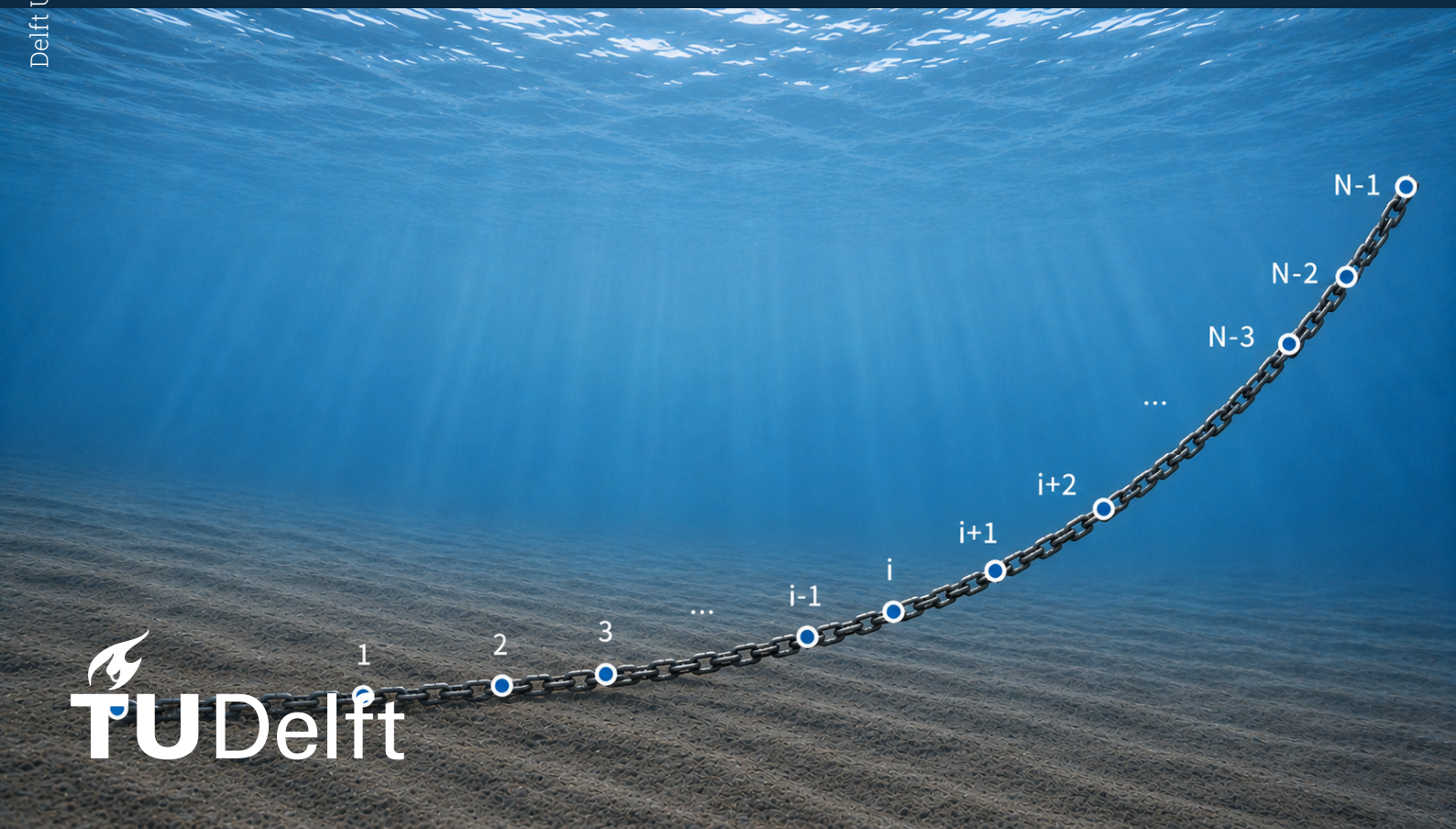


Graph-Attention Long-Short-Term-Memory Network (GAT+LSTM) for Mooring Line Load-Field Estimation Across Variable Topologies

Athanasios Konstantaras



Graph-Attention Long-Short-Term- Memory Network (GAT+LSTM) for Mooring Line Load-Field Estimation Across Variable Topologies

by

Athanasios Konstantaras

Student Number:	6054757
Supervisor TU Delft:	Dr. ir. J.O. Colomés Gené
Thesis Committee:	Dr. J.P. Aguilar Lopez
Faculty:	Offshore & Dredging Engineering, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.
The majority of the code used in this thesis is available in the following GitHub repository,
https://github.com/AthKon/Thesis_GAT_LSTM.

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Preface

This thesis completes my Master of Science in Offshore and Dredging Engineering at Delft University of Technology. It started from a straightforward question, whether a neural network could stand in for a finite-element mooring solver, and it ended somewhere rather more interesting, with a careful account of what such a surrogate genuinely knows and where its knowledge stops. Establishing that boundary honestly took considerably longer than producing the first encouraging number, and it is the part of the work I am most glad to have done.

I am grateful to my supervisor, Dr. ir. Oriol Colomés Gené, for the trust to let this project follow the problem wherever it led, and for the guidance and expertise offered at moments that were crucial for the continuation of the project. I would also like to thank Dr. Juan Aguilar Lopez for serving on my thesis committee and for the time spent reading and assessing this work.

This research rests on simulation data that I did not generate myself. I am indebted to Shagun Agarwal, the main author of the geometrically nonlinear finite-element mooring model, who generated the simulations and made the dataset available to me, and to the authors of the environmental sampling framework used to define the sea states. A dataset of this scale would not have been possible otherwise. The training runs and hyperparameter studies were carried out on the DelftBlue supercomputer, and I thank the Delft High Performance Computing Centre for the computational resources and for the support that kept long jobs running.

To my friends in Delft, thank you for the support in the difficult moments and for the great experiences we shared over these past two years. To my family, thank you for the patience and the unwavering support across the whole of my studies. None of this would have reached an end without you.

Finally, to the reader, I hope the limitations set out in these pages prove as useful as the results that surround them.

*Thanos Konstantaras
Delft, July 2026*

Summary

Reliable prediction of mooring-line motion and tension underpins the station-keeping safety of every floating offshore platform, whether built for wind energy, wave energy, or oil and gas production. High-fidelity finite-element solvers resolve the nonlinear dynamics that govern this safety, including snap loads and peak tensions, but their computational cost makes repeated evaluation across design iterations or environmental conditions impractical. Deep-learning approaches promise to overcome this cost. Yet existing mooring-line surrogates either frame the problem as failure classification or reduce the line to a lumped fairlead reaction force, so none reconstructs the full spatial field of motion and tension along lines of different discretisations with a single trained model, nor biases its predictions toward the conservative peak estimates that structural design requires.

This thesis develops a surrogate that couples a graph attention network (GAT) spatial encoder to a long short-term memory (LSTM) temporal encoder, denoted GAT+LSTM, and that addresses this gap. A mooring line is represented as a chain graph whose nodes and edges carry static geometric and mechanical features alongside dynamic motion, tension, and contact features. A two-layer GATv2 encoder captures the coupling between neighbouring nodes, a per-node LSTM encoder captures each node's temporal evolution, and a compact prediction head returns the future motion and tension of the whole line in one non-autoregressive pass. The network is trained under a mixed-topology scheme on finite-element simulations spanning ten offshore locations and node counts from $N = 4$ to $N = 21$, with two locations withheld to test generalisation, under two compared training objectives, a fixed-weight loss combining data fidelity with soft physical penalty terms in physical units, and a Kendall multi-task uncertainty-weighting scheme intended to learn those weights automatically.

One set of trained weights serves every discretisation from $N = 4$ to $N = 21$ nodes and both withheld locations, with test accuracy meeting or exceeding validation accuracy at every node count. The surrogate reconstructs the spatial field with a global tension R^2 of 0.98. A variance decomposition shows this accuracy certifies the response's spatial profile and per-sea-state level, together at least 98.5 % of its variance, not the wave-frequency fluctuation, which the model returns as a near-flat mean. Under the fixed-weight objective this yields a positive mean peak-tension bias of +197.5 N and an underprediction rate of 17.6 %. The uncertainty-weighted alternative matches this safety margin only once its one-sided penalty is reformulated as a two-sided quantile loss, and even then it does not surpass it.

The thesis therefore contributes a validated, topology-general load-field surrogate, a safety-aware training methodology with the diagnosis of its failure mode, and a rigorous characterisation of what the surrogate's accuracy does and does not certify. The surrogate is deliberately not a wave-by-wave forecaster of mooring tension. Its value lies in reproducing the spatial load field, across topologies and locations, at a fraction of the finite-element solver's computational cost.

Contents

Preface	i
Summary	ii
1 Introduction	1
1.1 Background	1
1.2 Mooring Lines and Underlying Physics	2
1.2.1 Mooring Configurations	2
1.2.2 Environmental Loads	2
1.2.3 Governing Equations of Motion	5
1.3 Finite-Element Solvers and Lumped-Mass Approaches	6
1.3.1 Lumped-Mass Method	6
1.3.2 Finite-Element Method: Slender-Rod Theory	7
1.3.3 Industry Simulation Software	8
1.4 Computational Limitations and the Case for Deep Learning	9
1.4.1 The Fidelity-Efficiency Trade-off	9
1.4.2 Deep Learning as a Surrogate Modelling Solution	10
1.4.3 Theoretical Foundations of Graph Neural Networks	11
1.5 Research Gap and Thesis Contribution	12
1.5.1 Research Gap	12
1.5.2 Research Questions	13
1.5.3 Contribution and Scope	13
1.6 Research Methodology	13
2 Data	16
2.1 Simulation Framework	16
2.1.1 Continuum Formulation and Kinematics	16
2.1.2 Strain and Stress Measures	17
2.1.3 Dynamic Equilibrium and External Loading	17
2.1.4 Weak Form and Finite-Element Discretisation	18
2.1.5 Time Integration and Nonlinear Solution	19
2.2 Environmental Condition Sampling	19
2.3 Raw Data Structure	21
2.4 Preprocessing	21
2.5 Data Quality and Corrupt Case Handling	22
2.6 Dataset Construction	24
3 Model Architecture and Methodology	26
3.1 Problem Formulation	26
3.2 Graph Representation of the Mooring Line	27
3.3 Feature Engineering	28
3.3.1 Static Node Features	28
3.3.2 Dynamic Node Features	29
3.3.3 Static Edge Features	31
3.3.4 Dynamic Edge Features	31
3.3.5 Normalisation	32
3.4 Spatial Encoder	32
3.5 Temporal Encoder	35
3.6 Prediction Head	36
3.7 Mixed-Topology Training	37

3.8	Loss Function and Evaluation Metrics	38
3.8.1	Data Loss	38
3.8.2	Evaluation Metrics	39
3.8.3	Physical Loss Terms	40
3.8.4	Multi-Task Uncertainty Weighting	41
3.9	Stage B Hyperparameter Optimisation	41
3.10	Final Hyperparameter Configuration	43
4	Model Performance	45
4.1	Experimental Setup	45
4.2	Primary Model Held-out Test Performance	46
4.3	Load-Field Reconstruction and Topology Generalisation	48
4.3.1	Per-Topology and OOD Generalisation	48
4.3.2	Spatial Reconstruction of the Load Field	50
4.4	A Spatial–Temporal Capability Anatomy	54
4.4.1	The Architectural Ceiling	54
4.4.2	The Variance Anatomy	55
4.4.3	The Temporal Blind Spot	56
4.4.4	Why Spatial but Not Temporal: the Mechanism	59
4.4.5	Comparison with a Persistence Baseline	60
4.5	Multi-task Uncertainty Weighting	61
4.5.1	Learned Uncertainties and the Reproducible Term Ranking	62
4.5.2	The σ -Collapse Failure Mode and the Pinball Fix	63
4.5.3	Accuracy versus Safety	64
4.6	Computational Time	67
4.7	Summary	67
5	Discussion	69
5.1	A Spatial Load-Field Surrogate, Not a Forecaster	69
5.2	Physical Consistency and Peak Safety	69
5.3	Topology Generalisation	69
5.4	The Capability Boundary and Its Two Regimes	70
5.5	What the Learned Uncertainties Reveal	70
5.6	Limitations	70
6	Conclusion	71
6.1	Answering the Research Questions	71
6.2	Contributions	72
7	Recommended Future Work	73
7.1	Reconstructing the Full Load Field from Sparse Motion Sensing	73
7.2	Forcing-Free Forecasting of Future Tension	74
	References	76

List of Figures

1.1	Principal offshore mooring-line configurations [9].	2
1.2	Loads acting on a floating structure [9].	3
1.3	An irregular sea state represented as the sum of many regular wave components [9].	4
1.4	Forces and displacement acting on an element of a mooring line [1].	5
1.5	Mooring-line discretisation and node indexing under the Lumped-Mass Method [14].	6
1.6	A single node aggregating messages from its local graph neighbourhood [20].	11
1.7	Overview of the research methodology.	14
2.1	Kinematic setup of the TDC cable model. The one-dimensional reference domain Ω_r is mapped to the undeformed configuration Γ_X via $X(r)$ and to the deformed configuration Γ_x via $x(r, t)$, with $u = x - X$ denoting the displacement field [29].	16
2.2	Schematic of the catenary mooring-line configuration modelled by GNL-FEM, illustrating the four distributed external loads, self-weight, buoyancy, hydrodynamic drag, and the spring-dashpot seabed contact force, applied along the line between the fairlead and the anchor [29].	18
3.1	End-to-end architecture of the GAT+LSTM surrogate f_θ of Equation (3.3). The feature-matrix cell values shown are illustrative.	27
3.2	The GATv2 attention mechanism of Equations (3.12)–(3.14), shown for a single attention head. Illustrative weights shown, not measured from the trained model.	34
3.3	The LSTM temporal encoder of Equations (3.18)–(3.24), shown for a single node i . The input sequence shown is illustrative, not extracted from the trained model.	36
3.4	The prediction head of Equation (3.25), shown for a single node i	37
4.1	Stage-C1 training history over 42 epochs. (a) Loss curves: training total loss, data loss, physical loss, and validation loss, with the learning-rate reduction events marked on the secondary axis. (b) Validation global R^2 for all three targets. (c) Validation peak-tension underprediction rate and bias. (d) Validation MAE for position and tension targets. (e) Gradient norm. The vertical dashed line marks the defended checkpoint at epoch 12 (minimum validation loss).	48
4.2	Best per-epoch global R^2_{tension} on the validation set as a function of node count N . Performance decreases from 0.985 at $N = 4$ to 0.977 at $N = 21$	50
4.3	Predicted (dashed) and true (solid) per-node peak tension, the maximum over the 5 s forecast window, along the mooring line for a representative out-of-distribution test window (location 11, $N = 21$). Node 0 is the anchor and node 20 is the fairlead.	51
4.4	Predicted (dashed) and true (solid) mooring line geometry at the $t = +5.0$ s forecast horizon, for a representative out-of-distribution test window (location 10). Node 0 is the anchor (left) and node 20 is the fairlead (right).	52
4.5	Predicted (dashed) and true (solid) tension profiles for the same out-of-distribution sea-state window (location 11), shown as four panels at node counts $N \in \{4, 8, 12, 21\}$. Each panel plots tension against node index from anchor (0) to fairlead ($N - 1$) at the $t = +5.0$ s forecast horizon.	53
4.6	Spatial tension load field reconstructed by the C1 surrogate at three discretisations ($N = 4, 8$ and 21) from a single set of weights, plotted against normalised arc length s/L from anchor ($s/L = 0$) to fairlead ($s/L = 1$) for an out-of-distribution sea-state window (location 11).	53
4.7	Variance decomposition of the held-out test set for each model target. Each bar represents 100 % of the total variance for that target, divided into level (between windows), spatial (between nodes), and temporal (within-node oscillation) components.	56

4.8	Predicted (dashed) versus true (solid) fairlead tension over the 50-step forecast horizon (5 s at $\Delta t = 0.1$ s) for two out-of-distribution test windows.	57
4.9	Temporal R^2 of the model (vertical axis) versus the best baseline R^2 (horizontal axis) for 400 held-out OOD fairlead windows. Each point is one 5-second future.	58
4.10	Predicted versus true per-node peak tension T^{peak} for a random subsample of test data points, coloured by node count N . The diagonal is the parity line, the shaded region marks the underprediction half-plane. Full held-out test metrics (30 000 windows): underprediction rate 17.6 %, bias +197.5 N (Table 4.1).	59
4.11	Comparison of the C1 surrogate against the persistence baseline (last observed frame repeated across the $F = 50$ horizon). (a) Global R^2 per target pooled over 20 003 test windows. (b) Per-window temporal R^2 at the fairlead versus fairlead oscillation amplitude, for 400 out-of-distribution windows. Rolling medians (solid lines) show that the surrogate's advantage over persistence grows monotonically with amplitude.	61
4.12	Evolution of the learned uncertainty σ_k for the peak-underprediction term. Left: C2a (one-sided hinge) showing σ -runaway. Right: C2c ($\tau = 0.9$ pinball) showing stable convergence to a plateau at ≈ 0.18 . Vertical markers indicate the R^2 -selected checkpoint (ep 5) and, for C2c, the safety-selected checkpoint (ep 20).	63
4.13	Per-node peak tension (maximum over the 5 s forecast window) for the defended C1 model (penalty on) and the ablation (penalty off), against the finite-element truth, for a representative out-of-distribution window (location 11, $N = 21$).	66

List of Tables

1.1	Comparison of the Lumped-Mass Method (LMM) and the Finite-Element Method (FEM) for offshore mooring-line analysis.	8
1.2	Comparison of model fidelity, normalised computational time, and accuracy for coupled floating offshore wind turbine simulations, adapted from Glasspool et al. [19].	10
2.1	Water depths and current-sampling depths for the ten thesis locations.	20
2.2	Distribution of per-case peak tensions across the 2 953 finite simulation cases below the original 5 MN screening level.	23
2.3	Corrupt simulation cases excluded from all dataset splits. A case is flagged if it contains non-finite tension values or if its peak tension exceeds 1.5 MN.	24
2.4	Window budgets for Stage B (50 cases per location) and Stage C (300 cases per location). The per-epoch sample is redrawn randomly each epoch from the training pool.	25
3.1	The fourteen static node features, assembled from the initial static-equilibrium configuration of each case.	29
3.2	The nineteen dynamic node features evaluated at every history step τ . Columns 0–5 vary per time step; columns 6–18 are the environmental descriptors of the case, constant over time and identical for every node.	30
3.3	The ten static edge features. Both directed edges of a physical segment carry the same feature vector.	31
3.4	The four dynamic edge features, derived from the endpoint seabed features at every history step.	32
3.5	Peak-tension safety metrics, evaluated on the per-node peak tension over the $F = 50$ step horizon and aggregated across all validation windows.	39
3.6	Stage B ablation summary at the best validation checkpoint per run, all with seed 42. Underprediction rates are averaged across all ten node counts. Bold denotes the selected configuration.	43
3.7	S-series seed robustness results at the best validation checkpoint per seed. Three-seed aggregates are given as mean $\pm$ sample standard deviation. Bold denotes the selected configuration.	43
3.8	Final hyperparameter configuration for Stage C. Bold entries differ from the B0 baseline.	44
4.1	Held-out test performance of the Stage C primary model (C1, defended checkpoint epoch 12) compared with the Stage-B BE1 three-seed mean. All metrics are computed in physical units.	46
4.2	Stage-C1 per-topology test metrics (a): position and tension accuracy. Absolute positions x_{abs} and z_{abs} in metres, tension in newtons.	49
4.3	Stage-C1 per-topology test metrics (b): peak-tension safety. A positive bias denotes safe overprediction, and the underprediction rate is the fraction of nodes whose predicted peak falls below the true peak.	49
4.4	Stage-C1 per-topology test metrics (c): temporal-difference MAE, the consecutive-step rate-of-change error with one step equal to 0.1 s.	49
4.5	Evidence for an architectural accuracy ceiling. Increasing the training data by $6\times$ (Stage B to Stage C) and then adding mild dropout regularisation both fail to lift the tension R^2 beyond the observed plateau.	54

4.6	Decomposition of total variance in the held-out test set into level (between sea-state windows), spatial (between nodes within a snapshot), and temporal (within-node over the forecast horizon) components. Values are percentages of total variance per target, computed over 2 000 windows drawn from the full held-out test set (development-location temporal test windows and OOD locations 10–11) with the 53-case corrupt blacklist applied. Entries are rounded independently, so a row need not sum to exactly 100 %. . . .	55
4.7	Held-out global R^2 of the C1 surrogate versus a persistence baseline (last observed frame repeated across the $F = 50$ horizon), pooled over 20 003 test windows.	61
4.8	Learned uncertainties σ_k and effective weights $1/(2\sigma_k^2)$ for the four physical-loss terms across C2 variants, all read from the converged training-history tail of each run (ep 51 for C2a, ep 41 for C2b, ep 55 for C2c). Small σ implies a precisely fitted term and therefore a large effective weight. The three two-sided terms rank consistently as temporal $\succ$ physical MAE $\succ$ peak tension in all runs.	62
4.9	Held-out test performance of all Stage-C runs, each at its own R^2 -selected checkpoint. Bold denotes the best value in each column, where best means the largest R^2 , the smallest underprediction rate, and the smallest MAPE and temporal-difference MAE. The bias column is not bolded, since the preferable value there is a moderate positive margin rather than an extremum.	64
4.10	Effect of removing the peak-underprediction penalty from C1. The two runs are identical except for the penalty weight ($w_{\text{underpred}} = 0.02$ versus 0). Held-out test set, 30 000 windows. Bold denotes the preferable value in each row, so the accuracy rows favour the ablation and the safety rows favour C1.	65
4.11	C2c safety-selected checkpoint (epoch 20) versus C1. Safety selection identifies the epoch at which the pinball weight has matured sufficiently to constrain underprediction, recovering a positive peak bias comparable to C1's at a modest accuracy cost. Bold denotes the better value in each column, with the bias column left unbolded because both entries are safely positive.	66
4.12	Training cost of the two Stage-C surrogate models on a single NVIDIA A100 GPU. . . .	67

Introduction

1.1. Background

The ocean has historically provided humanity with vast resources such as crude oil and natural gas beneath the seabed, and the discovery of large offshore hydrocarbon fields necessitated the construction of floating platforms to exploit them [1]. In the last decade, however, international mandates established by the Paris Agreement to restrict global warming to well below 2° C have created an urgent need to harness offshore wind energy [2].

According to the Global Offshore Wind Report of 2025 [3], approximately 83 GW of offshore wind capacity is currently operational worldwide, with a further 48 GW under construction. The European Commission's ambitious targets aim to deploy 60 GW of offshore wind by 2030 and to expand this capacity to 300 GW by 2050 [4].

Fixed-bottom platforms have historically dominated the offshore wind market, but their deployment is geographically constrained to shallow waters. To overcome this limitation, the industry is increasingly turning toward Floating Offshore Wind Turbines (FOWTs), a technological evolution that unlocks deep-water sites characterised by stronger and more consistent wind resources. This shift significantly broadens the potential development area, estimated to be four times larger than that available to fixed-bottom foundations [5]. Current projections suggest that 2.6 GW of floating capacity will be installed by 2030, paving the way for full commercialisation and multi-gigawatt annual installations in the following decade [3].

The floating offshore renewable landscape is not limited to wind energy alone. Among the developing renewable technologies, ocean energy is gaining increasing attention, albeit at a slower pace of development than wind [6].

The feasibility of every one of these floating platforms, whether wind, wave, or oil and gas, rests fundamentally on the integrity of its mooring system. The primary purpose of a mooring system is station-keeping, providing sufficient restoring forces to restrict the platform's horizontal offsets in surge, sway, and yaw against the combined environmental loads of wind, waves, and currents. Limiting these motions is essential for the safety of operations ranging from drilling and production to wind power generation, not merely for positioning but also for preventing the rupture of electrical power cables or risers, which are highly sensitive to excessive excursions [1]. The mooring system further contributes to the global stiffness of the platform, thereby shaping its dynamic response, ensuring efficient operation, and mitigating fatigue-induced structural damage [7].

Accurate prediction of mooring-line dynamics is therefore paramount for both structural safety and continuous energy production. The remainder of this chapter first establishes the physical fundamentals of mooring-line behaviour and the numerical simulation techniques traditionally used to model it, before turning to the computational limitations of these high-fidelity solvers and the resulting case for deep-learning-based surrogate models.

1.2. Mooring Lines and Underlying Physics

Meeting these station-keeping requirements under combined wind, wave, and current loading, together with additional dynamic loads such as the asymmetric tension induced during pipe-laying, calls for different physical strategies for generating restoring force. The offshore industry has consequently converged on a small number of distinct mooring-line configurations, each suited to a different water depth and loading regime [1].

1.2.1. Mooring Configurations

The geometric profile of a mooring line determines how it generates restoring force and compliance, and the three principal configurations are compared in Figure 1.1.

- **Catenary:** the most widely used configuration in shallow to intermediate water depths. A significant portion of the line rests on the seabed at static equilibrium, and the restoring force is generated primarily by the self-weight of the suspended catenary shape (Figure 1.1a) [1].
- **Taut-leg:** lines run directly under tension from the seabed anchor to the fairlead, with no segment resting on the seafloor, reducing the anchor footprint and material requirements. Because compliance is derived from the axial elasticity of the line rather than its weight, taut-leg systems can become excessively stiff in shallow water, and are therefore deployed predominantly in deep and ultra-deep water, typically using synthetic-fibre lines (Figure 1.1b) [1, 8].
- **Tension-leg:** used exclusively on Tension Leg Platforms (TLPs), whose buoyancy substantially exceeds their structural weight. The resulting net upward force is counteracted by vertical, highly pre-tensioned tendons that remain taut at all times, virtually eliminating heave, roll, and pitch. When environmental loads displace the platform, the tendons tilt and the horizontal component of their tension restores the platform to station (Figure 1.1c) [8, 9].

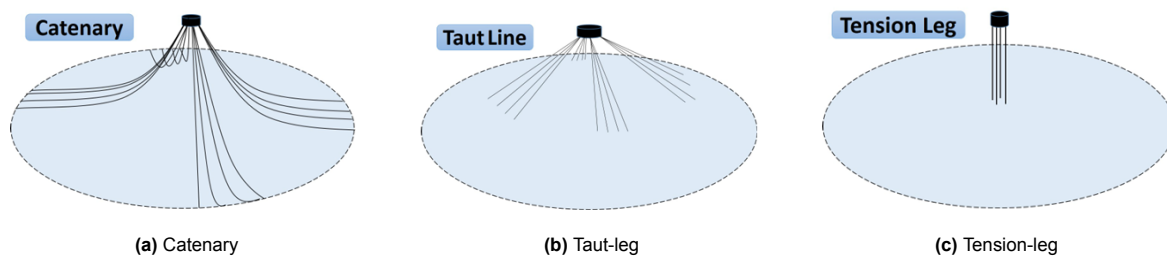


Figure 1.1: Principal offshore mooring-line configurations [9].

1.2.2. Environmental Loads

A floating structure in open water is subjected to combined loading from wind, waves, currents, and, in specific regions, ice (Figure 1.2). The mooring system must be evaluated across the most unfavourable combinations of wind, wave, and current directions consistent with the site-specific metocean conditions, without exceeding structural or fatigue limits [1].

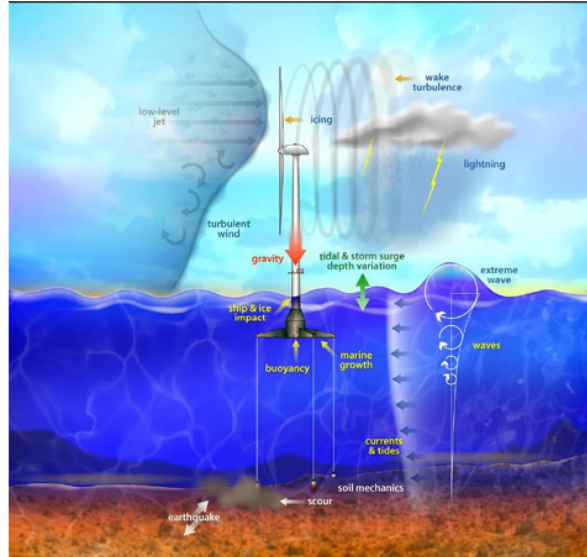


Figure 1.2: Loads acting on a floating structure [9].

To predict mooring dynamics accurately, these loads are typically grouped by frequency band [1]:

- **Steady loads:** mean wind, current, and mean wave-drift forces exert a constant pressure on the structure, pushing the floater to a static offset that is directly counteracted by the restoring force of the mooring system.
- **Wave-frequency loads:** cyclic loads with periods of roughly 5–30 s induce wave-frequency vessel motions and cyclic line tensions, governing both peak mooring strength and long-term fatigue damage.
- **Low-frequency loads:** slowly varying (slow-drift) loads excite the floating system at its natural periods in surge, sway, and yaw, typically 3–10 minutes (180–600 s), and can induce significant dynamic tension in the mooring lines.

Wind loads: Wind acts on the floating structure both directly, through steady and low-frequency dynamic pressure on the exposed hull and superstructure, and indirectly, by generating the waves and currents that load it in turn. Wind speed varies with both time and elevation above the sea surface, and the sustained wind velocity profile is commonly expressed as a power-law function of a time-averaged reference speed at a 10 m elevation [1]:

$$U(z) = U_{10} \left(\frac{z}{10} \right)^b \quad (1.1)$$

where $U(z)$ is the wind speed at height z above the sea surface, U_{10} is the time-averaged reference wind speed at 10 m, and b is the power-law exponent, typically taken as 0.125.

The instantaneous wind force on the structure is obtained by summing the contribution of every exposed member above the waterline [1]:

$$F_w = \frac{1}{2} \rho_a C_s A (V_Z + u' - \dot{x}) |V_Z + u' - \dot{x}| \quad (1.2)$$

where F_w is the instantaneous wind force, ρ_a is the mass density of air, C_s is a shape coefficient that may also account for shielding effects, A is the projected area of the member, V_Z is the mean wind speed at the elevation of that member, obtained from the profile of Equation (1.1) as $V_Z = U(z)$, u' is the instantaneous speed variation from the sustained wind, and $\dot{x}$ is the instantaneous velocity of the structural member itself.

Wave loads: Of the three load bands, wave loads are the dominant source of the wave-frequency tensions defined above, and are the most directly relevant to the dynamic mooring response considered in this thesis. Ocean waves are broadly classified into wind waves, generated locally by wind blowing over the sea surface, with heights ranging from minor ripples to extremes exceeding 30 m, and swells, generated by distant weather systems and propagating with longer periods, often 20 s or more, and a narrow-band spectrum largely decoupled from the local wind [1]. Because a real sea state comprises a broad range of individual wave heights, its characteristic scale is conventionally expressed through the significant wave height H_s , the average height of the highest one-third of the waves recorded over a given duration [1].

The simplest and most widely used description of ocean waves in offshore engineering is linear, or Airy, wave theory, valid when the wave height is small relative to both the wavelength and the water depth. Under this theory, an irregular sea state is represented as the superposition of many regular wave components of different amplitude, frequency, and phase (Figure 1.3) [8].

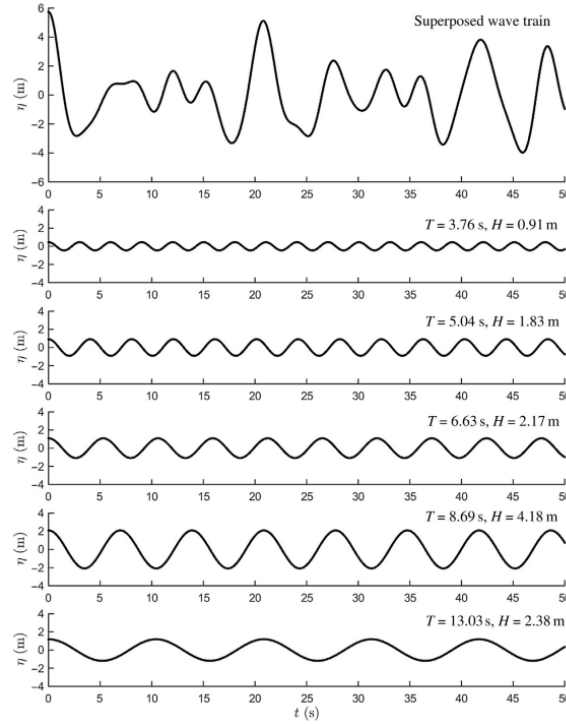


Figure 1.3: An irregular sea state represented as the sum of many regular wave components [9].

The random, irregular nature of a real sea state is captured statistically through an energy spectrum that distributes wave energy across frequency. Among the spectral models in common use, such as the Bretschneider spectrum, the Joint North Sea Wave Project (JONSWAP) spectrum is particularly widely adopted for wind-generated seas in fetch-limited waters, and is the spectrum adopted for the environmental forcing used throughout this thesis (Section 2.2). Its parametric engineering form is [8]:

$$S_{\zeta}(\omega) = \frac{320 H_s^2}{T_p^4} \omega^{-5} \exp \left\{ \frac{-1950}{T_p^4} \omega^{-4} \right\} \gamma^A \quad (1.3)$$

where the peak-shape exponent A is given by

$$A = \exp \left\{ - \left(\frac{\omega/\omega_p - 1}{\sigma\sqrt{2}} \right)^2 \right\} \quad (1.4)$$

Here $S_{\zeta}(\omega)$ is the wave spectral density as a function of the angular frequency ω , H_s is the significant wave height, T_p is the peak wave period, γ is a peakedness factor (typically taken as 3.3), $\omega_p = 2\pi/T_p$

is the circular frequency at the spectral peak, and σ is a spectral width parameter equal to 0.07 for $\omega < \omega_p$ and 0.09 for $\omega > \omega_p$ [8]. Parameterising the spectrum directly by H_s and T_p makes it possible to generate a physically consistent, non-repeating wave time series for any sea state of interest from only these two summary statistics.

Current loads: Currents are driven by ocean circulation, tides, and storm surges, and are generally treated as a steady-state phenomenon whose velocity varies with depth through the water column (Section 2.2). Beyond the steady drag and lift forces they exert on the hull, risers, and mooring lines, currents flowing past bluff bodies can also trigger vortex-induced motion (VIM), a transverse oscillation that increases the effective in-line drag coefficient, amplifies low-frequency line tensions, and accelerates fatigue damage [1].

The high-fidelity numerical solvers introduced in the next section are built precisely to resolve this dynamic, wave-frequency response, together with the mean and low-frequency components, at the temporal resolution required to capture peak tensions and fatigue-relevant cycles.

1.2.3. Governing Equations of Motion

The fundamental behaviour of a mooring line is captured by analysing the forces acting on a differential element. Consider a small segment of length dl in the two-dimensional plane (x, z) , subject to hydrodynamic forces, self-weight, and tension (Figure 1.4) [1].

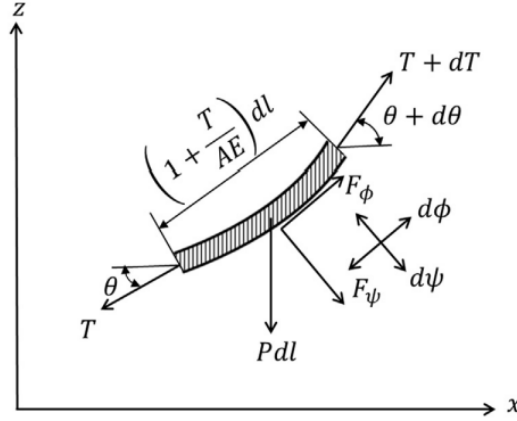


Figure 1.4: Forces and displacement acting on an element of a mooring line [1].

Applying Newton's second law to this element and decomposing the forces into tangential (ϕ) and normal (ψ) directions yields the governing differential equations of motion [1]:

$$dT - P \sin \theta \, dl + F_\phi \left(1 + \frac{T}{AE}\right) dl = m \frac{d^2 \phi(l)}{dt^2} \quad (1.5)$$

$$T \, d\theta - P \cos \theta \, dl - F_\psi \left(1 + \frac{T}{AE}\right) dl = m \frac{d^2 \psi(l)}{dt^2} \quad (1.6)$$

where T is the effective tension in the line, P is the wet weight per unit length, θ is the angle of the line tangent relative to the horizontal, AE is the axial stiffness, F_ϕ and F_ψ are the hydrodynamic (drag and inertia) forces in the tangential and normal directions, respectively, and m is the mass of the element.

