

**Probing Unfolding Forces of Proteins with Heterogeneous
Domains: Machine Learning Enables Force Spectroscopy**

**A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY**

Cailong Hua

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY**

Advisor: Murti Salapaka

October, 2025

© Cailong Hua 2025
ALL RIGHTS RESERVED

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Murti Salapaka, for his unwavering support and kind guidance throughout my Ph.D. journey. His mentorship has been instrumental not only in shaping my research but also in guiding my personal and professional development. He truly cares about his students and is deeply committed to helping them succeed in their future endeavors.

I am also grateful to Prof. James Ervasti for the unique opportunity to work on characterizing muscle proteins and for his generous support throughout this project. I would like to thank my thesis committee members—Prof. Andrew Lampeski, Prof. Nicola Elia, Prof. James Ervasti, and Prof. Ju Sun—for their encouragement and insightful suggestions.

Special thanks to Dr. Sivaraman Rajaganapathy, who guided me through my early years in the lab and made my transition smooth. I am glad we had the chance to collaborate on several projects together. I appreciate Dr. Rachit Shrivastava and Dr. Connor Edlund for their numerous discussions and the spirit of collaboration they brought to our work. I am thankful to Mohammed Tuhin Rana for the engaging conversations and the warmth of his friendship. My time in the lab was made immensely more enjoyable thanks to the camaraderie and support of my current and former labmates—Dr. Vivek Khatana, Dr. Arnab Dey, and Dr. Mishfad Veedu. I also thank Yiyuan Zhang and Vinitendra Singh—it was a pleasure working with you both on various projects.

I am especially grateful to Xinnong Li, who has been an indispensable partner and steadfast supporter in all aspects of life. Her patience and encouragement have been a constant source of strength throughout this journey.

Finally, I want to express my heartfelt appreciation to the most important people in my life—my parents. Their unconditional love, support, and encouragement laid the

foundation for this journey. I could not have come this far without them.

Dedication

To my family and friends.

Abstract

Protein molecules are the fundamental building blocks of life, and their ability to respond to mechanical forces through folding and unfolding is central to many biological processes as well as to debilitating conditions such as Duchenne muscular dystrophy (DMD), cardiomyopathies, and neurodegenerative disorders. Single-molecule force spectroscopy (SMFS) provides a powerful means of probing proteins with nanometer spatial and piconewton force resolution. However, extracting biologically meaningful insights at this scale remains challenging due to the inherent complexity, heterogeneity, and volume of SMFS data.

This dissertation combines rigorous SMFS experiments with deep learning methods to advance studying the mechanical properties of protein molecules. Experimentally, we present our findings on the muscle proteins dystrophin and utrophin using atomic force microscopy under constant speed and constant force operational modes. These results reveal distinct unfolding behaviors between dystrophin and utrophin, with important implications for the therapeutic potential of utrophin in DMD. Computationally, we introduce a physics-augmented deep learning model that incorporates the physics of protein unfolding for identifying trials originating from single molecule, which reduces reliance on manual inspection and accelerates SMFS analysis. We further develop a physics-aware deep clustering framework that enables domain-level inference, uncovering heterogeneous mechanical properties of protein folding domains in synthetic proteins as well as in natural proteins such as dystrophin and utrophin. Together, these experimental and computational advances establish a complementary framework for deepening our understanding of protein molecules while expanding the analytical toolkit available for single molecule studies.

Contents

Acknowledgements	i
Dedication	iii
Abstract	iv
List of Tables	ix
List of Figures	xi
1 Introduction	1
2 Single Molecule Force Spectroscopy of Muscle Proteins	4
2.1 Introduction	4
2.2 Multiple Modes of Atomic Force Microscope (AFM) Experiments	7
2.2.1 AFM Experiments	7
2.2.2 Constant Speed and Constant Force Experiments	8
2.2.3 Data Analysis	10
2.3 Evidence Supporting Single Molecule Measurements	11
2.3.1 Characterization of unfolding events per unfolding event count .	12
2.3.2 Consistent Unfolding Transition Rates across Multiple Modes of AFM	15
2.3.3 Concentration Optimization to Minimize Multiple-Protein Pulls	21
2.4 Mechanical Properties of Dystrophin and Utrophin	24

2.4.1	The Mechanical Properties of Dys Differ Significantly from Utr, with Dys Exhibiting Brittle Unfolding Behavior	24
2.4.2	DysN-R10 and Dp260 Exhibit Comparable Mechanical Properties to Dys, in Contrast to the Stiffening Spring Behavior of Utr . . .	27
2.4.3	The DHS Model Reveals Similarity among Dys Constructs and Distinction from Utr	28
2.5	Discussion	31
2.6	Supporting Information	33
2.6.1	Cloning	33
2.6.2	Protein Expression and Purification	34
2.6.3	Calibration Protein	35
2.6.4	Negligible Effect of Asparagine-tRNA Ligase	37
2.6.5	Histograms of Unfolding Forces and Lifetimes Across All Biological Replicates	37
2.6.6	Unfolded Length Increases Linearly with Event Count	38
3	Deep Learning Models for Classifying Single Molecule Force Spectroscopy Data	51
3.1	Introduction	51
3.2	Related Work	53
3.3	Problem Formulation	55
3.4	The Physics-augmented Model	56
3.4.1	Polymer Elastic Models	58
3.4.2	Fusion Module	59
3.5	Datasets	62
3.5.1	Experimental Dataset	64
3.5.2	Simulation dataset	64
3.6	Results	68
3.6.1	Evaluation Setup	68
3.6.2	PemNN Classification Performance	71
3.6.3	Functionality of Two Branches	72
3.6.4	Ablation Study of Model Architecture	75

3.6.5	Performance with Limited Data	76
3.6.6	Application to SMFS Data Analysis	77
3.7	Conclusions	80
3.8	Supporting Information	80
3.8.1	Data Preprocessing	80
3.8.2	Results of the Discoidin Domain Receptors (DDR) Dataset . . .	81
3.8.3	Performance under Varying Train/Test Splits	82
3.8.4	Physics-guided Pretraining Strategy	83
4	A Method for Identifying Heterogeneous Protein Folding Domains	90
4.1	Introduction	90
4.2	Literature Review	92
4.2.1	Identification of heterogeneous protein domains in SMFS data. .	92
4.2.2	Clustering methods for time series data.	92
4.3	Problem Statement	93
4.4	Methods	95
4.4.1	state-of-the-art methods	95
4.4.2	LatentUnfold, a novel physics-aware model	96
4.5	Datasets	98
4.5.1	Simulation data	98
4.5.2	Experimental data	100
4.6	Results	101
4.6.1	Evaluation	101
4.6.2	Performance on Simulated Data with Known Ground Truth . . .	102
4.6.3	Effective Identification of Heterogeneous Domains from Real Ex- perimental Data	108
4.7	Discussion and Conclusion	113
4.8	Supporting Information	113
4.8.1	Pre-processing of experimental data	113
4.8.2	Robustness across top-performing models	114
5	Conclusion and Discussion	118

List of Tables

2.1	The probability of successful pulls for the first and last unfolding event for all molecules (Dys, DysN-R10, Dp260, and Utr) from all constant speed data, where the number of unfolding events per each unfolding event count is summarized in Fig. 2.3	12
2.2	Decay rate for all molecules (Dys, DysN-R10, Dp260, and Utr) of both constant speed and constant force data, where the number of unfolding events per each unfolding event count is summarized in Fig. R6.	13
2.3	DHS model parameters of four molecules: Dys, DysN-R10, Dp260, and Utr.	29
2.4	The experimental most probable unfolding forces and lifetime averages of four molecules: Dys, DysN-R10, Dp260, and Utr.	50
2.5	Titin I27O modeling parameters	50
3.1	Details of experimental datasets	63
3.2	The DHS model parameters of four real protein molecules are: Titin I27O, bact UtrN-R3, insect UtrN-R3, DysN-R3	67
3.3	The most probable forces (in pN), along with the interquartile range (IQR) in parentheses, for all four protein molecules analyzed using different methods during AFM data analysis.	79
3.4	Overall accuracy, class accuracy, and F1-score of DDRs dataset for Triplet, MLP, FCN, ResNet, and InceptionTime are presented in the format of average (standard deviation) from 5 runs. Among these models, InceptionTime demonstrates superior performance compared to the others.	82

3.5	Model performance on experimental datasets with varying train/test split ratios. Values represent mean accuracy (%) with standard deviations over five runs in parentheses.	83
3.6	The most probable forces (in pN), along with the interquartile range (IQR) in parentheses, for both experimental and simulated data across all four protein.	85
3.7	ROC-AUC of ROC curves in Figure 3.14.	87
4.1	Simulation parameters for parameter sets 0 and 1.	102
4.2	Clustering accuracy and Silhouette score (mean with standard deviation in parentheses) for each model at 1000 nm/s. Results at 500 and 2000 nm/s are similar and therefore omitted for brevity. Best values at this speed are highlighted in bold.	103
4.3	Silhouette scores (mean and standard deviation across five runs and three pulling speeds) for the LatentUnfold(LSTM) model with different cluster numbers on four protein molecules. Additional results and detailed trends are shown in Figure 4.11.	108

List of Figures

2.1	Single Molecule Force Spectroscopy of Dystrophin and Utrophin. (A) Domain structure of Dys, DysN-R10, Dp260 and Utr. Circles, spectrin-like repeats (SLRs); diamonds, hinge domains; NT, N terminus; CT, C terminus; CR, cysteine-rich domain; ABD1 & 2, actin-binding domains; DgBD, dystroglycan binding domain. (B) Coomassie blue-stained SDS polyacrylamide gel with purified proteins. (C) Illustration of the SMFS experiment.	6
2.2	SMFS using two operational modes of AFM, constant speed and constant force. (A) A representative force-extension curve of molecule Dys obtained in constant speed mode. (B) Unfolding force histograms with fitted kernel density estimation from one biological replicate of molecule Dys at 1000 nm/s. (C) A representative curve of molecule Dys from constant force mode, with the extension-time curve in the top panel and the force-time curve in the bottom panel. (D) Lifetime histograms with fitted kernel density estimation from one biological replicate of molecule Dys at 70 pN. Unfolding events are highlighted with blue circles in both (A) and (C).	9

2.3	Number of unfolding events per unfolding event count for molecules (Dys, DysN-R10, Dp260, and Utr; from top to bottom), shown for constant speed data (left column) and constant force data (right column). The x-axis labels are formatted as $\{Ci : n = x\}$, indicating that ' $n = x$ ' unfolding events resulted in an unfolding event count of ' Ci ', $i = \{1, 2, 3, \dots\}$. The y-axis represents the probability, which generally decreases with increasing unfolding event count. A y-axis break at 0.02 is included to enhance visualization of lower probability values.	14
2.4	Schematic energy landscape along the reaction coordinate, where the intrinsic energy landscape U_0 is titled by $-Fx$ as the external force F increases, illustrated in blue.	15
2.5	Validation of consistency between constant-speed and constant-force AFM experiments using Monte Carlo simulations (Dys data shown). (A) Force-extension curve at constant speed. (B–C) Unfolding force distributions at various pulling speeds (500–5000 nm/s). (D) Transition rates extracted from constant-speed data. (E) Extension- and force-time curves under constant force. (F–G) Lifetime distributions across forces (50–90 pN). (H–I) Simulated unfolding force and lifetime statistics. (J–K) Comparison of $k_{\text{off}}(F)$ between experimental and simulated data across both AFM modes. (L–M) Energy landscapes fitted with cusp ($\nu = 1/2$) and linear-cubic ($\nu = 2/3$) models.	19
2.6	Number of unfolding events per unfolding event count for different concentrations, 1 nMol, 10 nMol, and 100 nMol (left to right) for both constant speed (top) and constant force (bottom) data. The x-axis labels are formatted as $\{Ci : n = x\}$, indicating that ' $n = x$ ' unfolding events resulted in an unfolding. Total pulls refer to the combined number from both constant speed and constant force. The probability of successful pulls is calculated as the number of unfolding events at $C1$ (unfolding event count = 1) from both constant speed and constant force, divided by the total number of pulls.	22

2.7	Detailed comparison of unfolding behavior at two concentrations: 10 nM (left) and 100 nM (right). (A) Unfolding force statistics across pulling speeds from 200 to 5000 nm/s, with the black stars indicating the most probable forces. (B) Lifetime statistics across multiple regulated forces ranging from 50 to 90 pN, with red triangles showing the average lifetimes. (C) Transition rates $k_{off}(F)$ for both experimental and simulated data using constant force and constant speed modes. Simulations were performed using the parameters listed in Table 1 of the main manuscript. Fitted energy landscapes with $\nu = 1/2$ and $\nu = 2/3$ are shown in (D) and (E), respectively.	23
2.8	Statistical summary of $N = 3$ biological replicates for (A) constant speed and (B) constant force experiments. Each column presents results for different molecules, namely Dys, DysN-R10, Dp260, and Utr from left to right. The data from three biological repeats are visualized in purple, pink, and yellow. Unfolding force distributions from constant speed experiments and lifetime distributions from constant force experiments are depicted using violin plots. The violin plots illustrate data distributions using kernel density estimation as black lines on each side, with the width of each curve indicating the relative frequency of data points. Meanwhile, black stars signify the positions of values with the highest probability, visualized by the widest section of the violin plot, and red triangles indicate the lifetime averages, calculated from identical forces and replicates.	24
2.9	Protein mechanical behavior as a function of unfolding event count (the number of domains unfolded). (A) Most probable forces from constant speed experiments. (B) Mean dwell lifetime, the time interval between consecutive unfolding events, derived from constant force experiments. Error bars represent 95% confidence intervals, calculated using bootstrapping.	26

2.10	Transition rates and energy landscapes of four molecules, Dys, DysN-R10, Dp260, and Utr. (A) k_{off} -vs- F curves showing mean values (lines) and standard deviations (error bars). (B) Cusp-like ($\nu = 1/2$) energy landscapes and (C) linear cubic ($\nu = 2/3$) energy landscapes, with mean values (lines) and standard deviations (shaded area). Mean and standard deviations are computed across different operational modes and biological replicates.	30
2.11	Summary of Titin I27O characterization. (A) Unfolding force statistics across pulling speeds from 500 to 5000 nm/s, with the black stars indicating the most probable forces. Typical curves are provided in SI Fig. 2.15A-C. (B) Lifetime statistics across multiple regulated forces ranging from 80 to 170 pN, with red triangles showing the average lifetimes. A zoomed-in view for regulated forces of 150 and 170 pN is shown on the right. Representative curves are displayed in SI Fig. 2.16A-C. (C) Unfolding force statistics from simulation data at pulling speeds of 500 to 5000 nm/s, with energy landscape parameters adopted from [1]. (D) Lifetime statistics from simulation data at regulated forces of 80 to 170 pN, with a zoomed-in view for forces of 150 and 170 pN. (E) Transition rates $k_{off}(F)$ for both experimental and simulated data using constant force and constant speed modes. Fitted energy landscapes with $\nu = 1/2$ and $\nu = 2/3$ are shown in (F) and (G), respectively. The corresponding energy landscape parameters are documented in Table 2.5.	36
2.12	Unfolding force data from constant speed AFM experiments. Rows represent pulling speeds of 200, 500, 1000, 2000, and 5000 nm/s from top to bottom; columns correspond to Dys, DysN-R10, Dp260, and Utr from left to right. In each panel, the histogram (blue bars) is shown together with Gaussian kernel density estimation (KDE) in solid red and two Gaussian components (dashed blue and orange) of the two-Gaussian mixture modal (GMM). The most probable force (Mode), full width at half-maximum (FWHM), and Kolmogorov–Smirnov (KS) distance difference between two fitting methods $\Delta KS = KS_{KDE} - KS_{GMM}$ are annotated.	39

2.13	Lifetime data from constant force AFM experiments. Rows represent regulated forces of 50, 60, 70, 80, and 90 pN from top to bottom; columns correspond to Dys, DysN-R10, Dp260, and Utr from left to right. In each panel, the histogram (blue) is shown together with Gaussian kernel density estimation (red); the mean lifetime (Mean) and full width at half-maximum (FWHM) are annotated.	40
2.14	Unfolded contour length as a function of unfolding event count for proteins (A) Dys, (B) DysN-R10, (C) Dp260, (D) Utr, and (E) pairwise comparison between Dys and Utr. Each boxplot represents the distribution of unfolded lengths at a given unfolding event count. The central line indicates the median; box edges denote the 25th and 75th percentiles, and whiskers represent the minimum and maximum values excluding outliers. These figures show a linear increase in unfolded contour length with unfolding event count. A weighted linear fit (in red) was applied to the median values, using 1/variance as the weighting factor and including the origin (0,0). The resulting slope (in nm/event) is annotated in the upper-left corner of each panel.	41
2.15	Representative curves from constant speed experiments. Each row presents results for different molecules, including Titin I27O, Dys, DysN-R10, Dp260, and Utr from top to bottom.	42
2.16	Representative curves from constant force experiment. Each row represents results namely Titin I27O, Dys, DysN-R10, Dp260, and Utr from top to bottom.	43
2.17	Example traces with a large number of unfolding events for constant speed experiments (left column; ≥ 10 unfolding events) and constant force experiments (right column; ≥ 7 unfolding events), shown for Dys (top row) and Utr (bottom row). In the constant speed data (A and C), Dys maintains relatively low unfolding forces (< 500 pN) across all events, whereas Utr shows increasing unfolding forces with higher unfolding event counts. Similarly, in constant force data (B and D), Utr exhibits longer total unfolding lifetimes compared to Dys.	44

- 2.18 A summary of statistics of (A) simulated constant speed and (B) simulated constant force. Simulations were conducted independently and distinguished by biological replicate to be consistent with experimental data. Each column presents results for different molecules, namely Dys, DysN-R10, Dp260, and Utr from left to right. The data from three biological repeats are visualized in purple, pink, and yellow. Unfolding force and lifetime distributions are depicted using violin plots. The violins illustrate data distributions using kernel density estimation as black lines on each side, with the width of each curve indicating the relative frequency of data points. Black stars signify the positions of unfolding forces with the highest probability, and red triangles indicates the lifetime average. 45
- 2.19 (A) Histograms of unfolding forces from constant speed data for unfolding event ranges 1–8 and 10–14. (B) Dwell time distributions from constant force data for unfolding event ranges 1–4 and 5–9. Data are shown for both Dys (left column) and Utr (right column). Here, the range x–y denotes data from unfolding events with counts satisfying $x \leq \text{unfolding event count} \leq y$. Dys exhibits similar distributions for both unfolding forces and dwell times across early and late unfolding event counts. In contrast, Utr shows marked differences, where later unfolding events (range 10–14 for constant speed, and range 5–9 for constant force) display flatter distributions with shifted peaks compared to earlier events. 46

2.20	Supplementary analysis corresponding to Fig. 2.9, in which constant speed data are filtered using a substantially stricter filtering with the physics-augmented model developed in Chapter 3, removing any traces that have any possibility of not being single molecule trial. We increased the classification threshold to maintain high precision, ensuring that only very high-confidence single-molecule curves were retained. Most probable forces from constant speed experiments. Error bars represent 95% confidence intervals, calculated using bootstrapping. It shows the same trend as Fig. 2.9A: the most probable unfolding force for Utr begins to rise at an unfolding event count of 7 and remains higher than that of Dys for counts greater than 7. This consistency under more rigorous data selection further reinforces the robustness of our conclusions.	47
2.21	A summary of results across four distinct scenarios: experimental constant speed, experimental constant force, simulated constant speed, and simulated constant force. Each row presents results for different molecules, namely Dys, DysN-R10, Dp260 and Utr from top to bottom. The graphs in the first column show k_{off} -vs- F curves, while the second and third columns show fitted energy landscapes with $\nu = 1/2$ and $\nu = 2/3$, respectively.	48
2.22	Statistics for all four molecules, Dys, DysN-R10, Dp260, and Utr, from both (A) constant speed, and (B) constant force data of all biological replicates. We conducted the two sample Kolmogorov-Smirnov (KS) test [2], which tests the null hypothesis that the distributions of two samples are identical, with the level of confidence $\alpha = 0.05$. The p-values over brackets correspond to comparisons of paired data.	49

3.1	Illustration of AFM based SMFS. (a) Schematic showing the desirable case of a single protein molecule with four folded domains under tension between the tip of the AFM cantilever and the substrate. The deflection d and the separation between the cantilever and the substrate z are measured. The tensile force on the protein molecule is computed from the deflection d . (b) Depictions of possible scenarios, categorized into three classes: (1) no molecule present between the tip and substrate, (2-3) a single molecule or a section of a single molecule present between the tip and the substrate, and (4-6) multiple molecules or sections of multiple molecules between the tip and the substrate. (c-e) Show example force curves representative of the three different classes, with blue circles highlighting unfolding events.	54
3.2	The overall architecture of the Polymer Elastic Models Neural Networks (PemNN) comprises a physics-based branch and a force trace branch, illustrated with the early fusion approach (the late fusion approach is shown in Figure 3.3).	57
3.3	Overall architecture of the PemNN model with late fusion strategy. The data augmentation block is not enabled by default but is utilized through the physics-guided pretraining strategy described in Section 3.8.4. . . .	62
3.4	Diagrams of dystrophin and utrophin. (a) Full-length dystrophin. (b) Full-length utrophin. (c) UtrN-R3 construct (d) DysN-R3 construct. The ovals are spectrin-like repeats (SLRs); the diamonds are hinge domains; NT is the N terminus; CT is the C terminus; CR is a cysteine-rich domain; ABD1 & 2 are actin-binding domains; DgBD is the dystroglycan binding domain. Figure courtesy of previous study [3]	64
3.5	Critical difference diagram of different deep learning models across the simulated and experimental testing sets of all protein molecules based on average accuracies. The most accurate model is assigned a rank of 1, with a thick horizontal line representing a group of classifiers that do not exhibit statistically significant differences in accuracy.	71

3.6	a) The average accuracy of all models across the four simulated datasets, with error bars indicating the standard deviations over five runs. b) The average accuracy across four simulated datasets over five runs as the contour length threshold varies, depicted in a radar chart where longer radii indicate higher accuracy. c) The average accuracy across four simulated datasets over five runs as the persistence length changes.	73
3.7	Contour length calculations using the WLC model with varying persistence lengths (L_p), visualized with a simulated force curve of Titin I270.	74
3.8	Ablation study of PemNN architecture. a) Critical diagram comparing various fusion strategies, where 'E-' and 'L-' denote early fusion and late fusion strategies, respectively. b) Critical diagram evaluating the incorporation of LSTM layers, with 'F1P1' indicating LSTM layers included in both the force trace and physics-based branches. c) Critical diagram of different polymer elastic models. d) Performance of PemNN across datasets with three distinct polymer elastic models.	75
3.9	The performance of models trained with different proportions of experimental datasets (training proportion), with error bars indicating standard deviations across all experimental datasets over five runs.	76
3.10	Application to SMFS data analysis of Titin I270: <i>RawData</i> as the Baseline Method, <i>Heuristic</i> as non-machine learning method, and <i>PemNN</i> as the our proposed model.	78
3.11	Application of PemNN to AFM Data Analysis: <i>RawData</i> as the Baseline Method, <i>Heuristic</i> as non machine learning method, and <i>PemNN</i> as the our proposed model. Unfolding force distributions are depicted using violin plots, where black stars signify the positions of values with the highest probability, with values detailed in Table 3.3. The violins illustrate data distributions using kernel density estimation as black lines on each side, with the width of each curve indicating the relative frequency of data points.	79

3.12	Data preprocessing example. (a) The original force curve. (b) The force curve after trimming. (c) The force data after normalization and resampling in the force trace branch. (d) The force curve with calculated L_c in the physics-based branch.	81
3.13	Pretraining strategy results with PemNN. a)Critical difference diagram comparing different data augmentation methods. b)The physics-guided pretraining strategy (train size = 0%) slightly underperforms compared to direct training with experimental data (train sizes = 5% to 20%). . .	84
3.14	ROC curves from a single run for four datasets, (a) Titin I27O, (b) Insect UtrN-R3, (c) Bact UtrN-R3 and (d) DysN-R3 using PemNN. ROC curves are plotted using One-vs-Rest strategy, with no molecule, single molecule and multiple molecules classes depicted in yellow, red and blue, respectively. Solid lines represent results trained with simulation data, while dashed lines indicate results trained with experimental data. These ROC curves are generated using 80% of experimental data, with the corresponding ROC-AUC in Table 3.7.	86
3.15	Performance of PemNN with varying probability thresholds for all four experimental datasets. The x-axis represents values of α , with 'w/o Threshold' indicating no threshold is used.	88
4.1	Illustration of AFM-based SMFS. (a) Schematic of the experimental configuration with the protein molecule composed of three folded domains. The deflection d and the cantilever–substrate distance z are measured during pulling. (b–d) Illustration of sequential unfolding of three domains, including two type 0 domains (blue) and one type 1 domain (red). (e) Representative force-extension curve with the unfolding segment indicated by a dashed oval. (f) Matching unfolding curve segments to domain types 0 and 1, enabling identification of heterogeneous domains.	93
4.2	Model architecture of LatentUnfold, a novel physics-aware deep clustering framework.	96

4.3	Diagrams of protein molecules. (a) ddFLN4–Titin I27, consisting of one ddFLN4 domain and one Titin I27 domain. (b) DysN-R3, a fragment of dystrophin spanning the N-terminus (NT) through spectrin-like repeat 3 (SLR3). (c) UtrN-R3, a corresponding utrophin fragment, further divided into bacterial UtrN-R3 and insect UtrN-R3 based on the expression system. Ovals represent spectrin-like repeats (SLRs), and NT indicates the N-terminus domain.	99
4.4	Predicted versus ground truth results on simulated data using LatentUnfold(LSTM). (a) Predicted unfolding force distributions (dashed lines) closely match the ground truth distributions (solid lines) for the dataset at 1000 nm/s. (b) Distributions of unfolding force (top) and contour length increment (bottom), with medians shown as center points and error bars indicating standard deviations. (c) Example force-extension curves with two distinct unfolding segments identified.	104
4.5	Robustness study under systematic parameter variation. Each parameter is modified by a specified fold relative to values of ‘P0’ (Table 1 in the main paper, and annotated accordingly (e.g., +99% k_0 indicates a 99-fold increase in k_0). Ground truth distributions are shown as solid lines, and predictions as dashed lines.	105
4.6	Critical difference (CD) diagrams comparing all models across the 51 simulated datasets using (a) clustering accuracy and (b) Silhouette score. The top-performing model receives a rank of 1. Models connected by a thick horizontal line form a group with no statistically significant difference in performance. Statistical significance is assessed following [4, 5].	106
4.7	Critical difference (CD) diagram for the ablation study with (a) clustering accuracy and (b) Silhouette score. Both LatentUnfold(CNN-LSTM) and Ours (LSTM) consistently outperform the single-autoencoder variants: the force-only autoencoder (FAE), the CNN-LSTM relationship autoencoder (RAE-CNN-LSTM), and the LSTM only relationship autoencoder (RAE-LSTM).	107

4.8	Clustering results for a synthesized protein with well-characterized domains. Clustering statistics for ddFLN4 (cluster 0) and Titin I27 (cluster 1) at pulling speeds of (a) 500 nm/s, (b) 1000 nm/s, and (c) 2000 nm/s. The first column shows representative force-extension curves with color-coded segments by predicted cluster labels. The second column displays the unfolding force distributions, and the third column presents the distributions of contour length increase (ΔL_c). Cluster 0 is shown in blue, Cluster 1 in orange, and the combined distribution is shown in red. . . .	109
4.9	Results of natural protein molecules. Clustering statistics at 1000 nm/s for (a) DysNR3, (b) Bact UtrNR3, and (c) Insect UtrNR3.	112
4.10	Clustering statistics of ddFLN4–Titin I27 across three pulling speeds: 500, 1000, and 2000 nm/s (from left to right). Dots indicate the most probable values, with error bars representing standard deviations. The first row shows the unfolding force distributions; the second row shows the contour length increase distributions; and the third row presents mismatch ratios, computed by treating the predicted labels from LatentUnfold(CNN-LSTM) as the reference and comparing them against those from the other algorithms. Mismatch ratios are shown in black.	115
4.11	Silhouette score vs number of clusters for all four protein molecules: ddFLN4–Titin I27, bact UtrN-R3, insect UtrN-R3, DysN-R3.	116
4.12	Clustering statistics at 500 nm/s for (a) DysNR3, (b) Bact UtrNR3, and (c) Insect UtrNR3, and at 2000 nm/s for (d) DysNR3, (e) Bact UtrNR3, and (f) Insect UtrNR3. The first column shows representative force-extension curves with color-coded segments by predicted cluster labels. The second column displays the unfolding force distributions, and the third column presents the distributions of contour length increase (ΔL_c). Cluster 0 is shown in blue, Cluster 1 in orange, and the combined distribution is shown in red.	117

Chapter 1

Introduction

Protein molecules, the functional building blocks of life, are composed of more than 20 different types of amino acids. Interactions among these amino acids drive the folding of the protein into an intricate three-dimensional shape, which determines its function and behavior. They play essential roles in many biological processes via regulating mechanical forces through the folding and unfolding [6, 7]. Understanding how protein molecules respond to mechanical forces offers valuable insight into their functional roles and development of treatments for a range of debilitating and fatal conditions [3, 8, 9, 10]. For example, mutations in the titin gene can impair its ability to respond to mechanical tension, resulting in dilated cardiomyopathy [11, 12]. Similarly, characterizing the mechanical properties of shock-absorbing muscle proteins such as dystrophin and utrophin may contribute to effective therapies for Duchenne Muscular Dystrophy (DMD), a fatal genetic disorder affecting approximately 1 in 5,000 male births [13, 14]. Additionally, alterations in the mechanical behavior and aggregation of misfolded proteins such as α -synuclein are increasingly recognized as key factors in the pathogenesis of Parkinson's and Alzheimer's diseases [15, 16]. Therefore, elucidating how protein molecules respond to mechanical forces is critical for understanding their functional roles and for interpreting disease-associated mutations.

Single-molecule force spectroscopy (SMFS) is a well-established method that directly probes structural changes of macromolecules under the influence of mechanical force. The development of instruments such as atomic force microscopes (AFM) [17] and optical tweezers [18] have enabled SMFS with the resolution of measuring forces in the

femto to multiple pico Newton range with spatial resolution at the nanometer scale. In an SMFS experiment, a force probe interacts with target molecules, and the resulting force-extension data are recorded. A typical workflow includes protein purification (from bacterial or insect expression systems), sample preparation, and automated AFM measurements that yield thousands of force-extension curves per session. Post-experiment, expert users manually filter curves to identify those that correspond to interactions with single molecules. To ensure reproducibility, proteins are typically expressed in at least three independent batches, each with multiple measurement sessions.

This dissertation employs both experimental and computational methods to advance our understanding of protein molecules. The outline of the thesis is as follows:

Chapter 2: Duchenne muscular dystrophy (DMD) is a lethal muscle disease that affects 1 in 5000 boys born in the United States [13] characterized by progressive muscle degeneration and weakness due to loss of the protein dystrophin [19]. Utrophin, a natural homolog of dystrophin, has been proposed as a therapeutic substitute due to its similar structural and functional properties [20]. However, the mechanical properties of full-length dystrophin have not been thoroughly characterized, and comparisons with utrophin remain limited. In this chapter, we report the first mechanical characterization of single full-length dystrophin (Dys) molecules using two operational modes of atomic force microscopy (AFM); constant speed and constant force as well as Monte Carlo simulations. Furthermore, we have compared Dys with large fragments encoding the N-terminus through spectrin repeat 10 (DysN-R10), the C-terminal retinal isoform of dystrophin (Dp260), and full-length utrophin (Utr).

Chapter 3: Since the size of a typical force probe is orders greater than a typical bio-molecule, the probe may come in contact with one or more molecules. Measurements obtained from interactions involving more than one molecule confound the interpretation of results and are a significant challenge in characterizing the behavior of single molecules. How to identify and isolate measurements and data that result from a single-molecule is an important objective for SMFS. Currently, manual visual inspection is the primary method for classifying single-molecule data; a process that is both time-consuming and requires significant expertise that take 12 to 24 hours per session. Each session of SMFS experiments typically involves approximately 2,000–5,000 trials.

ere, we both apply state-of-the-art machine learning models and present a novel physics-augmented deep learning model tailored to SMFS data. The proposed model employs a dual-branch fusion strategy; one branch integrates the physics of protein molecules, and the other operates independently of physical constraints. Our physics-augmented model reduces the time required to extract statistically meaningful single-molecule data from a full session to under one hour.

Chapter 4: Natural proteins are typically composed of heterogeneous folding domains with distinct mechanical properties; deciphering domain-level behavior provides insights into disease mechanisms [21, 22]. For example, in dystrophin (Figure 4.3, the N-terminal domain anchors the protein to actin filaments beneath the sarcolemma, while the spectrin-like triple-helix repeats are hypothesized to unfold in response to mechanical stress. In this work, we present the first automated framework to identify heterogeneous folding domains in SMFS data, including both existing clustering models and a novel physics-aware deep clustering architecture, LatentUnfold. LatentUnfold learns latent representations from two perspectives, force magnitude and protein unfolding physical relationship, followed by joint optimization for clustering assignments. This work enables domain-level biological inference, overcoming prior limitations that relied on averaging and overlooked heterogeneity, thus advancing the understanding mechanical behavior of protein unfolding.

Chapter 5: We conclude with a summary of the key contributions of this thesis and discuss potential future directions to further expand the scope and impact of this work.

Chapter 2

Single Molecule Force Spectroscopy of Muscle Proteins

Duchenne muscular dystrophy (DMD) is a lethal muscle wasting disorder caused by loss of the protein dystrophin. Utrophin is a fetal homologue of dystrophin that is under active investigation as a dystrophin replacement therapy for DMD. However, it is unknown if utrophin can substitute for dystrophin from a mechanical perspective. In this chapter, we directly compared the mechanical properties of full-length dystrophin with utrophin using atomic force microscopy (AFM) through two operational modes; constant speed and constant force. Our data reveal distinct mechanical behaviors for dystrophin and utrophin that suggest their functions are different *in vivo*.

2.1 Introduction

Dystrophin is a 427 kDa protein expressed primarily at the muscle cell membrane, or sarcolemma, in striated muscle tissue and is a crucial component of the dystrophin-glycoprotein complex (DGC) [23]. It is hypothesized that dystrophin acts as a molecular shock absorber to stabilize the sarcolemma and protect muscle against mechanical forces during muscle stretching and contraction [24, 25, 26, 27, 28, 14, 29]. Dystrophin deficiency increases sarcolemmal fragility ultimately leading to myofiber death [30].

The functional domain architecture of dystrophin provides insights into how its

absence leads to disease [14]. At the N-terminus, an actin-binding domain anchors dystrophin to actin filaments beneath the sarcolemma (Fig. 2.1A). This is followed by a central rod domain composed of 24 spectrin-like triple-helix repeats interspersed with four flexible hinge regions, forming a mechanical linkage between the actin-binding domain and the dystrophin-glycoprotein complex (DGC)-binding domains. DGC-binding is mediated by globular domains located C-terminal to the spectrin-like repeats. It is hypothesized that the central spectrin-like repeats unfold in response to mechanical stress, allowing dystrophin to absorb mechanical energy and protect the sarcolemma from damage. However, the mechanical properties of full-length dystrophin have yet to be fully characterized, and it remains unclear whether they support its proposed role as a molecular shock absorber.

Upregulation and subsequent overexpression of a naturally homologous protein, utrophin, is proposed as a potential DMD therapy with the goal of substituting utrophin for dystrophin at the sarcolemma [20]. Utrophin (Fig. 2.1A) is abundantly expressed during fetal development, and like dystrophin, forms a complex with actin filaments and the DGC at the sarcolemma but is later replaced by dystrophin after birth [31]. In adult skeletal muscle, utrophin primarily localizes to the myotendinous and neuromuscular junctions [32]. Utrophin overexpression is shown to compensate for dystrophin deficiency in *mdx* mice [33] and could do so in humans as well. How utrophin’s mechanical properties compare with dystrophin is not known.

In this study, we characterized the mechanical properties of full-length dystrophin (Dys) using atomic force microscopy (AFM), and compared it with full-length utrophin (Utr) and large dystrophin fragments encoding the N-terminus through repeat 10 (DysN-R10) and Dp260 encoding ~R11-C-terminus (Fig. 2.1A). We performed single molecule force spectroscopy (SMFS) using both constant speed and constant force modes of AFM. The statistics of force magnitude data from constant speed mode and the time to unravel statistics of the constant force mode were independently consistent. We analyzed these data globally using the Dudko Hummer Szabo (DHS) model [34], which serves as a cornerstone for relating statistics from the two operational modes and extracting kinetic information to model domain unfolding via Monte Carlo simulations. Such consistency ensures data reliability [35, 36]. Our data demonstrate that Dys, DysN-R10 and Dp260, all exhibit brittle unfolding behavior while the mechanical unfolding of Utr

is distinct and more complex, marked by a stiffening spring behavior. Our direct *in vitro* comparison of Dys and Utr suggest different *in vivo* functions. Notably, although the early unfolding behaviors of Dys and Utr are similar, significant differences arise during the later unfolding events. We hypothesize these differences will impact their interchangeability as therapeutics.

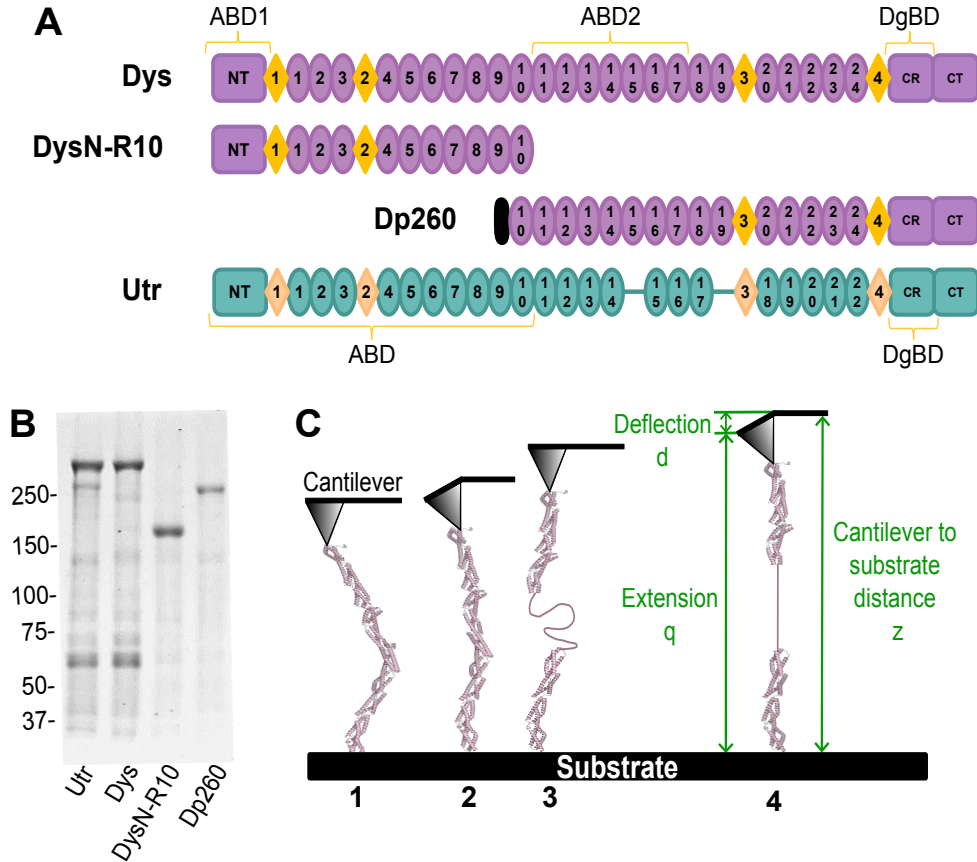


Figure 2.1: Single Molecule Force Spectroscopy of Dystrophin and Utrophin. (A) Domain structure of Dys, DysN-R10, Dp260 and Utr. Circles, spectrin-like repeats (SLRs); diamonds, hinge domains; NT, N terminus; CT, C terminus; CR, cysteine-rich domain; ABD1 & 2, actin-binding domains; DgBD, dystroglycan binding domain. (B) Coomassie blue-stained SDS polyacrylamide gel with purified proteins. (C) Illustration of the SMFS experiment.

2.2 Multiple Modes of Atomic Force Microscope (AFM) Experiments

2.2.1 AFM Experiments

Single molecule force experiments were conducted using MFP-3D atomic force microscope (AFM) (Asylum Research, an Oxford Instruments Company, Santa Barbara, CA). The AFM configuration consists of a flexible cantilever with a sharp tip, a laser-photodiode based sensor for tracking the cantilever tip's position, and a piezoelectric nano-positioner capable of moving the substrate in three dimensions relative to the cantilever base [17]. We employed Bruker MLCT-BIO triangular cantilevers, made of silicon nitride and coated with reflective gold on the backside, with a nominal spring constant of 10 pN/nm and a typical tip radius of 20 nm. Prior to each experiment, the spring constant was determined by analyzing the thermally excited deflection of the cantilever [37]. A droplet ($\sim 100 \mu\text{l}$) of purified protein suspended in PBS, with a dilution ranging from 3 to 30 nanomolar concentration (nMol), was placed on a freshly cleaved and ionized mica substrate. The protein concentrations were chosen to ensure a 5-10% probability of successful pulls, a success percentage used in previous studies to minimize the probability of pulling multiple proteins [10, 38]. This setup was incubated for 15 minutes to allow the protein to adhere to the mica surface. The droplet containing the protein solution was then removed and the mica substrate was washed once with 100 μl of PBS to eliminate proteins that had not adhered. A new droplet containing 100 μl of PBS was introduced to establish the environment for the force spectroscopy experiments. Repeated approach-retraction cycles were conducted at room temperature, where the cantilever's tip was pressed against the substrate for a duration of 2 seconds, applying an indentation force of 600 pN. Subsequently, in the constant speed mode, the cantilever base was retracted at a speed ranging from 200 to 5000 nm/s. The pulling force on the molecule is balanced by the force experienced by the cantilever, thus the force, F on the protein was determined as,

$$F = k_c d, \quad (2.1)$$

where k_c is the spring constant of the cantilever and d is the deflection of the cantilever. In constant force mode, the cantilever base was regulated to maintain a steady deflection,

resulting in a regulated force on the molecule, ranging from 50 to 90 pN. Data from 300 – 2000 successful force spectroscopy experiments (with at least one identifiable unfolding event) were collected for each protein under a specific speed or force condition and used to determine the unfolding statistics.

