showing consistently strong correlations throughout the temperature range (PCC = 0.97 at 320K to 0.92 at 450K), confirming DeepFlex's ability to capture region-specific thermal adaptation.

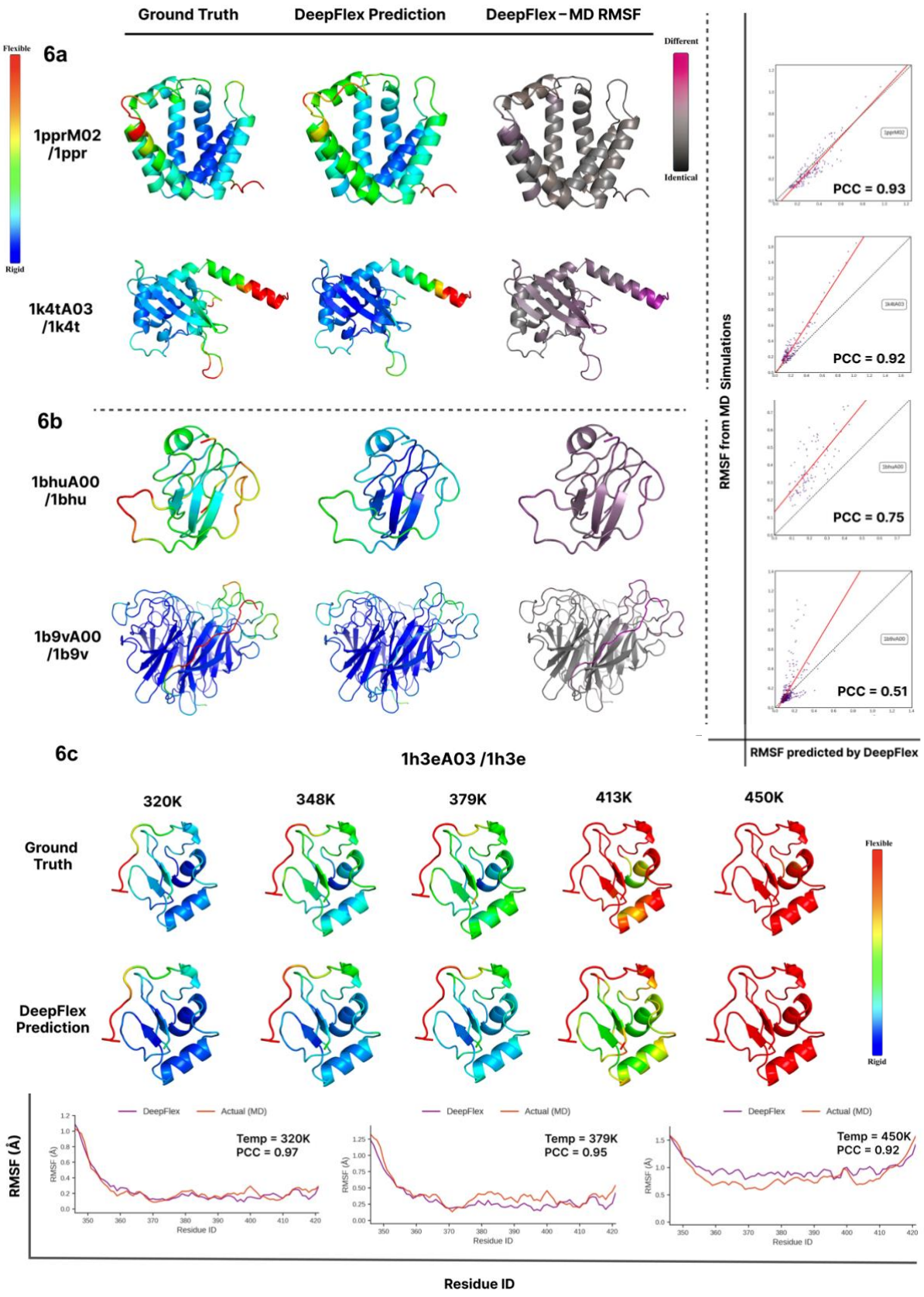

Figure 6 | Context specific performance of DeepFlex across diverse protein folds. a, High fidelity cases (320 K). Peridinin–chlorophyll protein C terminus (1PPR) and DNA topoisomerase I Core III (1K4T). Columns: MD RMSF map, DeepFlex prediction, prediction–MD difference (purple ≤ –0.8 Å → grey ≈ 0 → brown ≥ 0.8 Å) and residue level scatter plot (Predicted vs MD; Pearson r annotated). **b,** Challenging cases (320 K). Disulfide stabilised SMPI (1BHU) and inhibitor bound neuraminidase (1B9V) reveal underestimation of constrained or ligand modulated loops**. c**, Temperature generalisation. Thermophilic TyrRS C terminal domain (1H3E): top, RMSF heat maps from 320 to 450 K (MD vs DeepFlex); bottom, 1D RMSF traces highlighting the N terminal linker (346–351). All structures are coloured blue (low) → red (high) by RMSF; visualisations rendered in PyMOL).

4. Discussion:

4.1 Accurate and Computationally Efficient Prediction

DeepFlex represents a signficant advancement in protein dynamics prediction, achieving high correlation (PCC = 0.81 ± 0.02) and low error (MAE = 0.14 ± 0.06 Å) against MD-derived RMSF values across diverse thermal environments. This correlation approaches the inter-replicate reproducibility of the simulation data itself (inter-replicate PCC = 0.83 ± 0.04)[17], validating our hypothesis that a multi-modal temperature-conditioned neural network architecture can effectively capture the complex relationship between protein structure, sequence, and thermal flexibility while delivering predictions orders of magnitude faster than MD (40ms/protein).

