Mechanical Health Monitoring of Robotic Systems Using Graph Neural Networks

AUTHOR

Kane Jackson

SUPERVISOR

Dr. Ang Liu

School of Mechanical and Manufacturing Engineering

Faculty of Engineering

UNSW Sydney • 2026

Abstract

Graph neural networks (GNNs) have proven effective for structural and machinery health monitoring, yet their application to multi-joint robots remains nascent, and no published diagnostic incorporates the robot's configuration. This thesis asks how structural knowledge should enter a robot diagnostic GNN, testing two mechanisms on a simulated 6-DoF manipulator instrumented with per-joint vibration and acceleration sensing: (i) pose-conditioned adjacency, in which edge connectivity is gated by joint configuration, and (ii) a learned joint–sensor coupling matrix driving an anomaly-residual localisation head. A ROS 2 pipeline generates a fault-rich corpus spanning six fault modes across trajectory, speed and payload regimes; nine model variants are evaluated under a leakage-controlled, multi-seed protocol. Pose conditioning yields no measurable improvement - a negative result reported in full. The learned-coupling head, by contrast, localises faults at 0.95 accuracy in-distribution and - under sensor placements unknown to the model, where every fixed-graph and non-graph baseline collapses to chance ($\approx$0.17) - still at $\approx$0.79, recovering the true coupling at 0.97 correlation. Structure, as the results suggest, is a necessity for a system to spatially attribute it's mechanical health to its physical form, and is better learned than assumed.

Contents

List of Figures

List of Tables

1 Introduction

Multi-joint robots are increasingly deployed in settings such as production lines, logistics chains, and remote or hazardous environments, where an undetected mechanical fault is costly and an unlocalised one is barely more useful than no diagnosis at all. A robot is also, structurally, a graph: joints and links form a kinematic chain, sensors attach to components, and disturbances propagate along physical connections. Graph neural networks (GNNs), which learn directly on such relational structure, have delivered state-of-the-art results in the neighbouring domains of structural and machinery health monitoring, yet their application to robots remains nascent - and no published diagnostic incorporates the robot's configuration, despite the fact that what is "normal" for a robot's sensors depends strongly on its pose and load. This thesis asks how structural knowledge of a robot should enter a graph-based diagnostic model, and answers it empirically on a simulated 6-DoF manipulator. This chapter reviews the foundations and prior applications of GNN-based health monitoring (sections 1.1–1.4), distils the open gaps in the literature (section 1.5), and states the research questions, aims and contributions that follow from them (section 1.6).

1.1 Graph Neural Networks for Sensor Data

Traditional neural networks compute on regular, grid-like inputs such as images or sequences. Graph neural networks instead learn representations on graph-structured data: given a system represented as a graph $G = (V, E)$, with nodes V corresponding to sensors or components and edges E encoding relationships between them, each node iteratively aggregates information from its neighbours to update its own embedding - a process called message passing [8]. This topological reflection of a system lets a GNN learn interactions between interconnected components that would be missed by processing each signal independently, and GNNs have accordingly achieved state-of-the-art results across Internet of Things and sensor-network tasks by exploiting connectivity structure in the data [5, 1].

Two broad families of graph convolution exist. Spectral GNNs, exemplified by Kipf and Welling's Graph Convolutional Network (GCN), define convolution in the graph's frequency domain via the Laplacian eigenbasis, localised through polynomial filter approximations [1]. Spatial GNNs define convolution directly on the graph by aggregating neighbour features under the message-passing paradigm - the approach of GraphSAGE [6] and the Graph Attention Network (GAT) [14] - and are generally the more flexible formulation for varying graph structures, making them particularly applicable to dynamic robotic systems. Stacking layers propagates information across multi-hop neighbourhoods, so local connectivity yields progressively global context; the caveat is that excessive depth over-smooths node representations, eroding exactly the per-node distinctions that spatially resolved tasks depend on [12] - a constraint that shapes the architecture of

section 2.4.1.

Health monitoring is inherently temporal, which motivates spatio-temporal GNNs (ST-GNNs): GNN layers capture spatial relations while recurrent units (LSTMs, GRUs) or temporal convolutions model how those relations evolve, a combination pioneered for traffic forecasting (STGCN, DCRNN) and since adapted widely to mechanical and structural monitoring [12]. For multi-sensor mechanical systems, the resulting joint spatio-temporal representation offers a principled way to fuse heterogeneous signals - vibration, current, temperature - placed across a physical structure, learning both *when* and *where* anomalies emerge rather than treating channels as independent features.

1.2 GNN Applications in Structural and Mechanical Health Monitoring

Structural health monitoring (SHM) supplied the earliest evidence that graph-structured learning pays off on real sensor networks. Bloemheuvel et al. [1] designated each sensor on an instrumented bridge as a graph node, with edges encoding the sensor network's physical layout, and showed a GCN could predict strain at one sensor from the others more accurately than models that disregarded the topology. Liu et al. [12] extended the idea temporally with TPS-GNN, encoding daily and seasonal periodicity alongside GCN spatial layers to improve bridge-response prediction by up to 78% over prior models. Most relevant to this thesis, Xu et al. [15] observed that a single fixed graph struggles to capture all damage-relevant relationships, and proposed a dual-graph GCN that adaptively fuses a fixed geometry-based graph with a second graph learned from data features, achieving accurate localisation of multiple damage sites. The lesson - that a physical structure supplies one natural graph, latent data relationships supply another, and fusing them can outperform either - directly informs the learnable-adjacency mechanism evaluated in section 2.4.4.

A parallel machinery-diagnosis literature reports the same pattern for rotating equipment. Chen et al. [4] treated each sensor channel of a subway transmission system as a graph node and connected per-time-step subgraphs into a spatio-temporal graph, improving fault identification especially in small-sample settings; related work has shown multi-view graph fusion and limited-data Siamese graph architectures outperforming CNN and MLP baselines [13, 11], and a survey by Chen et al. [3] catalogues the field's rapid growth. A notable emerging thread is the integration of domain physics into the graph algorithm itself - for instance the Energy-Propagation GNN of Li et al. [10], whose message passing mimics physical energy flow and improves out-of-distribution robustness. These systems are machines of interconnected parts whose behaviours are interdependent - precisely the situation of a robot composed of joints, links and sensors.

1.3 GNNs for Health Monitoring of Robotic Systems

Despite this momentum, explicit applications of GNNs to robot fault diagnosis are scarce. Zhang et al. [16] proposed a dual-graph convolutional network for a mobile robot, combining a prior-knowledge fault-mode graph with a data-driven sensor-relationship graph to outperform non-graph baselines. Beyond isolated studies of this kind, the multi-joint industrial-robot case is essentially open: recent robot-diagnosis work using classical signal-processing pipelines explicitly notes that GNN models have demonstrated strong performance in mechanical diagnostics but that their application to real-time, multi-joint robotic fault detection remains limited, and recommends their integration as future work. The likely obstacles are the difficulty of obtaining large labelled fault datasets for robots and the additional complexity that a robot's configuration changes over time, altering load distributions and sensor correlations. Adjacent work hints at the flexibility of the tool - Jiao et al. [7] detect anomalies in robot *trajectories* by treating time-steps as graph nodes - but component-level fault diagnosis on articulated robots under a GNN framework remains largely unrealised, placing this project at the frontier of the method's application.

The need is sharpest in safety-critical and extreme environments, where related spacecraft work demonstrates both value and feasibility. Kiprit et al. [8] forecast each satellite telemetry channel from the others with a GCN, flagging anomalies on prediction error at an F1-score of 0.89 on NASA's SMAP/MSL datasets, and Lai et al. [9] showed with STGLR that *learning* a dynamic graph over telemetry variables - rather than fixing one - captures the cross-variable correlations through which system-level faults manifest. Both points transfer to robots in space, underwater or nuclear settings: graph models can fuse many channels into system-level insight, and they run within realistic on-board budgets. Model robustness and adaptability of the graph as the system changes are correspondingly treated as design considerations throughout this thesis.

1.4 Graph-Based Modelling of Robotic Systems

Applying a GNN to a physical system requires deciding what the graph is, and the model's effectiveness is intrinsically bounded by that choice. For nodes, the literature offers two poles and a hybrid. Component-level graphs mirror the mechanism - each joint or link a node, edges along the kinematic chain - so a disturbance at one joint can propagate to its neighbours exactly as it does physically; sensor-channel graphs, common in machinery diagnosis, instead make every transducer a node, capturing finer instrumentation detail. A hybrid, argued here to suit robots best, represents both: component nodes carry mechanical state, sensor nodes carry their readings, and explicit attachment edges join each sensor to the component it observes, as depicted in Figure 15 (Appendix). This is the schema adopted in section 2.2.

For edges, three strategies recur. *Physical connectivity* hard-codes the mechanism's design -

clear and data-efficient, but blind to indirect couplings such as resonance between distant joints. *Data-driven construction* learns the graph from signal correlations or by attention over an initially dense graph, as in STGLR's dynamic graph learning [9]; this can discover couplings designers never anticipated, at the cost of stability, interpretability and computation. *Hybrid physics-plus-data* approaches start from a structural graph and let the model adjust or add edges - the dual-graph fusion of Xu et al. [15], or the adaptive topology module of Chen et al. [2], which perturbs a complete multichannel graph into an optimised per-domain topology. Edges may further carry weights, direction or physical attributes, up to embedding physical law in the propagation rule itself [10]. The ablation design of section 2.4.2 operationalises exactly this spectrum: a fixed kinematic graph, a learned residual on it, and - ultimately - a coupling learned outright.

1.5 Identified Research Gaps and Limitations in the Literature

Figure 1 positions the reviewed literature along the three axes of this thesis - the *problem* (health monitoring and fault diagnosis), the *means* (graph neural networks), and the *context* (robotic systems). Every pairwise intersection is well populated: GNNs are established for structural and machinery health monitoring (section 1.2), robot fault diagnosis is established under classical signal-processing methods, and GNNs appear in robotics for non-diagnostic tasks. The triple intersection, however, is nearly empty - a single mobile-robot study [16] - and the specific cell this thesis occupies, GNN-based fault diagnosis of a multi-joint manipulator with configuration integrated into the model, is unoccupied. The gaps that follow make this positioning precise:

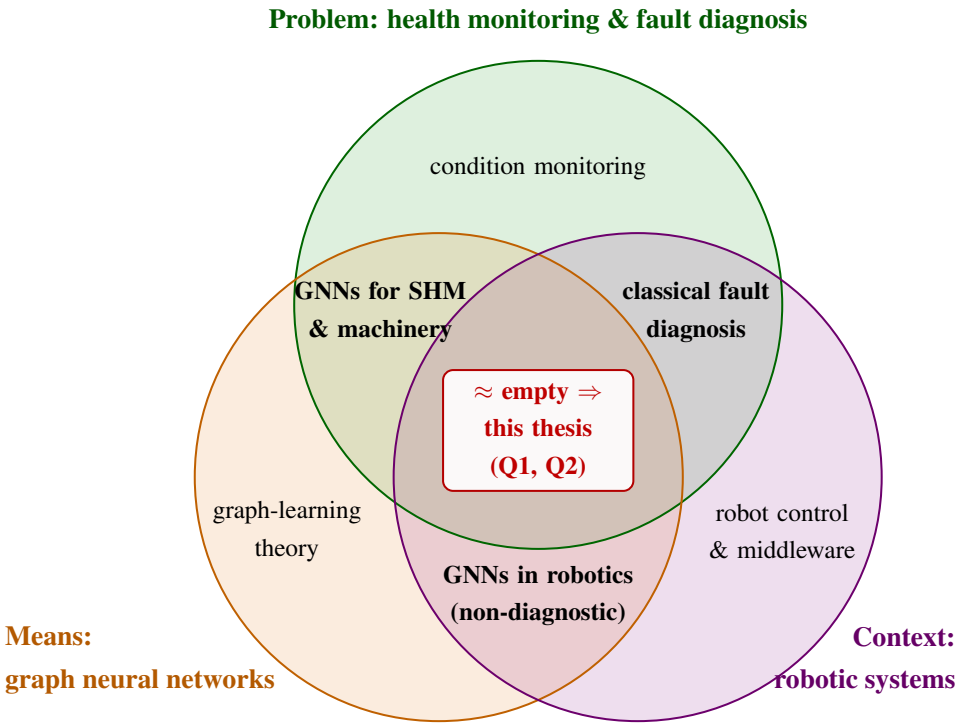

Figure 1: Positioning of the reviewed literature at the intersection of problem, means and context. Pairwise intersections are well populated (representative works shown); the triple intersection is nearly empty, and the cell this thesis occupies - multi-joint manipulator diagnosis with configuration in the model - is unoccupied.