2.2.2 Constant Speed and Constant Force Experiments

The proteins analyzed in this study are depicted in Fig. 2.1A and their quality assessed on Coomassie blue-stained SDS polyacrylamide gels (Fig. 2.1B). Dys and Utr preparations each contained 10% of a ~ 60 kDa band (Fig. 2.1B), which was identified by mass spectrometry to be asparagine-tRNA ligase derived from the insect cell host. We do not believe that the presence of this minor contaminant impacted subsequent analyses by AFM because Dys and Utr presented with significantly different unfolding profiles (see Fig. 2.9) even though both were contaminated to similar extents while the unfolding profiles for DysN-R10 and Dp260 matched closely that of Dys (Fig. 2.9) even though preparations of both lacked the contaminating protein (Fig. 2.1B). We also assessed the potential contribution of asparagine-tRNA ligase to our data and found it represents less than 5% of traces, confirming that it does not meaningfully affect our results (see Section 2.6.4 for details). AFM operational modes of constant speed and constant force were utilized for SMFS experiments. One portion of the protein randomly attaches to the substrate and the remainder is left free to attach to the force probe, as illustrated by state 1 in Fig. 2.1C. The cantilever to substrate distance, z , is retracted at a constant rate in constant speed mode or is controlled to maintain the cantilever deflection d constant, thus ensuring a constant force on protein, in constant force mode. Under such applied mechanical tension, the protein is stretched, transitioning into state 2 in Fig. 2.1C. Subsequently, one of the folded protein domains unfolds stochastically, leading to state 3, and the cantilever continues to stretch the protein to state 4. This process is repeated until all domains are unfolded or the protein link between the cantilever and the substrate is broken.

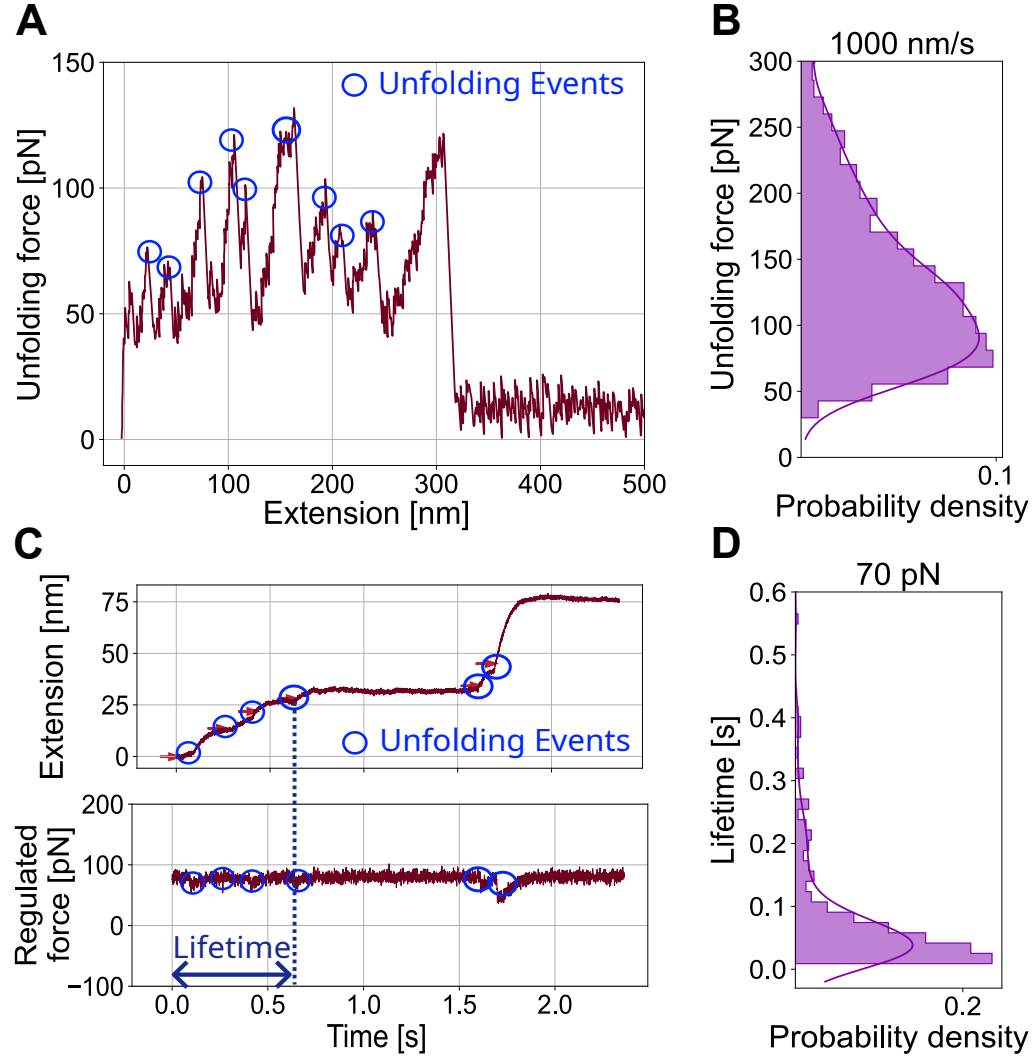


Figure 2.2: SMFS using two operational modes of AFM, constant speed and constant force. (A) A representative force-extension curve of molecule Dys obtained in constant speed mode. (B) Unfolding force histograms with fitted kernel density estimation from one biological replicate of molecule Dys at 1000 nm/s. (C) A representative curve of molecule Dys from constant force mode, with the extension-time curve in the top panel and the force-time curve in the bottom panel. (D) Lifetime histograms with fitted kernel density estimation from one biological replicate of molecule Dys at 70 pN. Unfolding events are highlighted with blue circles in both (A) and (C).

In constant speed mode, the applied force drops abruptly when a domain unfolds, resulting in a saw-tooth pattern in the force versus extension curves. Fig. 2.2A illustrates a representative force-extension curve for a Dys molecule with unfolding events highlighted in blue circles. Unfolding forces were recorded as a function of unfolding events, leading to statistics of force magnitude data as shown in Fig. 2.2B, with lines representing the kernel density estimation of the corresponding histograms. Here, a Gaussian kernel was employed to estimate the underlying probability distribution of the data collected at 1000 nm/s of molecule Dys.

In constant force mode, unfolding events were identified by discrete steps in the molecular extension accompanied by corresponding spikes in force, as illustrated with a Dys molecule in Fig. 2.2C. These unfolding events are marked with blue circles and a staircase-like pattern can be observed in the extension-time curve. The parameter of interest is the lifetime of a bound state which is the time for an unfolding event to occur from the moment the protein is subject to tension. The lifetime histograms together with kernel density estimation is presented in Fig. 2.2D, where the data was collected at 70 pN of molecule Dys.

2.2.3 Data Analysis

The raw data obtained from the force spectroscopy experiments are primarily composed of three main variables: 1) the time for a bond to unravel and how the forces vary with time, 2) the deflection of the cantilever, and 3) the distance between the base of the cantilever and the substrate. It includes instances where there is no protein between the tip and substrate, as well as instances where a protein was effectively adsorbed and subsequently stretched. To identify successful experiments and extract unfolding events from data, we utilize a custom algorithm developed in MATLAB R2023b (MathWorks) [10, 3]. The algorithm calculates the applied force by multiplying the cantilever deflection with the spring constant and computes the protein extension by subtracting the deflection of the cantilever from the distance between the cantilever base and the substrate. The algorithm first computes the standard deviation σ of noise in the force signal measured from the approach curve. Subsequently, in constant speed mode, it detects characteristic peaks in the force signal that exceed 3σ . In constant force mode, it identifies steps larger than 10σ in the extension signal coinciding with a spike in the

force signal. Here, the significant events are the peaks in constant speed mode and steps in constant force mode. The first significant event arises from the adhesive force between the cantilever tip and the substrate; this event is excluded from the dataset for subsequent analysis. Similarly, the last significant event, often affected by a detachment event, is also discarded. The remaining significant events are recognized as domain unfolding events, which are of interest. The number of significant events is then used to distinguish between successful and unsuccessful experiments.

The force-extension curves obtained from the constant speed mode are fitted to a worm-like chain (WLC) model. The WLC model [39] relates the force applied on protein F_{WLC} to its extension q , and is given by,

$$F_{WLC}(q, L_c, L_p) = \frac{k_B T}{L_p} \left[\frac{1}{4(1 - \frac{q}{L_c})^2} - \frac{1}{4} + \frac{q}{L_c} \right], \quad (2.2)$$

where k_B is the Boltzmann constant, T is temperature, and L_c and L_p are the contour length and the persistence length of the protein, respectively. For the constant force mode, extension-time curves are fitted to a staircase function. These curves are then inspected manually. Curves that show irregularities, such as an excessive number of unfolding events or distorted significant events that deviate from the characteristics of the fitting curve, are excluded for further analysis.

To analyze mechanical properties across different unfolding event counts, we first categorized data from all biological replicates, retaining only categories with sample sizes greater than 10 to minimize bias. After removing outliers (greater than three standard deviations from the mean), we performed bootstrapping by resampling with replacement to generate estimate distributions. This approach allowed us to determine confidence intervals using percentile methods for both: (1) the most probable unfolding forces from constant speed measurements and (2) the mean dwell lifetimes from constant force experiments.

2.3 Evidence Supporting Single Molecule Measurements

To minimize the probability of pulling multiple protein molecules, we optimized the concentration of molecules deposited (Fig. 2.6). We set the working concentration

to ensure that the probability of successful pulls remained in the 5–10% range (Table 2.1, Fig. 2.3), which lies within the recommended 5–25% window established previously [10, 3]. As evidence that single molecules are being pulled, we observed that unfolded lengths increased linearly with event count, with slopes consistent with single spectrin-like repeats (Fig. 2.14). Furthermore, when high concentrations of proteins are employed (100 nMol), transition rates $k_{off}(F)$ estimated based on constant speed and constant force data do not remain consistent (see Fig. 2.7), indicating multiple proteins are unfolded at higher concentrations. At concentrations used for the results and conclusions reached, the data from constant speed and constant force experiments exhibit consistency in transition rates $k_{off}(F)$ when analyzed using the DHS model, as illustrated in Fig. 2.5, indicating that data primarily results from single molecule trials. Transition rates were estimated from the unfolding force distributions in the constant speed mode using Eq. 2.5, and from lifetime averages in the constant force mode by Eq. 2.10. In the following analysis, we focus on mean lifetimes derived from the constant force data and the most probable forces from the constant speed data, as the latter better represents the distribution than the mean.

Table 2.1: The probability of successful pulls for the first and last unfolding event for all molecules (Dys, DysN-R10, Dp260, and Utr) from all constant speed data, where the number of unfolding events per each unfolding event count is summarized in Fig. 2.3

Molecule	Total force spectroscopy pulls	Probability for the first unfolding event	Probability for the last unfolding event Ci
Dys	35762	9.16%	$C13$: 0.045%
DysN-R10	27376	7.54%	$C8$: 0.047%
Dp260	26205	5.28%	$C11$: 0.053%
Utr	37660	6.86%	$C14$: 0.032%

2.3.1 Characterization of unfolding events per unfolding event count

To minimize the probability of pulling multiple protein molecules, the concentration is generally lowered to the point where the probability of successful pulls is in the 5-25%

range, as established in previous studies [10, 3]. In the present study, we maintained this probability in the 5–10% range, derived from all constant speed data, as shown in Table 2.1. The protein concentrations used in constant force experiments were identical to those in constant speed and were drawn from the same biological replicates, thus the data collection rate in constant force experiments is comparable to that observed in constant speed.

These data also reflect that the successful probability decreases markedly with increasing unfolding event count (e.g., Dys: 9.16% at the first event vs. 0.045% at C13). To quantitatively capture this trend, we fit the number of unfolding events versus unfolding event count using an exponential model $y = ae^{-bx}$, where b is the exponential decay rate and a is the scaling factor. The average fitted decay rate across all constructs from both constant-speed and constant-force data is $b = 0.54$, with detailed decay rates provided in Table 2.2. This corresponds to a decay factor of $e^{-0.54} \approx 0.58$, meaning the number of unfolding events decreases by approximately 42% per additional unfolding event count. Practically, this means that, based on the average total pulls across four molecules in Table R1, we would need to perform approximately 51,000 additional experiments to observe just one more unfolding event. Thus, observing traces with a larger number of unfolding events than 14 is quite improbable and impractical.

Table 2.2: Decay rate for all molecules (Dys, DysN-R10, Dp260, and Utr) of both constant speed and constant force data, where the number of unfolding events per each unfolding event count is summarized in Fig. R6.

Molecule	Dys	DysN-R10	Dp260	Utr
CS decay rate	0.41	0.48	0.37	0.37
CF decay rate	0.52	0.71	0.74	0.70

Furthermore, the distribution of unfolding events per unfolding event count remains consistent across molecules (Fig. 2.3), which further supports that our data are not biased by nonspecific desorption artifacts. If nonspecific desorption events were a variable contributor, we would expect deviations from this trend between molecules. Therefore, the distinct trends observed between Dys and Utr are more likely due to intrinsic

differences in mechanical unfolding behavior, rather than desorption artifacts.

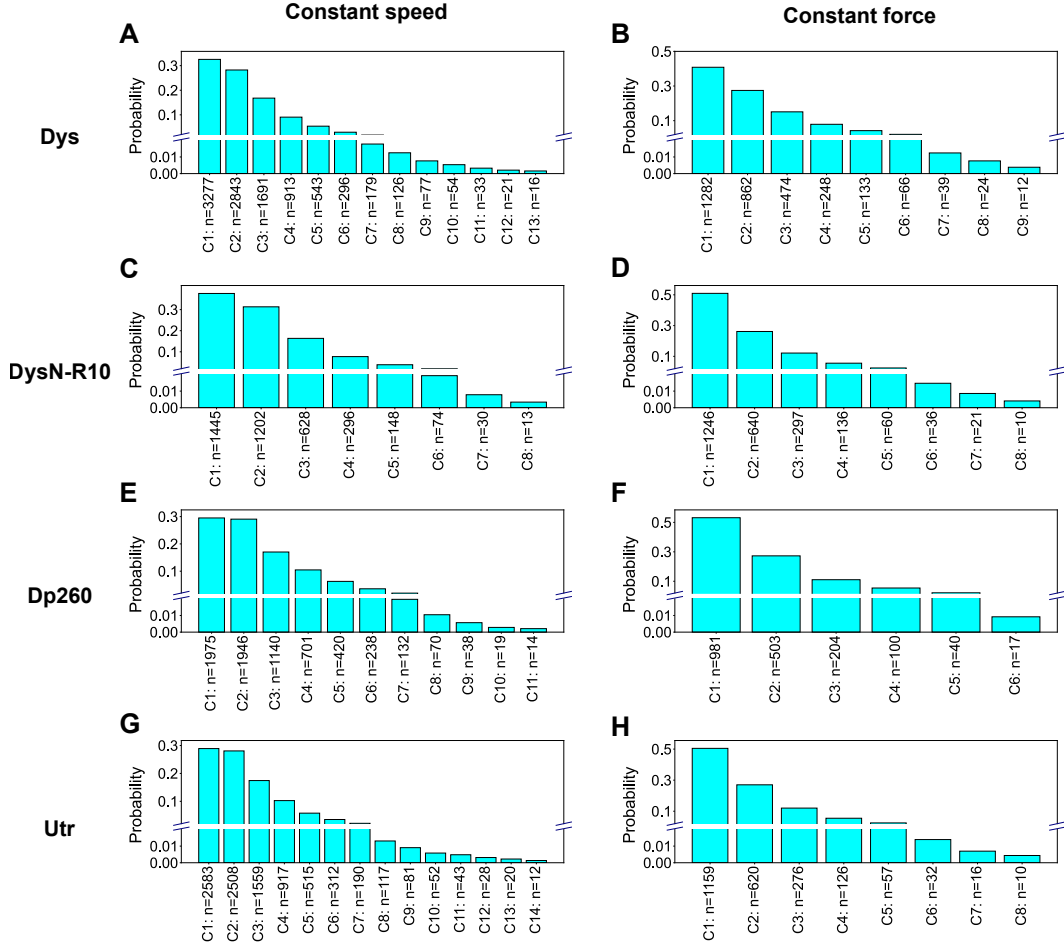


Figure 2.3: Number of unfolding events per unfolding event count for molecules (Dys, DysN-R10, Dp260, and Utr; from top to bottom), shown for constant speed data (left column) and constant force data (right column). The x-axis labels are formatted as $\{C_i : n = x\}$, indicating that ' $n = x$ ' unfolding events resulted in an unfolding event count of ' C_i ', $i = \{1, 2, 3, \dots\}$. The y-axis represents the probability, which generally decreases with increasing unfolding event count. A y-axis break at 0.02 is included to enhance visualization of lower probability values.

2.3.2 Consistent Unfolding Transition Rates across Multiple Modes of AFM

Two State System

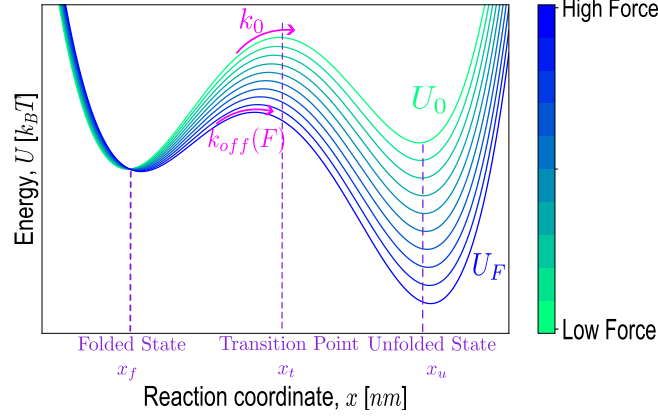


Figure 2.4: Schematic energy landscape along the reaction coordinate, where the intrinsic energy landscape U_0 is tilted by $-Fx$ as the external force F increases, illustrated in blue.

In many cases, the unfolding process of a single folded domain of protein can be modeled with a one-dimensional projected energy landscape along the reaction coordinate with two stable states being the local minimas in the projected energy landscape. The intrinsic free energy landscape $U_0(x)$ along reaction coordinate x (see Fig. 2.4), is assumed to have minimas at folded state x_f and unfolded state x_u with a barrier of height $\Delta G^\ddagger$ at the transition point x_t , where $\Delta G^\ddagger := U_0(x_t) - U_0(x_f)$ is the energy difference between energy levels at x_t and x_f . The distance between the folded state and the transition point is defined as $\Delta x^\ddagger := x_t - x_f$. Additionally, the rate of transition from the folded to the unfolded state under intrinsic energy landscape k_0 is exponentially related to the energy barrier, i.e., $k_0 \propto e^{-\frac{\Delta G^\ddagger}{k_B T}}$ [40], where k_B is the Boltzmann constant and T is temperature.

An external force F results in a tilted energy landscape as shown in Fig. 2.4. The modified energy landscape is given by $U_F(x) = U_0(x) - Fx$ with a smaller barrier height. Additionally, the associated transition rate is denoted as $k_{off}(F)$ and is related to the

force-dependent lifetime $\tau(F)$, which is the time it takes to unfold under such force, through $k_{off}(F) := 1/\langle\tau(F)\rangle$; where $\langle\cdot\rangle$ denotes the mean.

The Dudko-Hummer Szabo Model

In the context of two-state process, the transition rate $k_{off}(F)$ can be obtained from the probability distribution $p(F)$ of unfolding forces measured from constant speed mode as [1, 34],

$$k_{off}(F) = \frac{p(F)r(F)}{\int_F^\infty p(F')dF'}, \quad (2.3)$$

where $r(F) := \frac{dF}{dt}$ is the loading rate. This equation illustrates how the distributions of unfolding forces, measured under constant speed mode, can be converted into lifetime statistics measured in constant force mode, and is applicable regardless of the specific shape of the energy landscape.

The histogram of unfolding forces is constructed with equal bin width ΔF ; starting from F_0 and ending at $F_N = F_0 + N\Delta F$. Let the height of k th bin be,

$$h_k = \frac{C_k}{C_{tot}\Delta F}, \quad (2.4)$$

where C_k is the number of counts in k th bin and $C_{tot} = \sum_i C_i$ is the total number of counts in the histogram. The transition rate k_{off} corresponding to the k th bin is found by [34],

$$k_{off}\left(F_0 + (k - \frac{1}{2})\Delta F\right) = \frac{h_k r\left(F_0 + (k - \frac{1}{2})\Delta F\right)}{\left(\frac{h_k}{2} + \sum_{i=k+1}^N h_i\right)\Delta F}, \quad (2.5)$$

where $r\left(F_0 + (k - \frac{1}{2})\Delta F\right)$ is the average loading rate of k th bin.

The Dudko-Hummer-Szabo (DHS) model [34] considers two different shapes of energy landscape to derive analytical expressions for $k_{off}(F)$ using Kramers theory [40]. The two energy landscapes are: 1) the cusp-like energy landscape which is described by $U_0^{cl}(x) = \Delta G^\ddagger (x/\Delta x^\ddagger)^2$ for $x < x_t$ and $U_0^{cl}(x) = -\infty$ otherwise [41], and 2) the linear-cubic energy landscape which is described as $U_0^{lc}(x) = \frac{3}{2}\Delta G^\ddagger (x/\Delta x^\ddagger) - 2\Delta G^\ddagger (x/\Delta x^\ddagger)^3$ [1]. The transition rate is given by,

$$k_{off}(F) = k_0 \left(1 - \frac{\nu F \Delta x^\ddagger}{\Delta G^\ddagger}\right)^{\frac{1}{\nu}-1} e^{\beta \Delta G^\ddagger \left[1 - \left(1 - \frac{\nu F \Delta x^\ddagger}{\Delta G^\ddagger}\right)^{1/\nu}\right]}, \quad (2.6)$$

with $\nu = 1/2$ or $2/3$ for the cusp-like and linear-cubic energy landscape, respectively.

Monte Carlo Simulation

Algorithm 1 Protein Unfolding Monte Carlo Simulation

Require: $v, N, L_c, L_p, \Delta L_c, \Delta L_p, k_0, \Delta x^\ddagger, \Delta G^\ddagger$

```

1: Initialization:  $z \leftarrow 0$ 
2: for  $t \leftarrow 0 : \Delta t : T$  do
3:    $z \leftarrow z + v\Delta t$ 
4:   Solve Eq. 2.7 for  $q$ 
5:   Calculate  $F$  using Eq. 2.8
6:   Compute  $k_{off}(F)$  using Eq. 2.6
7:   Compute  $P_u(F)$  using Eq. 2.9
8:   Draw  $\eta \sim \mathcal{U}_{[0,1]}$ 
9:   if  $\eta < P_u(F)$  then
10:     $L_c \leftarrow L_c + \Delta L_c, L_p \leftarrow L_p + \Delta L_p, N \leftarrow N - 1$ 
11:   end if
12: end for

```

For simulations, a protein of N folded domains is considered with one end attached to the substrate and the other end attached to the force probe. For simulating constant speed experiments, the base of the cantilever probe is moved away at a constant speed v ; here the position of base of the cantilever z is initialized at zero and is updated every Δt seconds. The protein extension q is determined by solving the equation,

$$F_{WLC}(q, L_c, L_p) = k_c d. \quad (2.7)$$

Here, $F_{WLC}(q, L_c, L_p)$ is the force on the protein determined by the WLC model [39] given by,

$$F_{WLC}(q, L_c, L_p) = \frac{k_B T}{L_p} \left[\frac{1}{4(1 - \frac{q}{L_c})^2} - \frac{1}{4} + \frac{q}{L_c} \right], \quad (2.8)$$

where L_c and L_p are the contour length and the persistence length of the protein, respectively. The transition rate $k_{off}(F)$ is computed using Eq. 2.6, and the probability of a domain unfolding during the time interval Δt is found by

$$P_u(F) = N(1 - e^{-k_{off}(F)\Delta t}). \quad (2.9)$$

For determining unfolding events, a random number η is generated uniformly from 0 to 1 and is compared to the unfolding probability $P_u(F)$. No unfolding event is triggered if the random number is larger than $P_u(F)$ and the simulation will continue to the next time slot by adding time interval Δt . Otherwise, one of the domains is unfolded, leading to a decrease in the number of folded domains, N , by 1, and the simulation continues to the next unfolding event if folded domains still remain. After each unfolding event, contour length and persistence length are updated by adding increments ΔL_c , ΔL_p respectively. The constant speed pulling simulation algorithm is described in Algorithm 1, when performed with constant force, the force F is required instead of the speed v , and lines 2-4 in Algorithm 1 are skipped.

Transition Rates Correction

For the constant force mode, where the force is regulated, the typical force feedback response time τ_I for Atomic Force Microscopy (AFM) is around 2 ms [42]. Consequently, corrections are required to account for unfolding events that might not be observed due to finite temporal resolution [43]. Under ideal conditions, with $\tau_I = 0$, the transition rate is defined as the inverse of the average lifetime and is determined by,

$$\langle \tau \rangle = \int_0^\infty \tau P_s(\tau) d\tau = \int_0^\infty \tau k_{mol} e^{-k_{mol}\tau} d\tau = \frac{1}{k_{mol}}, \quad (2.10)$$

where the probability density of lifetime is given by $P_s(\tau) = k_{mol} e^{-k_{mol}\tau}$ (according to Bell model [44]) and k_{mol} is the actual molecular transition rate. However, our observations of lifetimes are limited to $\tau > \tau_I$. This limitation leads to the experimentally observed transition rate, k_{obs} , which is determined by

$$\frac{1}{k_{obs}} = \int_{\tau_I}^\infty \tau \left(\frac{k_{mol}}{e^{-k_{mol}\tau_I}} \right) e^{-k_{mol}\tau} d\tau = \frac{1 + k_{mol}\tau_I}{k_{mol}}, \quad (2.11)$$

where the term enclosed within parentheses represents the normalization of the distribution over the restricted domain. By rearranging Eq. 2.11, we derive the correction formula that compensates for unobserved unfolding events attributed to the constraints posed by the finite instrumental response:

$$k_{mol} = \frac{k_{obs}}{1 - k_{obs}\tau_I}. \quad (2.12)$$

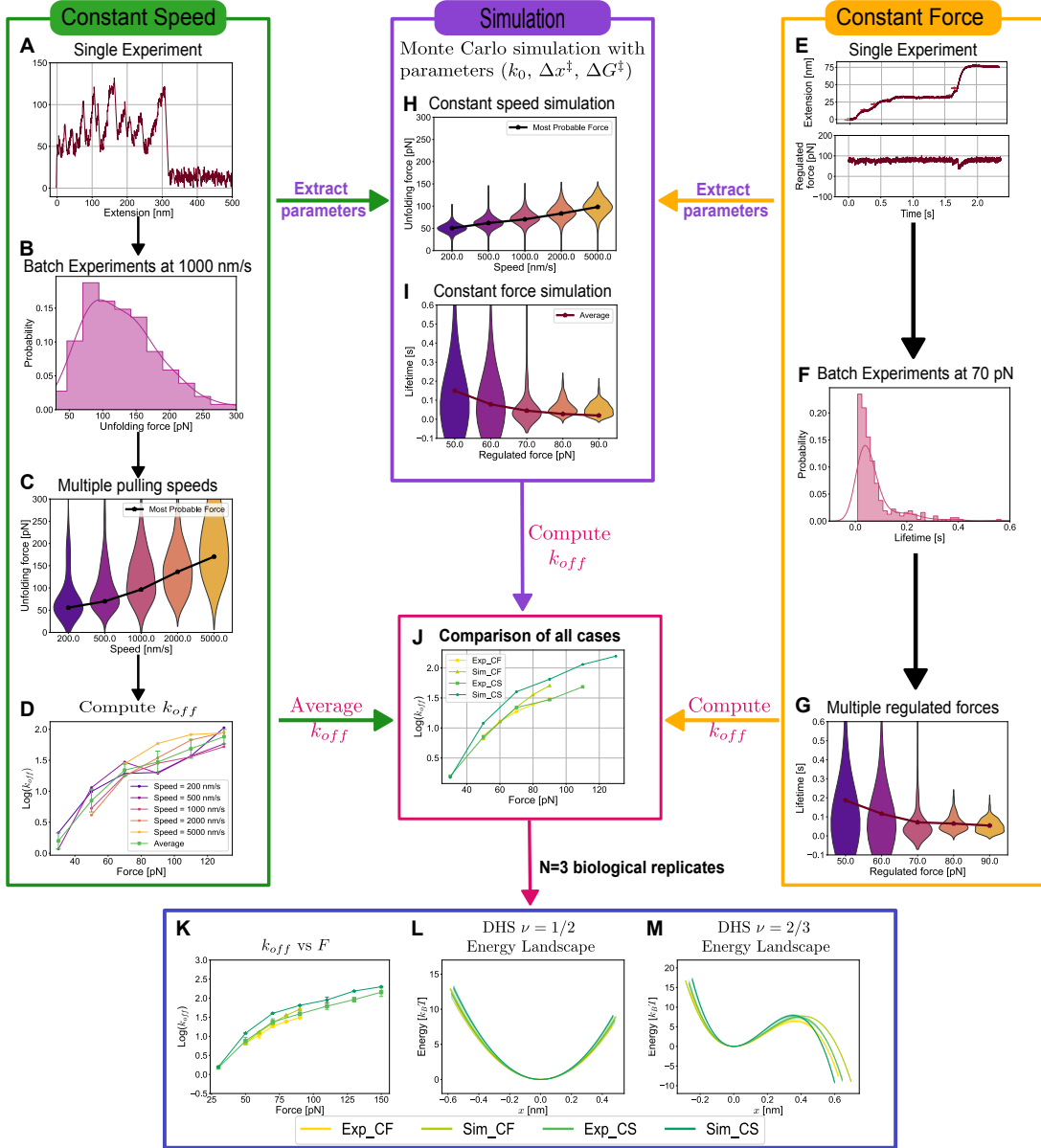


Figure 2.5: Validation of consistency between constant-speed and constant-force AFM experiments using Monte Carlo simulations (Dys data shown). (A) Force-extension curve at constant speed. (B–C) Unfolding force distributions at various pulling speeds (500–5000 nm/s). (D) Transition rates extracted from constant-speed data. (E) Extension- and force-time curves under constant force. (F–G) Lifetime distributions across forces (50–90 pN). (H–I) Simulated unfolding force and lifetime statistics. (J–K) Comparison of $k_{off}(F)$ between experimental and simulated data across both AFM modes. (L–M) Energy landscapes fitted with cusp ($\nu = 1/2$) and linear-cubic ($\nu = 2/3$) models.

Consistency of Multiple Modes of AFM

We utilized the DHS model to relate obtained statistics from the two different operational modes and further comparison was done with Monte Carlo simulations per each biological replicate. The procedure is detailed in this section using the molecule Dys (Fig. 2.5); results for other molecules are provided in Fig. 2.21.

In the constant speed mode, multiple pulls of the molecule at pulling speed at 1000 nm/s are conducted, with unfolding force distributions summarized in Fig. 2.5B, where the line represents the kernel density estimation of the corresponding histogram. The pulling speeds were then varied (500, 1000, 2000, and 5000 nm/s) with unfolding force distributions shown in Fig. 2.5C as violin plots, where lines on each side represent the kernel density estimation. Subsequently, the unfolding force distributions from different pulling speeds were used to estimate transition rate $k_{off}(F)$ using Eq. 2.5, leading to Fig. 2.5D. The average transition rate across all pulling speeds is used for comparison, represented by the green square line Fig. 2.5J. In the constant force mode, experiments were conducted at one regulated force (Fig. 2.5F) and repeated at different regulated forces ranging from 50 to 90 pN, with lifetime statistics depicted in Fig. 2.5G. The transition rates were determined using Eq. 2.10 followed by the correction with Eq. 2.12, which results in the yellow line in Fig. 2.5J.

Our experimental data show consistent transition rates across constant speed and constant force modes. The DHS model was then utilized to extract energy landscape parameters, including k_0 , $\Delta G^\ddagger$, and $\Delta x^\ddagger$, from curve fitting of Eq. 4.7 using the least squares method. Here, all domains were assumed to be homologous and independent. Monte Carlo simulations, which require extracted energy landscape parameters as well as molecular parameters such as contour length, persistence length, and their corresponding increments (estimated from experiments and molecular structure), were conducted to further validate the consistency of the results. Simulations in both constant speed and constant force modes, detailed in Algorithm 1, produced force magnitude and lifetime statistics with corresponding kernel density estimations illustrated in Fig. 2.5H and Fig. 2.5I, respectively. Transition rates from simulated data were also estimated, with results depicted in Fig. 2.5J. The green line with stars and the light green line with triangles represent transition rates obtained from simulated constant speed and constant force data, respectively.

Notably, the transition rates were closely aligned (with $CV < 0.2$ in $\log(k_{off})$) across four different scenarios: experimental constant speed, experimental constant force, simulated constant speed, and simulated constant force, as evidenced by Fig. 2.5J. This process was then expanded to compute transition rates k_{off} from different biological replicates. The mean values of k_{off} across these repeats are illustrated in Fig. 2.5K, with error bars representing standard deviations. Corresponding energy landscapes, fitted with both cusp-like and linear cubic energy landscape shapes, are shown in Fig. 2.5L and M. The shaded areas in these plots represent standard deviations of extracted parameters observed across various biological replicates. The same analysis was performed for DysN-R10, Dp260 and Utr. The results are consistent across across four distinct scenarios as shown in Fig. 2.21. Experimental and simulated data for these molecules are summarized in Figs. 2.8 and 2.18, respectively. This entire procedure was calibrated using Titin I27O (Fig. 2.11).

2.3.3 Concentration Optimization to Minimize Multiple-Protein Pulls

To further validate that our protein concentration was appropriate, we conducted additional experiments using Dys at three concentrations, 1 nMol, 10 nMol, and 100 nMol, as shown in Fig. 2.6. The probability of successful pulls at 10 nMol was 7.2%, within the recommended range. However, when the concentration was reduced tenfold to 1 nM, the probability of successful pulls dropped to 1.5%, and very few usable traces were collected. Conversely, increasing the concentration tenfold to 100 nM resulted in a much higher success rate of 26.2%, which exceeds the suggested range of 5–25%.

We analyzed the data obtained at 10 nMol and 100 nMol concentrations, as shown in Fig. 2.7. Data from 1 nMol were excluded due to the low number of useful data. In Fig. 2.7, newly collected experimental data from both constant speed and constant force measurements are compared with simulated data generated using the parameters reported in Table 1 of the main manuscript. For constant speed data (Fig. 2.7A), the most probable unfolding force increases with pulling speed at 10 nMol, as expected. In contrast, the 100 nMol data deviate from the expected trend. For constant force data (Fig. 2.7B), the average lifetime consistently decreases with increasing regulated force at 10 nMol, while no clear trend is observed at 100 nMol. Consequently, the transition rates $k_{off}(F)$ derived from 10 nMol data align well with the simulated data (Fig. 2.7C),

whereas the transition rates at 100 nMol shift toward lower values. This discrepancy may result from data of multiple molecules at the higher concentration. Furthermore, the fitted energy landscapes with $\nu = 1/2$ (Fig. 2.7D) and $\nu = 2/3$ (Fig. 2.7E) show substantial disagreement 100 nMol.

Overall, these results suggest that 100 nMol is too high a concentration, likely leading to data from multiple molecules. Conversely, 1 nMol yields too few successful events to support reliable analysis. Therefore, 10 nMol represents a good choice, and the present study is consistent with this condition, as the probability of successful pulls at 10 nMol closely matches those in our data reported in Table 2.1. In addition, a global fit with the DHS model across both operational modes produced consistent transition rates and energy-landscape parameters; Monte Carlo simulations further corroborated these results, mitigating the potential influence of multiple protein events.

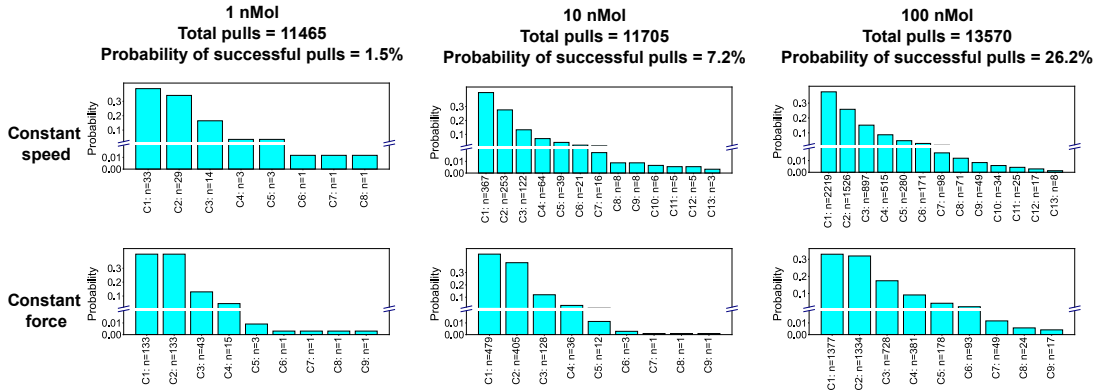


Figure 2.6: Number of unfolding events per unfolding event count for different concentrations, 1 nMol, 10 nMol, and 100 nMol (left to right) for both constant speed (top) and constant force (bottom) data. The x-axis labels are formatted as $\{C_i : n = x\}$, indicating that ‘ $n = x$ ’ unfolding events resulted in an unfolding. Total pulls refer to the combined number from both constant speed and constant force. The probability of successful pulls is calculated as the number of unfolding events at $C1$ (unfolding event count = 1) from both constant speed and constant force, divided by the total number of pulls.

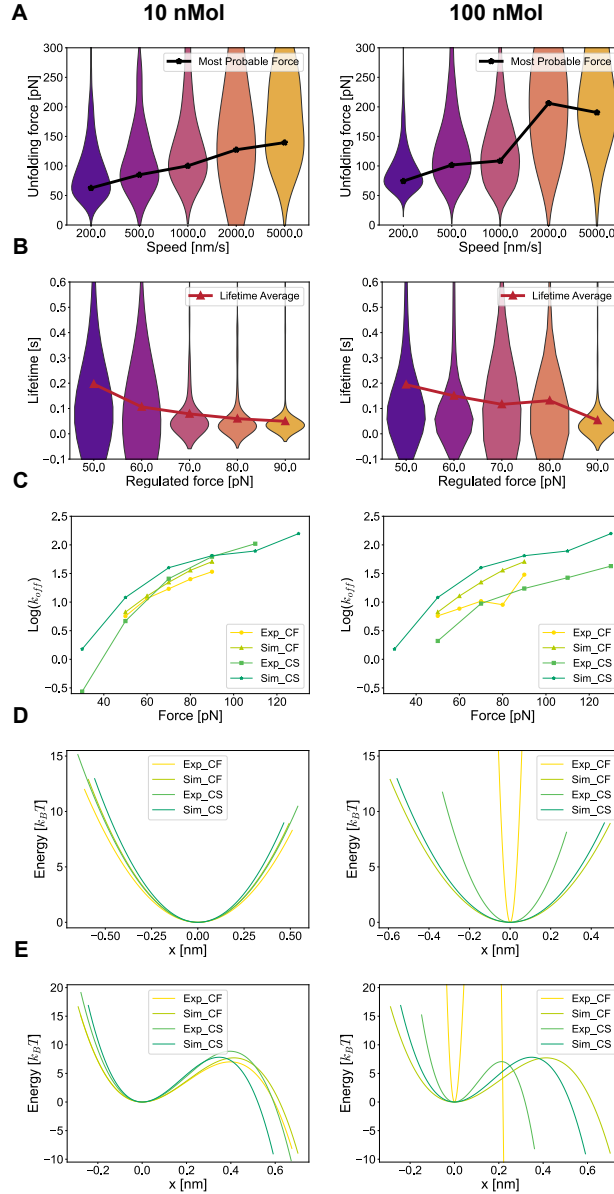


Figure 2.7: Detailed comparison of unfolding behavior at two concentrations: 10 nM (left) and 100 nM (right). (A) Unfolding force statistics across pulling speeds from 200 to 5000 nm/s, with the black stars indicating the most probable forces. (B) Lifetime statistics across multiple regulated forces ranging from 50 to 90 pN, with red triangles showing the average lifetimes. (C) Transition rates $k_{off}(F)$ for both experimental and simulated data using constant force and constant speed modes. Simulations were performed using the parameters listed in Table 1 of the main manuscript. Fitted energy landscapes with $\nu = 1/2$ and $\nu = 2/3$ are shown in (D) and (E), respectively.

2.4 Mechanical Properties of Dystrophin and Utrophin

2.4.1 The Mechanical Properties of Dys Differ Significantly from Utr, with Dys Exhibiting Brittle Unfolding Behavior

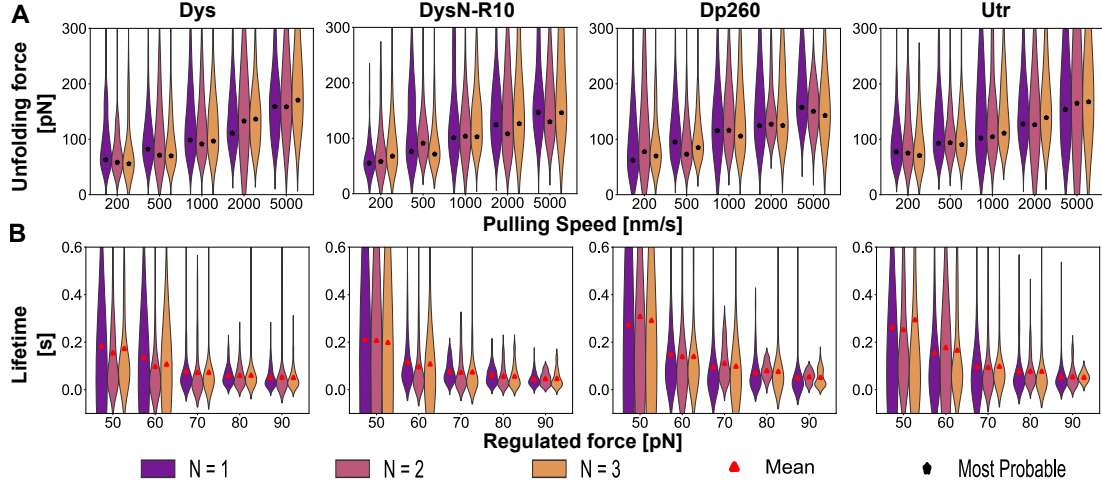


Figure 2.8: Statistical summary of $N = 3$ biological replicates for (A) constant speed and (B) constant force experiments. Each column presents results for different molecules, namely Dys, DysN-R10, Dp260, and Utr from left to right. The data from three biological repeats are visualized in purple, pink, and yellow. Unfolding force distributions from constant speed experiments and lifetime distributions from constant force experiments are depicted using violin plots. The violin plots illustrate data distributions using kernel density estimation as black lines on each side, with the width of each curve indicating the relative frequency of data points. Meanwhile, black stars signify the positions of values with the highest probability, visualized by the widest section of the violin plot, and red triangles indicate the lifetime averages, calculated from identical forces and replicates.