4.2 Biophysical Validation and Structure-Function Relationships

DeepFlex's predictions align with established principles of protein dynamics, confirming the model has learned meaningful physical relationships rather than statistical artifacts. The model correctly reproduces three fundamental biophysical patterns: (1) the hierarchy of stability among secondary structures (β-sheets > α-helices > loops), (2) the distinction between rigid core and flexible surface residues, and (3) the temperature-dependent "thermal softening"[5] behaviour expected from statistical mechanics.

Our case studies (Fig. 6) demonstrate these capabilities across diverse structural contexts. Well-structured domains show atomic-level precision, while thermophilic proteins reveal correct prediction of differential thermal responses—with specific regions becoming progressively mobile at high temperatures while cores remain stable. These region-specific adaptations capture realistic thermal behaviour missing from static models, demonstrating the model's ability to learn biologically meaningful thermal response patterns.

4.3 Mechanistic Basis for Temperature Generalization

Training across a wide temperature range (320K-450K) serves multiple purposes: it accelerates sampling of rare but functionally relevant high-energy conformations, establishes a robust temperature-flexibility relationship preventing overfitting, and reveals the hierarchy of structural vulnerability through progressive thermal destabilization - insights that would remain hidden at physiological temperatures alone.

DeepFlex's integrated temperature-aware design explains its performance advantage over feature-matched baselines. Error correlation analysis revealed standard models (Random Forest, LightGBM, Neural Network) exhibit highly correlated errors ($R^2 \geq 0.83$), suggesting shared systematic limitations. In contrast, DeepFlex's error profile correlates weakly with these baselines ($R^2 \approx 0.43$-0.46), demonstrating its temperature-conditioned attention mechanism enables fundamentally different feature integration.

This mechanism operates through several key pathways. First, the learnable temperature embedding creates a continuous representation of thermal energy that modulates query-key interactions in the self-attention mechanism, dynamically adjusting how residues "attend" to each other based on thermal context. Second, these embedding biases value projections, enabling temperature-specific feature weighting that captures the differential impact of thermal energy on various structural elements (e.g., loops versus β-sheets). Finally, by integrating temperature information throughout the network rather than as a simple scalar input, DeepFlex learns complex, non-linear relationships between thermal energy and local mobility that standard architectures struggle to capture.

Input baseline models exhibit complementary mechanistic strengths reflecting their architectures. ESM-Only leverages evolutionary conservation patterns and excels in conformationally variable regions where sequence-encoded flexibility propensities dominate

rather than strict packing—specifically loops termini, and solvent-exposed surfaces (Fig. 5a-d). Conversely, VoxelFlex-3D captures spatial geometry and non-covalent interaction networks, performing better in regions governed by strict structural constraints, such as β-sheets and hydrophobic cores. DeepFlex's integrates these complementary signals, achieving superior performance across all structural elements by adaptively weighting features according to both local environment and thermal context.

This advantage is most pronounced at physiological temperatures (18% improvement at 320K) but diminishes as temperature increases (3% at 450K). This convergence occurs because at extreme temperatures, protein dynamics become increasingly stochastic as thermal energy progressively disrupts native contacts and secondary structures. When random thermal motion dominates over the specific structural constraints that DeepFlex is designed to model, all prediction methods face similar fundamental limitations, regardless of their architectural sophistication.

4.4 Positioning DeepFlex Relative to Existing Methods

Recent protein flexibility predictors have shown considerable promise. Flexpert[19] achieves high correlation (PCC ≈ 0.83) with MD simulations at 300K using protein language model embeddings and structural features. Similarly, RMSF-net[21] successfully integrates cryo-EM data with 3D convolution networks, reporting PCC ≈ 0.76 at 300K.

Direct comparison between these methods is challenging due to differences in training datasets and primary tasks - Flexpert was trained on ATLAS at 300K while RMSF-net focuses on cryo-EM data integration. Nevertheless, DeepFlex's most distinctive contribution is its robust temperature generalization. While Flexpert shows rapidly deteriorating performance outside its training temperature (from PCC ≈ 0.66 at 320K to PCC ≈ 0.33 at 450K on mdCATH[19]), DeepFlex maintains strong performance across the entire 320K-450K

span, never dropping below PCC ≈ 0.78. This thermal robustness is enabled by the temperature-conditioning architecture, as evidenced by our single-temperature model variants which similarly degrade when tested far from their training conditions (Fig. 4f).

While the performance improvement over feature-matched baselines is modest (ΔPCC ≈ 0.06, Table 1), it remains consistent across all structural contexts and temperature ranges. More importantly, DeepFlex's architecture offers a substantial qualitative advantage: it provides a differentiable end-to-end solution that integrates directly with gradient-based optimization frameworks for protein design, unlike ensemble methods that require separate training and inference pipelines.

4.5 Methodological Limitations and Scope

Despite its advances, DeepFlex exhibits systematic limitations. Analysis by amino acid type reveals that hydrophobic residues consistently show lower errors, while polar residues, residues with unusual backbone geometry (GLY, PRO) or those participating in specific chemical interactions (CYS) present greater challenges. This pattern suggests DeepFlex would benefit from improved representation of non-canonical residue properties.