Crucially, the geometry of the line is coupled to the tension through the material's elasticity, since the stretched length of the element relates to the global coordinate system through

$$dx = \left(1 + \frac{T}{AE}\right) \cos \theta \, dl, \quad dz = \left(1 + \frac{T}{AE}\right) \sin \theta \, dl \quad (1.7)$$

This coupling between geometry and tension, mediated by the line's axial elasticity, is exactly what the numerical solvers discussed next must resolve, whether by discretising the line directly or by iterating between geometry and tension until convergence.

1.3. Finite-Element Solvers and Lumped-Mass Approaches

A quasi-static analysis neglects the mooring line's own inertia and hydrodynamic damping, treating the line at every instant as a static catenary configuration determined solely by the fairlead position. This assumption is computationally efficient and adequate for preliminary sizing or for systems in benign environments, but it cannot reproduce the dynamic amplification caused by wave-frequency fairlead motion, and it has been shown to systematically underpredict the peak and fatigue loads that govern detailed design [1, 10]. A dynamic analysis instead solves the line's equation of motion directly in the time domain, and two numerical strategies dominate offshore engineering practice for doing so, the Lumped-Mass Method and the Finite-Element Method (FEM).

1.3.1. Lumped-Mass Method

The Lumped-Mass Method (LMM) discretises a continuous mooring line into a finite number of straight, massless segments connecting a set of node points, at which the line's mass, weight, buoyancy, and drag properties are concentrated. The formulation has evolved from early two-dimensional finite-difference models [11] to three-dimensional, multi-component implementations [12], with axial elasticity later incorporated to capture the stretch of the line under tension [13]. In the widely used formulation of Hall and Goupee [14], which also underlies the open-source MoorDyn program discussed in Section 1.3.3, a line of unstretched length L is divided into N segments connecting $N+1$ nodes (Figure 1.5), with the elastic and damping properties assigned to the massless segments and the inertial and hydrodynamic properties lumped at the nodes.

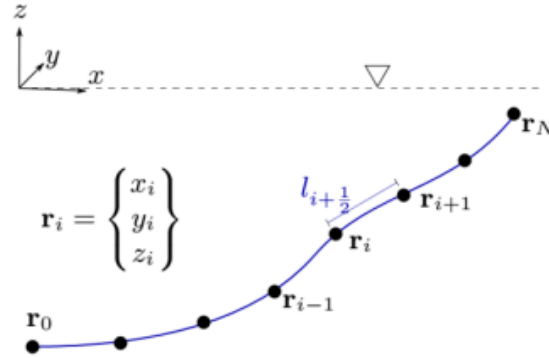


Figure 1.5: Mooring-line discretisation and node indexing under the Lumped-Mass Method [14].

The dynamic behaviour of each node i follows from Newton's second law, expressed as a system of second-order ordinary differential equations that balance the node's inertial, internal, and external forces [14]:

$$(\mathbf{m}_i + \mathbf{a}_i) \ddot{\mathbf{r}}_i = \mathbf{F}_{\text{int},i} + \mathbf{F}_{\text{ext},i} \quad (1.8)$$

where $\mathbf{r}_i$ is the position vector of node i , $\mathbf{m}_i$ is its mass matrix, and $\mathbf{a}_i$ is the hydrodynamic added-mass matrix. Expanding the internal and external force terms gives the complete equation solved at every time step [14]:

$$(\mathbf{m}_i + \mathbf{a}_i) \ddot{\mathbf{r}}_i = \underbrace{(\mathbf{T}_{i+1/2} + \mathbf{C}_{i+1/2}) - (\mathbf{T}_{i-1/2} + \mathbf{C}_{i-1/2})}_{\text{internal tension and damping}} + \underbrace{\mathbf{W}_i + \mathbf{B}_i}_{\text{weight and seabed reaction}} + \underbrace{\mathbf{D}_{p,i} + \mathbf{D}_{q,i}}_{\text{hydrodynamic drag}} \quad (1.9)$$

where the subscripts $i \pm 1/2$ denote the two segments adjacent to node i , $\mathbf{T}$ and $\mathbf{C}$ are the segment tension and internal damping forces, $\mathbf{W}_i$ and $\mathbf{B}_i$ are the node's weight and seabed reaction, and $\mathbf{D}_{p,i}$ and $\mathbf{D}_{q,i}$ are the normal and tangential hydrodynamic drag forces obtained from Morison's equation.

Accuracy and Computational Cost: Hall and Goupee [14] validated this formulation against 1:50-scale model test data for the DeepCwind semisubmersible. With fairlead motions prescribed directly from the experiment, the LMM predicted fairlead tension amplitudes to within 8 % and phases to within 7° of the measured values for regular waves, and reproduced snap loads, that is, slack-line events, which a quasi-static model could not. Equivalent fatigue and extreme loads matched the experimental data to within 10 %, compared with underpredictions of 65–75 % and 10–25 %, respectively, from a quasi-static model. Accuracy degraded once the mooring model was fully coupled to a floating-platform simulator, an effect attributed to the platform's own hydrodynamic model rather than to the mooring formulation itself. The LMM remains considerably cheaper than a full finite-element solution, although the short time steps required for numerical stability in its explicit time integration still make it markedly more expensive than a quasi-static model.

1.3.2. Finite-Element Method: Slender-Rod Theory

A higher-fidelity alternative treats the mooring line not as a chain of lumped points but as a spatially continuous, extensible elastic rod capable of resisting bending and torsional loads in addition to axial tension. This nonlinear slender-rod theory was originally derived by Garrett [15] and underlies most modern finite-element mooring and riser solvers.

Kinematics and Strain: Let $\mathbf{r}(s, t)$ denote the position vector of the rod centreline at arc length s and time t , and let $\mathbf{t}$ be the unit tangent vector along the rod axis. The local axial strain ϵ relates the two through

$$\mathbf{r}' = \frac{\partial \mathbf{r}}{\partial s} = (1 + \epsilon) \mathbf{t}, \quad \epsilon = |\mathbf{r}'| - 1 \quad (1.10)$$

Constitutive Relations: The internal effective tension T and bending moment $\mathbf{M}$ follow from the material's constitutive laws,

$$T = EA\epsilon, \quad \mathbf{M} = EI(\mathbf{r}' \times \mathbf{r}'') \quad (1.11)$$

where EA is the axial stiffness, EI is the bending stiffness, and $\mathbf{r}''$ is the second derivative of the centreline with respect to arc length, so that the product $\mathbf{r}' \times \mathbf{r}''$ is the curvature vector of the rod.

Equation of Motion: For a rod element of distributed mass m subject to distributed external hydrodynamic loads $\mathbf{f}$, comprising gravity, buoyancy, and Morison drag, the balance of linear momentum gives [15]

$$-(EI\mathbf{r}'')' + (T\mathbf{r}')' + \mathbf{f} = m\ddot{\mathbf{r}} \quad (1.12)$$

where the first term is the shear-force gradient due to bending stiffness and the second is the restoring force due to axial tension.

Finite-Element Discretisation: Equation (1.12) is solved numerically by applying the principle of virtual work to derive its weak form, then discretising the position field within each element using cubic Hermite shape functions, which enforce C^1 continuity of the interpolated centreline, that is, continuity of slope, as required to support the second-derivative bending terms. The position within an element is approximated as

$$\mathbf{r}(s, t) = \sum_{i=1}^4 H_i(\xi) \mathbf{Q}_i(t) \quad (1.13)$$

where $\xi = s/L_e \in [0, 1]$ is the normalised element coordinate, $H_i(\xi)$ are the Hermite shape functions, and the nodal degree-of-freedom vector $\mathbf{Q} = \{\mathbf{r}_j, \mathbf{r}'_j, \mathbf{r}_{j+1}, \mathbf{r}'_{j+1}\}^T$ collects the positions and tangents at the two element end-nodes j and $j + 1$. Substituting this discretisation into the weak form yields the assembled system of nonlinear matrix equations solved at every time step,

$$\mathbf{M}\ddot{\mathbf{Q}} + \mathbf{K}(\mathbf{Q})\mathbf{Q} = \mathbf{F}_{\text{ext}} \quad (1.14)$$

where $\mathbf{M}$ is the consistent mass matrix and $\mathbf{K}$ is the stiffness matrix, which includes both material and geometric contributions and depends on the current configuration $\mathbf{Q}$ because of the theory's inherent geometric nonlinearity.

Table 1.1 summarises the principal differences between the two formulations.

Table 1.1: Comparison of the Lumped-Mass Method (LMM) and the Finite-Element Method (FEM) for offshore mooring-line analysis.

Feature	Lumped-Mass Method	Finite-Element Method
Theoretical basis	Discrete mechanics, Newton's second law applied to point masses	Continuum mechanics, principle of virtual work applied to the weak form of rod theory
Discretisation	Point masses connected by massless spring-damper segments	Spatially continuous elements approximated by shape functions
Interpolation	Linear	Cubic Hermite polynomials
Degrees of freedom	3 per node, translations x, y, z	6 per node, positions and tangents
Stiffness physics	Axial stiffness EA only, bending and torsion neglected	Axial (EA), bending (EI), and torsional (GJ) stiffness
Mass representation	Lumped, diagonal mass matrix concentrated at the nodes	Consistent mass matrix capturing inertial coupling between nodes
Governing equation	Second-order system of ordinary differential equations, Equation (1.8)	Nonlinear matrix equation, Equation (1.14)

1.3.3. Industry Simulation Software

Both formulations underlie widely used commercial and open-source mooring simulators. Three representative tools are introduced briefly below.

OrcaFlex: OrcaFlex, developed by Orcina, models a line using a hybrid finite-element and lumped-mass structure [16]. The line is divided into straight, massless model segments connecting a node at each end, and each segment represents only the axial and torsional properties of the line, while all other properties, namely mass, weight, buoyancy, and drag, are lumped to the nodes by assigning each node the properties of its two adjacent half-segments. Axial and torsional stiffness are represented by spring-dampers acting along the segment, and bending stiffness is represented by rotational spring-dampers connecting each segment end to its node, allowing independent stiffness values in orthogonal bending planes for non-axisymmetric lines. Forces and moments computed at the nodes are then integrated forward in the time domain to obtain the line's dynamic response.

MoorDyn: MoorDyn is an open-source lumped-mass mooring simulator developed by Hall [17], implementing the LMM formulation of Section 1.3.1 directly. A line is discretised into N segments connecting $N + 1$ nodes, with the anchor as node zero, and its ends attach to Fixed, Vessel, or Connect points, depending on whether they are stationary, driven by an external platform model, or free to move under the local force balance. Hydrodynamic drag is computed at the nodes from Morison's equation, and contact with the seabed is represented by a vertical spring-damper reaction proportional to the local penetration depth. At initialisation, MoorDyn first estimates each line's configuration from a quasi-static catenary solution, then runs a dynamic relaxation stage to settle the whole system to its true equilibrium before the coupled simulation begins. During time-stepping, MoorDyn integrates the nodal accelerations forward using an explicit second-order Runge–Kutta scheme, with its own internal time step set as a fraction of the coupling time step imposed by the calling program, which may be a standalone driver or a floating-platform simulator such as FAST. This formulation was validated against the same 1:50-scale DeepCwind semisubmersible test data discussed in Section 1.3.1 [14].

RIFLEX: RIFLEX, maintained by SINTEF Ocean as part of the SIMA workbench, is a finite-element solver for slender marine structures such as risers, pipelines, and mooring lines [18]. It combines two element formulations, a total-Lagrangian bar element with three translational degrees of freedom per node for purely axial members, and a co-rotational beam element with six degrees of freedom per node, three translational and three rotational, for members that also resist bending and torsion. Both element types accommodate the large displacements and rotations typical of slender offshore

structures while remaining computationally efficient, and RIFLEX solves the same class of nonlinear finite-element equations introduced in Section 1.3.2.

Despite their different levels of fidelity, the lumped-mass and finite-element solvers described above share the same computational bottleneck, and it is this trade-off between physical accuracy and simulation speed that motivates the data-driven alternative explored next.

1.4. Computational Limitations and the Case for Deep Learning

1.4.1. The Fidelity-Efficiency Trade-off

To quantify this trade-off directly, Glasspool et al. [19] define a hierarchy of five model fidelities for the coupled simulation of floating offshore wind turbines, ranging from a fully coupled high-fidelity time-domain solution to a simplified frequency-domain solver. Four of the five are described below. Fidelity 2 is omitted, because it removes only the turbine controller and leaves the hydrodynamic and mooring modelling untouched, so it carries no consequence for the mooring response of interest here. Fidelity 1 is the high-fidelity baseline, a fully coupled time-domain formulation that resolves the turbine's aerodynamics through blade-element-momentum theory alongside a nonlinear finite-element mooring model. Fidelity 3 retains this fully coupled time-domain formulation, including the turbine controller, but neglects the second-order wave-drift loads and wave-drift damping, reducing the hydrodynamic forcing to its first-order components. This simplification bears directly on the mooring system, since the slow-drift loads introduced in Section 1.2.2 are precisely a second-order wave effect. Fidelity 4 introduces a structural simplification in which the rotor-nacelle assembly is replaced by a representative point mass driven by an aerodynamic thrust curve, while the mooring system itself remains dynamic. Fidelity 5 is the lowest fidelity level, a linearised frequency-domain solver that abandons time-stepping entirely in favour of spectral solutions built from linearised stiffness and damping matrices.

To compare these fidelity levels on a common basis, Glasspool et al. [19] define a normalised computational time coefficient as the ratio of a given model's execution time to that of the high-fidelity baseline,

$$T_{\text{norm}} = \frac{T_{\text{model}}}{T_{\text{Fidelity 1}}} \quad (1.15)$$

where a value of 1.0 indicates parity with the baseline and lower values indicate a proportional speed-up.

The resulting comparison, summarised in Table 1.2, reveals a strongly nonlinear relationship between cost and accuracy. The Fidelity 3 model achieves no meaningful cost reduction, $T_{\text{norm}} = 1.00$ at matched environmental fidelity, yet neglecting the second-order hydrodynamics produces a notable overestimation of mooring tension in the most severe wind state considered, evidence that slow-drift forcing is important for accurately modelling mooring loads while contributing almost nothing to the computational burden. The Fidelity 4 model reaches $T_{\text{norm}} = 0.06$, a 94 % reduction in computational cost relative to the baseline, but this simplification introduces an epistemic error of approximately 10 % in key load predictions and fails to capture controller-induced dynamics. The Fidelity 5 model offers a far more extreme reduction, $T_{\text{norm}} = 4.05 \times 10^{-3}$, around 0.4 % of the baseline cost, but excludes all transient nonlinearities and is consequently unable to predict snap loads, shock loads, or extreme peak tensions, limiting its use to preliminary station-keeping analysis rather than detailed load validation.

Table 1.2: Comparison of model fidelity, normalised computational time, and accuracy for coupled floating offshore wind turbine simulations, adapted from Glasspool et al. [19].

Fidelity		Physical Definition	T_{norm}	Accuracy and Limitations
Fidelity (baseline)	1	Fully coupled time-domain solution, finite-element mooring and blade-element-momentum aerodynamics	1.00	Ground truth
Fidelity 3		Time-domain solution with second-order wave-drift loads and damping neglected, first-order hydrodynamics only	1.00	Overestimates mooring tension in severe sea states, no meaningful cost saving
Fidelity 4		Time-domain solution with a lumped-mass turbine, point mass and thrust curve, mooring system remains dynamic	0.06	$\approx 10\%$ error, neglects controller dynamics
Fidelity 5		Linearised frequency-domain solver	4.05×10^{-3}	Misses nonlinear transients, snap loads, and extreme peaks

1.4.2. Deep Learning as a Surrogate Modelling Solution

This fidelity hierarchy exposes a fundamental trade-off between physical accuracy and computational efficiency that constrains the numerical modelling of floating structures. The high-fidelity Fidelity 1 solution, and the equally detailed lumped-mass and finite-element solvers of Sections 1.3.1 and 1.3.2, capture the nonlinear phenomena, such as snap loads and transient peak tensions, that govern safe mooring design, but their computational cost renders repeated evaluation impractical. Conversely, the reduced-order Fidelity 4 and Fidelity 5 approaches are cheap enough for repeated use but fail to resolve the nonlinear dynamics that make the high-fidelity result trustworthy in the first place.

A deep-learning surrogate model offers a way out of this trade-off. Rather than integrating the equations of motion at every time step, it learns the nonlinear input-output mapping realised by a high-fidelity solver directly from its simulation output, reproducing mooring-line motions and tensions with comparable accuracy at a fraction of the computational cost. Training such a model is itself computationally expensive, but this cost is paid once, offline. Once trained, the surrogate requires no further time integration or iterative solution. A single forward pass through the network returns a full prediction in milliseconds rather than the hours typically required by a coupled time-domain solver, making repeated evaluation across many environmental conditions, successive design iterations, or real-time monitoring practical for the first time.

A further advantage, central to the architecture developed in this thesis, is that a single trained model and a single set of weights can serve mooring-line configurations of different node counts and node positions along the line, without retraining for each new topology, a capability derived and explained in Section 3.7.

This flexibility across line configurations is not incidental, but reflects a deeper structural property of the problem. A mooring line is most naturally described not as a grid or a single time series but as a graph, a chain of discrete nodes joined by segments, whose response is distributed spatially along the line and evolves in time. The deep-learning families most commonly applied to mooring analysis do not capture this structure directly. A recurrent sequence model applied to the tension at the fairlead collapses the entire line to one point and discards the coupling between neighbouring nodes, while a convolutional network presupposes the regular Euclidean lattice that a variable-length, irregularly spaced mooring chain does not possess. A graph neural network (GNN) instead passes information along the physical connectivity of the line itself, preserving exactly the topological dependencies that these architectures omit [20]. Combining such a spatial graph encoder with a recurrent temporal encoder yields a spatio-temporal graph network that captures node-to-node coupling and temporal evolution together, and it is this pairing, refined with an attention mechanism that lets the more influential neighbours contribute

more strongly to each node, that has made the graph paradigm a natural and increasingly prominent choice for the spatially distributed structural problems of offshore engineering. The theoretical foundations of graph neural networks, on which the model developed in this thesis is built, are set out next.

1.4.3. Theoretical Foundations of Graph Neural Networks

Many relationships in physical and engineering systems are most naturally expressed as graphs, and graph neural networks are designed to process this structured information directly, without discarding the topological dependencies between components. The foundational model was introduced by Scarselli et al. [21]. Whereas those early architectures relied on recursive functions iterated to a fixed-point equilibrium, modern graph networks are formalised through a framework known as neural message passing [20], which is the formulation adopted here.

Graph representation and notation: A graph is defined as a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. The local topology of a node $u \in \mathcal{V}$ is described by its neighbourhood $\mathcal{N}(u)$, the set of all nodes connected to u by an edge. To encode the physical properties of the system rather than a discrete label, each node u carries a real-valued feature vector $\mathbf{x}_u \in \mathbb{R}^d$, where d is the dimension of the feature space.

Neural message passing: The core operation of a modern graph network is iterative message passing, in which vector messages are exchanged between neighbouring nodes and combined through learnable functions. At each iteration, a hidden embedding $\mathbf{h}_u^{(k)}$ for every node u is updated from the information aggregated over its neighbourhood $\mathcal{N}(u)$ (Figure 1.6). A single iteration is governed by two differentiable functions, an aggregation and an update [20],

$$\mathbf{m}_{\mathcal{N}(u)}^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{\mathbf{h}_v^{(k)} : v \in \mathcal{N}(u)\right\}\right), \quad (1.16)$$

$$\mathbf{h}_u^{(k+1)} = \text{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right). \quad (1.17)$$

The aggregation function in Equation (1.16) takes the embeddings of the neighbours of u at the current layer k and returns a single summary message $\mathbf{m}_{\mathcal{N}(u)}^{(k)}$. Because a node's neighbours have no natural ordering, this function must be permutation invariant, such as a sum, a mean, or a maximum. The update function in Equation (1.17) then combines this neighbourhood message with the node's own current embedding to produce its state for the next layer. The embeddings are initialised with the input features, $\mathbf{h}_u^{(0)} = \mathbf{x}_u$, and after K iterations each embedding carries structural and feature information drawn from its K -hop neighbourhood, so that the final output $\mathbf{z}_u = \mathbf{h}_u^{(K)}$ can be used for the downstream prediction task [20].

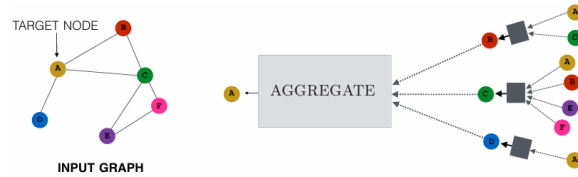


Figure 1.6: A single node aggregating messages from its local graph neighbourhood [20].

The basic graph neural network layer: The abstract aggregation and update functions are realised with trainable neural-network parameters. In its most basic form, a message-passing layer is

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right), \quad (1.18)$$

where $\mathbf{W}_{\text{self}}^{(k)}$ and $\mathbf{W}_{\text{neigh}}^{(k)}$ are trainable weight matrices for layer k , $\mathbf{b}^{(k)}$ is a bias vector, and σ is an element-wise non-linearity such as a rectified linear unit. Here the neighbourhood message is the sum

of the incoming neighbour embeddings, which is combined linearly with the node's own transformed embedding before the non-linearity is applied, in effect a multilayer perceptron adapted to a graph topology.

Generalised message passing: Modelling a physical system such as a mooring line often benefits from an explicit representation of the connections themselves, and not only of the nodes. The message-passing framework generalises to maintain, and update, an embedding for every edge alongside the node embeddings. In the formulation of Battaglia et al. [22], the network holds a hidden embedding $\mathbf{h}_{(u,v)}^{(k)}$ for each edge and, optionally, a global embedding $\mathbf{h}_G^{(k)}$ for the whole graph, and each iteration proceeds in stages [20],

$$\mathbf{h}_{(u,v)}^{(k)} = \text{UPDATE}_{\text{edge}}\left(\mathbf{h}_{(u,v)}^{(k-1)}, \mathbf{h}_u^{(k-1)}, \mathbf{h}_v^{(k-1)}, \mathbf{h}_G^{(k-1)}\right), \quad (1.19)$$

$$\mathbf{m}_{\mathcal{N}(u)}^{(k)} = \text{AGGREGATE}_{\text{node}}\left(\left\{\mathbf{h}_{(u,v)}^{(k)} : v \in \mathcal{N}(u)\right\}\right), \quad (1.20)$$

$$\mathbf{h}_u^{(k)} = \text{UPDATE}_{\text{node}}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}, \mathbf{h}_G^{(k-1)}\right), \quad (1.21)$$

$$\mathbf{h}_G^{(k)} = \text{UPDATE}_{\text{graph}}\left(\mathbf{h}_G^{(k-1)}, \left\{\mathbf{h}_u^{(k)} : u \in \mathcal{V}\right\}, \left\{\mathbf{h}_{(u,v)}^{(k)} : (u,v) \in \mathcal{E}\right\}\right). \quad (1.22)$$

The edge embeddings are updated first, from their previous state and the embeddings of their two incident nodes. The node update then aggregates the newly updated *edge* embeddings incident on u , rather than the raw neighbour features, before combining them with the node's own previous state, and a global embedding, when used, is refreshed from all node and edge embeddings. This edge-aware formulation underlies the architecture developed in this thesis, in which the mechanical properties of each line segment enter the network as edge features that actively shape the messages passed between adjacent nodes.

1.5. Research Gap and Thesis Contribution

1.5.1. Research Gap

Graph neural networks, and in particular their spatio-temporal variants, have recently proven effective across a range of offshore-wind structural and operational problems. Jiang and Chen [23] developed a graph-based digital twin that reconstructs the full stress field of an offshore wind turbine support structure in real time for structural health monitoring, showing that a graph model can recover a spatially distributed response quantity rather than a single scalar. Daenens et al. [24] applied a spatio-temporal graph neural network to operational supervisory control and data acquisition (SCADA) data to predict power across an offshore wind farm, using the graph to encode the spatial layout of the turbines together with their temporal interactions. Liu et al. [25] used a physics-guided graph neural network to predict the wake field behind a floating offshore wind turbine. Together these studies establish the graph as a natural and effective representation for spatially distributed offshore systems.

For mooring lines specifically, however, the picture remains incomplete. De Melo Costa et al. [26] applied a spectral graph-based network to a moored FPSO, but framed the problem as failure detection, a classification task that flags a broken line rather than reconstructing the continuous tension response. The graph surrogates closest to the present work, those of Yung et al. [27, 28], model the complete floating wind turbine and represent the mooring system only as a set of lumped reaction forces applied at the fairlead, so the distribution of load along the line itself is never resolved. No existing surrogate reconstructs the full spatial field of motion and tension along a mooring line, and none does so for lines of different discretisations with a single trained model.

This thesis addresses that gap with a surrogate that couples a graph attention network (GAT) spatial encoder to a long short-term memory (LSTM) temporal encoder, referred to throughout as the GAT+LSTM model. In place of a single fairlead quantity, it reconstructs the motion and tension of every node along the line. In place of one model per configuration, a single set of weights serves lines of different node counts and generalises to environmental locations not seen during training. In place of an objective that treats all errors equally, the training loss is shaped to remain conservative about the peak tensions that govern structural safety. The contribution is therefore not the architecture itself, since graph-attention

recurrent surrogates for floating structures already exist, but a topology-general full-field load surrogate together with a rigorous characterisation of what such a surrogate can and cannot certify.

1.5.2. Research Questions

To address this gap, this thesis poses the following main research question (MRQ).

MRQ: *How can a graph-attention LSTM architecture be formulated as a finite-element surrogate for offshore mooring lines, one that reconstructs the full spatial field of nodal motions and tensions across a range of line discretisations and environmental conditions, while remaining conservative with respect to the peak loads that govern structural safety?*

This question is addressed through four sub-questions (SQs).

- **SQ1 (Graph representation and generalisation):** How can the geometry and mechanical properties of a mooring line be encoded as a graph, and to what extent can a single set of network weights serve mooring configurations of differing node counts and generalise to environmental locations not seen during training?
- **SQ2 (Spatial reconstruction and its meaning):** With what accuracy can the coupled graph-attention and recurrent encoders reconstruct the spatial distribution of motion and tension along the line, and which components of the response, its spatial profile, its per-sea-state magnitude, and its wave-frequency fluctuation, does that accuracy actually certify?
- **SQ3 (Safety-aware objective):** How can soft penalty terms in physical units, together with a multi-task uncertainty-weighting scheme, be combined in the training objective to bias the surrogate toward conservative prediction of the peak tensions that govern mooring design, without manual tuning of the individual loss weights?
- **SQ4 (Value and consistency):** What does the surrogate provide beyond a trivial persistence baseline, how physically consistent are its reconstructed fields, and where does its value lie relative to the finite-element solver it is intended to emulate?

1.5.3. Contribution and Scope

Two points delimit the nature of this contribution. The first concerns how its safety-oriented behaviour arises, namely through the training objective rather than through the architecture. That objective combines a data-fidelity term with several soft penalty terms expressed in physical units, not physical constraints embedded in the model, so the network remains data-driven and the terms bias its predictions toward physically reasonable behaviour without enforcing the governing equations.

The second concerns the intended use of the surrogate, whose scope is deliberately bounded. It is developed and evaluated as a spatial load-field emulator for chain mooring lines across a range of discretisations and sea states. It is not intended to act, and is shown in Chapter 4 not to act, as a wave-by-wave forecaster of the tension time history at a fixed point. The forecasting-oriented extensions that this boundary motivates, together with the reconstruction of dense fields from sparse sensing, are set out as future research directions in Chapter 7.

1.6. Research Methodology

The methodology of this thesis follows a single pipeline that turns high-fidelity simulation data into a validated surrogate model, and this same pipeline organises the chapters that follow. Its stages are summarised in Figure 1.7, which traces the path from the environmental and structural inputs, through the construction of the training dataset and the graph-based model, to the two training strategies that are compared and the final model that is analysed. Each stage is developed in detail in a later chapter, and the remainder of this section walks through the path at a high level.

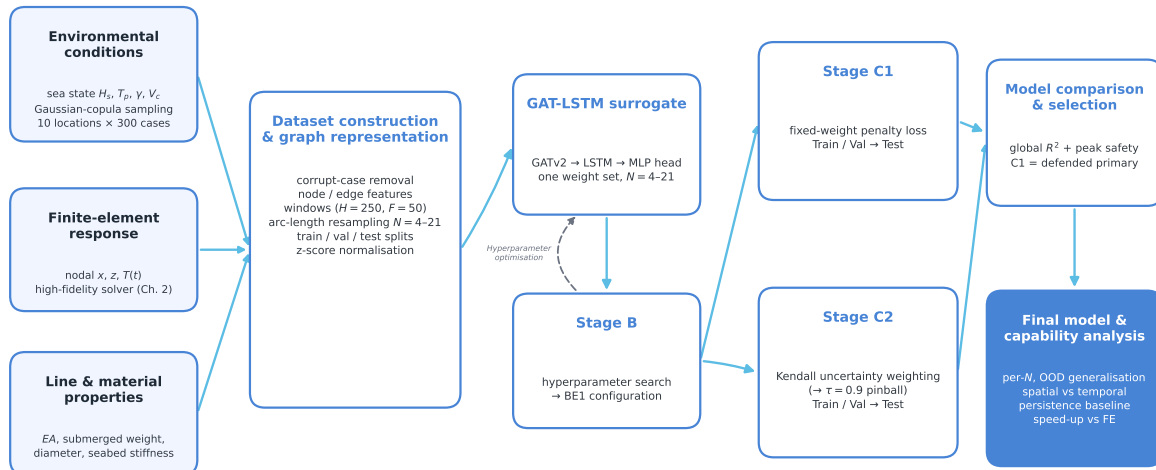


Figure 1.7: Overview of the research methodology.

Data generation: The starting point is a set of high-fidelity simulations. Environmental conditions are drawn for ten offshore locations of differing water depth, with the sea state of each case obtained from a Gaussian-copula model so that the joint distribution of wave height, wave period, and current is preserved rather than sampled from a small set of design points. Three hundred load cases per location are simulated with the finite-element solver of Chapter 2, which resolves the nonlinear mooring dynamics and returns the time histories of position and tension at every node. These simulations are the ground truth that the surrogate is trained to reproduce.

Dataset construction and graph representation: The raw simulations are then organised into a machine-learning dataset. Corrupt cases are removed, static and dynamic features are attached to the nodes and edges of the mooring-line graph, and the response is cut into overlapping history and future windows. Because the model must serve lines of different node counts, each case is resampled by arc length onto the target discretisation, and the windows are divided into training, validation, and test sets, with two locations held out entirely to probe generalisation to unseen conditions. This construction is the subject of Chapter 2.

Model and training: The surrogate itself is a hybrid graph-attention and recurrent network, developed in full in Chapter 3. A graph-attention encoder captures the spatial coupling between nodes, a recurrent encoder captures the temporal evolution of each node, and a small prediction head returns the future motion and tension of the whole line in a single pass. The training proceeds in two stages. Stage B is a hyperparameter search that fixes the architecture, carried out by changing one factor at a time and confirming the outcome across several random seeds, which yields the configuration used thereafter. Stage C then trains that configuration under two different loss strategies, a fixed-weight objective and an uncertainty-weighted objective, so that the effect of the safety-oriented terms can be isolated.

Evaluation and selection: The two trained models are compared on the held-out test set, using both accuracy and the peak-tension safety metrics that matter for design, and the more reliable of the two is retained as the primary model. Its behaviour is then characterised in depth in Chapter 4, where its accuracy is decomposed into the spatial and temporal components it does and does not capture, it is measured against a trivial persistence baseline, and its physical consistency and computational cost are assessed.

Thesis outline: The structure of the remainder of this thesis follows the same path. Chapter 2 describes the finite-element simulation framework, the environmental sampling, and the construction of the dataset. Chapter 3 develops the model architecture, the feature engineering, the loss function, and the two training stages. Chapter 4 reports the performance of the final model together with the analysis of its capabilities and limits. Chapter 5 discusses what these results mean for the use of the surrogate, Chapter 6 answers the research questions and states the contributions, and Chapter 7 sets out the research directions that the capability boundary of the present model motivates.

2

Data

2.1. Simulation Framework

The mooring-line time histories used to train and evaluate the GAT+LSTM model were generated with the geometrically nonlinear finite-element model GNL-FEM introduced by Agarwal et al. [29]. The formulation is based on *Tangential Differential Calculus* (TDC), following the finite-strain rope and membrane framework of Fries and Schöllhammer [30], as adapted there to mooring-line dynamics. TDC allows differential operators, strain measures, and internal stresses to be defined consistently on a one-dimensional curved manifold embedded in two-dimensional Euclidean space. The equilibrium is enforced in the deformed configuration, while the finite-element problem is formulated on the undeformed (reference) configuration. The implementation is written in Julia using the open-source finite-element library Gridap.jl [31].

2.1.1. Continuum Formulation and Kinematics

The mooring cable is modelled as a one-dimensional continuum in $\mathbb{R}^2$. The undeformed configuration is denoted Γ_X and the deformed configuration Γ_x . A material point at $X \in \Gamma_X$ moves to its current position $x \in \Gamma_x$ according to:

$$x(X, t) = X + u(X, t) \quad (2.1)$$

where the displacement vector is written $u = [u_x, u_z]^T$ in terms of its horizontal and vertical components. Both configurations are parameterised through a one-dimensional reference domain Ω_r via the mapping $X(r) : \Omega_r \rightarrow \Gamma_X \subset \mathbb{R}^2$. Figure 2.1 shows the resulting kinematic setup.

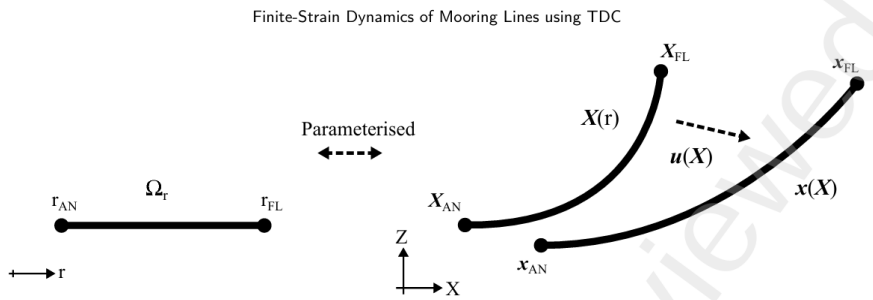


Figure 2.1: Kinematic setup of the TDC cable model. The one-dimensional reference domain Ω_r is mapped to the undeformed configuration Γ_X via $X(r)$ and to the deformed configuration Γ_x via $x(r, t)$, with $u = x - X$ denoting the displacement field [29].

The TDC framework constructs geometric quantities along the cable from the Jacobian of this mapping:

$$J(r) = \frac{\partial X}{\partial r}, \quad T = \frac{J}{|J|} \quad (2.2)$$

where T is the unit tangent to the undeformed cable. The matrices $G = J^T J$ and $Q = JG^{-1}$ are used to form the directional surface gradient of the displacement field:

$$\nabla_{\Gamma_X}^{\text{dir}} u = \nabla_r u Q^T \quad (2.3)$$

which replaces the standard Cartesian gradient for quantities defined only on the curved cable manifold. The surface deformation gradient:

$$F_\Gamma = I + \nabla_{\Gamma_X}^{\text{dir}} u \quad (2.4)$$

maps tangent vectors from Γ_X to Γ_x . The stretch ratio $\Lambda = |t^*|/|T^*|$, where T^* is a tangent vector of the undeformed cable and $t^* = F_\Gamma T^*$ its image in the deformed configuration, quantifies the elongation of a differential cable element [29].

2.1.2. Strain and Stress Measures

The model employs finite-strain kinematics. The directional Green–Lagrange strain tensor in the undeformed configuration is:

$$E_{\text{dir}} = \frac{1}{2} (F_\Gamma^T F_\Gamma - I) \quad (2.5)$$

and its projection onto the unit tangent direction extracts the mechanically relevant axial component:

$$E_{\text{tang}} = P E_{\text{dir}} P, \quad P = T \otimes T \quad (2.6)$$

The cable material is modelled as Saint Venant–Kirchhoff elastic, for which the second Piola–Kirchhoff stress takes the general form:

$$S = \lambda \text{tr}(E_{\text{tang}}) P + 2\mu E_{\text{tang}} \quad (2.7)$$

where λ and μ are the Lamé constants, expressed through the Young's modulus E and the Poisson ratio ν of the chain material. For the cable formulation, $\lambda = 0$ and $\mu = E/[2(1 + \nu)]$, so this reduces to $S = 2\mu E_{\text{tang}}$. The first Piola–Kirchhoff stress $K = F_\Gamma S$ relates forces in the deformed configuration to area elements in the undeformed configuration. Together these two-point stress measures allow the model to represent large rotations and large deformations while keeping the constitutive law linear in the Green–Lagrange strain. These kinematic and constitutive definitions follow the TDC finite-strain mooring-line formulation of Agarwal et al. [29].