Data were collected across $N = 3$ preparations of both Dys and Utr using constant speed and constant force AFM modes. Violin plots of constant speed experiments conducted at five different pulling speeds (200, 500, 1000, 2000, and 5000 nm/s) are shown in Fig. 2.8A while the constant force data collected at five different forces (50, 60, 70, 80, and 90

pN) are presented in Fig. 2.8B. To further confirm that our measurements reflect single molecule events, we provide detailed histograms of the unfolding force distributions from the constant speed experiments and the lifetime distributions from the constant force experiments in Figs. 2.12 and 2.13, respectively. The unfolding force distributions for Dys and Utr shift toward higher values at higher pulling speeds (Fig. 2.8A). To quantify the consistency across different biological replicates, we used the coefficient of variation (CV), defined as the ratio of the standard deviation to the mean. The largest CV observed were 0.088 for Dys and 0.044 for Utr, both observed at 2000 nm/s (Table 2.4). The lifetime averages also exhibited a high degree of consistency among different biological replicates measured at the same regulated force, and decreased with increasing regulated forces, as illustrated in Fig. 2.8B. The largest CV were 0.147 at 60 pN for Dys and 0.065 at 50 pN for Utr (Table 2.4).

Notably, Dys consistently exhibited shorter lifetime averages than Utr under the same regulated forces (Table 2.4). As the regulated force increased, the faster unfolding events reduced the resolution, narrowing the difference in lifetime averages between Utr and Dys from 0.100 s (58.8% of Dys' lifetime) at 50 pN to 0.002 s (3.8%) at 90 pN. In constant speed mode, Dys displayed lower most probable unfolding forces than Utr at same pulling speeds. The difference decreased from 15 pN (25.8% of Dys' most probable force) at 200 nm/s to 4 pN (3.3%) at 2000 nm/s, and approached zero at 5000 nm/s. Moreover, we conducted the two sample Kolmogorov-Smirnov (KS) test [2], which tests the null hypothesis that the distributions of two samples are identical, with the level of confidence $\alpha = 0.05$. Across all pulling speeds and regulated forces, all p-values comparing Dys and Utr were below 0.05 (Fig. 2.22), indicating significantly different mechanical properties (unfolding forces and lifetimes) between these Dys and Utr proteins.

We categorized data from all biological replicates by unfolding event counts and plotted the most probable forces from constant speed experiments of all pulling speeds, and mean dwell lifetimes from constant force measurements of all regulated forces, with their corresponding confidence intervals (Fig. 2.9; see Materials and Methods for plotting details). Dwell lifetime, the time interval between consecutive unfolding events, accurately quantifies bond strength as additional domains become unbound.

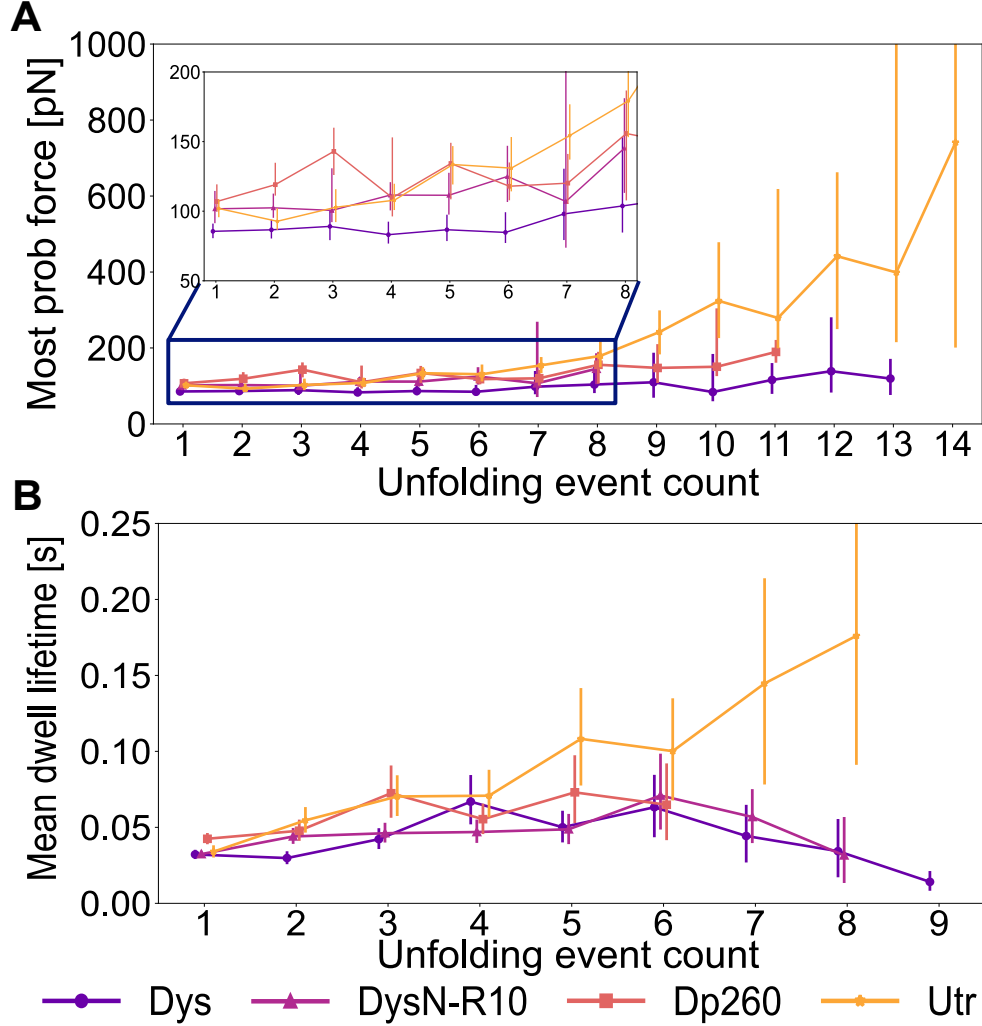


Figure 2.9: Protein mechanical behavior as a function of unfolding event count (the number of domains unfolded). (A) Most probable forces from constant speed experiments. (B) Mean dwell lifetime, the time interval between consecutive unfolding events, derived from constant force experiments. Error bars represent 95% confidence intervals, calculated using bootstrapping.

Our previous findings [10] demonstrated that Utr exhibits a stiffening spring behavior, with most probable forces increasing as the unfolding event count (the number of domains unfolded) rises. Here, we extend this analysis to include both constant speed

and constant force data, as shown in Fig. 2.9. The consistency of the data across these two operational modes further strengthens the interpretation established in our earlier work. For Dys, the most probable forces remained below 200 pN, and mean dwell lifetimes were consistently under 0.1 s across all unfolding event counts. When comparing Utr and Dys with a 95% confidence interval, Utr consistently demonstrated higher unfolding forces across all events and higher mean dwell lifetimes across all events. The divergence between Dys and Utr becomes increasingly pronounced at higher unfolding event counts. For large unfolding event counts (greater than 5), error bars overlapped only at counts of 9 and 12 from constant speed data and at 6 from constant force data. To further probe these trends, we plotted histograms for Dys and Utr at early and late unfolding event counts (Fig. 2.19). In both constant speed and constant force data, Dys exhibits similar distributions across early and late unfolding event count, whereas Utr shows marked differences with flatter distributions and shifted peaks under late unfolding events. These differences are also evident in additional example traces with a large number of unfolding events (Fig. 2.17). Finally, we applied a substantially stricter filtering criterion to the constant-speed data, which is produced in Fig. 2.20, which corroborates the trend in the data shown in Fig. 2.9A. In summary, these findings highlight significant differences in mechanical properties between Dys and Utr, showing that Dys exhibits a brittle unfolding behavior while Utr exhibits a stiffening spring behavior.

2.4.2 DysN-R10 and Dp260 Exhibit Comparable Mechanical Properties to Dys, in Contrast to the Stiffening Spring Behavior of Utr

To assess whether dystrophin’s brittle unfolding behavior is a property of the full-length protein or intrinsic to either the N-terminal or C-terminal half, we conducted experiments on two other dystrophin fragments, DysN-R10 and Dp260, under both constant speed and constant force modes across $N = 3$ biological replicates. The unfolding force and lifetime statistics are summarized in Fig. 2.8 and Table 2.4. The most probable unfolding forces ($CV < 0.11$) and lifetime averages ($CV < 0.08$) were consistent across biological replicates for both DysN-R10 and Dp260. When comparing the magnitudes of most probable forces and lifetime averages across various pulling speeds and regulated forces, Dys exhibited greater or equal magnitudes than DysN-R10 in six conditions

(2000 nm/s, 5000 nm/s, and 60-90 pN), and greater or equal magnitudes than Dp260 under three conditions (2000 nm/s, 5000 nm/s, and 90 pN). The results indicated no significant differences (p-values >0.05) between Dys and DysN-R10 under the conditions of 200 nm/s, 1000 nm/s, 5000 nm/s, 80 pN, and 90 pN; and between Dys and Dp260 at 90 pN (Fig. 2.22).

Analysis based on unfolding event counts (Fig. 2.9) revealed that neither DysN-R10 nor Dp260 exhibited the stiffening spring behavior observed in Utr. The most probable forces of DysN-R10 and Dp260 remained below 200 pN, and mean dwell lifetimes consistently stayed under 0.1 s across all unfolding event counts, mirroring the behavior observed in Dys. Quantitatively, the most probable force increases by $\sim 65\%$ for Dys (~ 85 pN to ~ 140 pN), by $\sim 45\%$ for DysN-R10 (~ 100 pN to ~ 145 pN), and by $\sim 90\%$ for Dp260 (~ 100 pN to ~ 190 pN), whereas Utr shows a much more pronounced increase from ~ 100 pN to over 400 pN (greater than 400% increase). In summary, the mechanical properties of DysN-R10, Dp260, and Dys are comparable, all exhibiting brittle unfolding behavior, in contrast to the stiffening spring behavior observed in Utr.

2.4.3 The DHS Model Reveals Similarity among Dys Constructs and Distinction from Utr

We applied the DHS model to extract key energy landscape parameters, including the intrinsic transition rate k_0 , the barrier height $\Delta G^\ddagger$, and the distance between the folded state and the transition point along the reaction coordinate $\Delta x^\ddagger$, by fitting Eq. 4.7 using the least squares method. The consistency of these results was further validated by Monte Carlo simulations, as detailed in Fig. 2.5 for the Dys molecule; similar consistency for other molecules was summarized in Fig. 2.21. Here we present the average transition rates, k_{off} , across different operational modes and biological replicates in Fig. 2.10A, where the mean values and standard deviations are represented by lines and error bars. Additionally, corresponding energy landscapes with both cusp-like ($\nu = 1/2$) and linear cubic ($\nu = 2/3$) shapes are plotted in Fig. 2.10B,C, with associated parameters listed in Table 2.3.

Table 2.3: DHS model parameters of four molecules: Dys, DysN-R10, Dp260, and Utr.

Molecules	DHS $\nu = 1/2$			
	$\ln(k_0)$	$\Delta x^\ddagger$ [nm]	$\Delta G^\ddagger$ [$k_B T$]	$\Delta x^\ddagger$ diff (%)
Dys	-2.363 ± 0.243	0.475 ± 0.028	8.528 ± 0.620	0.00
DysN-R10	-2.795 ± 0.453	0.493 ± 0.052	9.599 ± 0.633	3.79
Dp260	-2.678 ± 0.206	0.438 ± 0.018	9.408 ± 0.536	7.79
Utr	-2.719 ± 0.344	0.419 ± 0.040	9.489 ± 0.622	11.79

Molecules	DHS $\nu = 2/3$			
	$\ln(k_0)$	$\Delta x^\ddagger$ [nm]	$\Delta G^\ddagger$ [$k_B T$]	$\Delta x^\ddagger$ diff (%)
Dys	-1.809 ± 0.319	0.376 ± 0.026	7.376 ± 0.617	0.00
DysN-R10	-2.209 ± 0.417	0.396 ± 0.040	8.361 ± 0.643	5.32
Dp260	-2.242 ± 0.280	0.364 ± 0.020	8.250 ± 0.542	3.19
Utr	-2.287 ± 0.294	0.348 ± 0.025	8.256 ± 0.582	7.44

^a Energy landscapes depicted in Fig. 2.10 are based on the parameters provided here.

^b $\Delta x^\ddagger$ difference represents the absolute percentage change in $\Delta x^\ddagger$ relative to Dys.

Dys consistently exhibited larger transition rate averages (k_{off}) than Utr, with the difference decreasing as force increased (Fig. 2.10A). This observation aligns with earlier findings on comparison of lifetime averages and most probable unfolding forces (Table 2.4). Similarly, Utr had smaller transition rate averages compared to both DysN-R10, and Dp 260. Furthermore, crossover points were observed between DysN-R10 and Dys at 70 pN, and between DysN-R10 and Dp260 at 100 pN, indicating comparable mechanical properties among these molecules.

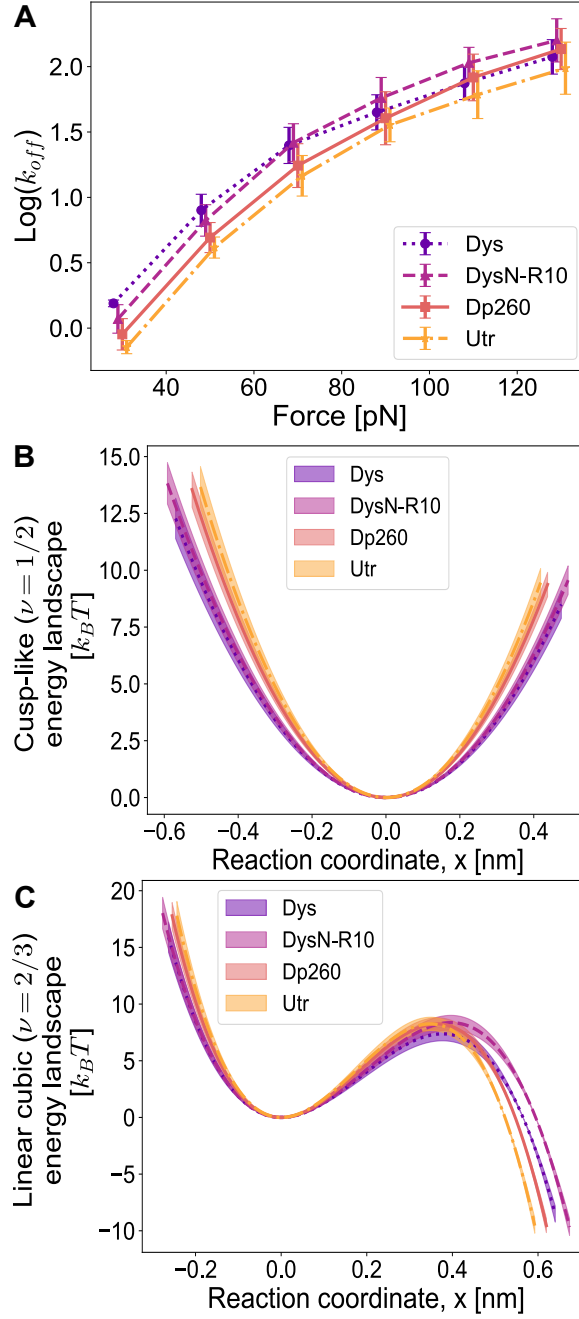


Figure 2.10: Transition rates and energy landscapes of four molecules, Dys, DysN-R10, Dp260, and Utr. (A) k_{off} -vs- F curves showing mean values (lines) and standard deviations (error bars). (B) Cusp-like ($\nu = 1/2$) energy landscapes and (C) linear cubic ($\nu = 2/3$) energy landscapes, with mean values (lines) and standard deviations (shaded area). Mean and standard deviations are computed across different operational modes and biological replicates.

From the perspective of energy landscape parameters, smaller $\Delta x^\ddagger$, k_0 and larger $\Delta G^\ddagger$ correspond to a steeper energy landscape (Fig. 2.4). We note that when k_0 are similar, $\Delta x^\ddagger$ values are generally robust to draw quantitative comparisons [45, 46]. This is because the tilted energy barrier $\Delta G_F^\ddagger$ decreases faster with larger $\Delta x^\ddagger$ under external force F , following the relationship $\Delta G_F^\ddagger = \Delta G^\ddagger - F\Delta x^\ddagger$. The absolute percentage change in $\Delta x^\ddagger$ relative to Dys when compared to Utr are 11.79% and 7.44% for $\nu = 1/2$ and $\nu = 2/3$, respectively (Table 2.3). However, smaller percentage changes are observed when Dys is compared to DysN-R10 or Dp260. Specifically, the larger absolute percentage change in $\Delta x^\ddagger$ for $\nu = 1/2$ was 7.79% for Dys versus Dp260, and it was for 5.32% for Dys versus DysN-R10. Taken together, DysN-R10 and Dp260 exhibited more similarity to Dys when compared to Utr.

2.5 Discussion

The key conclusion from our work is that both constant speed and constant force modes of atomic force microscopy show that the mechanical properties of full-length dystrophin are distinct from utrophin. We collected a large amount of data to ensure statistically meaningful results and optimized protein concentrations to ensure that results and conclusions are based on single molecule trials (Table 2.1). Our data from different modes exhibited consistency in transition rates k_{off} estimated from fitting the DHS model to both data sets. This conclusion is supported by Monte Carlo simulations (Fig. 2.21). At higher concentrations, however, this consistency breaks down (Fig. 2.7), further supporting that our measurements reflect single-molecule events rather than simultaneous pulling of multiple proteins.

Previous SMFS studies showed that small dystrophin fragments had unfolding forces of ~ 20 -50 pN [47, 48, 29], while we demonstrated that full-length utrophin and large utrophin fragments, comprising approximately half the molecule, unfold with forces of ~ 100 pN [10]. Our subsequent study showed that differences in protein expression system and phosphorylation partially account for the differences in unfolding forces between our study of utrophin and prior studies of dystrophin fragments [3]. Therefore, here we directly compared the mechanical properties of recombinant utrophin and dystrophin expressed using the same cell expression and AFM systems.

Our data showed that Dys exhibits uniform unfolding properties also displayed by its N- and C-terminal large fragments, DysN-R10 and Dp260 (Fig. 2.8, Fig. 2.22 and Table 2.4). All three molecules displayed brittle unfolding behavior, with both the most probable unfolding forces and lifetime averages remaining consistent as the unfolding event counts increases. Previous studies showed that different fragments of dystrophin’s central rod domain exhibited similar unfolding forces [29], and that forces to unfold a mini-dystrophin appeared the same as its shorter core construct [47]. These observations together with our results of longer dystrophin constructs further corroborate a brittle unfolding behavior for dystrophin. The unfolding behavior of dystrophin and its large fragments contrasts strikingly with the stiffening spring behavior that we previously reported for utrophin [10] and confirmed here (Fig. 2.9). Notably, these divergent mechanical behaviors become most apparent at higher unfolding event counts, as also observed in our previous findings [10], where the differences between utrophin and dystrophin are amplified and not explained by the properties of smaller fragments alone. Furthermore, energy landscapes supported these findings (Fig. 2.10, Table 2.3), showing that Utr is positioned significantly further away from Dys compared to DysN-R10 and Dp260.

Our study quantifies the mechanical properties of dystrophin, its large fragments (DysN-R10 and Dp260), and utrophin using both constant speed and constant force atomic force microscopy. Data from different modes, with consistency via the DHS model, revealed two distinct mechanical behaviors. Dystrophin and its fragments exhibit a brittle unfolding behavior, where sequentially unfolding domains display similar mechanical properties. In contrast, utrophin is a stiffening spring, with increasing resistance to unfolding events. Although AFM-based force spectroscopy probes forces higher than those expected *in vivo*, it offers valuable insight about how these proteins may behave. Notably, the mechanical differences between utrophin and dystrophin are increasingly pronounced at lower pulling speeds and regulated forces, suggesting that in the physiological regime of a few piconewtons, the contrast is likely to be even more significant.

We hypothesize that dystrophin’s brittle unfolding behavior allows it to dissipate mechanical energy during muscle stretching through stochastic domain unfolding, effectively protecting the sarcolemma from mechanical tearing or rupture. In contrast,

utrophin’s stiffening spring behavior, concentrates rather than dissipates mechanical stress, potentially exacerbating membrane damage through sudden energy release when forces are abruptly removed. These contrasting behaviors reinforce the hypothesis that dystrophin functions as a molecular shock absorber, protecting the sarcolemma from mechanical stress.

Utrophin’s stiffening behavior, however, appears well-suited to its native localization at the myotendinous junction [32], where its primary role involves efficient mechanical force transfer rather than energy dissipation. This functional specialization is further supported by *in vivo* evidence showing decreased passive stiffness measured at the tendons of skeletal muscles isolated from utrophin knockout mice [10]. Thus, utrophin’s mechanical behavior at the level of single molecules is consistent with its role as a stiff elastic element working in series with titin at myotendinous junctions. In contrast, the brittle unfolding behavior of dystrophin seems consistent with its hypothesized function as a membrane shock absorber. Interestingly, replacement of 4 dystrophin repeats with 4 α -actinin repeats in a micro-dystrophin transgene construct failed to rescue the dystrophic phenotypes of mdx mice [49]. While transgenic overexpression of full-length utrophin effectively rescues most mdx phenotypes [33], it does not protect mdx muscle from eccentric contraction-induced force loss *in vivo* as effectively as dystrophin or mini-dystrophin transgenes [50]. Taken together, these findings suggest that dystrophin and utrophin perform different mechanical functions in muscle cells, and that utrophin may not be an ideal therapeutic surrogate for dystrophin deficiency in DMD because it exhibits a stiffening spring mechanical response compared to the brittle shock absorber behavior exhibited by dystrophin.

2.6 Supporting Information

2.6.1 Cloning

Dys, DysN-R10, Dp260, and Utr were cloned as previously described [51]. All constructs contained an N-terminal FLAG-tag for affinity purification [51, 10].

2.6.2 Protein Expression and Purification

Insect cells used for protein expression were maintained at 1×10^6 cells/mL in 250 mL suspension cultures. Sf9 cells and Tni cells were grown in Sf-900™ II SFM (Thermo Fisher Scientific) and ESF 921™ (Expression Systems), respectively, both supplemented with penicillin/streptomycin (Sigma-Aldrich) and fungizone (Thermo Fisher Scientific) in a 250 mL suspension culture. The baculovirus generation for Utr was previously described [3]. For Dys, DysN-R10, and Dp260, higher-titer viral stocks were generated through successive infections of Sf9 cells in 6 cm (P0), 10 cm (P1), and 15 cm (P2) plates. For P0, 2.5×10^6 cells were plated and allowed to adhere for 1 hour before transfection with a bacmid and Cellfectin mix. After 5 hours at 27, extant media was removed, 4 mL Grace's complete media (Thermo Fisher Scientific), FBS, antibiotics, and antifungal were added, and the plates were incubated at 27 for 4 days. For P1, 10×10^6 cells were plated and incubated for 1 hour at room temperature before adding 1 mL of P0 virus to each plate. The cells were then incubated for another hour at room temperature, with gentle tilting every 15 minutes to ensure even virus distribution. Afterward, 9 mL Grace's complete media was added, and the plates were incubated at 27 for 3 days. For P2, 22×10^6 cells were plated on a 15 cm plate and incubated for 1 hour at room temperature before adding 0.5 mL of P1 virus. The cells were incubated for another hour, with gentle tilting every 15 minutes to ensure uniform virus distribution. 15 mL Grace's complete media was then added, and the plates were incubated at 27 for 3 days. Dys were expressed in Tni cells, Dp260 and DysN-R10 were expressed in Sf9 cells, and Utr were expressed in both Tni and Sf9 cells.

For protein expression, 250 mL 1×10^6 cells/mL cultures were transfected with 10 mL of conditioned media baculovirus and incubated at 140 rpm for 72 hours at 27. The cells were harvested by centrifuging at $500 \times g$ for 5 minutes at 4. Cell pellets were resuspended in I-PER lysis buffer (Thermo Fisher Scientific), vortexed on medium for 5 seconds, then incubated on ice for 10 minutes. Each lysate was then centrifuged at $15,000 \times g$ for 15 minutes at 4 to clear the lysates. After centrifugation, EDTA was added to the supernatant to reach a final concentration of 0.2 mM. The supernatant was applied to an anti-FLAG M2 agarose column (Sigma-Aldrich), washed with 30 mL of molecular-grade phosphate-buffered saline (PBS), and eluted with 100 μ g/mL FLAG peptide in PBS. The I-PER lysis buffer and PBS used for wash and elution had 100

nM aprotinin, 1 $\mu\text{g}/\text{mL}$ pepstatin A, 10 $\mu\text{g}/\text{mL}$ E-64, 10 μM leupeptin, 1 mM PMSF, and 2 mM benzamidine. For Dp260, DysN-R10, and Utr, 5 mL of the PBS + FLAG peptide (elution buffer) was applied to the resin, with 3 mL drained and collected in 1 mL fractions. The remaining 2 mL was left on top of the resin for 1 hour. Another 5 mL of elution buffer was added to the resin, drained, and collected in 1 mL fractions. The fourth fraction of elution was used for atomic force microscopy experiments. For Dys, 5 mL of the PBS + FLAG peptide (elution buffer) was applied to the resin, with 3 mL drained and collected in 1×3 mL, 3×1 mL, or 6×0.5 mL fractions. The remaining 2 mL was left on top of the resin for 1 hour. Another 5 mL of elution buffer (or 25 mL in the case of 1×3 mL fractions) was added to the resin, drained, and collected in 3 mL, 1 mL, or 0.5 mL fractions, respectively. The second, fourth, or eighth fraction (respectively) of elution was used for atomic force microscopy experiments. The eluted protein was dialyzed for 1 hour in 1 L of 20 mM HEPES (pH 7.5), 150 mM NaCl, 1 mM DTT, followed by overnight dialysis in another liter of buffer with the same composition. Protein concentration was determined by Bradford assay using an Albumin standard curve. Purified proteins were run on a 3-12% sodium dodecyl sulfate (SDS) polyacrylamide gradient gel and stained with Coomassie blue stain for verification (Fig. 2.1B). Coomassie stained gels were visualized and imaged using Licor Odyssey[®] Infrared Imaging System. Sucrose was added to unused protein elutions to reach a final concentration of 150 mM, and the protein elutions were then flash-frozen in liquid nitrogen.

2.6.3 Calibration Protein

Titin I27O (Athena Enzyme Systems[™]), an AFM reference protein composed of eight repeats of the Ig 27 domain of human titin, serves as the calibration protein. The most probable unfolding forces at a pulling speed of 1000 nm/s is 219.40 pN (Fig. 2.11c), closely matching the reported value of 225 pN [52]. Additionally, energy landscape parameters align with values previously reported in the literature [1], where the literature values are represented by dashed lines in Fig. 2.11f-g.

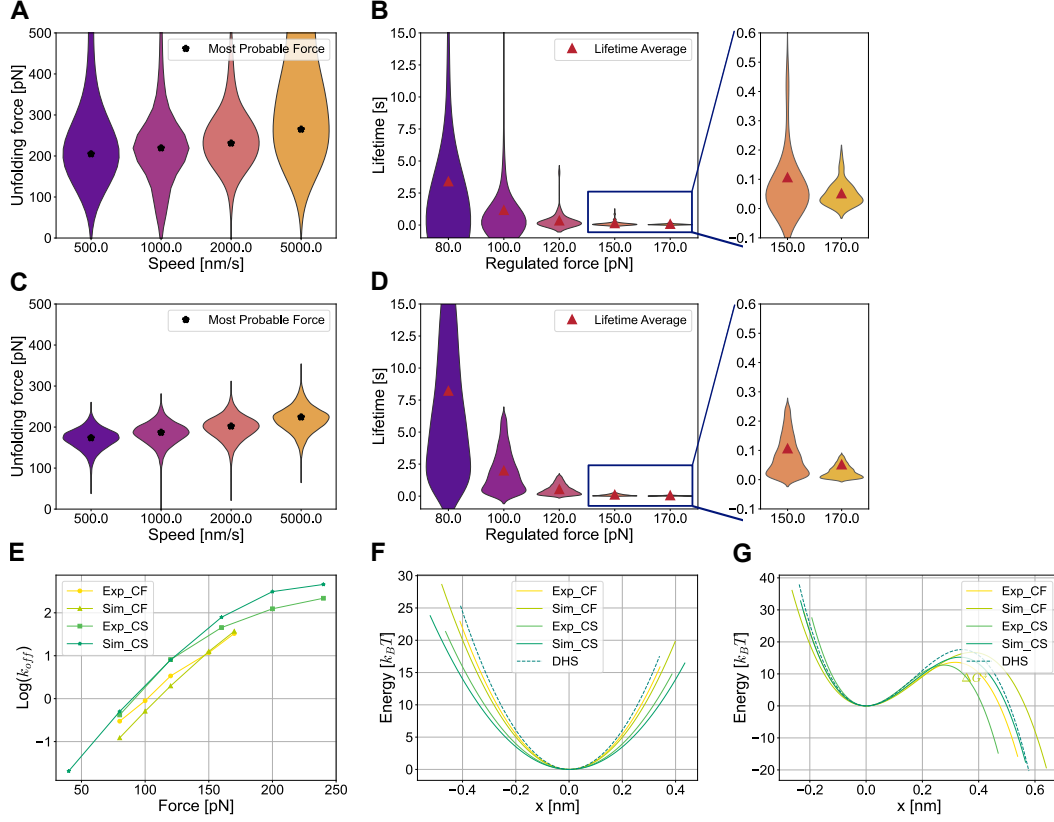


Figure 2.11: Summary of Titin I27O characterization. (A) Unfolding force statistics across pulling speeds from 500 to 5000 nm/s, with the black stars indicating the most probable forces. Typical curves are provided in SI Fig. 2.15A-C. (B) Lifetime statistics across multiple regulated forces ranging from 80 to 170 pN, with red triangles showing the average lifetimes. A zoomed-in view for regulated forces of 150 and 170 pN is shown on the right. Representative curves are displayed in SI Fig. 2.16A-C. (C) Unfolding force statistics from simulation data at pulling speeds of 500 to 5000 nm/s, with energy landscape parameters adopted from [1]. (D) Lifetime statistics from simulation data at regulated forces of 80 to 170 pN, with a zoomed-in view for forces of 150 and 170 pN. (E) Transition rates $k_{off}(F)$ for both experimental and simulated data using constant force and constant speed modes. Fitted energy landscapes with $\nu = 1/2$ and $\nu = 2/3$ are shown in (F) and (G), respectively. The corresponding energy landscape parameters are documented in Table 2.5.

2.6.4 Negligible Effect of Asparagine-tRNA Ligase

Dys and Utr preparations each contained 10% of a ~ 60 kDa band (Fig. 2.1B), which was identified by mass spectrometry to be asparagine-tRNA ligase. To further assess whether the contaminant contributed to our single-molecule data, we considered its biophysical signature. Asparagine-tRNA ligase consists of two domains (225 and 321 amino acids), giving a predicted total contour length of 199 when fully unfolded (assuming 0.365 nm per amino acid). We screened our datasets for traces with total unfolded lengths within $\pm 20\%$ of this value (i.e., 159–239 nm) and containing at most two unfolding events. The results show that only 128 out of 2,583 traces ($\sim 4.96\%$) for Utr, and 124 out of 3,277 traces ($\sim 3.79\%$) for Dys within this range. These findings indicate that traces potentially attributable to the contaminant represent a very small fraction of the total dataset, further supporting our conclusion that the contaminant does not meaningfully affect our results.

2.6.5 Histograms of Unfolding Forces and Lifetimes Across All Biological Replicates

Full histograms of unfolding forces for each construct and pulling speed across biological replicates are depicted in Fig. 2.12. We fit each distribution with two methods: (i) Gaussian kernel density estimation (KDE) and (ii) a two-Gaussian mixture model (GMM) with independent means and variances. In the GMM fits, one component aligns with the KDE mode; the second lies at a higher force with a much smaller weight and, in several cases, is barely visible (e.g., 500 nm/s for Dys; 1000 nm/s for DysN-R10; 2000 nm/s for Dys and Utr; 5000 nm/s for Utr). To quantify goodness of fit, we use the Kolmogorov–Smirnov (KS) distance [2] between the fitted cumulative distribution $F(x)$ and the empirical distribution of the unfolding-force data $\{X_1, X_2, \dots, X_n\}$,

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{(-\infty, x]}(X_i), \quad (2.13)$$

where $\mathbf{1}_{(-\infty, x]}(X_i)$ is the indicator function, equal to 1 if $X_i \leq x$ and 0 otherwise. The KS distance,

$$\text{KS} = \max_x |F_n(x) - F(x)|, \quad (2.14)$$

ranges from 0 (identical distributions) to 1 (maximally dissimilar distributions).

To compare fits, we report $\Delta\text{KS} = \text{KS}_{\text{KDE}} - \text{KS}_{\text{GMM}}$. Across all pulling speeds and proteins we find $|\Delta\text{KS}| \leq 0.03$. This very small discrepancy indicates that the minor GMM component primarily overlaps with the right-skewed tail of the main peak rather than constituting a distinct mode. We also include full histograms of lifetime for each construct and regulated forces in Fig. 2.13.

2.6.6 Unfolded Length Increases Linearly with Event Count

Spectrin-like repeats (SLRs) contain ~ 110 amino acids on average in both dystrophin and utrophin, whereas terminal domains (N-terminus, cysteine-rich, and C-terminus domains) contain at least double that number (≥ 230 amino acids). Each SLR has an approximate folded length of ~ 6 nm [53], while the folded length of the N-terminus is ~ 12 nm [54]. We can estimate the contour length increase upon unfolding a domain using $Nb - L_f$, where N is the number of amino acids, $b = 0.365\text{nm/aa}$ is the approximate contour length for each amino acid, and L_f is the folded length. Using this relation, the estimated contour length increase is ~ 34.1 nm ($0.365 \times 110 - 6$) for single SLR and at least ~ 72.0 nm ($0.365 \times 230 - 12$) for terminal domains. Our measured slopes (31.67–35.66 nm), as presented in Fig. 2.14, align closely with the single SLR value and are far below the ~ 72.0 nm expected for terminal domains.

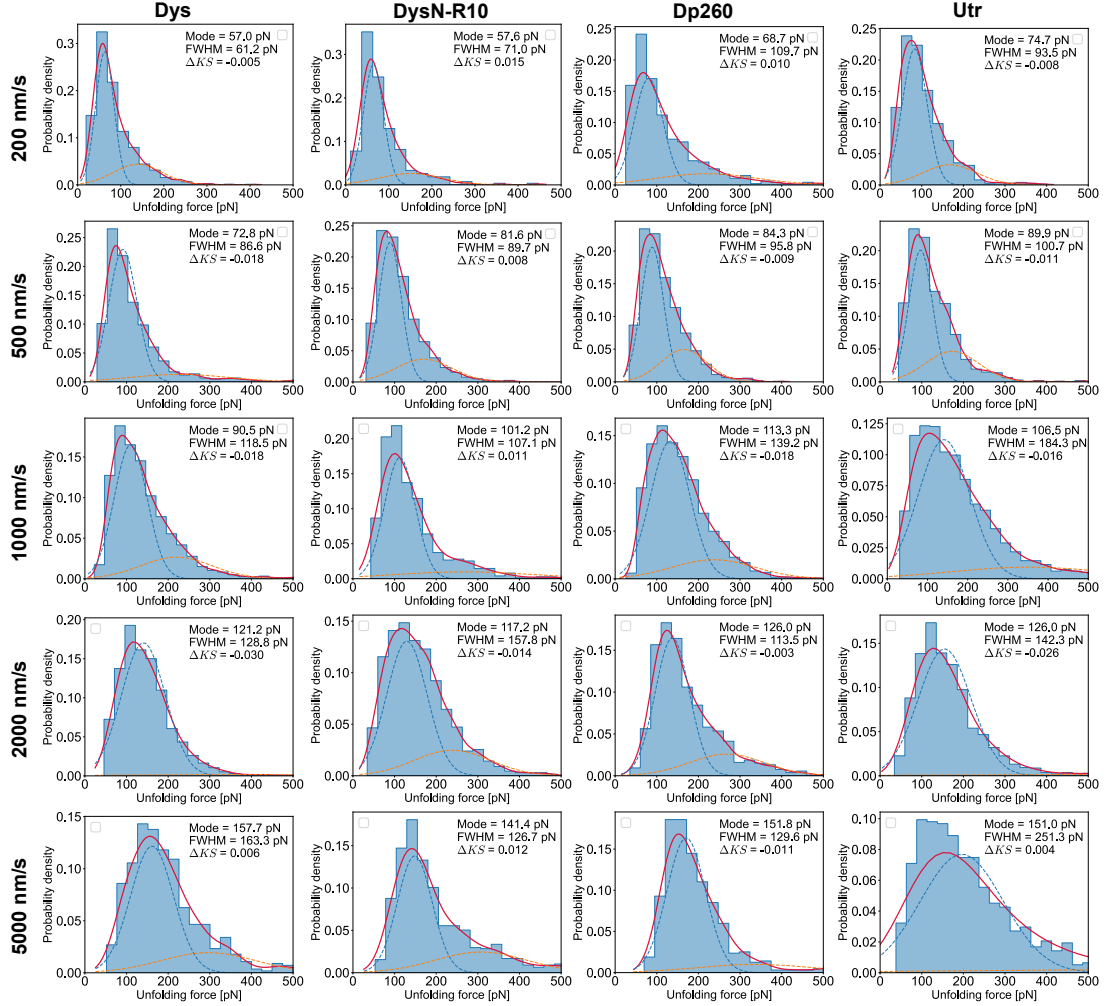


Figure 2.12: Unfolding force data from constant speed AFM experiments. Rows represent pulling speeds of 200, 500, 1000, 2000, and 5000 nm/s from top to bottom; columns correspond to Dys, DysN-R10, Dp260, and Utr from left to right. In each panel, the histogram (blue bars) is shown together with Gaussian kernel density estimation (KDE) in solid red and two Gaussian components (dashed blue and orange) of the two-Gaussian mixture modal (GMM). The most probable force (Mode), full width at half-maximum (FWHM), and Kolmogorov–Smirnov (KS) distance difference between two fitting methods $\Delta KS = KS_{KDE} - KS_{GMM}$ are annotated.

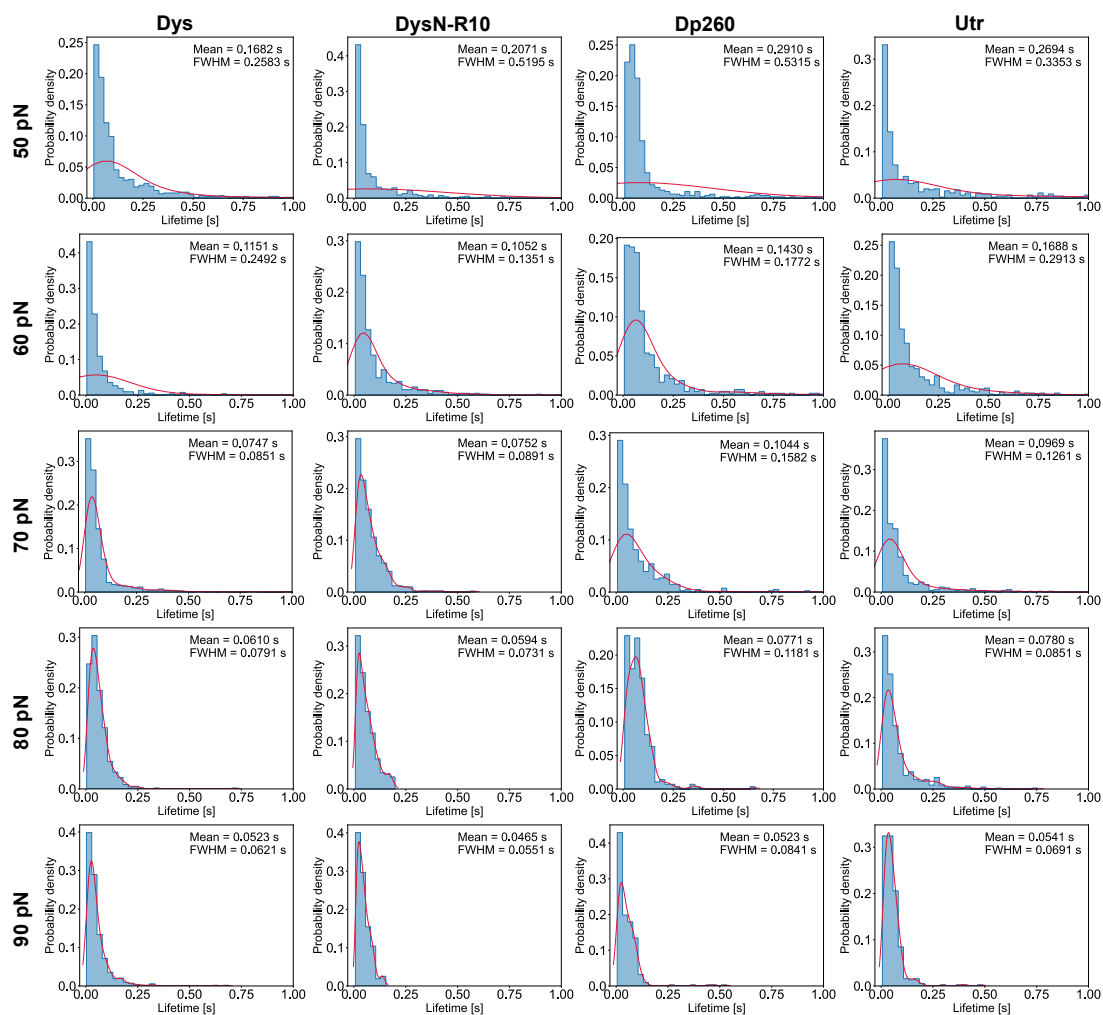


Figure 2.13: Lifetime data from constant force AFM experiments. Rows represent regulated forces of 50, 60, 70, 80, and 90 pN from top to bottom; columns correspond to Dys, DysN-R10, Dp260, and Utr from left to right. In each panel, the histogram (blue) is shown together with Gaussian kernel density estimation (red); the mean lifetime (Mean) and full width at half-maximum (FWHM) are annotated.