The mdCATH dataset[17] composition introduces bias toward single, globular domains in apo states, limiting generalization to complex contexts. Our failure analysis highlights two specific scenarios where this limitation manifests: a disulfide-rich soybean protease inhibitor (1BHU) and inhibitor-bound influenza neuraminidase (1B9V). In both cases, DeepFlex underestimated loop flexibility, failing to capture the specialized dynamics of functional loops constrained by specific structural features - particularly the disulfide-stabilized inhibitory loop in SMPI and the ligand-interacting catalytic 150-loop in neuraminidase. This underestimation likely stems from the limited representation of these specialized motifs in the training data.

Performance degrades for extremely flexible segments (RMSF > 1.0 Å) and at the highest temperature (450K), where unfolding occurs. Statistical analysis reveals poorer correlation for rigid residues compared to moderately flexible ones, indicating difficulty distinguishing small differences among the most rigid structural elements despite their low absolute error values. Additionally, our validation relies exclusively on MD-derived data rather than direct experimental measurements—while MD simulations provide physically realistic approximations of protein dynamics, they themselves represent computational models with inherent limitations.

4.6 Future Directions and Broader Impact

To address DeepFlex's current limitations, we propose several key directions for future development: (i) Data Expansion: Broaden training data to include multi-domain architectures, membrane proteins, and ligand-bound states to improve generalization to diverse protein contexts; (ii) Architectural Improvements: Implement end-to-end training with the VoxelFlex 3D CNN to better capture structural constraints like disulfide bonds, allowing the model to better integrate spatial relationships (iii) Experimental Validation: Directly validate DeepFlex predictions against experimental dynamics data.

Wayment-Steele et al.[37] recently demonstrated that deep learning can predict microsecond-millisecond protein dynamics by leveraging missing NMR peak assignments as indicators of conformational exchange. Integrating their RelaxDB dataset as additional training signals could bridge the computational-experimental gap, providing comprehensive dynamics predictions spanning ps-ms timescales under varying thermal conditions and potentially revealing how temperature modulates both fast local fluctuations and slower conformational exchanges essential for function.

Overall, DeepFlex represents a significant advancement toward routinely incorporating the essential dynamic dimension of proteins into computational biology workflows. By integrating evolutionary information, multi-scale structural features, and explicit temperature conditioning via attention, our model achieves accurate, temperature-aware prediction of residue-level protein flexibility orders of magnitude faster than molecular dynamics simulations. This computational efficiency enables applications previously hindered by the prohibitive cost of MD, particularly in evolutionary biophysics where it facilitates analyses of thermal adaptation mechanisms across protein families. In protein engineering, where the objective is shifting from designing stable scaffolds towards controlling functional motions[20], DeepFlex introduces a differentiable flexibility objective that can be jointly optimized alongside stability—a capability particularly valuable for the rational design of enzymes, biosensors, and therapeutic proteins whose activities rely on carefully tuned conformational dynamics. The model's ability to generalize across a 130K temperature range while capturing key biophysical determinants of flexibility enables a paradigm shift from static structural analysis to more realistic modelling of the conformational ensembles that underlie protein function, providing a powerful new framework for dynamics-aware protein engineering and analysis.

Acknowledgements:

I would like to express my heartfelt thanks to everyone who supported me throughout this Honours dissertation journey including the full Biochemistry Honours Programme Faculty.

I am incredibly grateful to my supervisor, Dr. Chris Wood, whose expertise, patience, and enthusiasm made this research possible. His thoughtful guidance helped shape not only this project but also my approach to scientific inquiry.

I also owe special thanks to Dr. Eugene Shrimpton-Phoenix for his day-to-day mentorship. His willingness to troubleshoot challenges, discuss ideas, and share his technical knowledge was a constant source of support. From initial brainstorming sessions to the final draft reviews, his insights have been invaluable, and this would not have been possible without him.

The entire Wells Wood Research group deserves my sincere appreciation. The collaborative atmosphere and feedback during lab meetings created an ideal environment for exploring new ideas. I'm particularly thankful for the computational resources, debugging help, and the constant support that made even the most challenging days enjoyable.

This dissertation represents not just my work, but the collective support of these wonderful mentors and colleagues who believed in this project and in me.

Bibliography:

1. Henzler-Wildman KA, Kern D (2007) Dynamic personalities of proteins, Nature 450, 964-972. https://doi.org/10.1038/nature06522
2. Guo J, Zhou H-X (2016) Protein allostery and conformational dynamics, Chem Rev 116, 6503-6515. https://doi.org/10.1021/acs.chemrev.5b00590
3. Agarwal PK, Bernard DN, Bafna K, Doucet N (2020) Enzyme dynamics: Looking beyond a single structure, ChemCatChem 12, 4704-4720. https://doi.org/10.1002/cctc.202000665
4. Motlagh HN, Wrabl JO, Li J, Bahar I (2014) The ensemble nature of allostery, Nature 508, 331-339. https://doi.org/10.1038/nature13001.
5. Saavedra HG, Wrabl JO, Anderson JA, Li J, Hilser VJ (2018) Dynamic allostery can drive cold adaptation in enzymes, Nature 558, 324-328. https://doi.org/10.1038/s41586-018-0183-2.
6. Dong YW, Liao ML, Meng XL, Somero GN (2018) Structural flexibility and protein adaptation to temperature: Molecular dynamics analysis of malate dehydrogenases of marine molluscs, Proc Natl Acad Sci U S A 115, 1274-1279. https://doi.org/10.1073/pnas.1718910115.
7. Gerasimavicius L, Liu X, Marsh JA (2020) Identification of pathogenic missense mutations using protein stability predictors, Sci Rep 10, 15387. https://doi.org/10.1038/s41598-020-72404-w.
8. Bramer D, Wei G-W (2018) Blind prediction of protein B-factor and flexibility, J Chem Phys 149, 134107. https://doi.org/10.1038/s41598-020-72404-w
9. Kay LE (2016) New views of functionally dynamic proteins by solution NMR spectroscopy, J Mol Biol 428, 323-331. https://doi.org/10.1016/j.jmb.2015.12.003.
10. Mittermaier AK, Kay LE (2009) Observing biological dynamics at atomic resolution using NMR, Trends Biochem Sci 34, 601-611. https://doi.org/10.1016/j.tibs.2009.07.004.
11. Murata K, Wolf M (2022) Cryo-EM for the study of dynamic biological macromolecules, Adv Protein Chem Struct Biol 128, 1-50. doi:10.1016/bs.apcsb.2021.11.001
12. Karplus M, McCammon JA (2002) Molecular dynamics simulations of biomolecules, Nat Struct Biol 9, 646-652.
13. Bahar I, Lezon TR, Yang LW, Eyal E (2010) Global dynamics of proteins: bridging between structure and function, Annu Rev Biophys 39, 23-42.
14. Bakan A, Meireles LM, Bahar I (2011) ProDy: protein dynamics inferred from theory and experiments, Bioinformatics 27, 1575-1577.