2.1.3. Dynamic Equilibrium and External Loading

The dynamic equilibrium is first written in the deformed configuration:

$$\rho \ddot{u} - \text{div}_{\Gamma_x} \sigma(u) - b(x) = 0 \quad \forall x \in \Gamma_x, \quad (2.8)$$

where ρ is the material density, σ is the Cauchy stress, and b collects distributed external forces. Four physical load contributions are included:

1. **Self-weight:** $b_{\text{weight}} = -\rho A g \hat{k}$
2. **Buoyancy:** $b_{\text{buoy}} = \rho_{\text{water}} A_{\text{disp}} g \hat{k}$
3. **Seabed contact**, modelled with a spring-dashpot law activated only when the cable penetrates the seabed:

$$b_{\text{bed}} = \begin{cases} -k_b (z - z_{\text{bed}}) \hat{k} - c_b \dot{z} \hat{k} & z \leq z_{\text{bed}} \\ 0 & z > z_{\text{bed}} \end{cases} \quad (2.9)$$

4. **Hydrodynamic drag**, decomposed into components normal and tangential to the cable axis using a Morison-type expression:

$$b_{\text{drag},n} = \frac{1}{2} \rho_{\text{water}} C_{d,n} v_{r,n} |v_{r,n}|, \quad b_{\text{drag},t} = \frac{1}{2} \rho_{\text{water}} C_{d,t} v_{r,t} |v_{r,t}|, \quad (2.10)$$

where $v_r = v_{\text{wave}} + v_{\text{current}} - \dot{u}$ is the relative fluid velocity and $v_{r,n}$, $v_{r,t}$ are its normal and tangential projections, respectively. Added mass and added damping are excluded from the formulation.

At the domain boundaries, the anchor end is held fixed (zero displacement, a Dirichlet condition), while the fairlead end follows a prescribed motion time series that supplies the dynamic excitation (Section 2.1.5). The material and geometric parameters entering the load terms above are constants of the physical mooring system [29]. Those of them that the surrogate model receives as inputs, namely the cross-sectional area, the line diameter, the drag coefficients, and the derived nodal mass and submerged weight, are listed in Table 3.1 (Chapter 3). Figure 2.2 shows the complete configuration together with the four distributed loads defined above.

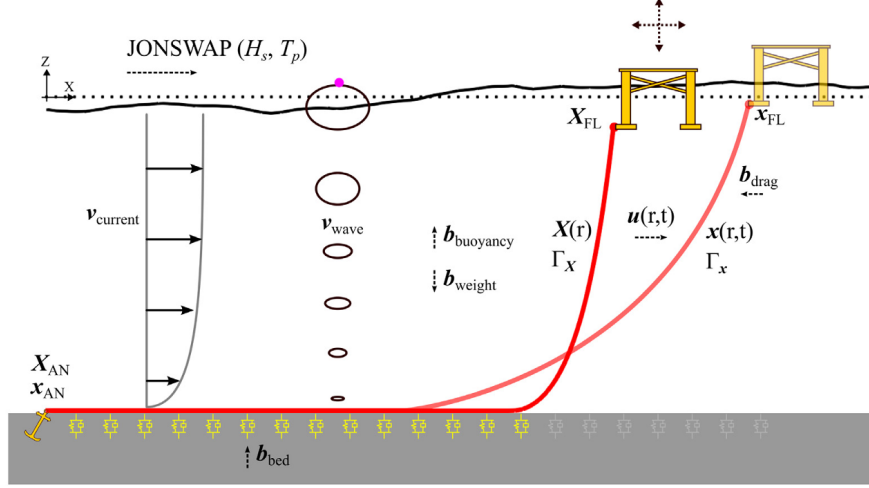


Figure 2.2: Schematic of the catenary mooring-line configuration modelled by GNL-FEM, illustrating the four distributed external loads, self-weight, buoyancy, hydrodynamic drag, and the spring-dashpot seabed contact force, applied along the line between the fairlead and the anchor [29].

Pulling Equation (2.8) back to the undeformed configuration through the first Piola–Kirchhoff stress yields the strong form underlying the finite-element discretisation:

$$\rho_0 \ddot{u}(X) - \text{Div}_{\Gamma_X} K(u) - B(X) = 0 \quad \forall X \in \Gamma_X \quad (2.11)$$

where ρ_0 is the reference density and $B(X)$ is the external force vector expressed in the undeformed configuration.

2.1.4. Weak Form and Finite-Element Discretisation

The weak form is obtained by multiplying Equation (2.11) by a test function $w \in H^1(\Gamma_X)$, integrating over the undeformed cable domain, and applying integration by parts on the manifold. In the manifold integration-by-parts operation, a curvature contribution appears in the intermediate form, but it vanishes for in-plane tensors such as the first Piola–Kirchhoff stress K . This leads to the final weak form used in the numerical implementation. Expressed in the one-dimensional reference domain Ω_r , the variational statement is to find a discrete displacement field u_h for which

$$A \int_{\Omega_r} \rho_0 w_h \cdot \ddot{u}_h |J| d\Omega + A \int_{\Omega_r} (\nabla_r w_h Q^T) : K(u_h) |J| d\Omega = A \int_{\Omega_r} w_h \cdot B |J| d\Omega + \int_{\partial\Omega_r} w_h \cdot H d(\partial\Omega) \quad (2.12)$$

holds for all admissible test functions w_h , where A is the cross-sectional area and H is the prescribed Neumann traction at the cable boundary. The finite-element space consists of continuous, element-wise Lagrange polynomial shape functions of order q , denoted $\mathbb{P}_q$, defined over the discretised reference mesh $\Omega_{r,h}$. The reference domain is discretised as a regular one-dimensional grid that is mapped to the undeformed cable geometry Γ_X . Because the stress $K(u_h)$, the hydrodynamic drag, and the seabed contact all depend nonlinearly on displacement and velocity, the resulting semi-discrete system is nonlinear. The weak form and its reference-domain finite-element discretisation follow the GNL-FEM formulation of Agarwal et al. [29].

2.1.5. Time Integration and Nonlinear Solution

Before the dynamic simulation is started, the cable is brought to its static equilibrium by solving the nonlinear static version of (2.12) under self-weight, buoyancy, and seabed contact. The resulting displacement and internal force state serve as initial conditions for the dynamic run.

Time integration is performed with the Generalised- α method [32, 33], which introduces controlled numerical dissipation of high-frequency oscillations while retaining accuracy at the physically relevant low frequencies. In GNL-FEM, Agarwal et al. use this scheme with $\rho_\infty = 0$, so that modes above approximately $f = 1/(10 \Delta t)$ are effectively damped, preventing spurious high-frequency content from polluting the solution [29]. A constant time step Δt is used throughout each simulation. At each time step the nonlinear algebraic system is solved using Newton–Raphson iterations [29]. Dynamic excitation is introduced through prescribed fairlead motion.

GNL-FEM was validated against MoorDyn [14], a widely used lumped-mass mooring-line solver, for catenary-line configurations subject to sinusoidal fairlead excitation [29]. The comparison shows good agreement at typical wave-frequency excitation, while larger differences are observed under very high-frequency excitation and strong slack-line dynamics, indicating that the two modelling approaches may diverge under these conditions [29]. A reduced-order analytical companion model, GNL-ANA, is also presented by Agarwal et al. [29] for the special case of a taut vertical line, using a Galerkin modal expansion and a Runge–Kutta time integrator. GNL-ANA is useful for rapid parametric studies but is not used to generate the dataset in this thesis.

For the dataset used in this thesis, the relevant stored GNL-FEM outputs are nodal positions $(x_{\text{abs}}, z_{\text{abs}})$ and line tension at each discretisation node. These outputs, sampled at a fixed time step, constitute the supervised-learning targets used to train the GAT+LSTM model. The environmental conditions under which the simulations were run and the resulting dataset structure are described in Sections 2.2–2.6.

2.2. Environmental Condition Sampling

Each GNL-FEM mooring-line simulation corresponds to one irregular sea-state realisation, defined by a set of environmental variables that fully specify the wave and current forcing at a given location. The environmental inputs required by the simulator are water depth h_0 , significant wave height H_s , peak wave period T_p , five current magnitudes $v_1, \dots, v_5$, and the five corresponding current-sampling depths $d_1, \dots, d_5$ through the water column. Together these thirteen variables constitute the forcing definition for one simulated case.

Hydrodynamic data source: The environmental conditions are derived from the Dutch Continental Shelf Model (DCSM), a large-scale three-dimensional hydrodynamic model covering the north-western European continental shelf [34, 35]. The DCSM uses the D-Flow Flexible Mesh (D-Flow FM) component of the Delft3D Flexible Mesh Suite [36], with a spatially varying horizontal grid resolution that is coarser in deep offshore waters and refined near shallow coastal areas. Wave conditions are provided by the SWAN spectral wave model [37], coupled on top of the hydrodynamic solver to account for shallow-water wave processes. Current and wave datasets from the coupled model cover the period 2013–2017 [35].

Environmental samples representative of energetic sea states are constructed as follows, following the Gaussian-copula-based Bayesian Network (GCBN) approach of Santjer et al. [35]. Daily maxima of significant wave height H_s are first identified from the historical DCSM wave record, together with the concomitant peak wave period T_p and the current magnitudes at the five sampling depths, taken as maxima within a ± 4 hour window centred on the daily wave-height maximum. These historical daily extremes are used to fit, per location, a Gaussian-copula-based Bayesian Network over the full set of thirteen environmental variables, describing their joint dependence structure. For each location, $n = 300$ environmental cases are then drawn from the fitted network rather than taken directly as the historical daily extremes themselves. This procedure produces sea-state samples that are simultaneously physically consistent, since the Gaussian copula preserves the empirical dependence between wave height, period, and current profile, and more representative of rare, design-relevant extremes than the underlying five-year historical record alone, which by construction under-samples the tail of the extreme-event distribution [35].

Copula-based versus independent synthetic sampling: An alternative synthetic-sampling design would draw each environmental variable independently, for instance via Latin Hypercube sampling of $(H_s, T_p, v_1, \dots, v_5)$, which guarantees uniform coverage of the feasible region but breaks the physically observed dependence between wave height, period, and current, potentially producing hydrodynamically implausible combinations. The Gaussian-copula-based Bayesian Network avoids this by sampling from a fitted joint distribution rather than the marginal ranges of each variable separately, so the generated cases remain physically consistent while still representing the full range of the fitted distribution, including its tails, rather than only the frequency with which extreme combinations happened to occur in the five-year historical record. This directly addresses the concern that a purely frequency-based extraction would under-represent rare, high-severity conditions relevant to mooring-line design.

Wave and current forcing: For each sampled case, the irregular wave environment is described by a JONSWAP spectrum parameterised by H_s and T_p . A wave time series is generated from the spectrum via an inverse Fourier transform, and wave-particle velocities are computed using linear wave theory. The current velocity profile is defined by the five magnitudes $v_1, \dots, v_5$ at depths $d_1, \dots, d_5$ and interpolated over the full water column using the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) method [35]. The first depth d_1 lies approximately 1 m above the seabed and d_5 is near the free surface. The top and bottom 1 m of the water column are excluded from the profile definition. Waves and currents are assumed to propagate in the same direction as the mooring line, which represents a conservative loading scenario for the cable [35].

Selected locations: The original study considered multiple south-eastern North Sea locations with different water depths and environmental statistics [35]. In this thesis, ten of those locations are used, with identifiers:

$$\text{loc} \in \{1, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

Table 2.1 lists the water depth and current-sampling depths for each location. These values are taken directly from the input data files used in this thesis and may differ slightly from those reported by Santjer et al. [35].

Table 2.1: Water depths and current-sampling depths for the ten thesis locations.

Location	h_0 [m]	d_1 [m]	d_2 [m]	d_3 [m]	d_4 [m]	d_5 [m]
loc01	30.096	1	8	15	22	29
loc03	35.662	1	9	18	26	35
loc04	44.669	1	12	22	33	44
loc05	39.195	1	10	20	29	38
loc06	14.460	1	4	7	10	13
loc07	40.724	1	11	20	30	40
loc08	30.163	1	8	15	22	29
loc09	33.578	1	9	17	25	33
loc10	14.998	1	4	7	11	14
loc11	38.614	1	10	19	28	38

The water depths range from approximately 14 m at loc06 and 15 m at loc10 to 45 m at loc04, covering a realistic range of near-shore and mid-shelf deployment conditions. The current-sampling depths are scaled with the local water depth, with d_1 close to the seabed and d_5 close to the free surface.

Case definition and dataset stages: For each location, the environmental sampling procedure generates $n = 300$ sea-state realisations, referred to as *cases* [35]. Each case is identified by a location index l and a 1-indexed case number, forming the pair (l, case) used throughout the dataset. The thirteen environmental variables $(h_0, H_s, T_p, d_1, \dots, d_5, v_1, \dots, v_5)$ are stored in a semicolon-delimited CSV file per location, with one row per case. The column SN contains the 1-indexed case number, h_0 gives the water depth, H_s and T_p define the JONSWAP sea state, and the pairs (d_1, v_1) through (d_5, v_5) specify the discrete current profile.

The thesis distinguishes two dataset stages based on the number of cases used per location:

- **Stage B:** the first 50 cases per location are used for the hyperparameter optimisation experiments described in Chapter 3 (Section 3.9 defines the search precisely).
- **Stage C:** all 300 cases per location are used for the final model training and evaluation reported in Chapter 4.

Each sampled environmental realisation is passed to the geometrically nonlinear finite-element mooring-line model of Agarwal et al. [29], as used in the environmental sampling framework of Santjer et al. [35], to produce mooring-line time histories of nodal positions and tension. These simulated time histories are subsequently preprocessed and organised into the graph-temporal samples required by the GAT+LSTM model, as described in Sections 2.4–2.6.

2.3. Raw Data Structure

The GNL-FEM simulation outputs are organised in a two-level directory hierarchy. Each of the ten locations occupies a dedicated top-level folder, within which the individual simulation cases are stored in numbered sub-directories. The number of cases per location is $n_{\text{case}} = 50$ for Stage B and $n_{\text{case}} = 300$ for Stage C. The symbol N is reserved throughout for the number of discretisation nodes on a mooring line (Chapter 3).

Each case sub-directory contains a single `.dat` output file produced by the GNL-FEM solver. The file is structured as a two-dimensional array with one row per time step. Each row stores the dynamic state of the mooring line at that instant, comprising the horizontal and vertical nodal positions ($x_{\text{abs}}, z_{\text{abs}}$) and the scalar line tension at each of the 21 discretisation nodes, together with the environmental forcing variables, wave kinematics and current velocities, evaluated at the same instant.

The time series comprises 13 800 records at a uniform interval of $\Delta t = 0.1$ s, giving a total simulation duration of 1 380 s (23 minutes) per case. The first record, at $t = 0.1$ s, corresponds to the static equilibrium configuration of the cable under self-weight, buoyancy, and seabed contact, computed prior to any dynamic loading (see Section 2.1.5). This static snapshot serves as the initial condition from which the dynamic wave and current excitation is applied for every case. The remaining 13 799 records capture the transient mooring-line response under the irregular sea state defined by the case’s environmental variables.

All cases are output at the standard spatial resolution of 21 nodes along the mooring-line arc, regardless of the topology used during model training. The conversion from this uniform 21-node representation to the variable-resolution graph inputs required by the GAT+LSTM model is performed during preprocessing, as described in Section 2.4.

2.4. Preprocessing

Binary cache conversion: The raw `.dat` output files are text-based and require line-by-line parsing each time they are accessed. For a dataset of this scale, ten locations with up to 300 cases each and 13 800 time steps per case, repeated parsing during model training would be prohibitively slow. Each `.dat` file is therefore converted once to a binary `.npy` array and stored in a dedicated cache directory. The `.npy` format supports memory-mapped access, which allows the operating system to load only the portions of the array that are actually needed at any given moment, without bringing the full time series into RAM. This is essential because the complete dataset exceeds available system memory. Memory-mapped loading keeps RAM usage bounded regardless of dataset size, with data paged in and out transparently as training proceeds.

Spatial resampling to variable node counts: Every GNL-FEM simulation produces output at 21 nodes along the mooring-line arc. In a real offshore deployment, however, continuous knowledge of the cable state at all 21 positions simultaneously is not achievable. Physical sensors provide measurements at only a handful of discrete locations along the line, and deploying the number of instruments required to resolve 21 points in real time is neither practical nor economical. A surrogate model intended for real-time monitoring must therefore be capable of producing accurate predictions from sparse, low-resolution inputs.

To enable this, each 21-node simulation output is resampled to ten target resolutions:

$$N \in \{4, 5, 6, 7, 8, 10, 12, 15, 18, 21\}$$

by interpolating nodal positions and tensions onto evenly spaced arc-length coordinates. This converts every simulated case into ten separate training instances, one per target topology, all derived from the same underlying FE solution and sharing the same environmental forcing. The technical implementation of the resampling is described in Chapter 3, where it is presented as part of the model data pipeline.

2.5. Data Quality and Corrupt Case Handling

In a small fraction of simulated cases the GNL-FEM solver undergoes numerical divergence during the Newton–Raphson iterations, producing tension values that are physically impossible. These diverged cases corrupt the feature normaliser. The tension standard deviation fitted on the training data is inflated by several orders of magnitude, rendering all physical-unit accuracy metrics unreliable.

Why exclude at the input rather than filter at the output: These diverged cases are not valid physical solutions to be learned and later filtered. They are numerical artefacts of the Newton–Raphson iteration failing to converge, and their tension values bear no reproducible relationship to the input sea-state variables. Because a supervised model learns the input-output mapping present in its training data, including such cases would teach the network to reproduce arbitrary numerical noise rather than genuine mooring-line physics, and would inflate the tension normalisation statistics (Section 3.3.5) by orders of magnitude, degrading the input scaling for every other case. Filtering after training, at the model’s output, cannot repair this, since the corrupted statistics and spurious input-output associations would already be embedded in the fitted weights. Exclusion prior to training is therefore the only point at which these cases can be removed without contaminating the model.

Initial screening (Stage B): When the first 50 cases per location were assembled for the Stage B hyperparameter search, each cached tension file was screened by computing the maximum absolute tension recorded across all nodes and time steps. Any case with a peak tension exceeding 5 MN was flagged as diverged. The affected cases exhibit tensions in the gigaNewton to petaNewton range, several orders of magnitude above the physically expected tens to hundreds of kilonewtons. This initial screen identified 36 corrupt cases spanning nine of the ten locations, with location 3 containing no corrupt cases. All 36 were added to a blacklist and excluded from every dataset split before any training began.

Extended rescan for Stage C: When the dataset was expanded to 300 cases per location for Stage C, a full rescan of all cached tension files was performed. The original magnitude-only screen had two blind spots, both of which became apparent only at the 300-case scale.

1. **Large-finite divergence in the 1.4–5 MN range:** Several cases recorded finite peak tensions below the original 5 MN cut, yet still far exceeding any physically plausible mooring load. These passed the original screen undetected.
2. **Non-finite values:** A cluster of 11 cases concentrated in location 7 contained not-a-number or infinity entries in their tension arrays, while their largest finite tension remained in the normal range of 13–17 kN. A magnitude-only scan returns a finite result for such files and passes them undetected. In practice, a single non-finite value in a sampled training window propagates through the Huber loss computation and yields a non-finite total loss, halting the optimiser. This failure mode was confirmed when the first Stage C training run crashed at epoch 20 due to a non-finite total loss.

The rescan therefore checked for the presence of any non-finite element in the tension array and for any finite peak tension exceeding a revised magnitude threshold.

Threshold calibration: A global magnitude threshold alone cannot cleanly separate physically valid extreme events from diverged cases because the raw tension distribution has a heavy legitimate tail

with no obvious gap. Peak tensions in the valid range reach up to the order of hundreds of kilonewtons, and approximately 1 % of cases naturally produce peaks exceeding 100 kN, confirming that high tension values are not in themselves evidence of solver divergence. A fixed global cut would therefore either exclude genuine extreme events or permit lightly diverged cases to pass.

The threshold was instead calibrated relative to each location's 99th-percentile peak tension, taken as the reference for the typical maximum load at each site. Cases whose peak tension lies within eight times this value are treated as physically plausible, while cases exceeding it are treated as numerically diverged. Two entries, however, required individual consideration. Case 83 at location 8 records a peak of 1.42 MN, equal to 8.4 times the 99th percentile for that location, placing it marginally beyond the eight-times limit. Mooring snap loads, arising when a previously slack section of line suddenly becomes taut, are known to produce short-duration tension amplifications substantially above the cyclic mean, and this ratio is consistent with such an event. The case was therefore retained. Case 221 at location 7 records 2.6 MN, equal to 4.5 times that location's 99th-percentile peak of 584 kN, which falls within the eight-times rule. It is, however, the only case at location 7 that exceeds 1 MN, with the next highest case peaking at 932 kN. An isolated jump of this magnitude in an otherwise consistent distribution is a recognised signature of solver divergence rather than a physical extreme event, and the case was therefore excluded. All remaining finite cases above 1.0 MN lie between 20 and 95 times their location's 99th percentile, placing them clearly outside any physically explainable range. A threshold of 1.5 MN was therefore selected, retaining case 83 as a plausible snap-load event while excluding all numerically diverged cases. No additional finite cases fall between 1.0 MN and 1.42 MN, so any cut placed in this interval would yield an identical set of excluded cases.

Table 2.2 shows the distribution of per-case peak tensions across all 2 953 finite cases below the original 5 MN screening level. The global median of 10.7 kN and 90th percentile of 14.3 kN confirm that typical sea states produce moderate mooring loads well within the expected operational range. The 99th percentile of 103 kN reflects natural variability at the more energetic locations.

Table 2.2: Distribution of per-case peak tensions across the 2 953 finite simulation cases below the original 5 MN screening level.

Location	p_{50} (kN)	p_{90} (kN)	p_{99} (kN)
loc01	8.8	11.0	60.8
loc03	10.8	13.6	17.8
loc04	13.6	16.7	364.8
loc05	11.9	14.4	103.1
loc06	3.6	4.6	22.4
loc07	12.3	14.9	584.3
loc08	8.8	11.2	169.6
loc09	10.2	13.0	78.3
loc10	3.6	4.7	37.0
loc11	11.6	14.8	68.7
All	10.7	14.3	103.0

Final blacklist: Applying the combined criterion (non-finite values or peak tension > 1.5 MN) to the full 300-case pool identified 53 corrupt cases, listed in Table 2.3. Compared to the initial Stage B blacklist of 36, the Stage C rescan added 17 entries, six finite cases in the 1.4–5 MN range and eleven non-finite cases in location 7. As a secondary safeguard, each training batch is checked for a non-finite loss value before the backward pass, and any such batch is counted and skipped. With the 53-case blacklist in place, that counter remains at zero throughout all training runs, confirming that the blacklist is complete.

Table 2.3: Corrupt simulation cases excluded from all dataset splits. A case is flagged if it contains non-finite tension values or if its peak tension exceeds 1.5 MN.

Location	Corrupt case IDs
loc01	63, 202, 215
loc04	45, 133, 164, 270
loc05	31, 35, 46, 108, 194, 197, 202, 262, 264
loc06	22, 131, 134
loc07	16, 32, 83, 132, 137, 174, 212, 214, 220, 221, 225, 229
loc08	37, 94, 136, 143, 247
loc09	43, 167, 193
loc10	22, 56, 77, 187, 188, 224, 234, 239, 275, 278
loc11	3, 22, 61, 89

2.6. Dataset Construction

Sliding window extraction: Each preprocessed time series of 13 800 records is divided into non-overlapping blocks of $L = 500$ time steps. Within each block, all valid sliding windows are extracted with a fixed history length of $H = 250$ steps and a prediction horizon of $F = 50$ steps, yielding

$$n_w = L - H - F + 1 = 201 \text{ windows per block,}$$

extracted at every valid stride-1 offset so that consecutive windows overlap by $H + F - 1$ steps, providing a dense, augmented sampling of each block's dynamics. Only the outer $L = 500$ -step blocks themselves are non-overlapping. With 27 complete blocks per case ($\lfloor 13\,800/500 \rfloor = 27$), this gives approximately $27 \times 201 = 5\,427$ candidate windows per (location, case, node count) triplet before any cap is applied.

The history and horizon lengths are set by the physics of the response. At the simulation time step of $\Delta t = 0.1$ s, the history window spans $H \times \Delta t = 25$ s, which captures approximately five oscillation cycles of the mooring-line response, as observed in the node-position and tension time series. The prediction horizon of $F \times \Delta t = 5$ s then covers approximately one such response cycle.

The block length $L = 500$, by contrast, is not physically motivated. It is a partition unit for the train and validation split described below, chosen to be long enough to contain many complete windows per block while short enough that block-level shuffling produces a fine-grained, well-mixed split. No window is allowed to span a block boundary, which ensures that the history sequence of every sample is drawn from contiguous simulation data and that no future information leaks into the history. Because the 500-step blocks are later shuffled independently into the training and validation pools, a window whose history spanned a block boundary could draw part of its 25 s history from a neighbouring block assigned to a different split, silently mixing training and validation data. Forbidding windows from crossing a block boundary rules this out entirely.

Dataset objects and lazy loading: One dataset object is created per (location, case, target N) triplet. Each object stores only the path to the memory-mapped `.npy` file and the corresponding window index. The numerical data is read from disk only when a specific window is requested during training. This design keeps RAM usage negligible at dataset construction time regardless of the total pool size, and allows the full 300-case Stage C dataset to be handled without modification to the loading code.

Train, validation, and test splits: The splits are constructed along two orthogonal axes, temporal position within each case and location identity, to evaluate generalisation across both time and environmental conditions.

For each case belonging to the eight development locations (1, 3, 4, 5, 6, 7, 8, and 9), the time series is divided at the 70 % mark. The first 70 % of time steps, approximately 9 660 steps covering the first 16 minutes of each sea state, form the development set, while the last 30 % form a temporal test set drawn from the later portion of each realisation. Within the development set, the 500-step blocks are

randomly shuffled and partitioned into training (70 % of blocks) and validation (30 % of blocks). This block-level shuffle prevents temporally adjacent blocks from the same case from appearing in both pools simultaneously. Because blocks are assigned to training or validation independently of their temporal position within the case, and every case belonging to a development location contributes blocks to both pools, the two splits draw from the same underlying distribution of sea-state dynamics by construction, rather than relying on a post-hoc statistical check.

Locations 10 and 11 are withheld entirely, with all their windows assigned to the test pool, providing an out-of-distribution test of generalisation to unseen geographical locations with different environmental statistics. The final test pool therefore combines temporal holdout windows from the eight development locations and all windows from the two held-out locations. Locations 10 and 11 span a shallow ($h_0 = 15.0$ m) and a mid-depth ($h_0 = 38.6$ m) site respectively, both within the depth range already spanned by the eight development locations (14.5–44.7 m, Table 2.1). The out-of-distribution test therefore evaluates generalisation to unseen environmental statistics at previously interpolated depths, rather than extrapolation to depth extremes never seen during training. A depth-stratified holdout targeting the shallowest and deepest sites specifically was not performed and is left as a sharper test of extrapolation for future work.

Window caps and dataset summary: Because the total number of candidate windows far exceeds what can be processed in a feasible training run, each pool is capped. The training sampler draws a fresh random subset, of the per-epoch size given in Table 2.4, at the start of every epoch, so the full pool is covered progressively across epochs. Table 2.4 summarises the window budgets for Stage B and Stage C.

Table 2.4: Window budgets for Stage B (50 cases per location) and Stage C (300 cases per location). The per-epoch sample is redrawn randomly each epoch from the training pool.

Split	Stage B	Stage C
Training pool	20 000	500 000
Per-epoch sample	10 000	30 000
Validation	5 000	10 000
Test	5 000	30 000

The twenty-five-fold increase in the training pool from Stage B to Stage C, which follows the six-fold increase in the number of available cases per location, reflects the full 300-case dataset becoming available, while the per-epoch cap keeps individual epoch duration manageable on the A100 GPU. Together, Sections 2.1–2.6 describe the complete data pipeline, from FE simulation to the graph-temporal windows consumed by the GAT+LSTM model.

Model Architecture and Methodology

The finite-element solver described in Chapter 2 resolves the dynamics of a mooring line by integrating the semi-discrete equations of motion at every time step, which makes a single irregular sea-state realisation expensive to compute. The objective of this chapter is to specify a surrogate model that replaces this solver with a learned spatio-temporal map. Given a short history of the line state, the model predicts the subsequent motion and tension of every node directly, at a small fraction of the computational cost. Because the mooring line is naturally a chain of discrete masses connected by elastic segments, it is represented as a graph, and the surrogate is built as a hybrid spatio-temporal graph neural network. A graph attention encoder first captures the instantaneous spatial coupling between nodes, and a recurrent encoder then captures the temporal evolution of each node along the history window. Yung et al. [27, 28] have recently shown the same construction, a physics-guided spatio-temporal graph neural network acting as a finite-element surrogate, to be effective for predicting the response of floating offshore wind structures. The present work adapts the idea to mooring-line dynamics and, in contrast to those fixed-topology surrogates, trains a single model that generalises across mooring discretisations ranging from four to twenty-one nodes.

This chapter is organised as follows. Section 3.1 states the prediction task formally. Section 3.2 defines the graph representation of the mooring line, and Section 3.3 specifies the static and dynamic features attached to its nodes and edges. Sections 3.4–3.6 derive the three components of the network, the GATv2 spatial encoder, the LSTM temporal encoder, and the prediction head. Section 3.7 describes the mixed-topology training strategy that lets one set of weights serve all discretisations. Section 3.8 defines the composite loss function, comprising the supervised data term, the evaluation metrics, the physically motivated auxiliary terms, and the multi-task uncertainty weighting. Finally, Section 3.9 describes the staged hyperparameter search and Section 3.10 reports the final configuration it selected.

3.1. Problem Formulation

A mooring line is discretised into N nodes, indexed from the anchor ($i = 0$) to the fairlead ($i = N - 1$). The number of nodes is not fixed. The model is trained and evaluated on ten discretisations, $N \in \{4, 5, 6, 7, 8, 10, 12, 15, 18, 21\}$, so that one set of weights serves every topology (Section 3.7). The line is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$ nodes and a bidirectional chain connectivity, giving $|\mathcal{E}| = 2(N - 1)$ directed edges. The graph carries two kinds of information:

1. **Static attributes:** these describe the time-invariant geometry and material of the line, through a static node-feature matrix $\mathbf{X}_s \in \mathbb{R}^{N \times 14}$ and a static edge-feature matrix $\mathbf{A}_s \in \mathbb{R}^{|\mathcal{E}| \times 10}$.
2. **Dynamic attributes:** these describe the evolving state of the line at a discrete time step τ , through a dynamic node-feature matrix $\mathbf{X}_\tau \in \mathbb{R}^{N \times 19}$ and a dynamic edge-feature matrix $\mathbf{A}_\tau \in \mathbb{R}^{|\mathcal{E}| \times 4}$.

The contents of all four matrices are defined in Section 3.3.

The model performs a sliding-window prediction. Let t denote the index of the current time step. The

input is the static graph together with a *history window* of the $H = 250$ most recent dynamic states,

$$\mathcal{H}_t = \left\{ (\mathbf{X}_\tau, \mathbf{A}_\tau) \mid \tau = t - H + 1, \dots, t \right\}, \quad (3.1)$$

and the output is the predicted state of the line over a *future horizon* of $F = 50$ steps. For each future step the model predicts three physical quantities at every node, namely the absolute horizontal position x^{abs} , the absolute vertical position z^{abs} , and the line tension T . Stacking these over the horizon and the nodes gives the target tensor

$$\mathbf{Y}_{t+1:t+F} \in \mathbb{R}^{F \times N \times 3}, \quad (\mathbf{Y}_\tau)_{i,:} = [x_{i,\tau}^{\text{abs}}, z_{i,\tau}^{\text{abs}}, T_{i,\tau}]. \quad (3.2)$$

The two position channels are absolute nodal coordinates in the global frame, taken directly from the finite-element output, rather than displacements relative to a reference configuration. The surrogate is the parametric map f_θ , with trainable parameters θ , that produces the entire horizon from a single evaluation.

$$\hat{\mathbf{Y}}_{t+1:t+F} = f_\theta(\mathcal{G}, \mathbf{X}_s, \mathbf{A}_s, \mathcal{H}_t) \in \mathbb{R}^{F \times N \times 3}. \quad (3.3)$$

The sampling interval of the underlying simulations is $\Delta t = 0.1$ s, so the history window spans 25 s of motion, capturing approximately five oscillation cycles of the mooring-line response as observed in the node-position and tension time series. The prediction horizon spans 5 s, approximately one such response cycle. Providing several response cycles of context lets the temporal encoder infer the phase and amplitude of the mooring-line dynamics, while restricting the horizon to approximately one cycle keeps the prediction well conditioned.

Direct multi-step prediction: The map in Equation (3.3) emits all F future steps in one forward pass, rather than predicting one step at a time and feeding each prediction back as input. This direct, non-autoregressive design is chosen for three reasons. First, it avoids the accumulation of errors that an autoregressive rollout incurs when its own imperfect predictions are reused as inputs, a compounding effect that is particularly damaging for tension, whose peaks the model is required to track accurately. Second, a horizon of $F = 50$ steps is short enough that a single-shot decoder remains well conditioned. Third, it is far cheaper at inference, since the horizon is produced by one network evaluation instead of fifty sequential ones, which is the primary motivation for building a surrogate in the first place. The trade-off, that the horizon length is fixed at training time, is acceptable for the present application and is revisited as a possible extension in Chapter 7. Figure 3.1 summarises the complete map of Equation (3.3), from the four input feature matrices to the predicted horizon, and locates each component developed in the sections that follow.

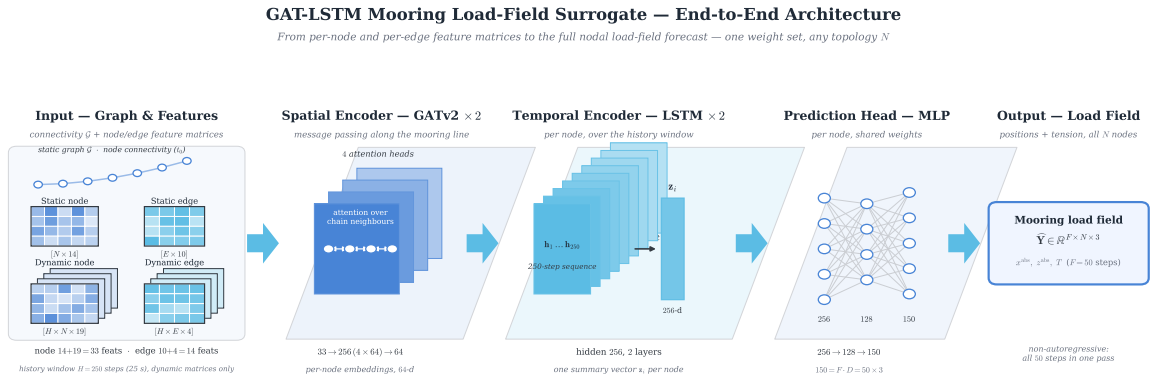


Figure 3.1: End-to-end architecture of the GAT+LSTM surrogate f_θ of Equation (3.3). The feature-matrix cell values shown are illustrative.

3.2. Graph Representation of the Mooring Line

The finite-element discretisation described in Chapter 2 partitions the mooring line into N nodes separated by $N - 1$ elastic segments. Each node accumulates half the mass and submerged weight of

its two neighbouring segments, with the anchor and fairlead endpoints receiving contributions from one segment only. The discretisation is therefore precisely a lumped-mass chain, and this structure maps directly onto a graph. Nodes of $\mathcal{G}$ represent the lumped masses, and edges represent the elastic segments that transmit axial forces between adjacent nodes.

Chain connectivity: The nodes are ordered from the anchor at $i = 0$ to the fairlead at $i = N - 1$ along the arc length of the line, following the same indexing used in the finite-element solver. Every physical segment between nodes i and $i + 1$ is represented by two directed edges, $i \rightarrow i + 1$ and $i + 1 \rightarrow i$, so that information can propagate in both directions along the chain during message passing. The full edge set is:

$$\mathcal{E} = \{ (i, i + 1), (i + 1, i) \mid i = 0, \dots, N - 2 \}, \quad |\mathcal{E}| = 2(N - 1). \quad (3.4)$$

No self-loops are included in the adjacency. Self-information is instead propagated via a residual skip connection in the spatial encoder, which is detailed in Section 3.4. Because the mooring line is a single catenary without branches, the graph is a path graph, in which every interior node has exactly two neighbours while the anchor and fairlead nodes each have one.