- **Lack of Robot Pose/State Integration in Models:** The most striking gap is that existing GNN-based monitoring methods do not incorporate the robot's pose or dynamic state (joint angles, velocities, accelerations) into the diagnostic model: sensor signals become node features while configuration is ignored. Yet a robot's normal sensor behaviour is highly state-dependent - a motor's current draw or vibration level depends on the arm's position and load - so what is normal in one pose may be anomalous in another, inviting false alarms and missed detections. The literature acknowledges the need to handle continuously changing operational conditions, but at the time of writing no published work integrates these state variables directly into a diagnostic GNN. This is the novel direction the thesis explores.

- **Limited Application within Complex Robotic Systems:** Case studies applying GNNs to full multi-degree-of-freedom robots are nearly absent (section 1.3). The gap is one of validation and domain-specific tailoring rather than a flaw in GNNs per se - real-time execution on robot controllers, fusion of proprioceptive and exteroceptive sensing, and integration with robotic middleware are largely unaddressed. This project fills the gap by demonstrating a complete GNN health-monitoring framework on a robotic platform.

- **Challenges in Graph Construction and Uncertainty:** It remains unclear how to optimally construct or learn a robot's graph (section 1.4). Fixed graphs may be suboptimal; unconstrained learned graphs can be unstable and hard to interpret; dynamic graph learning carries a real computational cost [9]. An open question is how a robot's graph should capture both physical couplings and latent data relationships - and how it should respond when the system itself changes, as through damage or re-tooling.

- **Handling of Distribution Shifts and Novel Conditions:** Varying loads, speeds and tasks shift the sensor-data distribution, and a GNN trained under one regime may falter under another. Domain-adaptation work exists for rotating machinery [2], but for robots this is largely uncharted; robustness across poses and operating regimes is a necessity for real-world deployment and ties directly back to Gap #1.

- **Real-time Constraints and Scalability:** Industrial robots require timely fault response, yet comparative studies have reported deep-model inference times above 9 s - far too slow - and GNN runtime on robot hardware is rarely reported. Demonstrating deployable latency is therefore a completeness requirement for any proposed framework.

1.6 Research Questions, Aims and Contributions

The gaps identified above reduce to a single overarching question: *how should structural knowledge of a robot enter a graph-based diagnostic model?* Two candidate mechanisms are investigated, deliberately framed as open empirical questions rather than foregone conclusions:

Q1 (configuration). Does conditioning the graph's connectivity on the robot's joint configuration - making the adjacency an explicit function of pose - improve fault diagnosis? This responds directly to Gap #1, for which no published precedent was found.

Q2 (coupling). Can the joint↔sensor coupling itself be *learned* rather than fixed by the structural graph, such that faults can be localised to the correct joint even when the physical sensor placement is unknown to the model - the retrofit or aftermarket-mounting scenario that a fixed graph cannot represent?

These questions are pursued through three aims. **Aim 1** constructs a fault-rich simulated environment: a ROS 2 pipeline generating labelled multi-sensor data for a 6-DoF manipulator across a curriculum of trajectories, speeds, payloads, and six injected fault modes. **Aim 2** trains spatio-temporal graph neural networks on this corpus, within an ablation design built so that each structural mechanism's marginal contribution - pose kernel, learned adjacency residual, learned coupling

- can be isolated. **Aim 3** evaluates the *specificity* of diagnosis along three clauses: detecting that a fault is present, classifying its mode, and localising it to the affected joint.

To preview the arc of the thesis: the evidence answers Q1 in the negative - pose conditioning yields no measurable improvement, a result reported in full because a clean null on an untested mechanism is itself a contribution - while Q2 is answered strongly in the affirmative. The learned-coupling anomaly-residual head localises faults under sensor placements no baseline can handle, and recovers the true coupling pattern as an inspectable artefact. The thesis's primary contribution is therefore the coupling-agnostic localisation mechanism (Q2); its secondary contributions are the negative result on pose conditioning (Q1), obtained under a multi-seed protocol that overturned a single-seed artefact, and the reproducible simulation-to-inference pipeline itself.

2 Methodology

2.1 Overview and chapter roadmap

This chapter is organised around the three aims of the project - simulating a fault-rich environment (Aim 1), training a graph model on it (Aim 2), and evaluating the accuracy of the model's diagnosis (Aim 3) - preceded by the shared graph schema on which all three depend (section 2.2). It serves two questions of unequal eventual weight. The Q1 mechanism - pose-conditioned adjacency (section 2.4.4) - is the direct response to Gap #1 and is evaluated under a 2×2 ablation built so that a null result is as informative as a positive one; the results (section 3) present this, and it is reported in full as a secondary contribution. The primary contribution is the learned-coupling anomaly-residual head (section 2.4.3), which answers Q2: it localises faults through a *learned* joint↔sensor coupling rather than a fixed structural one, evaluated under the unknown-placement protocol of section 2.6.4. Both mechanisms are specified with equal rigour precisely so that the asymmetry of the findings is attributable to the mechanisms, ensuring methodological validity. Finally, the study is simulation-only: the analytic ROS 2 node-graph simulator of section 2.3.1 offers rapid model exploration and is defended in section 2.3.1. The sim-to-real gap this incurs is logged explicitly as a limitation in section 2.8.

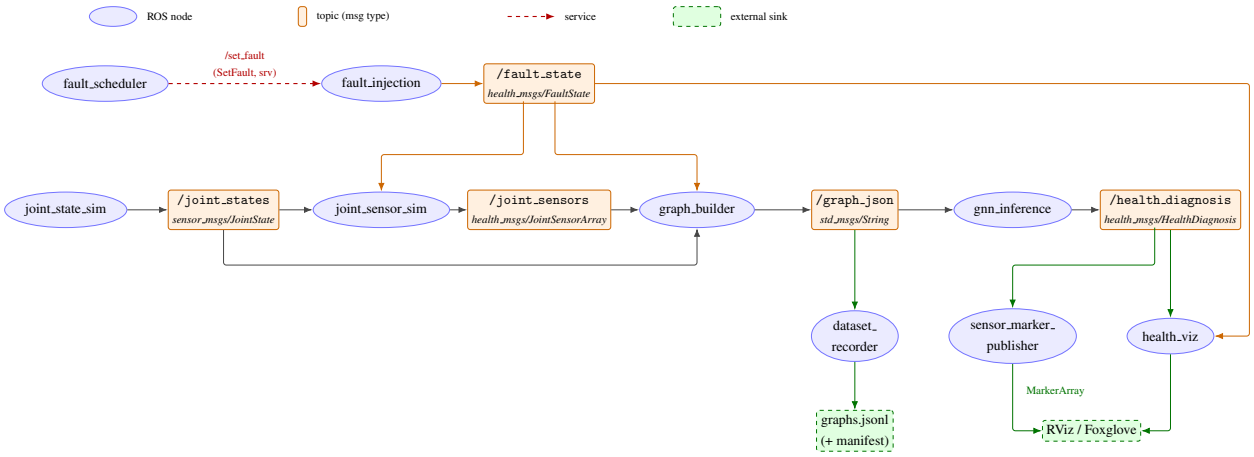

Figure 2: Pipeline block diagram spanning the three stages of the study - simulation, training and evaluation, and on-line inference - with the common `GraphSample` schema threaded through all three. This figure serves as the map for the chapter; subsequent sections refer back to it at each stage boundary.

2.2 System under study and graph construction

The platform is a simulated 6-DoF manipulator modelled on the UR5e kinematic structure, comprising six revolute joints in serial chain: `shoulder_pan`, `shoulder_lift`, `elbow`, `wrist_1`, `wrist_2` and `wrist_3`. Each joint is instrumented with a co-located sensor package consisting of a piezo vibration element and a tri-axis accelerometer, giving twelve signal sources across the arm. Every graph snapshot uses the same hybrid joint+sensor topology (Figure 3): six *joint* nodes carrying the joint-state triple (position, velocity, effort) and six *sensor* nodes carrying the vibration and accelerometer readings. Two edge types connect them: *Kinematic-backbone* edges link each pair of adjacent joints bidirectionally, mirroring the serial chain along which mechanical disturbances physically propagate; *sensor-attachment* edges link each joint to its co-located sensor. The graph is constructed per snapshot in `src/graph_builder/graph_builder/features.py`, which also fixes the canonical node ordering (six joint nodes followed by six sensor nodes) that every downstream consumer - training, evaluation and the on-line runtime - validates against. This topology is the hybrid that section 1.4 argued would benefit robotics: it sits between the pure component-location graphs of bridge SHM (Bloemheuvel et al. [1]) and the per-sensor-channel graphs of machinery diagnosis (Chen et al. [4]). Components and their instrumentation are distinct node types joined by explicit attachment edges, so the model can reason about a joint's mechanical state and its observed signals as related but separate entities - a distinction the learned-coupling head of section 2.4.3 later exploits by severing exactly this attachment assumption.

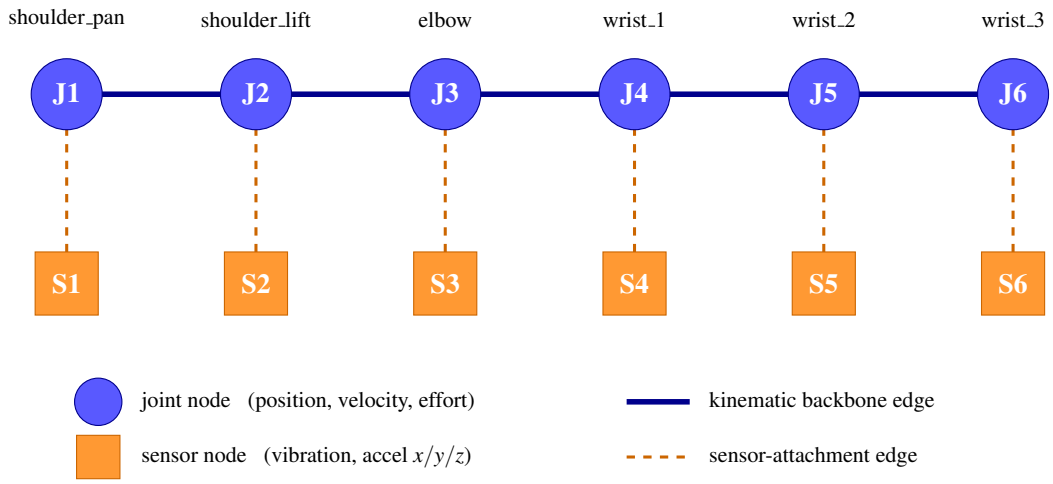

Figure 3: Graph schema used by every snapshot: six joint nodes and six sensor nodes in the canonical ordering, joined by bidirectional kinematic-backbone edges along the serial chain and sensor-attachment edges between each joint and its co-located sensor.

Every node carries the same nine-dimensional feature vector: two node-type indicators, the joint state triple (position, velocity, effort) and the sensor quadruple (piezo vibration; accelerometer x, y, z). Joint nodes zero the sensor channels; sensor nodes carry their co-located joint's state alongside their own readings, so a sensor is never presented to the model in isolation from the joint it observes. This replication has a consequence exploited later: pairwise position differences across joint–sensor attachment edges are identically zero, so the pose kernel of section 2.4.4 acts only on the kinematic backbone.

2.3 Aim 1 - Simulated environment

2.3.1 Simulator architecture and the node-graph decision

The simulated environment is implemented as an analytic ROS 2 node graph rather than a Gazebo physics session. The six fault signatures required by this study - backlash chatter, friction drag, jam impulses, structural resonance scaling, sensor drift and sensor noise - are far easier to inject deterministically and analytically than to elicit from contact dynamics, where reproducing a specific fault at a specific severity is itself a separate research problem. Analytic injection yields perfectly labelled, exhaustive coverage of the joint$\times$fault$\times$severity space: every sample carries its ground-truth fault class, target joint and severity, and the fault scheduler can guarantee that no cell of that space is missing from the corpus. Realism is addressed by having every synthetic signal carrying a Gaussian noise floor, and the fault signatures embed bounded per-sample randomness.

This, however, introduces a trade-off: the pipeline exchanges physical fidelity for controllability and complete class coverage. This choice is what makes the class-balance guarantee of section 2.3.3 possible, but it also introduces the signature-circularity limitation examined in section 2.8 and section 4.6 - the model is, in part, learning to invert its own generator.