15. Haliloglu T, Bahar I, Erman B (1997) Gaussian dynamics of folded proteins, Phys Rev Lett 79, 3090-3093.

16. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, Bridgland A, Meyer C, Kohl SAA, Ballard AJ, Cowie A, Romera-Paredes B, Nikolov S, Jain R, Adler J, Back T, Petersen S, Reiman D, Clancy E, Wrotek P, Zielinski M, MacCarthy M, Vinyals O, Hubers N, Dieleman S, Sifre L, Kavukcuoglu K, Hassabis D (2021) Highly accurate protein structure prediction with AlphaFold, Nature 596, 583-589. doi:10.1038/s41586-021-03819-2

17. Mirarchi A, Giorgino T, De Fabritiis G (2024) mdCATH: a large-scale MD dataset for data-driven computational biophysics, Sci Data 11, 1299.

18. Vander Meersche Y, Cretin G, Gheeraert A, Gelly J-C, Galochkina T (2024) ATLAS: protein flexibility description from atomistic molecular dynamics simulations, Nucleic Acids Res 52, D384-D392. doi:10.1093/nar/gkad1084

19. Kouba P, Planas-Iglesias J, Damborsky J, Sedlar J, Mazurenko S, Sivic J (2024) Learning to engineer protein flexibility, arXiv:2412.18275v2 [q-bio.BM] [preprint].

20. Campbell E, Kaltenbach M, Correy GJ, Carr PD, Porebski BT, Livingstone EK, Afriat-Jurnou L, Buckle AM, Weik M, Hollfelder F, Tokuriki N, Jackson CJ (2016) The role of protein dynamics in the evolution of new enzyme function, Nat Chem Biol 12, 944-950.

21. Song X, Bao L, Feng C, Chen Z, Huang Y, Li Y, Kihara D (2024) Accurate Prediction of Protein Structural Flexibility by Deep Learning Integrating Intricate Atomic Structures and Cryo-EM Density Information, Nat Commun 15, 5538. doi:10.1038/s41467-024-49858-x

22. Hayes T, Rao R, Akin H, Sofroniew NJ, Oktay D, Lin Z, Verkuil R, Tran VQ, Deaton J, Wiggert M, Badkundri R, Shafkat I, Gong J, Derry A, Molina RS, Thomas N, Khan YA, Mishra C, Kim C, Bartie LJ, Nemeth M, Hsu PD, Sercu T, Candido S, Rives A (2025) Simulating 500 million years of evolution with a language model, Science 387, 850-858

23. Ramakrishnan G, Baakman C, Heijl S, Vroling B, van Horck R, Hiraki J, Xue LC, Huynen MA (2023) Understanding structure-guided variant effect predictions using 3D convolutional neural networks, Front Mol Biosci 10, 1204157.

24. Burton F (2025) VoxelFlex [Software]. Available at: https://github.com/wells-wood-research.

25. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need, Adv Neural Inf Process Syst 30, 5998-6008.

26. Sillitoe I, Bordin N, Dawson N, Waman VP, Ashford P, Scholes HM, Pang CS, Woodridge L, Rauer C, Sen N, Abbasian M, Le Cornu S, Lam SD, Berka K, Varekova IH, Svobodova R, Lees J, Orengo CA (2021) CATH: increased structural coverage of functional space, Nucleic Acids Res 49, D266-D273.

27. Burton F (2025) mdcath-processor code [Software]. Available at: https://github.com/wells-wood-research.

28. Burton F (2025) mdcath-sampling code [Software]. Available at: https://github.com/wells-wood-research.

29. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, de Hoon MJL, Harvey M, Hetherington J, Holland RCG, Klenk M, Robinson A, Stajich JE, Tiessen A, van der Walt S, Weber J (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics, Bioinformatics 25, 1422-1423. doi:10.1093/bioinformatics/btp163

30. Kabsch W, Sander C (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, Biopolymers 22, 2577-2637. doi:10.1002/bip.360221211

31. Tien MZ, Meyer AG, Sydykova DK, Spielman SJ, Wilke CO (2013) Maximum allowed solvent accessibilities of residues in proteins, PLoS One 8, e80635. doi:10.1371/journal.pone.0080635

32. Wood C (2025) Aposteriori tool [Software]. Available at: https://github.com/wells-wood-research/aposteriori.

33. Loshchilov I, Hutter F (2019) Decoupled Weight Decay Regularization, arXiv:1711.05101 [cs.LG] [preprint].

34. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library, Adv Neural Inf Process Syst 32. arXiv:1912.01703 [cs.LG] [preprint].

35. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree, Adv Neural Inf Process Syst 30, 3146-3154.

36. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay É (2011) Scikit-learn: Machine Learning in Python, J Mach Learn Res 12, 2825-2830.

37. Wayment Steele HK, El Nesr G, Hettiarachchi R, Kariyawasam H, Ovchinnikov S, Kern D (2025) Learning millisecond protein dynamics from what is missing in NMR spectra, bioRxiv [preprint]. doi:10.1101/2025.03.19.642801

Appendix:

Methods:

This appendix provides expanded details on the methods summarized in the main manuscript body, corresponding to the references.

Appendix S1: Holdout Set Representation Index Details

The creation of a sequestered holdout set *prior* to any model development was critical for unbiased evaluation of model generalization. The protocol, implemented in the mdcath-sampling package, employed topology-aware stratified sampling based on the CATH v4.2.0 hierarchy.

Sampling Protocol:

1. **Grouping:** Domains were initially grouped by their full CATH classification (Class, Architecture, Topology, Homologous Superfamily).
2. **Homology Control:** A connectivity graph was constructed where nodes represent domains, and edges connect domains originating from the same PDB entry or belonging to the same CATH Homologous Superfamily. Entire connected components of this graph were treated as single units during sampling to prevent splitting highly homologous structures across the holdout and development sets.
3. **Stratified Sampling:**
    - **Large Topologies (≥10 unique domains/components):** K-means clustering (k determined dynamically, typically 3-7) was performed on aggregated domain-level features (including secondary structure percentages, mean RMSF across temperatures, core/exterior residue ratio, average relative solvent accessibility, and thermal stability profile metrics derived from RMSF vs. Temperature curves). Representative domains/components closest to cluster centroids were selected for the holdout set pool.
    - **Smaller Topologies (<10 unique domains/components):** Domains/components were sampled proportionally based on their representation in the overall dataset and feature space uniqueness to ensure coverage of rarer folds.
4. **Iteration and Validation:** The initial partition was evaluated based on distribution similarity (Kolmogorov-Smirnov tests on key features like length, RMSF, RSA), hierarchy coverage (ensuring representation across CATH levels), and stability profile coverage (comparing distributions of thermal response metrics). A composite **Representation Index (RI)** was calculated to quantify partition quality:

RI = (Similarity_Metrics_Geometric_Mean * Hierarchy_Coverage * Stability_Profile_Coverage)^(1/3)

Where Similarity_Metrics_Geometric_Mean is the geometric mean of p-values or similarity scores from statistical tests comparing feature distributions between the holdout and

development sets. Hierarchy_Coverage measures the fraction of CATH levels represented proportionally in both sets. Stability_Profile_Coverage measures the overlap in the distribution of thermal stability metrics. The sampling process (steps 2-3) was iterated with adjustments until RI ≥ 0.9, indicating high similarity and representative coverage in the holdout set.

This rigorous, iterative process yielded the final holdout set of 401 unique CATH domains (representing 287,866 residue instances across all temperatures, ~7.5% of initial domains), which was strictly isolated before any model training or hyperparameter tuning commenced.

Appendix S2: Development Dataset Preprocessing Details

The domains remaining after holdout set sequestration formed the initial pool for the development dataset. Preprocessing steps were applied using custom scripts within the mdcath-processor pipeline and pandas [*v1.3.0*].

Exclusion Criteria:

A total of 514 domains were excluded from the development set due to the following technical issues encountered during data loading or initial feature calculation:

- **BioPython Parsing Errors:** Incompatibility with BioPython.PDB parser (e.g., corrupted PDB files, unusual formatting).
- **DSSP Failures:** Errors during secondary structure assignment using the DSSP algorithm via BioPython's wrapper (e.g., missing atoms, chain breaks preventing DSSP execution).
- **PDB B-factor Query Errors:** Failures when querying experimental B-factors from the PDB API (e.g., PDB ID obsolete, B-factor data missing in the specific PDB file).
- **Residue Numbering Discontinuities:** Detected gaps or non-sequential resid values within the simulation data for a given domain, indicating potential simulation setup or data extraction problems.

Loading and Cleaning:

Residue-level data for the remaining domains was loaded from the aggregated mdCATH data file (originally derived from HDF5 sources). Initial cleaning steps using pandas included:

- Removing rows missing any essential fields: domain_id, temperature, resid, resname, rmsf.
- Standardizing common histidine protonation state variants (HSD, HSE, HSP) to the standard HIS residue name for consistent feature mapping and embedding lookup.

Instance Definition and Sequence Reconstruction:

- A unique instance key was created for each domain-temperature pair using the format domain_id@temperature.1f (e.g., 1a05A00@320.0) to uniquely identify each experimental condition instance.

- Data was grouped by this instance_key.
- Within each group, the data was sorted by the original residue ID (resid).
- The amino acid sequence and corresponding per-residue feature arrays (see Appendix S3) and target RMSF arrays were reconstructed based on this sorted order. This ensured correct alignment between sequence positions, features, and the target variable. The target RMSF (target_rmsf) used for training was the mean value across the five independent simulation replicas for that residue at that specific temperature, as provided by mdCATH.