Physical interpretation of message passing: The local connectivity of the graph has a direct physical meaning. In the governing equations of motion (Chapter 2), the internal force on node i depends on the positions and velocities of the immediately adjacent nodes $i - 1$ and $i + 1$. Tension arises from the elongation of the segment between two neighbouring nodes, and the catenary geometry at any point along the line is determined by its local neighbourhood. The message-passing operation of a GNN mirrors this structure. At each layer, every node aggregates features from its direct neighbours, which corresponds to propagating the local force signal one step along the chain. With the two-layer GATv2 encoder used here, the effective receptive field extends to two hops, covering both the nearest and the next-nearest neighbours through which the dominant tension and curvature coupling of a mooring catenary acts.

Variable topology: The chain connectivity defined by Equation (3.4) holds identically for all ten discretisations $N \in \{4, 5, 6, 7, 8, 10, 12, 15, 18, 21\}$. Only the number of nodes and the length of each segment change. The connectivity pattern, the feature definitions (Section 3.3), and the model architecture are all independent of N . This is the structural property that enables a single set of model weights to generalise across all discretisations. The mechanism by which training exploits it is described in Section 3.7.

3.3. Feature Engineering

The predictive information available to the network is fixed by the four feature matrices introduced in Section 3.1. The static matrices $\mathbf{X}_s \in \mathbb{R}^{N \times 14}$ and $\mathbf{A}_s \in \mathbb{R}^{|\mathcal{E}| \times 10}$ are assembled once per case, and the dynamic matrices $\mathbf{X}_\tau \in \mathbb{R}^{N \times 19}$ and $\mathbf{A}_\tau \in \mathbb{R}^{|\mathcal{E}| \times 4}$ are evaluated at every history step τ . Their design follows two principles. First, the features are restricted to information that would realistically be available in a monitoring deployment, including the measured response of the line, the known geometric and material properties of the system, and summary parameters of the sea state. Second, known physics is made explicit wherever practical. Quantities that the network could only recover indirectly, such as the seabed contact state of each node, are precomputed and supplied as inputs. All features are local quantities attached to a single node or segment, so the same definitions apply unchanged to every discretisation N , as required for the variable-topology training of Section 3.7. At each history step the spatial encoder consumes the concatenation of static and dynamic features, giving $14 + 19 = 33$ inputs per node and $10 + 4 = 14$ inputs per edge (Section 3.4). This section defines the four feature groups in turn and closes with the normalisation applied before training.

3.3.1. Static Node Features

The static node features describe the time-invariant geometric, inertial, and material properties of each node. Because these properties do not change over time, they are extracted from only the first time record of each simulation file, which captures the unloaded static equilibrium configuration of the cable before any wave or current loading is applied (Section 2.3). Since every simulation file stores 21 nodes

regardless of the target discretisation, this equilibrium snapshot is then resampled to N nodes by the procedure of Section 3.7, giving a fixed static node-feature matrix $\mathbf{X}_s$ that is shared across all sliding windows drawn from the same (case, N) pair.

The first group of columns locates each node within the line topology, comprising the normalised arc-length coordinate s_i/L , which runs from 0 at the anchor to 1 at the fairlead, and three mutually exclusive indicator columns marking the node as anchor, fairlead, or interior. The line length L is obtained as the arc-length coordinate of the fairlead in the initial configuration. The indicators duplicate information already contained in s_i/L , but make the boundary distinction explicit by directly flagging which nodes carry the qualitatively different boundary conditions of the anchor and fairlead, rather than requiring the encoder to infer this from the continuous value of s_i/L . The initial coordinates x_i^0 and z_i^0 and the initial tension T_i^0 embed each node in the global frame and provide the static-equilibrium reference about which the dynamic response evolves.

The inertial columns follow the lumped-mass picture of Section 3.2. Because the resampled nodes are evenly spaced in arc length, every node is associated with a nominal segment length $\Delta s = L/(N - 1)$, and interior nodes receive the lumped mass and submerged weight

$$m_i = \rho A \Delta s, \quad w_i = (\rho A - \rho_{\text{water}} A) g \Delta s, \quad (3.5)$$

where $\rho A = 28.23 \text{ kg/m}$ is the mass of the chain per unit length, $A = 3.619 \times 10^{-3} \text{ m}^2$ its cross-sectional area, $\rho_{\text{water}} = 1025 \text{ kg/m}^3$ the water density, and $g = 9.81 \text{ m/s}^2$ the gravitational acceleration. The anchor and fairlead nodes receive half of both values, since their control volume covers a single half-segment (Section 3.2).

The remaining columns are constants shared by every node, the line diameter $d = 0.068 \text{ m}$, the cross-sectional area A , the normal and tangential drag coefficients $C_{d,n} = 2.6$ and $C_{d,t} = 1.4$, and the seabed elevation z_{bed} , taken as the anchor elevation in the initial configuration. The chain properties match those of the mooring system studied in the source dataset [35] and are identical for all ten locations. Only the line length, and with it the segment length and the lumped quantities of Equation (3.5), varies between locations and discretisations. Table 3.1 summarises all fourteen columns.

Table 3.1: The fourteen static node features, assembled from the initial static-equilibrium configuration of each case.

Col.	Feature	Units	Description
0	s_i/L	–	Normalised arc-length coordinate: 0 at the anchor, 1 at the fairlead
1	anchor flag	–	Indicator: 1 for the anchor node $i = 0$, else 0
2	fairlead flag	–	Indicator: 1 for the fairlead node $i = N - 1$, else 0
3	interior flag	–	Indicator: 1 for interior nodes $0 < i < N - 1$, else 0
4	x_i^0	m	Initial horizontal coordinate (static equilibrium)
5	z_i^0	m	Initial vertical coordinate (static equilibrium)
6	m_i	kg	Lumped nodal mass, Equation (3.5), halved at the end nodes
7	d	m	Line diameter, 0.068 m
8	A	m^2	Cross-sectional area, $3.619 \times 10^{-3} \text{ m}^2$
9	w_i	N	Lumped submerged weight, Equation (3.5), halved at the end nodes
10	z_{bed}	m	Seabed elevation, equal to the anchor's initial elevation
11	$C_{d,n}$	–	Normal drag coefficient, 2.6
12	$C_{d,t}$	–	Tangential drag coefficient, 1.4
13	T_i^0	N	Initial tension (static equilibrium)

3.3.2. Dynamic Node Features

The dynamic node features supply the state of the line at one history step τ . Stacked over all H steps of the window, they form the sequence of Equation (3.1) from which the temporal encoder infers the

motion. The nineteen columns split into six per-step quantities and thirteen per-case environmental descriptors (Table 3.2).

The first three columns are the response state of the node, namely the absolute positions $x_{i,\tau}^{\text{abs}}$ and $z_{i,\tau}^{\text{abs}}$ and the tension $T_{i,\tau}$, read from the stored 21-node trajectories and resampled to the working discretisation (Section 3.7). These coincide with the three target quantities of Equation (3.2). The network observes the past trajectory of exactly the quantities it is asked to extrapolate, conditioned on the geometry and the forcing descriptors.

Seabed interaction features: Columns 3–5 (Table 3.2) make the contact nonlinearity of the grounded line section explicit. After resampling, the binary contact flag and the penetration depth are evaluated per node and time step as:

$$c_{i,\tau} = \begin{cases} 1, & z_{i,\tau}^{\text{abs}} \leq z_{\text{bed}} + \varepsilon_c, \\ 0, & \text{otherwise,} \end{cases} \quad p_{i,\tau} = \max(z_{\text{bed}} - z_{i,\tau}^{\text{abs}}, 0), \quad (3.6)$$

with $\varepsilon_c = 10^{-6}$ m a numerical tolerance. The third column is an indicative vertical seabed reaction. For a node in contact it is the sum of the lumped submerged weight and an elastic spring term, and it is set to zero when the node is suspended,

$$r_{i,\tau} = c_{i,\tau} (w_i + k_b p_{i,\tau}), \quad (3.7)$$

where w_i is the lumped submerged weight of Equation (3.5) and k_b is the seabed spring stiffness of the finite-element contact law (Section 2.1.3). The reaction equals the submerged weight at incipient contact and grows linearly with penetration as the node presses into the seabed. All three quantities are deterministic functions of the vertical position and the static features, so the network could in principle recover them on its own. Supplying them directly, however, spares the encoder from having to learn the threshold behaviour at the touchdown point, which is among the strongest nonlinearities of catenary mooring dynamics.

Environmental descriptors: Columns 6–18 (Table 3.2) carry the thirteen sea-state variables defined in Section 2.2, namely the water depth h_0 , the significant wave height H_s , the peak wave period T_p , and the current profile given by the depth-velocity pairs $(d_1, v_1), \dots, (d_5, v_5)$. These values are constant within a case and are broadcast over every time step and node, which is the mechanism by which the sea-state conditioning reaches each graph snapshot processed by the spatial encoder. The instantaneous wave kinematics along the line are deliberately not part of the input. The wave forcing is represented only through its summary parameters, and the network must infer the present phase and severity of the excitation from the 25 s response history rather than observe it directly. This restriction matches the intended monitoring scenario, in which the line response is measured by sensors and the sea state is available as summary statistics from metocean forecasts or wave buoys, whereas distributed wave-particle velocities along the cable are not measurable in practice.

Table 3.2: The nineteen dynamic node features evaluated at every history step τ . Columns 0–5 vary per time step; columns 6–18 are the environmental descriptors of the case, constant over time and identical for every node.

Col.	Feature	Units	Description
0	$x_{i,\tau}^{\text{abs}}$	m	Horizontal nodal position (target quantity)
1	$z_{i,\tau}^{\text{abs}}$	m	Vertical nodal position (target quantity)
2	$T_{i,\tau}$	N	Line tension (target quantity)
3	$c_{i,\tau}$	–	Seabed contact indicator, Equation (3.6)
4	$p_{i,\tau}$	m	Seabed penetration depth, Equation (3.6)
5	$r_{i,\tau}$	N	Indicative vertical seabed reaction, Equation (3.7), zero for suspended nodes
6	h_0	m	Water depth at the location
7	H_s	m	Significant wave height of the case sea state
8	T_p	s	Peak wave period of the case sea state
9–18	$(d_k, v_k), k = 1, \dots, 5$	m, m/s	Current profile: alternating depth-velocity pairs

3.3.3. Static Edge Features

The static edge features characterise the elastic segment that an edge represents (Table 3.3). The geometric columns derive from the initial configuration, comprising the reference segment length ℓ_e , computed as the chord distance between the initial positions of the two endpoint nodes, its normalised counterpart ℓ_e/L , and the unit tangent components $(t_{x,e}, t_{z,e})$, which encode the initial orientation of the segment, near-horizontal in the grounded section of the line and progressively steeper towards the fairlead. The chord length coincides with the arc-length spacing Δ_s only where the line is straight, so the two carry complementary geometric information in the curved section.

Three material columns follow, namely the axial stiffness $EA = 232.7$ MN, which governs the elastic tension response of the segment, the mass per unit length ρA , and the line diameter d , all constants of the chain shared by every segment and location. Finally, three indicator columns mark the position of the segment within the chain, distinguishing the anchor-side segment, the fairlead-side segment, and interior segments. As established in Section 3.2, each physical segment is represented by two directed edges, and both carry the same ten-component feature vector, since every column is a symmetric property of the segment.

Table 3.3: The ten static edge features. Both directed edges of a physical segment carry the same feature vector.

Col.	Feature	Units	Description
0	ℓ_e	m	Reference segment length: chord distance between the initial endpoint positions
1	ℓ_e/L	–	Segment length normalised by the line length
2	EA	N	Axial stiffness, 232.7 MN
3	ρA	kg/m	Mass per unit length, 28.23 kg/m
4	d	m	Line diameter, 0.068 m
5	$t_{x,e}$	–	Horizontal component of the initial unit tangent
6	$t_{z,e}$	–	Vertical component of the initial unit tangent
7	anchor-side flag	–	Indicator: 1 for the first segment, else 0
8	interior flag	–	Indicator: 1 for interior segments, else 0
9	fairlead-side flag	–	Indicator: 1 for the last segment, else 0

3.3.4. Dynamic Edge Features

The dynamic edge features extend the seabed description from the nodes to the segments. For the edge pair representing the segment between nodes i and j , the four columns are evaluated at each history step τ as:

$$\bar{c}_e = \frac{1}{2}(c_i + c_j), \quad \bar{p}_e = \frac{1}{2}(p_i + p_j), \quad \bar{r}_e = \frac{1}{2}(r_i + r_j), \quad \chi_e = |c_i - c_j|, \quad (3.8)$$

where the time index is omitted for brevity. The averaged columns describe the contact state of the segment as a whole, with $\bar{c}_e$ equal to 0 for a fully suspended segment, 1 for a fully grounded one, and $\frac{1}{2}$ precisely when the segment straddles the touchdown point. The touchdown flag χ_e isolates this last case. Because the contact flags are binary, $\chi_e = 1$ exactly when the two endpoints are in different contact states, that is, for the segment on which the line lifts off the seabed. The touchdown point migrates along the line as the catenary responds to the fairlead motion, and the segment on which this transition occurs experiences the strongest change in boundary condition. Flagging it explicitly gives the attention mechanism a direct handle on this region. As with the static edge features, all four columns are symmetric in i and j , so the two directed edges of a segment carry identical values (Table 3.4). One candidate dynamic edge feature not included here is the current chord length $\ell_{e,\tau}$ between the two endpoint positions, which encodes segment elongation and hence strain directly on the edge. It was omitted because the GATv2 attention score already has access to both endpoint positions through the node features, so the encoder can derive elongation implicitly. Adding an explicit $\ell_{e,\tau}$ column, or equivalently the normalised elongation $(\ell_{e,\tau} - \ell_e)/\ell_e$, is a natural extension that could accelerate convergence in a future retrain.

Table 3.4: The four dynamic edge features, derived from the endpoint seabed features at every history step.

Col.	Feature	Units	Description
0	$\bar{c}_{e,\tau}$	–	Mean endpoint contact flag: 0 suspended, 1 grounded, $\frac{1}{2}$ straddling the touchdown point
1	$\bar{p}_{e,\tau}$	m	Mean endpoint penetration depth
2	$\bar{r}_{e,\tau}$	N	Mean endpoint seabed reaction
3	$\chi_{e,\tau}$	–	Touchdown indicator: 1 when the endpoint contact states differ

3.3.5. Normalisation

The feature columns span widely different physical scales. Tensions of order 10^4 – 10^5 N coexist with coordinates of order 10 – 10^2 m and dimensionless indicators. Left unscaled, the largest columns would dominate the input projections and condition the optimisation poorly. All continuous dynamic feature columns and all three target columns are therefore standardised to zero mean and unit variance,

$$\tilde{\phi}^{(k)} = \frac{\phi^{(k)} - \mu^{(k)}}{\sigma^{(k)}}, \quad (3.9)$$

where $\phi^{(k)}$ denotes the k -th feature column and $\mu^{(k)}$ and $\sigma^{(k)}$ are its mean and standard deviation estimated from the training split alone. The statistics are computed from a random subset of training windows, 5 000 windows in the final configuration (Section 3.10), pooled over all sampled windows, time steps, and nodes, and accumulated in a single pass with a numerically stable batched variant of Welford’s online algorithm [38]. The standard deviation is floored at 10^{-8} to avoid division by zero on near-constant columns. Fitting on the training split alone prevents information from the validation and test data from leaking into the input scaling. The fitted transform is then applied unchanged to all splits, on the fly as windows are loaded. For a summary of the resulting tension magnitude distribution across the dataset, see Table 2.2 (Section 2.5).

Two binary columns are excluded from the standardisation and passed through unchanged, namely the nodal contact flag $c_{i,\tau}$ and the touchdown flag $\chi_{e,\tau}$. Both are already bounded and dimensionless, and applying Equation (3.9) to a rarely active binary column is numerically fragile, since its standard deviation approaches zero and the rare activations would be mapped to arbitrarily large values.

The targets are standardised with their own column statistics, and the data term of the loss is computed in this normalised space, so that the three target quantities contribute on a comparable scale despite their different physical units. Predictions are mapped back to physical units by the inverse of Equation (3.9) wherever physical interpretability is required, namely for the physics-based loss terms and for all reported error metrics (Section 3.8).

The static feature matrices enter the encoder without standardisation. Most of their columns are material constants shared by every sample, which act as fixed offsets after the first affine projection of the encoder, and the remaining columns vary only between cases and discretisations, not along the time axis. Their absolute scale is absorbed by the input projection of the first attention layer together with the layer normalisation applied after every attention layer (Section 3.4). This layer normalisation acts on the hidden activations produced after each attention layer, not on the raw input features, and the attention implementation applies no further rescaling of its own to the input tensors. The z-score transform of Equation (3.9), together with the deliberately unstandardised static columns above, therefore constitutes the complete normalisation applied to the network’s inputs.

3.4. Spatial Encoder

The spatial encoder processes one snapshot of the mooring-line graph at a single history step τ and returns a fixed-dimensional embedding for each node. Each of the N nodes represents a single lumped-mass point along the line rather than the line as a whole (Section 3.2). The encoder receives the concatenated static and dynamic node features $\mathbf{h}_i^{\text{in}} \in \mathbb{R}^{33}$ and the concatenated static and dynamic edge features $\mathbf{e}_{ij} \in \mathbb{R}^{14}$, as defined in Section 3.3, and outputs node embeddings $\mathbf{h}_i^{\text{out}} \in \mathbb{R}^{64}$. Stacking

these embeddings over the $H = 250$ history steps produces the per-node input sequence consumed by the temporal encoder (Section 3.5). The encoder is a two-layer Graph Attention Network v2 (GATv2) [39], following the spatio-temporal graph-neural-network surrogate approach applied to floating wind turbine response prediction by Yung et al. [27, 28]. The first layer uses four attention heads whose outputs are concatenated to form a 256-dimensional intermediate representation. The second layer is a single-head attention that projects to the 64-dimensional output. A linear residual skip connection bypasses the graph aggregation, and layer normalisation together with an exponential linear unit (ELU) activation and dropout (rate 0.1) are applied after each layer. This attention operates spatially, over each node's graph neighbours at a fixed history step τ . It is distinct from the temporal, or sequence, attention used in Transformer architectures. The temporal dimension is handled entirely by the LSTM encoder of Section 3.5, which contains no attention mechanism of its own.

GATv2 attention mechanism: Graph Attention Networks [40] weight each neighbour's contribution to the aggregated node representation by a learned attention score. In the original formulation, the score e_{ij} assigned by the query node i to its neighbour j , along the message direction $j \rightarrow i$, is:

$$e_{ij}^{\text{GAT}} = \text{LeakyReLU}\left(\mathbf{a}^\top \left[\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j \right]\right) \quad (3.10)$$

where $\mathbf{W}$ is a shared linear projection, $\mathbf{a}$ is a learnable attention vector, and $\parallel$ denotes concatenation. Writing $\mathbf{a} = [\mathbf{a}_l \parallel \mathbf{a}_r]$ for the partition of $\mathbf{a}$ that acts on the two concatenated halves, the expression is separable, so the term $\mathbf{a}_r^\top \mathbf{W}\mathbf{h}_j$ alone determines the ranking of all neighbours j , independently of the query node i . Brody et al. [39] formalise this as *static* attention and show that it restricts the expressiveness of the attention head. GATv2 resolves the limitation by placing the nonlinearity inside the sum of the two transformed inputs:

$$e_{ij} = \mathbf{a}^\top \cdot \text{LeakyReLU}(\Theta_l \mathbf{h}_i + \Theta_r \mathbf{h}_j) \quad (3.11)$$

where Θ_l and Θ_r are separate learnable left and right projections. Equation (3.11) is not separable. The left projection $\Theta_l \mathbf{h}_i$ shifts the operating point of the nonlinearity through which $\Theta_r \mathbf{h}_j$ passes, making the attention *dynamic*, that is, query-dependent. For the mooring chain this matters because the relative importance of the anchor-side and fairlead-side neighbours changes as the touchdown point migrates along the line.

Edge-conditioned attention: The 14-dimensional edge features $\mathbf{e}_{ij}$ carry segment-level information including the contact fractions and the touchdown indicator that mark the most dynamically active region of the line (Section 3.3.4). They enter the attention score as a third additive term [41],

$$e_{ij} = \mathbf{a}^\top \cdot \text{LeakyReLU}(\Theta_l \mathbf{h}_i + \Theta_r \mathbf{h}_j + \Theta_e \mathbf{e}_{ij}) \quad (3.12)$$

where Θ_e is a learnable edge projection. The scores are normalised by a softmax over the in-neighbourhood of each node:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})} \quad (3.13)$$

and the aggregated node representation is the attention-weighted sum of the right-projected neighbour features:

$$\mathbf{h}_i^{\text{new}} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \Theta_r \mathbf{h}_j \quad (3.14)$$

The edge features enter the attention scores but not the aggregated values, so the attention mechanism uses segment properties to decide how much weight to assign to each neighbour while the transmitted information remains a projection of the neighbour's node state. Figure 3.2 illustrates the two stages of this operation for a single attention head.

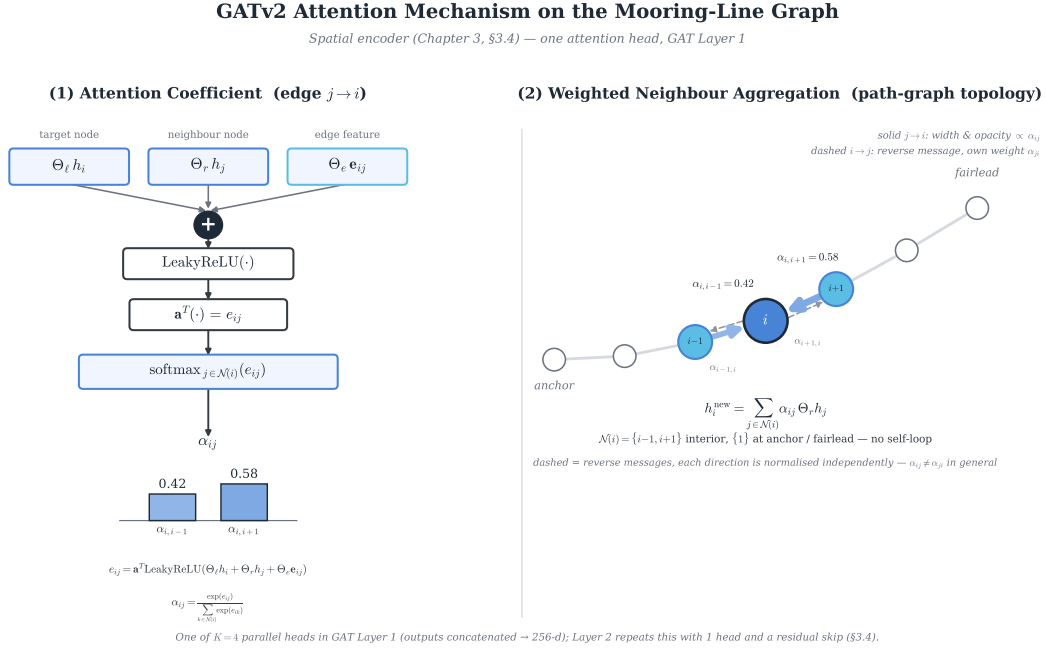


Figure 3.2: The GATv2 attention mechanism of Equations (3.12)–(3.14), shown for a single attention head. Illustrative weights shown, not measured from the trained model.

Multi-head attention and first layer: Layer 1 employs $K = 4$ independent attention heads, each applying Equations (3.12)–(3.14) with its own learnable parameters $\{\Theta_l^{(k)}, \Theta_r^{(k)}, \Theta_e^{(k)}, \mathbf{a}^{(k)}\}$ and mapping the 33-dimensional concatenated input to a 64-dimensional per-head embedding. The four outputs are concatenated:

$$\tilde{\mathbf{h}}_i^{(1)} = \left\| \sum_{k=1}^K \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(k)} \Theta_r^{(k)} \mathbf{h}_j^{\text{in}} \right\| \in \mathbb{R}^{256} \quad (3.15)$$

giving a 256-dimensional intermediate representation. Layer normalisation is then applied, followed by an ELU activation and dropout with rate 0.1.

Second layer and residual connection: Layer 2 is a single-head GATv2 convolution that projects the 256-dimensional intermediate representation to the 64-dimensional output,

$$\tilde{\mathbf{h}}_i^{(2)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \Theta_r \mathbf{h}_j^{(1)} \in \mathbb{R}^{64} \quad (3.16)$$

Both layers receive the same concatenated edge features $\mathbf{e}_{ij}$ for their attention scores. The edge representation is not updated between layers. Layer normalisation is applied to $\tilde{\mathbf{h}}_i^{(2)}$. As established in Section 3.2, no self-loops are present in the chain graph, so the aggregations in Equations (3.15)–(3.16) sum only over the one or two physical neighbours of each node. A linear skip connection projecting the original 33-dimensional concatenated input then restores the node's own information,

$$\mathbf{h}_i^{\text{out}} = \text{ELU}(\tilde{\mathbf{h}}_i^{(2)} + \mathbf{W}_{\text{res}} \mathbf{h}_i^{\text{in}}) \in \mathbb{R}^{64} \quad (3.17)$$

where $\mathbf{W}_{\text{res}} \in \mathbb{R}^{64 \times 33}$ is a learnable weight matrix. Dropout with rate 0.1 is applied to $\mathbf{h}_i^{\text{out}}$. The residual ensures that the identity map is always reachable, which stabilises training when the attention weights are still uninformative in the early epochs.

Vectorised forward pass: A training batch of $B = 32$ windows, each spanning $H = 250$ history steps, presents $B \times H = 8\,000$ graph snapshots to the spatial encoder per training step. Processing these sequentially would require 8 000 separate GATv2Conv kernel calls on small N -node graphs, leaving the

GPU idle between launches. The bucketed batch sampler (Section 3.7) guarantees that all B windows in a batch share the same node count N , so the $B \times H$ graphs are topologically identical. A vectorised forward method assembles them into a single block-diagonal super-graph with $B \times H \times N$ nodes and $B \times H \times |\mathcal{E}|$ edges, where the node indices of the m -th sub-graph are offset by $m \times N$, preventing any cross-graph message passing. A single GATv2Conv call then processes all nodes simultaneously, fully utilising the GPU tensor cores.

3.5. Temporal Encoder

The temporal encoder receives the sequence of per-node spatial embeddings produced by the spatial encoder at each of the $H = 250$ history steps and condenses it into a single fixed-size representation per node. For every history step $\tau \in \{t - H + 1, \dots, t\}$ the spatial encoder returns a 64-dimensional embedding $\mathbf{h}_i^{\text{out}}(\tau) \in \mathbb{R}^{64}$ for each node i (Section 3.4). Stacking these over the window gives a sequence $\mathbf{H}_i \in \mathbb{R}^{H \times 64}$ per node, and the temporal encoder maps this sequence to a single 256-dimensional summary $\mathbf{z}_i \in \mathbb{R}^{256}$ that is passed to the prediction head (Section 3.6). A Long Short-Term Memory network [42] is used as the temporal encoder because its gating mechanism can selectively retain information across the full 250-step history window, which spans approximately five oscillation cycles of the mooring response and therefore contains both the slow drift of the catenary shape and the faster wave-frequency oscillations.

LSTM gate equations: An LSTM cell processes one input step $\mathbf{x}_t \in \mathbb{R}^{d_{\text{in}}}$ together with the previous hidden state $\mathbf{h}_{t-1} \in \mathbb{R}^{d_h}$ and cell state $\mathbf{c}_{t-1} \in \mathbb{R}^{d_h}$, where d_h is the hidden dimension. The update proceeds through four gates [42, 43]:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \quad (3.18)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \quad (3.19)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c), \quad (3.20)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \quad (3.21)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (3.22)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \quad (3.23)$$

where σ is the sigmoid function and $\odot$ the element-wise (Hadamard) product. The weight matrices $\mathbf{W}_{(\cdot)} \in \mathbb{R}^{d_h \times d_{\text{in}}}$ and $\mathbf{U}_{(\cdot)} \in \mathbb{R}^{d_h \times d_h}$ and the bias vectors $\mathbf{b}_{(\cdot)} \in \mathbb{R}^{d_h}$ are all learnable. The forget gate $\mathbf{f}_t$ determines which part of the cell state is discarded, the input gate $\mathbf{i}_t$ and candidate cell $\tilde{\mathbf{c}}_t$ determine what new information is written, and the output gate $\mathbf{o}_t$ selects which part of the updated cell state flows into the hidden state $\mathbf{h}_t$. The cell state $\mathbf{c}_t$ is updated additively in Equation (3.21), which provides a near-unobstructed gradient path through time and allows the network to retain relevant context over sequences far longer than is feasible for plain recurrent networks [42].

Node-wise temporal processing: At each history step τ , the spatial encoder returns an embedding matrix of shape $[N, 64]$ containing one vector per node. These are assembled into the per-node input sequence

$$\mathbf{H}_i = (\mathbf{h}_i^{\text{out}}(\tau))_{\tau=t-H+1}^t \in \mathbb{R}^{H \times 64}$$

Each node's sequence is processed by the LSTM independently. Because the spatial encoder has already propagated inter-node information through message passing (Section 3.4), the 64-dimensional embedding at each step already encodes the neighbourhood context of node i . The temporal encoder therefore does not need to observe other nodes simultaneously. This node-wise design separates spatial and temporal responsibilities cleanly. The GATv2 encoder handles *what* the neighbourhood looks like at each instant, while the LSTM encoder handles *how* each node's local state evolves over time. In practice, the N independent sequences are stacked into a tensor of shape $[N, H, 64]$ and processed in a single LSTM call, equivalent to a batch of N independent sequences [43].

Two-layer configuration and readout: The final model uses two stacked LSTM layers with hidden dimension $d_h = 256$, selected by the Stage B hyperparameter search (Section 3.9). The output sequence of the first layer is passed as input to the second layer, with dropout of rate 0.1 applied between

layers. The temporal embedding for each node is taken as the output of the second layer at the last step of the history window:

$$\mathbf{z}_i = \mathbf{h}_t^{(2)} \in \mathbb{R}^{256} \quad (3.24)$$

where the superscript denotes the second LSTM layer and t is the index of the most recent history step. Layer normalisation followed by dropout with rate 0.1 is applied to $\mathbf{z}_i$ before it enters the prediction head. Taking the last-step output rather than a mean-pool over the sequence preserves temporal order, so the embedding encodes the most recent state of the line, which is the natural context for predicting the near-future horizon. The LSTM is unidirectional, so each hidden state is computed from past inputs only, maintaining causal consistency between the observation window and the prediction horizon. Figure 3.3 shows the cell mechanics of Equations (3.18)–(3.23) together with the unrolled processing of the history window.

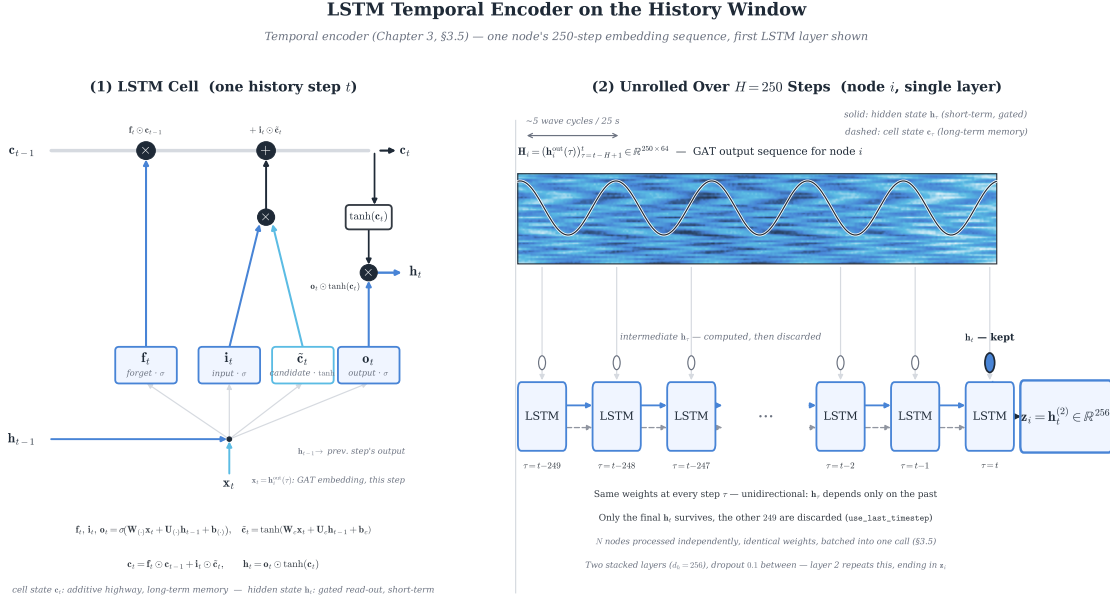


Figure 3.3: The LSTM temporal encoder of Equations (3.18)–(3.24), shown for a single node i . The input sequence shown is illustrative, not extracted from the trained model.

3.6. Prediction Head

The prediction head receives the per-node embedding $\mathbf{z}_i \in \mathbb{R}^{256}$ produced by the temporal encoder (Section 3.5) and maps it to the predicted future trajectory of node i over the full $F = 50$ step horizon. The head is a two-layer multilayer perceptron with a hidden width of 128,

$$\hat{\mathbf{y}}_i = \mathbf{W}_2 \cdot \text{ELU}(\mathbf{W}_1 \mathbf{z}_i + \mathbf{b}_1) + \mathbf{b}_2 \in \mathbb{R}^{F \cdot D} \quad (3.25)$$

where $\mathbf{W}_1 \in \mathbb{R}^{128 \times 256}$ and $\mathbf{W}_2 \in \mathbb{R}^{150 \times 128}$ are learnable weight matrices, $\mathbf{b}_1 \in \mathbb{R}^{128}$ and $\mathbf{b}_2 \in \mathbb{R}^{150}$ are biases, $F = 50$, and $D = 3$ is the number of targets. A dropout layer with rate 0.1 is applied after the ELU activation and before the second linear projection. The 150-dimensional output is reshaped to $[F, D] = [50, 3]$ per node, and stacking over all N nodes gives the full prediction tensor $\hat{\mathbf{Y}} \in \mathbb{R}^{F \times N \times D}$ matching the form of Equation (3.2).

Non-autoregressive direct prediction: Equation (3.25) produces all $F = 50$ future steps in a single evaluation of the two-layer mapping, without feeding any predicted step back as input. This directly implements the non-autoregressive design stated in Section 3.1. The temporal embedding $\mathbf{z}_i$ encodes a compressed summary of the 250-step history for node i , and the head decodes it into the complete future trajectory in one pass. The 128-dimensional hidden layer was found sufficient in the Stage B search (Section 3.9) to map the 256-dimensional temporal representation onto the 150 scalar outputs, and widening it was not among the changes that improved accuracy.

Output space: The prediction head operates in the normalised target space of Section 3.3.5. Predicted values in $\hat{\mathbf{Y}}$ are in the standardised frame of Equation (3.9) and are mapped back to physical units by the inverse transform wherever physical interpretation is required, specifically in the physically motivated auxiliary loss terms of Section 3.8 and in all reported accuracy metrics. Figure 3.4 shows the structure of the head and the reshaping of its output into the prediction tensor.

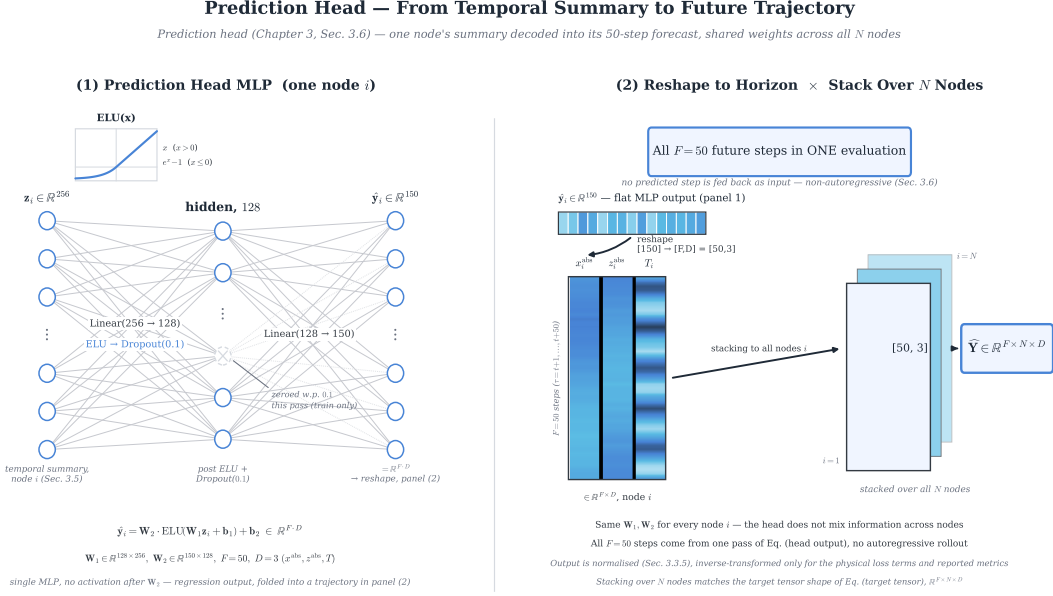


Figure 3.4: The prediction head of Equation (3.25), shown for a single node i .