The pipeline consists of six ROS 2 nodes in dependency order, from trajectory generation through fault-modulated sensor synthesis to `GraphSample` assembly and JSONL recording; Table 1 states each node's role and Figure 4 outlines the dataflow between them. Two details are significant downstream: `fault_scheduler` enumerates healthy and faulted intervals automatically over all joints, fault types and severities (section 2.3.3), and `dataset_recorder` tags every sample with a `recording_run_id` so that run-aware splitting (section 2.4.6) does not affect the integrity of the data.

Table 1: ROS 2 simulation pipeline nodes and their roles.

Node	Role
`joint_state_sim`	Publishes `JointState` along the trajectory curriculum.
`joint_sensor_sim`	Synthesises vibration + accelerometer readings, fault-modulated.
`fault_injection`	Owns `FaultState`; accepts `SetFault`; expires faults on a timer.
`fault_scheduler`	Enumerates healthy/faulted intervals over joints $\times$ types $\times$ severities.
`graph_builder`	Joins state + sensors + fault into one `GraphSample`.
`dataset_recorder`	Appends JSONL with `recording_run_id` and writes the manifest.

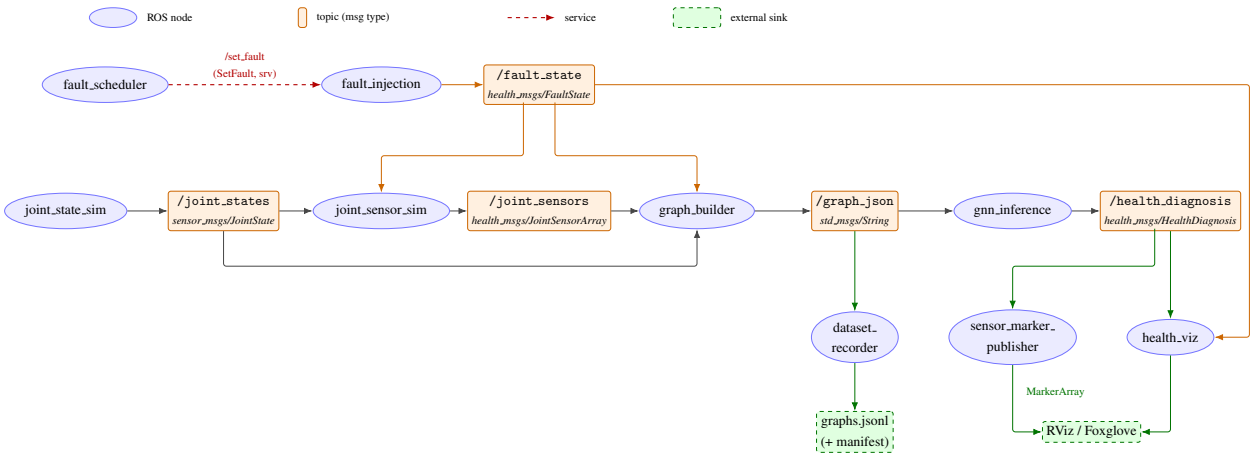

Figure 4: ROS2 node/topic dataflow graph.

2.3.2 Trajectory curriculum

The `joint_state_sim` node cycles through a curriculum of three trajectory families:

1. *Single-joint sweeps* move each joint through its range in isolation while the others are held static, repeated for every combination of speed scale and payload scale.

2. *Two-joint coordinated pairs* exercise every pair of joints simultaneously with fixed phase offsets so the resulting load is non-trivial.

3. *Whole-arm motion* activates all six joints with staggered phase offsets, cycling three waveforms - sine, triangle and a compound two-harmonic waveform - over the full speed and payload grid.

Speed enters the curriculum as a multiplier on the base excitation frequency, and payload as a multiplier inside the effort model, so the two parameters shift the healthy signal distribution along physically distinct axes. With speed scales $\{0.6, 1.0, 1.5\}$ and payload scales $\{0.5, 1.0, 1.6\}$, the curriculum comprises 126 distinct segments, shuffled per session by a seed so no two recordings exhibit the same operating envelope in the same order.

Making speed and payload first-class curriculum parameters is the design-time answer to Gap #4 (section 1.5): the recorded healthy distribution must span regimes rather than orbit a single operating point. The setup for Q1 uses this design - the curriculum deliberately produces a pose-dependent healthy distribution, so if configuration-gated adjacency has any value, this design gives provides the opportunity to show it.

2.3.3 Fault catalogue

Six fault modes are simulated, each with a distinct analytic signature applied inside `joint_sensor_sim` (Table 2). Four are mechanical faults and two are sensor faults, so the catalogue covers both the plant and the instrumentation. The effects of mechanical faults are spatially transmissible: the injected severity is attenuated by $\exp(-0.45\,d)$ where d is the hop distance from the target joint along the kinematic chain, so a fault at the elbow leaves a decayed imprint on the wrist and shoulder sensors. It is this propagation structure that a graph model is positioned to exploit. The two sensor faults differ deliberately in their spatial footprint: `SENSOR_DRIFT` applies an accumulating per-channel integrator bias strictly at the target joint's sensor (a corrupted transducer, not a mechanical event), while `SENSOR_NOISE` injects additive Gaussian noise whose variance scales with the decayed influence.

The `fault_scheduler` ROS2 node enumerates the full Cartesian product,

$$\{6 \text{ joints}\} \times \{6 \text{ fault types}\} \times \{\text{severities } 0.3, 0.6, 0.9\},$$

equating to 108 scenarios per pass, alternating each 12 s fault interval with an 8 s healthy interval. The schedule is reshuffled per the session seed on every pass. Coverage is therefore accounted for

Table 2: Fault catalogue and analytic signatures in `joint_sensor_sim`

Code	Class	Signature
1	`BACKLASH`	Chatter on velocity sign reversals; opposite-axis acceleration kicks.
2	`FRICTION`	Effort-proportional vibration; sustained z-axis acceleration bias.
3	`JAM`	Periodic impulses on vibration and all accelerometer axes.
4	`STRUCT_WEAKEN`	Resonance-scaled multiplicative gain on vibration and acceleration.
5	`SENSOR_DRIFT`	Slow per-channel integrator bias at the target sensor only.
6	`SENSOR_NOISE`	Additive Gaussian noise of fault-scaled variance.

by construction; no joint, fault type or severity can be absent from a sufficiently long recording. Two consequences of this schedule are carried forward. First, the class composition is balanced across the six fault classes, with a deliberate healthy majority, since every fault interval is bracketed by healthy time; section 2.7 reports macro-averaged per-class metrics alongside accuracy precisely so this structural majority cannot mask per-fault weaknesses (analysed in section 3.2). Second, each signature embeds bounded Gaussian randomness, so no two instances of the same scenario produce identical waveforms, mitigating fault-model determinism (elaborated in section 2.8).

2.3.4 Recording protocol

Each invocation of `ros2 launch robot_bringup dataset.launch.py` produces two artefacts: a JSONL file containing one `GraphSample` per publication tick ($\sim$20 Hz), and a `manifest.json` recording the schema version, the session's `recording_run_id`, the start time and the recorder configuration. The recorder stamps every individual sample with its `recording_run_id` (auto-generated per session from a timestamp and unique suffix) together with a monotonic sample index, so provenance survives any later file merging. The `recording_run_id` is load-bearing: it is the grouping key that makes the run-aware split of section 2.4.6 true, since without a per-session identifier on every sample, windows cut from a merged corpus could not be partitioned by recording session. The thesis corpus comprises at minimum eight distinct recording sessions (twelve were ultimately collected; section 3.1), varied across the fault scheduler's random seed (different fault orderings), the trajectory curriculum's speed and payload scales (different operating regimes), and session length. A subset of sessions is reserved as held-out and are not merged into training. The `merge_datasets.py` script consolidates the training sessions into a single corpus, while each held-out session remains in its own JSONL file for evaluation (section 2.6.1).

2.4 Aim 2 - GNN model and training

2.4.1 Shared backbone

All ST-GNN variants share the same backbone (Figure 5). Two spectral `GraphConv` layers (linear transform of the symmetrically normalised adjacency's propagation, LayerNorm, ReLU, dropout) each carry a residual skip connection that preserves per-node identity across propagation rounds, guarding against the over-smoothing that would otherwise erode the spatially discriminative signal that joint-level localisation depends on (section 1.1's depth caveat). A per-node GRU then encodes the temporal window, after which a single cross-node Transformer self-attention layer lets each node compare itself against its peers - identifying itself as the spatial outlier - before scoring. Two heads follow: a graph-level classifier over the seven classes (healthy plus six fault modes), and a per-node localisation head that receives the refined embeddings concatenated with a multi-scale skip from the first `GraphConv` layer, supplying the least-smoothed spatial evidence alongside the temporally-refined features. Localisation is reported at joint resolution throughout - sensor-fault classes included localise to the affected *joint* node - and sensor-node logits are excluded from the localisation argmax by construction. The GAT and MLP baselines deliberately do not share the full backbone (section 2.8 notes the resulting comparison caveat): the GAT variant swaps the spectral convolutions for multi-head graph-attention layers over the same structural mask but omits the cross-node attention and multi-scale skip, and the MLP encodes each node's flattened window independently with no cross-node pathway at all.

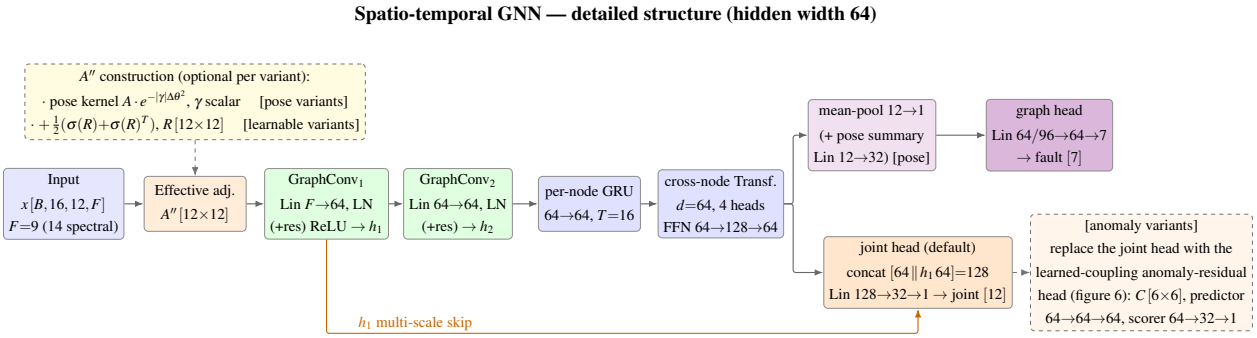

Figure 5: Shared ST-GNN architecture. Dashed components are optional, variant-specific; solid elements are present in every ST-GNN variant.

2.4.2 Evaluated model variants

The evaluation compares nine model variants (Table 3); their ordering is an argument by construction. Two baselines establish what is achievable without an explicit structural graph: `mlp`, which encodes each node's window independently with no cross-node pathway - the control whose localisation collapse (section 3.4) proves the graph is necessary for joint attribution - and `gat` [14], which

learns edge weights implicitly from node-feature pairs over the fixed structural mask, included as the strongest in-distribution classifier and as the implicit counterpart to the explicit, interpretable pose kernel (section 4.5). Four ST-GNN variants then form a 2×2 design over two factors - whether the sigmoid-gated symmetric learned residual R of Eq. (3), adapting the fixed-plus-learned dual-graph fusion of Xu et al. [15] (section 1.2), is added to the adjacency, and whether the pose kernel is applied - so that each factor's marginal effect can be read directly off the matrix (section 3.5). A fifth spectral-feature probe, `stgnn_pose+spectral`, appends window-level statistical and spectral summaries to the node features of the pose-conditioned model (per-window RMS, peak, spectral centroid and high-band energy ratio of the vibration channel, plus the RMS of the accelerometer x–z magnitude), ruling out feature poverty as the explanation for any pose null; its checkpoints alone are not deployable without porting the dataloader-side feature block (section 2.6.6).

Where the adjacency variants ask how a *known* structure should be weighted, the final two variants remove the assumption that the joint–sensor coupling is known at all in order to address Q2. The primary contribution model, `anomaly_residual`, predicts each sensor embedding from its coupled joints through a *learned* coupling matrix C; a fault breaks that learned relation, and the resulting residual energy is scattered back through C to localise it (section 2.4.3, Figure 6). Since this model's coupling is not tied to the known structural graph, it is able to localise under unknown sensor placement (section 3.7). The `pose+anomaly` model combines the head with the pose kernel, confirming that the Q1 result holds in the head's presence and does not interact with it.