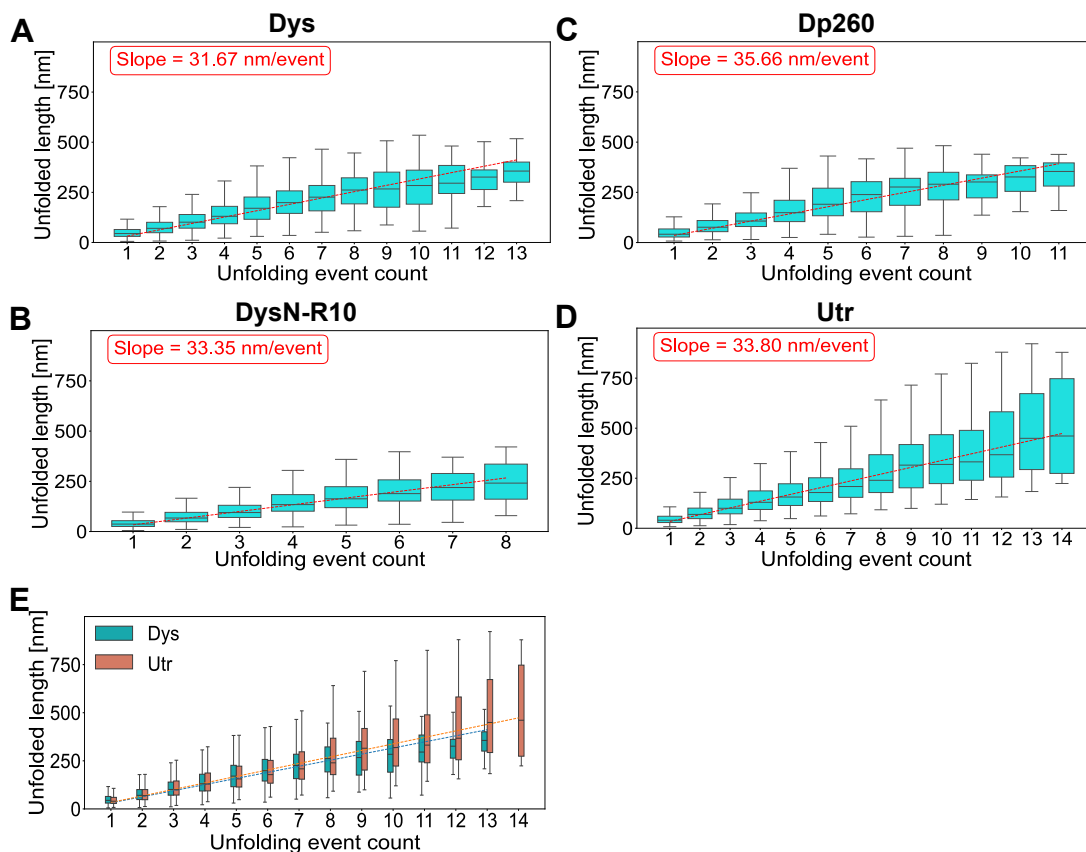


Figure 2.14: Unfolded contour length as a function of unfolding event count for proteins (A) Dys, (B) DysN-R10, (C) Dp260, (D) Utr, and (E) pairwise comparison between Dys and Utr. Each boxplot represents the distribution of unfolded lengths at a given unfolding event count. The central line indicates the median; box edges denote the 25th and 75th percentiles, and whiskers represent the minimum and maximum values excluding outliers. These figures show a linear increase in unfolded contour length with unfolding event count. A weighted linear fit (in red) was applied to the median values, using $1/\text{variance}$ as the weighting factor and including the origin (0,0). The resulting slope (in nm/event) is annotated in the upper-left corner of each panel.

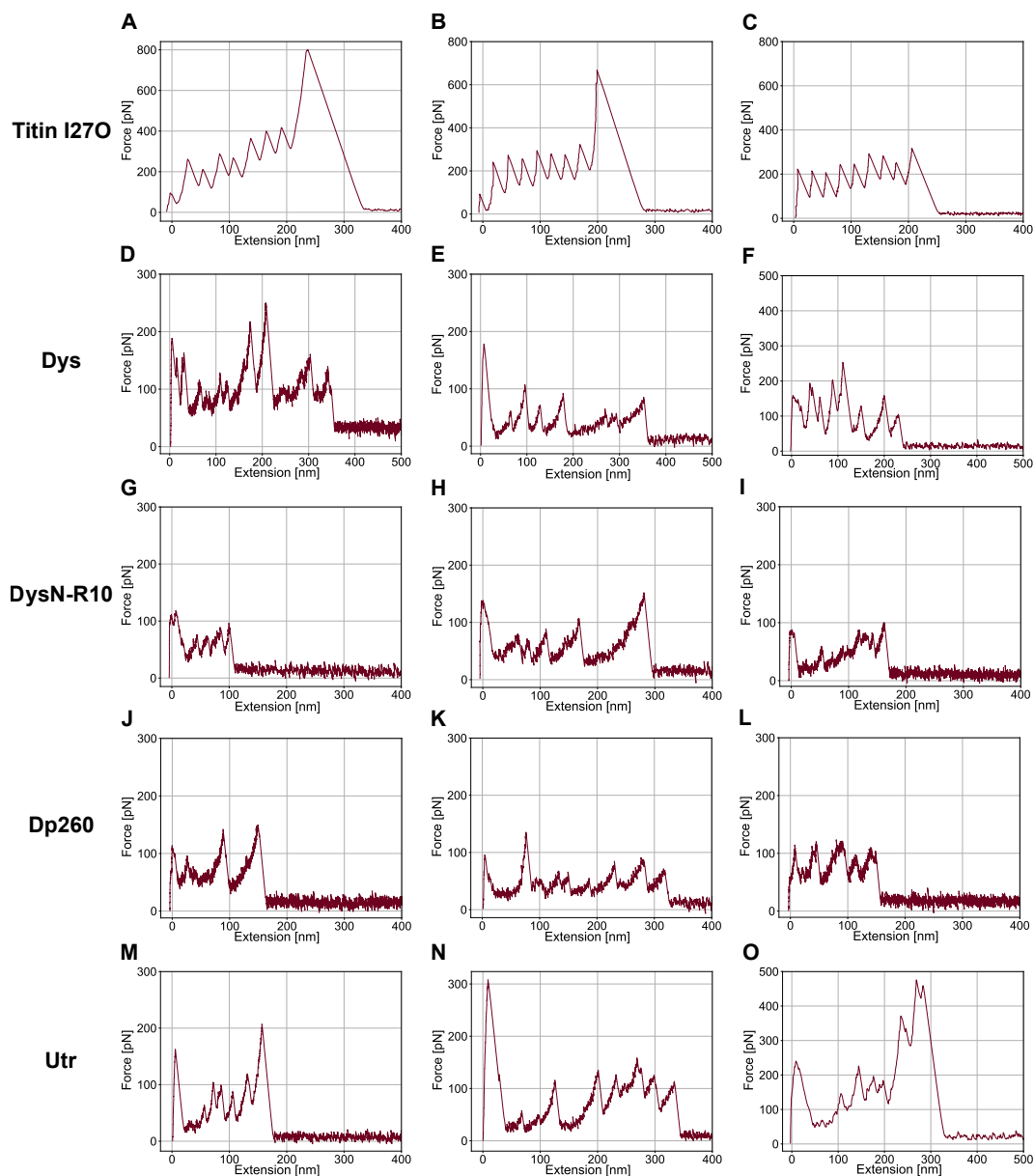


Figure 2.15: Representative curves from constant speed experiments. Each row presents results for different molecules, including Titin I270, Dys, DysN-R10, Dp260, and Utr from top to bottom.

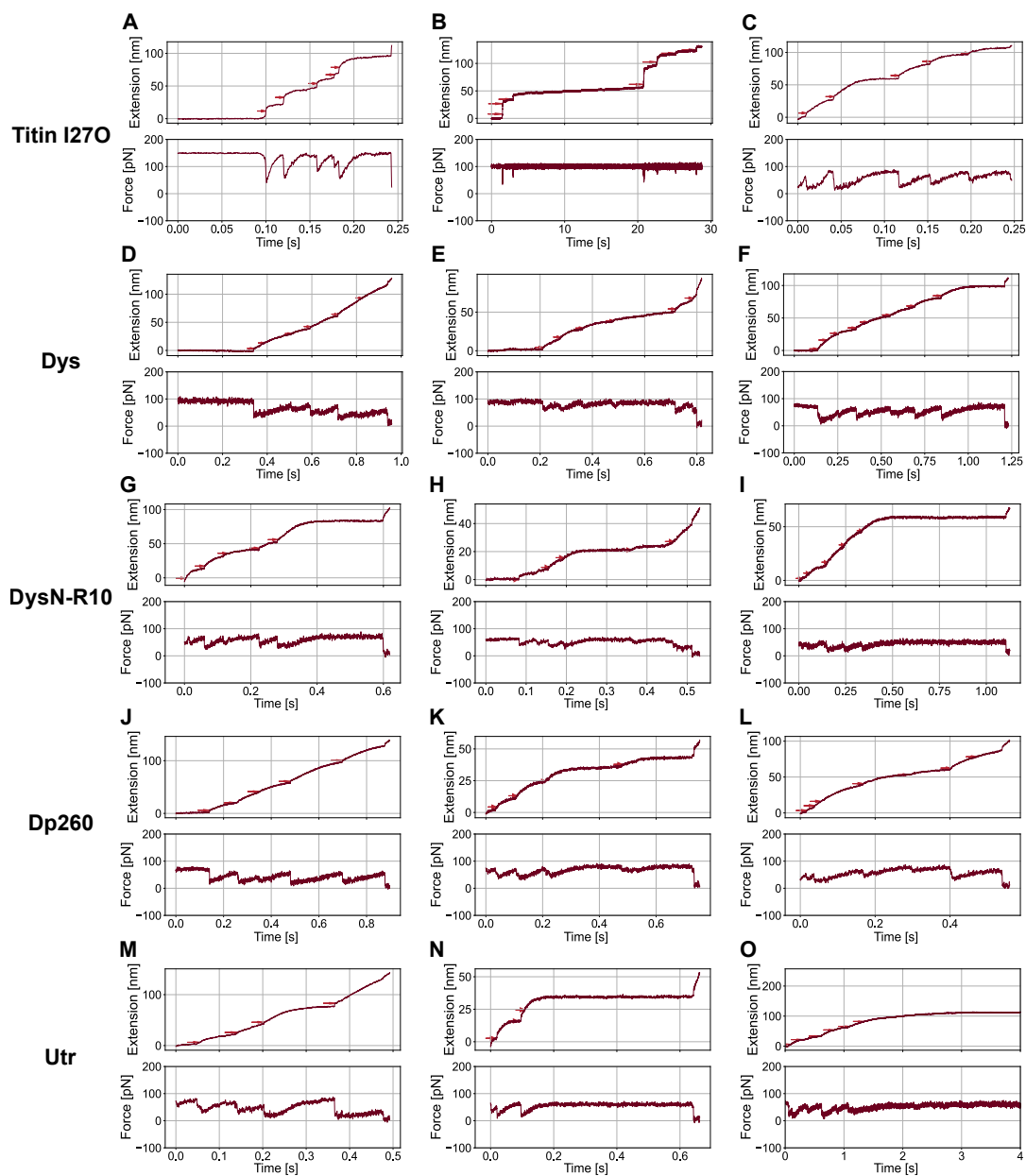


Figure 2.16: Representative curves from constant force experiment. Each row represents results namely Titin I27O, Dys, DysN-R10, Dp260, and Utr from top to bottom.

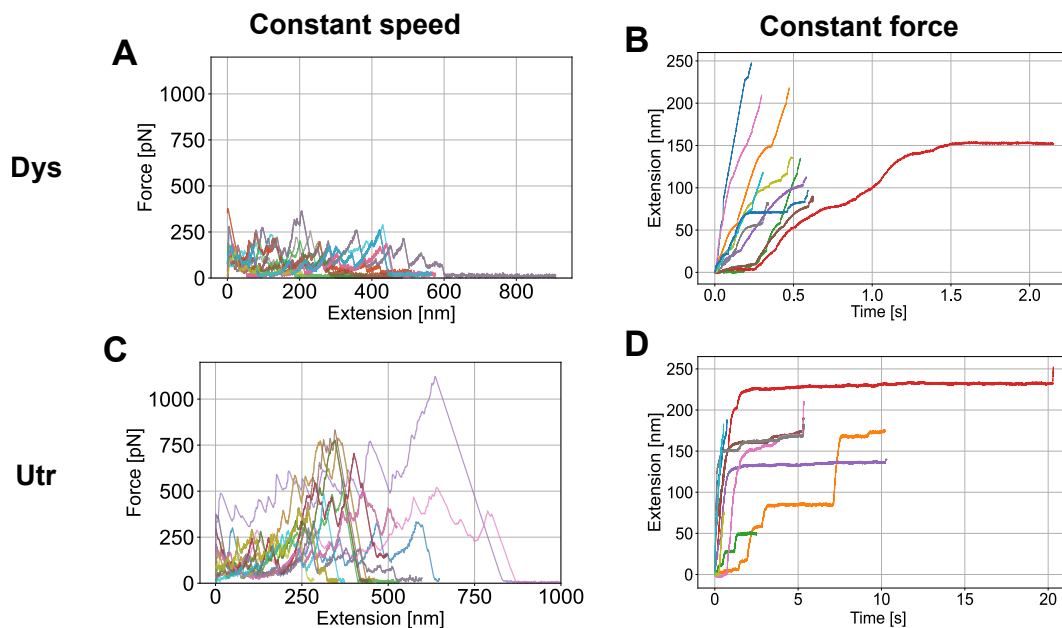


Figure 2.17: Example traces with a large number of unfolding events for constant speed experiments (left column; ≥ 10 unfolding events) and constant force experiments (right column; ≥ 7 unfolding events), shown for Dys (top row) and Utr (bottom row). In the constant speed data (A and C), Dys maintains relatively low unfolding forces (< 500 pN) across all events, whereas Utr shows increasing unfolding forces with higher unfolding event counts. Similarly, in constant force data (B and D), Utr exhibits longer total unfolding lifetimes compared to Dys.

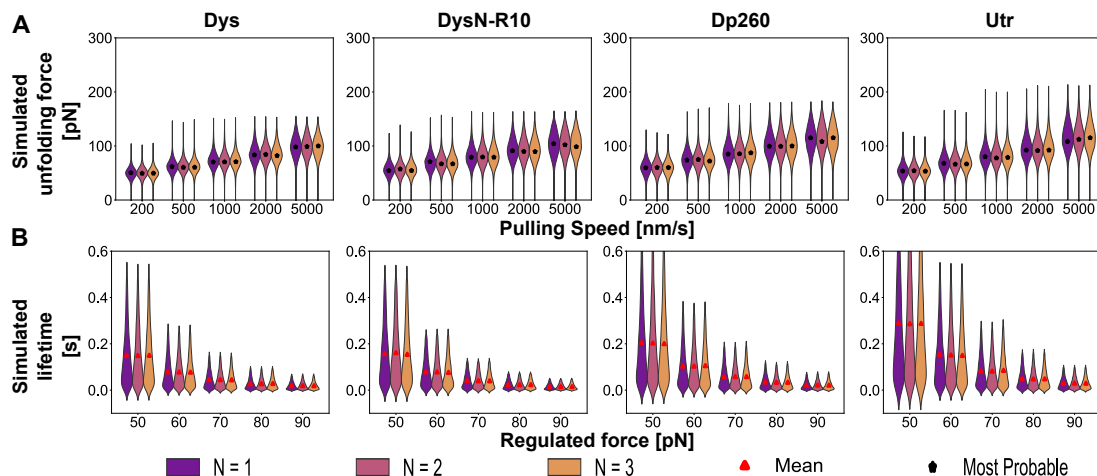


Figure 2.18: A summary of statistics of (A) simulated constant speed and (B) simulated constant force. Simulations were conducted independently and distinguished by biological replicate to be consistent with experimental data. Each column presents results for different molecules, namely Dys, DysN-R10, Dp260, and Utr from left to right. The data from three biological repeats are visualized in purple, pink, and yellow. Unfolding force and lifetime distributions are depicted using violin plots. The violins illustrate data distributions using kernel density estimation as black lines on each side, with the width of each curve indicating the relative frequency of data points. Black stars signify the positions of unfolding forces with the highest probability, and red triangles indicates the lifetime average.

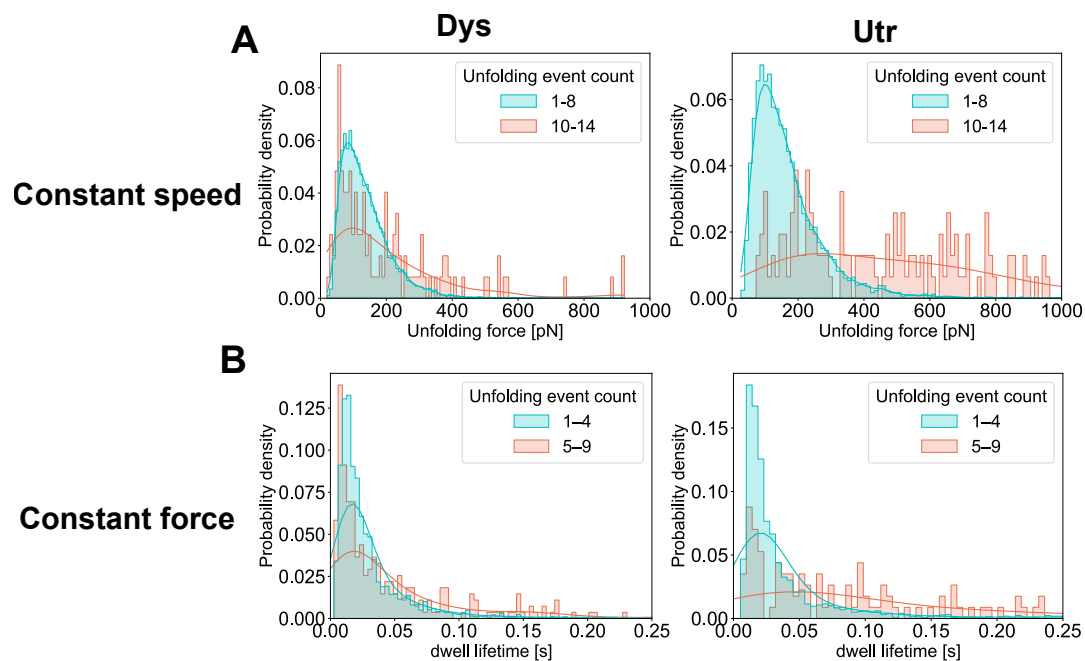


Figure 2.19: (A) Histograms of unfolding forces from constant speed data for unfolding event ranges 1–8 and 10–14. (B) Dwell time distributions from constant force data for unfolding event ranges 1–4 and 5–9. Data are shown for both Dys (left column) and Utr (right column). Here, the range x – y denotes data from unfolding events with counts satisfying $x \leq \text{unfolding event count} \leq y$. Dys exhibits similar distributions for both unfolding forces and dwell times across early and late unfolding event counts. In contrast, Utr shows marked differences, where later unfolding events (range 10–14 for constant speed, and range 5–9 for constant force) display flatter distributions with shifted peaks compared to earlier events.

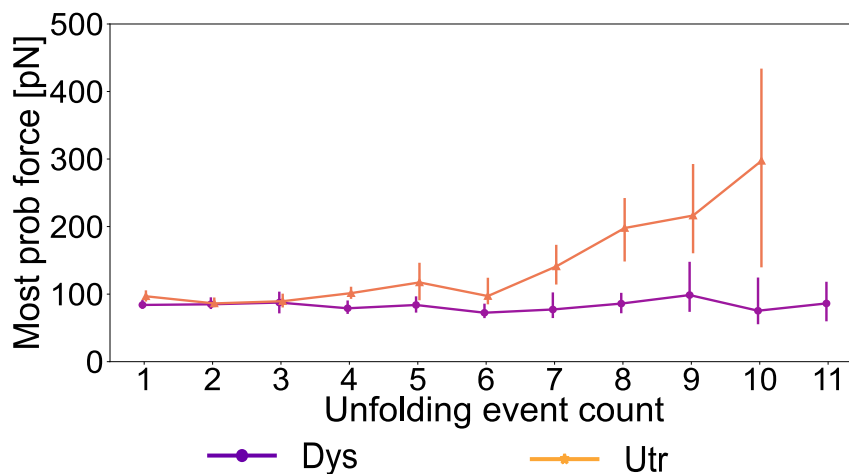


Figure 2.20: Supplementary analysis corresponding to Fig. 2.9, in which constant speed data are filtered using a substantially stricter filtering with the physics-augmented model developed in Chapter 3, removing any traces that have any possibility of not being single molecule trial. We increased the classification threshold to maintain high precision, ensuring that only very high-confidence single-molecule curves were retained. Most probable forces from constant speed experiments. Error bars represent 95% confidence intervals, calculated using bootstrapping. It shows the same trend as Fig. 2.9A: the most probable unfolding force for Utr begins to rise at an unfolding event count of 7 and remains higher than that of Dys for counts greater than 7. This consistency under more rigorous data selection further reinforces the robustness of our conclusions.

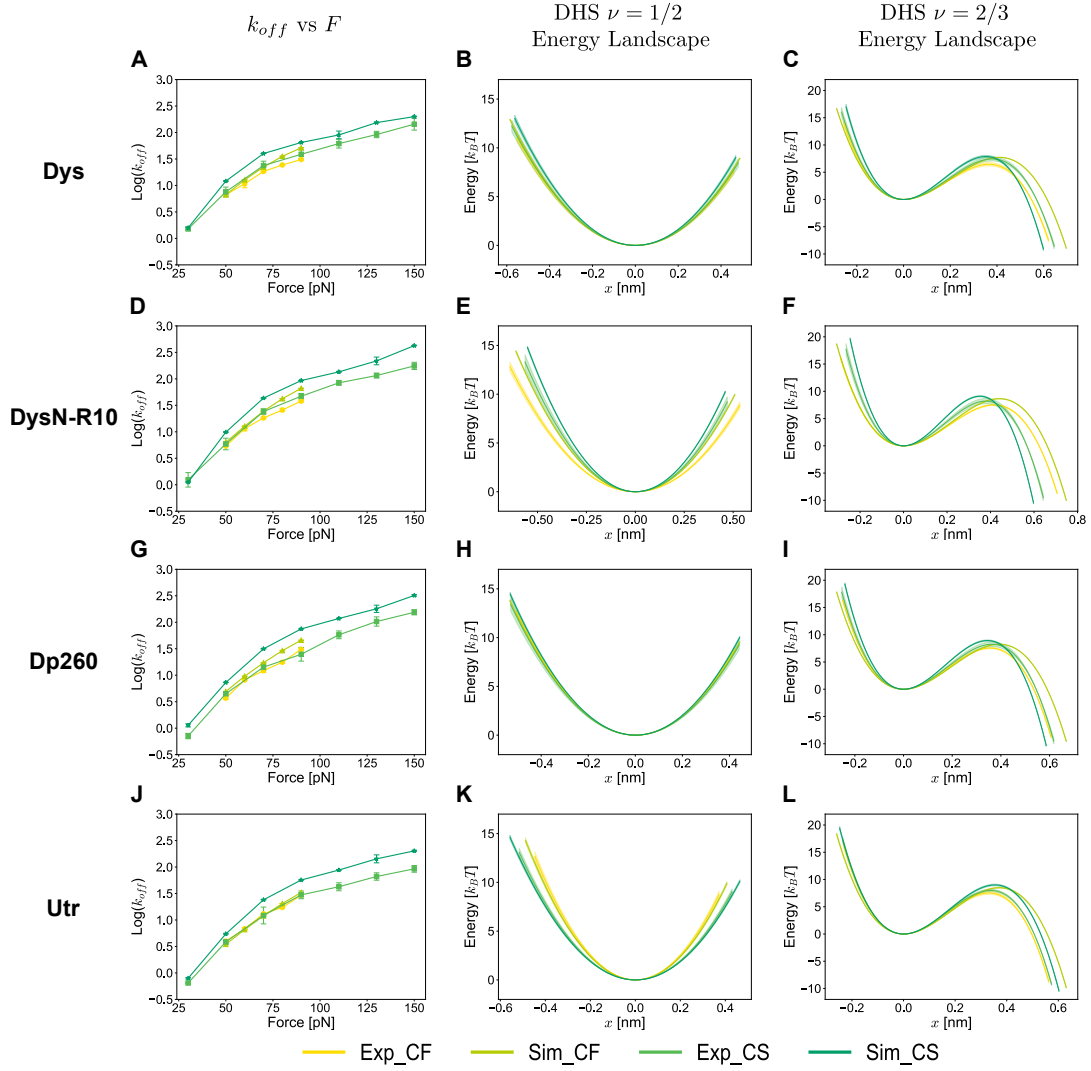


Figure 2.21: A summary of results across four distinct scenarios: experimental constant speed, experimental constant force, simulated constant speed, and simulated constant force. Each row presents results for different molecules, namely Dys, DysN-R10, Dp260 and Utr from top to bottom. The graphs in the first column show k_{off} -vs- F curves, while the second and third columns show fitted energy landscapes with $\nu = 1/2$ and $\nu = 2/3$, respectively.

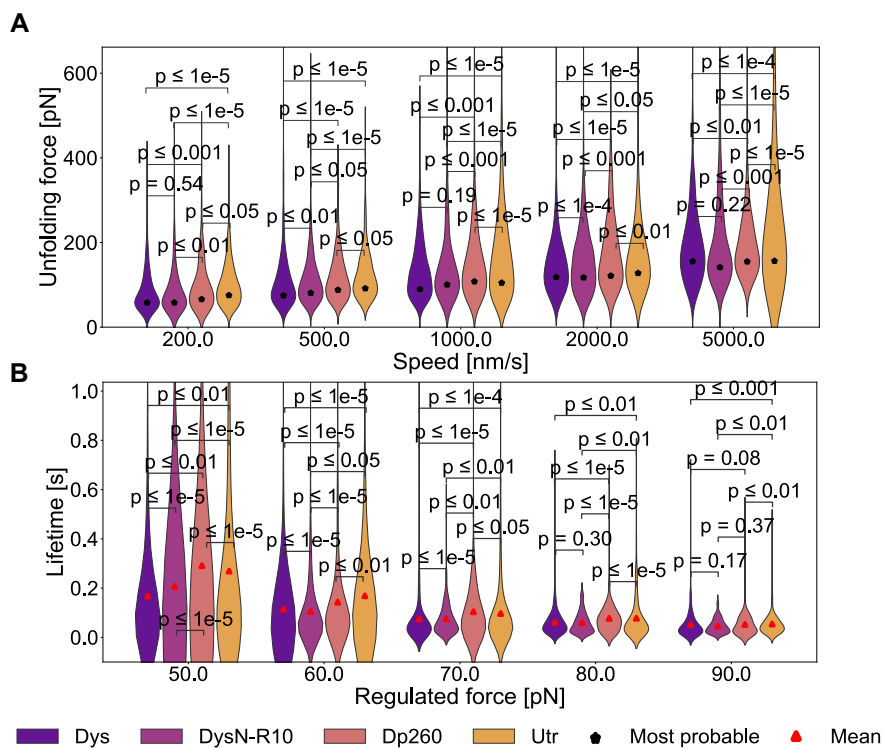


Figure 2.22: Statistics for all four molecules, Dys, DysN-R10, Dp260, and Utr, from both (A) constant speed, and (B) constant force data of all biological replicates. We conducted the two sample Kolmogorov-Smirnov (KS) test [2], which tests the null hypothesis that the distributions of two samples are identical, with the level of confidence $\alpha = 0.05$. The p-values over brackets correspond to comparisons of paired data.

Table 2.4: The experimental most probable unfolding forces and lifetime averages of four molecules: Dys, DysN-R10, Dp260, and Utr.

Conditions		Dys	DysN-R10	Dp260	Utr
Constant Speed [nm/s]	200	59.02 $\pm$ 3.05 (0.052)	60.27 $\pm$ 5.38 (0.089)	69.74 $\pm$ 6.33 (0.091)	74.23 $\pm$ 2.71 (0.036)
	500	74.42 $\pm$ 5.53 (0.074)	79.54 $\pm$ 8.32 (0.105)	84.25 $\pm$ 8.96 (0.106)	92.18 $\pm$ 1.38 (0.015)
	1000	95.49 $\pm$ 3.04 (0.032)	102.35 $\pm$ 1.14 (0.011)	112.42 $\pm$ 4.78 (0.042)	105.81 $\pm$ 3.69 (0.035)
	2000	126.66 $\pm$ 11.21 (0.088)	119.63 $\pm$ 8.28 (0.069)	125.41 $\pm$ 1.19 (0.009)	130.82 $\pm$ 5.74 (0.044)
	5000	162.68 $\pm$ 5.52 (0.034)	140.82 $\pm$ 7.91 (0.056)	150.20 $\pm$ 5.96 (0.040)	162.02 $\pm$ 5.93 (0.037)
Constant Force [pN]	50	0.171 $\pm$ 0.012 (0.069)	0.206 $\pm$ 0.005 (0.025)	0.292 $\pm$ 0.014 (0.049)	0.271 $\pm$ 0.018 (0.065)
	60	0.115 $\pm$ 0.017 (0.147)	0.108 $\pm$ 0.008 (0.079)	0.143 $\pm$ 0.004 (0.029)	0.167 $\pm$ 0.009 (0.052)
	70	0.075 $\pm$ 0.001 (0.013)	0.075 $\pm$ 0.001 (0.019)	0.102 $\pm$ 0.007 (0.067)	0.097 $\pm$ 0.002 (0.019)
	80	0.061 $\pm$ 0.001 (0.016)	0.059 $\pm$ 0.003 (0.048)	0.077 $\pm$ 0.003 (0.041)	0.078 $\pm$ 0.001 (0.013)
	90	0.052 $\pm$ 0.001 (0.019)	0.046 $\pm$ 0.001 (0.022)	0.052 $\pm$ 0.002 (0.032)	0.054 $\pm$ 0.001 (0.019)

^a Values are mean $\pm$ standard deviations from $N = 3$ biological repeats.

^b The most probable unfolding force is recorded in constant speed case, with the unit of pN.

^c The lifetime average is recorded in constant force case, with the unit of second.

^d Coefficient of variation is depicted in parentheses.

Table 2.5: Titin I27O modeling parameters

Molecules	DHS $\nu = 1/2$			DHS $\nu = 2/3$		
	$\ln(k_0)$	$\Delta x^\ddagger$ [nm]	$\Delta G^\ddagger$ [$k_B T$]	$\ln(k_0)$	$\Delta x^\ddagger$ [nm]	$\Delta G^\ddagger$ [$k_B T$]
Exp_CF	-6.911	0.342	15.936	-6.672	0.317	13.623
Sim_CF	-9.090	0.403	19.970	-8.852	0.373	17.586
Exp_CS	-7.604	0.423	15.299	-6.388	0.329	13.482
Sim_CS	-8.385	0.465	17.01	-6.954	0.341	15.506

Chapter 3

Deep Learning Models for Classifying Single Molecule Force Spectroscopy Data

Single molecule force spectroscopy (SMFS) is a powerful technique for investigating forces involved in protein domains folding and unfolding. However, SMFS trials often involve multiple protein molecules, necessitating filtering to isolate measurements from single-molecule trials. Currently, manual visual inspection is the primary method for classifying single-molecule data; a process that is both time-consuming and requires significant expertise. In this chapter, we both apply state-of-the-art machine learning models and present a novel deep learning model tailored to SMFS data. With our deep learning approach, the time required to extract meaningful statistics from single-molecule SMFS trials is reduced from a day to under an hour.

3.1 Introduction

The use of chemical functionalization of probes, along with molecular fingerprints has emerged as an approach for identifying single molecule trials [46]. Chemical functionalization modifies the surface of the force probe and substrate to enable site-specific attachment of molecules, where fingerprints are well-characterized molecules that yield

distinct unfolding patterns. Here fingerprints in the trials can be leveraged to discern single molecule trials from trials that result from multiple molecules. However, surface chemical functionalization is time-consuming, often taking at least 6 hours [55], and demands careful handling and practice. Moreover, advanced filtering techniques, informed by an understanding of all molecules involved in the complex, are essential for effectively identifying single molecule trials [46].

In contrast, conducting experiments without functionalizing probes and introducing fingerprints into the native molecule has significant advantages. Probes without functionalization are easier and less expensive to manufacture, and bio-molecules without fingerprints engineered into their structure are easier to synthesize. Moreover, there is greater confidence that the experimental data characterizes the unaltered native bio-molecule without any confounding effects introduced by fingerprints. Despite these advantages, when there are no fingerprints, distinguishing data that result from single molecules and multiple molecules is more challenging. Currently, prevalent accepted method for distinguishing the data is based on visual inspection, which is a time-consuming process that demands a high level of expertise [56, 57]. Additionally, trials need to be collected from a large number of experiments, not only to ensure statistical confidence but also because the molecule concentration is generally lowered to minimize the possibility of involving multiple molecules [3, 38]. These factors hinder obtaining precise statistics of single molecular trials without functionalization and the generation of a large, annotated dataset suitable for training deep learning models.

In this work, we introduce state-of-the-art machine learning models and present a novel deep learning model that augments the unfolding physics of protein molecules to accurately classify SMFS data into three classes: 1) no molecule, 2) single molecule, and 3) multiple molecules. The model employs a dual-branch fusion strategy, one branch incorporating the physics of protein molecules and the other operating independently of physical constraints. To train and validate our approach, we present the first publicly accessible datasets, both simulated and experimental, obtained from non-specific pulling of multi-domain molecules: titin, utrophin, and dystrophin [58]. The simulated datasets, comprising of SMFS trials that originate from a single molecule, multiple molecules, or no molecules, are created with a novel physics-based Monte Carlo engine. Extensive evaluations on these datasets, in comparison with five state-of-the-art

baseline models, demonstrate efficacy of our proposed model. Specifically, our model achieves nearly 100% accuracy on the simulated datasets, outperforming baselines by 6%. When testing on experimental datasets, our model achieves average accuracies of $79.6 \pm 5.2\%$, surpassing baselines by 11.4%, with only approximately 30 training samples. Even without expert annotations on experimental data, our model can still achieve average accuracies of $72.0 \pm 5.9\%$ when pre-trained with corresponding simulated datasets. With this deep learning approach, the time required to extract meaningful statistics from single-molecule SMFS trials is reduced from a day to under an hour. It is expected that this pilot effort will stimulate significant activity with associated impact of ML methods on the understanding of protein folding and unfolding that emphasize mechanical properties. This work results in SMFS experimental datasets from four important protein molecules crucial to many biological pathways. To support further research, we have made our datasets publicly available and provided a Python-based toolbox (<https://github.com/SalapakaLab-SIMBioSys/SMFS-Identification>).

3.2 Related Work

Single molecule classification A 1D convolutional neural network trained using a triplet loss function [59] was utilized to classify force curves into single, multiple, or no molecule classes, with reported accuracy ranging from 65 – 70% [60]. Moreover, a machine learning workflow was proposed to iteratively classify different unfolding pathways of single molecule curves [61]. However, these two datasets were collected with chemically functionalized probes and fingerprints. The chemical functionalization process is dependent on the specific molecule and thus cannot be made agnostic to the molecule under investigation. Moreover, in the first dataset, each single molecule curve contains only one unfolding event [60], simplifying the classification problem; the second dataset comprises images rather than time series data [61], which introduces redundancy given that force curves are inherently time series data. There are currently no time series datasets available, which are from non-specific pulling of multi-domain protein molecules; most naturally occurring protein molecules have multiple domains. Here, we construct such datasets for classification purposes.

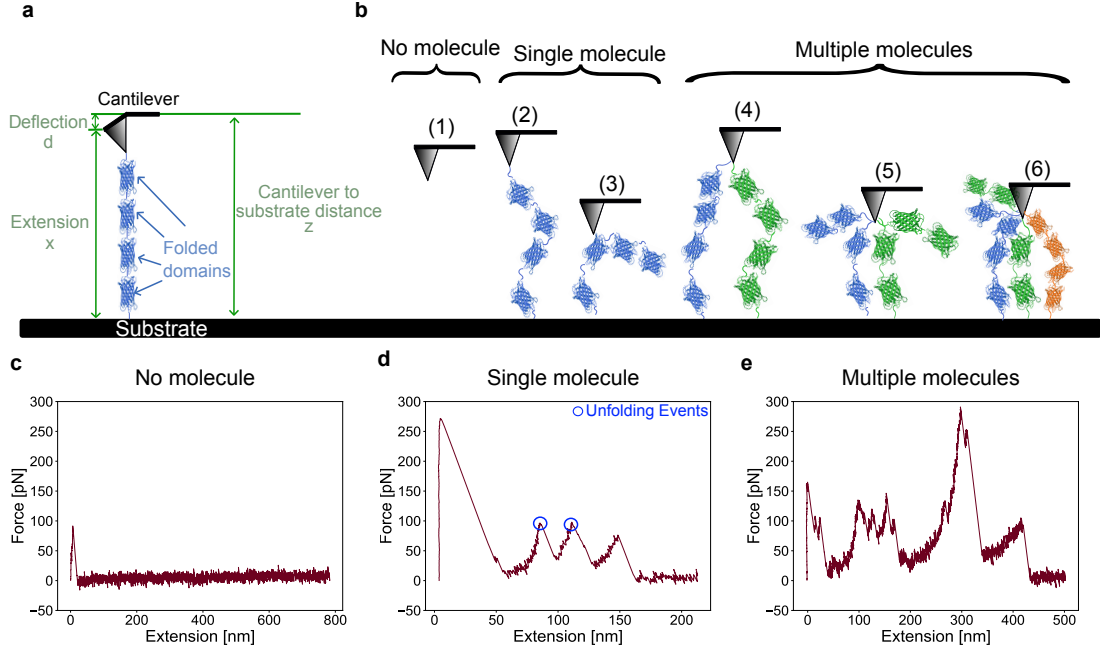


Figure 3.1: Illustration of AFM based SMFS. (a) Schematic showing the desirable case of a single protein molecule with four folded domains under tension between the tip of the AFM cantilever and the substrate. The deflection d and the separation between the cantilever and the substrate z are measured. The tensile force on the protein molecule is computed from the deflection d . (b) Depictions of possible scenarios, categorized into three classes: (1) no molecule present between the tip and substrate, (2-3) a single molecule or a section of a single molecule present between the tip and the substrate, and (4-6) multiple molecules or sections of multiple molecules between the tip and the substrate. (c-e) Show example force curves representative of the three different classes, with blue circles highlighting unfolding events.

Time series classification (TSC) More than hundreds of time series classification (TSC) algorithms, including both non-deep learning methods [62] and deep learning methods [5, 63], are present in prior-art. Although more than 80 different datasets from the University of California, Riverside (UCR) time series classification repositories [64] are evaluated with these methods, none of these datasets include SMFS data. Methods

that do not use deep learning become computationally intensive and impractical to execute on large-scale datasets [62, 5]. In this article, we focus on deep learning methods to classify our SMFS datasets.

3.3 Problem Formulation

We first describe an atomic force microscope (AFM) based SMFS setup. Here, a micro-cantilever with a sharp tip is pressed against a substrate on which the protein molecules under study are deposited. Under applied force, parts of one or more protein molecules are non-specifically attached to the cantilever tip; characterized by a stochastic adhesion event [65]. Upon retraction of the cantilever from the surface, sections of the protein molecules between the tip of the cantilever and the substrate experience a tensile force. The record of the force F experienced by the cantilever (and therefore the molecule) versus molecule extension x is known as a force curve, as depicted in Figure 4.1c-e. If only one protein molecule is present between the cantilever tip and substrate, the force curve unveils important mechanical properties of the protein molecule. We illustrate such a scenario in Figure 4.1a, where a single protein molecule with four folded domains is attached between the substrate and the tip of the cantilever. As the cantilever retracts, the molecule experiences mechanical tension, eventually causing a folded domain to unfold. The applied force drops abruptly when a domain unfolds, as highlighted by the blue circles in Figure 4.1d. This process continues until either all domains are unfolded or the connection between the cantilever and the substrate is broken [8], producing a saw-tooth pattern of force curves (Figure 4.1d).

In practice, the force curves can be categorized into one of three classes - 1) **No molecule**: where no molecule is present between the tip and substrate, 2) **Single molecule**: when only a single molecule or a section of a single molecule is present, or 3) **Multiple molecules**: where multiple molecules or sections of multiple molecules are present between the tip and substrate. Example experimental force curves corresponding to the three classes are depicted in Figure 4.1c, 4.1d, and 4.1e respectively. The force curves originating from multiple molecules typically exhibit larger unfolding forces than those with a single molecule (Figure 4.1e) and have a mixture of unfolding events that cannot be traced back to a specific protein molecule (Fig. 4.1b (4-6)),

confounding useful interpretation. Here, excluding force curves with no molecule and multiple molecules is necessary to obtain accurate and interpretable data from SMFS. The identification of the *single molecule* force curves is challenging due to a number of reasons: 1) a large number of force curves (2000-5000) need to be collected in a single experiment since protein molecule capture success rates are kept at 1-5% [38, 3], 2) the study of a specific protein molecule involves at least three replications for confidence on results, 3) the force curves are corrupted from instrument measurement noise and intrinsic thermal noise of the molecules, and 4) often the protein molecules under investigation have no prior characterization, which makes the adjudication time consuming and difficult even for domain experts.

The i -th force curve, of length $T^{(i)}$, consists of force data $\mathcal{F}^{(i)} = [F_1^{(i)}, F_2^{(i)}, \dots, F_{T^{(i)}}^{(i)}]$ and extension data $\mathcal{X}^{(i)} = [X_1^{(i)}, X_2^{(i)}, \dots, X_{T^{(i)}}^{(i)}]$. Each force curve $(\mathcal{F}^{(i)}, \mathcal{X}^{(i)})$ is associated with a class label $Y_i \in \{0, 1, 2\}$, corresponding to one of the three classes. Given a dataset of n samples, $\mathcal{D} = [(\mathcal{F}^{(1)}, \mathcal{X}^{(1)}, Y^{(1)}), \dots, (\mathcal{F}^{(n)}, \mathcal{X}^{(n)}, Y^{(n)})]$, the goal is to design an effective deep learning model capable of accurately predicting the class label of a force curve.