3.7. Mixed-Topology Training

The finite-element simulator stores all output trajectories at $N_{\text{orig}} = 21$ nodes per mooring line, while the surrogate is required to predict at any of the ten discretisations $N \in \{4, 5, 6, 7, 8, 10, 12, 15, 18, 21\}$ with a single set of weights. Two mechanisms enable this. A spatial resampling function converts a stored 21-node trajectory to the working discretisation before a sliding window is assembled, so that training samples at any target N are available without maintaining separate caches for each topology. A bucketed batch sampler then ensures that every training batch is drawn from a single node count, a prerequisite for the block-diagonal vectorisation of the spatial encoder (Section 3.4).

Arc-length resampling: Let $s_j = j / (N_{\text{orig}} - 1)$, $j = 0, \dots, N_{\text{orig}} - 1$, be the normalised arc-length positions of the 21 stored nodes, and $s'_i = i / (N - 1)$, $i = 0, \dots, N - 1$, the uniformly spaced target positions for a discretisation of N nodes. Both grids span $[0, 1]$, so the anchor at $s = 0$ and the fairlead at $s = 1$ are endpoints of both grids and are always preserved exactly. For each interior target position s'_i , the enclosing source interval $[s_j, s_{j+1}]$ is located by binary search, and the signal value is estimated by linear interpolation along the arc:

$$f(s'_i) = f(s_j) + \frac{s'_i - s_j}{s_{j+1} - s_j} (f(s_{j+1}) - f(s_j)) \quad (3.26)$$

Equation (3.26) is applied to the initial equilibrium snapshot (the first stored time step of each case) to produce the initial-position and initial-tension entries of the static node-feature matrix $\mathbf{X}_s$ (Section 3.3.1), and independently to the x^{abs} , z^{abs} , and T channels at every time step of the sliding window to produce the dynamic node features and the target tensor $\mathbf{Y}$ of Equation (3.2). All time steps are processed simultaneously through vectorised array operations over the time dimension. When $N = N_{\text{orig}} = 21$, the resampling step is bypassed and the original FE data is passed through unchanged.

Bucketed batch sampling: The block-diagonal super-graph construction described in Section 3.4 requires that all $B = 32$ samples in a training batch share the same node count. A bucketed node-count batch sampler enforces this constraint by partitioning the training window index list into buckets by node count at initialisation. At the start of each training epoch, the sampler draws a fresh subsample from the full window pool using a random number generator seeded with the sum of a fixed base seed and the epoch index, re-buckets the subsample by node count, shuffles the indices within each bucket, assembles batches of size B from each bucket in turn, and finally shuffles the resulting batch list. Each batch therefore contains only topologically identical graphs, while the shuffled ordering across batches exposes the model to all ten discretisations within every epoch.

Topology-agnostic weight sharing: Because the GATv2 message-passing operations act only on the one- or two-hop local neighbourhood of each node, and because all feature definitions in Section 3.3 are local quantities attached to individual nodes or segments, the model contains no parameters that depend on N . Training jointly on all ten discretisations exposes the weights to a broader distribution of catenary shapes, segment lengths, and seabed contact patterns than would be available from a single fixed topology. The Stage B hyperparameter search (Section 3.9) confirmed that the resulting model achieves a validation coefficient of determination $R^2_{\text{tension}} > 0.96$ across all ten node counts, with accuracy declining only modestly as N increases from four to twenty-one.

3.8. Loss Function and Evaluation Metrics

The training objective combines a primary data-fidelity loss with a set of physically motivated auxiliary terms. These terms encode domain knowledge about what a physically plausible and safe prediction should satisfy, accurate peak loads, conservative rather than unsafe underprediction, and smooth step-to-step dynamics, but they are evaluated on the model's own output against the ground truth, not on a residual of the governing partial differential equation of Section 2.1. They are therefore distinct from physics-informed neural networks in the stricter sense, where a PDE residual is evaluated directly at collocation points and enters the loss as a constraint on the differential operator itself. This section first defines the supervised data loss (Section 3.8.1), then the evaluation metrics used to assess model quality and to drive the Stage B configuration search (Section 3.8.2), and finally the physical loss terms in their two forms, the fixed-weight formulation used throughout Stage B (Section 3.8.3) and the learnable uncertainty-weighted formulation reserved for one of the final Stage C models (Section 3.8.4).

3.8.1. Data Loss

The data loss compares the predicted and ground-truth trajectories directly in the normalised target space of Section 3.3.5. Let $\mathbf{r} = \hat{\mathbf{Y}} - \mathbf{Y}$ denote the residual between the predicted tensor and the target tensor of Equation (3.2), both standardised, and let r_k index its individual scalar entries over the $F \times N \times D$ elements of one window. The smooth L_1 (Huber) loss [44] is applied element-wise:

$$\ell(r_k) = \begin{cases} \frac{1}{2\beta} r_k^2, & |r_k| < \beta, \\ |r_k| - \frac{1}{2}\beta, & |r_k| \geq \beta, \end{cases} \quad \beta = 1, \quad (3.27)$$

and averaged over all elements of the window to give the data loss

$$\mathcal{L}_{\text{data}} = \frac{1}{FND} \sum_k \ell(r_k). \quad (3.28)$$

The smooth L_1 loss behaves quadratically for residuals smaller than the threshold β and linearly beyond it. This combines the stable, well-conditioned gradients of a squared-error loss near the optimum with the reduced outlier sensitivity of an absolute-error loss for large residuals, which is important because the tension channel contains occasional sharp snap-load peaks that a pure mean-squared-error loss would over-weight. The threshold is kept at the default $\beta = 1$, which in the standardised frame corresponds to one standard deviation of the per-target training distribution.

3.8.2. Evaluation Metrics

All evaluation metrics are computed in physical units, obtained by applying the inverse of the z-score transform of Equation (3.9) to both the prediction and the target before the metric is evaluated. The three target channels are the horizontal position x^{abs} in metres, the vertical position z^{abs} in metres, and the tension T in newtons. Each metric is accumulated over every validation window and aggregated across the full validation set. The shorthand $\mathbf{e}_{t,i} = \hat{y}_{t,i} - y_{t,i}$ denotes the physical-unit error at forecast step t and node i for a given target channel.

Regression accuracy: The mean absolute error and root-mean-square error per target channel are

$$\text{MAE} = \frac{1}{FN} \sum_{t,i} |\mathbf{e}_{t,i}| \quad \text{RMSE} = \sqrt{\frac{1}{FN} \sum_{t,i} \mathbf{e}_{t,i}^2} \quad (3.29)$$

averaged over all $F = 50$ forecast steps and N nodes within a window and then across windows. The root-mean-square error is the more outlier-sensitive of the two. The principal quality metric is the coefficient of determination, for which two variants are tracked. The per-window variant computes, for each window, $R_w^2 = 1 - \sum_{t,i} \mathbf{e}_{t,i}^2 / \sum_{t,i} (y_{t,i} - \bar{y}_w)^2$ against that window's own mean $\bar{y}_w$ and then averages over windows. Because each window is referenced to its local mean, this variant is blind to a systematic offset between windows representing different sea states and is therefore used only as a within-window waveform diagnostic. The global variant accumulates the residual and total sums of squares across all validation windows before forming a single ratio:

$$R_{\text{global}}^2 = 1 - \frac{\sum_{\text{windows}} \sum_{t,i} \mathbf{e}_{t,i}^2}{\sum_{\text{windows}} \sum_{t,i} (y_{t,i} - \bar{y}_{\text{global}})^2} \quad (3.30)$$

where $\bar{y}_{\text{global}}$ is the mean over all windows, forecast steps, and nodes. Equation (3.30) is the standard definition of the coefficient of determination and is sensitive to both the within-window wave-frequency oscillation and the between-window variation in mean load level across sea states. It is therefore the more stringent and physically meaningful of the two, and R_{global}^2 for the tension channel is the primary physical metric for model selection, alongside validation loss, throughout the Stage B search (Section 3.9).

Peak-tension safety metrics: For each window the per-node peak tension is extracted over the forecast horizon, $T_i^{\text{peak}} = \max_t T_{t,i}$ for the ground truth and $\hat{T}_i^{\text{peak}} = \max_t \hat{T}_{t,i}$ for the prediction, and the resulting per-node errors are accumulated across all windows. These metrics target the design-critical question of whether the surrogate reproduces extreme loads, summarised in Table 3.5. The most important is the underprediction rate, which is the fraction of nodes whose peak tension is underestimated. An underestimated peak load is unconservative and potentially unsafe, so this quantity is the primary safety metric and the direct motivation for the asymmetric loss term introduced in Section 3.8.3.

Table 3.5: Peak-tension safety metrics, evaluated on the per-node peak tension over the $F = 50$ step horizon and aggregated across all validation windows.

Metric	Definition	Engineering meaning
Peak MAE	$\frac{1}{N} \sum_i \hat{T}_i^{\text{peak}} - T_i^{\text{peak}} $	Mean peak-load error [N]
Peak bias	$\frac{1}{N} \sum_i (\hat{T}_i^{\text{peak}} - T_i^{\text{peak}})$	Signed: > 0 safe, < 0 unsafe [N]
Underprediction rate	$\frac{100\%}{N} \sum_i \mathbf{1}[\hat{T}_i^{\text{peak}} < T_i^{\text{peak}}]$	Primary safety metric [%]
Peak p_{95} error	95th percentile of $ \hat{T}_i^{\text{peak}} - T_i^{\text{peak}} $	Worst-typical case [N]

Temporal accuracy: Two metrics characterise the temporal structure of the forecast. The horizon-resolved error reports the mean absolute error restricted to a single forecast step index, evaluated

at steps 1, 10, 25, and 50, corresponding to lookahead times of 0.1, 1.0, 2.5, and 5.0 s. A monotonic increase of the horizon error with lookahead is physically expected and confirms that the non-autoregressive decoder degrades gracefully towards the end of the horizon rather than failing abruptly. The temporal-difference error measures the accuracy of the predicted rate of change:

$$\text{TempDiff-MAE} = \frac{1}{(F-1)N} \sum_{t=1}^{F-1} \sum_i \left| (\hat{y}_{t+1,i} - \hat{y}_{t,i}) - (y_{t+1,i} - y_{t,i}) \right| \quad (3.31)$$

in units of metres or newtons per step, where one step is 0.1 s. It quantifies whether the model reproduces physically smooth dynamics consistent with the wave-induced loading, and it is the metric directly addressed by the temporal-smoothness loss term of Section 3.8.3.

Threshold-gated MAPE: For the tension channel a mean absolute percentage error is reported, restricted to time-and-node pairs whose true tension exceeds a threshold $\varepsilon = 1000$ N:

$$\text{MAPE}_T = \frac{100\%}{n_{\text{valid}}} \sum_{\{t,i: |T_{t,i}| > \varepsilon\}} \frac{|\hat{T}_{t,i} - T_{t,i}|}{|T_{t,i}|} \quad (3.32)$$

The threshold excludes near-zero tension values arising from slack or nearly slack cable, which would otherwise produce division-by-near-zero artefacts. The accompanying valid fraction, the proportion of pairs that pass the threshold, indicates how informative the percentage error is. A low valid fraction signals substantial slack-cable behaviour in the validation set. All metrics defined above are additionally broken down per node count using the per-topology metric log. In the Stage B search of Section 3.9, this breakdown shows the validation R_{global}^2 for tension decreasing near-monotonically from approximately 0.977 at $N = 4$ to 0.964 at $N = 21$, indicating that finer spatial discretisation increases prediction difficulty.

3.8.3. Physical Loss Terms

In addition to the data loss of Equation (3.28), four auxiliary terms, each motivated by a physical requirement, regularise the model towards behaviour consistent with mooring mechanics. Each term is evaluated in physical units, after inverse transformation of the prediction and target, and then made dimensionless by dividing by a characteristic scale taken as the per-target standard deviation of the training distribution, σ_x , σ_z , and σ_T for the three channels. Writing the physical-unit prediction and target as $\hat{y}$ and y , the four terms are:

$$\mathcal{L}_{\text{mae}} = \frac{1}{FND} \sum_{t,i,d} \frac{|\hat{y}_{t,i,d} - y_{t,i,d}|}{\sigma_d} \quad (3.33)$$

$$\mathcal{L}_{\text{peak}} = \frac{1}{N \sigma_T} \sum_i |\hat{T}_i^{\text{peak}} - T_i^{\text{peak}}| \quad (3.34)$$

$$\mathcal{L}_{\text{under}} = \frac{1}{N \sigma_T} \sum_i \max(0, T_i^{\text{peak}} - \hat{T}_i^{\text{peak}}) \quad (3.35)$$

$$\mathcal{L}_{\text{temp}} = \frac{1}{(F-1)ND} \sum_{t,i,d} \frac{|(\hat{y}_{t+1,i,d} - \hat{y}_{t,i,d}) - (y_{t+1,i,d} - y_{t,i,d})|}{\sigma_d} \quad (3.36)$$

The scaled mean absolute error in Equation (3.33) reinforces accuracy in physical units across all three channels. The symmetric peak term of Equation (3.34) penalises any error in the peak load, while the asymmetric term of Equation (3.35), built from a rectified-linear penalty, activates only when the predicted peak falls below the true peak and therefore directly discourages the unsafe underprediction tracked by the primary safety metric. The temporal-smoothness term of Equation (3.36) penalises errors in the step-to-step increments, encouraging physically realistic dynamics. The complete Stage B objective is the sum of the data loss and the four weighted physical terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + w_{\text{mae}} \mathcal{L}_{\text{mae}} + w_{\text{peak}} \mathcal{L}_{\text{peak}} + w_{\text{under}} \mathcal{L}_{\text{under}} + w_{\text{temp}} \mathcal{L}_{\text{temp}} \quad (3.37)$$

with the fixed weights $w_{\text{mae}} = 0.0025$, $w_{\text{peak}} = 0.005$, $w_{\text{under}} = 0.02$, and $w_{\text{temp}} = 0.0025$ for the final configuration. These values were established by the Stage B search (Section 3.9). In particular, the underprediction weight is held at the comparatively large value of 0.02 because reducing it was found to worsen the peak-tension underprediction rate, a safety regression. The weights are kept deliberately small so that the auxiliary terms act as a gentle regulariser. Under the final configuration the physical terms contribute roughly 21–23 % of the data-loss magnitude (Section 3.9), rather than dominating the gradient as they did at the baseline weights.

3.8.4. Multi-Task Uncertainty Weighting

The fixed weights of Equation (3.37) require manual calibration and impose a single static balance between the competing objectives for the entire training run. One of the final Stage C models replaces this fixed weighting with the homoscedastic-uncertainty formulation of Kendall et al. [45], in which the relative weight of each auxiliary term is learned jointly with the network parameters. Each of the four physical terms $\mathcal{L}_k \in \{\mathcal{L}_{\text{mae}}, \mathcal{L}_{\text{peak}}, \mathcal{L}_{\text{under}}, \mathcal{L}_{\text{temp}}\}$ is assigned a learnable log-variance parameter $s_k = \log \sigma_k^2$, where the index k runs over loss terms rather than over nodes or target channels, and the weighted objective becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + \sum_{k=1}^4 \left[\frac{1}{2 \exp(s_k)} \mathcal{L}_k + \frac{1}{2} s_k \right] \quad (3.38)$$

where the four parameters s_k are initialised to zero, recovering unit weights at the start of training. The first term in the bracket downweights a loss component with large learned uncertainty σ_k^2 , while the second term acts as a regulariser that prevents the trivial solution of driving every σ_k to infinity. Optimising the s_k alongside the weights lets the model discover the appropriate balance automatically and adapt it over the course of training rather than fixing it in advance.

Differentiating the bracket in Equation (3.38) with respect to s_k and setting the result to zero gives the stationary point

$$-\frac{\mathcal{L}_k}{2 \exp(s_k)} + \frac{1}{2} = 0 \quad \implies \quad s_k^* = \ln \mathcal{L}_k, \quad (3.39)$$

so at equilibrium each learned log-variance tracks the logarithm of its own term. Equation (3.39) is the condition against which the convergence of each σ_k is assessed in Chapter 4, and it shows immediately that a term able to reach zero possesses no stationary point in s_k .

The learned variances are themselves an interpretable result. A larger σ_k^2 identifies a loss term that the model fits less precisely relative to its own scale, with the expectation that the peak-tension terms, which depend on the sharpest dynamics of the response, will be assigned greater uncertainty than the temporal-smoothness term, which measures a short-range difference. The data loss of Equation (3.28) retains unit weight as the primary supervised objective, so the uncertainty weighting governs only the relative influence of the four physical regularisers.

3.9. Stage B Hyperparameter Optimisation

The Stage B search was conducted as a controlled, one-at-a-time ablation study on the mixed-topology dataset with 50 training cases per location. Between runs, only the training configuration was modified. All model-architecture, data-loading, and training-function components were kept identical, so that each comparison isolated the effect of a single configuration change. Every run used 40 training epochs, an initial learning rate of 10^{-3} , and the AdamW optimiser [46] with decoupled weight decay $\lambda = 10^{-4}$.

Runs were accepted, rejected, or partially accepted on a two-level criterion. A change was accepted outright if $R_{\text{global}, \text{tension}}^2$ improved by more than 0.0015 at the best validation checkpoint. Because that margin exceeds the seed-to-seed scatter reported in Table 3.7, a smaller movement in $R_{\text{global}, \text{tension}}^2$ was treated as indistinguishable from noise, and the decision then fell to the secondary metrics, namely the peak-tension underprediction rate as the safety criterion and the position errors MAE_x and R_x^2 as the accuracy criterion. A change was accepted on this second level only if it improved a secondary metric substantially while leaving $R_{\text{global}, \text{tension}}^2$ within that same noise band, and it was rejected if any secondary metric regressed materially. Runs A1 and BBF were accepted on the second level, and run D1 was accepted only in part, for the reasons given below.

B0 baseline: The baseline used a single-layer LSTM with hidden dimension $d_{\text{LSTM}} = 128$, dropout $p = 0.1$ throughout, and the initial physical loss weights of Equation (3.37). After 40 epochs it achieved $R_{\text{global,tension}}^2 = 0.9755$ and a peak underprediction rate of 29.8%. Two diagnostic ratios motivated the subsequent experiments. The validation-to-training loss ratio reached 1.36–1.46 over the final five epochs, indicating mild overfitting, and the ratio of physical to data loss was approximately 44%, indicating that the physical terms dominated the gradient signal. These diagnostics triggered three parallel follow-up runs.

First-round experiments: Run A1 doubled the LSTM hidden dimension to 256, widening the temporal encoder’s capacity to represent multi-scale wave-induced dynamics. The increase reduced the horizontal-position error by 40% ($\text{MAE}_x = 1.36$ m versus 2.29 m for B0), raised R_x^2 from 0.979 to 0.993, reduced the underprediction rate from 29.8% to 27.1%, and largely closed the overfitting gap (val/train ratio 0.85–1.30). Its $R_{\text{global,tension}}^2$ moved by -0.0019 , within the noise band defined above, so run A1 was accepted on the second level, on the strength of the large position-accuracy gain and the improved safety metric.

Run BC1 raised all dropout probabilities to 0.2 to address the overfitting signal. The intervention proved too aggressive. $R_{\text{global,tension}}^2$ fell to 0.9662, the underprediction rate worsened to 41.7%, and early stopping triggered at epoch 35. The result indicates that the model relies on dense local-neighbourhood attention to propagate load information along the chain, and that regularising those connections is counterproductive. Run BC1 was rejected.

Run D1 halved all four physical loss weights to reduce the dominant phys/data ratio. The data loss recovered gradient dominance (phys/data ratio fell to 21–23%), R_x^2 improved to 0.993, and the overfitting gap closed (val/train ratio 0.95–1.13). However, reducing w_{under} from 0.020 to 0.010 worsened the underprediction rate from 29.8% to 38.5% across all node counts, a direct safety regression. Run D1 was therefore only partially accepted, in that w_{mae} , w_{peak} , and w_{temp} were halved while $w_{\text{under}} = 0.020$ was restored to its B0 value.

Combined configuration and extended experiments: The BBF run combined the A1 and partial D1 decisions. The outcome was substantially stronger than either parent. The underprediction rate fell to approximately 19% at the best checkpoint, versus 27% for A1 and 39% for D1 alone. The synergy reflects two mutually reinforcing effects, the wider LSTM enabling the model to track the build-up of peak loads over the history window, and the recalibrated weight balance giving the asymmetric underprediction penalty clear gradient precedence. BBF became the intermediate best architecture.

Alongside the three runs above, the first round also tested two capacity changes that were rejected at the time. Run A2 tested a second LSTM layer while keeping the B0 loss weights. Under that imbalanced loss the underprediction rate worsened to 32.8%, establishing that the second layer requires recalibrated weights to be beneficial. Run B1 tested a wider GAT output dimension ($d_{\text{GAT,out}} = 128$) while retaining $d_{\text{LSTM}} = 128$. The mismatched spatial-to-temporal capacity ratio produced unstable training, with $R_{\text{global,tension}}^2$ oscillating between 0.963 and 0.974 across epochs. Both were therefore revisited once BBF had fixed the loss balance.

Run BE1 retested the second LSTM layer within the BBF configuration ($d_{\text{LSTM}} = 256$, recalibrated weights). With the capacity mismatch resolved and the loss balance corrected, the additional depth produced the highest $R_{\text{global,tension}}^2 = 0.9783$ and $\text{MAE}_x = 1.05$ m across all runs, an improvement of $+0.0028$ over B0 that clears the primary acceptance threshold outright. Two further runs, BE2 ($d_{\text{GAT,out}} = 128$, $d_{\text{LSTM}} = 256$, B0 weights) and BE3 (BBF weights with $d_{\text{GAT,out}} = 128$), confirmed that widening the GAT output without recalibrated weights produces a competitive but late-epoch overfitting configuration (BE2), while adding a wider GAT on top of the already-recalibrated BBF introduces additional overfitting without improving $R_{\text{global,tension}}^2$ (BE3). Table 3.6 summarises the six runs that defined the accepted configuration path. Runs A2, B1, BE2, and BE3 are omitted from the table, since each was rejected on the qualitative grounds described above rather than on a competitive final metric.

Table 3.6: Stage B ablation summary at the best validation checkpoint per run, all with seed 42. Underprediction rates are averaged across all ten node counts. Bold denotes the selected configuration.

Run	Key change	R^2_{tension}	MAE_x [m]	Underpred [%]	Verdict
B0	Baseline	0.9755	2.29	29.8	Baseline
A1	$d_{\text{LSTM}} = 256$	0.9736	1.36	27.1	Accepted
BC1	Dropout = 0.2	0.9662	3.07	41.7	Rejected
D1	Halved $w_{\text{mae}}, w_{\text{peak}}, w_{\text{temp}}$	0.9760	1.55	38.5	Partial
BBF	A1 + partial D1	0.9762	1.34	18.9	Accepted
BE1	BBF + 2 LSTM layers	0.9783	1.05	17.7	Final

Seed robustness: The two leading configurations, BBF and BE1, were each retrained with three random seeds (42, 123, and 7) to confirm that the reported accuracy was not an artefact of a single random initialisation. Table 3.7 reports best-checkpoint metrics for each seed together with three-seed aggregates.

Table 3.7: S-series seed robustness results at the best validation checkpoint per seed. Three-seed aggregates are given as mean $\pm$ sample standard deviation. Bold denotes the selected configuration.

Run	R^2_{tension}	R^2_x	R^2_z	Underpred [%]	Peak MAE [N]
<i>BBF ($d_{\text{LSTM}} = 256$, 1 layer)</i>					
Seed 42	0.9771	0.9918	0.9943	18.9	352
Seed 123	0.9758	0.9929	0.9959	21.5	358
Seed 7	0.9770	0.9969	0.9968	15.3	402
Mean $\pm$ std	0.9766 ± 0.0007	0.9939 ± 0.0027	0.9957 ± 0.0013	18.6 ± 3.1	371 ± 27
<i>BE1 ($d_{\text{LSTM}} = 256$, 2 layers) — selected</i>					
Seed 42	0.9783	0.9965	0.9977	17.7	393
Seed 123	0.9786	0.9974	0.9980	24.7	328
Seed 7	0.9772	0.9972	0.9960	18.8	360
Mean $\pm$ std	0.9780 ± 0.0007	0.9970 ± 0.0005	0.9972 ± 0.0010	20.4 ± 3.8	360 ± 33

BE1 exceeded BBF on every quality metric, with lower validation loss, higher R^2_{global} for all three targets, and marginally lower peak MAE. The only metric favouring BBF was the underprediction rate (18.6% versus 20.4%), a difference of 1.8 percentage points that falls within one standard deviation of both configurations and is expected to narrow at Stage C scale, where six times as many training cases give the model broader exposure to high-load sea states. BE1 was therefore selected as the final architecture, with reported accuracy $R^2_{\text{tension}} = 0.9780 \pm 0.0007$.

3.10. Final Hyperparameter Configuration

Table 3.8 lists the complete model and training configuration selected by the Stage B search, comparing the B0 baseline with the final Stage C values. Entries that changed from B0 are shown in bold. The two architectural modifications, the LSTM hidden dimension and the number of LSTM layers, encode the finding that temporal representational capacity was the primary bottleneck at the Stage B data scale.

Table 3.8: Final hyperparameter configuration for Stage C. Bold entries differ from the B0 baseline.

Parameter	B0 Baseline	Final (Stage C)
<i>Spatial encoder (GATv2)</i>		
Hidden dimension per head, layer 1	64	64
Output dimension, layer 2	64	64
Attention heads, layer 1	4	4
<i>Temporal encoder (LSTM)</i>		
Hidden dimension d_{LSTM}	128	256
Number of layers	1	2
<i>Prediction head</i>		
Hidden dimension	128	128
<i>Regularisation</i>		
Dropout p (all modules)	0.1	0.1
<i>Physical loss weights</i>		
w_{mae}	0.005	0.0025
w_{peak}	0.010	0.005
w_{under}	0.020	0.020
w_{temp}	0.005	0.0025
<i>Optimisation</i>		
Optimiser	AdamW	AdamW
Learning rate η	10^{-3}	10^{-3}
Weight decay λ	10^{-4}	10^{-4}
LR scheduler factor / patience	0.5 / 5	0.5 / 5
Batch size	32	32
Early-stopping patience (epochs)	20	30
Maximum epochs	40	100

All runs used the AdamW optimiser [46], which applies weight decay directly to the parameter values rather than to the gradients, decoupling the regularisation effect from the per-parameter adaptive step sizes of the Adam update. Default coefficients $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$ were retained throughout. The learning rate followed a reduce-on-plateau schedule, halving η whenever the validation loss did not decrease for five consecutive epochs. The extended epoch ceiling and patience for Stage C accommodate the larger training distribution, in which each epoch draws a fresh stratified subsample of 30 000 windows from a pool of approximately 500 000. The model is expected to converge more gradually than under Stage B's 50-case per-location distribution, and the higher patience threshold prevents premature early stopping before convergence.

4

Model Performance

The mixed-topology GAT+LSTM surrogate described in Chapter 3 is evaluated on a held-out test set comprising temporal windows withheld from the development locations and all windows from two completely unseen deployment sites. This chapter presents the results across two complementary lines of investigation. The first is the engineering result. Run C1, a single model trained on 300 sea-state realisations per location, is assessed on its held-out test accuracy, its per-topology behaviour, and its convergence characteristics. The second is a methodological contribution, multi-task uncertainty weighting via the Kendall framework, which replaces the four hand-tuned physical-loss weights with jointly learned task uncertainties and yields an interpretable ranking of the physical loss terms. Three variants, C2a through C2c, trace the path from an initial σ -collapse failure mode to a stable four-term formulation using a $\tau = 0.9$ pinball loss. Run C1 remains the defended primary model. The Kendall variants constitute an independent methodological finding.

This chapter is organised as follows. Section 4.1 recaps the experimental configuration and hardware. Section 4.2 presents the held-out test performance of C1 and analyses its training dynamics. Section 4.3 demonstrates spatial load-field reconstruction and includes per-topology and out-of-distribution generalisation results. Section 4.4 opens with architectural ceiling evidence, then decomposes the variance structure of the predictions and characterises the spatial–temporal capability boundary. Section 4.5 presents the multi-task uncertainty weighting investigation. Section 4.6 reports the measured training cost and discusses inference cost qualitatively. Section 4.7 closes with a summary of findings.

4.1. Experimental Setup

Run C1 uses the final architecture and hyperparameter configuration described in Section 3.10 and summarised in Table 3.8.

Dataset and split: Stage C scales the training corpus from the 50 cases per location used in Stage B to 300 cases per location, increasing the available sea-state diversity by a factor of six. After applying the 53-case corrupt blacklist described in Section 2.5, the effective training pool contains approximately 500 000 candidate sliding windows. A stratified resampler draws 30 000 of these per epoch, cycling through different subsets across epochs so that the full pool is covered over the course of training. Development locations 1, 3, 4, 5, 6, 7, 8, and 9 supply training, validation, and temporal-holdout test windows. Locations 10 and 11 serve exclusively as out-of-distribution (OOD) test locations, providing entirely unseen deployment sites with different environmental conditions. The validation cap is 10 000 windows and the combined test cap is 30 000 windows (see Table 2.4).

Hardware and software: All computations were performed on the DelftBlue supercomputer at TU Delft [47]. Stage B hyperparameter search runs each used a single NVIDIA A100 GPU (80 GB HBM2e), 8 CPU cores, and 32 GB of host RAM. Stage C runs, including C1 and all C2 variants, used the same A100 GPU with 16 CPU cores and 64 GB of host RAM, to accommodate the larger dataset index and

DataLoader memory footprint that accompany the 300-case pool. Before each Stage C job, the 13 GB preprocessed cache was copied from the parallel BeeGFS file system to the node-local NVMe storage, eliminating file-system latency from the per-epoch data-loading path. The software stack comprised PyTorch 2.1.0 [48], PyTorch Geometric 2.7.0 [49], and NumPy 1.24.1. Peak GPU memory usage across all runs was 5.2 GB, well within the A100 capacity, indicating that compute throughput rather than memory is the dominant resource.

Checkpoint selection: For the primary run C1, the defended checkpoint is selected by the minimum of the composite validation loss $\mathcal{L}_{\text{val}} = \mathcal{L}_{\text{data}} + \sum_k w_k \mathcal{L}_k^{\text{phys}}$, which mirrors the training objective directly. For the Kendall uncertainty-weighting runs (C2a, C2b, C2c), the validation loss under the Kendall objective accumulates the $\frac{1}{2}s_k$ regularisation terms, so it can become negative and, in the presence of σ -runaway, anti-correlated with predictive accuracy as σ_k evolves. Consequently, the checkpoint is selected by the maximum validation global R_{tension}^2 for those variants. All inter-run comparisons are made at each run’s respective selected checkpoint, so the comparison is always among operating-point models, not among arbitrary epochs.

4.2. Primary Model Held-out Test Performance

Run C1 was trained from random initialisation on the full 300-case dataset, using seed 42 and early stopping with patience 30. Training ran for 42 epochs before the patience criterion was satisfied. The defended checkpoint is epoch 12, at which the composite validation loss reached its minimum of 0.00658. Table 4.1 summarises the held-out test metrics at that checkpoint, together with the Stage-B BE1 three-seed mean for comparison.

Table 4.1: Held-out test performance of the Stage C primary model (C1, defended checkpoint epoch 12) compared with the Stage-B BE1 three-seed mean. All metrics are computed in physical units.

Metric	Stage-B BE1 (3-seed mean $\pm$ std)	Stage-C C1 (test)
R_{tension}^2 (global)	0.9780 ± 0.0007	0.9800
$R_{x_{\text{abs}}}^2$ (global)	0.9970 ± 0.0005	0.9957
$R_{z_{\text{abs}}}^2$ (global)	0.9972 ± 0.0010	0.9954
MAE_T [N]	257 ± 7	231
$\text{MAE}_{x_{\text{abs}}}$ [m]	0.98 ± 0.07	1.00
$\text{MAE}_{z_{\text{abs}}}$ [m]	0.36 ± 0.06	0.35
Peak underprediction rate [%]	20.4 ± 3.8	17.6
Peak-tension bias [N]	$+190.6 \pm 51.0$	+197.5
$\text{MAPE}_{\text{tension}}$ [%]	10.5 ± 0.1	9.06
Peak-tension MAE [N]	360.4 ± 32.8	332.5
Peak max abs. error [kN]	5.7 ± 0.2	22.2
Temporal diff. MAE_T [N/step]	53.1 ± 2.8	45.4
Temporal diff. $\text{MAE}_{x_{\text{abs}}}$ [m/step]	0.0157 ± 0.0015	0.0131
Temporal diff. $\text{MAE}_{z_{\text{abs}}}$ [m/step]	0.0151 ± 0.0005	0.0125

Before the two columns of Table 4.1 are compared, one qualification is needed. The Stage-B BE1 figures are validation metrics computed on the development locations with 50 cases per location, whereas the Stage-C C1 figures are held-out test metrics evaluated over 300 cases per location including the two out-of-distribution sites. The two columns therefore describe different quantities on different data, and the comparison below indicates consistency of scale rather than a controlled improvement.

With that caveat, the model achieves a global $R_{\text{tension}}^2 = 0.9800$ on the held-out test set, against the Stage-B BE1 validation mean of 0.9780 ± 0.0007 . The peak underprediction rate is 17.6 % against 20.4 %, and the peak-tension bias is safely positive at +197.5 N, indicating that the model tends to overestimate rather than underestimate tension peaks. The global R^2 for horizontal and vertical displacement is marginally lower than the Stage-B mean, reflecting that the primary quality gain from the full dataset is concentrated in the tension target, which carries a larger variance across sea states and is harder to generalise.

One entry in Table 4.1 moves in the opposite direction and should be read carefully. The maximum absolute peak error rises from 5.7 kN at Stage B to 22.2 kN at Stage C. This is a property of the evaluation set rather than a regression in the model. The Stage-C figure is a maximum over a test pool that is six times larger in cases, spans the two withheld locations, and therefore contains more extreme sea states than the Stage-B validation pool, and a maximum over a larger and more energetic sample can only grow. What the figure does establish is that the worst single peak error remains at the kilonewton scale. The 53-case corrupt blacklist has removed the meganewton-scale outliers that contaminated the pre-blacklist baseline, and the per-topology breakdown in Table 4.3 localises the two largest values to $N = 12$ and $N = 18$ rather than spreading them across all discretisations.

Training dynamics: Figure 4.1 shows the full training history over the 42 epochs of run C1. The composite training loss decreases from approximately 0.101 at epoch 1 to 0.0094 by epoch 4, and then continues to fall gradually to 0.0057 at epoch 42, while the validation loss reaches its minimum of 0.00658 at epoch 12 and then rises to values between 0.013 and 0.017 by the end of training. The tension R^2 peaks at epoch 4 (the best-validation- R^2 checkpoint, 0.9788), and then drifts downward to approximately 0.965 by epoch 40. The position R^2 values for x_{abs} and z_{abs} remain high throughout, mostly above 0.97 after the first epoch, with a less pronounced late-epoch decline. The gradient norm falls from about 3.1 at epoch 1 to below 0.5 by epoch 5 and remains low and stable thereafter, in the range 0.14–0.30 over the second half of training, ruling out gradient explosion or collapse as a cause of the diverging gap between the training and validation losses. This late-epoch divergence, and the evidence that the resulting accuracy plateau is architectural rather than data-driven, is taken up in Section 4.4.1.