Table 3: Model variants evaluated, grouped by role: baselines, the 2×2 adjacency ablation serving Q1, and the learned-coupling localisation head serving Q2 (the primary contribution). Every model named in the results chapter is defined here.

Variant	Adjacency / graph	Mechanism added	Isolates
Baselines			
`mlp`	none	-	[a]
`gat`	structural mask	attention weights	[b]
Adjacency ablation (2×2: pose $\times$ transmissibility, Q1)			
`stgnn_fixed`	structural	-	[c]
`stgnn_fixed+pose`	structural, pose-gated	pose kernel	[d]
`stgnn_learnable`	structural + residual[†]	learned residual	[e]
`stgnn_pose-cond.`	structural + residual, pose-gated	pose + residual	[f]
Spectral-feature probe			
`stgnn_pose+spec.`	structural + residual, pose-gated	+ spectral features	[g]
Learned-coupling localisation head (Q2, primary contribution)			
`anomaly-resid.`	structural + coupling[‡]	anomaly-residual head	[h]
`pose+anomaly`	structural + coupling, pose-gated	head + pose kernel	[i]

[†] Learned symmetric adjacency residual R (Eq. 3).

[‡] Learned joint–sensor coupling matrix C (Eq. 1).

[a] No cross-node message passing; its localisation collapse proves the graph is necessary for joint attribution.

[b] Implicit learned edge weighting, as the counterpart to the explicit pose kernel.

[c] The structural-graph reference point.

[d] The pose kernel alone, without a learned residual.

[e] The learned residual alone, without pose.

[f] The full Q1 mechanism (both factors together).

[g] Whether the pose null persists under richer frequency-domain node features.

[h] Localisation via learned C, untied from the structural graph; the only variant robust to unknown sensor placement.

[i] Whether the Q1 null holds in the presence of the Q2 head.

2.4.3 Learned-coupling anomaly-residual head (primary contribution, Q2)

A single learnable coupling matrix acts as a property of the machine, shared across samples: $C = \mathrm{softplus}(C_{\mathrm{logits}}) \in \mathbb{R}_{\geq 0}^{J \times M}$ over J joint and M sensor nodes, initialised near-uniform with no structural prior. C is consumed under two normalisations: C^{pred}, column-normalised over joints, for predicting each sensor from the joints it couples to; and C^{scat}, row-normalised over sensors, for attributing evidence back to joints. Each sensor embedding is predicted from its coupled joint

embeddings and a reconstruction residual formed,

$$\hat{s}_m = f\Big(\sum_j C^{\text{pred}}_{jm}\, h_j\Big), \qquad r_m = s_m - \hat{s}_m, \tag{1}$$

after which two quantities are scattered back through the coupling: a *learned* per-sensor anomaly score $g(r_m)$ yields the per-joint localisation logits $\ell_j = \sum_m C^{\text{scat}}_{jm}\, g(r_m)$, while the raw residual energy $\|r_m\|^2$ is scattered separately to drive a self-supervised auxiliary penalty (weight 0.1) that pushes residuals toward zero on *healthy* joint–sensor pairs. On healthy data the learned relation therefore holds and residuals vanish; a fault at joint j breaks the relation at precisely the sensors C couples to j, and the scatter attributes the anomalous energy back to j. Sensor-node logits are pinned low so the localisation argmax always lands on a joint. Because C is learned rather than read from the structural graph - and because the head supports $M \neq J$ natively - localisation does not require the sensor placement to be known.

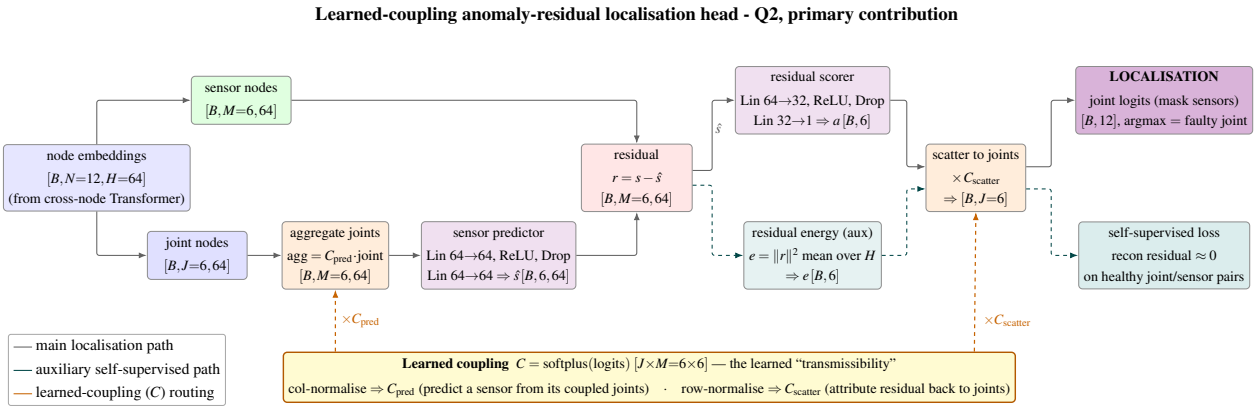

Figure 6: Primary contribution model architecture, designed to predict each sensor from its coupled joints; a fault at a joint breaks that relation, spiking residual energy at coupled sensors, which the learned coupling scatters back to the joint

2.4.4 Pose-conditioning mechanism (Q1)

The most recent joint-position vector in the window modulates edge weights so that effective connectivity is a function of configuration. Positions are standardised before entering the model, having all features z-scored per channel against training-split statistics in the data pipeline. Through this method, pose differences are measured in units of training-set standard deviations, so the kernel does not operate on raw radians. A single learned scalar bandwidth γ is initialised to 1.0 and optimised jointly with the network, gating each structural edge:

$$A'_{ij} \;=\; A_{ij}\exp\!\big(-|\gamma|\,(\hat{\theta}_i - \hat{\theta}_j)^2\big), \tag{2}$$

where $\hat{\theta}$ denotes the standardised positions. Three properties fix the interpretation. First, $|\gamma|$ is the inverse bandwidth of a Gaussian radial basis function (RBF) kernel over configuration distance, and the absolute value constrains the kernel to decay only. The effect of this is that connectivity is gated by pose difference, but not amplified. Secondly, $\gamma \to 0$ recovers the unconditioned graph, so the mechanism can learn to switch itself off; the converged $|\gamma|$ is therefore itself a reportable, dimensionless estimate of how strongly the robot's configuration gates inter-component transmission. Thirdly, because sensor nodes replicate their joint's position (section 2.4.1), attachment-edge differences vanish and the kernel acts only on the kinematic backbone in which pose changes the physical transmission path. Pose enters at a second site as well: the last position vector is passed through a small MLP and concatenated to the mean-pooled graph embedding before classification, giving the classifier direct access to *absolute* configuration where the kernel sees only relative differences. This is a physics-informed dynamic adjacency: distinct from STGLR's purely data-driven dynamic graph [9], which infers connectivity from the stream at the computational cost noted in Gap #3, and from the static dual-graph fusion of Xu et al. [15], which adapts structure to data but not to the operating state.

A learnable symmetric residual adds data-driven edges to the (optionally pose-gated) structural graph:

$$A'' \;=\; A' + \tfrac{1}{2}\big(\sigma(R) + \sigma(R)^{\top}\big), \tag{3}$$

with self-loops enforced and the result symmetrically normalised before convolution. R is initialised to a strong negative bias (-4.0, so $\sigma(R) \approx 0.018$). With this, the learned graph starts equal to the kinematic graph and can only add transmissibility weight where doing so reduces the loss. A zero initialisation for $\sigma(R) = 0.5$ on every node pair densifies the sparse graph from the first step and over-smooths the per-joint signal that localisation depends on.

2.4.5 Loss, optimisation, hyperparameters

The training objective is

$$\mathscr{L} = \mathrm{CE}(\text{graph head}) + \lambda\,\mathrm{BCE}(\text{joint head}) + 0.1\,\mathscr{L}_{\text{recon}}, \tag{4}$$

with $\lambda = 0.5$, where the reconstruction term applies to anomaly-residual variants only (Eq. (1)). Optimisation uses Adam (learning rate 10^{-3}, weight decay 10^{-4}), dropout 0.1, batch size 128, window 16 / stride 4 at the base configuration, and a run-aware training/validation/test split of $70/15/15$. Each window is labelled by its *final* snapshot, so windows straddling a fault onset carry the fault label on partial evidence - a policy whose cost is quantified in the per-class analysis (section 3.6) and the long-window collapse (section 3.8). Each run trains for 30 epochs with the best-validation checkpoint retained; selection landed between epochs 5 and 24 across variants.

2.4.6 Splitting protocol and leakage

The *run-aware* split groups windows by the `recording_run_id` of their final snapshot and then partitions whole recording sessions into train, validation and test with a seeded greedy assignment that tracks the 70/15/15 target proportions.

The failure mode that the run-aware split eliminates is the most obvious criticism on a synthetic-data result. With a stride of 4 and a window of 16, adjacent windows from the same recording would overlap in twelve of their sixteen snapshots; a window-level shuffle therefore places near-duplicates of training windows into the test set, and the resulting test accuracy measures memorisation of a session rather than generalisation across sessions. Grouping by `recording_run_id` guarantees that no two windows from the same session - overlapping or otherwise - can ever land in different splits, so the test partition consists entirely of recording sessions the model has never seen. Feature normalisation statistics are similarly computed from the training partition only, so no test-set information leaks through the z-scoring either. The splitter is guarded to not run on fewer than three distinct recording sessions, which structurally forces the recording protocol of section 2.3.4 to be followed.

2.4.7 Reproducibility

All reported results are the mean $\pm$ standard deviation over three seeds $\{42, 43, 44\}$; a single seed controls Python, NumPy and PyTorch randomness as well as the run-aware splitter, so each seed is a full independent replicate of initialisation, data order and split assignment. The multi-seed protocol was adopted after a single-seed pilot produced an apparent pose-conditioning gain that did not survive replication (section 4.1). Checkpoints store weights together with node order, adjacency, feature-normalisation statistics, the fault-label table and the full configuration dictionary. In this, evaluation, benchmarking and ONNX exports reconstruct the exact model; the converged $|\gamma|$ and best epoch are logged per run.

2.5 Aim 3 - Specificity of diagnosis

Aim 3 asks how specific the diagnosis can be made, and its three clauses - detect that a fault is present, classify its mode, and localise it within the mechanism - map one-to-one onto three outputs computed from the model's two heads for every window. *Fault presence* is the binary decision obtained by testing whether the classification argmax lands on any class other than `none`. *Fault classification* is the argmax over the seven classes (healthy plus the six fault modes of section 2.3.3). *Fault localisation* is the argmax over the per-joint logits, evaluated only on windows where a fault is actually active, so the metric answers the operative question - when there is a fault, did the model point at the right joint? - rather than rewarding arbitrary joint scores on healthy data.

Severity is deliberately not a fourth output. It is a first-class experimental variable - the scheduler enumerates severities and every sample records its injected value - but it is not a prediction target of this thesis (section 2.8 scope; severity regression is future work as detailed in section 4.7). The recorded severities are instead collected in the interest of severity-stratified error analysis in the discussion.

2.6 Experimental protocol

2.6.1 Datasets

Three dataset roles are distinguished (Table 4): the merged *training corpus*, consumed under the run-aware 70/15/15 split of section 2.4.6; the *held-out generalisation* session, kept separate from the training set, from which every cited test number comes; and the *distribution-shift probe*, the same fault mix recorded at a payload outside the training range (Gap #4). Every file passes through `summarize_dataset` before use, which reports per-class and per-joint support, per-run coverage and severity statistics, and acts as a gate: any file with missing fault classes or a degenerate joint-target distribution is rejected and re-recorded rather than silently trained on.

Table 4: Dataset roles in the experimental protocol.

Role	File	Purpose
Training corpus	`graphs_training_merged.jsonl`	Run-aware 70/15/15 split.
Held-out generalisation	`graphs_eval.jsonl`	Unseen session; the cited test numbers.
Distribution-shift probe	`graphs_eval_payload_high.jsonl`	Out-of-range payload, same fault mix.