NaN Feature Imputation:

Per-residue feature columns were checked for missing values (NaN) within each domain-temperature instance.

- NaN values were imputed using the **median** value of that specific feature calculated across all residues *within that same domain-temperature instance*. Using the instance-specific median preserves the local context of that protein under that specific thermal condition.
- In rare cases (<0.1% of all imputation events across the dataset) where the instance median could not be computed (e.g., if all values for a feature in that instance were NaN), a fallback value of **0.0** was used.

This comprehensive preprocessing pipeline yielded the final high-quality development set comprising 4,483 unique domains, representing 3,064,132 residue-temperature instances used for subsequent model training, validation, and testing.

Appendix S3: Per-Residue Descriptor Definitions and Processing

A set of per-residue descriptors, derived from both sequence information and the processed experimental PDB structures, were engineered to provide the model with relevant physicochemical and structural context. These calculations were performed using BioPython and custom scripts within the mdcath-processor pipeline, using the cleaned PDB structures described in the main Methods.

Feature List and Calculation Details:

- **Positional Information:**
    - normalized_resid: Residue index (resid) linearly scaled to the range [0, 1] within its domain. Calculated as (resid - min_resid) / (max_resid - min_resid) for the domain, handling single-residue domains by setting the denominator to 1. Provides relative position information.
    - protein_size: The total number of unique residues in the CATH domain. This is a global feature, constant for all residues within a given domain.
- **Secondary Structure:**

- secondary_structure_encoded: A three-state numerical encoding based on DSSP assignments obtained via BioPython's DSSP module.
    - 0: Helix (DSSP codes: H, G, I)
    - 1: Sheet (DSSP codes: E, B)
    - 2: Coil/Other (DSSP codes: T, S, C, -)

- **Solvent Exposure:**
    - relative_accessibility: Relative Solvent Accessibility (RSA), calculated as the residue's Solvent Accessible Surface Area (SASA) divided by its theoretical maximum ASA. SASA was computed using the Shrake-Rupley algorithm implemented in BioPython (Bio.PDB.SASA.ShrakeRupley). Theoretical maximum ASA values were taken from Tien et al., 2013
    - core_exterior_encoded: A binary encoding derived from relative_accessibility.
        - 0: Core (RSA $\leq$ 0.20)
        - 1: Exterior (RSA > 0.20)

- **Backbone Conformation:**
    - phi_norm, psi_norm: Backbone dihedral angles (φ and ψ) extracted from the cleaned PDB structures using BioPython (Bio.PDB.Polypeptide.PPBuilder). Missing dihedral values (e.g., for N-terminal residues lacking φ or C-terminal residues lacking ψ) were imputed as 0.0 degrees. The raw degree values were then linearly scaled to the range [-1, 1] by dividing by 180.0.

- **Experimental Flexibility:**
    - bfactor_norm: Experimental B-factors obtained from the ATOM records of the corresponding cleaned PDB structure. To account for variations in scale and resolution across different PDB files, B-factors were standardized within each domain by calculating the Z-score: (B_factor - mean(B_factors_domain)) / stddev(B_factors_domain). For domains where the standard deviation of B-factors was near zero (< 1e-6), or where B-factors were missing, a value of 0.0 was assigned to bfactor_norm.

Imputation Summary: As noted in Appendix S2, missing values (NaN) arising during feature calculation (e.g., for terminal residues missing dihedrals, or PDBs missing B-factors) were imputed using the instance-specific median, with a 0.0 fallback used in rare cases (<0.1% of all feature values).

Final Scaling: All resulting numerical descriptors, along with voxel_rmsf and temperature (see Appendix S4 and main Methods), were Min-Max scaled to the [0, 1] range using parameters derived *solely* from the training set partition. These scaling parameters were saved

(feature_normalization.json, temp_scaling_params.json) and applied consistently to the validation and test sets during evaluation and prediction.

Appendix S4: VoxelFlex Architecture and Training Details

To incorporate detailed local 3D structural context, the voxel_rmsf feature was generated using **VoxelFlex**, an independent deep learning model developed specifically for this project VoxelFlex predicts RMSF directly from a voxelized representation of the local atomic environment around each residue's Cα atom.

Voxelization:

- **Input Structures:** The same cleaned PDB structures used for per-residue descriptor calculation (one per CATH domain) served as input.
- **Tool:** Voxelization was performed using the Aposteriori tool
- **Grid Parameters:** For each residue, a cubic voxel grid was generated, centered on its Cα atom.
    - Dimensions: 21 x 21 x 21 voxels.
    - Resolution: 12 Å edge length (~0.57 Å per voxel).
- **Encoding Scheme (CNOCBCA):** Each voxel grid contained 5 channels, representing the atomic density of specific backbone/Cβ atom types within that voxel space:
    - Channel 0: Carbon (C) atoms (excluding Cα).
    - Channel 1: Nitrogen (N) atoms.
    - Channel 2: Oxygen (O) atoms.
    - Channel 3: Alpha Carbon (Cα) atoms.
    - Channel 4: Average Beta Carbon (Cβ) position (a single point density representing the average Cβ location for non-glycine residues within the box, primarily useful for sidechain orientation context). Density was typically represented using Gaussian smoothing around atom positions.