3.4 The Physics-augmented Model

We introduce **P**olymer **E**lastic **M**odels **N**eural **N**etworks (PemNN), a novel deep learning model designed to classify force curves as originating from no molecule, single molecule or multiple molecules. PemNN contains two branches, the force trace branch, which uses the force data $\mathcal{F}^{(i)}$, and the physics-based branch, which incorporates polymer elastic models (Section 3.4.1) with both extension and force data. Features extracted from these branches are fused using either early fusion (depicted in Figure 3.2) or late fusion strategies (see Section 3.4.2).

Both branches pass through a convolutional block, comprising a 1-Dimensional convolutional layer, batch normalization layer [66] and a Rectified Linear Unit (ReLU) [67] activation layer. Convolutional layers have demonstrated compelling performance and efficiency in time series classification [63, 5, 68, 69, 70, 71, 72]. Then features from the first convolutional block of both branches are fused with one of four methods detailed in Section 3.4.2. The fused output undergoes two additional convolutional blocks, and

a Global Average Pooling (GAP) layer is applied to reduce parameters by averaging across the time dimension [63, 5].

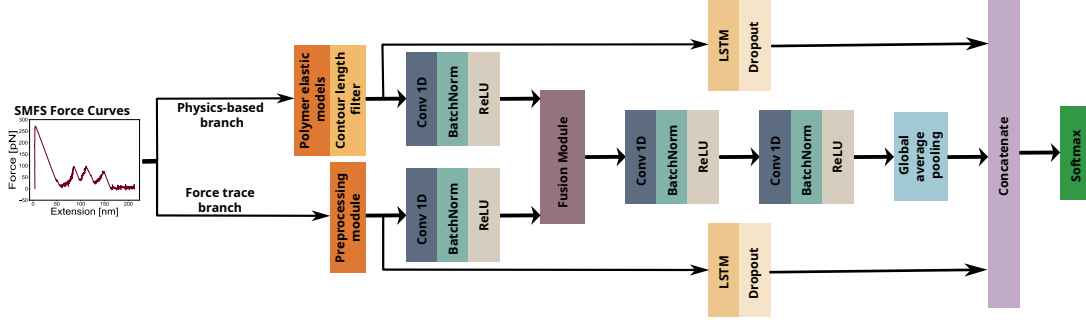


Figure 3.2: The overall architecture of the Polymer Elastic Models Neural Networks (PemNN) comprises a physics-based branch and a force trace branch, illustrated with the early fusion approach (the late fusion approach is shown in Figure 3.3).

To further enhance temporal encoding, a long short-term memory (LSTM) [73, 74] layer is applied to each branch, followed by a dropout layer [75] to mitigate overfitting [68]. Earlier studies have shown that augmenting convolutional layers with LSTM significantly improves performance in time series classification with only a modest increase in the number of parameters [68, 70, 76].

The outputs of the GAP and LSTM layers are concatenated and fed into a fully connected layer with three neurons, corresponding to the three classes: 1) no molecule, 2) single molecule, and 3) multiple molecules. A softmax activation function is used in the final fully connected layer, and the model is trained with categorical cross-entropy loss:

$$L(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) = - \sum_{j=1}^3 Y_j^{(i)} \log(\hat{Y}_j^{(i)}), \quad (3.1)$$

where $L(\mathcal{F}^{(i)}, \mathcal{X}^{(i)})$ represents the loss for classifying the force curve $(\mathcal{F}^{(i)}, \mathcal{X}^{(i)})$. Here, $Y_j^{(i)}$ is the label for class j of i -th force curve ($Y_j^{(i)} \in \{0, 1\}$), and $\hat{Y}_j^{(i)}$ is predicted probability for class j of i -th force by the neural network.

3.4.1 Polymer Elastic Models

Various polymer elasticity models have been developed to model the force-extension ($F - x$) behavior of entropic elasticity of polymers (proteins included). The widely used models are the worm-like chain (WLC) model [77], the freely jointed chain (FJC) model [78], and the freely rotating chain (FRC) model [79].

The force-extension relationship for each model is given by the following equations:

- The WLC model:

$$F = \frac{k_B T}{L_p} \left[\frac{1}{4 \left(1 - \frac{x}{L_c}\right)^2} - \frac{1}{4} + \frac{x}{L_c} \right], \quad (3.2)$$

where k_B is the Boltzmann constant, T is temperature, L_c is the contour length, and L_p is the persistence length.

- The FJC model:

$$\frac{x}{L_c} = \coth\left(\frac{F L_k}{k_B T}\right) - \frac{k_B T}{F L_k}, \quad (3.3)$$

where L_k is the Kuhn length.

- The FRC model:

$$\frac{x}{L_c} = 1 - \frac{k_B T}{2 F L_b}. \quad (3.4)$$

where L_b is the bond length.

In these models, F and x represent the force and extension respectively, k_B is the Boltzmann constant, T is temperature, L_c is the contour length, and L_p is the persistence length. The contour length L_c represents the maximal length of physically possible extension and the persistence length L_p quantifies the bending stiffness of the polymer. [80] first proposed solving for contour length L_c using each data point, rather than fitting each loading region with a polymer model. This approach has been widely adopted in SMFS studies [81, 6, 82, 83, 84, 85]. Given the physically relevant constraints ($L_c > 0, x > 0, F > 0, x < L_c$) and fixed values for the persistence length L_p , Kuhn length L_k , and the bond length L_b , the contour length L_c can be calculated using the following equations for three polymer elasticity models:

- The WLC model [82]:

$$L_c^{WLC} = \frac{x}{6u} \left(3 + 4u + \frac{9 - 3u + 4u^2}{g(u)} + g(u) \right), \quad (3.5)$$

where $u = \frac{FL_p}{k_B T}$ and

$$g(u) = \left(27 - \frac{27}{2}u + 36u^2 - 8u^3 + \frac{3\sqrt{3}}{2}\sqrt{-u^2[(4u-3)^3 - 108]} \right)^{1/3}.$$

- The FJC model:

$$L_c^{FJC} = \frac{x}{\coth\left(\frac{FL_k}{k_B T}\right) - \frac{k_B T}{FL_k}}. \quad (3.6)$$

- The FRC model, assuming bonds are connected by a fixed angle γ [79, 81]:

$$L_c^{FRC} = \begin{cases} \frac{3xk_B T}{Fa} & \text{for } \frac{FL_b}{k_B T} < \frac{L_b}{p} \\ \frac{x}{1 - \left(\frac{4Fp}{k_B T}\right)^{-1/2}} & \text{for } \frac{L_b}{p} < \frac{FL_b}{k_B T} < \frac{p}{L_b}, \\ \frac{x}{1 - \left(\frac{2FL_b}{k_B T}\right)^{-1}} & \text{for } \frac{p}{L_b} < \frac{FL_b}{k_B T} \end{cases} \quad (3.7)$$

where $a = L_b \frac{1+\cos\gamma}{(1-\cos\gamma)\cos(\gamma/2)}$ is the Kuhn length and $p = L_b \frac{\cos(\gamma/2)}{|\ln(\cos\gamma)|}$ is the persistence length.

In the physics-based branch, the contour length $L_{cp}^{(i)}$ is estimated for each force-extension pair $(F_p^{(i)}, \mathcal{X}_p^{(i)})$ using the WLC model with a fixed persistence length L_p . A subsequent filtering step selects P samples with $L_{cp}^{(i)} \in [0, M]$, where M is the filter threshold. If the number of qualified data points is less than P , sampling is performed with replacement.

3.4.2 Fusion Module

The early fusion approach contains four methods: (i) Early-Sum: sums convolutional feature maps; (ii) Early-Max: selects the maximum value at each element from convolutional feature maps, (iii) Early-Wavg: computes a weighted sum of convolutional feature maps with learnable weights; and (iv) Early-Conv: concatenates convolutional feature maps along the filter dimension, followed by a convolution with a kernel size of 1.

With late fusion method (depicted in Figure 3.3), features from the first convolutional block are independently processed through two additional convolutional blocks for each channel, with shared parameters. GAP layers are applied to the outputs of both branches, and their results are fused using one of the following methods: (i) Late-Concat: directly concatenates the outputs of GAP layers; and (ii) Late-Gating [86]: Concatenates GAP layer outputs with learnable weights.

PemNN contains two branches: the force trace branch $\mathcal{T}$ and the physics-based branch $\mathcal{P}$. To uniquely identify a block, we use the notation $(\mathcal{C}, pos)$, where $\mathcal{C} \in \{\mathcal{P}, \mathcal{T}, \mathcal{PT}\}$ specifies the channel with $\mathcal{PT}$ representing the fusion of two branches, and $pos \in \{pre-module, CNN1, BN1, RELU1, CNN2, fusion, GAP, \dots\}$ indicates the name. For example, $(\mathcal{T}, CNN1)$ identifies the first convolutional layer in the force trace branch. We use $f_{\mathcal{C}, pos}$ denotes the output of a block. PemNN supports two fusion strategies—late fusion and early fusion—which differ based on the dimensionality of the features being fused.

Early fusion module is applied to fuse outputs of layers $(\mathcal{T}, RELU1)$ and $(\mathcal{P}, RELU1)$ (Figure 3.2). We discuss four different methods as follows:

- Early-Sum:

$$f_{Early-fusion}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) = f_{\mathcal{T}, RELU1}(\mathcal{F}^{(i)}) + f_{\mathcal{P}, RELU1}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) \quad (3.8)$$

which sums convolutional feature maps across branches.

- Early-Max:

$$f_{Early-fusion}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) = \max \left\{ f_{\mathcal{T}, RELU1}(\mathcal{F}^{(i)}), f_{\mathcal{P}, RELU1}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) \right\} \quad (3.9)$$

where the maximum value at each element is selected.

- Early-Wavg:

$$f_{Early-fusion}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) = \alpha_{\mathcal{T}} f_{\mathcal{T}, RELU1}(\mathcal{F}^{(i)}) + \alpha_{\mathcal{P}} f_{\mathcal{P}, RELU2}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) \quad (3.10)$$

$$\text{subject to } \alpha_{\mathcal{T}} + \alpha_{\mathcal{P}} = 1$$

which computes a weighted sum of feature maps with coefficients $\alpha_{\mathcal{T}}, \alpha_{\mathcal{P}}$ learned during optimization.

- **Early-Conv:** The three fusion strategies described above require that the outputs of the $(\mathcal{T}, \text{RELU1})$ and $(\mathcal{P}, \text{RELU1})$ layers have consistent shapes. For each spatial location (i, j) , the early fusion output is computed using the corresponding values from both branches, but without any interaction between them. That is, the computation at each location only combines the transformer-derived and protein-derived features at that same location. To enable cross-branch information exchange, we introduce a convolutional operation termed *Early-Conv*. This approach applies a set of shared 1×1 convolution filters, denoted $\omega^{\text{Early-conv}}$, to the concatenated feature maps from both branches, allowing local fusion of features across branches while maintaining spatial alignment.

$$f_{\text{Early-fusion}}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) = \left(f_{\mathcal{T}, \text{RELU1}}(\mathcal{F}^{(i)}) \| f_{\mathcal{P}, \text{RELU1}}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) \right) \otimes \omega^{\text{Early-conv}} \quad (3.11)$$

where $\otimes$ is the convolution operator, and the vertical concatenating operator $\|$ is defined as $v_i \| v_j = (v_{i0}, v_{i1}, \dots, v_{in}, v_{j0}, \dots, v_{jn})^T$ given $v_i = (v_{i0}, v_{i1}, \dots, v_{in})^T$ and $v_j = (v_{j0}, v_{j1}, \dots, v_{jn})^T$.

Late-fusion strategy, where the fusion module is applied after the GAP layers, as depicted in Figure 3.3. Two methods are considered:

- **Late-Concat:**

$$f_{\text{Late-fusion}}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) = f_{\mathcal{T}, \text{GAP}}(\mathcal{F}^{(i)}) \| f_{\mathcal{P}, \text{GAP}}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)})$$

where GAP layers are concatenated directly.

- **Late-Gating:**

$$f_{\text{Late-fusion}}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) = \left(g_{\mathcal{T}} f_{\mathcal{T}, \text{GAP}}(\mathcal{F}^{(i)}) \right) \| \left(g_{\mathcal{P}} f_{\mathcal{P}, \text{GAP}}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) \right)$$

where GAP layers are concatenated with learnable weights. The gating weights $g_{\mathcal{T}}, g_{\mathcal{P}}$ are calculated as

$$(g_{\mathcal{T}}, g_{\mathcal{P}}) = \sigma \left(W \cdot \left(f_{\mathcal{T}, \text{GAP}}(\mathcal{F}^{(i)}) \| f_{\mathcal{P}, \text{GAP}}(\mathcal{F}^{(i)}, \mathcal{X}^{(i)}) \right) + b \right),$$

where W and b are the weights and bias of a fully connected layer and $\sigma(\cdot)$ is softmax activation function.

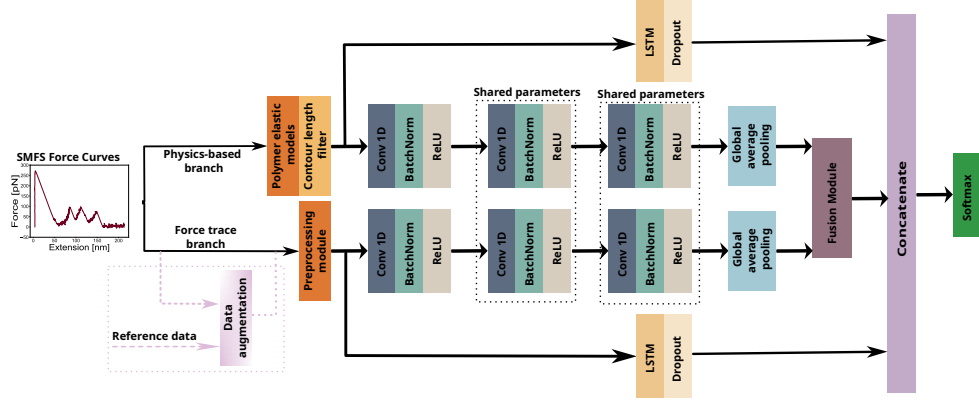


Figure 3.3: Overall architecture of the PemNN model with late fusion strategy. The data augmentation block is not enabled by default but is utilized through the physics-guided pretraining strategy described in Section 3.8.4.

In multivariate time series classification, the widely used approach is late fusion, which directly concatenates features extracted from Global Average Pooling (GAP) layers. For example, [68] concatenated the GAP layer of a convolutional block with output of an LSTM layer, [71] concatenated GAP layers of convolutional blocks originated from different input branches, [70] integrated both LSTM and GAP features from different branches, and [86] not only concatenated features from two transformer encoders but also introduced a gating mechanism to assign weights to these features. Unlike late fusion, early fusion focuses on combining convolutional feature maps from multiple branches. Examples includes [87], which employed an element-level maximum operator for feature fusion in image classification; [88], which summed feature maps for image detection; and [69], which concatenated feature maps extracted with different kernel sizes for anomaly detection.

3.5 Datasets

Datasets from four different protein molecules are employed. Out of these four protein molecules, only Titin I27O (Athena Enzyme SystemsTM) is an engineered protein molecule composed of eight repeats of the Ig 27 domain that serves as a reference protein for calibrating and validating our methods. The other three protein molecules

come from natural proteins with considerable variations in their sequence and structure, dystrophin and utrophin. Dystrophin is a protein molecule expressed primarily at the muscle cell membrane, or sarcolemma, in striated muscle tissue. Deficiencies of this protein molecule lead to severe muscle wasting disorder like Duchenne muscular dystrophy (DMD), a fatal disease occurring in 1 out of 4000 male births [13]. Structurally, dystrophin is composed of four major domains: an amino terminal (NT) actin-binding domain (ABD1), a large central rod domain with 24 triple helical spectrin-like repeats (SLRs) interspersed with 4 hinge domains, including a second actin-binding domain (ABD2), a cysteine-rich domain and a carboxy-terminal (CT) domain (Figure 4.3a). Utrophin (Figure 4.3b) is a fetal homologue of dystrophin and is under active investigation as a dystrophin replacement therapy for DMD. We include dystrophin and utrophin fragments encoding the NT through SLR 3 domains, referred to as DysN-R3 (Fig. 4.3c) and UtrN-R3 (Fig. 4.3d), respectively. Previous studies have demonstrated that the mechanical properties of UtrN-R3 are influenced by the expression system used, such as insect or bacterial cells [3, 58]. Consequently, we further categorize UtrN-R3 into insect UtrN-R3 and bact UtrN-R3 to reflect these variations. In summary, our four real protein molecules are: Titin I27O, bact UtrN-R3, insect UtrN-R3, DysN-R3. Additionally, we include the Discoidin Domain Receptors (DDRs) dataset (Appendix 3.8.2) [60] to compare the performance of different deep learning methods.

Table 3.1: Details of experimental datasets

DATASET	NUMBER OF CURVES PER CLASS	LENGTHS	DIFFERENT DAYS	PULLING SPEEDS [nm/s]
DDRs [60]	[102,102,136]	400	NA	2000
TITIN I27O	[181,164,191]	736-4859	5	500, 1000, 2000
BACT UTRN-R3 [58]	[181,181,178]	957-9974	10	500, 1000, 2000, 5000
INSECT UTRN-R3 [58]	[175,166,200]	1777-8890	11	500, 1000, 2000
DYSN-R3 [58]	[191,185,177]	1659-4814	6	500, 1000, 2000

For each protein molecule, we have two datasets, one is simulation dataset generated with our physics-based Monte Carlo simulation engine described in Section 3.5.2, and one is experimental dataset that is obtained from physical experiments conducted via

AFM on real protein molecule samples.

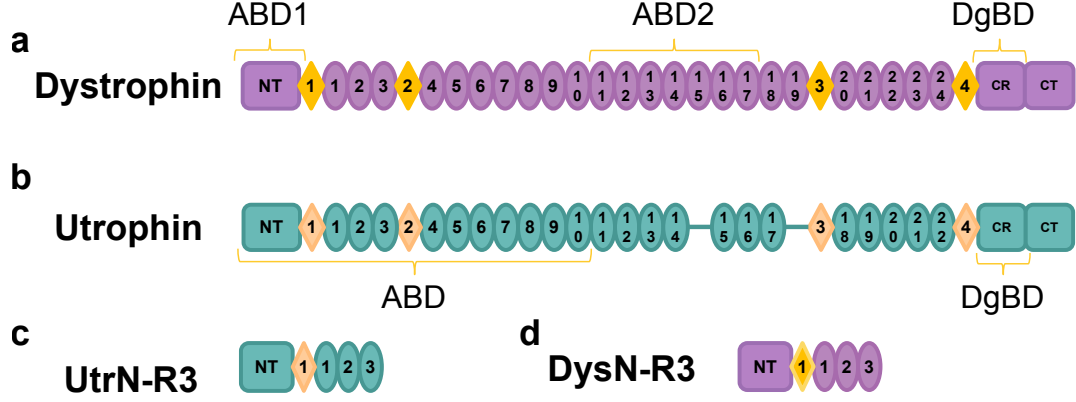


Figure 3.4: Diagrams of dystrophin and utrophin. (a) Full-length dystrophin. (b) Full-length utrophin. (c) UtrN-R3 construct (d) DysN-R3 construct. The ovals are spectrin-like repeats (SLRs); the diamonds are hinge domains; NT is the N terminus; CT is the C terminus; CR is a cysteine-rich domain; ABD1 & 2 are actin-binding domains; DgBD is the dystroglycan binding domain. Figure courtesy of previous study [3]

3.5.1 Experimental Dataset

For the experimental data, as previously described in [10, 3, 58], force curves were collected on different days and at various pulling speeds, as outlined in Table 3.1. As a result, the lengths of the force curves are different, and substantial variations exist even within the same class. Annotation is performed through visual inspection of the unfolding force curves.

3.5.2 Simulation dataset

The simulation dataset, comprising 600 force curves, is generated for three classes; Class 0 has no protein molecule between the substrate and cantilever tip, Class 1 has one protein molecule, and Class 2 has two protein molecules attached. Different protein

molecules are distinguished by different model parameters listed in Table 3.2.

Algorithm 2 Monte Carlo Simulation of Multiple Molecules

Require: $v, L_c^{(i)}, L_p^{(i)}, \Delta L_c^{(i)}, \Delta L_p^{(i)}, k_0, \Delta x^\ddagger, \Delta G^\ddagger, N, D^{(i)}$

```

1: Initialization:  $z \leftarrow 0, U^{(i)} \leftarrow 0$ 
2: for  $t \leftarrow 0 : \Delta t : T$  do
3:    $z \leftarrow z + v\Delta t$ 
4:   Solve (4.5) for extension  $x$ 
5:   for  $i \leftarrow 1 : 1 : N$  do
6:     Calculate  $F_{WLC}^{(i)}$  using (3.13)
7:     Compute  $k_{off}(F_{WLC}^{(i)})$  using (4.7)
8:     Compute  $P_u^{(i)}(F_{WLC}^{(i)})$  using (4.6)
9:     Draw  $\eta^{(i)} \sim \mathcal{U}_{[0,1]}$ 
10:    if  $\eta^{(i)} < P_u^{(i)}(F_{WLC}^{(i)})$  then
11:       $L_c^{(i)} \leftarrow L_c^{(i)} + \Delta L_c, L_p^{(i)} \leftarrow L_p^{(i)} + \Delta L_p$ 
12:       $U^{(i)} \leftarrow U^{(i)} + 1$ 
13:    end if
14:    Draw  $\eta_d^{(i)} \sim \mathcal{U}_{[0,1]}$ 
15:    if  $U^{(i)} == D^{(i)}$  and  $\eta_d^{(i)} < P_d(F_{WLC}^{(i)})$  then
16:       $L_c^{(i)} \leftarrow L_c^{(i)} + C_{Lc}, L_p^{(i)} \leftarrow L_p^{(i)} + C_{Lp}$ 
17:    end if
18:  end for
19: end for

```

Monte Carlo Simulation based methods are used widely in the SMFS studies, yielding results that closely align with experimental data [81, 89]. However, prior simulation frameworks are restricted to the idealized case of force curves arising from single molecule, as what we previously described in Section 2.3.2. To build a comprehensive training dataset, we incorporate the real cases of force curves generated by no molecule and multiple molecules. Additionally, we model the cases where only partial sections of molecules are present as well as the stochastic adhesion and detachment events, of the cantilever to the protein, to better approximate experimental data.

For the simulations, N proteins are considered with i -th protein having $D^{(i)}$ folded

domains attached between the substrate and the force probe. The base of the cantilever probe is moved away at a constant speed v ; here the position of the base of the cantilever z is initialized at zero and is updated every Δt seconds. The protein extension x is determined by solving the equation,

$$k_c d = \sum_{i=1}^N F_{WLC}^{(i)}(x, L_c^{(i)}, L_p^{(i)}), \quad (3.12)$$

where k_c is the spring constant of the cantilever, and d is the deflection (Fig 4.1a). Here, $F_{WLC}^{(i)}(x, L_c^{(i)}, L_p^{(i)})$ is the worm-like chain (WLC) model that relates the force applied to the extension of i -th protein (See Section 3.4.1), given by [39],

$$F_{WLC}^{(i)}(x, L_c^{(i)}, L_p^{(i)}) := \frac{k_B T}{L_p^{(i)}} \left[\frac{1}{4 \left(1 - \frac{x}{L_c^{(i)}}\right)^2} - \frac{1}{4} + \frac{x}{L_c^{(i)}} \right], \quad (3.13)$$

where k_B is the Boltzmann constant, T is temperature, and $L_c^{(i)}$ and $L_p^{(i)}$ are the contour length and the persistence length of the i -th protein, respectively. For the i -th protein, the probability of a domain unfolding during the time interval Δt is found by

$$P_u^{(i)}(F_{WLC}^{(i)}) = (D^{(i)} - U^{(i)})(1 - e^{-k_{off}(F_{WLC}^{(i)})\Delta t}), \quad (3.14)$$

where $U^{(i)}$ is the number of unfolded domains which is initially set to zero, and the transition rate $k_{off}(F_{WLC}^{(i)})$ is computed using the Dudko–Hummer–Szabo model [34]; see Section 2.3.2 for details.

For determining unfolding events of i -th protein, a random number $\eta^{(i)}$ is generated uniformly from 0 to 1 and is compared to the unfolding probability $P_u^{(i)}(F_{WLC}^{(i)})$. No unfolding event is triggered if the random number is larger than $P_u^{(i)}(F_{WLC}^{(i)})$; the simulation will continue to the next time slot by adding time interval Δt . Otherwise, one of the domains is unfolded, leading to a increase in the number of unfolded domains, $U^{(i)}$, by 1, and the simulation continues to the next unfolding event if folded domains still exist after updating contour length and persistence length via adding increments $\Delta L_c^{(i)}$, $\Delta L_p^{(i)}$ respectively.

Once all domains in i -th protein unfold, the protein detaches from either the cantilever tip or substrate based on the detachment probability,

$$P_d(F) = \begin{cases} C_d & F \geq F_{td} \\ 0 & F < F_{td} \end{cases}, \quad (3.15)$$

where C_d is a constant and F_{td} is a random number sampled from the Gaussian distribution, denoting the threshold at which the detection of detachment begins. Upon detachment of the i -th protein, its WLC force is reduced to zero by adding large constants (C_{Lc} and C_{Lp}) to contour length $L_c^{(i)}$ and persistence length $L_p^{(i)}$ respectively. To better replicate experimental force curves, we introduce Gaussian noise to the WLC force immediately after its calculation at line 5 of Algorithm 2. The adhesion force is added at the end of the simulation process.

Table 3.2: The DHS model parameters of four real protein molecules are: Titin I27O, bact UtrN-R3, insect UtrN-R3, DysN-R3

MOLECULES	DHS $\nu = 1/2$		
	$\ln(k_0)$	$\Delta x^\ddagger$ [nm]	$\Delta G^\ddagger$ [$k_B T$]
TITIN I27O [1]	-9.21	0.40	20.00
BACT UTRN-R3 [58]	-6.50	0.85	14.50
BACT DYSN-R3 [58]	-4.50	0.60	11.50
INSECT UTRN-R3 [58]	-2.50	0.41	9.80

The adhesive force can be composed of various components like van der Waals force, capillary force, and chemical forces, which depend on environmental conditions such as roughness, interacting angles, and wetness [65, 90]. However, quantifying these environmental conditions is challenging, and they can vary significantly between experiments. Consequently, we adopt a straightforward yet versatile method to model adhesive force

rather than a more intricate approach,

$$F_a(t) = \begin{cases} \frac{t}{t_1} F_{ad} & 0 < t < t_1 \\ \frac{t-t_2}{t_2-t_1} F_{ad} & t_1 < t < t_2, \\ 0 & else \end{cases} \quad (3.16)$$

where the adhesive force increases linearly in the interval $[0, t_1]$, reaching the adhesive force threshold F_{ad} at t_1 , then adhesion between the cantilever tip and the substrate begins to disconnect at t_1 and vanishes at t_2 . The vanishing phase $[t_1, t_2]$ should be much faster than the adhesive phase $[0, t_1]$, with a common choice being to set t_2 close to t_1 , for example, $t_2 = 1.1t_1$. To introduce stochasticity and enhance generality, we assume both F_{ad} and t_1 to be Gaussian distributed random variables with user-specified mean and standard deviation.

3.6 Results

The performance of our proposed model, PemNN, is evaluated in this section. We outline the evaluation setup in Section 3.6.1, followed by a comparison of PemNN’s classification performance against baseline models in Section 3.6.2. Next, we analyze the functionality of the force trace branch and physics-based branch in Section 3.6.3. Finally, we assess performance under limited training data, a scenario frequently encountered in SMFS applications, in Section 3.6.5, and apply it to AFM data analysis through a comparison to non-machine learning methods in Section 3.6.6.

3.6.1 Evaluation Setup

We evaluate our proposed model, PemNN, against five baselines: 1) Triplet network (Triplet) [60]: The model designed for analyzing SMFS specific pulling data. 2) Fully convolutional neural networks (FCN) and 3) the residual networks (ResNet): Two of the highest performing deep neural networks on the UCR time series classification archive [5]. 4) InceptionTime: The current state-of-the-art deep learning model on the UCR archive [91]. 5) LSTMFCN: It outperforms FCN and ResNet on the UCR time series classification archive and demonstrates robust performance on multivariate time series classification [92, 68]. Further details about these baselines are provided below.

Triplet The triplet network [59] takes three samples: the anchor x , the positive sample x^+ , and the negative sample x^- . Here, x is the sample under classification, x^+ comes from the same class as x , and x^- is of different class. All three samples are passed through the same network architecture, where the weights are shared, to learn representations in the embedding space. The triplet loss is employed,

$$L_{triplet} = \max\{0, ||Net(x^+) - Net(x)||_2^2 - ||Net(x^-) - Net(x)||_2^2 + m\}, \quad (3.17)$$

where $||\cdot||$ denotes the L_2 norm, and m represents the margin parameter that controls the separation between positive and negative samples in the embedding space. The objective is to minimize the distance between the anchor and the positive sample while maximizing the distance between the anchor and the negative sample in the embedding space. Subsequently, the resulting embeddings are fed into the MLP described earlier to learn classification. The embedding network adopts the ResNet structure, which will be elaborated on below.

FCN Fully convolutional neural networks (FCN) [63] is composed of three convolutional blocks followed by a Global Average Pooling (GAP) layer and a softmax layer. Each convolutional block includes a 1D convolutional layer, batch normalization, and a Rectified Linear Unit (RELU) activation layer. The output of the final convolutional block is passed through a GAP layer, followed by a fully connected softmax layer. A stride of 1 with zero padding is used to preserve the length of time series data. The filter sizes and kernel sizes of the three convolutional layers are $\{128, 256, 128\}$ and $\{8, 5, 3\}$, respectively.

ResNet The ResNet [93] extends neural networks to very deep structures by incorporating shortcut connections. These connections enable the gradient flow directly through the network, easing the training of deeper models. Our Residual Network (ResNet) comprises three residual blocks followed by a GAP layer and a softmax layer. Each residual block contains three convolutional blocks similar to those in the FCN architecture. The output of the last convolutional block is added to the input of the residual block before proceeding to the next layer. The kernel size for the three convolutional layers in each residual block remains consistent with FCN architecture, set as $\{8, 5, 3\}$. The

number of filters is the same within the same residual block, with filter sizes specified as $\{64, 128, 128\}$ for three residual blocks, respectively.

InceptionTime InceptionTime [91] ensembles the predictions of several InceptionNet models, each with different initializations, to overcome the high variance across different runs. The InceptionNet model consists of two residual blocks, followed by a global average pooling (GAP) layer and a softmax layer. Within each residual block are three Inception modules. Each Inception module begins with a bottleneck layer that reduces dimensionality while preserving the original sequence length through a kernel size and stride of 1. Subsequently, 1D convolutions with varying kernel sizes are applied to the bottleneck output, capturing patterns across different scales. Additionally, a max pooling layer, followed by another bottleneck layer, is applied to the original time series.

LSTMFCN LSTMFCN [68] employs the augmentation an LSTM layer and CNN layers along with a squeeze-and-excitation block to generate feature maps, followed by a softmax layer for time series classification. The three convolutional layers in LSTMFCN are configured with filter sizes of $\{128, 256, 128\}$ and kernel sizes of $\{8, 5, 3\}$, respectively. The LSTM layer has a hidden dimension of 8, followed by a dropout layer with a rate of 0.8.

Implementation Details The baselines were implemented using default parameters from sktime [94], which is a python framework for ML with time series data. For PemNN, the default configuration incorporates the WLC model, LSTM layers in both branches, and Early-Conv fusion method. In PemNN, the CNN blocks contains $\{128, 256, 128\}$ filters with kernel sizes of $\{8, 5, 3\}$, respectively; the LSTM layers are configured with a hidden dimension of 8, followed by a dropout layer with a rate of 0.8 to mitigate overfitting. The contour length threshold is set to 1000 nm, the persistence length $L_p = 300$ pm, Kuhn length $L_k = 300$ pm, and bond length $L_b = 150$ pm by default. The batch size is 16 for all models except for InceptionTime with 64, and the number of training epochs are kept at 200 epochs for all models. All models are trained with Adam [95] with the learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e - 8$. The best-performing model, determined by achieving the lowest training loss, is then selected and evaluated on the testing data. These models are trained with Apple M1 Pro,

which has 10-core CPU and 16-core GPU. An ablation study of PemNN’s architecture is presented in Section 3.6.4.

For each dataset, 20% of the data was used for training, and the remaining 80% was reserved for testing, as adjudication is a time-consuming task that requires significant expertise. Stratified resampling was applied to maintain class distributions. Each train-test split was seeded for reproducibility. Identical resamples were applied to all models within a single run, and performance was evaluated using overall accuracy of the test data across five runs.

3.6.2 PemNN Classification Performance

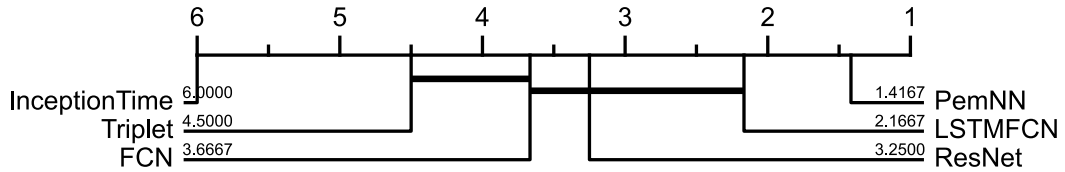


Figure 3.5: Critical difference diagram of different deep learning models across the simulated and experimental testing sets of all protein molecules based on average accuracies. The most accurate model is assigned a rank of 1, with a thick horizontal line representing a group of classifiers that do not exhibit statistically significant differences in accuracy.

Per each dataset, we evaluated all models on testing data and ranked them based on their mean classification accuracy over five runs, assigning a rank of 1 to the most accurate model and 6 to the least accurate. The average ranking is then computed across all datasets, including both simulated and experimental testing sets for all protein molecules. These average rankings are summarized in the critical difference diagram [4], as presented in Figure 3.5. PemNN achieves the lowest ranking of 1.4167, indicating that our model is more accurate than baseline models.

For statistical analysis, we employed the Wilcoxon signed-rank test with Holm correction as the post-hoc test following the Friedman test [5, 4]. In Figure 3.5, thick

horizontal lines represent groups of models that are not significantly different in terms of classification accuracy. Thus we conclude that PemNN is significantly more accurate than all baseline models. Among the baselines, LSTMFCN has the lowest rank, with its performance statistically similar to ResNet and FCN, but significantly different from Triplet and InceptionTime.

3.6.3 Functionality of Two Branches

This section evaluates the functionality of the force trace and physics-based branches in PemNN using simulated datasets with known ground truths.

Physics-based branch functionality The importance of the physics-based branch, which integrates polymer elastic models, was evaluated by comparing PemNN (using both force trace and physics-based branches) to baselines that only utilized the force trace branch. As shown in Figure 3.6a, PemNN achieves near perfect accuracy (98.9 ± 1.9 %) across four simulated datasets, significantly outperforming the baseline. The largest performance gap is in the Titin I27O simulated dataset, where PemNN outperformed baselines by 15%; the smallest gap occurs in the insect UtrN-R3 dataset, with PemNN maintaining a 6% higher accuracy.

Force trace branch Functionality Baselines were provided with the same input data as the physics-based branch to assess the contribution of the force trace branch. This was analyzed under two scenarios: one is changing contour length threshold in the filter, the other one is varying the persistence length L_p in the WLC model. Radar charts were employed to visualize model performance, with models represented by different transparent colors and higher radii indicating better accuracy. In Figure 3.6b, the contour length threshold is changed from 1,000 to 1,024,000 nm in multiples of four, with "inf" denoting no threshold. PemNN maintained near-perfect accuracy for thresholds between 1,000 and 64,000 nm, a range in which some baselines, such as LSTMFCN and ResNet, also perform well. However, as the threshold increases, baseline performance dropped sharply. In contrast, PemNN maintains robust performance, outperforming the baselines by at least 7% at a threshold of 1,024,000 nm, with this margin growing to 25% when no threshold was applied. In Figure 3.6c, the persistence length L_p is varied logarithmically from 0.3 to 30,000 pm. Both PemNN and baselines achieved near-perfect accuracy for L_p values between 30 and 30,000 pm. However, at $L_p = 0.3$ pm and $L_p = 3$

pm, baseline performance dropped significantly, whereas PemNN maintained at least 3% and 10% higher accuracy, respectively.

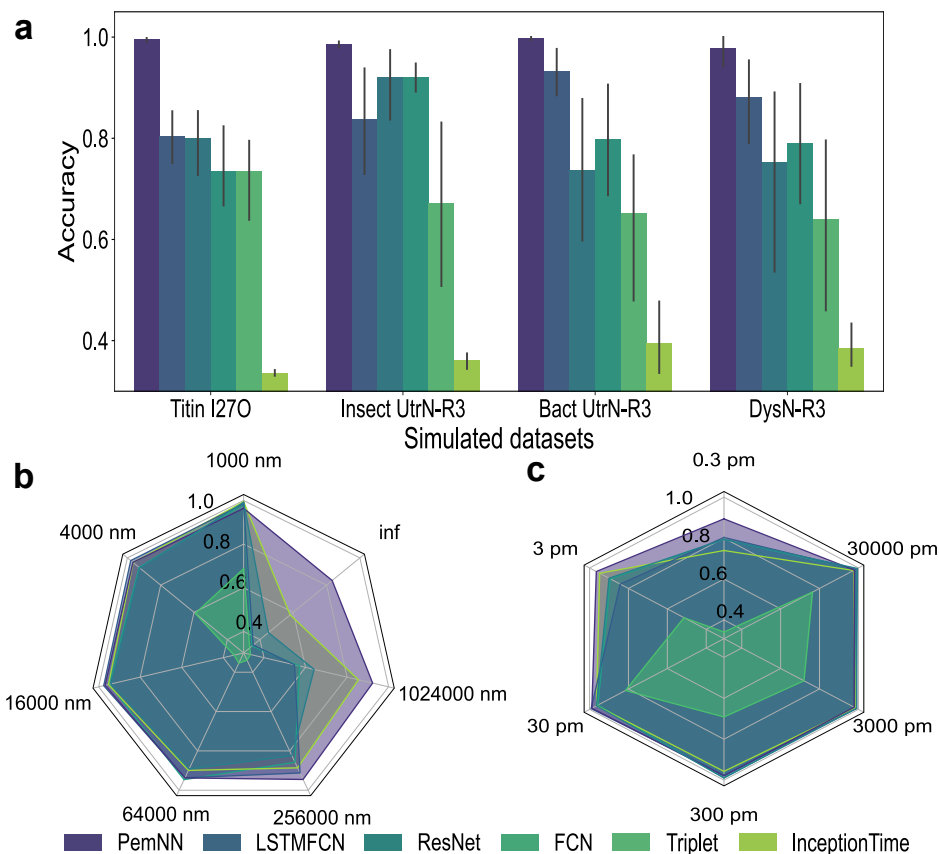


Figure 3.6: a) The average accuracy of all models across the four simulated datasets, with error bars indicating the standard deviations over five runs. b) The average accuracy across four simulated datasets over five runs as the contour length threshold varies, depicted in a radar chart where longer radii indicate higher accuracy. c) The average accuracy across four simulated datasets over five runs as the persistence length changes.

Contour lengths typically span the micrometer scale. However, factors such as signal noise or deviations from the polymer elastic models can lead to exaggerated contour length values, potentially degrading model performance. Despite these challenges,

PemNN exhibits robust performance even when datasets include such extreme values.

The persistence length L_p quantifies the bending stiffness of a polymer. When L_p is underestimated, the polymer appears excessively flexible, causing the contour length L_c to be overestimated. Conversely, when L_p is overestimated, the polymer appears stiffer, and L_c approaches the extension x . Figure 3.7 illustrates the relationship between L_p and L_c using the WLC model applied to a force curve from the Titin I27O dataset. At $L_p = 300$, contour length L_c remains consistent within unfolding events, indicating a correctly chosen L_p . Underestimated L_p (0.3-30) results in large and inconsistent L_c , leading to performance degradation. When L_p is overestimated (3000, 30000), L_c becomes linear with x , converging toward an identity function (red line in the inset of Figure 3.7).

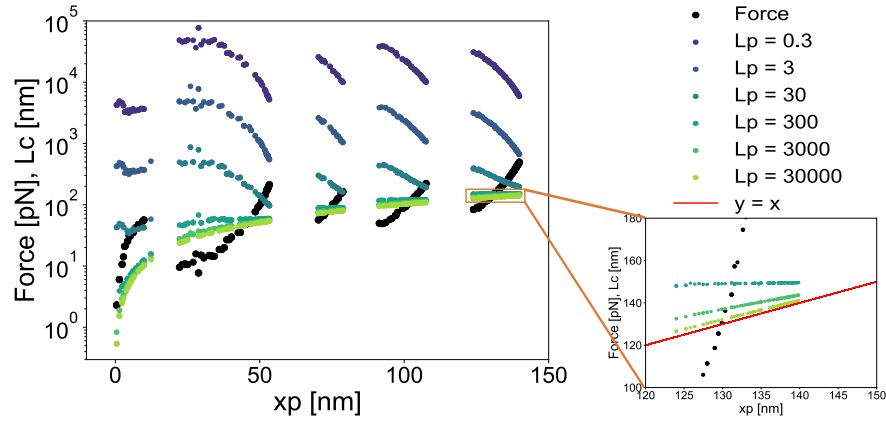


Figure 3.7: Contour length calculations using the WLC model with varying persistence lengths (L_p), visualized with a simulated force curve of Titin I27O.

In conclusion, the physics-based branch enhances performance by incorporating polymer elastic models; the force trace branch increases robustness, ensuring reliable classification even when physics models are corrupted by parameter errors. By combining the strengths of both branches, PemNN consistently outperformed baselines in SMFS classification tasks, demonstrating its effectiveness under various conditions.

3.6.4 Ablation Study of Model Architecture

In this section, we explore the architectures of our model, PemNN, in greater detail to provide insights into optimizing model designs for SMFS classification. We examine 1) different fusion strategies, 2) the inclusion or exclusion of LSTM layers, and 3) different polymer elastic models.