2.6.2 Campaign: variant matrix and window-size ablation

A single resumable campaign command drives the full protocol: the nine-variant matrix at window 16 repeated over the three seeds, the window-size ablation (window $\in \{8, 16, 32, 48\}$ with stride locked to a quarter of the window) on the anomaly-residual and pose-conditioned variants, and held-out plus payload-shift evaluation of every checkpoint. It emits per-run raw rows and a per-(variant, window) mean $\pm$ std aggregate, which together are the source of every table in the results chapter. Runs whose held-out evaluation already exists are skipped, so the campaign is re-launchable after interruption without redoing work.

2.6.3 Held-out evaluation

Every checkpoint produced by the campaign is evaluated on the full held-out session with `evaluate.py`, using the entire file rather than any internal test subset. The evaluator reconstructs the exact trained

model from the checkpoint - weights, configuration, node ordering, adjacency and the training-split normalisation statistics - and refuses to run if the evaluation file's node order or graph topology differs from what the checkpoint was trained on, so a held-out number can never silently come from a mismatched graph definition.

2.6.4 Coupling-agnostic localisation protocol (Q2)

The retrofit scenario of Q2 is implemented with a synthetic coupling harness applied to the held-out session. A fixed ground-truth coupling $C_0 \in \mathbb{R}^{6\times4}$ mixes the four per-joint sensor channels (vibration; accelerometer x, y, z) into four harness sensors, each picking up a weighted combination of several joints in a setup where sensors are placed off-joint, in a many-to-many relationship. C_0 is designed so that every joint's row signature across the four sensors is distinct, enabling it to distinguish all six joints, and every joint sufficiently reaches at least one sensor; harness sensor nodes carry only their mixed sensor channels, with joint-state channels zeroed. imperatively, C_0 is withheld from the model twice over: no variant sees the matrix, and the graph itself is rebuilt with a *dense bipartite* joint–sensor edge set, so message passing is not informed of the coupling either. As such, fixed-graph variants retain assumptions on their structural attachment that are now physically incorrect, while the anomaly-residual head must recover the effective coupling from data alone through its learned C. Evaluation reports (i) localisation accuracy under the harness, and (ii) the mean per-joint correlation between softplus(C) and C_0, quantifying whether the recovered structure is physically meaningful rather than merely discriminative. The harness is synthetic by design, demonstrating the mechanism rather than a particular physical mounting - a limitation carried to section 2.8.

2.6.5 Latency and scalability

Per-window inference latency is measured off-line with `benchmark.py`: each checkpoint is run for 200 timed iterations with a batch size of 1 (after 10 warm-up iterations) on real held-out windows, on both CPU and Apple MPS backends, reporting `latency_ms_median`, `latency_ms_p95` and `throughput_windows_per_sec`. This measurement is the evidence base for Gap #5 (section 1.5): deployability at the simulator's $\sim$20 Hz tick (a 50 ms budget per window) is substantiated with a direct measurement rather than asserted. The off-line figures are later cross-checked against the latency the on-line runtime itself publishes (section 2.6.6, section 3.10).

2.6.6 On-line ROS deployment (best model only)

On-line inference is a deployment demonstration of the single best-performing checkpoint, not a per-variant evaluation. Once the variant matrix and held-out evaluation have selected the winner,

the checkpoint is exported to ONNX with `export_onnx` and the inference stack is launched with `infer.launch.py`. At runtime, `ModelRuntime` buffers a sliding window of incoming `GraphSample` snapshots, applies the stored training-split normalisation, runs the ONNX session at every tick, and publishes a `HealthDiagnosis` message carrying the fault decision, per-joint scores and a measured `inference_latency_ms`. This closes the loop between training and on-robot inference - the same `GraphSample` schema flows unmodified from the simulator through training to live diagnosis - and makes the reproducible pipeline of the secondary contribution concrete. Two constraints attach to this step. First, spectral-feature checkpoints are not deployable without porting the dataloader-side feature block (section 2.4.2); the headline anomaly-residual model carries no such dependency and deploys unchanged. Second, if the ONNX session or model file is unavailable, the runtime emits an explicit no-model result stating a fault is absent with zero confidence, and logs a warning so an apparently working demonstration always reflects a loaded model rather than an echo of the injected fault label. Quantitative on-line evaluation, logging `/health_diagnosis` against `/fault_state` to report on-line accuracy and detection latency, is a planned extension (section 4.7); until then on-line behaviour is reported qualitatively together with the published `inference_latency_ms`, corroborated against the off-line benchmark.

2.7 Metrics

Every model is scored on every dataset with the same metric set, each reported as mean $\pm$ standard deviation over seeds. *Classification accuracy* is top-1 over the seven classes. *Fault-present accuracy* is the binary detection rate (healthy versus any fault). *Per-class precision, recall and F1*, together with *macro-F1*, are reported because the corpus has a deliberate healthy majority (section 2.3.3): macro-averaging prevents the majority class from masking weakness on any fault mode, and the confusion matrices expose which fault pairs are conflated. *Localisation accuracy* is the argmax of the per-joint logits against the injected target joint, conditioned on a fault being present (section 2.5). *Inference latency* is the median, p95 and throughput from the benchmark protocol of section 2.6.5, and the *converged pose bandwidth* $|\gamma|$ is logged per seed for every pose-conditioned run (section 2.4.4). Each metric is reported over the grid {train, validation, held-out, distribution-shift probe}; that grid, with mean $\pm$ std over seeds for the headline variants, constitutes the substantive results table of the thesis. Severity is not a metric - per-severity stratification of the metrics above appears in the discussion instead.

2.8 Threats to validity and design mitigations

Table 5 collects the principal threats to the validity of a simulation-only diagnostic study and states, for each, the mitigation designed into this methodology or the explicit acknowledgement where

no design choice could remove it. Two deserve emphasis: the *sim-to-real gap* is acknowledged rather than addressed empirically, and *signature circularity* follows from the analytic simulator - the model partly learns to invert its own generator - with bounded per-sample randomness and regime variation as the in-corpus mitigations and per-run signature-parameter randomisation as future work.

Table 5: Threats to validity and their design mitigations.

Threat	Mitigation
Sim-to-real gap	Acknowledged limitation of a simulation-only study; not addressed empirically. Curriculum breadth makes the simulated regime as demanding as possible.
Signature circularity	Bounded per-sample randomness in every signature; severity and regime variation; per-run signature randomisation as future work. Stated explicitly as a limitation.
Run leakage between splits	Run-aware splitter keyed on `recording_run_id`; window-level shuffling prohibited for reported numbers (section 2.4.6).
Class imbalance	Full Cartesian fault schedule - balanced across fault classes with a deliberate healthy majority; macro-F1 and per-class metrics reported; `summarize_dataset` gates every file.
Window-label noise at fault onset	Windows labelled by final snapshot; onset windows carry partial evidence, examined in the per-class analysis (most acutely for `SENSOR_DRIFT`).
Hyperparameter cherry-picking	One configuration drives the entire ablation matrix; sweeps are window-size only; headline numbers averaged over three seeds.
Fault-model determinism	Bounded Gaussian randomness at fixed scales; the literal channel modulation is auditable in `joint_sensor_sim`.
Baseline asymmetry	GAT/MLP do not share the full localisation backbone (section 2.4.1); the resulting comparison caveat is carried to section 4.5.
Latency measurement	Off-line benchmark and the on-line `inference_latency_ms` corroborate one another (section 2.6.5, section 2.6.6).
Silent fallback in deployment	`ModelRuntime`'s no-model fallback returns an explicit zero-confidence, fault-absent result and warns; it cannot echo injected labels (section 2.6.6).

2.9 Mapping to literature gaps

Returning to the gaps of section 1.5: *pose and state integration* (Gap #1) is addressed directly by the pose-conditioned adjacency of section 2.4.4 and indirectly by the learned-coupling head of section

2.4.3, which models the joint→sensor transmission that pose was hypothesised to gate; *limited application to complex robotic systems* (Gap #2) by the complete ROS 2 path from simulation through training to on-line inference on a 6-DoF manipulator; *graph construction and uncertainty* (Gap #3) empirically, across four adjacency regimes plus an implicit-attention baseline, with the eventual finding that learning the coupling outright beats perturbing a structural prior; *distribution shifts* (Gap #4) by the curriculum's speed and payload variation at design time and the out-of-range payload probe at evaluation time; and *real-time constraints* (Gap #5) by direct measurement. Harsh-environment evaluation is out of scope for a simulation-only study; the two sensor-fault classes serve as in-simulation proxies for degraded instrumentation.

3 Results

Table 6: Held-out performance by variant (mean ± std over 3 seeds, window = 16). Localisation is conditioned on a fault being present; Payload is classification accuracy under high-payload distribution shift; γ is the converged pose-bandwidth (pose-conditioned variants only). Bold = best per column. The headline `anomaly_residual` row (bold label) is the best GNN on both heads at w=16 and improves further at its operating window w=32 (Table 8).

Variant	Cls. acc	Localisation	Macro-F1	Payload	γ
Baselines					
MLP (baseline)	0.934 ± 0.002	0.168 ± 0.012	0.932 ± 0.002	0.917 ± 0.004	–
GAT (baseline)	$\mathbf{0.963 \pm 0.001}$	0.888 ± 0.013	$\mathbf{0.964 \pm 0.002}$	$\mathbf{0.936 \pm 0.005}$	–
Adjacency ablation (2×2: pose × transmissibility, Q1)					
ST-GNN fixed	0.878 ± 0.023	0.930 ± 0.005	0.903 ± 0.013	0.861 ± 0.025	–
ST-GNN fixed+pose	0.863 ± 0.019	0.929 ± 0.001	0.896 ± 0.009	0.864 ± 0.014	0.22 ± 0.05
ST-GNN learnable	0.837 ± 0.063	0.915 ± 0.002	0.883 ± 0.034	0.840 ± 0.042	–
ST-GNN pose-cond.	0.822 ± 0.017	0.916 ± 0.004	0.874 ± 0.007	0.825 ± 0.013	0.36 ± 0.09
Spectral-feature probe					
ST-GNN pose+spectral	0.887 ± 0.015	0.910 ± 0.005	0.907 ± 0.009	0.890 ± 0.009	0.18 ± 0.07
Learned-coupling localisation head (Q2, primary contribution)					
ST-GNN anomaly-resid.	0.916 ± 0.013	$\mathbf{0.937 \pm 0.001}$	0.925 ± 0.008	0.910 ± 0.008	–
ST-GNN pose+anomaly	0.914 ± 0.036	0.933 ± 0.004	0.926 ± 0.025	0.911 ± 0.019	0.24 ± 0.05

3.1 Experimental setup and reporting conventions

The corpus for all reported experiments is the merge of twelve recording sessions, split 70/15/15 under the run-aware protocol keyed on `recording_run_id` (section 2.4.6). Every headline number is the mean $\pm$ standard deviation over three full replicates (seeds 42, 43, 44), each seed controlling initialisation, data order and split assignment; training runs for 30 epochs with the best-validation checkpoint retained, and selection landed between epochs 5 and 24 across variants. Two evaluation files sit outside the corpus entirely: the unseen held-out session and the out-of-range high-payload probe, neither of which is ever merged into training (section 2.6.1).

Four metrics recur throughout the chapter: seven-class classification accuracy, binary fault-present accuracy, localisation accuracy (argmax over the joint logits, conditioned on a fault being active), and macro-F1. Macro-F1 accompanies accuracy for one stated reason: the corpus carries a deliberate healthy majority (section 2.3.3), and macro-averaging prevents that structural majority from masking weakness on any individual fault class.

3.2 Dataset characterisation

The class composition follows the schedule by construction: support is balanced across the six fault classes, with a structural healthy majority arising from the healthy interval that brackets every fault injection. Figure 7 shows the recorded vibration and accelerometer traces at the affected joint's sensor for each fault class against a healthy reference, and confirms that the signatures are visually distinct and physically interpretable: backlash produces high-frequency chatter locked to velocity reversals, jams produce sparse large impulses, `sensor_drift` appears as a slow accumulating ramp, and `struct_weaken` as a resonance-dependent scaling of the healthy envelope. That final pair - a scaled version of healthy behaviour and a slowly accumulating bias - foreshadows the two hardest classes in the per-class analysis of section 3.6.

Operating-regime coverage is complete - every combination of trajectory family, speed scale and payload scale appears in the corpus - so the "healthy" class spans regimes rather than a single operating point, denying the classifier the shortcut of treating any unfamiliar regime as a fault (section 3.10 probes beyond even this envelope).