VoxelFlex Architecture (Multipath ResNet):

- **Rationale:** VoxelFlex employs a 3D Convolutional Neural Network (CNN) to process the voxel grids. Initial experiments compared standard 3D CNNs, Dilated ResNets, and a Multipath ResNet. The Multipath ResNet architecture demonstrated superior performance in capturing multi-scale spatial features relevant to local flexibility.
- **Architecture Overview:**

    - The architecture uses parallel convolutional paths with differing kernel sizes (e.g., 3x3x3, 5x5x5, 7x7x7) to analyze the voxel grid at multiple spatial resolutions simultaneously.
    - Residual connections are employed within each path to facilitate training deeper networks.
    - Features from parallel paths are fused (e.g., via concatenation followed by a 1x1x1 convolution) before passing through final layers.
    - Global average pooling reduces the spatial dimensions before the regression head.
- **Specific Parameters:** Dilated ResNet variant used base_filters=32, num_residual_blocks=4, channel_growth_rate=1.5, dropout_rate=0.3.

VoxelFlex Training:

- **Dataset:** VoxelFlex was trained **independently** from the main DeepFlex model, using *only* the **training split** of the mdCATH dataset. This prevents data leakage from the validation/test/holdout sets into the voxel_rmsf feature.
- **Input (X):** The 5-channel voxel grids (21x21x21) generated as described above.
- **Target (Y):** The corresponding replica-averaged ground truth Cα RMSF values from the mdCATH training set for each residue.
- **Training Objective:** Minimize Mean Squared Error (MSE) between VoxelFlex predictions and the ground truth RMSF values.

Output Feature (voxel_rmsf):

- After training, the optimized VoxelFlex model was used to predict RMSF values for *all* residues (across train, validation, test, and holdout sets) using only their respective voxel grids as input.
- These VoxelFlex-generated RMSF predictions (scalar value per residue) were saved and used directly as the voxel_rmsf numerical input feature for the main **DeepFlex** model. This feature provides DeepFlex with a representation of the local 3D structural environment, implicitly learned by the 3D CNN.

Appendix S5: DeepFlex Architecture Layer Specifications & Training Details

DeepFlex Architecture Layer Specifications

The **DeepFlex** architecture integrates multi-modal inputs via specialized MLP blocks and a temperature-conditioned attention mechanism. Below are the specific layer configurations:

- **ESM Embeddings:** Pre-trained esmc_600m provided 1152-dimensional embeddings. These weights were **frozen**.
- **Positional Encoding:** Standard sinusoidal positional encodings with 1152 dimensions were added element-wise to ESM embeddings.
- **Feature Fusion MLP:** This block fuses the concatenated per-residue descriptors and voxel_rmsf feature with the position-aware ESM embeddings.
    - Input: Concatenated vector (1152D ESM + N_features * D_per_residue_descriptor).
    - Layers:
        1. nn.Linear(in_features=1152 + N_features, out_features=1152): Projects concatenated features to match ESM dimension.
        2. nn.GELU(): Activation function.
        3. nn.LayerNorm(normalized_shape=1152): Layer normalization.
        4. nn.Dropout(p=0.1): Dropout for regularization (rate based on config in original code).
    - Output: Unified 1152-dimensional representation per residue.
- **Temperature Encoding MLP (TemperatureEncoding):** Transforms the [0, 1] scaled scalar temperature into a dense embedding.
    - Input: Scalar temperature (reshaped to [batch_size, 1]).
    - Layers:
        1. nn.Linear(in_features=1, out_features=16): Project to 16 dimensions.
        2. nn.GELU(): Activation.
        3. nn.LayerNorm(normalized_shape=16): Layer normalization.
    - Output: 16-dimensional temperature embedding per instance ([batch_size, 16]).
- **Temperature-Conditioned Attention (AttentionWithTemperature):** Modulates standard Multi-Head Self-Attention (MHA) with temperature.
    - **MHA:** Standard implementation with:
        - embed_dim = 1152
        - num_heads = 8
        - dropout = 0.1
        - batch_first = True

    - **Temperature Conditioning:** The 16D temperature embedding is passed through three separate linear layers to generate additive biases for Query (Q), Key (K), and Value (V) projections *before* the scaled dot-product attention calculation within the MHA module.
        - temp_to_q = nn.Linear(16, 1152)
        - temp_to_k = nn.Linear(16, 1152)
        - temp_to_v = nn.Linear(16, 1152)
        - (Usage: q = projected_q + temp_bias_q, etc.)
    - **Normalization/Residuals:** Pre-Layer Normalization (nn.LayerNorm(1152)) applied before the attention block, and a residual connection adds the input of the LayerNorm to the output of the attention block's final projection.
- **Regression Head MLP:** Predicts the final scalar RMSF value.
    - Input: 1152-dimensional final residue representation from the attention block.
    - Layers:
        1. nn.LayerNorm(normalized_shape=1152)
        2. nn.Linear(in_features=1152, out_features=128)
        3. nn.GELU()
        4. nn.Dropout(p=0.1)
        5. nn.Linear(in_features=128, out_features=1): Final prediction layer.
    - Output: Scalar RMSF prediction per residue.