Stage-C1 training dynamics (42 epochs) — defended checkpoint at epoch 12 (dashed line); LR reductions at epochs [11, 19, 25, 31, 37]

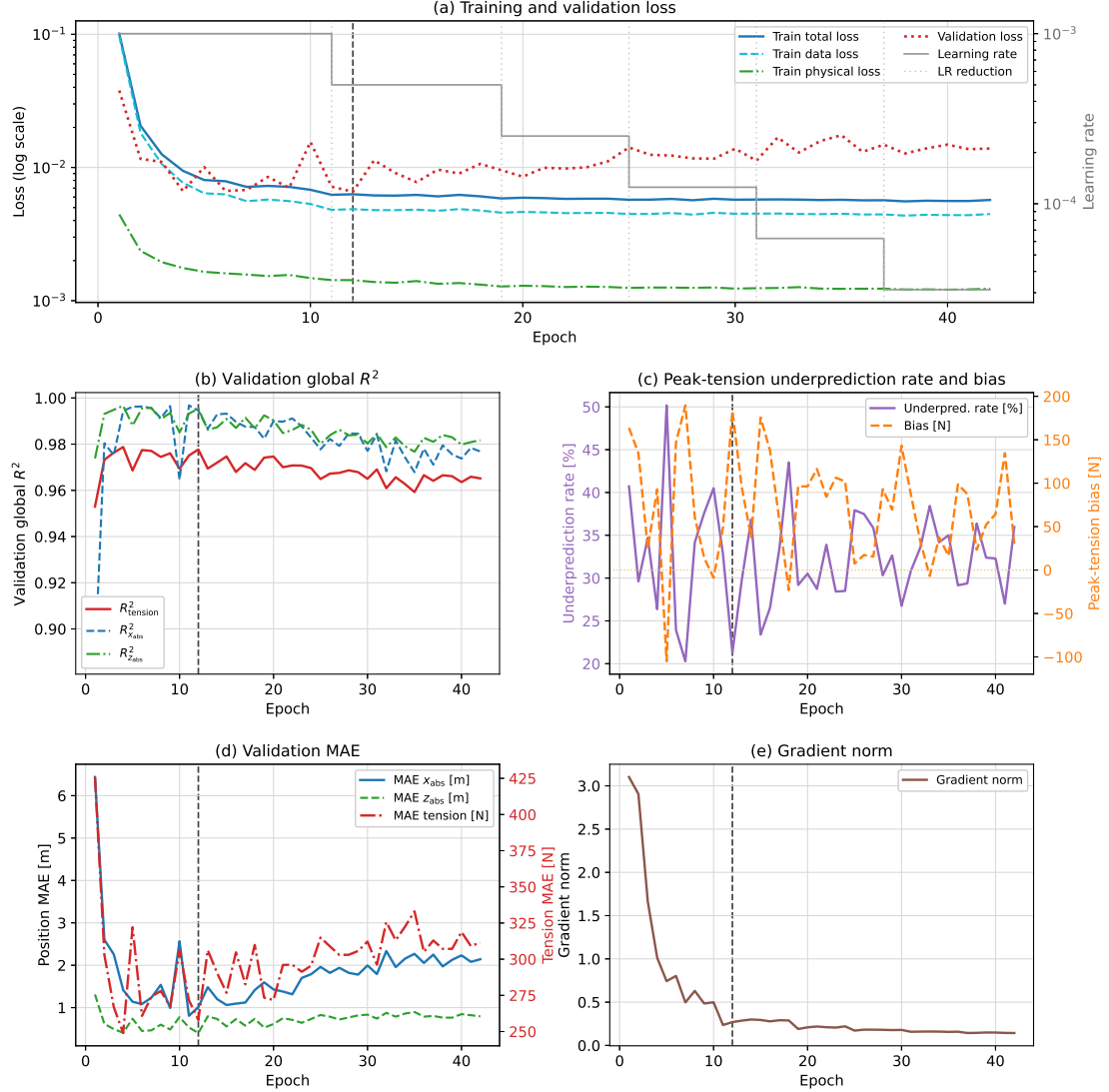


Figure 4.1: Stage-C1 training history over 42 epochs. (a) Loss curves: training total loss, data loss, physical loss, and validation loss, with the learning-rate reduction events marked on the secondary axis. (b) Validation global R^2 for all three targets. (c) Validation peak-tension underprediction rate and bias. (d) Validation MAE for position and tension targets. (e) Gradient norm. The vertical dashed line marks the defended checkpoint at epoch 12 (minimum validation loss).

4.3. Load-Field Reconstruction and Topology Generalisation

The aggregate metrics of Section 4.2 characterise overall accuracy but do not break down performance by topology or reveal the physical character of the predictions. Section 4.3.1 addresses the per-topology breakdown, and Section 4.3.2 then demonstrates the spatial reconstruction capability through qualitative figures drawn from the two out-of-distribution test locations. The full spatial-temporal anatomy of the R^2 figure is presented in Section 4.4.

4.3.1. Per-Topology and OOD Generalisation

The model is evaluated separately for each of the ten target discretisations defined in Section 3.1, from $N = 4$ to $N = 21$ nodes. The per-topology test metrics are reported in Tables 4.2–4.4, with position and tension accuracy in Table 4.2, peak-tension safety in Table 4.3, and temporal-difference accuracy

in Table 4.4.

Table 4.2: Stage-C1 per-topology test metrics (a): position and tension accuracy. Absolute positions x_{abs} and z_{abs} in metres, tension in newtons.

N	R^2_{tension}	$R^2_{x_{\text{abs}}}$	$R^2_{z_{\text{abs}}}$	MAE_x [m]	MAE_z [m]	MAE_T [N]	MAPE [%]
4	0.9863	0.9983	0.9983	0.70	0.30	233.2	8.64
5	0.9861	0.9976	0.9982	0.84	0.27	219.6	7.95
6	0.9824	0.9973	0.9973	0.97	0.31	232.7	8.48
7	0.9828	0.9970	0.9967	0.96	0.34	232.2	8.63
8	0.9827	0.9964	0.9965	1.07	0.34	223.6	8.74
10	0.9807	0.9967	0.9956	0.96	0.37	230.1	9.62
12	0.9791	0.9958	0.9947	1.01	0.39	232.1	9.85
15	0.9790	0.9949	0.9945	1.12	0.38	228.0	9.19
18	0.9765	0.9946	0.9938	1.13	0.39	237.0	9.49
21	0.9761	0.9941	0.9939	1.21	0.39	240.7	10.01

Table 4.3: Stage-C1 per-topology test metrics (b): peak-tension safety. A positive bias denotes safe overprediction, and the underprediction rate is the fraction of nodes whose predicted peak falls below the true peak.

N	Underpred. [%]	Bias [N]	Peak MAE [N]	Peak max abs. err. [kN]
4	12.8	295.8	394.3	4.2
5	15.8	221.7	333.7	4.6
6	19.1	177.0	333.9	8.4
7	15.6	227.1	343.8	4.9
8	15.9	204.2	321.2	4.4
10	17.5	197.6	324.3	5.9
12	18.5	174.4	323.0	21.8
15	18.4	176.4	318.7	5.8
18	20.2	158.0	318.1	22.2
21	21.9	143.0	314.5	13.6

Table 4.4: Stage-C1 per-topology test metrics (c): temporal-difference MAE, the consecutive-step rate-of-change error with one step equal to 0.1 s.

N	ΔT [N/step]	Δx [m/step]	Δz [m/step]
4	49.4	0.0159	0.0142
5	46.2	0.0144	0.0122
6	46.7	0.0136	0.0122
7	45.8	0.0131	0.0128
8	44.5	0.0130	0.0124
10	44.5	0.0125	0.0122
12	44.3	0.0123	0.0122
15	43.7	0.0120	0.0121
18	44.7	0.0121	0.0124
21	44.4	0.0121	0.0123

The tension R^2 decreases near-monotonically from 0.9863 at $N = 4$ to 0.9761 at $N = 21$. This gradient arises because a finer discretisation places more nodes in the region of the line that undergoes rapid spatial variation, including the touchdown zone and the upper catenary, and requires the model to resolve fine-scale tension and curvature gradients that are averaged out at coarser resolutions. At every node count, the held-out test R^2_{tension} equals or exceeds the validation R^2_{tension} of the defended checkpoint, and at eight of the ten node counts it also exceeds the best validation value reached at any epoch during training (Figure 4.2), an outcome that reflects two complementary effects. First, the test set includes windows drawn from OOD locations 10 and 11, which expose the model to deployment sites with different water depths and environmental forcing than the development set. The absence of

performance degradation on these windows demonstrates genuine spatial generalisation. Second, the overfitting visible in the late-epoch training dynamics (Section 4.2) affects the validation trajectory but is arrested at the checkpoint level by the min-validation-loss selection rule, as the defended checkpoint at epoch 12 lies before the regime of late-epoch overfit.

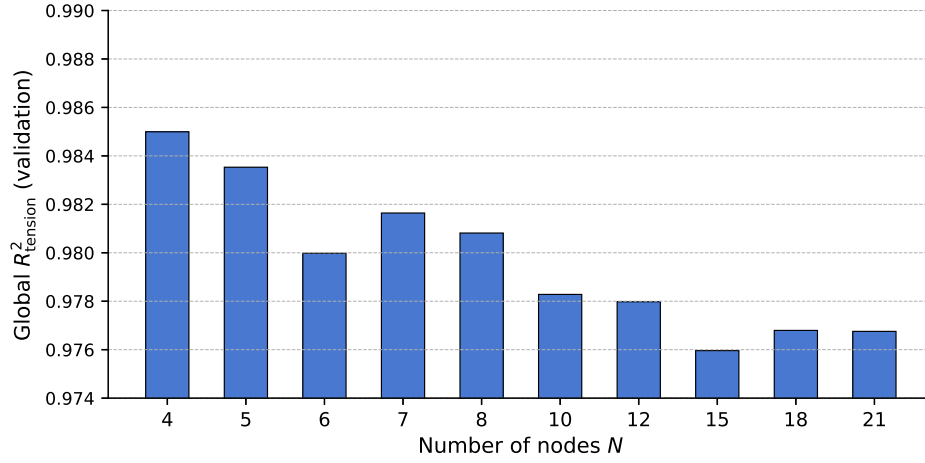


Figure 4.2: Best per-epoch global R^2_{tension} on the validation set as a function of node count N . Performance decreases from 0.985 at $N = 4$ to 0.977 at $N = 21$.

4.3.2. Spatial Reconstruction of the Load Field

Spatial tension profile: Figure 4.3 shows the predicted and true per-node peak tension, defined as the maximum over the 5 s forecast window, as a function of node index from anchor to fairlead for a representative OOD window. The peak is the design-relevant quantity, since a structural assessment is governed by the largest load each segment carries rather than by the value at an arbitrary instant. The surrogate reconstructs the characteristic peak distribution, near-uniform and low through the seabed-resting catenary section and rising monotonically toward the fairlead, and it tracks the high-tension fairlead region almost exactly. In the low-tension grounded section it sits slightly above the true peak, a conservative bias of the same sign as the dataset-wide positive peak-tension bias of +197.5 N (Table 4.1) and examined in detail in Section 4.5.3. This profile reconstruction is the quantity most directly relevant to design assessments that need the load distribution across the full line rather than the time history at a single node.

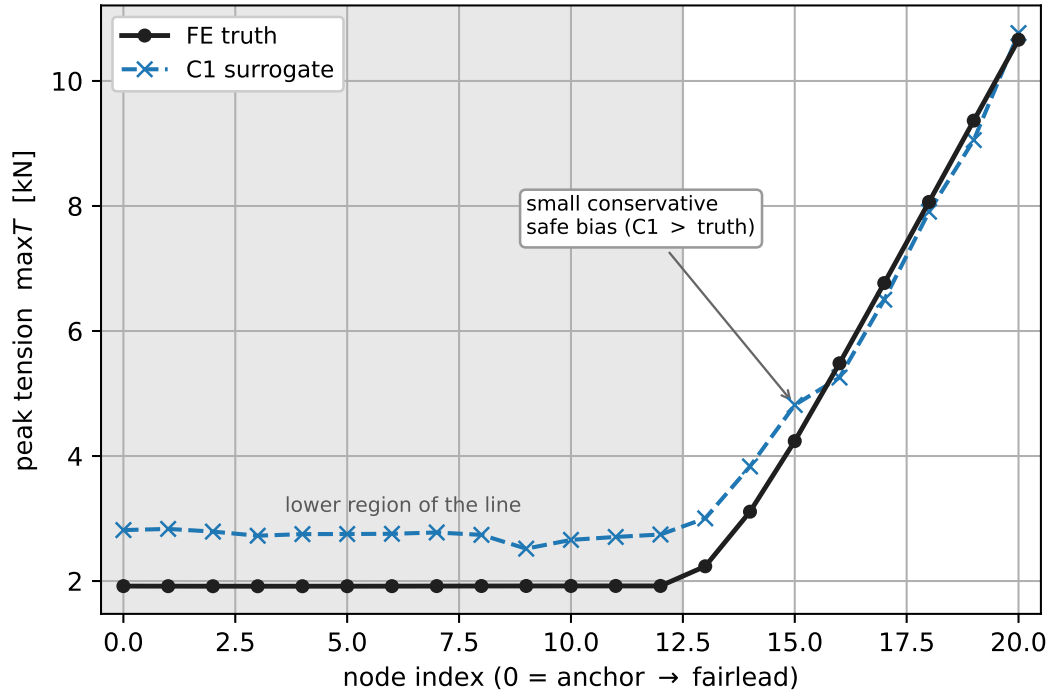


Figure 4.3: Predicted (dashed) and true (solid) per-node peak tension, the maximum over the 5 s forecast window, along the mooring line for a representative out-of-distribution test window (location 11, $N = 21$). Node 0 is the anchor and node 20 is the fairlead.

Mooring line shape: Figure 4.4 shows the predicted and true mooring line geometry, plotted as the node coordinates $(x_i^{\text{abs}}, z_i^{\text{abs}})$ for all N nodes at the $t = +5.0$ s forecast horizon. The predicted catenary follows the true shape closely, including the touchdown region where the chain transitions from resting on the seabed to hanging freely. The spatial R^2 for both horizontal and vertical position exceeds 0.99 (Table 4.1), consistent with the near-perfect overlap visible in the figure.

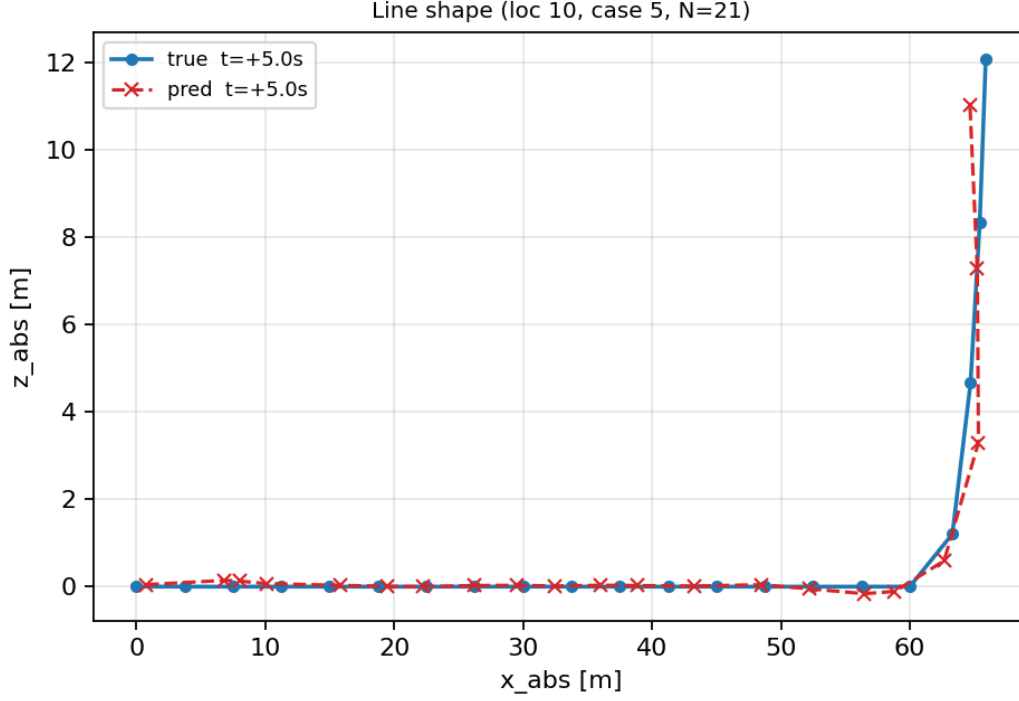


Figure 4.4: Predicted (dashed) and true (solid) mooring line geometry at the $t = +5.0$ s forecast horizon, for a representative out-of-distribution test window (location 10). Node 0 is the anchor (left) and node 20 is the fairlead (right).

Topology generalisation: Figure 4.5 shows the predicted and true tension profile for the same OOD sea-state window at four representative node counts, $N = 4$, $N = 8$, $N = 12$, and $N = 21$, one panel per discretisation. A single set of model weights reproduces the spatial tension distribution across all four discretisations. The coarser $N = 4$ model captures the anchor-to-fairlead gradient at its four sampling positions, while the finer $N = 21$ model resolves the profile at full resolution, with $N = 8$ and $N = 12$ showing intermediate behaviour. This one-weight, many-topology design is the architectural contribution that enables the surrogate to serve across a range of mooring line discretisations without retraining, and the test R^2_{tension} at every node count confirms that the single model does not sacrifice per-topology accuracy for this generality (Section 4.3.1). Figure 4.6 shows the same reconstruction plotted against normalised arc length, where the shaded band makes the surrogate’s small conservative tension bias directly visible.

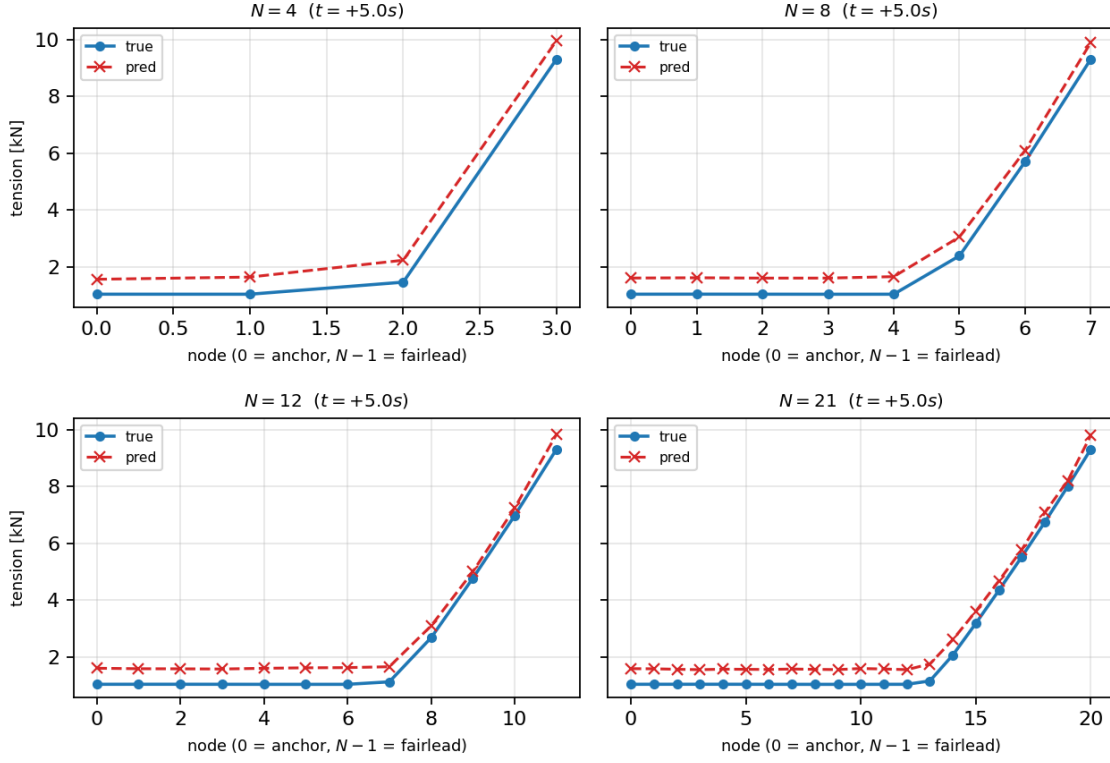


Figure 4.5: Predicted (dashed) and true (solid) tension profiles for the same out-of-distribution sea-state window (location 11), shown as four panels at node counts $N \in \{4, 8, 12, 21\}$. Each panel plots tension against node index from anchor (0) to fairlead ($N - 1$) at the $t = +5.0$ s forecast horizon.

In each panel of Figure 4.5 the predicted profile shows a near-uniform absolute overprediction of approximately 500 N along the grounded section. This is the same conservative behaviour that the dataset-wide peak-tension bias of +197.5 N (Table 4.1) measures on average, although this particular window sits well above that average. Because the offset is small relative to the fairlead tension but comparable to the tension itself in the grounded section, it appears as a $\approx 2\text{--}7\%$ relative gap near the fairlead but a $\approx 50\%$ relative gap near the anchor, an artefact of the safety margin being spatially uniform while the load magnitude it is calibrated against is not.

One weight set, every topology: the reconstructed mooring load field ($N = 4 \rightarrow 21$, global $R^2_{\text{tension}} = 0.98$)

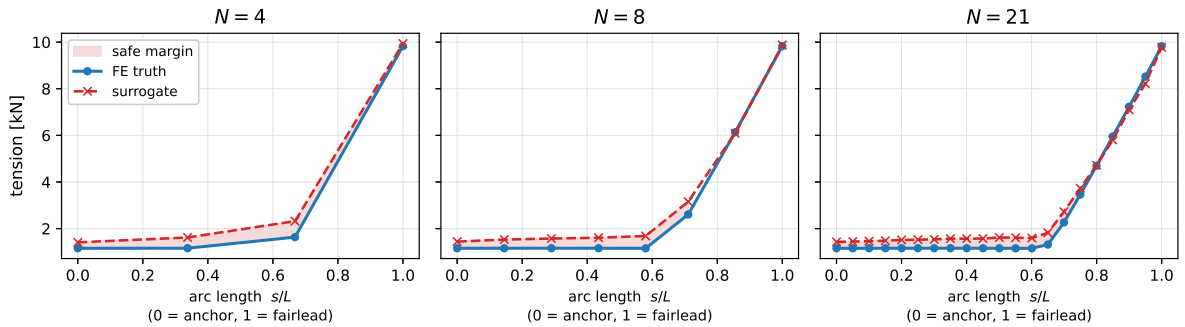


Figure 4.6: Spatial tension load field reconstructed by the C1 surrogate at three discretisations ($N = 4, 8$ and 21) from a single set of weights, plotted against normalised arc length s/L from anchor ($s/L = 0$) to fairlead ($s/L = 1$) for an out-of-distribution sea-state window (location 11).

4.4. A Spatial--Temporal Capability Anatomy

The headline tension $R^2 = 0.9800$ is both accurate and honest, yet it certifies a specific and bounded kind of accuracy, and it does not rise with more data or stronger regularisation. This section first establishes that the ≈ 0.98 tension accuracy is an architectural ceiling rather than a data-limited one, and then decomposes the total tension variance to show that this ceiling is in fact a *saturation* of the predictable spatial and level variance. It then characterises the residual temporal blind spot and explains the mechanism. The analysis draws on held-out test windows processed through the C1 checkpoint, 2 000 for the variance decomposition and 400 for the predictability study, together with the data-scaling and regularisation evidence of Table 4.5.

4.4.1. The Architectural Ceiling

The divergence between the decreasing training loss and the increasing validation loss from epoch 12 onward, visible in the training history of Figure 4.1, is characteristic of overfitting. The same early-peak-then-overfit signature was observed in the Stage-B BE1 S-series at 50 cases per location, where the two-layer LSTM peaked early and then degraded over later epochs. The six-fold increase to 300 cases per location substantially reduces per-epoch variance in the sampled windows, yet does not eliminate the phenomenon. The model still peaks before epoch 12 on the R^2 criterion and overfits thereafter.

This finding argues strongly against the data-starvation hypothesis, namely that BE1’s Stage-B accuracy of 0.9780 was limited primarily by insufficient training data. The final test accuracy of 0.9800 is consistent with BE1’s Stage-B operating envelope, and the trajectory after epoch 12 confirms that the model has saturated its representational capacity against the given distribution of windows before the training corpus is exhausted. The ceiling is therefore a property of the architecture and task, not of the dataset size. Table 4.5 presents the three data points that collectively support this conclusion.

Table 4.5: Evidence for an architectural accuracy ceiling. Increasing the training data by $6\times$ (Stage B to Stage C) and then adding mild dropout regularisation both fail to lift the tension R^2 beyond the observed plateau.

Configuration	Cases	R^2_{tension}	Underpred.[%]	Best ep
Stage-B BE1 (3-seed mean)	50	0.9780 ± 0.0007	20.4 ± 3.8	18 / 40
Stage-C C1	300	0.9800	17.6	12 / 42
C1-dropout 0.15	300	0.9792	15.9	7 / 100

The Stage-B result at 50 cases per location gives $R^2_{\text{tension}} = 0.9780 \pm 0.0007$ across three seeds, already displaying the early-peak-then-overfit pattern. Scaling to 300 cases per location (Stage-C C1) yields $R^2_{\text{tension}} = 0.9800$ and a marginally lower underprediction rate. The improvement of $+0.002$ has the direction expected of a data-limited model but not the magnitude, since a six-fold data increase would typically drive a substantially larger accuracy gain. A mild dropout increase from the base 0.1 to 0.15 on all three modules yields $R^2_{\text{tension}} = 0.9792$, an earlier best epoch of 7, and validation loss collapsing to values of 0.01–0.03 from epoch 8 onward. The dropout worsened rather than alleviated the overfit, confirming that at 300-case scale overfitting is not the dominant accuracy constraint. No further regularisation or data-scaling levers were explored.

What “architectural” does and does not mean here: Ruling out data quantity and regularisation strength narrows, but does not by itself identify, the limiting factor. It does not imply that a larger version of the current encoder-decoder would perform better. Indeed, the Stage-B search already tested increased temporal-encoder capacity (LSTM hidden size $128 \rightarrow 256$, one to two layers) and found large gains in position accuracy but negligible movement in tension R^2 (Section 3.9), while additional regularisation here moves the ceiling in the wrong direction. Parsimony is therefore not in conflict with this finding. The present, comparatively compact architecture already sits at the ceiling that its design class, a single-shot, non-autoregressive decoder conditioned on sea-state summary statistics, can reach, and neither more parameters nor more data close the remaining gap. Section 4.4.4 identifies what would move the ceiling, a qualitatively different decoding strategy or additional input information, a change in architectural *class* rather than architectural *size*.

The variance decomposition of Section 4.4.2 reframes this ceiling as a saturation of the predictable variance rather than a failure, and the implications are discussed in Chapter 5.

4.4.2. The Variance Anatomy

The total tension variance across the test set arises from three sources. The *level* component captures variation in the overall magnitude from one sea-state window to the next, reflecting changes in wave height, period, and current across the 300 simulated cases. The *spatial* component captures variation from node to node within a single time step, reflecting the catenary geometry, seabed reaction, and drag forces that create the anchor-to-fairlead tension gradient. The *temporal* component captures within-node variation over the $F = 50$ forecast steps at fixed sea-state conditions, reflecting the wave-frequency tension oscillation.

Table 4.6 reports the fraction of total variance attributable to each source for the three model targets, computed over 2 000 randomly sampled held-out windows.

Table 4.6: Decomposition of total variance in the held-out test set into level (between sea-state windows), spatial (between nodes within a snapshot), and temporal (within-node over the forecast horizon) components. Values are percentages of total variance per target, computed over 2 000 windows drawn from the full held-out test set (development-location temporal test windows and OOD locations 10–11) with the 53-case corrupt blacklist applied. Entries are rounded independently, so a row need not sum to exactly 100 %.

Target	Level [%]	Spatial [%]	Temporal [%]
Tension T	13	85	1.5
x_{abs}	8	92	0.0
z_{abs}	6	94	0.1

The spatial component dominates for all three targets. For positions, the temporal component is essentially zero because the mooring line is quasi-static over the 5-second forecast horizon. For tension, the spatial component accounts for 85 % of the total variance and the sea-state level for a further 13 %, leaving 1.5 % as the wave-frequency oscillation that drives the tension peaks.

Formal derivation of the decomposition: The percentages in Table 4.6 follow from a nested sum-of-squares decomposition of the ground-truth future tension, computed on the future block of $F = 50$ steps only. Let $T_{w,t,i}$ denote the true tension at node $i \in \{0, \dots, N-1\}$, future step $t \in \{1, \dots, F\}$, and window $w \in \{1, \dots, W\}$, with $F = 50$ and $N = 21$, retaining the anchor-to-fairlead node indexing of Section 3.1. Three nested sample means are defined,

$$\begin{aligned}\mu &= \frac{1}{WFN} \sum_{w,t,i} T_{w,t,i} \\ \bar{T}_w &= \frac{1}{FN} \sum_{t,i} T_{w,t,i} \\ \bar{T}_{w,i} &= \frac{1}{F} \sum_t T_{w,t,i}\end{aligned}$$

where μ is the grand mean across the full test pool, $\bar{T}_w$ is the mean over all nodes and timesteps within window w and represents the sea-state level, and $\bar{T}_{w,i}$ is the time-averaged tension at node i during window w and represents that node's position on the spatial profile once the sea-state level is removed. Every entry decomposes exactly by adding and subtracting these two intermediate means:

$$T_{w,t,i} - \mu = \underbrace{(\bar{T}_w - \mu)}_{\text{level}} + \underbrace{(\bar{T}_{w,i} - \bar{T}_w)}_{\text{spatial}} + \underbrace{(T_{w,t,i} - \bar{T}_{w,i})}_{\text{temporal}}$$

The three cross-terms that would otherwise appear in the expanded squared sum vanish due to the zero-sum properties of the nested means. For any fixed (w, i) , summing the temporal deviation over t

gives $\sum_t (T_{w,t,i} - \bar{T}_{w,i}) = 0$ by definition of $\bar{T}_{w,i}$, and for any fixed w , summing the spatial deviation over i gives $\sum_i (\bar{T}_{w,i} - \bar{T}_w) = 0$ by definition of $\bar{T}_w$. The total sum of squares therefore partitions exactly as:

$$SS_{\text{total}} = SS_{\text{level}} + SS_{\text{spatial}} + SS_{\text{temporal}},$$

where each component is given by:

$$\begin{aligned} SS_{\text{level}} &= FN \sum_w (\bar{T}_w - \mu)^2 \\ SS_{\text{spatial}} &= F \sum_{w,i} (\bar{T}_{w,i} - \bar{T}_w)^2 \\ SS_{\text{temporal}} &= \sum_{w,t,i} (T_{w,t,i} - \bar{T}_{w,i})^2 \end{aligned}$$

The multiplicative factors $FN = 1050$ and $F = 50$ reflect how broadly each deviation is shared across entries. A window-level deviation $(\bar{T}_w - \mu)$ takes the same value at all FN entries within that window, a node deviation $(\bar{T}_{w,i} - \bar{T}_w)$ is constant across the F timesteps of that node, and temporal deviations are unique per entry and carry no factor. The three variance shares in Table 4.6 are obtained as SS_k/SS_{total} for each k , pooled over $W = 2000$ windows drawn from the full held-out test set with the 53-case blacklist applied. The blacklist is essential, as a single near-corrupt window with a within-node spike near the 1.5 MN boundary would inflate SS_{temporal} until it dominated the entire sum, raising the apparent temporal share from the genuine 1.5 % to approximately 96 %.

The implication for the R^2 figure is direct. The global R^2 equals the fraction of total variance explained. A model that reconstructs the spatial profile and sea-state level accurately and outputs a flat line for the temporal component leaves exactly the temporal share unexplained, so it would explain the complement of that share, $100 - 1.5 = 98.5$ % of the tension variance, reaching a ceiling of approximately $R^2 = 0.985$. All subsequent references to the 98.5 % figure denote this complement of the temporal share. The additional 0.015 would require accurate wave-frequency temporal tracking. The reported $R_{\text{tension}}^2 = 0.9800$ is consistent with this analysis and confirms that the surrogate is excellent at the dominant (spatial and level) components of the signal.

Figure 4.7 visualises the decomposition.

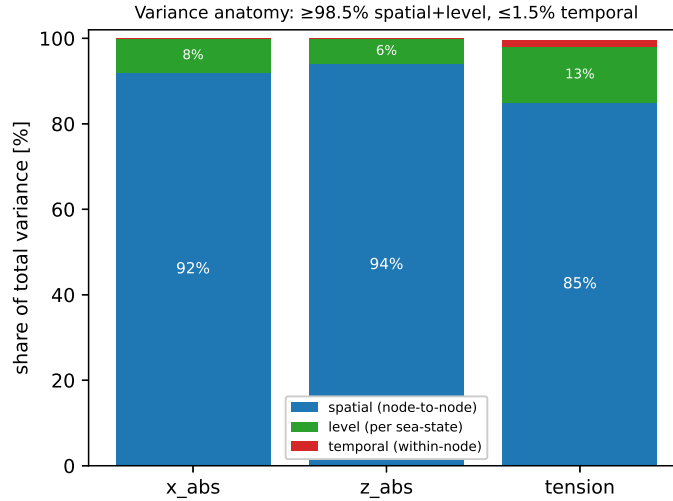


Figure 4.7: Variance decomposition of the held-out test set for each model target. Each bar represents 100 % of the total variance for that target, divided into level (between windows), spatial (between nodes), and temporal (within-node oscillation) components.

4.4.3. The Temporal Blind Spot

Although the temporal component accounts for only 1.5 % of the total tension variance, it contains the wave-frequency oscillation that drives the tension peaks. This subsection characterises how well the

model tracks that oscillation.

Time-series comparison: Figure 4.8 shows the predicted and true tension time series at the fairlead node over the 50-step forecast horizon for two OOD test windows. In both cases, the model produces a near-flat output close to the conditional mean of the fairlead tension under that sea state, while the true tension oscillates at the wave frequency. The level is reproduced accurately, but the wave-by-wave variation is absent from the prediction.

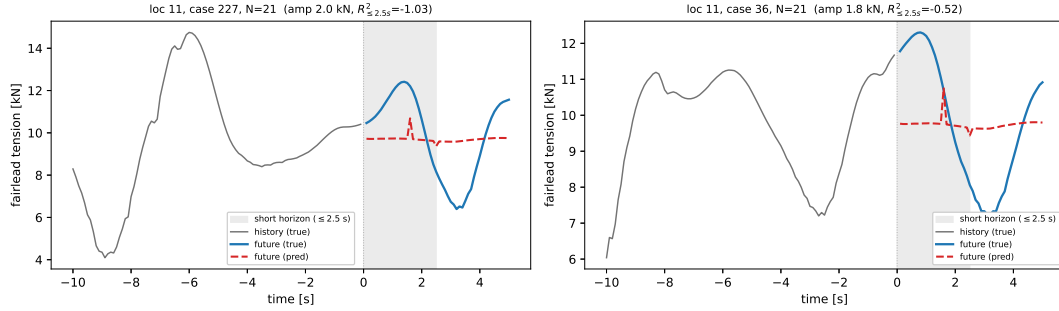


Figure 4.8: Predicted (dashed) versus true (solid) fairlead tension over the 50-step forecast horizon (5 s at $\Delta t = 0.1$ s) for two out-of-distribution test windows.

Predictability study: Not all 5-second futures are equally difficult to forecast. The incident wave train can produce a partially regular tension response when swell components dominate, or a fully irregular response under broad-band forcing. To quantify how much of the temporal oscillation is in principle recoverable from the 250-step response history alone, a structured predictability study was carried out on fairlead-tension windows drawn from the two out-of-distribution test locations (locations 10 and 11), at $N = 21$ nodes. Up to three windows were sampled per simulation case with a fixed random seed. Windows with a future tension standard deviation below 0.3 kN were then excluded, as near-flat futures yield numerically unstable temporal R^2 estimates and carry negligible engineering significance. All 400 windows referred to below are those retained after this amplitude filter, so every percentage quoted in this subsection has 400 as its denominator. The study is therefore scoped to oscillating, high-energy windows at the fairlead node, the most dynamically loaded node on the chain, under conditions the model has never seen at training time.

Each retained window was evaluated against three history-only baselines that use only the observed fairlead tension over the 250-step history and require no knowledge of the future forcing. The persistence baseline repeats the last observed value across all 50 future steps. The one-harmonic baseline computes the discrete Fourier transform of the history, identifies the dominant spectral component, fits its amplitude and phase, and extrapolates the resulting cosine into the 50-step future window. The five-harmonic baseline sums the five strongest spectral components in the same way. The temporal R^2 for each baseline and for the surrogate is computed within the window against the window's own mean, isolating the wave-frequency oscillation from the sea-state level, already accounted for by the decomposition in Section 4.4.2. A window is classified as predictable if the best temporal R^2 across the three baselines exceeds 0.3, meaning that a straightforward frequency extrapolation of the response history alone explains at least 30 % of the within-window tension variance.