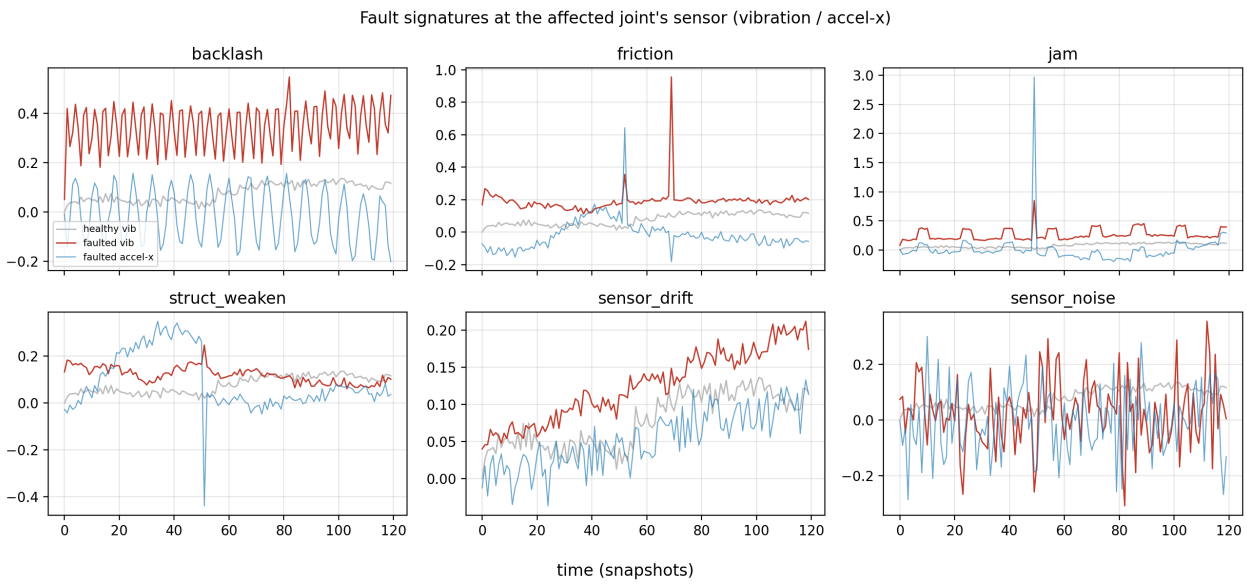

Figure 7: Vibration/accelerometer signatures at the affected joint's sensor for each fault class versus healthy.

3.3 Headline variant comparison

Table 6 and Figure 16 (Appendix) report held-out performance for all nine variants at the base window of 16. Read top-down, the classification ranking is: GAT 0.963 $>$ MLP 0.934 $>$ anomaly-residual 0.916 $\approx$ pose+anomaly 0.914 $>$ pose+spectral 0.887 $>$ fixed 0.878 $>$ fixed+pose 0.863 $>$ learnable 0.837 $>$ pose-conditioned 0.822. The headline, however, sits at the model's operating point rather than the shared base window: at w=32 (selected by the ablation of section 3.8) the anomaly-residual model reaches 0.935 classification and 0.950 localisation - the best localiser in the study, a near-best classifier, and the lowest-variance model of the nine.

The table is best read as two questions rather than one ranking. On *classification*, the baselines do conspicuously well - the attention baseline tops the column and even the non-graph MLP beats most of the structural variants. On *localisation*, the ordering inverts: the graph models separate cleanly from the MLP, which collapses outright. The next two sections take these questions in turn (section 3.4, section 3.7), and the discussion returns to why the graph's value concentrates in one and not the other (section 4.2).

Seed variance is tight for most variants ($\pm$0.01–0.02 on classification), with one outlier: the learnable-residual variant at $\pm$0.063, the least stable cell of the matrix - a point taken up in the 2$\times$2 decomposition of section 3.5.

3.4 Localisation requires the graph

The starkest result in the study is the localisation column. The MLP localises at 0.168 ± 0.012 - statistically indistinguishable from chance over six joints ($1/6 \approx 0.167$) - while every graph variant, of any adjacency regime, sits between 0.89 and 0.94 (Figure 8).

The collapse follows directly from the architecture. The MLP encodes each node's flattened window independently and mean-pools the node embeddings for classification; at no point does information from one joint's signals reach another node's representation. Attributing a fault to a specific joint, however, is inherently a relational judgement - the affected joint is the one whose signals are anomalous *relative to its neighbours*, especially when the fault's influence physically propagates down the chain (section 2.3.3). Without any cross-node pathway, each node's score is computed blind to its peers, and joint attribution degenerates to chance even while the same pooled features support strong classification.

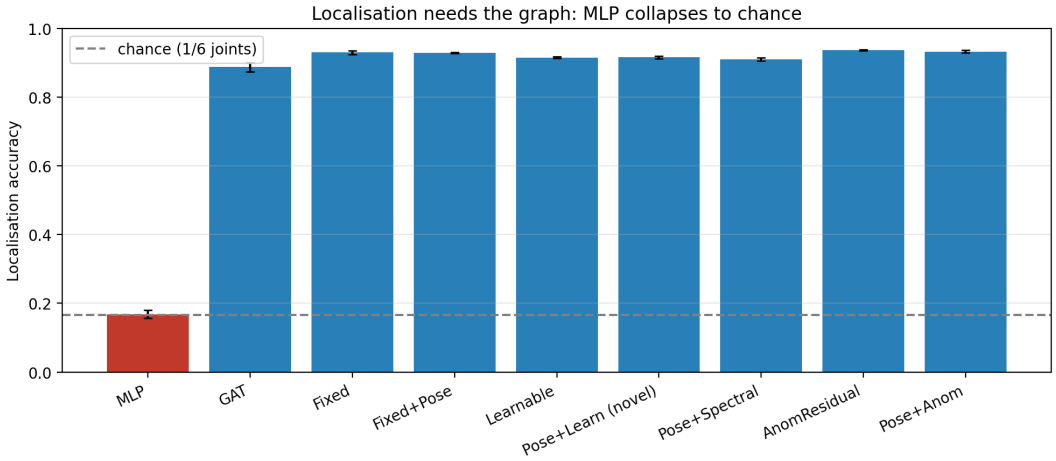

Figure 8: Localisation accuracy by variant; MLP collapses to chance.

3.5 Decomposing the adjacency mechanism (pose $\times$ transmissibility)

The four ST-GNN adjacency variants form the 2×2 design of section 2.4.2, and Figure 9 lays out the held-out classification cells: fixed 0.878, fixed+pose 0.863, learnable 0.837 ± 0.063, pose-conditioned 0.822. Two findings can be read directly off the marginals.

Finding 1 - pose conditioning does not improve classification. Applying the pose kernel costs -0.015 on the fixed graph and -0.015 on the learnable graph - neutral-to-slightly-negative on both rows, with the differences inside overlapping standard deviations. The effect is consistent in sign across both transmissibility conditions, which is what allows the Q1 null to be attributed to the pose mechanism itself rather than to an interaction with the learned residual.

Finding 2 - the learned residual R alone hurts and destabilises. The learnable cell is si-

multaneously the weakest non-pose cell and by far the highest-variance run in the matrix (± 0.063 against ± 0.01–0.02 elsewhere). The interpretation offered in section 2.4.4 is borne out: even from its conservative near-zero initialisation, the added edges densify the sparse kinematic graph and over-smooth exactly the per-joint signal that in-distribution discrimination relies on.

One foreshadowing note bounds Finding 2. The residual's failure here is a statement about the *in-distribution* setting, where the fixed kinematic graph is already correct and added edges can only dilute it. The value of learning structure appears precisely where the structural prior is wrong - the unknown-coupling setting of section 3.7 - and it is the coupling matrix C, not the adjacency residual R, that delivers it.

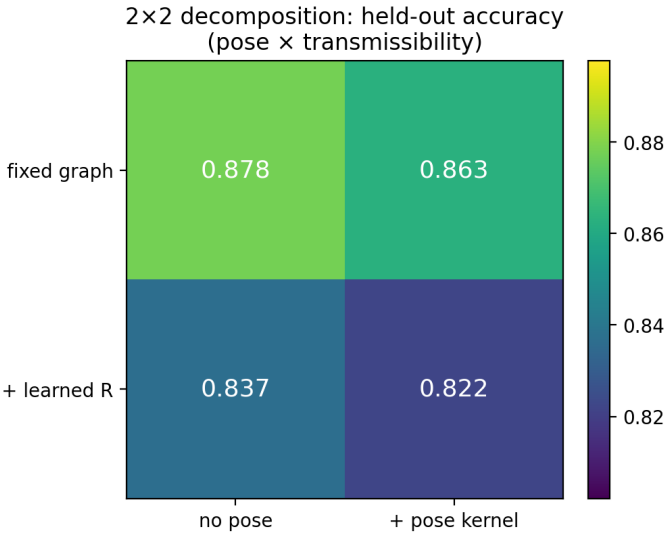

Figure 9: 2×2 decomposition of held-out fault classification accuracy.

3.6 Per-class behaviour and error structure

Figure 10 shows row-normalised held-out confusion matrices for the strongest classifier (GAT), the weakest (pose-conditioned) and the headline anomaly-residual model; Figure 18 (Appendix) gives per-class F1 across all nine variants. Three classes - `backlash`, `jam` and `sensor_noise` - are essentially solved by every model, including the MLP: their signatures (chatter, impulses, broadband noise) are large, distinctive and present throughout the fault interval. The discussion of error structure therefore concentrates on the two classes that are not.

The dominant error mode across all variants is `struct_weaken`↔`none` confusion. This is physically explicable rather than mysterious: structural weakening is implemented as a multiplicative, resonance-dependent scaling of the healthy signal envelope (section 2.3.3), so at low severity a weakened joint's traces are a mildly amplified copy of healthy behaviour - there is no qualitatively new feature to detect, only a gain change that must be distinguished from ordinary regime variation,

which the curriculum deliberately makes wide.

The second-weakest class is the slow-onset `sensor_drift`, and its errors trace to the window-labelling policy rather than to the signature itself. Windows are labelled by their final snapshot (section 2.4.5), so a window straddling fault onset carries the fault label while containing mostly healthy evidence; for an integrator fault whose bias accumulates from zero, the early faulted windows are nearly indistinguishable from healthy ones by construction. The onset-straddling label noise this creates is a known, documented cost of the policy, and it returns as the mechanism behind the long-window collapse in section 3.8.

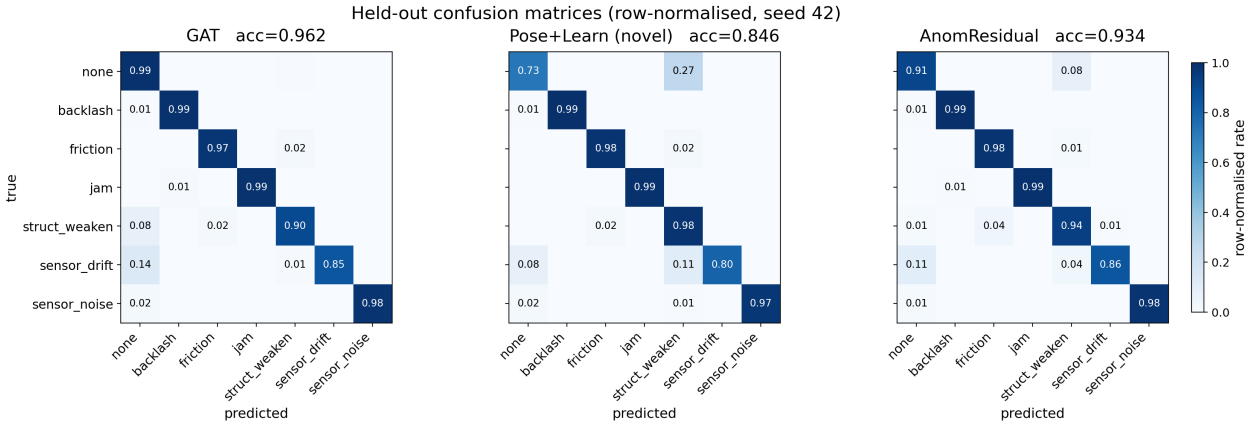

Figure 10: Row-normalised held-out confusion matrices.