DeepFlex Training Details

The DeepFlex model was trained using the PyTorch framework [*v2.5.1*]

- **Optimizer:** AdamW was used with the following parameters:
    - Learning Rate (LR): 1.0e-4
    - Betas: β1=0.9, β2=0.999
    - Epsilon (ε): 1.0e-8
    - Weight Decay: 0.01. Applied selectively only to weight parameters (typically excluding biases and LayerNorm parameters) to prevent decay of normalization terms.
- **Loss Function:** Mean Squared Error (MSE) between the model's predicted RMSF and the ground truth replica-averaged RMSF values.
- **Batching:**

    - Per-GPU Batch Size: 8
    - Gradient Accumulation Steps: 4
    - Effective Batch Size: 32 (8 * 4)
- **Regularization & Stability:**
    - Gradient Clipping: Applied with max_norm = 1.0 to prevent exploding gradients.
    - Dropout: Used within MLP blocks and attention as specified above (typically p=0.1).
- **Learning Rate Scheduling:** A ReduceLROnPlateau scheduler monitored the Pearson correlation coefficient on the **validation set**. The learning rate was reduced by a factor of 0.5 if the validation Pearson correlation did not improve for 5 consecutive epochs.
- **Early Stopping:** Training was stopped prematurely if the validation Pearson correlation did not improve for 10 consecutive epochs. This occurred after **33** epochs in the final training run.
- **Hardware:** Training was performed on NVIDIA Quadro RTX 8000 GPUs.
- **Training Time:** Approximately 12 hours.
- **Checkpointing:** The model checkpoint corresponding to the epoch with the highest validation Pearson correlation (epoch 33) was selected for final evaluation and prediction. Periodic checkpoints were also saved during training.
- **Figure S1 | DeepFlex Training Curves.** (a) Mean Squared Error (MSE) loss for training (blue) and validation (orange) sets per epoch. (b) Pearson correlation coefficient for training (blue) and validation (orange) sets per epoch. The dashed vertical line indicates the epoch (33) with the best validation Pearson correlation, selected for the final model. The right y-axis (green, log scale) shows the learning rate adjustments made by the scheduler.

Appendix S6: EnsembleFlex Baseline Hyperparameter Tuning Details

To provide a robust comparison against traditional machine learning methods using the same rich feature set, ensemble baselines (**EnsembleFlex**) were trained and evaluated.

- **Models:** LightGBM and Random Forest (RF) implemented using Scikit-learn.

- **Feature Set:** Crucially, these models were trained using the **identical, complete feature set** fed into the final DeepFlex MLP layers *before* the attention mechanism. This included:
    - Frozen ESM-C 600M embeddings (1152 dimensions per residue).
    - All per-residue descriptors (normalized position, SS, RSA, core/exterior, normalized dihedrals, normalized B-factors - see Appendix S3).
    - The voxel_rmsf feature generated by the VoxelFlex model (see Appendix S4).
    - The Min-Max scaled simulation temperature.

This ensures a direct comparison of the DeepFlex architecture's ability to integrate features versus standard ensemble techniques given the same information.

- **Hyperparameter Optimization:** To ensure fair comparison and optimize baseline performance, hyperparameters for both LightGBM and Random Forest were tuned using **Randomized Search Cross-Validation** (RandomizedSearchCV from Scikit-learn) on the **training set**.
    - **Cross-Validation:** 5-fold cross-validation was typically used within the training set.
    - **Search Space:** A predefined grid of relevant hyperparameters was explored (e.g., for RF: n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features; for LightGBM: n_estimators, learning_rate, num_leaves, max_depth, regularization terms, subsampling fractions).
    - **Iterations:** A fixed number of parameter combinations (e.g., 50-100 iterations) were randomly sampled and evaluated.
    - **Scoring Metric:** Optimization typically aimed to minimize negative Mean Squared Error (equivalent to maximizing R^2 or minimizing MSE).
- **Final Model:** The best hyperparameter set found during Randomized Search CV was used to train the final EnsembleFlex model on the entire training set.
- **Ablation Study (RF (No ESM Feats)):** A separate Random Forest model was trained using the identical procedure and feature set *except* for the exclusion of the ESM-C embeddings. This specifically quantified the contribution of the deep evolutionary information provided by the language model embeddings.
- **Single-Temperature Models:** As mentioned in the main text, DeepFlex variants trained on individual temperature slices were also evaluated. Performance was comparable to the unified model at those specific temperatures, demonstrating the unified model's ability to generalize without significant performance loss compared to specialized models trained on single temperature slices only. Detailed results for these models can be found in the associated analysis code and output files.

Appendix S7: Monte Carlo Dropout Uncertainty Estimation Details

Uncertainty estimation for DeepFlex predictions was performed using the Monte Carlo (MC) Dropout technique

- **Mechanism:** MC Dropout leverages the standard dropout layers present in the network (specifically within the Fusion MLP and Regression Head MLPs in DeepFlex) during *inference* time, which are normally deactivated during evaluation. By performing multiple stochastic forward passes (N=10 in this work) through the network with dropout *active* (using the same dropout rates as during training, e.g., p=0.1), we obtain a distribution of predictions for each residue.
- **Number of Passes (N=10):** The choice of 10 forward passes was based on a balance between obtaining a reasonably stable estimate of the prediction variance and computational cost during inference. Preliminary tests indicated that increasing passes beyond 10-20 yielded diminishing returns in the stability of the estimated standard deviation for this architecture and dataset, while significantly increasing prediction time. 10 passes provided a practical compromise.
- **Calculation:**
    - **Prediction (μ):** The final reported RMSF prediction for a residue is the **mean** of the predictions obtained across the N=10 stochastic forward passes.
    - **Uncertainty (σ):** The uncertainty estimate is the **standard deviation** of the predictions across the N=10 passes. A higher standard deviation indicates greater variance in the model's output when dropout is active, suggesting lower confidence or higher sensitivity to network perturbations for that specific residue's prediction.
- **Interpretation:** While correlated with prediction error, the MC Dropout standard deviation (σ) is not guaranteed to be a perfectly calibrated probabilistic uncertainty. It serves as a useful *qualitative* measure of model confidence, where higher σ suggests potentially less reliable predictions.