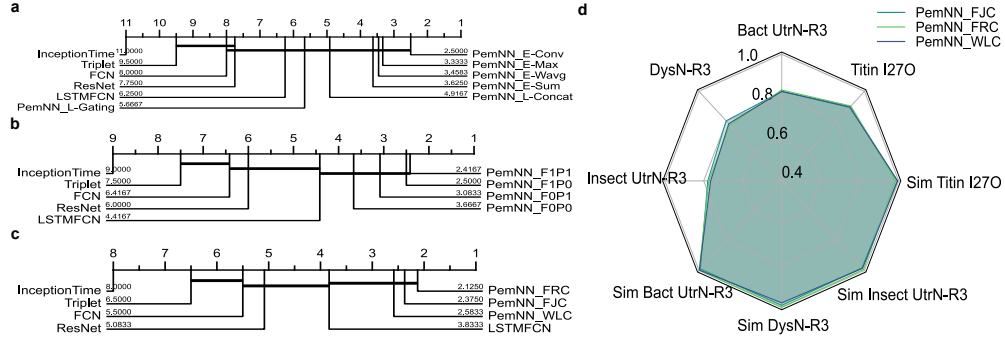


Figure 3.8: Ablation study of PemNN architecture. a) Critical diagram comparing various fusion strategies, where 'E-' and 'L-' denote early fusion and late fusion strategies, respectively. b) Critical diagram evaluating the incorporation of LSTM layers, with 'F1P1' indicating LSTM layers included in both the force trace and physics-based branches. c) Critical diagram of different polymer elastic models. d) Performance of PemNN across datasets with three distinct polymer elastic models.

Fusion Strategies Figure 3.8a presents results of various fusion strategies. Our model, PemNN, consistently outperforms the baselines regardless of the fusion strategy used. Moreover, all early fusion methods exhibit superior performance compared to late fusion methods. For late fusion methods, the GAP layers from the last convolutional blocks are concatenated, leading to significant information loss since each element in the GAP layer represents a single channel from the convolutional block. In contrast, early fusion methods directly integrate branches from first convolutional blocks, preserving more information. Among the early fusion strategies, the Early-Conv strategy stands out, likely due to its use of filters to effectively process information across branches.

LSTM Layers Figure 3.8b shows the impact of incorporating LSTM layers into the two branches of PemNN. Regardless of the specific use of LSTM layers, PemNN consistently outperforms the baselines, with all models showing statistically significant improvements over the baselines, except for LSTMFCN. Including LSTM layers in both the trace and physics-based branches achieves the best overall performance, with the average ranking of 2.4167.

3.6.5 Performance with Limited Data

In this section, we focus on experimental datasets and analyze model performance under limited training data. The testing data is fixed at 80% of the entire dataset, while the deep learning models are trained using varying proportions of the training dataset. When the training proportion ranges from 5% to 20% (approximately 30 to 120 samples), PemNN consistently outperforms baselines, achieving accuracies of 79.6% and 85.2% at training proportions of 5% and 20%, respectively (Figure 3.9). Among the baselines, LSTMFCN performs the best but remains 11.4% and 2.9% less accuracy than PemNN at training proportions of 5% and 20%, respectively.

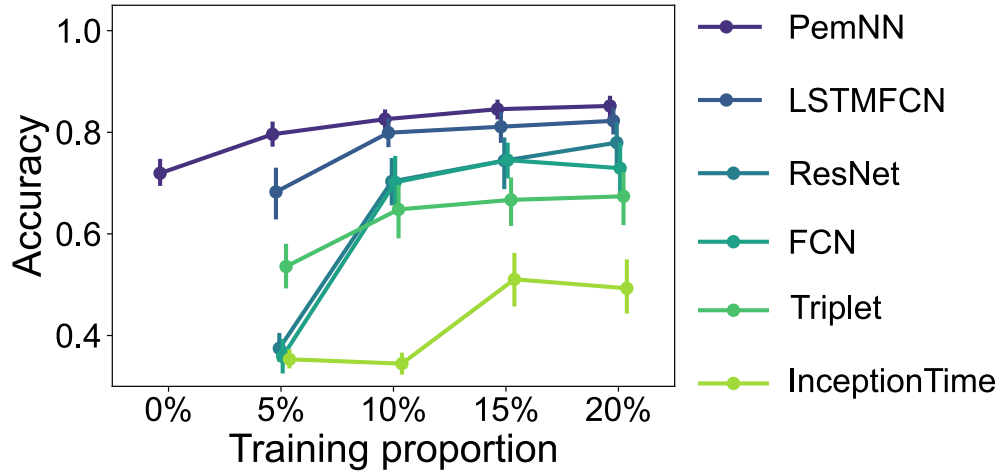


Figure 3.9: The performance of models trained with different proportions of experimental datasets (training proportion), with error bars indicating standard deviations across all experimental datasets over five runs.

At a training proportion of 0%, no experimental data is used during the training. Instead, a physics-guided pretraining strategy (see Section 3.8.4) is used to eliminate the need for experimental labeling and mitigate human bias. In this approach, deep learning models are pre-trained on simulated datasets generated using physics-based models and subsequently evaluating them on corresponding experimental datasets. The pretraining strategy achieves comparable performance to models directly trained on experimental data, with an average accuracy of 72.0%, despite not utilizing any experimental data during its training. In our physics-based protein molecule unfolding model, we assume every protein domain behaves identically. However, many protein molecules, including utrophin and dystrophin have folded domains that are significantly different from each other. Despite being trained on simulation data using a single double-well potential model for the domains, *PemNN* demonstrates the capability to classify the number of protein molecules involved in experiments with protein molecules exhibiting heterogeneous domains (See Section 3.8.4 for details).

3.6.6 Application to SMFS Data Analysis

Here, we first apply both machine learning and non-machine learning methods to Titin I27O data collected from a one-day experiment (~ 3000 curves). Titin I27O is a well-calibrated protein molecule with a most probable unfolding force of 204 ± 26 pN. In Figure 3.10, *RawData* includes all unfolding events. The *Heuristic* method, a non-machine learning approach, filters data with the WLC model [10, 3] that requires manual adjustments by experts to fine-tune parameters. *PemNN* was trained on our Titin I27O dataset and subsequently tested on the raw data to extract curves originating from single molecule. Notably, *PemNN* takes less than an hour to compute statistics, while the *Heuristic* method takes several hours, and *RawData* can take up to a full day.

PemNN achieves a most probable unfolding force of 206.68 pN, closely aligning with the expected value (204 ± 26 pN). In contrast, the *Heuristic* and *RawData* yield most probable unfolding forces of 217.41 pN and 192.21 pN, respectively (Table 3.3). Furthermore, the inclusion of data not originating from single-molecule events resulted in broader force distributions. We quantified the sharpness of these distributions using the interquartile range (IQR), as listed in Table 3.3. Our method, *PemNN*, achieved an IQR of 52.63, which is only a quarter of the IQR observed from the *Heuristic* or

the *RawData*, effectively filtering out confounding factors. These results highlight that *PemNN* effectively analyzes AFM data by accurately capturing key statistical features while effectively filtering out confounding factors from multiple molecules.

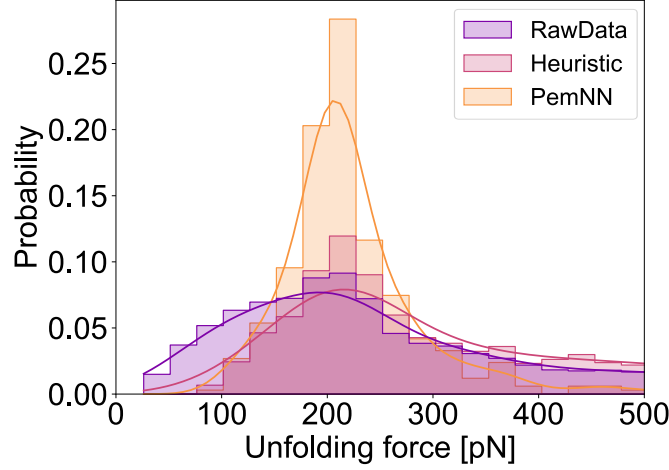


Figure 3.10: Application to SMFS data analysis of Titin I27O: *RawData* as the Baseline Method, *Heuristic* as non-machine learning method, and *PemNN* as the our proposed model.

We then provide additional details for utrophin and dystrophin molecules, as illustrated in Figure 3.11. Unfolding force distributions are visualized using violin plots, where black stars indicate the values with the highest probability. The violins depict data distributions with kernel density estimation (shown as black lines on either side), and the width of each curve represents the relative frequency of data points. For each protein molecule, Titin I27O, bact UtrN-R3, insect UtrN-R3, and DysN-R3, the unfolding distributions generated by *PemNN* are significantly more concentrated compared to those from *RawData* and *Heuristic*, as quantified by the interquartile range (IQR) listed in Table 3.3. Furthermore, the most probable forces obtained with *PemNN* differ from those achieved by the *Heuristic* method by no more than 10 pN. These results highlight that *PemNN* effectively analyzes AFM data by accurately capturing key statistical features while effectively filtering out confounding factors from multiple molecules.

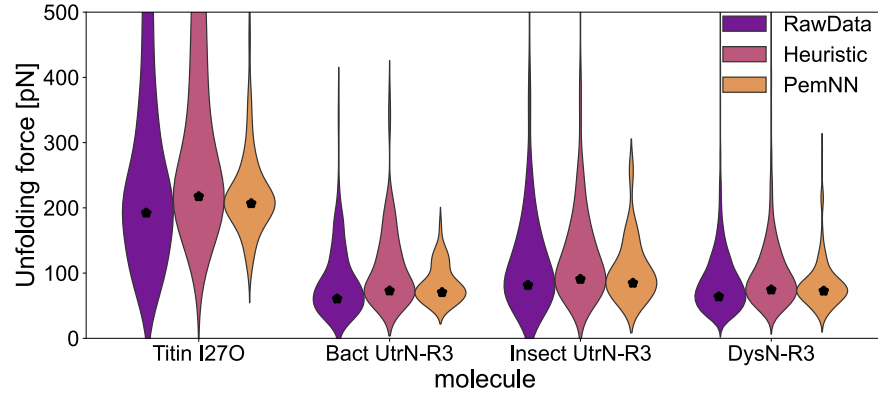


Figure 3.11: Application of PemNN to AFM Data Analysis: *RawData* as the Baseline Method, *Heuristic* as non machine learning method, and *PemNN* as the our proposed model. Unfolding force distributions are depicted using violin plots, where black stars signify the positions of values with the highest probability, with values detailed in Table 3.3. The violins illustrate data distributions using kernel density estimation as black lines on each side, with the width of each curve indicating the relative frequency of data points.

Table 3.3: The most probable forces (in pN), along with the interquartile range (IQR) in parentheses, for all four protein molecules analyzed using different methods during AFM data analysis.

TITIN I27O			BACT UTRN-R3		
RAWDATA	HEURISTIC	PEMNN	RAWDATA	HEURISTIC	PEMNN
192.21(234.52)	217.41(270.46)	206.68(52.63)	60.50(63.22)	72.84(66.55)	70.40(36.97)
INSECT UTRN-R3			DYSN-R3		
RAWDATA	HEURISTIC	PEMNN	RAWDATA	HEURISTIC	PEMNN
81.41(85.29)	90.58(80.57)	84.66(39.98)	63.76(53.18)	74.38(53.10)	72.54(26.79)

3.7 Conclusions

Single-molecule force spectroscopy (SMFS) data of protein molecules are time and resource intensive to collect. Currently, manual visual inspection remains the primary method for classifying force curves resulting from single molecules. This process typically requires a full day per experimental iteration of a specific molecule, with multiple iterations being standard practice in the field. These factors make it challenging to obtain precise statistics of single molecular force curves and to generate a large, annotated dataset appropriate for training deep learning models.

We developed a novel deep learning model that fuses a physics model based branch with another that does not employ physics to efficiently classify SMFS data as originating from no molecules, single molecules, or multiple molecules. We also applied state-of-the-art machine learning models (including ResNet, FCN, InceptionTime, and LSTMFCN) to SMFS data. Experimental datasets, obtained from four molecules (Titin I27O, bact UtrN-R3, insect UtrN-R3, and DysN-R3) are used to test and train the deep learning model. A Monte-Carlo engine, based on the physics of proteins, is developed and employed to provide simulated data. The presented approach achieves superior performance compared to state-of-the-art baseline methods on all simulated or experimental datasets. Remarkably, strong performance on experimental data even when trained solely on simulated data is corroborated. Furthermore, the model surpasses non-machine learning approaches in SMFS data analysis, demonstrating its effectiveness and reliability, while reducing processing time from a day to under an hour.

3.8 Supporting Information

3.8.1 Data Preprocessing

In both single molecule and multiple molecule cases, the force curve typically consists of two parts, separated by the detachment of the event. Our interest lies in the region before detachment, where unfolding events occur. Here, the force curves are trimmed to retain the region before detachment by identifying the detachment point. This trimming process is illustrated in the transition from Figure 3.12a to Figure 3.12b. In the force trace branch, data is resampled using linear interpolation to a default of 400 points,

followed by min-max normalization, $x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$, where x_{min} and x_{max} are minimum and maximum values of the data x . This ensures the models focus on learning shapes rather than magnitudes, as shown in the transition from Figure 3.12b to Figure 3.12c. In the physics-based branch, 400 samples are extracted from the trimmed force curve by default, and the contour length L_c is calculated using polymer elastic models, illustrated in Figure 3.12d. We utilize tools from the publicly available time series library Tslearn [96] to implement the two preprocessing steps: resampling and normalization.

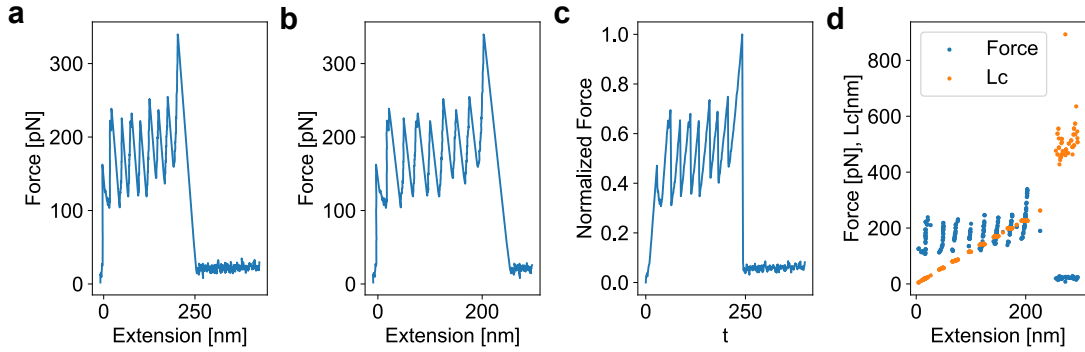


Figure 3.12: Data preprocessing example. (a) The original force curve. (b) The force curve after trimming. (c) The force data after normalization and resampling in the force trace branch. (d) The force curve with calculated L_c in the physics-based branch.

3.8.2 Results of the Discoidin Domain Receptors (DDRs) Dataset

The **DDRs** dataset contains unfolding curves of interaction forces between DDRs and its ligand, collagen, measured by AFM at the single molecule level [60]. The dataset contains three classes: 1) no molecule, 2) single molecule, and 3) multiple molecules. However, force curve collection was conducted at a single pulling speed, and the dataset is standardized to have equal lengths for each sample. However, our method, PemNN, is not applicable as extension data is not provided. Instead, we compare the performance of the remaining deep learning methods.

All deep learning models, except the Triplet model, were trained using 80% of the total dataset as the training dataset and tested on the remaining dataset. Their performance is compared to that of the triplet model reported in [60], as presented in Table

3.4. The dataset was pre-processed as proposed in [60], which involves applying a numerical first-order derivative, followed by a moving average filter with a window size of 13, and finally min-max normalization. Among the models, InceptionTime demonstrated the best performance, achieving approximately 20% higher overall accuracy and F1 score compared to the Triplet model.

Table 3.4: Overall accuracy, class accuracy, and F1-score of DDRs dataset for Triplet, MLP, FCN, ResNet, and InceptionTime are presented in the format of average (standard deviation) from 5 runs. Among these models, InceptionTime demonstrates superior performance compared to the others.

MODELS	OVERALL ACCURACY (%)	CLASS ACCURACY (%)			F1-SCORE(%)
		CLASS 0	CLASS 1	CLASS 2	
TRIPLET [60]	66.70	73.30	63.30	63.30	66.70
FCN	77.10(5.15)	81.22(4.96)	77.49(13.59)	73.48(10.06)	77.30(5.32)
RESNET	79.03(4.84)	81.90(10.52)	83.97(4.74)	70.92(7.44)	79.02(4.88)
LSTMFCN	75.81(3.42)	70.91(2.49)	88.24(4.16)	71.30(9.02)	76.17(3.32)
InceptionTime	85.48(1.14)	97.14(2.61)	84.00(8.94)	72.50(12.96)	85.26(1.02)

3.8.3 Performance under Varying Train/Test Splits

We evaluated our model and baseline methods using a range of train/test split ratios on experimental datasets. Table 3.5 reports the mean accuracy (with standard deviation in parentheses) across all experimental datasets over five independent runs. Our model consistently outperforms baseline methods, particularly under low-data conditions, achieving at least 11.4% higher average accuracy when trained on only 5% of the data. As the training data increases, our approach maintains superior accuracy while exhibiting smaller standard deviations.

Table 3.5: Model performance on experimental datasets with varying train/test split ratios. Values represent mean accuracy (%) with standard deviations over five runs in parentheses.

TRAIN/TEST RATIO	5/80	40/60	60/40	80/20
PEMNN	79.61(5.24)	86.82(4.86)	88.50(5.09)	88.54(5.68)
LSTMFCN	68.26(12.12)	85.12(7.49)	85.51(6.87)	87.66(7.66)
RESNET	37.47(6.33)	81.01(9.13)	84.62(7.13)	84.47(9.26)
FCN	35.88(7.37)	75.50(8.92)	75.48(13.15)	79.43(10.18)
TRIPLET	55.58(10.17)	66.78(12.06)	69.15(12.26)	66.92(11.45)
INCEPTIONTIME	35.32(3.97)	80.78(12.78)	81.35(9.07)	84.76(11.58)

3.8.4 Physics-guided Pretraining Strategy

Given the challenge of constructing a large, well-annotated dataset using SMFS experimental data and to eliminate experimental labeling and human bias, we employed a physics-guided pretraining strategy. In this approach, deep learning models are pre-trained on simulated datasets generated using physics-based models and subsequently evaluating them on corresponding experimental datasets.

Pretraining deep learning models with the physics of protein unfolding

By utilizing simulation data, we effectively incorporate the underlying physics of protein unfolding into our analysis. Our simulation (Section 3.5.2) is carried out using the WLC model. The WLC model encapsulates the physics of the protein unfolding, accurately describing the entropic spring like behavior of the protein between two unfolding events of the protein, which is also corroborated by experimental data. Subsequently, we test the performance of these pre-trained deep learning models using experimental data.

Data augmentation via linear combinations of examples from different classes is shown to be effective in image classification [97, 98, 99] and sound recognition [100]. Here, we incorporate $M \in \mathbb{N}$ reference data into the force trace branch of PemNN to augment the force data under classification (Figure 3.3). The reference data are

randomly sampled simulated force data from the training dataset. Each reference data $\mathcal{F}_j^{(i)}$ is augmented with the force data $\mathcal{F}^{(i)}$ via the difference $(\mathcal{F}^{(i)} - \mathcal{F}_j^{(i)})$ as additional channels, resulting in a total of $M + 1$ channels. The force data $\mathcal{F}^{(i)}$ can be either a simulated or experimental force curve undergoing classification. This augmented input $[\mathcal{F}^{(i)}, \mathcal{F}^{(i)} - \mathcal{F}_1^{(i)}, \dots, \mathcal{F}^{(i)} - \mathcal{F}_M^{(i)}]^T$, comprising the force data and the reference data, is then passed through the force trace branch.

Physics-guided pretraining strategy performance

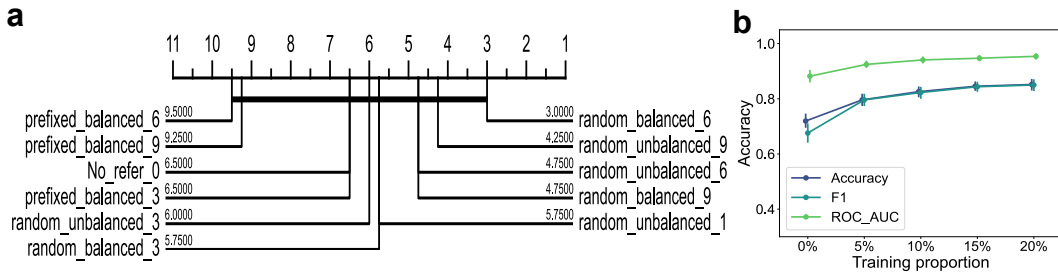


Figure 3.13: Pretraining strategy results with PemNN. a) Critical difference diagram comparing different data augmentation methods. b) The physics-guided pretraining strategy (train size = 0%) slightly underperforms compared to direct training with experimental data (train sizes = 5% to 20%).

We first evaluate the data augmentation block discussed in Appendix 3.8.4 by empirically testing three methods for selecting reference data:

- random balanced: Reference data are randomly selected while maintaining an equal number of samples from each class for every input data.
- random unbalanced: Reference data are randomly selected without class balance for every input data.
- prefixed balanced: Reference data are fixed for all input data and are selected equally from each class.

Figure 3.13a presents the critical diagram comparing these selection criteria for different numbers of reference data M . Both the random balanced and random unbalanced criteria achieve higher rankings compared to the prefixed balanced criterion and the case where no reference data is used. Among these, the random balanced approach with $M = 6$ achieves the highest ranking and is adopted for subsequent evaluations.

The physics-guided pretraining strategy demonstrates performance comparable to cases where the model is directly trained using experimental data, despite not utilizing any experimental data during its training. Specifically, the physics-guided pretraining strategy with PemNN achieves an average accuracy of 72.0% and an average ROC-AUC of 0.88 without involving any experimental data (Figure 3.13). These metrics are only 13% and 0.07 lower, respectively, compared to when 20% of the experimental data is incorporated during training.

Similarities between simulated and experimental data

We compared unfolding forces from experimental and simulated data based on their most probable values and interquartile ranges (IQR) (see Table 3.6). Most probable values closely match, with differences from 1 pN (Titin I27O) to ~ 10 pN (10%) (Bact UtrN-R3, Insect UtrN-R3, DysN-R3), indicating reasonable simulation accuracy. The IQR discrepancy likely arises from using a single double-well potential for domains in simulations. We note a key insight from our study; neural networks pre-trained on simulated data with homogeneous domains can effectively classify the number of proteins involved in experiments with heterogeneous protein domains.

Table 3.6: The most probable forces (in pN), along with the interquartile range (IQR) in parentheses, for both experimental and simulated data across all four protein.

	TITIN I27O	BACT UTRN-R3	INSECT UTRN-R3	DYSN-R3
EXPERIMENTAL	216.35(50.17)	81.58(64.07)	89.45(70.86)	91.25(65.64)
SIMULATED	217.75(36.17)	85.38(22.28)	96.65(40.20)	79.75(25.09)

Non-homogeneity introduces more challenges

The ROC curves (Figure 3.14) for PemNN were generated using the One-vs-Rest strategy, where a given class is regarded as the positive class and the remaining classes are regarded as the negative class as a bulk. For Titin I27O, the ROC-AUC remains consistently high, above 0.85 in all cases, regardless of whether the training data is experimental or simulated (Figure 3.14a, Table 3.7). For both utrophin and dystrophin, the no-molecule class consistently performs well whether experimental data is used for training or not. However, significant improvements are observed for the single-molecule and multiple-molecule classes when experimental data is used for training instead of simulation data, with ROC-AUC of single-molecule class increasing by at least 0.2 for insect UtrN-R3, bact UtrN-R3, and DysN-RR3 (Figure 3.14b-d, Table 3.7). This suggests notable differences between simulation and experimental data for both utrophin and dystrophin.

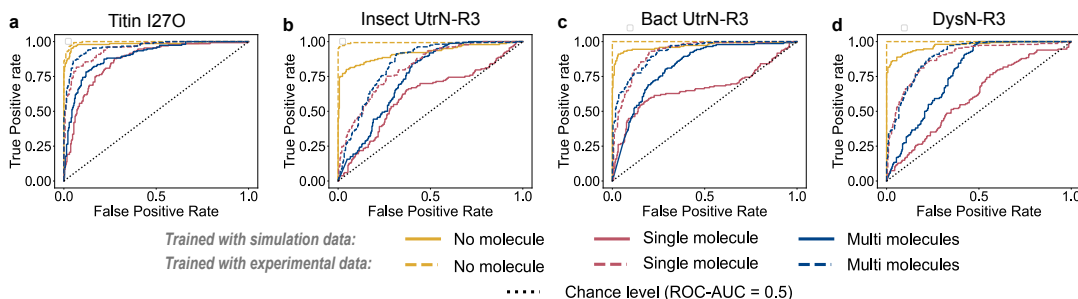


Figure 3.14: ROC curves from a single run for four datasets, (a) Titin I27O, (b) Insect UtrN-R3, (c) Bact UtrN-R3 and (d) DysN-R3 using PemNN. ROC curves are plotted using One-vs-Rest strategy, with no molecule, single molecule and multiple molecules classes depicted in yellow, red and blue, respectively. Solid lines represent results trained with simulation data, while dashed lines indicate results trained with experimental data. These ROC curves are generated using 80% of experimental data, with the corresponding ROC-AUC in Table 3.7.

The ROC curves allow us to adjust the optimal probability threshold. We can achieve high precision but accept low sensitivity to ensure the reliability of single molecule data statistics. The probability threshold can be chosen to tune for a minimum sensitivity or

for a minimum specificity. For example, we have selected the thresholds to achieve an average of 93% of the positive identifications of single molecule are correct across four experimental datasets. Thus the method can effectively filter data such that inferences can be drawn from true single molecule experiments. Alternatively, we can aim for high sensitivity but accept low precision to avoid losing data, given the scarcity and high cost of producing SMFS data (Appendix 3.8.4).

We further investigated the possible reasons for the superior performances of Titin I27O compared to insect UtrN-R3, bact UtrN-R3, and DysN-R3. Titin I27O has identical domains whereas utrophin and dystrophin have heterogeneous domains. The simulation model we use assumes identical folded domains in a protein. A model that can simulate a molecule with heterogeneous domains requires the model parameters for each domain. These parameters are not available for insect UtrN-R3, bact UtrN-R3, and DysN-R3. Thus, we hypothesize that when information about the molecular domains is missing in the simulation model, the usage of experimental data in training provides substantial improvements in performance. This is the case with molecules with heterogeneous domains.

Table 3.7: ROC-AUC of ROC curves in Figure 3.14.

	TRAINED WITH SIMULATION DATA			TRAINED WITH EXPERIMENTAL DATA		
	NO MOLECULE	SINGLE MOLECULE	MULTIPLE MOLECULES	NO MOLECULE	SINGLE MOLECULE	MULTIPLE MOLECULES
TITIN I27O	0.98	0.86	0.90	1.00	0.95	0.96
INSECT UTRN-R3	0.92	0.61	0.75	0.99	0.81	0.84
BACT UTRN-R3	0.97	0.67	0.82	1.00	0.92	0.93
DYSN-R3	0.97	0.61	0.77	1.00	0.88	0.90

Adjusting probability thresholds

The primary goal is to identify single molecule force curves, so we simplify the problem from multi-class to binary classification, focusing on distinguishing force curves from single molecules versus those from no molecule and multiple molecules. By leveraging the ROC curve, which illustrates the true positive rates (TPR) against the false positive

rates (FPR) by varying the probability threshold t_p , we can choose the optimal probability thresholds t_{op} to classify these binary classes. The optimal probability threshold t_{op} is determined by maximizing the difference between TPR and FPR with a weight $\alpha \in \mathbb{R}$ on FPR,

$$t_{op} = \operatorname{argmax}_{t_p} (TPR - \alpha \cdot FPR). \quad (3.18)$$

Subsequently, optimal thresholds with varying α are applied to the binary classification task. The results, compared to the original threshold, are presented in Figure 3.15 for ResNet. The models are trained on simulation data and then evaluated on experimental data.

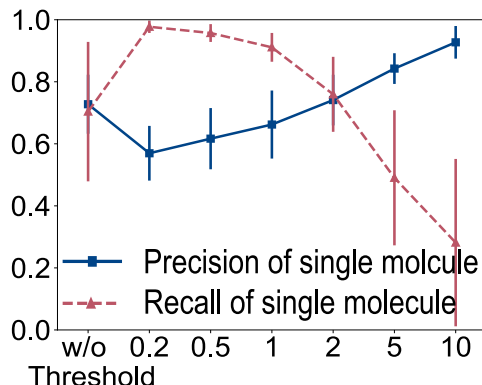


Figure 3.15: Performance of PemNN with varying probability thresholds for all four experimental datasets. The x-axis represents values of α , with 'w/o Threshold' indicating no threshold is used.

A larger weight α places greater emphasis on false positive, thereby increasing precision. For example, the precision of single molecule class improves to 0.93 when $\alpha = 10$. However, this improvement comes with a trade-off in recall. This adjustment enhances the reliability of single molecule data statistics, as we can be more confident that no-molecule and multiple-molecules force curves are not misclassified as single-molecule class. Conversely, if the preference is to accept some false positives rather than exclude any true positives as the data are expensive and scarce, $\alpha = 0.2$ would be the choice to achieve high recall, albeit with lower precision.

Discussion and limitations

A significant insight of the study is that we can perform the task of classification of data under no molecule, single molecule, and multiple molecules (for proteins with heterogeneous domains) being pulled where the training is done based on simulation data, where a single double-well potential model of the domains is employed. This implies that there is enough other information on the force-curves that allows discrimination between no, single, and multiple proteins-based force-curves. We emphasize that neural networks trained on simulated data using homogeneous domains have the capability to classify between the number of proteins being involved in the experiments of protein molecules with heterogeneous domains.

We identify the following limitations for the physics-guided pretraining strategy. First, the simulation algorithm relies on energy landscape parameters, which are not straightforward to obtain and typically necessitate experiments involving multiple pulling speeds or constant forces. Transforming our physics-guided pretraining strategy to be agnostic to the energy landscape parameters is beyond the scope of this study. Second, in our physics-based protein unfolding model, we assume every protein domain behaves identically. However, many proteins, including utrophin and dystrophin have folded domains that are significantly different from each other. Developing methods that can extract the individual energy landscapes of the different dissimilar domains in a protein is likely to aid in identifying single molecule force curves arising from proteins with dissimilar domains. The development of techniques to extract distinct energy landscapes from a small amount of experimental data should be investigated in the future.

Chapter 4

A Method for Identifying Heterogeneous Protein Folding Domains

Natural proteins consist of heterogeneous folding domains with distinct mechanical characteristics, and understanding these domain-level behaviors is crucial for uncovering disease mechanisms. Single-molecule force spectroscopy (SMFS) provides a way to probe the mechanical response of whole proteins, but existing methods face challenges in resolving heterogeneous domains, especially without prior information. In this chapter, we introduce the first automated framework for detecting heterogeneous folding domains in SMFS data, including both existing clustering methods and a novel physics-aware deep clustering model. By enabling domain-level biological inference, our approach addresses previous limitations that relied on averaging and overlooked heterogeneity, thereby advancing the study of protein unfolding mechanics.

4.1 Introduction

Advances in instrumentation—such as atomic force microscopy (AFM) [17] and optical tweezers [18]—have enabled SMFS measurements with force sensitivity in the femtonewton to piconewton range and nanometer spatial resolution. However, interpreting SMFS

data is challenging, particularly when dealing proteins with multiple, heterogeneous domains. Prior analysis of SMFS data often employed well-characterized proteins with known structural or mechanical properties [101], with a single domain or multiple heterogeneous domains; thus, excluding applicability to proteins with no prior knowledge and/or proteins with heterogeneous domains. These limitations, coupled with the therapeutic potential of such insights, motivate the development of automated, data-driven tools for identifying heterogeneous domains in natural proteins.

In this work, we use clustering models to identify heterogeneous domains in natural proteins. We both apply existing clustering models and provide a novel deep clustering physics-aware architecture, LatentUnfold. LatentUnfold employs two autoencoders to learn latent representations; the first captures latent features from SMFS data, while the second is designed to model relationships within the data that are informed by physical models, such as the Worm-Like Chain (WLC) model [77]. The latent representations are then jointly optimized for better clustering assignments. We validated all clustering models using both simulated data, generated by Monte Carlo simulation, and real SMFS experimental data from four biologically important protein molecules. Across 51 simulation datasets, LatentUnfold achieved an average ranking of 2.59, outperforming existing clustering methods by at least 16.5% in terms of Silhouette score. For a synthetic protein composed of two well-characterized domains, we successfully identified and attributed the domains with mechanical properties consistent with previously reported values. Last, we applied to natural protein molecules, utrophin and dystrophin, and uncovered two distinct types of heterogeneous domains with different mechanical properties.

This work represents the first reported effort of automating the identification of heterogeneous domain in SMFS data. It provides insight into the mechanical behavior of protein unfolding and highlights the value of domain-level analysis. Previously, a lack of such a facility has limited biological inferences to be drawn in an average sense without accounting for the impact of heterogeneity which is a substantial limitation.

4.2 Literature Review

4.2.1 Identification of heterogeneous protein domains in SMFS data.

Natural protein molecules with heterogeneous domains are typical, but current methods are limited to mechanical molecular fingerprints, which are well-characterized protein molecules with known structural or mechanical properties [101], to confirm the presence of single molecules and identify specific domain unfolding events. However, these methods can not be applied to natural proteins comprised of multiple heterogeneous domains [102, 103]. Moreover, incorporated molecular fingerprints can influence the native behavior of the studied protein, which may result in misleading experimental outcomes [104]. Domain expert knowledge is necessary to identify different molecular fingerprints, polymer elasticity models, such as the worm-like chain (WLC) model [77], are employed to fit individual unfolding events [105]. To identify these heterogeneous domains in natural protein molecules, we are the first to report the automation of identifying heterogeneous domains and the first to both apply and innovate clustering models in a data driven manner.

4.2.2 Clustering methods for time series data.

Time series clustering is widely explored in machine learning and deep learning contexts and has proved effective for discovering structural patterns in unlabeled data [106]. Classical approaches like K-Means and K-medoids with dynamic time warping (K-means DTW and K-medoids DTW) can handle temporal misalignments [107]. However, these methods suffer from increase in data dimensionality [108]. Autoencoders (AEs) are designed to learn low-dimensional features through neural networks, enabling more effective clustering with algorithms like K-Means [109]. Different neural network architectures have been explored within autoencoder frameworks, multiple linear perceptron (MLP) and convolutional neural network (CNN) can extract latent features efficiently, and recurrent neural network (RNN) and long short-term memory (LSTM) can capture long-range temporal dependencies [110]. More recently, deep clustering methods, such as deep embedded clustering (DEC) [111], and deep temporal clustering representation (DTCR) [112], further optimize feature learning by clustering objectives, yielding compact and cluster-friendly embeddings. In this article, apart from adapting

extant machine learning techniques, we propose LatentUnfold, a novel deep clustering framework for identifying heterogeneous domains from SMFS data by learning latent representations.

4.3 Problem Statement

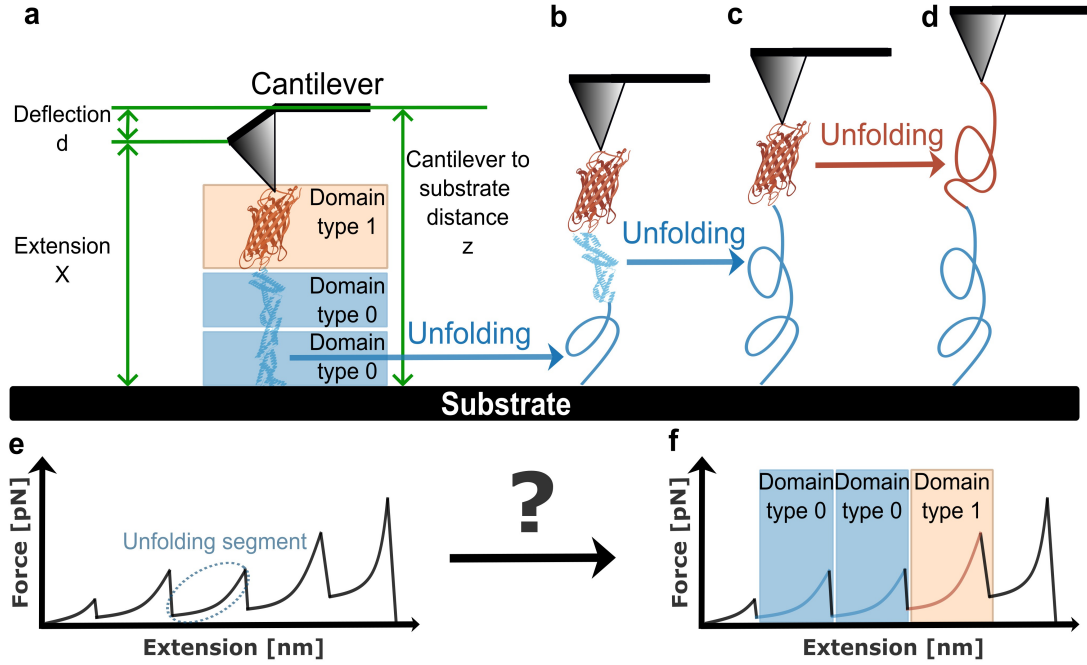


Figure 4.1: Illustration of AFM-based SMFS. (a) Schematic of the experimental configuration with the protein molecule composed of three folded domains. The deflection d and the cantilever-substrate distance z are measured during pulling. (b–d) Illustration of sequential unfolding of three domains, including two type 0 domains (blue) and one type 1 domain (red). (e) Representative force-extension curve with the unfolding segment indicated by a dashed oval. (f) Matching unfolding curve segments to domain types 0 and 1, enabling identification of heterogeneous domains.

In AFM-based SMFS experiments a protein molecule of study is tethered between the tip of micro-cantilever and the substrate. When the cantilever is retracted from the

substrate at a constant speed, the force F experienced by the cantilever is calculated by $F = kd$, where k is the spring constant of the cantilever and d is the deflection of the cantilever tip. The force F acts as a proxy for the tensile force applied to the protein molecule, while the extension X is the protein length (Figure 4.1a), resulting in force-extension curve. The example shown in Figure 4.1e corresponds to a single protein molecule with two different types of domains, labeled as type 0 and 1. As the cantilever retracts, the first force drop reflects the rupture of nonspecific adhesion between the tip and the substrate [65]. Subsequent three force drops are caused by the sequential unfolding of three folded protein domains under mechanical tension (Figure 4.1b–d). Finally, the protein detaches either from the cantilever tip or the substrate, resulting in the final force drop [8].

The force-extension curve (Figure 4.1e) reflects the unfolding process of proteins composed of multiple heterogeneous domains. Every domain unfolds with its own characteristic force-extension response, which we term as an unfolding segment. Identifying which unfolding segment corresponds to which domain type remains challenging, yet it is crucial to further the understanding of the biological function of the protein. The transition from Figure 4.1d to Figure 4.1e illustrates the goal: resolving the curve into domain-specific unfolding segments—type 0 and type 1.

However, clustering unfolding segments from force-extension curves is inherently challenging due to several factors: (1) SMFS experiments exhibit low success rates of single-molecule capture, necessitating the collection of thousands of force curves (typically 2000–5000) to obtain statistically meaningful inferences [3]; (2) force curves often contain noise, nonspecific adhesion, and detachment artifacts, complicating the identification of true domain unfolding segments; and (3) often proteins under study possess unknown or complex domain architectures with no prior characterization, making manual annotation by domain experts [113] infeasible and error-prone.

Given these challenges, it is difficult to do manual visualization, which needs expert knowledge, introduces subjectivity, and lacks scalability. Therefore, there is a pressing need for robust, automated methods capable of identifying heterogeneous domains from noisy, unlabeled force-extension curves. We separate force-extension curves into unfolding segments, resulting in a dataset $\mathcal{D} = [(\mathcal{F}^{(1)}, \mathcal{X}^{(1)}), \dots, (\mathcal{F}^{(n)}, \mathcal{X}^{(n)})]$. Each force-extension pair $(\mathcal{F}^{(i)}, \mathcal{X}^{(i)})$ corresponds to one unfolding segment, comprising T_i ordered

real values of force $\mathcal{F}^{(i)} = [F_1^{(i)}, F_2^{(i)}, \dots, F_{T_i}^{(i)}]$ and extension $\mathcal{X}^{(i)} = [X_1^{(i)}, X_2^{(i)}, \dots, X_{T_i}^{(i)}]$. All segments are standardized to a fixed length T after pre-processing.

4.4 Methods

We first apply existing methods described in Section 4.4.1 and further propose a novel physics-aware deep clustering framework, LatentUnfold in Section 4.4.2, to identify heterogeneous domains of natural protein molecules.

4.4.1 state-of-the-art methods

K-Means is a widely used clustering algorithm that assigns data points to the nearest centroid to minimize intra-cluster variance:

$$J = \sum_{k=1}^K \sum_{x_i \in C_k} dist(x_i - \mu_k) \quad (4.1)$$

where K is the number of clusters, x_i is a data point, μ_k is the centroid of cluster C_k and $dist(\cdot)$ denotes the distance. K-Medoids selects actual data points (medoids) as cluster centers, making it more robust to noise and outliers. Common options for distance are Euclidean norm and DTW. However, applying it directly to time series data using Euclidean norm is inadequate due to temporal misalignments. DTW addresses this by computing the minimum cumulative distance between aligned points of two sequences $X = [x_1, x_2, \dots, x_n]$ and $Y = [y_1, y_2, \dots, y_m]$, allowing for elastic shifts in the time axis.

$$DTW(X, Y) = \min_{\pi \in \mathcal{A}(n, m)} \sum_{(i, j) \in \pi} d(x_i, y_j) \quad (4.2)$$

where π is a warping path from start to end, $\mathcal{A}(n, m)$ is the set of all valid warping paths and $d(x_i, y_j)$ is the distance function (usually Euclidean distance).

Autoencoders (AEs) are unsupervised neural networks designed to learn low-dimensional latent representations by reconstructing the input [109]. For instance, MLP-AE consists of fully connected layers, and ResNet-AE uses multiple convolutional layers with residual connections, followed by a global average pooling and a dense embedding layer. These

autoencoders are trained solely with reconstruction loss, and clustering is assigned with the learned latent representations using K-Means.