3.7 Coupling-agnostic localisation under unknown sensor placement

The retrofit scenario of Q2 is evaluated with the harness protocol of section 2.6.4: the held-out session re-synthesised through the withheld coupling C_0, with the graph rebuilt as a dense bipartite joint–sensor edge set so that neither the model nor message passing is told the true placement. Fixed-graph variants therefore retain attachment assumptions that are now physically wrong, while the anomaly-residual head must recover the effective coupling from data alone through its learned C.

The result is categorical. The anomaly-residual head localises at ≈ 0.79 under the harness, while the fixed-graph and non-graph models collapse to ≈ 0.17, equivalent to the success rate of chance over six joints (Figure 11). The models that depend on the structural graph for attribution, whether explicitly through fixed attachment edges or implicitly through attention over them, have no mechanism by which to discover that sensor 2 now listens to the elbow and both wrists at once; the learned-coupling head does, because attribution is routed through C rather than through the graph.

The recovered coupling is, moreover, physically meaningful and not merely discriminative: the

learned C matches the withheld C_0 at a mean per-joint correlation of 0.97 (Figure 11), making it an inspectable artefact of training - a data-derived map of which sensors observe which joints, produced as a by-product of learning to localise. Simply, this is the capability that no baseline matches, and it is the empirical core of this thesis contribution. In-distribution, the graph variants trade a few accuracy points among themselves; under unknown sensor placement, one mechanism works and everything else is at chance.

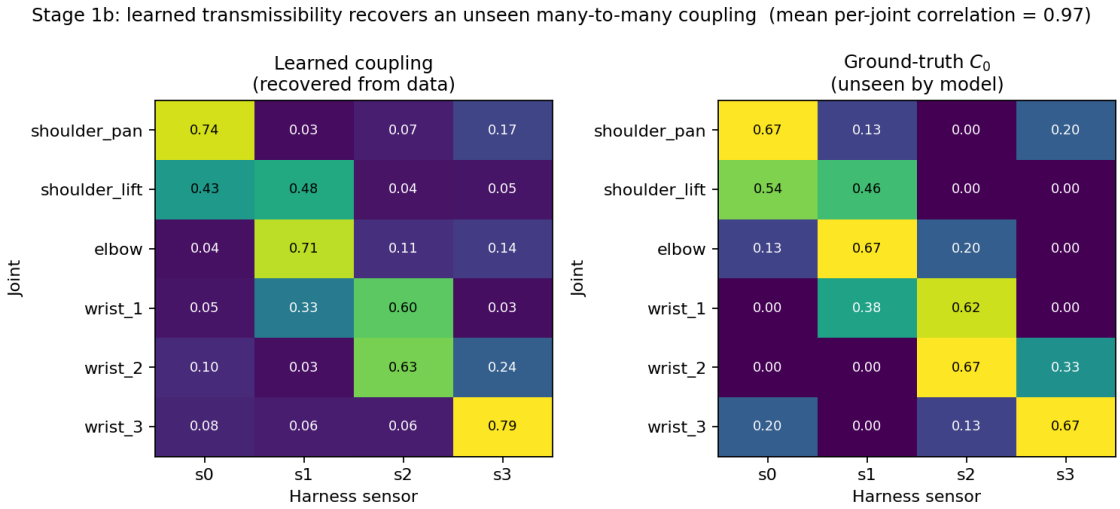

Figure 11: Localisation under unknown coupling; the learned C recovers the mounting pattern.

3.8 Window-size ablation

Table 8 and Figure 12 sweep the temporal window over $\{8, 16, 32, 48\}$ snapshots (0.4–2.4 s at the 20 Hz tick, stride locked to a quarter of the window) for the headline anomaly-residual model. The curve is a clean rise-then-drop: every metric improves monotonically from w=8 through w=16 to w=32 (classification 0.891 $\rightarrow$ 0.916 $\rightarrow$ 0.935; localisation 0.903 $\rightarrow$ 0.937 $\rightarrow$ 0.950), then classification collapses at w=48 to 0.865 with the variance inflating to $\pm$0.077. w=32 is therefore adopted as the operating point, and the headline numbers of section 3.3 are quoted there.

The drop at 48 has a mechanical explanation rather than a mysterious one, and it compounds two effects. Longer windows cut fewer training windows from a fixed-length corpus, so the model sees less data; and because each window is labelled by its final snapshot, longer windows straddle more fault-onset boundaries, so a larger fraction of training labels rests on partial evidence (section 3.6). Fewer windows with noisier labels is precisely the recipe for the observed combination of lower mean and higher variance. Localisation, notably, barely degrades at 48 (0.947) - the spatial attribution signal survives label dilution better than the class decision does.

The pose-conditioned model's window curve, by contrast, is noisier and non-monotonic (Fig-

ure 12), offering no comparably defensible operating point - a further reason the headline is reported for the anomaly-residual model at w=32 rather than for any pose variant at a window chosen post hoc.

Table 7: Window-size ablation on the headline `anomaly_residual` model (held-out, mean ± std). Steady rise in performance, then drops off past w=32.

Window (≈ s @20 Hz)	Cls. acc	Localisation	Macro-F1
8 (0.4)	0.891 ± 0.005	0.903 ± 0.004	0.905 ± 0.005
16 (0.8)	0.916 ± 0.013	0.937 ± 0.001	0.925 ± 0.008
32 (1.6)	$\mathbf{0.935 \pm 0.024}$	$\mathbf{0.950 \pm 0.002}$	$\mathbf{0.940 \pm 0.019}$
48 (2.4)	0.865 ± 0.077	0.947 ± 0.005	0.896 ± 0.039

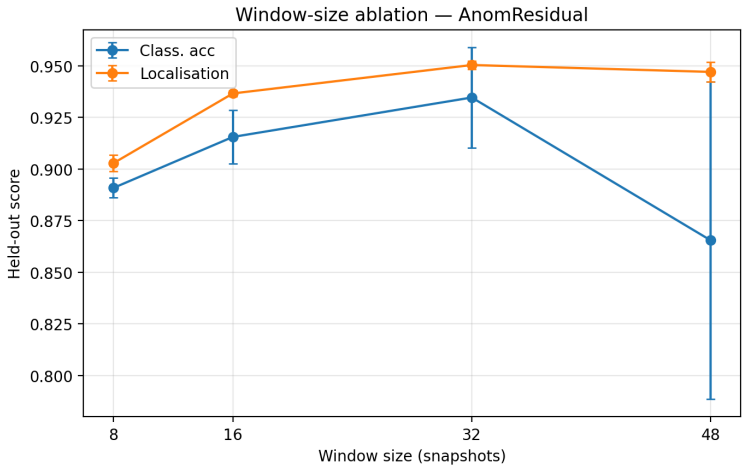

Figure 12: Window-size ablation result from the on the headline `anomaly_residual` model

3.9 Converged pose-bandwidth γ

Across every pose-conditioned variant at w=16, the learned bandwidth converges small: $|\gamma|$ lands between 0.18 and 0.36 across variants and seeds, well below its initialisation of 1.0 (Table 6; Figure 13). Recalling that $\gamma \to 0$ recovers the unconditioned graph (section 2.4.4), the optimiser is choosing to apply only gentle configuration gating - the mechanism partially switches itself off - and this converged, dimensionless value is itself the interpretable output the mechanism was retained for: an estimate that inter-component transmission in this simulated arm is only weakly gated by relative configuration, of a kind a black-box weighting such as GAT attention cannot supply in a single number. One systematic trend is visible: γ grows with window length, from roughly 0.31 at w=8 to 0.59 at w=48, plausibly because more temporal context lets the recurrent encoder

absorb the variance that aggressive edge gating introduces. The framing to retain is that γ is an interpretable by-product, not an accuracy lever: pose conditioning does not raise classification or localisation accuracy in this study (section 3.5).

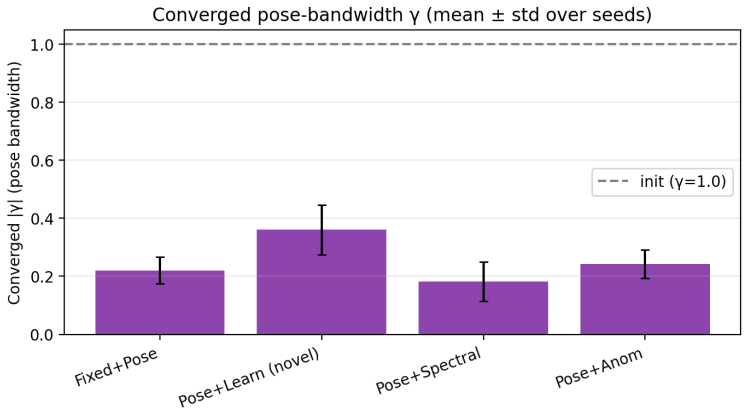

Figure 13: Converged pose bandwidth $|\gamma|$ per pose-conditioned variant (mean $\pm$ std over seeds, w=16).

3.10 Distribution shift and inference latency

Under the out-of-range payload probe, the picture is one of modest, consistent shift rather than failure (Figure 17, Appendix). Averaged over the nine variants, held-out classification accuracy moves by under one percentage point (about -0.7 points on average, computed from Table 6), with the largest drops on the two baselines (GAT -2.7, MLP -1.7 points) and the structural variants essentially flat; the headline anomaly-residual model loses 0.6 points. Crucially, the variant ranking is preserved under shift, so no conclusion drawn from the held-out comparison is overturned by the probe - evidence that the curriculum's speed and payload coverage (section 2.3.2) bought substantial regime robustness rather than interpolation within a narrow envelope.

Inference latency confirms real-time feasibility with a wide margin (Figure 14). At batch size 1, the ST-GNN variants run at sub-millisecond medians on CPU (with the MLP faster still), and around 3 ms on the MPS backend, whose per-call dispatch overhead dominates at this batch size - for single-window inference the CPU is the faster deployment target. Against the $\sim$50 ms budget of the 20 Hz tick, even the p95 latencies leave more than an order of magnitude of headroom, directly answering the $>$9 s inference-time concern cited under Gap #5. Finally, the off-line benchmark is corroborated on-line: the `inference_latency_ms` published per tick by `ModelRuntime` under `infer.launch.py` agrees with the CPU benchmark figures, so the latency claim rests on two independent measurements of the same quantity (section 2.6.5, section 2.6.6).

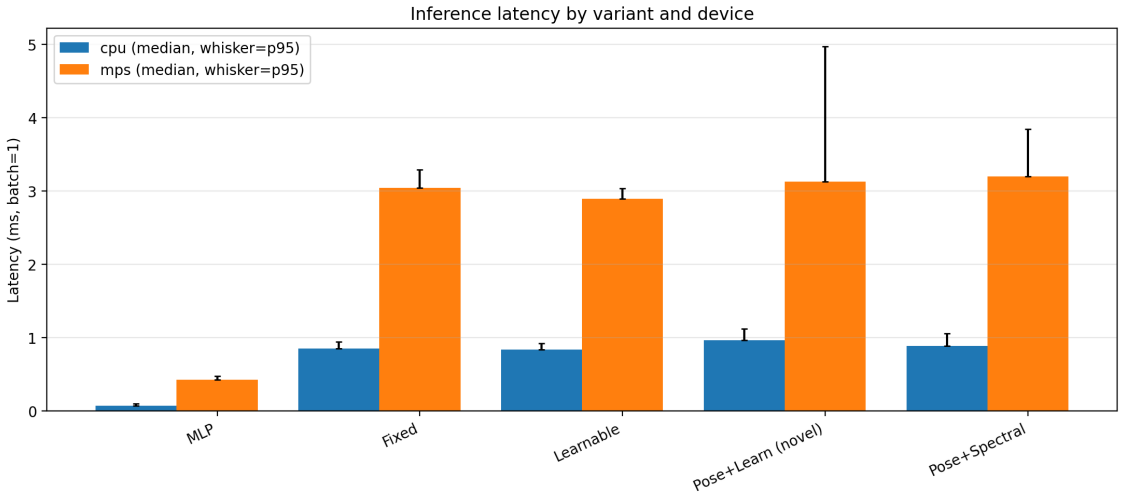

Figure 14: Inference latency by variant and device.

4 Discussion

4.1 Revisiting the hypothesis

The honest place to begin is with the hypothesis as originally posed: that a pose-conditioned, learned-transmissibility adjacency would be the winning mechanism - outperforming the non-graph baseline, the fixed-graph ST-GNN, and either mechanism applied alone. On in-distribution classification, that hypothesis is **not supported**. The 2×2 decomposition of section 3.5 shows the pose kernel accuracy-neutral at best and the learned residual actively harmful, and the combined pose-conditioned model is the weakest classifier in the matrix.