Figure 4.9 shows the model temporal R^2 as a function of the best baseline R^2 for each window. The diagonal marks the break-even line, and no model point lies above it in the predictable area. Of the 400 windows, 31 % are classified as predictable by this criterion, with a median baseline R^2 of 0.60 on that subset. Because only simple fixed-frequency baselines are considered, the 31 % figure is a lower bound on the truly recoverable fraction, as a more sophisticated history-only predictor could classify additional windows. On those same windows the surrogate achieves a median temporal R^2 of -0.29 , and it exceeds the best baseline in none of them. Pooled across all 400 windows, the model does beat the best baseline in 28.7 % of cases, but all of these instances lie in the non-predictable subset ($x < 0.3$), where both the model and the baselines score negative temporal R^2 and the comparison reflects which predictor fails less badly rather than genuine temporal skill. Across all 400 windows, 85.5 % of model

R^2 values are negative, meaning the surrogate is outperformed by a flat constant predictor in the large majority of cases.

The critical interpretation is that the predictable 31 % is recoverable from the response history by a simple frequency extrapolation, precisely the same signal that passes through the surrogate's LSTM encoder over 25 seconds of input. The limitation is therefore not a shortage of information in the model's inputs but the incentive structure of the training objective. Under an averaged SmoothL1 loss applied to a dataset in which wave phase is effectively random across windows, the loss-minimising output for the temporal oscillation is the conditional mean, a near-flat prediction, regardless of what the history encodes. Recovering the predictable 31 % is an architectural and loss-function problem, addressable in principle through a sequential decoder with step-to-step feedback or a spectral training objective, not through additional sensor inputs. The remaining 69 % of windows, where even the best baseline falls below the predictability threshold, represents a genuine input ceiling. Addressing it requires either supplying the future incident wave elevation as a decoder input or replacing the point prediction with a probabilistic output that quantifies the irreducible phase uncertainty. The distinction between these two regimes, a recoverable architectural fraction and a physically irreducible fraction, is interpreted in Chapter 5 and defines the two extension paths set out in Chapter 7.

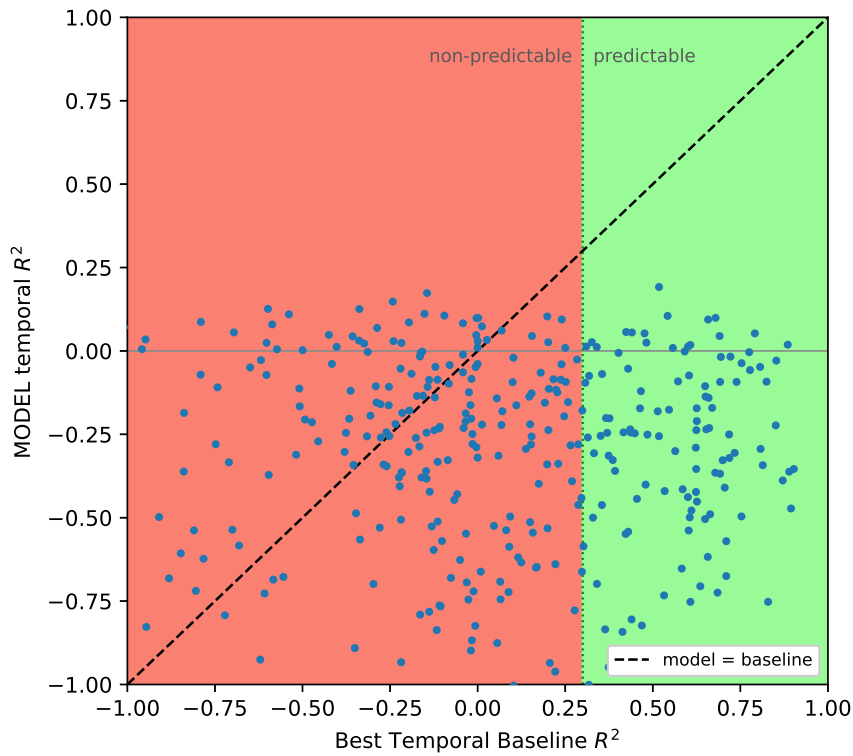


Figure 4.9: Temporal R^2 of the model (vertical axis) versus the best baseline R^2 (horizontal axis) for 400 held-out OOD fairlead windows. Each point is one 5-second future.

Temporal-difference metric reinterpretation: The temporal-difference MAE reported for C1 in Table 4.1, 45.4 N/step, quantifies the mean absolute error in consecutive-step tension differences. A flat-line predictor, which outputs the same tension value at every forecast step, scores approximately 26.8 N/step on this metric (averaged across the 21-node chain), because the true tension changes by a non-zero amount from step to step even in the absence of model prediction. The C1 model, at 45.4 N/step, thus scores worse than the flat-line predictor by a factor of approximately 1.7, confirming the absence of step-to-step temporal skill. The lower temporal-difference MAE of C1 relative to Stage-B BE1 (53.1 N/step) reflects that C1 produces a smoother output rather than that it tracks step-to-step changes more accurately.

Peaks and the temporal blind spot: Figure 4.10 shows the predicted versus true per-node peak tension, coloured by node count N , across all held-out test windows. The saturation region above approximately 9 kN, where predicted peaks cluster below the parity line, corresponds to the fraction of the tension distribution driven by the temporal oscillation. The model reproduces the quasi-static mean level accurately, reflected in a positive mean peak-tension bias of +197.5 N, but underestimates the peak excursions above that level, as these are determined by the instantaneous wave phase at the moment of maximum response, which the model does not track. The 17.6 % underprediction rate is therefore a direct consequence of the temporal blind spot characterised in this section. The denser scatter near the parity line at low tensions (quasi-static regime) contrasts with the systematic offset below the line above 9 kN, where wave-frequency oscillations drive the peaks. The figure displays a random subsample of test data points drawn from the prediction-dump run. The subsample underprediction rate (19.6 %) differs slightly from the authoritative full-test value (17.6 %, computed over 30 000 windows and reported in Table 4.1) because the subsample does not cover all sea states equally.

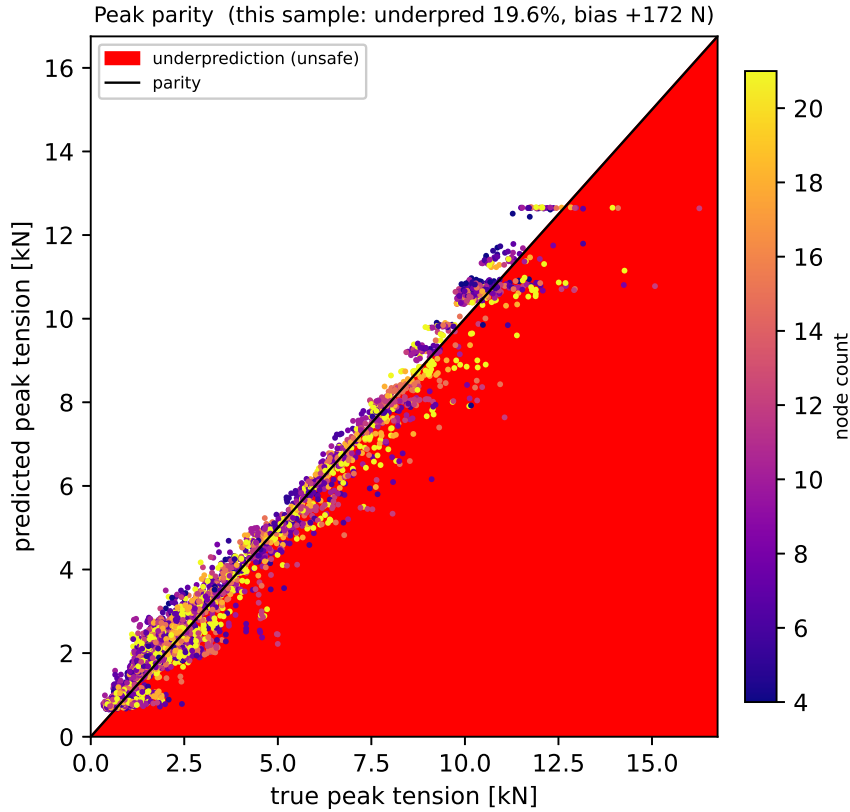


Figure 4.10: Predicted versus true per-node peak tension T^{peak} for a random subsample of test data points, coloured by node count N . The diagonal is the parity line, the shaded region marks the underprediction half-plane. Full held-out test metrics (30 000 windows): underprediction rate 17.6 %, bias +197.5 N (Table 4.1).

4.4.4. Why Spatial but Not Temporal: the Mechanism

The asymmetry between spatial and temporal performance is explained by three concurring factors.

Loss and variance structure (dominant): Under an averaged point loss over a dataset where wave phase is effectively random from window to window, the loss-minimising prediction for the temporal oscillation is its conditional mean, which is a flat line. Fitting the 98.5 % spatial and level components yields almost all of the available loss reduction, whereas fitting the 1.5 % temporal oscillation yields only a small and noisy gradient. The trained surrogate therefore allocates representational capacity to level and profile and outputs the conditional mean for the temporal component, even on the 31 % of windows where the oscillation is partially predictable from history.

Non-autoregressive decoder (amplifying): The temporal encoder uses the last LSTM hidden state only, compressing 25 seconds of history into a single 256-dimensional vector per node. The prediction head emits all 50 future steps at once from this fixed vector, without step-to-step feedback. The instantaneous wave phase is not preserved in the compression, so even the first forecast step (0.1 s ahead) is not anchored to the last observed value. The LSTM enlargement from Stage B (hidden size $128 \rightarrow 256$, one to two layers) improved level and position accuracy substantially (MAE_x reduced by 40–54 %), but had no measurable effect on the temporal R^2 of tension because the fundamental aggregation operation is unchanged, a sequence of 250 GAT embeddings is compressed into one vector and decoded in a single pass. This design was adopted deliberately in Section 3.1 to avoid autoregressive error accumulation over the horizon and to preserve the surrogate’s single-pass inference speed (Section 4.6). Chapter 7 revisits an autoregressive decoder specifically to recover the predictable temporal fraction identified here.

Input ceiling (bounding): The model receives sea-state statistics (H_s , T_p) rather than the incident wave elevation time series. For the approximately 69 % of test windows that are broad-band or multi-directional, the 5-second tension future is largely under-determined given only these statistics. No architecture can recover the phase information that is absent from the inputs. For the predictable 31 % of windows, where regular swell components encode the upcoming wave in the response history, the phase information is in principle available through the LSTM encoder, but the loss structure described above prevents the model from exploiting it.

These three factors act independently and compound. The loss structure drives the model toward the conditional mean regardless of how much temporal information is available, the non-autoregressive decoder discards the instantaneous phase that was present in the history, and the input statistics limit the achievable ceiling for the irregular majority of windows. The result is a surrogate that is excellent at reconstructing the spatial load field and sea-state level, and that outputs a smooth near-flat temporal signal whose amplitude represents the conditional mean rather than an oscillation prediction. The 1.5 % temporal sliver contains the peaks, and the temporal blind spot is the direct cause of the 17.6 % underprediction rate characterised in Section 4.4.3. The implications for architectural extensions are discussed in Chapter 5.

4.4.5. Comparison with a Persistence Baseline

At a 5-second horizon the mooring state is highly persistent, so a trivial forecaster that repeats the last observed frame across the full horizon is a demanding yardstick. Table 4.7 compares the C1 surrogate against this persistence baseline on the global R^2 of each target, pooled over 20 003 held-out test windows, and Figure 4.11 visualises the comparison. This pool is the subset of the 30 000-window test cap of Table 2.4 for which both the model prediction and the last observed history frame were retained, the latter being required to construct the persistence forecast. On tension, the design-relevant target, the surrogate exceeds persistence by +0.011, so the reported headline accuracy stands on its own against the trivial baseline. On the two position targets persistence is marginally ahead, by 0.004 and 0.003 respectively, precisely because the line is quasi-static over the 5-second horizon and a no-change prediction is almost exactly correct, which is the near-zero temporal position variance established in Table 4.6 rather than a deficiency of the model.

On the isolated temporal axis the surrogate is clearly and consistently ahead of persistence. Across the 400 fairlead-tension OOD windows of the predictability study the surrogate achieves a median temporal R^2 of -0.28 against -0.73 for persistence, and it outperforms persistence in 71.8 % of windows. Persistence falls below a flat-mean predictor in every one of the 400 windows, so copying the last observed tension value is worse than predicting the sea-state mean in every case examined. The surrogate, while also largely below the flat-mean predictor (Section 4.4.3), is systematically less wrong because it predicts the correct quasi-static level and spatial profile rather than anchoring to the last-observed instantaneous value.

Critically, the surrogate’s advantage over persistence is not uniform across sea states but grows monotonically with oscillation amplitude, the characteristic shown in panel (b) of Figure 4.11. In the lowest-amplitude quintile of the study (below 0.38 kN) the two methods are nearly tied, with the surrogate leading in 55 % of windows. From the second quintile onward the gap widens steadily. In the top am-

plitude quintile (above 0.83 kN) the surrogate leads in 83.8 % of windows. In the most energetic subset, the 57 windows with fairlead amplitude above 1.0 kN, the median persistence temporal R^2 falls to -0.95 while the surrogate holds at -0.09 , a gap of 0.86 in temporal R^2 . The physical reason is direct. At high amplitude the instantaneous tension at the end of the history window is far from the window mean, so copying it forward produces large errors over the full 5-second horizon, whereas the surrogate's quasi-static level prediction is close to the within-window mean regardless of where the last observed value happened to fall.

The persistence baseline is not, however, a usable surrogate. It requires the current full nodal state to be available, which is precisely the finite-element output that the surrogate is built to replace, and it cannot transfer to unseen sea states, node counts, or longer horizons. The comparison therefore confirms that the surrogate adds genuine value on the tension target at the global pooled level and, particularly in energetic sea states, on the temporal axis where the 5-second persistence assumption is most likely to be violated. The position metrics reflect a degenerate quasi-static regime in which no forecasting method improves on no-change, a fact that is fully explained by the near-zero temporal position variance established in Section 4.4.2. The implications for the surrogate's utility over longer forecast horizons, where persistence degrades rapidly and the trained model is expected to maintain its level-prediction advantage, are discussed in Chapter 5.

Table 4.7: Held-out global R^2 of the C1 surrogate versus a persistence baseline (last observed frame repeated across the $F = 50$ horizon), pooled over 20 003 test windows.

Target	C1 surrogate	Persistence	Δ (model – persist.)
x_{abs}	0.9957	0.9999	-0.004
z_{abs}	0.9953	0.9987	-0.003
Tension T	0.9800	0.9687	$+0.011$

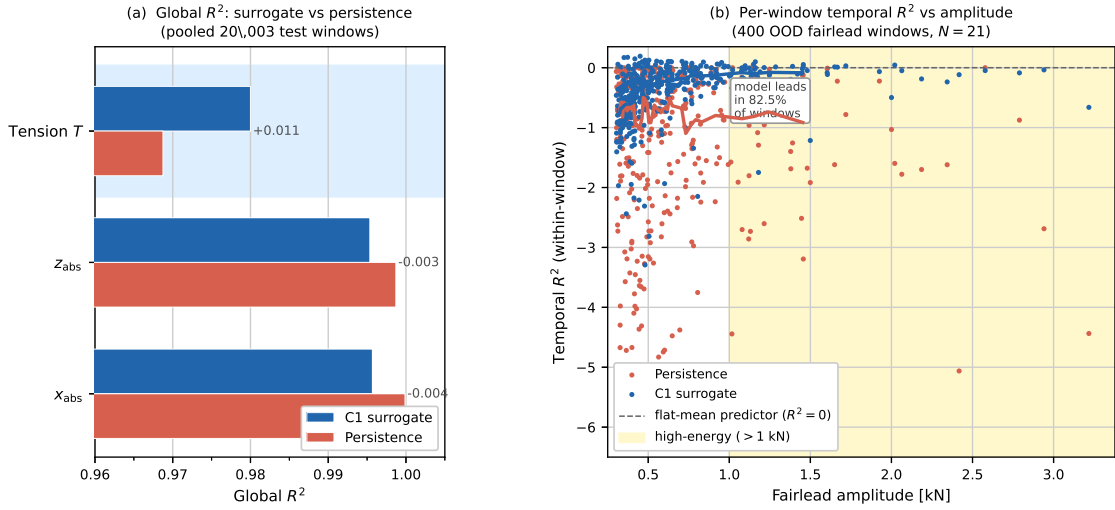


Figure 4.11: Comparison of the C1 surrogate against the persistence baseline (last observed frame repeated across the $F = 50$ horizon). (a) Global R^2 per target pooled over 20 003 test windows. (b) Per-window temporal R^2 at the fairlead versus fairlead oscillation amplitude, for 400 out-of-distribution windows. Rolling medians (solid lines) show that the surrogate's advantage over persistence grows monotonically with amplitude.

4.5. Multi-task Uncertainty Weighting

Chapter 3, Section 3.8.4 introduced a formulation that replaces the four hand-tuned physical-loss weights w_k with four learnable log-variance parameters $s_k = \log \sigma_k^2$, following the homoscedastic uncertainty weighting of Kendall et al. [45]. The resulting Stage-C2 investigation comprises three sequential variants, each motivated by a failure mode or limitation of its predecessor. The following three sub-

sections present the learned uncertainty dynamics, the failure modes and fixes, and the final accuracy-versus-safety comparison.

4.5.1. Learned Uncertainties and the Reproducible Term Ranking

Table 4.8 reports the learned $\sigma_k = \exp(\frac{1}{2}s_k)$ and the implied effective weight $1/(2\sigma_k^2)$ for each loss term, drawn from the training history of each C2 variant at its selected and converged epochs. All three C2 variants are selected by the maximum validation global R^2_{tension} , since the Kendall validation loss is dominated by the $\frac{1}{2}s_k$ regularisation terms and anti-correlates with model quality under σ -runaway, making it an unreliable selection criterion. All three rows in Table 4.8 are read from the converged tail of each run’s training history, so the σ values reflect the fully converged state rather than the R^2 -selected checkpoint. For C2a, the converged-tail values (ep 51) show the runaway endpoint and illustrate the failure mode most clearly.

Table 4.8: Learned uncertainties σ_k and effective weights $1/(2\sigma_k^2)$ for the four physical-loss terms across C2 variants, all read from the converged training-history tail of each run (ep 51 for C2a, ep 41 for C2b, ep 55 for C2c). Small σ implies a precisely fitted term and therefore a large effective weight. The three two-sided terms rank consistently as temporal $\succ$ physical MAE $\succ$ peak tension in all runs.

Variant / epoch	Temporal	Phys. MAE	Peak tension	Peak underpred.
<i>C2a (all-4 Kendall, one-sided hinge), ep 51 converged tail</i>				
σ_k	0.080	0.263	0.609	0.004 (runaway)
$1/(2\sigma_k^2)$	78.0	7.2	1.35	$\approx 27,800$ (runaway)
<i>C2b (3-term Kendall + fixed underpred.), ep 41 converged tail</i>				
σ_k	0.064	0.243	0.298	1.0 (fixed)
$1/(2\sigma_k^2)$	123	8.5	5.6	— (fixed at $w = 0.02$)
<i>C2c ($\tau = 0.9$ pinball, 4-term Kendall), ep 55 converged tail</i>				
σ_k	0.065	0.248	0.316	0.183 (plateau)
$1/(2\sigma_k^2)$	119	8.2	5.0	15.0 (plateau)

The three two-sided loss terms, temporal smoothness, physical MAE, and peak-tension magnitude, rank in the same order across all three C2 variants, specifically temporal $\succ$ physical MAE $\succ$ peak tension. This reproducibility, across runs with different loss formulations and different numbers of terms in the Kendall sum, constitutes a robust finding about the model’s uncertainty structure. The temporal-smoothness term receives the largest effective weight in every variant, indicating that the model can fit consecutive-step differences with the highest consistency, or equivalently that the task provides the most precise gradient signal for rate-of-change accuracy. The peak-tension magnitude term consistently receives the smallest effective weight, indicating that absolute peak-tension prediction carries the highest residual uncertainty relative to its scale. This ordering has a physical interpretation. The temporal-smoothness loss measures a short-range difference that is well constrained by the history window, while the peak-tension magnitude depends on the phase alignment of the wave train over the full future horizon, which is inherently less constrained.

Figure 4.12 illustrates the qualitative difference between a runaway trajectory and a stable convergent trajectory. In the left panel, for C2a with the one-sided hinge, $\sigma_{\text{underpred}}$ decreases monotonically from 0.72 to 0.004 with no floor, corresponding to an effective weight that diverges toward $+\infty$. In the right panel, for C2c with the $\tau = 0.9$ pinball, $\sigma_{\text{underpred}}$ decreases from 0.32 to a plateau around 0.18 by epoch 10 and remains stable, confirming that the two-sided formulation provides the irreducible noise floor required for Kendall convergence. All four C2c σ trajectories are overlaid.

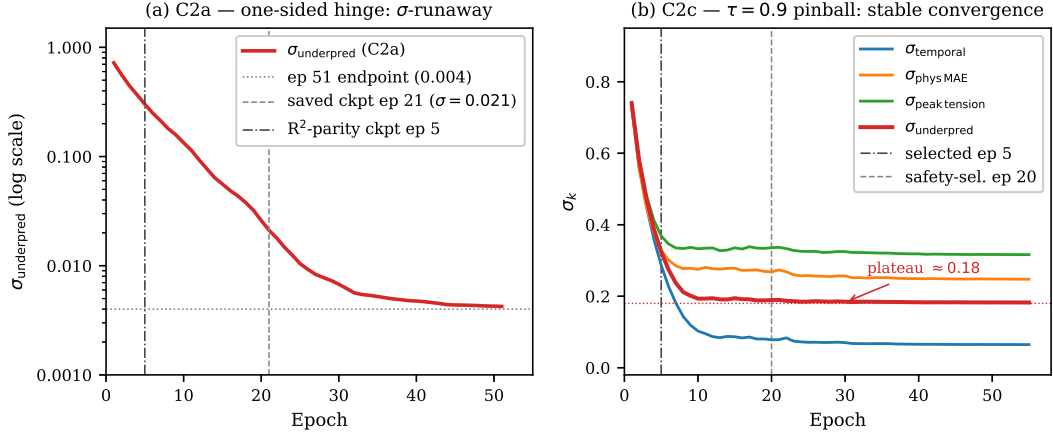


Figure 4.12: Evolution of the learned uncertainty σ_k for the peak-underprediction term. Left: C2a (one-sided hinge) showing σ -runaway. Right: C2c ($\tau = 0.9$ pinball) showing stable convergence to a plateau at ≈ 0.18 . Vertical markers indicate the R^2 -selected checkpoint (ep 5) and, for C2c, the safety-selected checkpoint (ep 20).

4.5.2. The σ -Collapse Failure Mode and the Pinball Fix

The three C2 variants trace a diagnostic ladder from a clean failure, through a corrected but still flawed design, to a working formulation. Two names are used below for the two sides of the same event. A σ -collapse is the learned uncertainty of a term falling without bound toward zero, and the resulting σ -runaway is the divergence of that term’s effective weight $1/(2\sigma_k^2)$ toward infinity. They are the same instability described from the parameter and from the weight, not two separate failure modes.

C2a (σ -collapse on the one-sided safety term): The initial formulation, run C2a, placed all four physical-loss terms inside the Kendall sum of Equation (3.38). The peak-underprediction term is a one-sided hinge, $\mathcal{L}_{\text{under}} = \max(T_{\text{true}}^{\text{peak}} - \hat{T}^{\text{peak}}, 0)$, which equals zero whenever the model overpredicts the peak. Because the model can drive $\mathcal{L}_{\text{under}}$ to zero simply by predicting a sufficiently large positive bias, the term has no irreducible noise floor. Without a floor, the Kendall equilibrium condition $s_k^* = \ln \mathcal{L}_k$ of Equation (3.39) has no solution, so $s_k \rightarrow -\infty$ and $\sigma_k \rightarrow 0$, creating a positive-feedback loop where a smaller σ_k assigns a larger effective weight to $\mathcal{L}_{\text{under}}$, which pushes the model to overpredict peaks more aggressively. The $\sigma_{\text{underpred}}$ trajectory in Table 4.8 and Figure 4.12 (left panel) confirm this runaway. The result is a model with no stable operating point. The selected R^2 -optimal checkpoint, epoch 5, is accurate but unsafe ($R^2 = 0.9785$, 39.1% underprediction, bias = -1.4 N), while the min-val-loss checkpoint (epoch 21) is over-conservative and inaccurate ($R^2 = 0.9336$, bias = $+2134$ N). The two checkpoints of the same run lie at opposite extremes of the accuracy-safety space, with no intermediate checkpoint satisfying both criteria simultaneously.

C2b (scale mixing between a fixed weight and Kendall weights): Run C2b addressed the C2a σ -collapse by removing the underprediction term from the Kendall sum and retaining it as a fixed penalty with $w_{\text{underpred}} = 0.02$, while the three two-sided terms remained Kendall-weighted. The three Kendall σ values converged without runaway, as shown in Table 4.8. However, the test underprediction rate at the selected checkpoint (epoch 11) was 50.0%, with a dangerous negative bias of -103 N. The cause is a scale-mixing incompatibility rather than a σ -collapse. The fixed weight of 0.02 was effective in Stage B and C1 because the other physical weights were of comparable magnitude (0.0025–0.005), making the underprediction penalty the dominant physical term. Inside C2b, the three Kendall terms self-calibrate to effective weights in the range 5.6–123 (converged tail, Table 4.8), making the fixed 0.02 the smallest physical penalty by a factor of 280–6150. The model therefore treats peak underprediction as a negligible objective, and the safety constraint collapses entirely. The failure is not in the σ dynamics themselves but in the design assumption that a fixed weight and auto-scaled weights can coexist at comparable magnitudes.

C2c ($\tau = 0.9$ pinball reformulation): Run C2c returns all four physical-loss terms to the Kendall sum, but replaces the one-sided hinge with a $\tau = 0.9$ pinball (quantile) loss for the underprediction

term. Following Koenker and Bassett [50], the pinball function is

$$L_\tau(d) = \max(\tau d, (\tau - 1)d), \quad d = \frac{T_{\text{true}}^{\text{peak}} - \hat{T}^{\text{peak}}}{\bar{\sigma}_T},$$

which penalises underprediction ($d > 0$) $9\times$ more than overprediction ($d < 0$), while remaining two-sided and strictly positive for any nonzero d . By the consistency result of Gneiting [51], the optimal point forecast under this loss is the τ -quantile of the true peak-tension distribution, a conservative upper bound appropriate for structural safety assessment. The two-sided character of the pinball provides the irreducible noise floor that the one-sided hinge lacks, as $L_\tau \geq 0$ with equality only at a perfect quantile predictor, which cannot be achieved by a finite model. The $\sigma_{\text{underpred}}$ trajectory under C2c (Table 4.8, Figure 4.12, right panel) confirms the fix. $\sigma_{\text{underpred}}$ decreases from 0.32 at epoch 5 to a stable plateau of approximately 0.18 by epoch 10, and remains there through epoch 55. All four σ values converge, and the clean four-way ranking (temporal $\succ$ peak underpred. $\succ$ physical MAE $\succ$ peak tension) becomes readable for the first time.

4.5.3. Accuracy versus Safety

Table 4.9 compares the four Stage-C runs on the held-out test set at each run’s selected checkpoint, all selected by the maximum validation global R^2_{tension} . For C1, this coincides with the minimum-validation-loss checkpoint (epoch 12), because the standard SmoothL1 objective tracks model quality monotonically. For the C2 variants, the Kendall validation loss is dominated by the $\frac{1}{2} \log \sigma_k$ regularisation terms and anti-correlates with quality under σ -runaway, making it meaningless as a selection criterion. R^2 -based selection is therefore mandatory for all C2 runs. For C2a this gives epoch 5, while C2b selects epoch 11 and C2c selects epoch 5.

Table 4.9: Held-out test performance of all Stage-C runs, each at its own R^2 -selected checkpoint. Bold denotes the best value in each column, where best means the largest R^2 , the smallest underprediction rate, and the smallest MAPE and temporal-difference MAE. The bias column is not bolded, since the preferable value there is a moderate positive margin rather than an extremum.

Run (sel. ep)	R^2_{tension}	$R^2_{x_{\text{abs}}}$	$R^2_{z_{\text{abs}}}$	Underpred. [%]	Bias [N]	MAPE [%]	Temp. Δ MAE [N/step]
C1 (ep 12)	0.9800	0.9957	0.9954	17.6	+197.5	9.06	45.4
C2a (ep 5)	0.9785	0.9898	0.9893	39.1	−1.4	11.2	37.3
C2b (ep 11)	0.9815	0.9956	0.9961	50.0	−103	9.5	25.8
C2c (ep 5)	0.9810	0.9880	0.9928	42.5	−42.6	10.2	40.0

Under R^2 -based selection, C1 is the only configuration that simultaneously achieves a competitive R^2_{tension} and a safe positive peak bias. The three Kendall variants reach comparable or marginally higher R^2_{tension} values at their R^2 -optimal checkpoints, but all three underprediction rates exceed 39 %, with negative or near-zero biases, making them unsafe in the sense that the model systematically underestimates the tension peaks that govern fatigue and extreme-load design. The root cause is a structural timing mismatch. The R^2 -optimal checkpoint is reached early in training (epochs 5–11), before the Kendall safety weight has grown to a level that substantially influences the peak-tension predictions. C1 avoids this mismatch because its hand-tuned $w_{\text{underpred}} = 0.02$ front-loads the safety emphasis from epoch 1.

Ablating the underprediction penalty: The claim that the hand-tuned penalty is the source of C1’s safe operating point can be tested directly by removing it. A controlled ablation sets $w_{\text{underpred}}$ to zero while holding the architecture, dropout, random seed, and the three remaining physical-loss weights fixed at their C1 values, so the run is identical to C1 except for the disabled penalty. Table 4.10 reports the outcome. Removing the penalty improves the symmetric-accuracy metrics marginally, raising the global tension R^2 from 0.9800 to 0.9817 and lowering the tension MAE from 230.9 N to 211.7 N, because the prediction is no longer biased away from the conditional mean that minimises the data loss. The safety metrics, in contrast, invert. The mean peak-tension bias flips from +197.5 N to −149.8 N, and the underprediction rate rises from 17.6 % to 56.7 %, so the model now underestimates the peak load on the majority of nodes. This confirms that the penalty is the single lever responsible for C1’s conservative

safety margin, purchased at a negligible accuracy cost of 0.0017 in R^2 , and it explains why the R^2 -selected Kendall variants, in which the equivalent safety weight has not yet matured, are unsafe. The accuracy ceiling of Section 4.4.1 is likewise unmoved, the tension R^2 remaining within 0.002 of the defended value, which reinforces that the ≈ 0.98 plateau is set by the architecture and inputs rather than by the loss weighting.

Table 4.10: Effect of removing the peak-underprediction penalty from C1. The two runs are identical except for the penalty weight ($w_{\text{underpred}} = 0.02$ versus 0). Held-out test set, 30 000 windows. Bold denotes the preferable value in each row, so the accuracy rows favour the ablation and the safety rows favour C1.

Metric	C1 (penalty on)	No-underpred. (off)
Global R^2_{tension}	0.9800	0.9817
Global $R^2_{x_{\text{abs}}}$	0.9957	0.9974
Global $R^2_{z_{\text{abs}}}$	0.9954	0.9963
Tension MAE [N]	230.9	211.7
Peak-tension bias [N]	+197.5	−149.8
Peak underpred. rate [%]	17.6	56.7
Peak-tension MAE [N]	332.5	241.4
Tension MAPE [%]	9.06	9.00

Spatial structure of the penalty: Figure 4.13 shows the per-node peak-tension profile for both models on a representative out-of-distribution window and reveals that the penalty acts non-uniformly along the line, because the load magnitude it is calibrated against varies by an order of magnitude from anchor to fairlead. In the low-tension grounded section the penalty drives a large relative overprediction of the peak, approximately +0.86 kN on this window against a true peak of order 1.9 kN, which is the lower-line overprediction that motivated the ablation. Removing the penalty brings the grounded-section peak onto the true value, to within −0.14 kN, but it simultaneously leaves the safety-critical fairlead peak underpredicted by −0.51 kN. A spatially uniform penalty therefore over-corrects the grounded section, where the fixed absolute margin is large relative to the small local load, while under-covering the fairlead, where the largest and most design-relevant loads occur. This observation motivates a spatially adaptive safety margin, wider in the low-load grounded region and tighter at the fairlead.

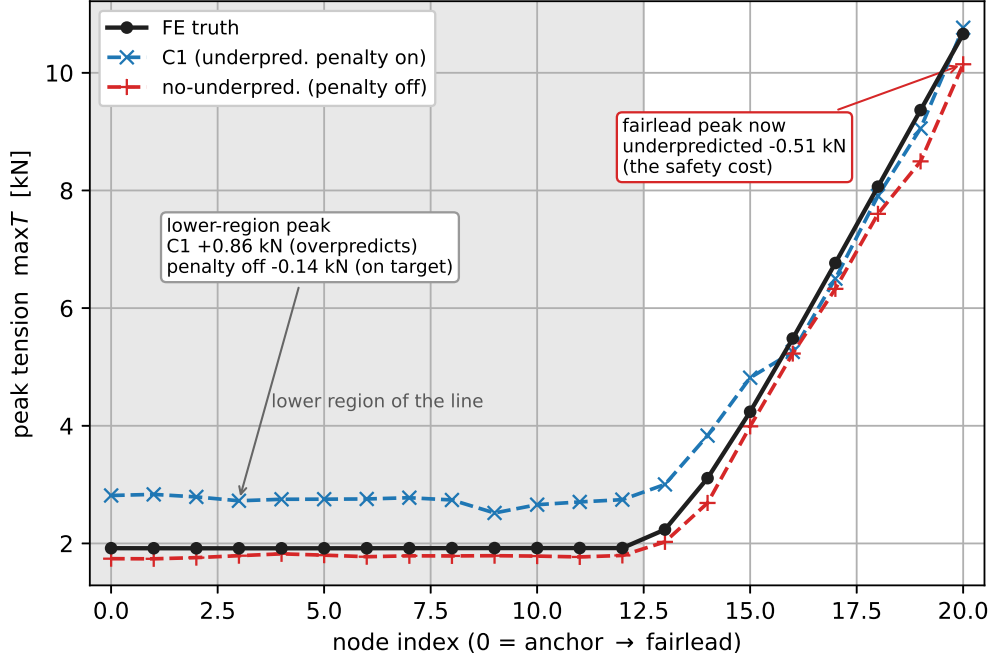


Figure 4.13: Per-node peak tension (maximum over the 5 s forecast window) for the defended C1 model (penalty on) and the ablation (penalty off), against the finite-element truth, for a representative out-of-distribution window (location 11, $N = 21$).

The C2c pinball formulation enables a different selection strategy. When the checkpoint is chosen on a safety criterion rather than an R^2 criterion, the pinball weight at the selected epoch is substantially larger, and the model recovers a safe operating point without hand-tuning. Table 4.11 presents this safety-selected comparison.

Table 4.11: C2c safety-selected checkpoint (epoch 20) versus C1. Safety selection identifies the epoch at which the pinball weight has matured sufficiently to constrain underprediction, recovering a positive peak bias comparable to C1's at a modest accuracy cost. Bold denotes the better value in each column, with the bias column left unbolded because both entries are safely positive.

Run	R^2_{tension}	Underpred.[%]	Bias [N]	Peak MAE [N]	MAPE [%]	Temp. Δ MAE [N/step]
C1 (ep 12)	0.9800	17.6	+197.5	332.5	9.06	45.4
C2c (ep20)	0.9752	15.5	+235.7	323.4	11.3	27.3

At epoch 20, the C2c $\sigma_{\text{underpred}}$ has stabilised at approximately 0.18 and the effective underprediction weight (≈ 15) is large enough to constrain the model. The resulting test performance has a peak underprediction rate of 15.5 % and a safe positive bias of +235.7 N, both comparable to C1. The R^2_{tension} at this checkpoint (0.9752) is 0.0048 below C1, reflecting the cost of safety-based selection relative to R^2 -based selection. The temporal-difference MAE at epoch 20 (27.3 N/step) is substantially lower than C1's (45.4 N/step), consistent with the finding in Table 4.9 that large Kendall temporal-smoothness weights improve rate-of-change accuracy. Neither configuration Pareto-dominates the other. C1 achieves higher R^2_{tension} and lower MAPE, while C2c-ep20 achieves a marginally lower underprediction rate, a larger conservative bias margin, and better temporal-difference accuracy. The principal contribution of C2c is therefore methodological. It demonstrates that the τ -quantile pinball loss, when used as the safety term within the Kendall framework, provides a principled mechanism for trading accuracy against safety via the choice of selection criterion, without requiring manual weight tuning.