DEC [111] is a deep clustering framework where a neural network is trained to jointly learn latent representations and cluster assignments. Deep temporal clustering representation (DTCR) [112] is a deep clustering method tailored for time series data. It uses stacked bidirectional gated recurrent units (GRUs) to encode temporal patterns and integrates three objectives: reconstruction loss, a binary classification loss to distinguish real from time-shuffled inputs, and a K-means loss to learn cluster assignments.

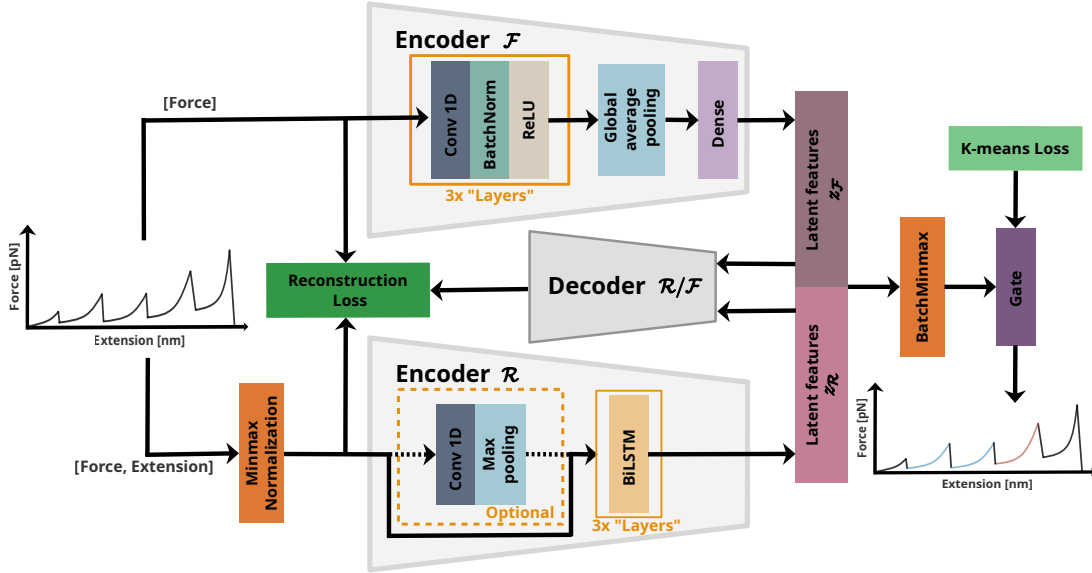


Figure 4.2: Model architecture of LatentUnfold, a novel physics-aware deep clustering framework.

4.4.2 LatentUnfold, a novel physics-aware model

LatentUnfold aims to capture two complementary features from unfolding segments, force magnitude and force-extension relationship, using two autoencoders to extract latent representations, as illustrated in Figure 4.2. The force autoencoder is trained exclusively on force magnitudes; extension values are omitted because identical domains can unfold at different points along the trace, yielding disparate extension readings that

would confound representation learning. Force alone, however, cannot fully describe mechanical behavior of protein molecules, which is governed by the relationship between force and extension modeled by protein unfolding physics. The relationship autoencoder learns from normalized force–extension pairs, embedding the underlying physical relationship into latent representations. After obtaining latent representations from both autoencoders, we apply a K-means loss to learn how much to trust each representation for accurate cluster assignments.

The force encoder $f_{enc_{\mathcal{F}}}$ maps $\mathcal{F}^{(i)}$ into a latent representation $z_{\mathcal{F}}^{(i)} \in \mathbb{R}^{m_{\mathcal{F}}}$ with $m_{\mathcal{F}} \in \mathbb{N}^+$ and the decoder $f_{dec_{\mathcal{F}}}$ reconstructs the data from the latent space. After both encoding and decoding process, the reconstructed force sequence $\hat{\mathcal{F}}^{(i)} \in \mathbb{R}^T$ is computed as $\hat{\mathcal{F}}^{(i)} = f_{dec_{\mathcal{F}}}(f_{enc_{\mathcal{F}}}(\mathcal{F}^{(i)}))$.

We use the mean square error (MSE) as the reconstruction loss to train the autoencoder with $\mathcal{Y}$ being the input and $\hat{\mathcal{Y}}$ being the reconstruction, defined as:

$$\mathcal{L}_{rec} = \frac{1}{n} \sum_{i=1}^n ||\mathcal{Y}^{(i)} - \hat{\mathcal{Y}}^{(i)}||^2. \quad (4.3)$$

The force encoder architecture consists of three convolutional blocks, followed by a global average pooling layer and a fully connected layer to produce the latent representation. Each convolutional block is composed of a 1-Dimensional convolutional layer, batch normalization layer [66] and a Rectified Linear Unit (ReLU) [67] activation layer. The decoder is constructed as the mirror of the encoder architecture with three convolutional blocks.

The relationship encoder $f_{enc_{\mathcal{R}}}$ extracts a latent representation $z_{\mathcal{R}}^{(i)} \in \mathbb{R}^{m_{\mathcal{R}}}$ with $m_{\mathcal{R}} \in \mathbb{N}^+$, which is then passed through a relationship decoder $f_{dec_{\mathcal{R}}}$ to reconstruct both sequences $(\hat{\mathcal{F}}^{(i)}, \hat{\mathcal{X}}^{(i)})$. This autoencoder is trained using the reconstruction loss defined in Equation 4.3. We tested two encoder architectures, the first one, referred to as LatentUnfold(LSTM), uses three stacked bidirectional long short-term memory (BiLSTM) layers [114] with the decoder is constructed as the mirror of the encoder architecture. The second, referred to as LatentUnfold(CNN-LSTM) inspired by deep temporal clustering [115], introduces a convolutional layer followed by a max-pooling layer prior to the BiLSTM stack to better capture cross-channel relationships, corresponding to the optional block in Figure 4.2.

After obtaining the two latent representations $z_{\mathcal{F}}$ and $z_{\mathcal{R}}$, we apply feature-wise

min-max normalization to remove scale differences between them. To adaptively control the influence of each representation on the clustering task, we introduce learnable parameters $\alpha_{\mathcal{F}}$ and $\alpha_{\mathcal{R}}$, computed as $\alpha_{\mathcal{F}}, \alpha_{\mathcal{R}} = \text{softmax}(W \cdot [z_{\mathcal{F}} \ z_{\mathcal{R}}] + b)$, with $W \in \mathbb{R}^{n \times (m_{\mathcal{F}} + m_{\mathcal{R}})}$ and $b \in \mathbb{R}^n$ are weights and bias of gate block.

These weights modulate the relative importance of each latent space. The final combined representation is $H = [\alpha_{\mathcal{F}} \cdot z_{\mathcal{F}} \ \alpha_{\mathcal{R}} \cdot z_{\mathcal{R}}] \in \mathbb{R}^{n \times (m_{\mathcal{F}} + m_{\mathcal{R}})}$. We optimize a spectral-relaxed K-means loss [116, 112] defined as:

$$\mathcal{L}_{K-Means} = \text{Tr}(H^T H) - \text{Tr}(F^T H^T H F), \quad (4.4)$$

where $\text{Tr}(\cdot)$ denotes the matrix trace and $F \in \mathbb{R}^{n \times K}$ is the cluster indicator matrix, where K is the number of clusters.

4.5 Datasets

4.5.1 Simulation data

A Monte Carlo simulation based method, adopted from Section 3.5.2, was used to generate simulated force-extension data. Below, we provide an overview of the simulation framework with minor modifications.

The system consists of a single protein composed of two distinct domain types, indexed by $i = 0, 1$. The cantilever base is pulled at constant velocity v , and at each time step Δt the protein extension X is calculated via a force-balance equation using the Worm-Like Chain (WLC) model [39]:

$$kd = \frac{k_B T}{L_p} \left[\frac{1}{4 \left(1 - \frac{X}{L_c}\right)^2} - \frac{1}{4} + \frac{X}{L_c} \right], \quad (4.5)$$

where k_B is the Boltzmann constant, T is temperature, and L_c and L_p are the contour length and the persistence length of the protein, respectively.

At each time step, the probability that a domain of type $i = 0, 1$ unfolds is given by

$$P_i(F) = (D_i - U_i)(1 - e^{-k_{off,i}(F)\Delta t}), \quad (4.6)$$

where D_i is total number of domains and U_i is the number of unfolded domains of type i , and $k_{off,i}(F)$ is the transition rate that can be determined with the Dudko-Hummer-Szabo model [34]:

$$k_{off,i}(F) = k_{0,i} \left(1 - \frac{\nu F \Delta x_i^\ddagger}{\Delta G_i^\ddagger} \right)^{\frac{1}{\nu}-1} e^{-\beta \Delta G_i^\ddagger \left[1 - \left(1 - \frac{\nu F \Delta x_i^\ddagger}{\Delta G_i^\ddagger} \right)^{1/\nu} \right]}, \quad (4.7)$$

where $k_{0,i}$ is the intrinsic transition rate, $\Delta x_i^\ddagger$ is the distance to energy barrier, $\Delta G_i^\ddagger$ is the energy barrier height for type i domain, and $\beta = \frac{1}{k_B T}$, and $\nu = 1/2$ or $2/3$, representing the cusp-like or linear-cubic energy landscape.

To determine whether unfolding occurs, a random number between 0 and 1 is generated and compared to $P_i(F)$. If the random number is smaller than $P_i(F)$, the type i domain unfolds. If both $P_0(F)$ and $P_1(F)$, exceed the random number, the domain with the higher probability unfolds, while the other remains folded. Upon unfolding, the contour length of the entire protein is updated by adding an increment $\Delta L_{c,i}$ corresponding to the unfolded domain type. The two domain types differ in both their kinetic parameters, k_0 , $\Delta x^\ddagger$, and $\Delta G^\ddagger$, and their elastic property such as the contour length increment ΔL_c .

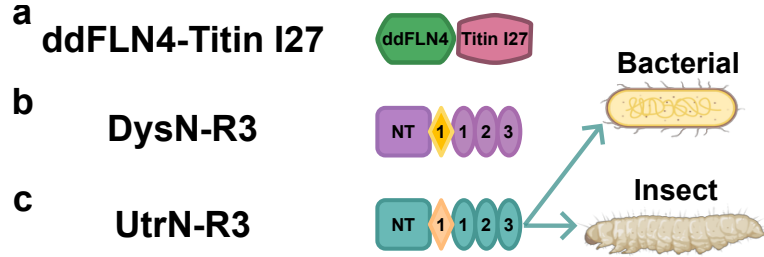


Figure 4.3: Diagrams of protein molecules. (a) ddFLN4–Titin I27, consisting of one ddFLN4 domain and one Titin I27 domain. (b) DysN-R3, a fragment of dystrophin spanning the N-terminus (NT) through spectrin-like repeat 3 (SLR3). (c) UtrN-R3, a corresponding utrophin fragment, further divided into bacterial UtrN-R3 and insect UtrN-R3 based on the expression system. Ovals represent spectrin-like repeats (SLRs), and NT indicates the N-terminus domain.

4.5.2 Experimental data

We employed experimental data that were obtained from physical experiments conducted via AFM on four different protein molecules. The first is ddFLN4–Titin I27 (Fig. 4.3a), a synthesized protein composed of two distinct domains: one ddFLN4 domain and one Titin I27 domain. Both are well-characterized, single-domain molecules widely used in SMFS as reference systems due to their robust and reproducible mechanical properties [6]. ddFLN4 is the fourth immunoglobulin (Ig)-like domain from *Dictyostelium discoideum* filamin, a cytoskeletal actin-binding protein that provides mechanical support to cells. Titin I27 is the 27th Ig-like domain of human titin, a giant sarcomeric protein responsible for the passive elasticity of muscle.

The remaining three protein constructs are derived from dystrophin and utrophin, two natural proteins with considerable variations in their sequences and structures. Dystrophin is primarily localized at the muscle cell membrane, or sarcolemma, in striated muscle tissue. Dystrophin deficiencies lead to Duchenne muscular dystrophy (DMD), a severe and fatal muscle wasting disorder occurring in 1 out of 4000 male births [13]. Utrophin, a fetal homologue of dystrophin, is currently under investigation as a potential therapeutic substitute for dystrophin in DMD treatment. We include dystrophin and utrophin fragments that span from the N-terminus (NT) through spectrin-like repeat (SLR) 3 domains. These constructs are referred to as DysN-R3 (Fig. 4.3b) for dystrophin and UtrN-R3 (Fig. 4.3c) for utrophin. Previous studies have shown that the mechanical properties of UtrN-R3 are influenced by the expression system used—whether derived from insect or bacterial cells [3, 58]. To account for the expression system, we further distinguish between insect UtrN-R3 and bact UtrN-R3 in our analysis. For all three constructs, the NT domain is presumed to be structurally distinct from the SLR domains, as it contains roughly twice the number of amino acids compared to an individual SLR domain [14]. No new experimental data were generated for these constructs; all data were obtained from previously studies [3, 58]. In summary, our four real protein molecules are: ddFLN4-Titin I27, bact UtrN-R3, insect UtrN-R3, DysN-R3.

4.6 Results

We outline the evaluation setup in Section 4.6.1, present performance on simulated data with known ground truth in Section 4.6.2, and finally evaluate on experimental data from four different protein molecules in Section 4.6.3.

4.6.1 Evaluation

Each dataset is partitioned into training and test subsets using a 20-80 split. Each model configuration is trained and evaluated five times independently to account for stochasticity and ensure statistical robustness. To evaluate clustering performance, we use both external and internal metrics.

Clustering accuracy [117] is an external metric that measures how well the predicted cluster labels y_{pred} align with the ground truth labels y_{true} . Since cluster labels are arbitrary, a permutation function $\pi(\cdot)$ is applied to find the best label alignment. The accuracy is defined as

$$Accuracy = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(y_{true}(i) = \pi(y_{pred}(i))), \quad (4.8)$$

where n is the number of samples and $\mathbb{1}(\cdot)$ is the indicator function. As an internal metric, the Silhouette score [118] assesses the compactness and separation of the clusters without using ground truth. For each sample i , $a(i)$ is the average distance to other points in the same cluster, and $b(i)$ is the minimum average distance to points in any other cluster. The Silhouette score is defined as

$$Silhouette = \frac{1}{n} \sum_{i=1}^n \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \quad (4.9)$$

Higher values indicate better-defined clusters, with 1 being ideal separation and values near 0 indicating overlapping clusters.

Because experimental SMFS data lack ground truths, we validate our clustering results by examining physical properties that quantify mechanical behavior. Two widely used physical properties are the unfolding force and the contour length increase [6]. The unfolding force, the peak force within an unfolding segment, can be described by the Dudko-Hummer-Szabo (DHS) model [34], whose parameters include the intrinsic

transition rate k_0 , the distance to the energy barrier $\Delta x^\ddagger$, and the energy barrier height $\Delta G^\ddagger$. The contour length L_c , the maximum physically possible extension, is obtained by fitting the WLC model (Equation 3.2); the contour-length increase ΔL_c is the difference in L_c between consecutive unfolding segments.

In our model, the force autoencoder employs convolutional layers with a fixed filter size of 64. The filter lengths for the first, second, and third convolutional layers are set to 8, 5, and 3, respectively. The default latent space dimension for the force encoder is $m_{\mathcal{F}} = 2$. For the relationship autoencoder, three stacked BiLSTM layers uses size of 64, 8, 2, respectively, resulting in a latent space dimension of $m_{\mathcal{R}} = 4$. In the CNN-LSTM variant, a convolutional layer with 100 filters of size 10 is applied before the BiLSTM layers, followed by a max-pooling layer of size 10. All models are trained with a batch size of 16 for 50 epochs. All models are trained with Adam [95] with the learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e - 8$. Experiments are conducted on a high-performance computing setup featuring NVIDIA and Apple M1 Pro GPUs, ensuring computational efficiency.

	k_0	$\Delta x^\ddagger$ [nm]	$\Delta G^\ddagger$ [$k_B T$]	ΔL_c [nm]
P0	0.0001	0.40	20.0	30.0
P1	0.08	0.37	10.0	40.0

Table 4.1: Simulation parameters for parameter sets 0 and 1.

4.6.2 Performance on Simulated Data with Known Ground Truth

The simulation data generation process was described in Section 4.5.1, where each dataset was created using a distinct pair of parameters. Each dataset produced approximately 1,000 simulated force-extension curves at a fixed pulling speed. We simulated datasets at three pulling speeds: 500, 1000, and 2000 nm/s. Since the ground truth labels are known for the simulated data, we evaluated clustering performance using both accuracy and Silhouette score.

Two well-separated simulation parameters

We first evaluate our approach on simulated datasets generated from two well-separated parameter sets, referred to as ‘P0’ and ‘P1’ (Table 4.1). As shown in Figure 4.4a, the predicted unfolding force distributions closely match the ground truth at 1000 nm/s. This agreement persists across other pulling speeds (500 and 2000 nm/s), and similar consistency is observed for contour length increments ΔL_c (Figure 4.4b). Predicted unfolding forces not only align well with the ground truth but also exhibit a consistent shift toward higher values with increasing pulling speed, while ΔL_c shows minimal dependence on pulling speed—both trends consistent with known biophysical behavior. Example force extension curves (Figure 4.4c) further illustrate the identification of distinct unfolding segments. Finally, LatentUnfold(LSTM) and LatentUnfold(CNN-LSTM) achieved an average clustering accuracy of 0.93, with Silhouette scores of 0.61 and 0.75, respectively. These physics-aware architectures outperformed existing clustering methods; for instance, LatentUnfold(CNN-LSTM) surpassed K-means DTW by 0.03 in accuracy and 0.17 in Silhouette score (Table S1 in Appendix).

Model	Accuracy	Silhouette
K-means DTW	0.90 (0.01)	0.57 (0.00)
K-medoids DTW	0.91 (0.01)	0.56 (0.00)
MLP-AE	0.77 (0.12)	0.51 (0.03)
LatentUnfold(CNN-LSTM)	0.93 (0.02)	0.75 (0.02)
LatentUnfold(LSTM)	0.93 (0.01)	0.61 (0.10)

Table 4.2: Clustering accuracy and Silhouette score (mean with standard deviation in parentheses) for each model at 1000 nm/s. Results at 500 and 2000 nm/s are similar and therefore omitted for brevity. Best values at this speed are highlighted in bold.

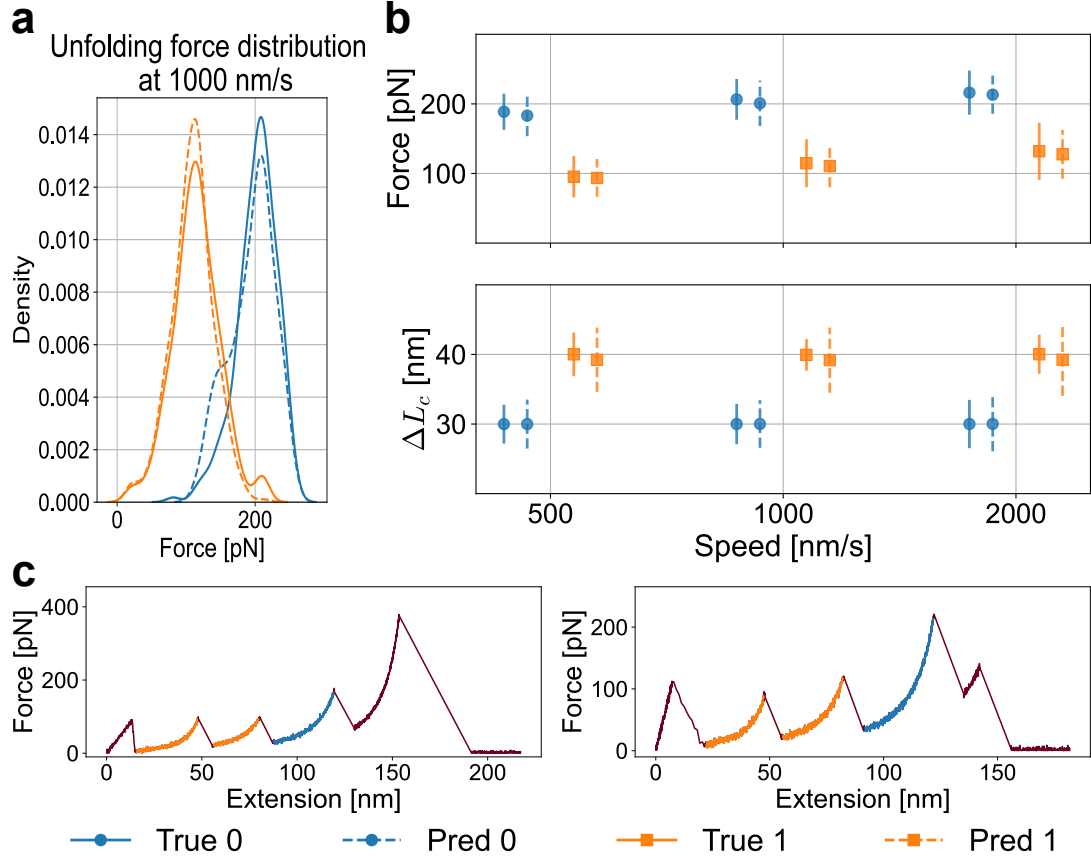


Figure 4.4: Predicted versus ground truth results on simulated data using LatentUnfold(LSTM). (a) Predicted unfolding force distributions (dashed lines) closely match the ground truth distributions (solid lines) for the dataset at 1000 nm/s. (b) Distributions of unfolding force (top) and contour length increment (bottom), with medians shown as center points and error bars indicating standard deviations. (c) Example force-extension curves with two distinct unfolding segments identified.

Robustness study with systematic parameter variation

Following our initial evaluation on a well-separated parameter set, we conducted a comprehensive robustness study to assess the generalizability of our method across varying

simulation conditions. Specifically, we generated 16 additional datasets by systematically varying one of four key parameters at a time, while keeping the remaining parameters fixed at values of ‘P0’, reported in Table 1 in main paper. The four parameters are: the intrinsic transition rate k_0 , the distance to the transition state $\Delta x^\ddagger$, the energy barrier $\Delta G^\ddagger$, and the contour length increase ΔL_c .

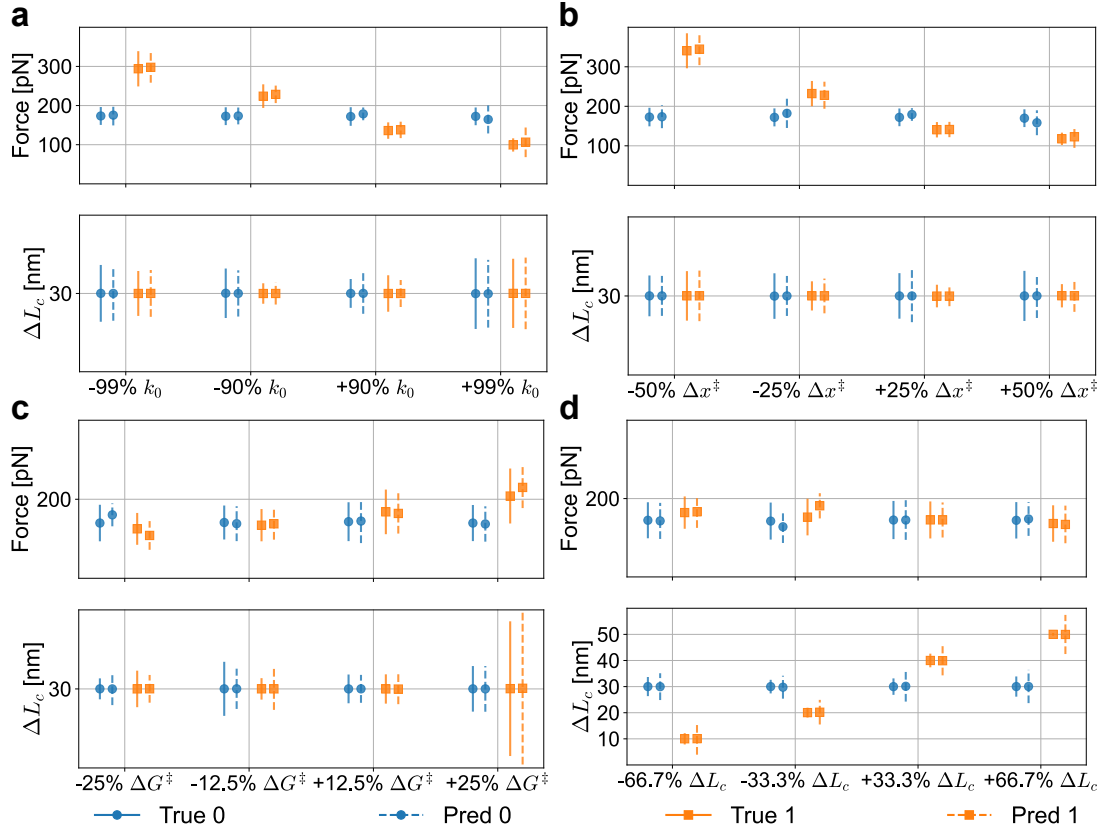


Figure 4.5: Robustness study under systematic parameter variation. Each parameter is modified by a specified fold relative to values of ‘P0’ (Table 1 in the main paper, and annotated accordingly (e.g., +99% k_0 indicates a 99-fold increase in k_0). Ground truth distributions are shown as solid lines, and predictions as dashed lines.

The corresponding results are presented in Figures 4.5. For example, in one case we increased k_0 from 0.0001 to 0.01—a 99-fold increase—denoted as +99% k_0 . Across all tested conditions, the predicted unfolding force and ΔL_c distributions remained closely

aligned with the ground truth, demonstrating the robustness of our method. Additionally, the observed effects of each parameter modification match theoretical expectations. Specifically, a smaller k_0 , a smaller $\Delta x^\ddagger$, or a larger $\Delta G^\ddagger$ each lead to higher unfolding forces without altering ΔL_c . In contrast, changes in ΔL_c primarily affect the contour length increment distribution while having minimal impact on unfolding forces. These results confirm that our clustering framework accurately captures key biophysical properties across a range of controlled parameter perturbations.

Ranking different methods

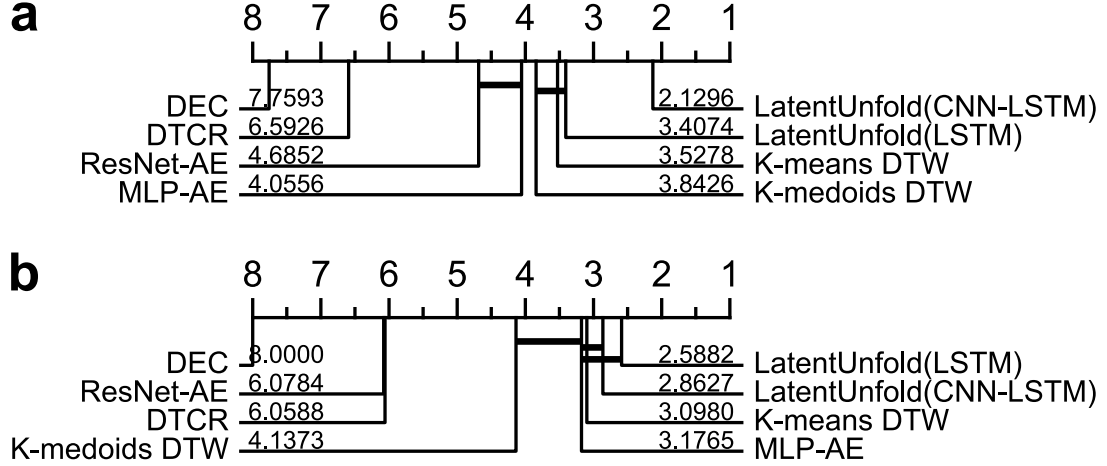


Figure 4.6: Critical difference (CD) diagrams comparing all models across the 51 simulated datasets using (a) clustering accuracy and (b) Silhouette score. The top-performing model receives a rank of 1. Models connected by a thick horizontal line form a group with no statistically significant difference in performance. Statistical significance is assessed following [4, 5].

We next ranking LatentUnfold with existing clustering models across a total of 51 simulated datasets, corresponding to 17 parameter configurations evaluated at three different pulling speeds. One configuration was generated using the well-separated parameters described above, while the remaining 16 were derived from the robustness analysis detailed in the Appendix, where each simulation parameter, k_0 , $\Delta x^\ddagger$, $\Delta G^\ddagger$,

and ΔL_c , was systematically varied across four values. For each simulation condition, we assessed all models on the corresponding test data and ranked them based on their average performance over five runs using two evaluation metrics: clustering accuracy and Silhouette score. The top-performing model for each parameter pair was assigned a rank of 1, and average rankings were then computed across all parameter combinations. These aggregated results are visualized using critical difference (CD) diagrams [4]. As shown in Figure 4.6a, where clustering accuracy is used as the evaluation metric, our physics-aware models—LatentUnfold(CNN-LSTM) and LatentUnfold(LSTM)—achieved the lowest (i.e., best) average ranks of 2.13 and 3.41, respectively. When Silhouette score is used instead (Figure 4.6b), LatentUnfold(LSTM) and LatentUnfold(CNN-LSTM) again outperformed, with average ranks of 2.59 and 2.86, respectively.

Ablation study

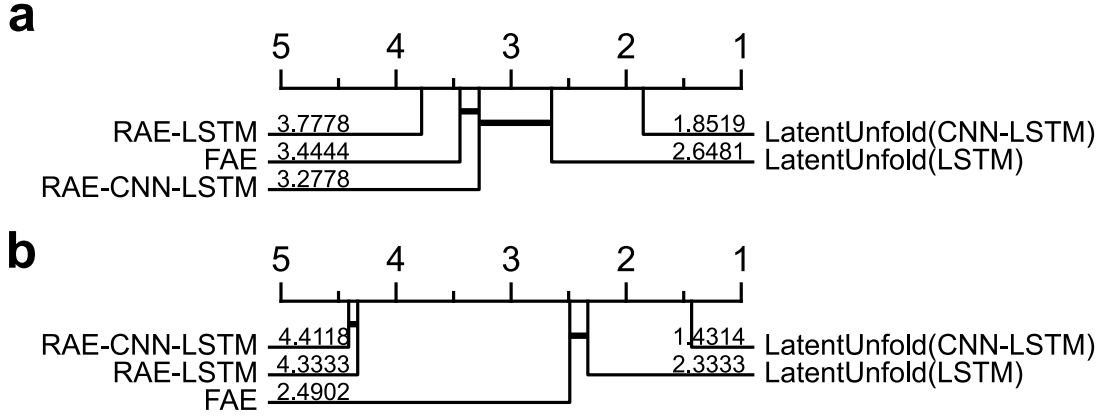


Figure 4.7: Critical difference (CD) diagram for the ablation study with (a) clustering accuracy and (b) Silhouette score. Both LatentUnfold(CNN-LSTM) and Ours (LSTM) consistently outperform the single-autoencoder variants: the force-only autoencoder (FAE), the CNN-LSTM relationship autoencoder (RAE-CNN-LSTM), and the LSTM only relationship autoencoder (RAE-LSTM).

We conducted an ablation study to evaluate the contribution of different components within our model architecture using all simulated datasets. Our proposed models combine a force autoencoder and a relationship autoencoder, and are further categorized into Ours (CNN-LSTM) and Ours (LSTM) based on the architecture of the relationship autoencoder. For comparison, we evaluate variants that use only a single autoencoder: we also evaluated three simplified variants that use only a single autoencoder: the force-only autoencoder (FAE), the CNN-LSTM relationship autoencoder (RAE-CNN-LSTM), and LSTM only relationship autoencoder (RAE-LSTM). In these single-autoencoder configurations, clustering is performed by directly applying K-means(DTW) to the learned latent representations. As shown in Figure 4.7, both LatentUnfold(CNN-LSTM) and LatentUnfold(LSTM) consistently outperform their single-autoencoder counterparts, demonstrating the effectiveness of jointly modeling force and relational information through a dual-autoencoder framework.

4.6.3 Effective Identification of Heterogeneous Domains from Real Experimental Data

Molecule	K=1	K=2	K=4
ddFLN4–Titin I27	-0.09 (0.23)	0.65 (0.04)	0.51 (0.04)
Insect UtrNR3	-0.08 (0.32)	0.67 (0.11)	0.53 (0.04)
Bact UtrNR3	-0.08 (0.21)	0.62 (0.07)	0.52 (0.03)
DysNR3	-0.12 (0.18)	0.61 (0.06)	0.52 (0.03)

Table 4.3: Silhouette scores (mean and standard deviation across five runs and three pulling speeds) for the LatentUnfold(LSTM) model with different cluster numbers on four protein molecules. Additional results and detailed trends are shown in Figure 4.11.

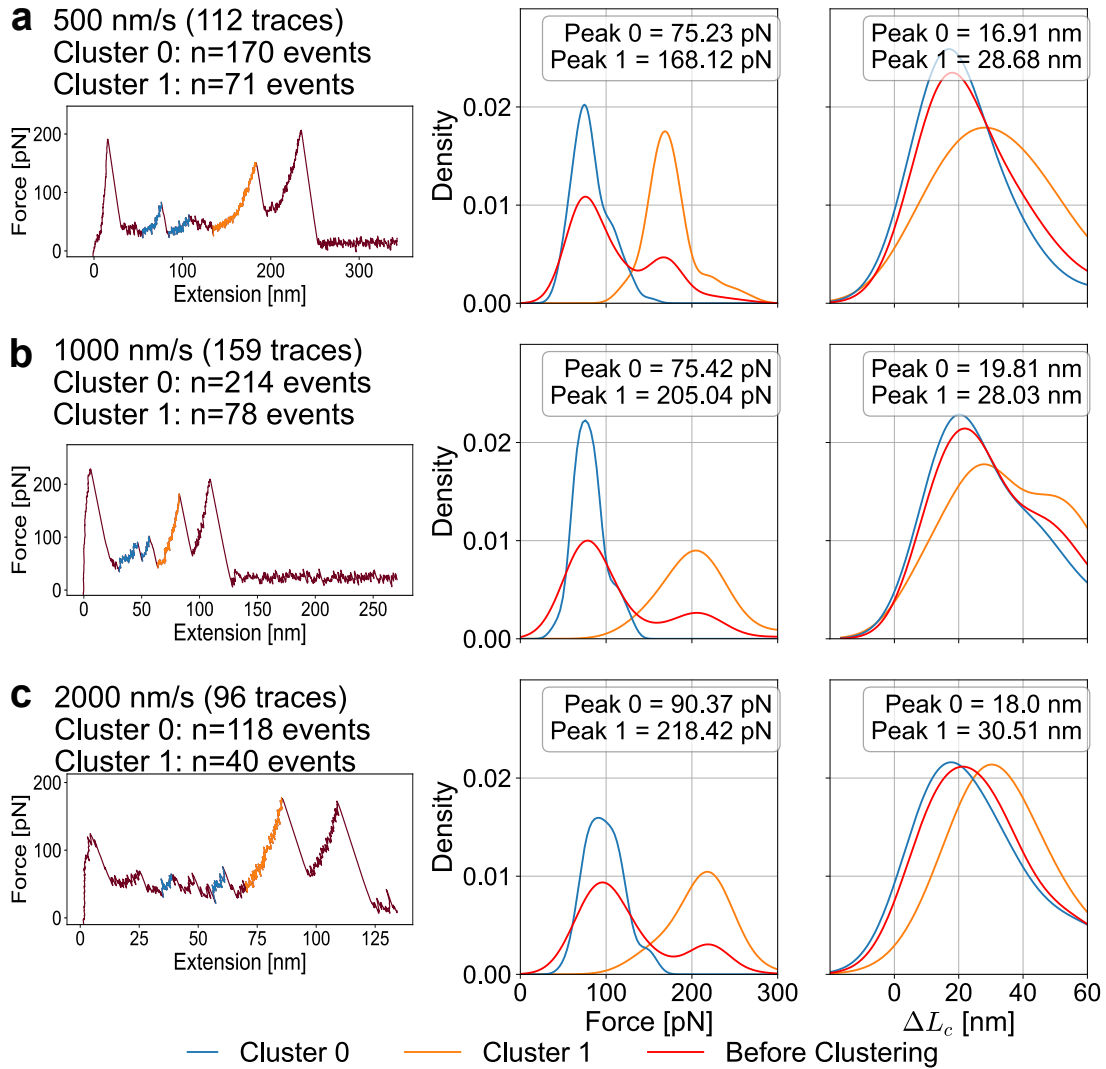


Figure 4.8: Clustering results for a synthesized protein with well-characterized domains. Clustering statistics for ddFLN4 (cluster 0) and Titin I27 (cluster 1) at pulling speeds of (a) 500 nm/s, (b) 1000 nm/s, and (c) 2000 nm/s. The first column shows representative force-extension curves with color-coded segments by predicted cluster labels. The second column displays the unfolding force distributions, and the third column presents the distributions of contour length increase (ΔL_c). Cluster 0 is shown in blue, Cluster 1 in orange, and the combined distribution is shown in red.

Unlike simulated data, real experimental data lacks ground-truth labels and may contain traces that do not originate from single molecule events. As a result, we use only the Silhouette score as the evaluation metric for experimental data. To ensure data quality, we apply a three-step filtering process, with further details provided in Section 4.8.1: 1) apply a physics-augmented deep learning model for initial filtering to remove traces not originating from single molecule proteins [119]; 2) perform expert review of the filtered traces; and 3) select unfolding segments based on the fitting quality to the WLC model. Since clustering labels are arbitrary, we assign label 0 to the cluster with the smaller median unfolding force for consistent presentation of results. We begin by evaluating our methods on the synthesized protein molecule ddFLN4–Titin I27, and then extend the analysis to natural protein molecules, including bact UtrN-R3, insect UtrN-R3, and DysN-R3.

Synthesized Protein: ddFLN4–Titin I27

AFM experiments on ddFLN4–Titin I27 were conducted across multiple sessions, with each session performed at a fixed pulling speed of either 500, 1000, or 2000 nm/s, yielding approximately 3,000–5,000 trials per session. To determine the appropriate number of clusters, we evaluated the Silhouette score for different cluster counts K (Table 4.3). When $K = 1$, the Silhouette score is negative, indicating a lack of meaningful clustering structure. The score peaks at 0.65 for $K = 2$, and subsequently decreases as K increases; suggesting the presence of two well-separated clusters in the dataset.

We applied the top five clustering algorithms; LatentUnfold(CNN-LSTM), LatentUnfold(LSTM), K-means(DTW), K-medoids(DTW), and MLP-AE, as identified in Figure 4.6; independently to the experimental dataset. A consensus voting strategy among the five methods was used to report the clustering statistics shown in Figure 4.8a. Each pulling speed condition (500, 1000, and 2000 nm/s) includes more than 95 force-extension curves, with each predicted cluster containing at least 40 unfolding events. According to previous literature, ddFLN4 unfolds via a two-step pathway with unfolding forces of around 60–80 pN at pulling speeds of 1000 nm/s, and the contour length increase of ~ 16 nm per step [120]. Titin I27 unfolds in a single step, with unfolding forces of 200–220 pN and a contour length increase of ~ 28 nm [8, 52]. In our clustering results, cluster 0 is assigned to the group with the lower median unfolding force,

corresponding to ddFLN4, while cluster 1 represents Titin I27. At 1000 nm/s, Cluster 0 (ddFLN4) exhibits a peak unfolding force of 75 pN, whereas Cluster 1 (Titin I27) peaks at 205 pN. The estimated contour length increases for ddFLN4 and Titin I27 are approximately 20 nm and 28 nm, respectively. These align well with known reported values. As the pulling speed increases from 500 nm/s to 2000 nm/s, the unfolding forces for both clusters increase, while the contour length increases remain largely unchanged. Along with the representative force-extension curves shown in Figure 4.8b, these results demonstrate that our method successfully identifies heterogeneous domains and accurately attributes them to ddFLN4 and Titin I27.

Natural proteins: Utrophin and Dystrophin

Consistent with the trend observed for ddFLN4–Titin I27, all dystrophin and utrophin constructs (insect UtrNR3, bacterial UtrNR3, and DysNR3) exhibit peak Silhouette scores of at least 0.61 at $K = 2$ (Table 4.3). This consistent trend, along with the known structural distinction between the NT domain and the SLRs (the NT domain contains roughly twice as many amino acids as a typical SLR), allows us to safely assume that cluster 0 corresponds to the SLRs and cluster 1 to the NT domain. A difference in contour length increase of approximately 35 nm is expected between the two clusters, with further justification provided in the following.

The NT domain is assumed to be structurally distinct from SLRs, as it contains roughly twice the number of amino acids as a single SLR domain [14]. Each SLR typically adopts a triple α -helical bundle conformation with an approximate folded length of ~ 6 nm of folded length [53, 121], while the folded length of the NT domain is estimated to be ~ 12 nm [54]. The theoretical contour length increase (ΔL_c) upon unfolding of a domain can be calculated using the equation [122, 80, 6]

$$\Delta L_c = (0.365 \text{ nm/AA}) \times (\text{No. of AAs in the domain}) - L_f, \quad (4.10)$$

where 0.365 nm/AA is the approximate contour length per amino acid in a fully extended polypeptide chain, and L_f is the folded length of the domain. Thus, the expected difference in contour length increase between the NT domain and a single SLR is approximately 35 nm. This estimate arises from the NT domain having roughly 120 additional amino acids ($0.365 \times 120 = 43.8$ nm) compared to an SLR, minus an estimated 6 nm

increase in folded length, yielding a net difference of ~ 35 nm.