The wave-frequency temporal component accounts for only 1.5 % of the total tension variance, so gains in rate-of-change accuracy do not register as meaningful improvements in the primary quality metric. The model's effective strategy of predicting the conditional mean at the temporal scale is rational under the SmoothL1 data loss, and it remains rational even when a large temporal-smoothness Kendall weight

is applied, because the spatial and level components dominate the loss gradient by orders of magnitude. The Kendall weighting therefore produces a better temporal metric at the cost of slightly worse spatial performance, confirming that the temporal limitation is a consequence of the loss structure and task variance distribution rather than a lack of capacity.

4.6. Computational Time

One of the primary engineering motivations for a surrogate model is the reduction of wall-clock time per sea-state evaluation. The GNL-FEM solver described in Chapter 2 integrates the semi-discrete equations of motion at every time step with an iterative Newton–Raphson loop. Per-case wall times are not available from the present run set. This section therefore reports the training cost of the surrogate and discusses inference cost qualitatively.

Table 4.12 reports the on-cluster training cost for the two Stage-C models. Both runs were executed on a single NVIDIA A100 (80 GB HBM2e) with 16 CPU cores and 64 GB RAM on the DelftBlue HPC cluster [47].

Table 4.12: Training cost of the two Stage-C surrogate models on a single NVIDIA A100 GPU.

Model	Epochs	Training time [min]	Time per epoch [s]	Peak VRAM [GB]
C1 (BE1, fixed weights)	42	48.8	69.7	5.48
C2c (Kendall + pinball)	55	65.7	71.7	5.48

C1 ran 42 epochs before the early-stopping criterion was satisfied, with the best checkpoint selected at epoch 12. C2c ran 55 epochs under an early-stopping patience extended from the Stage-C default of 30 (Table 3.8) to 50 epochs, allowing the learned uncertainties to converge fully, with the best checkpoint at epoch 5. Per-epoch cost is nearly identical between the two runs (69.7 vs 71.7 s), confirming that the Kendall uncertainty weighting adds negligible computational overhead over the fixed-weight formulation. The longer total training time for C2c reflects its larger epoch count rather than any per-step cost increase. Peak VRAM is identical at 5.48 GB for both runs, well within the 80 GB A100 capacity, confirming that GPU memory is not a bottleneck.

Once trained, surrogate inference replaces the GNL-FEM solve with a single forward pass, comprising a GATv2 aggregation over the graph, an LSTM readout over the 250-step history, and a two-layer MLP projection to all 50 future steps. All three operations reduce to batched matrix multiplications with no iterative convergence loop, and a single forward pass on the A100 processes a full batch of windows in milliseconds. The GNL-FEM solver, by contrast, advances 13 800 time steps per case and solves a nonlinear system by Newton–Raphson iteration at each one (Chapter 2). Because the per-case solver wall time was not recorded for the present dataset, no speed-up ratio is quoted here. The structural difference between a one-shot chain of matrix multiplications and an iterative nonlinear time integration nonetheless implies a substantial advantage per prediction window, and quantifying it on matched hardware is left for future work.

4.7. Summary

This chapter has presented the Stage-C results along five axes.

- **Engineering result (C1):** The BE1 architecture trained on 300 cases per location achieves a held-out test $R^2_{\text{tension}} = 0.9800$ with a safe positive peak bias (+197.5 N) and a peak underprediction rate of 17.6 %. The model generalises across all ten node counts without per-topology fine-tuning, and the test performance meets or exceeds the validation performance at every node count, confirming strong topology and OOD generalisation (Section 4.3.1).
- **Spatial load-field reconstruction:** The surrogate reconstructs the anchor-to-fairlead tension profile and full line geometry accurately for any topology and at OOD locations. The global R^2_{tension} stays between 0.9761 and 0.9863 across all ten discretisations, and the global R^2 for both position channels exceeds 0.99, confirming that the quantities most directly relevant to quasi-static design assessment are well represented (Sections 4.3.1 and 4.3.2).

- **Spatial–temporal capability boundary:** A variance decomposition of the held-out test set shows that $\geq 98.5\%$ of the total tension variance is the spatial profile and sea-state level, while the wave-frequency temporal component accounts for only 1.5% . The surrogate reproduces the dominant components accurately and outputs the conditional mean for the temporal component. Of the 5-second futures, 31% are predictable by a history-only baseline, and the model captures none of that predictable variance (median temporal $R^2 = -0.29$). This is a characterised capability boundary arising from the loss structure, the non-autoregressive decoder, and the sea-state-statistics input, not a hidden failure. The missed 1.5% temporal fraction contains the wave-frequency peaks, making it the direct cause of the 17.6% underprediction rate (Section 4.4).
- **Architectural ceiling:** The $R^2_{\text{tension}} \approx 0.98$ ceiling is architectural and task-limited, not data-limited or regularisation-addressable. A six-fold data increase delivers only $+0.002$ and a dropout increase worsens performance. This is a ceiling on what the current single-shot decoding strategy can extract from the given inputs, not evidence that a larger model is needed, the compact architecture already reaches it, and parsimony is preserved (Section 4.4.1). Reframed by the variance decomposition, the ceiling is a saturation of the predictable (spatial and level) variance rather than a failure. The implications are discussed in Chapter 5.
- **Multi-task uncertainty weighting:** The Kendall uncertainty framework replaces four hand-tuned weights with jointly learned task uncertainties. Three variants expose two distinct failure modes (a σ -collapse on the one-sided safety term in C2a, and scale mixing between fixed and Kendall weights in C2b) and arrive at a working formulation (C2c) in which the $\tau = 0.9$ pinball loss provides the irreducible noise floor required for Kendall convergence. The learned σ ranking, temporal $\succ$ physical MAE $\succ$ peak tension, is reproducible across all three variants with two-sided terms and constitutes an interpretable characterisation of the model's fitting precision. Safety-based checkpoint selection in C2c recovers C1's safety operating point without manual weight tuning, at a modest R^2 cost of 0.0048 .

5

Discussion

This chapter interprets the Stage-C results of Chapter 4. It asks what the surrogate is, what it is for, and where it stops working. The discussion covers physical consistency, topology generalisation, the boundary of the model's capability, and the lesson from the uncertainty-weighting experiment.

5.1. A Spatial Load-Field Surrogate, Not a Forecaster

The headline accuracy must be read alongside the variance decomposition of Section 4.4.2. The global $R^2_{\text{tension}} = 0.98$ certifies that the model reconstructs the spatial tension profile and the per-sea-state level. Together these carry at least 98.5 % of the variance. The figure does not certify wave-by-wave temporal tracking.

Two kinds of value are easily confused. The first is forecasting value, which means predicting how the state evolves. At a 5-second horizon this value is low, because the system barely decorrelates over that interval. This is why a trivial persistence baseline is competitive on the temporal axis (Section 4.4.5). The second is surrogate value, which means emulating the finite-element solver. The surrogate produces the full nodal load field for a condition, across topologies and locations, at a fraction of the cost. Persistence has no value here, because it needs the very state the surrogate is meant to replace. Judged as a surrogate, the model delivers a validated, transferable load field. Judged as a 5-second forecaster, it is unremarkable. The contribution of this work is the former, and it is the axis on which the existing graph surrogates for offshore structures are silent. Jiang and Chen [23] recover a distributed stress field but for a fixed support structure, while Yung et al. [27, 28] model the mooring system only through lumped fairlead reactions, so neither yields the along-line load distribution that this surrogate produces.

5.2. Physical Consistency and Peak Safety

The spatial predictions are physically consistent. The model reproduces the catenary geometry, the seabed touchdown transition, and the anchor-to-fairlead tension gradient (Section 4.3.2). The tension R^2 stays between 0.976 and 0.986 across every discretisation and the position R^2 is above 0.99. These are the quantities a quasi-static design assessment needs. The safety behaviour is also sound. The model carries a positive mean peak-tension bias of +197.5 N. On average it overestimates the peak load rather than underestimating it. A 17.6 % underprediction rate nevertheless remains, and it is the direct expression of the temporal blind spot, because the peak excursions live in the 1.5 % wave-frequency component that the model returns as its conditional mean (Section 4.4.3). The safety limitation and the temporal limitation are the same phenomenon seen through two metrics.

5.3. Topology Generalisation

The principal architectural result concerns generality. A single set of weights serves every discretisation from $N = 4$ to $N = 21$ nodes and both unseen deployment locations. Held-out test accuracy meets

or exceeds validation accuracy at every node count (Section 4.3.1). The mooring surrogates in the literature are instead trained for one fixed configuration, whether they predict a scalar fairlead quantity or, as in the spectral graph model of De Melo Costa et al. [26], classify the state of a fixed line layout. Such models must be retrained when the discretisation or the site changes. The mixed-topology graph formulation removes that constraint. This is what turns the model into a reusable building block rather than a single-case fit. The generality costs little accuracy, with tension R^2 declining only mildly from $N = 4$ to $N = 21$. This decline reflects the finer spatial gradients a denser chain must resolve, not a failure of transfer.

5.4. The Capability Boundary and Its Two Regimes

The most useful outcome of the analysis is a clear statement of where the model stops working and why. The temporal limitation splits into two regimes (Section 4.4.3). About 31 % of 5-second futures are recoverable from the response history alone. A simple single-harmonic frequency extrapolation of the past tension already explains them, yet the model leaves this signal unused. An averaged point loss over randomly phased windows is minimised by the conditional mean, and the non-autoregressive decoder discards the instantaneous phase in the history. This 31 % is addressable by an architectural and loss-function change, not by new measurements. The remaining 69 % is genuinely irreducible from the present inputs. The model is given sea-state statistics, not the incident wave train, so the 5-second phase is physically under-determined. Closing it requires supplying the future wave forcing or replacing the point prediction with a probabilistic output. Naming this split converts the temporal result from a hidden weakness into a characterised boundary. It also defines the two extension paths taken up in Chapter 7.

5.5. What the Learned Uncertainties Reveal

The multi-task uncertainty experiment (Section 4.5) yields a methodological lesson, not an accuracy gain. The learned σ_k rank the loss terms by how precisely each can be fitted. The order is reproducible across every variant, temporal $\succ$ physical MAE $\succ$ peak tension. The ranking is informative and it exposes why automatic weighting is hard to apply here. The Kendall framework up-weights the term it fits most precisely, which is the temporal-smoothness term. That term over-smooths the output and flattens the very peaks that matter. This is the opposite of what a safety objective demands. The hand-tuned C1 weights instead front-load the hardest term, peak underprediction, from the first epoch. This is why C1 alone is both accurate and safe under R^2 -based selection.

The corrected pinball formulation is still valuable. A two-sided $\tau = 0.9$ quantile loss restores the noise floor that the one-sided hinge lacked (Section 4.5.2). Under safety-based checkpoint selection, it then recovers C1's safe operating point automatically, at a modest accuracy cost (Section 4.5.3). The large learned temporal-smoothness weight in C2c also improves the temporal-difference metric, lowering the rate-of-change MAE from C1's 45.4 N/step to 27.3 N/step at the safety checkpoint. This gain reflects a smoother prediction rather than recovered wave-phase tracking, consistent with the loss-structure argument of Section 5.4. No Kendall variant Pareto-dominates C1, and C1 therefore remains the defended model. The uncertainty-weighting ladder stands as a transferable result for safety-asymmetric losses.

5.6. Limitations

Several limitations frame these results. The tension accuracy saturates near $R^2 = 0.98$. This ceiling is architectural rather than data-limited. A six-fold data increase and an added dropout both failed to lift it (Section 4.4.1). The model also predicts at the same node set it receives as input. It therefore does not yet perform sparse-to-dense reconstruction. Its inputs are response history and sea-state statistics, not the incident wave field. This bounds the achievable temporal forecast. Finally, the speedup over the finite-element solver is argued structurally, not measured on matched hardware. None of these undermines the core result, a transferable spatial load-field surrogate with a characterised capability boundary. Each maps directly onto a recommendation in Chapter 7.

6

Conclusion

Chapter 1 posed one main research question, addressed through four sub-questions, asking whether a graph-attention LSTM architecture can be formulated as a topology-general, safety-conscious finite-element surrogate for offshore mooring lines. This chapter answers each sub-question against the results of Chapters 4 and 5, returns to the main question, and states the contribution of the thesis.

6.1. Answering the Research Questions

SQ1 (Graph representation and generalisation): A mooring line is represented as a bidirectional chain graph whose nodes and edges carry static geometric and mechanical features alongside dynamic motion, tension, and contact features (Sections 3.2 and 3.3). Trained under a mixed-topology scheme, one set of weights serves every discretisation from $N = 4$ to $N = 21$ nodes and both environmental locations withheld from training, with held-out test accuracy meeting or exceeding validation accuracy at every node count (Section 4.3.1). The mild decline in tension R^2 over that range reflects the finer gradients a denser chain must resolve, not a failure of transfer (Section 5.3).

SQ2 (Spatial reconstruction and its meaning): The surrogate reconstructs the spatial field with a global R^2_{tension} of 0.98, holding between 0.976 and 0.986 across every discretisation and above 0.99 for both position channels (Sections 4.3.1 and 4.3.2). The variance decomposition of Section 4.4.2 identifies exactly what this figure certifies, the spatial profile and the per-sea-state level, which together hold at least 98.5 % of the variance, and not the remaining 1.5 % wave-frequency fluctuation, which the model returns as a near-flat conditional mean (Section 4.4.3).

SQ3 (Safety-aware objective): Four soft penalty terms expressed in physical units bias the fixed-weight objective toward safe peak-tension prediction, giving a positive mean bias of +197.5 N and an underprediction rate of 17.6 % (Section 5.2). A Kendall multi-task weighting scheme can replace these hand-tuned weights, but only once the one-sided underprediction hinge is reformulated as a two-sided $\tau = 0.9$ pinball loss, since the original hinge instead drove its own learned uncertainty to collapse (Section 4.5.2). Even corrected, it does not surpass the hand-tuned objective, recovering a comparable safe operating point only under safety-based checkpoint selection and at a small accuracy cost (Section 4.5.3). Automatic weighting is therefore possible here, but not yet advantageous.

SQ4 (Value and consistency): The surrogate reproduces the catenary geometry and the seabed touchdown transition (Section 4.3.2), and it exceeds a trivial persistence baseline on the headline tension accuracy, 0.9800 against 0.9687, and on the temporal axis, where across the 400 oscillating fairlead windows of the predictability study its median window R^2 of -0.28 beats persistence's -0.73 in 71.8 % of windows (Section 4.4.5). Persistence wins only on the quasi-static positions, where there is almost no temporal variance to forecast. Relative to the finite-element solver it is meant to emulate, the surrogate's value lies not in forecasting fidelity but in reproducing the spatial load field, across topologies and locations, at a fraction of the cost (Section 5.1).

The main research question: Together these answers show that a graph-attention LSTM architecture can be formulated as a full-field mooring surrogate that generalises across discretisations and locations and that is, on average, conservative with respect to peak tension. That conservatism is a property of the training objective rather than a guarantee, since 17.6 % of peaks are still underpredicted, for the same temporal reason identified in SQ2. The main research question is therefore answered positively for the spatial, cross-topology surrogate problem it targets, with the honest qualification that conservatism at every individual peak is not yet achieved.

6.2. Contributions

Three contributions follow. The first is a validated, topology-general load-field surrogate, one trained model spanning $N = 4$ to $N = 21$ nodes and generalising to unseen locations, in place of the single-configuration models common in the literature. The second is a safety-aware training methodology, the fixed-weight physical-penalty formulation together with the diagnosis of, and pinball-based fix for, uncertainty-weighting collapse on an asymmetric safety term. The third is a rigorous characterisation of what the surrogate's accuracy actually certifies, a spatial-versus-temporal variance decomposition and its predictability-baseline protocol, turning an implicit limitation into a stated, quantified boundary. None of the three rests on the model being a wave-by-wave forecaster of mooring tension, a framing this thesis deliberately set aside once the decomposition showed where the surrogate's value, and its limit, actually lie. The two research directions of Chapter 7 extend it along exactly that boundary.

Recommended Future Work

The model defended in this thesis is a validated, topology-general surrogate for the spatial mooring load field, with a capability boundary that Chapter 5 characterised precisely. That boundary is not a list of unrelated faults. It resolves into two limitations, one on each axis of the problem. Along the spatial axis the model predicts at the same node set it is given, so it cannot yet reconstruct a dense field from sparse sensing. Along the temporal axis it returns the conditional mean of the 5-second future, so it does not resolve the wave-frequency tension history (Section 5.6). Each limitation defines a self-contained research study. Both studies keep the GAT+LSTM backbone and the safety-aware loss formulation developed here, and change only the decoder and the training objective. The two studies are symmetric. The first holds the time window fixed and degrades the spatial input, measuring how accuracy falls as the line is sensed more sparsely. The second holds the spatial field and degrades the temporal horizon, measuring how accuracy falls as the forecast reaches further ahead.

7.1. Reconstructing the Full Load Field from Sparse Motion Sensing

This study addresses a physical problem distinct from the one posed in the rest of this thesis and from the second study below. It is not a prediction in time. Given the horizontal and vertical motion of the line over a time window, it predicts the tension at the same instants, so input and output share the time axis and the task is a spatial reconstruction rather than a forecast. The study proceeds in two stages. It first performs the reconstruction at matched resolution, predicting at the same node set it receives, across the full range of topologies used in this thesis. It then relaxes that constraint and attacks the super-resolution problem, reconstructing the tension at more nodes than are supplied as input.

The limitation it removes: The present model couples its input and output node sets. It receives the dynamic state at N nodes and predicts tension at the same N nodes, so it cannot infer the response at locations it does not already observe. The first study reframes the surrogate as a spatial reconstruction map. The input is the co-temporal motion, the horizontal and vertical displacement histories, measured at N_{in} nodes. The output is the tension at N_{out} nodes over the same time window. It mirrors a realistic monitoring scenario in which a line carries a few motion sensors but no load cells, and the full tension distribution must be inferred from the measured kinematics.

Why accuracy is not the contribution: Tension is, to leading order, a function of the instantaneous line geometry and its kinematics. A co-temporal motion-to-tension map is therefore a sharp, well-conditioned problem, and high accuracy is expected. The contribution of this study is not to be more accurate than the fixed-configuration recurrent baselines, which already perform well on the single fairlead quantity they are built to predict. It rests instead on three axes that those baselines do not address.

1. **Full-line field:** tension at every node from anchor to fairlead, not only the fairlead value reported by single-output surrogates.
2. **Topology generalisation:** one weight set for every discretisation from $N = 4$ to $N = 21$ nodes, already demonstrated in Section 4.3.1, whereas conventional surrogates need a separate model per configuration.
3. **Calibrated conservative peaks:** a $\tau = 0.9$ pinball objective biases the peak estimate safely upward. This axis is strongest in a reconstruction setting, because the sharp co-temporal map admits a conservative margin without sacrificing the accuracy a forecasting task would lose.

Spatial super-resolution: The decisive extension is to decouple N_{in} from N_{out} and sweep them independently. Reporting tension accuracy and peak safety over the full $N_{\text{in}} \times N_{\text{out}}$ grid yields a single diagnostic surface. Its diagonal recovers the topology generalisation already shown. Its upper triangle, where $N_{\text{in}} < N_{\text{out}}$, is the genuinely new capability, reconstructing the dense 21-node field from as few as four to six sensors. This is the deployment case of greatest industrial value, supporting sensor-placement studies and digital twins with limited instrumentation. The current prediction head cannot do this, because it is sized to the input node count. The required change is a query-based decoder. The GAT+LSTM encoder processes the sparse input nodes as before, and a decoder places N_{out} query positions at fixed arc-length stations that attend to the encoded node features and emit a prediction at each station. The study requires no additional simulation, since each 21-node ground-truth case can be subsampled to form a sparse input and scored against the full field.

7.2. Forcing-Free Forecasting of Future Tension

This study keeps the same physical problem as the rest of this thesis. It forecasts the future response of the line from its past, the setting used throughout this work. What changes is the objective. The study targets the temporal bottleneck identified in Chapter 5 and asks how far into the future the tension and position time series can be tracked before the forecast decays to the conditional mean. It does so under a positive bias on the predicted tension, so that whatever error remains is a safe overprediction of the peak load rather than a dangerous underprediction.

The limitation it removes: The temporal blind spot. The defended model returns a near-flat conditional mean over the 5-second future (Section 4.4.3), and the peak excursions that govern safety live in the unresolved wave-frequency component. The second study confronts this axis directly. The model sees only the response history, with no incident wave forcing and no future information, and predicts the future tension over a horizon F . Real-time monitoring never has access to the future forcing, so a forecaster that requires it is not deployable. The task is well-posed because Section 4.4.3 established that the recoverable part of the future is already present in the response history. A simple single-harmonic frequency extrapolation of the past tension recovers about 31 % of oscillating windows, which shows the missing skill is a loss-and-decoder problem on that fraction rather than a missing-input problem.

Accuracy is not the primary metric here: Unlike the reconstruction study, this study is not judged by absolute accuracy, and its R^2 values are not comparable with those of Section 7.1. The reconstruction task is a sharp co-temporal map on which high accuracy is expected, whereas forecasting operates under the temporal ceiling identified in Chapter 5, where most of the 5-second future is physically underdetermined from the present inputs. The model's value lies in how far into the future it can trace the tension time series from response history alone, and in the calibrated conservative envelope it sustains over that horizon, one that deliberately overpredicts to guard against dangerous underprediction.

The headline contribution, a degradation curve: The central deliverable is the forecast skill as a function of horizon, specifically how far into the future the model can trace the tension time series before the prediction decays to a flat conditional mean. Plotting tension R^2 against F traces this degradation curve and locates the horizon at which meaningful tracking ends. The recoverable fraction is itself horizon-dependent. Broadband sea states decorrelate quickly, while narrow-band swell remains predictable further out, so the deliverable is a curve of both the predictable fraction and the model's

capture of it against horizon, not a single number. Few mooring surrogates report this characterisation, and the honest limit is itself the contribution.

The architectural change: Recovering the predictable fraction requires two coupled changes, neither sufficient alone. The non-autoregressive head, chosen in Chapter 3 for a fixed short horizon, compresses the history into a single hidden state and discards the instantaneous phase. It is replaced by an autoregressive sequence decoder that reads the full LSTM sequence and rolls the forecast out step by step, so each step is anchored to the previous one. The averaged point loss is augmented by a term that no longer rewards the conditional mean, a spectral or derivative penalty, or a distributional objective. The physical loss terms and the Kendall uncertainty weighting carry over unchanged. The decoder gives the model the capacity to express an oscillation, and the loss gives it the incentive to.

The second contribution, a calibrated envelope: A conservative forecaster invites a specific trap. A model with little temporal skill predicts near the mean, so a blanket upward bias degenerates into the conditional mean plus a fixed margin, which is indistinguishable from persistence plus a safety factor. The defensible form is a calibrated and adaptive bound. The model predicts a high quantile of the future tension, the 90th percentile under a $\tau = 0.9$ pinball loss, and is judged by coverage and sharpness rather than by R^2 . Coverage asks whether the realised tension stays below the envelope at the intended rate, and sharpness asks how tight the envelope is given that coverage. The margin is made adaptive, widening with horizon and with sea-state energy. Adaptivity is what distinguishes the envelope from a fixed safety factor. The pinball formulation developed for the static peak term in Section 4.5.2 transfers directly to this dynamic setting. Following Koenker and Bassett [50], the quantile is the minimiser of the pinball loss, and by the consistency result of Gneiting [51] the trained model targets the τ -quantile of the true tension, a conservative upper bound appropriate for short-horizon alarming.

Keeping the degradation curve interpretable: The horizon sweep is only meaningful if the architecture is held fixed across horizons. Re-tuning per horizon would confound a harder task with a different model. The encoder backbone is reused without change, and only the decoder is new. The hyperparameters are tuned once at the longest and hardest horizon, and the Kendall weighting then balances the loss terms automatically at every shorter horizon, which is the direct payoff of the uncertainty-weighting study and removes a per-horizon weight search. The single horizon-sensitive control is the teacher-forcing schedule of the autoregressive rollout.

References

- [1] Kai-Tung Ma, Yong Luo, Chi-Tat Thomas Kwan, et al. *Mooring System Engineering for Offshore Structures*. Gulf Professional Publishing, 2019. ISBN: 978-0-12-818551-3. URL: <https://www.sciencedirect.com/book/9780128185513/mooring-system-engineering-for-offshore-structures>.
- [2] United Nations Framework Convention on Climate Change (UNFCCC). *The Paris Agreement*. 2015. URL: <https://unfccc.int/process-and-meetings/the-paris-agreement> (visited on 02/18/2026).
- [3] Global Wind Energy Council. *Global Offshore Wind Report 2025*. Brussels, Belgium: Global Wind Energy Council, 2025. URL: <https://gwec.net> (visited on 02/18/2026).
- [4] European Commission. *Offshore Renewable Energy*. 2024. URL: https://energy.ec.europa.eu/topics/renewable-energy/offshore-renewable-energy_en#offshore-wind-energy (visited on 02/18/2026).
- [5] International Renewable Energy Agency (IRENA). *Floating Offshore Wind Outlook*. Abu Dhabi, United Arab Emirates: International Renewable Energy Agency, 2024. URL: <https://www.irena.org> (visited on 02/18/2026).
- [6] Bruno Paduano. "Current status and perspectives on shared moorings for offshore floating renewable energy systems: A review". In: *Renewable and Sustainable Energy Reviews* 226 (2026), p. 116350. DOI: 10.1016/j.rser.2025.116350.
- [7] Zhiyu Jiang. "Installation of offshore wind turbines: A technical review". In: *Renewable and Sustainable Energy Reviews* 139 (2021), p. 110576. DOI: 10.1016/j.rser.2020.110576.
- [8] J. M. J. Journée and W. W. Massie. *Offshore Hydromechanics*. First. Lecture notes (TU Delft OCW). Delft University of Technology, Jan. 2001. URL: https://ocw.tudelft.nl/wp-content/uploads/OffshoreHydromechanics_Journee_Massie.pdf.
- [9] Kun Xu, Torgeir Moan, Zhen Gao, et al. "Design and Analysis of Mooring System for Semi-submersible Floating Wind Turbine in Shallow Waters". In: *NTNU Master Thesis Poster Exhibition*. Conference poster. Norway, May 2015. URL: https://www.researchgate.net/publication/277302726_Design_and_Analysis_of_Mooring_System_for_Semi-submersible_Floating_Wind_Turbine_in_Shallow_Waters.
- [10] C. T. Kwan and F. J. Bruen. "Mooring Line Dynamics: Comparison of Time Domain, Frequency Domain, and Quasi-Static Analyses". In: *Proceedings of the Offshore Technology Conference (OTC)*. Paper No. OTC-6657-MS (23rd Annual OTC, 6–9 May 1991). Houston, Texas, USA, May 1991. DOI: 10.4043/6657-MS. URL: <https://onepetro.org/OTCONF/proceedings-pdf/91OTC/A11-91OTC/OTC-6657-MS/1997242/otc-6657-ms.pdf>.
- [11] Thomas S. Walton and Harry Polachek. *Calculation of Nonlinear Transient Motion of Cables*. Tech. rep. Washington, D.C.: David W. Taylor Model Basin, Department of the Navy, July 1959. URL: <https://archive.org/details/calculationofnon00walt>.
- [12] Najeeb Ullah Khan and K. A. Ansari. "On the Dynamics of a Multicomponent Mooring Line". In: *Computers & Structures* 22.3 (1986), pp. 311–334. DOI: 10.1016/0045-7949(86)90037-4.
- [13] Shan Huang. "Dynamic Analysis of Three-Dimensional Marine Cables". In: *Ocean Engineering* 21.6 (1994), pp. 587–605. DOI: 10.1016/0029-8018(94)90008-6.
- [14] M. Hall and A. Goupee. "Validation of a lumped-mass mooring line model with DeepCwind semisubmersible model test data". In: *Ocean Engineering* 104 (2015), pp. 590–603. DOI: 10.1016/j.oceaneng.2015.05.035.
- [15] D. L. Garrett. "Dynamic Analysis of Slender Rods". In: *Journal of Energy Resources Technology* 104.4 (1982), pp. 302–306. DOI: 10.1115/1.3230419.

- [16] Orcina Ltd. *OrcaFlex Documentation, Line Theory*. URL: <https://www.orcina.com/webhelp/OrcaFlex/Content/html/Linetheory,Overview.htm> (visited on 07/10/2026).
- [17] Matthew Hall. *MoorDyn User's Guide*. Current for version 1.00.02, 15 December 2015. Department of Mechanical Engineering, University of Maine. Orono, Maine, USA, 2015. URL: <http://www.matt-hall.ca/files/MoorDyn%5C%20Users%5C%20Guide%5C%202015-12-15.pdf>.
- [18] SINTEF Ocean. *RIFLEX Theory Manual, Finite Element Formulation, SIMA Documentation*. URL: https://simasite.azurewebsites.net/docs/5.0/riflex/theory/finite_element_formulation.html (visited on 07/10/2026).
- [19] Jonathan Glasspool, Tiago Duarte, Julia Race, et al. "Balancing Fidelity and Computational Efficiency in Mooring System Simulations for Floating Offshore Wind Turbines". In: *Proceedings of the ASME 2025 44th International Conference on Offshore Mechanics and Arctic Engineering (OMAE)*. Volume 5: Ocean Renewable Energy. 2025, V005T09A014. DOI: 10.1115/OMAE2025-155460.
- [20] William L. Hamilton. *Graph Representation Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2020. DOI: 10.2200/S01045ED1V01Y202009AIM046.
- [21] Franco Scarselli, Marco Gori, Ah Chung Tsoi, et al. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.
- [22] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, et al. "Relational Inductive Biases, Deep Learning, and Graph Networks". In: *arXiv preprint arXiv:1806.01261* (2018). DOI: 10.48550/arXiv.1806.01261.
- [23] Chunhao Jiang and Nian-Zhong Chen. "G-Twin: Graph neural network-based digital twin for real-time and high-fidelity structural health monitoring for offshore wind turbines". In: *Marine Structures* 103 (2025), p. 103813. DOI: 10.1016/j.marstruc.2025.103813.
- [24] S. Daenens, T. Verstraeten, P.-J. Daems, et al. "Spatio-temporal graph neural networks for power prediction in offshore wind farms using SCADA data". In: *Wind Energy Science* 10.6 (2025), pp. 1137–1152. DOI: 10.5194/wes-10-1137-2025.
- [25] Mingqiu Liu, Jinguang Wang, Maokun Ye, et al. "Physics-Guided Graph Neural Networks for Floating Offshore Wind Turbine Wake Prediction". In: *Proceedings of the 35th International Ocean and Polar Engineering Conference (ISOPE)*. 2025, ISOPE-I-25–027.
- [26] José Lucas De Melo Costa, Fernando Kurike Matsumoto, Caio F. Deberaldini Netto, et al. "Spectral Graph-Based Networks for Mooring Line Failure Detection on FPSO". In: *Proceedings of the ASME 2024 43rd International Conference on Offshore Mechanics and Arctic Engineering (OMAE)*. Volume 1: Offshore Technology. 2024, V001T01A051. DOI: 10.1115/OMAE2024-136899.
- [27] Kobe Hoi-Yin Yung, Qing Xiao, Xiuqing Xing, et al. "Structural Surrogate Modelling of a Floating Offshore Wind Turbine with Physics-Guided Spatial-Temporal Graph Neural Network". In: *Journal of Physics: Conference Series* 3131.1 (2025). EERA DeepWind Conference 2025, p. 012033. DOI: 10.1088/1742-6596/3131/1/012033.
- [28] Kobe Hoi-Yin Yung, Qing Xiao, Xiuqing Xing, et al. "Physics-Guided Spatial-Temporal Graph Neural Network for Finite Element Surrogate Modelling of a Floating Offshore Wind Turbine With Wind-Wave Load Directionality Effect". In: *Proceedings of the ASME 2025 44th International Conference on Offshore Mechanics and Arctic Engineering (OMAE)*. Volume 5: Ocean Renewable Energy. 2025, V005T09A037. DOI: 10.1115/OMAE2025-155626.
- [29] Shagun Agarwal, Sergio Sánchez Gómez, and Oriol Colomé. "Dynamic Finite-Strain Analysis of Mooring Lines Based on Tangential Differential Calculus". In: *SSRN Electronic Journal* (2025). Preprint. DOI: 10.2139/ssrn.5271454. URL: <https://ssrn.com/abstract=5271454>.
- [30] T. P. Fries and D. Schöllhammer. "A unified finite strain theory for membranes and ropes". In: *Computer Methods in Applied Mechanics and Engineering* 365 (2020), p. 113031. ISSN: 0045-7825. DOI: 10.1016/j.cma.2020.113031. URL: <https://www.sciencedirect.com/science/article/pii/S0045782520302152>.

- [31] S. Badia and F. Verdugo. “Gridap: An extensible Finite Element toolbox in Julia”. In: *Journal of Open Source Software* 5.52 (2020), p. 2520. DOI: 10.21105/joss.02520.
- [32] J. Chung and G. M. Hulbert. “A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method”. In: *Journal of Applied Mechanics* 60.2 (1993), pp. 371–375. DOI: 10.1115/1.2900803.
- [33] J. I. Gobat and M. A. Grosenbaugh. “Application of the generalized- α method to the time integration of the cable dynamics equations”. In: *Computer Methods in Applied Mechanics and Engineering* 190.37 (2001), pp. 4817–4829. ISSN: 0045-7825. DOI: 10.1016/S0045-7825(00)00349-2. URL: <https://www.sciencedirect.com/science/article/pii/S0045782500003492>.
- [34] F. Zijl, T. Zijlker, S. Laan, et al. *3D DCSM-FM: a sixth-generation model for the NW European Shelf*. Tech. rep. Delft, The Netherlands: Deltares, 2022.
- [35] R. Santjer, S. Agarwal, O. Colomés, et al. “Gaussian copula-based Bayesian Networks for dynamic loads in mooring systems”. In: *Applied Ocean Research* 165 (2025), p. 104809. ISSN: 0141-1187. DOI: 10.1016/j.apor.2025.104809. URL: <https://www.sciencedirect.com/science/article/pii/S0141118725003955>.
- [36] Deltares. *D-Flow Flexible Mesh, User Manual*. Deltares. Delft, The Netherlands, 2024.
- [37] N. Booij, R. C. Ris, and L. H. Holthuijsen. “A third-generation wave model for coastal regions: 1. Model description and validation”. In: *Journal of Geophysical Research: Oceans* 104.C4 (1999), pp. 7649–7666. DOI: 10.1029/98JC02622.
- [38] B. P. Welford. “Note on a Method for Calculating Corrected Sums of Squares and Products”. In: *Technometrics* 4.3 (1962), pp. 419–420. DOI: 10.1080/00401706.1962.10490022.
- [39] Shaked Brody, Uri Alon, and Eran Yahav. “How Attentive are Graph Attention Networks?” In: *International Conference on Learning Representations (ICLR)*. 2022. DOI: 10.48550/arXiv.2105.14491. URL: <https://openreview.net/forum?id=F72ximsx7C1>.
- [40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, et al. “Graph Attention Networks”. In: *International Conference on Learning Representations (ICLR)*. 2018. DOI: 10.48550/arXiv.1710.10903. URL: <https://openreview.net/forum?id=rJXMpikCZ>.
- [41] PyG Team. *GATv2Conv – PyTorch Geometric Documentation*. 2024. URL: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GATv2Conv.html.
- [42] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- [43] PyTorch Team. *torch.nn.LSTM – PyTorch Documentation*. 2024. URL: <https://docs.pytorch.org/docs/2.12/generated/torch.nn.LSTM.html>.
- [44] PyTorch Team. *torch.nn.SmoothL1Loss – PyTorch Documentation*. 2024. URL: <https://docs.pytorch.org/docs/2.12/generated/torch.nn.SmoothL1Loss.html>.
- [45] Alex Kendall, Yarin Gal, and Roberto Cipolla. “Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics”. In: *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7482–7491. DOI: 10.1109/CVPR.2018.00781.
- [46] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations (ICLR)*. 2019. DOI: 10.48550/arXiv.1711.05101. URL: <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [47] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 2)*. 2024. URL: <https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2>.
- [48] Adam Paszke, Sam Gross, Francisco Massa, et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*. 2019, pp. 8024–8035. DOI: 10.48550/arXiv.1912.01703. URL: <https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html>.

- [49] Matthias Fey and Jan Eric Lenssen. “Fast Graph Representation Learning with PyTorch Geometric”. In: *ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds*. 2019. DOI: 10.48550/arXiv.1903.02428.
- [50] Roger Koenker and Gilbert Bassett. “Regression Quantiles”. In: *Econometrica* 46.1 (1978), pp. 33–50. DOI: 10.2307/1913643.
- [51] Tilmann Gneiting. “Making and Evaluating Point Forecasts”. In: *Journal of the American Statistical Association* 106.494 (2011), pp. 746–762. DOI: 10.1198/jasa.2011.r10138.