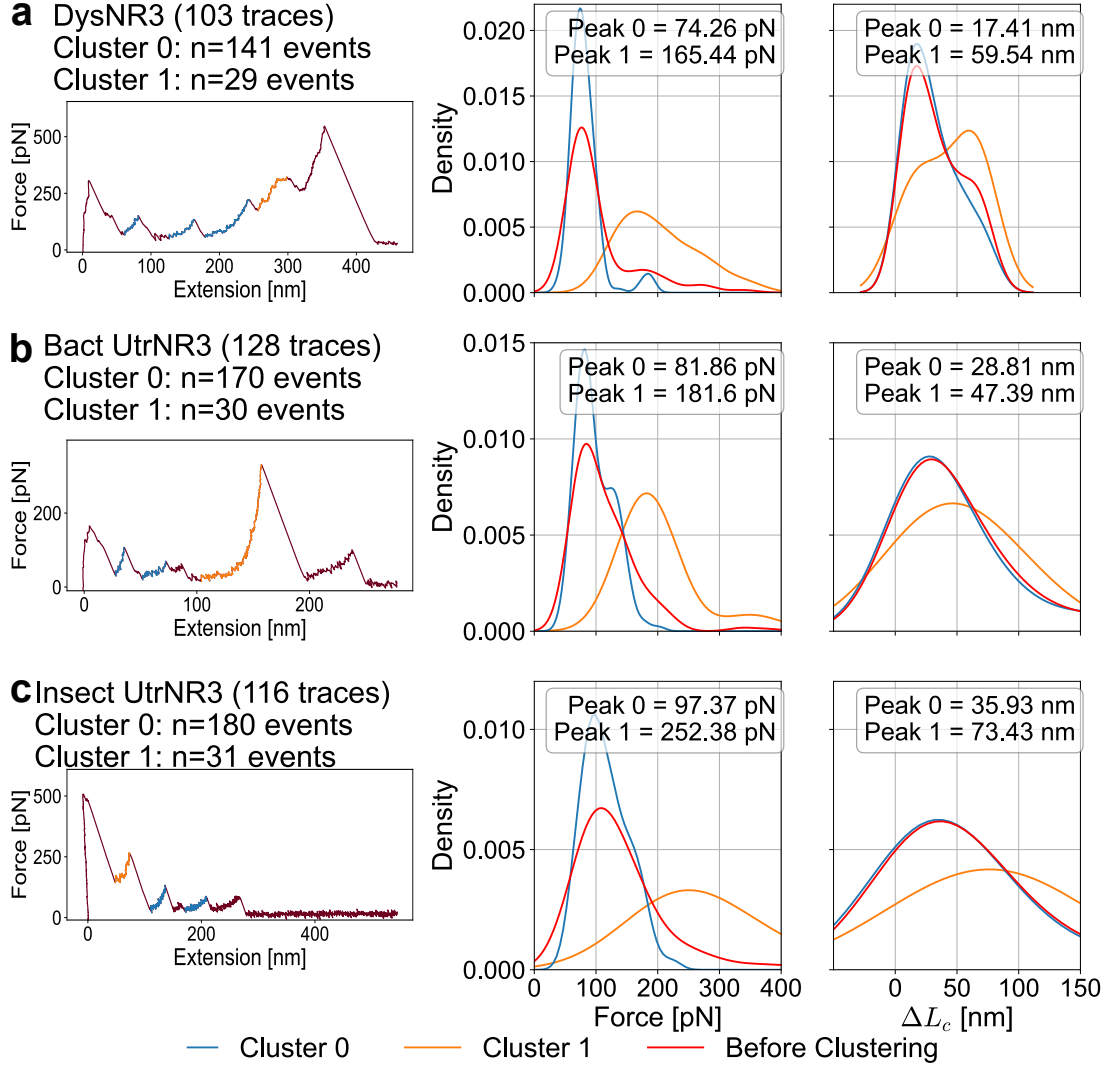


Figure 4.9: Results of natural protein molecules. Clustering statistics at 1000 nm/s for (a) DysNR3, (b) Bact UtrNR3, and (c) Insect UtrNR3.

The analysis pipeline follows the same procedure as described for ddFLN4–Titin I27. Clustering statistics at a pulling speed of 1000 nm/s are presented in Figure 4.9 for DysNR3, bacterial UtrNR3, and insect UtrNR3, from top to bottom, alongside representative force-extension curves. Results for 500 and 2000 nm/s are provided in

Figure 4.12. Across all pulling speeds and constructs, Cluster 1 consistently exhibits higher contour length increases ΔL_c and higher unfolding forces compared to Cluster 0. This observation supports our assumption that Cluster 1 corresponds to the NT domain, while Cluster 0 represents the SLRs. The difference in ΔL_c between the two clusters is at least approximately 18 nm across all speeds and constructs. For DysNR3, the most probable unfolding force differences between clusters are 68 pN at 500 nm/s, 91 pN at 1000 nm/s, and 73 pN at 2000 nm/s. Furthermore, unfolding forces increase with higher pulling speeds across all three constructs, consistent with biophysical behavior.

4.7 Discussion and Conclusion

In this study, we provide the first automated approach for identifying heterogeneous domains, leveraging both existing clustering models and a novel physics-aware deep clustering architecture, LatentUnfold. We validate our method on a synthetic protein composed of two well-characterized domains, showing that the recovered mechanical properties closely match previously reported values. We also apply our approach to natural proteins, utrophin and dystrophin, successfully uncovering heterogeneous domains exhibiting distinct mechanical behaviors. This facility enables drawing biological inferences that account for the impact of heterogeneity, beyond what was previously possible through an average sense.

4.8 Supporting Information

4.8.1 Pre-processing of experimental data

Because the size of a typical force probe is orders of magnitude larger than that of a single biomolecule, it may come into contact with one or more molecules during an experiment. To reduce the likelihood of multiple molecule interactions, the concentration of biomolecules is typically lowered [3, 38]. However, this introduces a trade-off: while lower concentrations reduce multi-molecule events, they also increase the frequency of measurements with no molecular interaction. To obtain accurate and interpretable data from single-molecule force spectroscopy (SMFS), it is essential to exclude traces arising from both cases—those containing only background noise due to the absence of a

molecule, and those involving multiple molecules, which introduce confounding interpretation.

To ensure data quality in experimental data, we employed a three-step filtering process. First, we apply a physics-augmented deep learning model [119] to pre-filter experimental data and identify those most likely originating from single-molecule measurements. Although deep learning models are not perfectly accurate, this initial screening substantially reduces the number of traces requiring manual inspection. Although deep learning models are not perfectly accurate, this initial screening substantially reduces the number of traces requiring manual inspection. Next, an expert manually reviews the pre-filtered dataset to remove any remaining traces that do not conform to expected single molecule behavior. The combination of deep learning-based preselection and expert validation ensures that the final dataset consists of high-confidence single-molecule measurements. Finally, we filter unfolding segments based on their fitting quality to the worm-like chain (WLC) model. Unfolding segments are first identified by detecting characteristic peaks in the force signal that exceed a threshold of 3σ , where σ is the standard deviation of noise. The first and last significant events typically arises from adhesive forces and detachment events, respectively, as described in Section 4.3, these are excluded from further analysis. The remaining significant peaks are interpreted as domain unfolding events of interest. These segments are then fitted to the WLC model, as defined in Equation 4.5, and only those with high fitting quality are retained for downstream analysis.

4.8.2 Robustness across top-performing models

We applied five top-performing clustering algorithms—LatentUnfold(CNN-LSTM), LatentUnfold(LSTM), K-means(DTW), K-medoids(DTW), and MLP-AE—independently to the experimental dataset. As shown in Figure 4.10, the predicted distributions of unfolding force and contour length increase are highly consistent across these models. Furthermore, the level of disagreement in predicted cluster labels, when compared to LatentUnfold(CNN-LSTM), remains relatively low. These findings support the conclusion that our clustering framework yields robust and consistent results across different algorithms, even in the absence of ground truth labels.

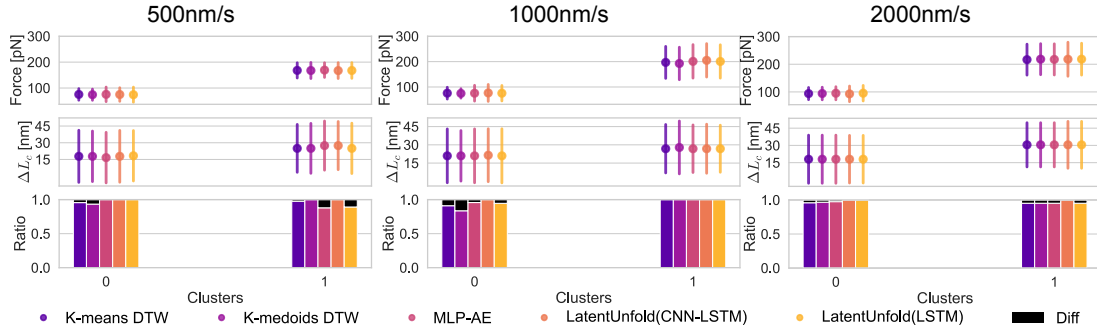


Figure 4.10: Clustering statistics of ddFLN4–Titin I27 across three pulling speeds: 500, 1000, and 2000 nm/s (from left to right). Dots indicate the most probable values, with error bars representing standard deviations. The first row shows the unfolding force distributions; the second row shows the contour length increase distributions; and the third row presents mismatch ratios, computed by treating the predicted labels from LatentUnfold(CNN-LSTM) as the reference and comparing them against those from the other algorithms. Mismatch ratios are shown in black.

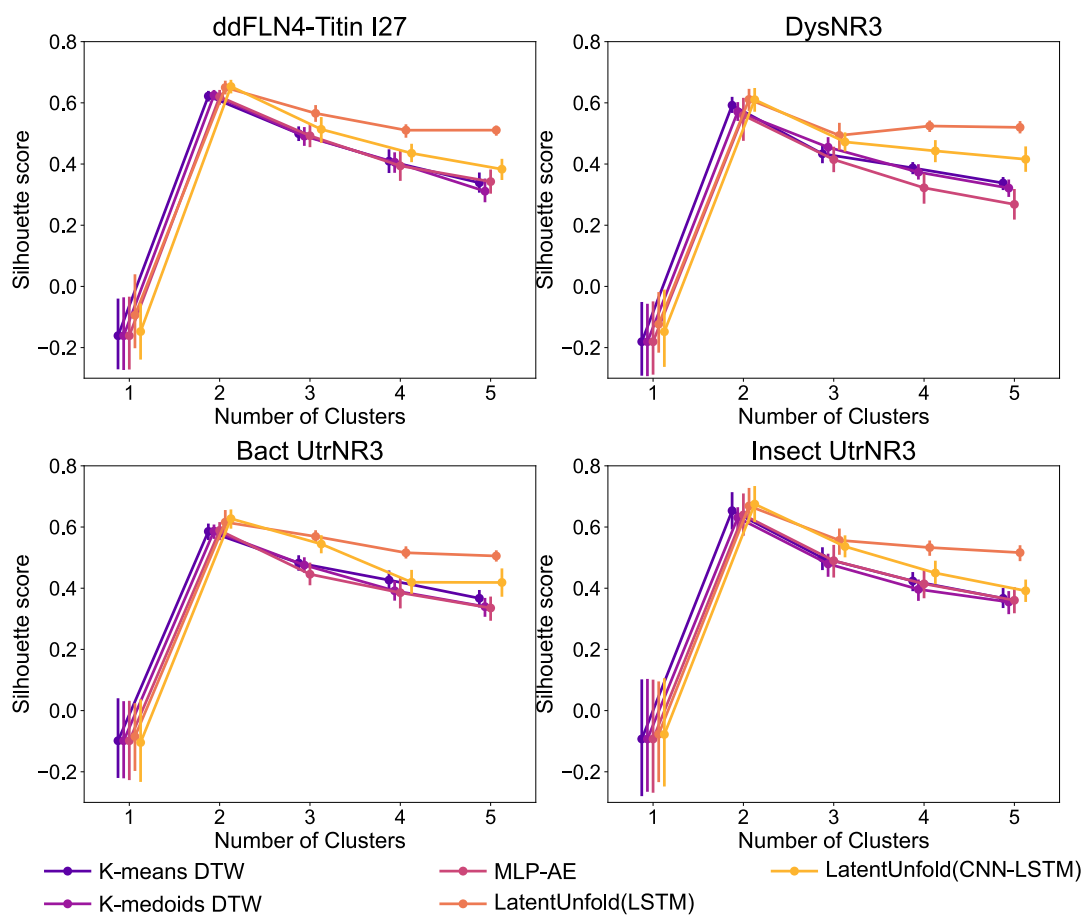


Figure 4.11: Silhouette score vs number of clusters for all four protein molecules: ddFLN4-Titin I27, bact UtrN-R3, insect UtrN-R3, DysN-R3.

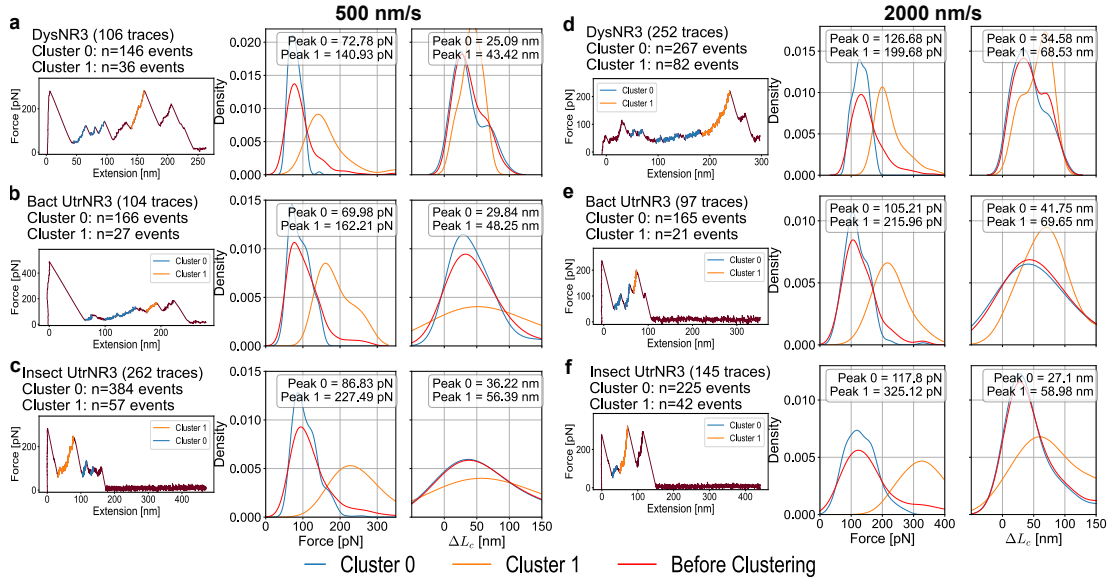


Figure 4.12: Clustering statistics at 500 nm/s for (a) DysNR3, (b) Bact UtrNR3, and (c) Insect UtrNR3, and at 2000 nm/s for (d) DysNR3, (e) Bact UtrNR3, and (f) Insect UtrNR3. The first column shows representative force-extension curves with color-coded segments by predicted cluster labels. The second column displays the unfolding force distributions, and the third column presents the distributions of contour length increase (ΔL_c). Cluster 0 is shown in blue, Cluster 1 in orange, and the combined distribution is shown in red.

Chapter 5

Conclusion and Discussion

In this dissertation, we studied mechanical properties of protein molecules through rigorous single molecule force spectroscopy (SMFS) experiments and the development of deep learning methods. SMFS provides unparalleled insight into how proteins respond to mechanical force at the single molecule level, revealing their functional roles in health and disease. Deep learning methods, in turn, address the complexity, heterogeneity, and scale of SMFS data, enabling more efficient and biologically meaningful interpretation. Together, these experimental and computational approaches form a complementary framework that advances the understanding of protein mechanics.

First, we presented the mechanical characterization of full-length dystrophin and utrophin using atomic force microscopy. This study provided the first mechanical characterization of single full-length dystrophin molecules using two operational modes of atomic force microscopy (AFM); constant speed and constant force. The consistency of results across two operational modes, further corroborated by Monte Carlo simulations, underscores the reliability of the data. Our direct *in vitro* comparisons of Dys and Utr suggest different *in vivo* functions. Notably, although the early unfolding behaviors of Dys and Utr are similar, significant differences arise during the later unfolding events. Given the central role of dystrophin in Duchenne muscular dystrophy (DMD), these findings raise important implications, suggesting that utrophin may not be an ideal therapeutic surrogate for dystrophin deficiency in DMD.

Second, we applied existing deep learning models and developed a physics-augmented

deep learning model for SMFS data identification. This approach effectively filters admissible single molecule data from experiments (distinguishing them from no molecule and multiple molecules events) without requiring expert knowledge. The proposed model also reduces the reliance on manual inspection, paving the way for faster and more consistent analysis of SMFS data. Additionally, this work provides SMFS experimental datasets from four important protein molecules, crucial to many biological pathways, which are made publicly available to support further research.

Third, we introduced the first automated approach for identifying heterogeneous protein folding domains in SMFS data, leveraging both existing clustering models and a novel physics-aware deep clustering architecture. Unlike traditional approaches that rely on drawing inferences in an average sense, this framework enables domain-level inference and accounts for the impact of heterogeneity. Validation with a synthetic protein composed of two well-characterized domains confirmed that the recovered mechanical properties closely match previously reported values. Application to natural proteins, dystrophin and utrophin, further demonstrated the ability of the method to uncover heterogeneous domains with distinct mechanical properties.

Several avenues exist to further extend the work presented in this thesis. One promising direction involves the use of chemical functionalization of probes in combination with molecular fingerprints. Chemical functionalization alters the surface properties of the force probe and substrate, enabling site-specific attachment of target molecules. Molecular fingerprints are well-characterized proteins or domains that exhibit distinct and reproducible unfolding signatures. Incorporating such fingerprints into SMFS trials can provide an internal reference, improving the reliability of single molecule identification and distinguishing them from multi-molecule interactions. Another important direction lies in applying deep learning to investigate domain-domain interdependencies in proteins. Since protein domains are connected in chains, the unfolding of one domain may influence the stability and behavior of others. Understanding these interdependencies represents a critical step toward linking molecular mechanics with its biological function.

In summary, this dissertation demonstrates that both experiments and deep learning methods together provide a powerful and complementary framework for studying the mechanical properties of proteins. Together, these approaches strengthen our understanding of protein behaviors while enabling more efficient analysis.

References

- [1] Olga K. Dudko, Gerhard Hummer, and Attila Szabo. Intrinsic rates and activation free energies from single-molecule pulling experiments. *Physical Review Letters*, 96(10):108101, 2006.
- [2] Vance W. Berger and YanYan Zhou. Kolmogorov–Smirnov Test: Overview. In *Wiley StatsRef: Statistics Reference Online*. John Wiley & Sons, Ltd, 2014.
- [3] Maria Paz Ramirez, Sivaraman Rajaganapathy, Anthony R. Hagerty, Cailong Hua, Gloria C. Baxter, Joseph Vavra, Wendy R. Gordon, Joseph M. Muretta, Murti V. Salapaka, and James M. Ervasti. Phosphorylation alters the mechanical stiffness of a model fragment of the dystrophin homologue utrophin. *Journal of Biological Chemistry*, 299(2):102847, 2023.
- [4] Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. 2006.
- [5] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.
- [6] Byeongseon Yang, Zhaowei Liu, Haipei Liu, and Michael A. Nash. Next Generation Methods for Single-Molecule Force Spectroscopy on Polyproteins and Receptor-Ligand Complexes. *Frontiers in Molecular Biosciences*, 7:85, 2020.
- [7] Carlos Bustamante, Yann R. Chemla, Nancy R. Forde, and David Izhaky. Mechanical Processes in Biochemistry. *Annual Review of Biochemistry*, 73(1):705–748, 2004.

- [8] Matthias Rief, Mathias Gautel, Filipp Oesterhelt, Julio M. Fernandez, and Hermann E. Gaub. Reversible Unfolding of Individual Titin Immunoglobulin Domains by AFM. *Science*, 276(5315):1109–1112, 1997.
- [9] Matthias Rief, Mathias Gautel, Alexander Schemmel, and Hermann E. Gaub. The Mechanical Stability of Immunoglobulin and Fibronectin III Domains in the Muscle Protein Titin Measured by Atomic Force Microscopy. *Biophysical Journal*, 75(6):3008–3014, 1998.
- [10] Sivaraman Rajaganapathy, Jackie L. McCourt, Sayan Ghosal, Angus Lindsay, Preston M. McCourt, Dawn A. Lowe, James M. Ervasti, and Murti V. Salapaka. Distinct mechanical properties in homologous spectrin-like repeats of utrophin. *Scientific reports*, 9(1):5210, 2019.
- [11] Brenda Gerull, Michael Gramlich, John Atherton, Mark McNabb, Karoly Trombitás, Sabine Sasse-Klaassen, J.G. Seidman, Christine Seidman, Henk Granzier, Siegfried Labeit, Michael Frenneaux, and Ludwig Thierfelder. Mutations of TTN, encoding the giant muscle filament titin, cause familial dilated cardiomyopathy. *Nature Genetics*, 30(2):201–204, 2002.
- [12] Daniel S. Herman, Lien Lam, Matthew R.G. Taylor, Libin Wang, Polakit Teekakirikul, Danos Christodoulou, Lauren Conner, Steven R. DePalma, Barbara McDonough, Elizabeth Sparks, Debbie Lin Teodorescu, Allison L. Cirino, Nicholas R. Banner, Dudley J. Pennell, Sharon Graw, Marco Merlo, Andrea Di Lenarda, Gianfranco Sinagra, J. Martijn Bos, Michael J. Ackerman, Richard N. Mitchell, Charles E. Murry, Neal K. Lakdawala, Carolyn Y. Ho, Paul J.R. Barton, Stuart A. Cook, Luisa Mestroni, J.G. Seidman, and Christine E. Seidman. Truncations of Titin Causing Dilated Cardiomyopathy. *New England Journal of Medicine*, 366(7):619–628, 2012.
- [13] Jerry R. Mendell, Chris Shilling, Nancy D. Leslie, Kevin M. Flanigan, Roula al-Dahhak, Julie Gastier-Foster, Kelley Kneile, Diane M. Dunn, Brett Duval, and Alexander Aoyagi. Evidence-based path to newborn screening for Duchenne muscular dystrophy. *Annals of neurology*, 71(3):304–313, 2012.

- [14] James M. Ervasti. Dystrophin, its interactions with other proteins, and implications for muscular dystrophy. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1772(2):108–117, 2007.
- [15] Rubén Hervás, Javier Oroz, Albert Galera-Prat, Oscar Goñi, Alejandro Valbuena, Andrés M. Vera, Àngel Gómez-Sicilia, Fernando Losada-Urzáiz, Vladimir N. Uversky, Margarita Menéndez, Douglas V. Laurents, Marta Bruix, and Mariano Carrión-Vázquez. Common Features at the Start of the Neurodegeneration Cascade. *PLoS Biology*, 10(5), 2012.
- [16] Francesco Simone Ruggeri, Fabrizio Benedetti, Tuomas P. J. Knowles, Hilal A. Lashuel, Sergey Sekatskii, and Giovanni Dietler. Identification and nanomechanical characterization of the fundamental single-strand protofilaments of amyloid-synuclein fibrils. *Proceedings of the National Academy of Sciences*, 115(28):7230–7235, 2018.
- [17] G. Binnig, C. F. Quate, and Ch. Gerber. Atomic Force Microscope. *Physical Review Letters*, 56(9):930–933, 1986.
- [18] Arthur Ashkin, James M. Dziedzic, John E. Bjorkholm, and Steven Chu. Observation of a single-beam gradient force optical trap for dielectric particles. *Optics letters*, 11(5):288–290, 1986.
- [19] Eric P Hoffman, Robert H Brown, and Louis M Kunkel. Dystrophin: The Protein Product of the Duchenne Muscular Dystrophy Locus. *Cell*, 51(6):919–928, 1987.
- [20] Simon Guiraud, Huijia Chen, David T. Burns, and Kay E. Davies. Advances in genetic therapeutic strategies for Duchenne muscular dystrophy. *Experimental Physiology*, 100(12):1458–1467, 2015.
- [21] Brian K. Shoichet, Walter A. Baase, Ryota Kuroki, and Brian W. Matthews. A relationship between protein stability and protein function. *Proceedings of the National Academy of Sciences*, 92(2):452–456, 1995.
- [22] H. Jane Dyson and Peter E. Wright. Intrinsically unstructured proteins and their functions. *Nature reviews Molecular cell biology*, 6(3):197–208, 2005.

- [23] James M. Ervasti, Kay Ohlendieck, Steven D. Kahl, Mitchell G. Gaver, and Kevin P. Campbell. Deficiency of a glycoprotein component of the dystrophin complex in dystrophic muscle. *Nature*, 345(6273):315–319, 1990.
- [24] Boaz Weller, George Karpati, and Stirling Carpenter. Dystrophin-deficient mdx muscle fibers are preferentially vulnerable to necrosis induced by experimental lengthening contractions. *Journal of the Neurological Sciences*, 100(1-2):9–13, 1990.
- [25] B J Petrof, J B Shrager, H H Stedman, A M Kelly, and H L Sweeney. Dystrophin protects the sarcolemma from stresses developed during muscle contraction. *Proceedings of the National Academy of Sciences USA*, 90(8):3710–3714, 1993.
- [26] Steven J Winder. REVIEW: The membrane–cytoskeleton interface: the role of dystrophin and utrophin. *Journal of Muscle Research Cell Motility*, 18:617–629, 1994.
- [27] C Pasternak, S Wong, and E L Elson. Mechanical function of dystrophin in muscle cells. *Journal of Cell Biology*, 128(3):355–361, 1995.
- [28] S. J. Winder, L. Hemmings, S. K. Maciver, S. J. Bolton, J. M. Tinsley, K. E. Davies, D. R. Critchley, and J. Kendrick-Jones. Utrophin actin binding domain: analysis of actin binding and cellular targeting. *Journal of Cell Science*, 108(1):63–71, 1995.
- [29] Shimin Le, Miao Yu, Ladislav Hovan, Zhihai Zhao, James Ervasti, and Jie Yan. Dystrophin As a Molecular Shock Absorber. *ACS Nano*, 12(12):12140–12148, 2018.
- [30] Inna N. Rybakova, Jitandrakumar R. Patel, and James M. Ervasti. The Dystrophin Complex Forms a Mechanically Strong Link between the Sarcolemma and Costameric Actin. *Journal of Cell Biology*, 150(5):1209–1214, 2000.
- [31] C. Rigoletto, A. Prella, P. Ciscato, M. Moggio, G. Comi, F. Fortunato, and G. Scarlato. Utrophin expression during human fetal development. *International Journal of Developmental Neuroscience*, 13(6):585–593, 1995.

- [32] Kay Ohlendieck, James M. Ervasti, Kiichiro Matsumura, Steven D. Kahl, Cynthia J. Leveille, and Kevin P. Campbell. Dystrophin-related protein is localized to neuromuscular junctions of adult skeletal muscle. *Neuron*, 7(3):499–508, 1991.
- [33] Jonathon Tinsley, Nicolas Deconinck, Rosie Fisher, David Kahn, Steve Phelps, Jean-Marie Gillis, and Kay Davies. Expression of full-length utrophin prevents muscular dystrophy in mdx mice. *Nat. Med.*, 4(12):1441–1444, 1998.
- [34] Olga K. Dudko, Gerhard Hummer, and Attila Szabo. Theory, analysis, and interpretation of single-molecule force spectroscopy experiments. *Proceedings of the National Academy of Sciences USA*, 105(41):15755–15760, 2008.
- [35] Ivet Bahar, Robert L. Jernigan, and Ken A. Dill. *Protein actions: Principles and modeling*. Garland Science, 2017.
- [36] José Nelson Onuchic, Zaida Luthey-Schulten, and Peter G. Wolynes. Theory of protein folding: the energy landscape perspective. *Annual Review of Physical Chemistry*, 48(1):545–600, 1997.
- [37] John E. Sader, James W. M. Chon, and Paul Mulvaney. Calibration of rectangular atomic force microscope cantilevers. *Review of Scientific Instruments*, 70(10):3967–3969, 1999.
- [38] Andres F Oberhauser, Paul K Hansma, Mariano Carrion-Vazquez, and Julio M Fernandez. Stepwise unfolding of titin under force-clamp atomic force microscopy. *Proceedings of the National Academy of Sciences USA*, 98(2):468–472, 2001.
- [39] Matthias Rief, Jaime Pascual, Matti Saraste, and Hermann E Gaub. Single molecule force spectroscopy of spectrin repeats: low unfolding forces in helix bundles. *Journal of Molecular Biology*, 286(2):553–561, 1999.
- [40] Hendrik Anthony Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.
- [41] Gerhard Hummer and Attila Szabo. Kinetics from nonequilibrium single-molecule pulling experiments. *Biophysical journal*, 85(1):5–15, 2003.

- [42] Phillip J. Elms, John D. Chodera, Carlos J. Bustamante, and Susan Marqusee. Limitations of Constant-Force-Feedback Experiments. *Biophysical Journal*, 103(7):1490–1499, 2012.
- [43] David R. Jacobson and Thomas T. Perkins. Correcting molecular transition rates measured by single-molecule force spectroscopy for limited temporal resolution. *Physical Review E*, 102(2):022402, 2020.
- [44] George I. Bell. Models for the specific adhesion of cells to cells: a theoretical framework for adhesion mediated by reversible bonds between cell surface molecules. *Science*, 200(4342):618–627, 1978.
- [45] David J. Brockwell, Godfrey S. Beddard, Emanuele Paci, Dan K. West, Peter D. Olmsted, D. Alastair Smith, and Sheena E. Radford. Mechanically Unfolding the Small, Topologically Simple Protein L. *Biophysical Journal*, 89(1):506–519, 2005.
- [46] Byeongseon Yang, Haipai Liu, Zhaowei Liu, Regina Doenen, and Michael A. Nash. Influence of Fluorination on Single-Molecule Unfolding and Rupture Pathways of a Mechanostable Protein Adhesion Complex. *Nano Letters*, 20(12):8940–8950, 2020.
- [47] Nishant Bhasin, Richard Law, George Liao, Daniel Safer, Jennifer Ellmer, Bohdana M. Discher, H. Lee Sweeney, and Dennis E. Discher. Molecular Extensibility of Mini-dystrophins and a Dystrophin Rod Construct. *Journal of Molecular Biology*, 352(4):795–806, 2005.
- [48] Christine Carag Krieger, Nishant Bhasin, Manorama Tewari, Andre E. X. Brown, Daniel Safer, H. Lee Sweeney, and Dennis E. Discher. Exon-skipped dystrophins for treatment of Duchenne muscular dystrophy: Mass spectrometry mapping of most exons and cooperative domain designs based on single molecule mechanics. *Cytoskeleton*, 67(12):796–807, 2010.
- [49] Scott Q. Harper, Robert W. Crawford, Christiana DelloRusso, and Jeffrey S. Chamberlain. Spectrin-like repeats from dystrophin and α -actinin-2 are not functionally interchangeable. *Human Molecular Genetics*, 11(16):1807–1815, 2002.

- [50] Joseph J. Belanto, Tara L. Mader, Michael D. Eckhoff, Dana M. Strandjord, Glen B. Banks, Melissa K. Gardner, Dawn A. Lowe, and James M. Ervasti. Microtubule binding distinguishes dystrophin from utrophin. *Proceedings of the National Academy of Sciences USA*, 111(15):5723–5728, 2014.
- [51] Davin M. Henderson, Joseph J. Belanto, Bin Li, Hanke Heun-Johnson, and James M. Ervasti. Internal deletion compromises the stability of dystrophin. *Human Molecular Genetics*, 20(15):2955–2963, 2011.
- [52] Mariano Carrion-Vazquez, Andres F. Oberhauser, Susan B. Fowler, Piotr E. Marszalek, Sheldon E. Broedel, Jane Clarke, and Julio M. Fernandez. Mechanical and chemical unfolding of a single protein: A comparison. *Proceedings of the National Academy of Sciences USA*, 96(7):3694–3699, 1999.
- [53] Muralidharan Muthu, Kylie A. Richardson, and Andrew J. Sutherland-Smith. The Crystal Structures of Dystrophin and Utrophin Spectrin Repeats: Implications for Domain Boundaries. *PLoS ONE*, 7(7):e40066, 2012.
- [54] Fiona Lm Norwood, Andrew J Sutherland-Smith, Nicholas H Keep, and John Kendrick-Jones. The structure of the N-terminal actin-binding domain of human dystrophin and how mutations in this domain may cause Duchenne or Becker muscular dystrophy. *Structure*, 8(5):481–491, 2000.
- [55] Julia L Zimmermann, Thomas Nicolaus, Gregor Neuert, and Kerstin Blank. Thiol-based, site-specific and covalent immobilization of biomolecules for single-molecule experiments. *Nature Protocols*, 5(6):975–985, 2010.
- [56] Thomas Bornschlöggl and Matthias Rief. Single-Molecule Protein Unfolding and Refolding Using Atomic Force Microscopy. In Erwin J. G. Peterman and Gijs J. L. Wuite, editors, *Single Molecule Analysis*, volume 783, pages 233–250. Springer, 2011.
- [57] Yuri L. Lyubchenko. *Nanoscale Imaging: Methods and Protocols*. Springer, 2018.
- [58] Cailong Hua, Rebecca A. Slick, Joseph Vavra, Joseph M. Muretta, James M. Ervasti, and Murti V. Salapaka. Two operational modes of atomic force microscopy

reveal similar mechanical properties for homologous regions of dystrophin and utrophin. *bioRxiv*, 2024.

- [59] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-based pattern recognition: third international workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, pages 84–92. Springer, 2015.
- [60] Joshua R. Waite, Sin Yong Tan, Homagni Saha, Soumik Sarkar, and Anwesha Sarkar. Few-shot deep learning for AFM force curve characterization of single-molecule interactions. *Patterns*, 4(1):100672, 2023.
- [61] Vanni Doffini, Haipai Liu, Zhaowei Liu, and Michael A. Nash. Iterative Machine Learning for Classification and Discovery of Single-Molecule Unfolding Trajectories from Force Spectroscopy Data. *Nano Letters*, 23(22):10406–10413, 2023.
- [62] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.
- [63] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.
- [64] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.
- [65] Fabio L. Leite, Carolina C. Bueno, Alessandra L. Da Róz, Ervino C. Ziemath, and Osvaldo N. Oliveira. Theoretical Models for Surface Forces and Adhesion and Their Measurement Using Atomic Force Microscopy. *International Journal of Molecular Sciences*, 13(12):12773–12856, 2012.

- [66] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [67] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [68] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate LSTM-FCNs for time series classification. *Neural Networks*, 116:237–245, 2019.
- [69] Tuan-Anh Pham, Jong-Hoon Lee, and Choong-Shik Park. MST-VAE: Multi-Scale Temporal Variational Autoencoder for Anomaly Detection in Multivariate Time Series. *Applied Sciences*, 12(19):10078, 2022.
- [70] Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. TapNet: Multivariate Time Series Classification with Attentional Prototypical Network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6845–6852, 2020.
- [71] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J. Leon Zhao. Exploiting multi-channels deep convolutional neural networks for multivariate time series classification. *Frontiers of Computer Science*, 10(1):96–112, 2016.
- [72] Seyed Navid Mohammadi Foumani, Chang Wei Tan, and Mahsa Salehi. Disjoint-cnn for multivariate time series classification. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 760–769. IEEE, 2021.
- [73] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [74] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. LSTM neural networks for language modeling. In *Interspeech 2012*, pages 194–197. ISCA, 2012.
- [75] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

- [76] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. *International Journal of Forecasting*, 37(1):388–427, 2021.
- [77] C. Bustamante, J. F. Marko, E. D. Siggia, and S. Smith. Entropic Elasticity of -Phage DNA. *Science*, 265(5178):1599–1600, 1994.
- [78] C. Ortiz and G. Hadziioannou. Entropic Elasticity of Single Polymer Chains of Poly(methacrylic acid) Measured by Atomic Force Microscopy. *Macromolecules*, 32(3):780–787, 1999.
- [79] L. Livadaru, R. R. Netz, and H. J. Kreuzer. Stretching Response of Discrete Semiflexible Polymers. *Macromolecules*, 36(10):3732–3744, 2003.
- [80] Elias M. Puchner, Gereon Franzen, Mathias Gautel, and Hermann E. Gaub. Comparing Proteins by Their Unfolding Pattern. *Biophysical Journal*, 95(1):426–434, 2008.
- [81] Zhaowei Liu, Haipai Liu, Andrés M. Vera, Rafael C. Bernardi, Philip Tinnefeld, and Michael A. Nash. High force catch bond mechanism of bacterial adhesion in the human gut. *Nature Communications*, 11(1):4321, 2020.
- [82] Markus A. Jobst, Constantin Schoeler, Klara Malinowska, and Michael A. Nash. Investigating Receptor-ligand Systems of the Cellulosome with AFM-based Single-molecule Force Spectroscopy. *Journal of Visualized Experiments*, (82):50950, 2013.
- [83] Marcus Otten, Wolfgang Ott, Markus A Jobst, Lukas F Milles, Tobias Verdorfer, Diana A Pippig, Michael A Nash, and Hermann E Gaub. From genes to protein mechanics on a chip. *Nature Methods*, 11(11):1127–1130, 2014.
- [84] Song Zhang, Hu-jun Qian, Zhonghua Liu, Hongyu Ju, Zhong-yuan Lu, Haiming Zhang, Lifeng Chi, and Shuxun Cui. Towards Unveiling the Exact Molecular Structure of Amorphous Red Phosphorus by Single-Molecule Studies. *Angewandte Chemie International Edition*, 58(6):1659–1663, 2019.
- [85] Constantin Schoeler, Klara H. Malinowska, Rafael C. Bernardi, Lukas F. Milles, Markus A. Jobst, Ellis Durner, Wolfgang Ott, Daniel B. Fried, Edward A. Bayer,

- Klaus Schulten, Hermann E. Gaub, and Michael A. Nash. Ultrastable cellulosome-adhesion complex tightens under load. *Nature Communications*, 5(1):5635, 2014.
- [86] Minghao Liu, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang Wang, and Wei Song. Gated transformer networks for multivariate time series classification. *arXiv preprint arXiv:2103.14438*, 2021.
- [87] Lifang Wang, Jin Zhang, Yang Liu, Jia Mi, and Jiong Zhang. Multimodal Medical Image Fusion Based on Gabor Representation Combination of Multi-CNN and Fuzzy Neural Network. *IEEE Access*, 9:67634–67647, 2021.
- [88] Yunfan Chen, Han Xie, and Hyunchul Shin. Multi-layer fusion techniques using a CNN for multispectral pedestrian detection. *IET Computer Vision*, 12(8):1179–1187, 2018.
- [89] William T. King, Meihong Su, and Guoliang Yang. Monte Carlo simulation of mechanical unfolding of proteins based on a simple two-state model. *International Journal of Biological Macromolecules*, 46(2):159–166, 2010.
- [90] Jacob N. Israelachvili. *Intermolecular and surface forces*. Academic press, 2011.
- [91] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.
- [92] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access*, 6:1662–1669, 2018.
- [93] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [94] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J. Király. sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872*, 2019.
- [95] Diederik P. Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [96] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. Tsllearn, A Machine Learning Toolkit for Time Series Data. 2020.
- [97] Cecilia Summers and Michael J. Dinneen. Improved mixed-example data augmentation. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1262–1270. IEEE, 2019.
- [98] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-Class Learning for Image Classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, Salt Lake City, UT, 2018. IEEE.
- [99] Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. *Advances in neural information processing systems*, 33:19365–19376, 2020.
- [100] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282*, 2017.
- [101] Megan L Hughes and Lorna Dougan. The physics of pulling polyproteins: a review of single molecule force spectroscopy using the AFM to study protein unfolding. *Reports on Progress in Physics*, 79(7):076601, 2016.
- [102] Amy EM Beedle, Ainhoa Lezamiz, Guillaume Stirnemann, and Sergi Garcia-Manyes. The mechanochemistry of copper reports on the directionality of unfolding in model cupredoxin proteins. *Nature communications*, 6(1):7894, 2015.

- [103] Andres F. Oberhauser, Carmelu Badilla-Fernandez, Mariano Carrion-Vazquez, and Julio M. Fernandez. The mechanical hierarchies of fibronectin observed with single-molecule AFM. *Journal of molecular biology*, 319(2):433–447, 2002.
- [104] Rafael Tapia-Rojo, Marc Mora, and Sergi Garcia-Manyes. Single-molecule magnetic tweezers to probe the equilibrium dynamics of individual proteins at physiologically relevant forces and timescales. *Nature protocols*, 19(6):1779–1806, 2024.
- [105] Hema Chandra Kotamarthi, Riddhi Sharma, Satya Narayan, Sayoni Ray, and Sri Rama Koti Ainavarapu. Multiple unfolding pathways of leucine binding protein (LBP) probed by single-molecule force spectroscopy (SMFS). *Journal of the American Chemical Society*, 135(39):14768–14774, 2013.
- [106] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering—a decade review. *Information systems*, 53:16–38, 2015.
- [107] Vit Niennattrakul and Chotirat Ann Ratanamahatana. On clustering multimedia time series data using k-means and dynamic time warping. In *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE’07)*, pages 733–738. IEEE, 2007.
- [108] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.
- [109] Hamid Hadipour, Chengyou Liu, Rebecca Davis, Silvia T. Cardona, and Pingzhao Hu. Deep clustering of small molecules at large-scale via variational autoencoder embedding and K-means. *BMC bioinformatics*, 23(Suppl 4):132, 2022.
- [110] Pengzhi Li, Yan Pei, and Jianqiang Li. A comprehensive survey on design and application of autoencoder in deep learning. *Applied Soft Computing*, 138:110176, 2023.
- [111] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

- [112] Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. Learning Representations for Time Series Clustering. *Advances in neural information processing systems*, 32, 2019.
- [113] Ivica Letunic and Peer Bork. 20 years of the SMART protein domain annotation resource. *Nucleic acids research*, 46(D1):D493–D496, 2018.
- [114] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.
- [115] Naveen Sai Madiraju. *Deep temporal clustering: Fully unsupervised learning of time-domain features*. PhD thesis, Arizona State University, 2018.
- [116] Hongyuan Zha, Xiaofeng He, Chris Ding, Ming Gu, and Horst D Simon. Spectral Relaxation for K-means Clustering. *Advances in neural information processing systems*, 14, 2001.
- [117] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):568–586, 2010.
- [118] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [119] Cailong Hua, Sivaraman Rajaganapathy, Rebecca A. Slick, Joseph Vavra, Joseph M. Muretta, James M. Ervasti, and Murti Salapaka. A Physics-Augmented Deep Learning Framework for Classifying Single Molecule Force Spectroscopy Data. In *Forty-second International Conference on Machine Learning*, 2025.
- [120] Ingo Schwaiger, Angelika Kardinal, Michael Schleicher, Angelika A Noegel, and Matthias Rief. A mechanical unfolding intermediate in an actin-crosslinking protein. *Nature Structural & Molecular Biology*, 11(1):81–85, 2004.

- [121] Quan Q. Gao. The Dystrophin Complex: Structure, Function, and Implications for Therapy. In *Comprehensive Physiology*, pages 1223–1239. Wiley, 1 edition, 2015.
- [122] Hendrik Dietz and Matthias Rief. Protein structure by mechanical triangulation. *Proceedings of the National Academy of Sciences*, 103(5):1244–1247, 2006.