Why this is a finding and not a failure is a methodological point worth making explicitly. A single-seed pilot run had suggested a pose-conditioning gain; under the three-seed mean $\pm$ std protocol, that gain vanished - it was an artefact of one initialisation and one split assignment, not a property of the mechanism. The multi-seed protocol was adopted in direct response, and it is the reason the thesis can state the null with confidence rather than report a spurious positive. If the study had stopped at one seed, the central claim would have been wrong. The results do, however, support a reframed and arguably stronger conclusion, which the remainder of this chapter develops: the contribution moves from configuration-gated accuracy to *coupling-agnostic localisation and interpretability* - graph structure earns its place in this problem not through stronger classification, but by making spatial attribution possible (section 4.2), portable to unknown sensor placements (section 4.3), and inspectable (section 4.4).

4.2 Why the graph matters for localisation but not classification

Classification is, in this regime, comparatively easy - and deliberately so. The fault signatures are analytic, per-channel functions (section 2.3.3) whose classes are near-separable from single-node evidence: chatter, impulses, drift ramps and noise floors each leave a distinctive imprint on the affected sensor's own window. A model that never mixes information across nodes can therefore classify well, which is exactly what the MLP's 0.934 and the GAT's 0.963 demonstrate. This is a property of the simulator as much as of the models - the signature-circularity limitation of section 4.6 - and it explains why adding structural machinery to the classifier buys nothing: the discriminative evidence is already local.

Localisation is a different kind of question. Pointing at the faulted joint requires deciding which node's behaviour is anomalous *relative to its peers*, under fault influence that physically propagates down the kinematic chain with distance decay (section 2.3.3); it is a comparison across nodes, not a property of any one of them. Cross-node mixing is therefore not an optimisation but a precondition, and the MLP's collapse to chance (section 3.4) is the clean control: remove the pathway and joint attribution becomes impossible, while classification is untouched. The implication reframes what a GNN buys in this diagnostic setting. The graph's value is *spatial attribution, not discrimination*: structure determines whether the system can say *where*, not how well it says *what*. For maintenance use - where the actionable output is which joint to service - that is the more valuable half of the diagnosis.

4.3 The primary contribution: learned-coupling anomaly-residual localisation

The mechanism, in brief (section 2.4.3, Figure 6): a learned coupling matrix C predicts each sensor's embedding from the joints it couples to; on healthy data a self-supervised penalty holds that reconstruction tight, so a fault - which breaks the normal joint→sensor relationship - spikes residual energy at precisely the sensors coupled to the faulted joint, and scattering that energy back through C localises it.

The problem this uniquely solves is unknown sensor placement - the retrofit or aftermarket-mounting scenario in which sensors are installed on an existing machine without a verified mapping of which transducer observes which joint, or where one sensor observes several. Every other variant hard-wires the sensor→joint mapping into the graph, explicitly (fixed attachment edges) or implicitly (attention over them), and section 3.7 shows what follows when that assumption is false: collapse to chance, against the coupling head's ≈ 0.79.

Three practical virtues attach to the mechanism beyond the headline capability. First, C is inspectable: it recovers the withheld ground-truth coupling at 0.97 correlation, so training yields

a physically meaningful map of sensor–joint transmissibility as a by-product, not just a black-box score. Second, the model is no worse for it in the standard setting - at its w=32 operating point it is the study's best localiser (0.950) and a near-top classifier (0.935) with the lowest seed variance. Third, it deploys unchanged: unlike the spectral-feature variant, it carries no dataloader-side computation, so the checkpoint exports through the ONNX/ROS runtime as-is (section 2.6.6).

4.4 Pose conditioning: interpretability, not accuracy (the Q1 result)

The negative result is owned rather than buried: the learned-γ pose kernel improves neither classification nor localisation in this study, on any graph, at any window size, under a three-seed protocol expressly adopted after a single-seed gain failed to replicate.

Three candidate explanations are consistent with the evidence, and they are not mutually exclusive. First, the kernel operates on standardised position differences and converged to a gentle bandwidth ($|\gamma| \approx 0.18$–0.36 at w=16, from an initialisation of 1.0), so the realised edge modulation was mild - the mechanism learned to mostly switch itself off, which is itself evidence that stronger gating hurt during training. Second, transmissibility in the analytic simulator may be only weakly pose-gated: fault influence propagates with a fixed per-hop decay that does not depend on configuration, so the physical premise the kernel encodes is only faintly present in the data-generating process. Third, the analytic per-channel signatures dominate the classification problem (section 4.2), leaving little residual error for any adjacency refinement - pose-conditioned or otherwise - to remove.

What is retained from the mechanism is its interpretability. The converged γ is a single, dimensionless, physically meaningful scalar - an estimate of how strongly the arm's configuration gates inter-component transmission - and its systematic growth with window length (section 3.9) is a legible trend in a way that a matrix of attention weights is not. The value of pose conditioning in this study is therefore as an analysis as opposed to a mechanism for accuracy; whether a physics-based simulator or a real arm, where transmission paths realistically change with configuration, would convert that analysis back into accuracy is precisely the question the negative result illuminates for future work.

4.5 Explicit versus implicit graph weighting (the GAT contrast)

The GAT baseline sharpens what the thesis's design choices actually trade. As the strongest in-distribution classifier (0.963), it demonstrates that learning edge weights implicitly from node-feature pairs is highly effective when the structural mask is correct and the test distribution matches training. But it is a black box - its attention weights are per-sample, per-head functions of features, yielding no single reportable quantity comparable to γ or inspectable map comparable to C (mech-

anism contrast in Figure 19, Appendix); it is a weaker localiser (0.888) than any spectral ST-GNN variant; and under unknown coupling it fails alongside the fixed-graph models (≈ 0.20), because its attention operates only over the structural mask it is given and cannot discover that the mask is wrong.

The trade-off, then, is between marginal black-box accuracy in-distribution and interpretable, coupling-agnostic structure that survives a broken structural prior - and the thesis takes a side: for a health-monitoring system whose outputs must justify maintenance interventions and whose sensor installation may not match the design drawing, the second bundle of properties is worth the classification points it concedes. One caveat bounds the in-distribution comparison: the GAT baseline does not share the full localisation backbone (section 2.4.1), so its localisation gap conflates the weighting mechanism with the missing components; the comparison this section rests on is therefore the unknown-coupling collapse, which no backbone addition could repair.

4.6 Limitations

Threats addressed by construction - leakage, imbalance, label noise at fault onset, and the baseline asymmetry - were tabled with their mitigations in section 2.8; what remains are the limitations no design choice could remove.

The study is *simulation-only*. No sim-to-real claim is made anywhere in the thesis: the performance bounds established here apply to the analytic signatures of section 2.3.3, and transfer to a physical arm - with real structural dynamics, sensor mounting compliance and unmodelled disturbance - is untested by design (section 2.1).

Closely related is *signature circularity*: because the fault signatures are authored analytic functions, the model in part learns to invert its own generator, and reported accuracy is accuracy on these signatures. The in-corpus mitigations are the bounded per-sample randomness inside every signature and the breadth of operating-regime variation; the full mitigation - randomising signature parameters per run so the model must learn fault *classes* rather than exact waveforms - is identified as future work rather than claimed.

The *harness coupling is synthetic*: C_0 is an authored mixing matrix, designed so every joint remains distinguishable, and the unknown-placement result demonstrates the mechanism rather than any particular physical mounting. A physical retrofit would introduce frequency-dependent transmission and mounting nonlinearities the linear mixing does not represent.

Finally, *scope*: severity is recorded and enumerated but not predicted (section 2.5), the fault catalogue is one authored set of six modes, and the platform is a single 6-DoF kinematic structure - generality across catalogues and platforms is asserted only to the extent that nothing in the mechanism is specific to either.

4.7 Implications and future work

Four extensions follow directly from the limitations and results, in rough order of value. The first is *sim-to-real fine-tuning on a physical arm*: the pipeline's ONNX/ROS deployment path was built so that a checkpoint trained in simulation can run against real sensor streams, and the natural experiment is to fine-tune the learned coupling C on a physically instrumented manipulator - not a deliverable of this thesis, but the step that would test whether coupling-agnostic localisation survives real transmission physics. The second is *severity regression as a fourth output*: severity is already recorded in every sample and enumerated by the scheduler, so extending the graph head with a severity regressor is a data-ready change, and the same recorded labels support expanded distribution-shift probes (speed-shift and severity-shift companions to the existing payload probe). The third is a *quantitative on-line evaluation script*, logging `/health_diagnosis` against `/fault_state` to report on-line accuracy and detection latency - the ticks from fault onset to first correct diagnosis - upgrading the qualitative deployment demonstration of section 2.6.6 into a measured one. The fourth is *per-run signature-parameter randomisation*, the full answer to signature circularity: by drawing each session's signature parameters from a distribution, the corpus would force the model to learn fault classes as families of behaviours rather than as fixed waveforms, and would make the reported accuracies claims about fault types rather than about one generator.

5 Conclusion

In considering the question of how structural knowledge of a robot should enter a graph-based diagnostic model, the research undertaken provides an empirical answer through simulation on a 6-DoF manipulator with a fault-rich, fully labelled corpus generated by a purpose-built ROS 2 pipeline. Two candidate mechanisms were posed as open questions: whether conditioning graph connectivity on the robot's configuration improves diagnosis (Q1), and whether the joint–sensor coupling can be learned rather than assumed, so that faults localise correctly even when the sensor placement is unknown to the model (Q2). Nine model variants - non-graph and attention baselines, a 2×2 pose $\times$ transmissibility adjacency ablation, and a learned-coupling anomaly-residual head - were evaluated under a leakage-controlled, run-aware, multi-seed protocol built so that a null result would be as attributable as a positive one.

Evidence produced by this research addressed these two questions asymmetrically. Q1 returned a clean null: the pose-conditioned adjacency improved neither classification nor localisation, a single-seed gain having failed to survive three-seed replication - a negative result reported in full, both because no published precedent existed to test and because the multi-seed rigour that overturned the artefact is itself a methodological lesson. The mechanism is retained for what it

does deliver: a single interpretable scalar, the converged bandwidth γ, estimating how strongly configuration gates inter-component transmission. Q2 was answered strongly in the affirmative. The learned-coupling head localises faults at 0.95 accuracy in-distribution and $\approx$0.79 under sensor placements withheld from the model, where every fixed-graph and non-graph baseline collapses to chance, and it recovers the true coupling at 0.97 correlation - an inspectable, physically meaningful artefact produced as a by-product of training.

Beneath both answers sits the study's most robust finding: graph structure is necessary for localisation but not for classification. The non-graph baseline classifies competitively yet localises at chance, while every graph variant localises above 0.89. In this diagnostic regime, structure buys *spatial attribution* rather than discrimination - it determines whether the system can say where the fault is, not how well it says what the fault is - and attribution is the actionable half of a maintenance diagnosis. The significance for the field follows directly: for GNN-based health monitoring of articulated machines, the productive question is not how to weight a known structural graph, but whether to assume the structure at all. Learning the coupling outright proved more valuable than either physically motivated refinement of the assumed graph, and it is the only approach that survived a broken structural prior - the retrofit scenario every fixed-graph method silently assumes away.

The thesis also leaves behind a reproducible platform: a simulation-to-inference pipeline in which one `GraphSample` schema flows unchanged from data generation through training and held-out evaluation to real-time ONNX inference on ROS 2, with run-aware splitting, seeded replication and per-checkpoint provenance built in. The natural continuations - fine-tuning the learned coupling on a physical arm, predicting severity from the already-recorded labels, and randomising signature parameters to close the circularity gap - are all incremental extensions of this platform rather than new builds, which is perhaps the most practical measure of the pipeline's contribution.

References

[1] Stefan Bloemheuvel, Jurgen van den Hoogen, and Martin Atzmueller. A computational framework for modeling complex sensor network data using graph signal processing and graph neural networks in structural health monitoring. *Applied Network Science*, 6(1):1–24, December 2021. Publisher: SpringerOpen.

[2] Zhiwen Chen, Haobin Ke, Jiamin Xu, Tao Peng, and Chunhua Yang. Multichannel Domain Adaptation Graph Convolutional Networks-Based Fault Diagnosis Method and With Its Application. *IEEE Transactions on Industrial Informatics*, 19(6):7790–7800, June 2023.

[3] Zhiwen Chen, Jiamin Xu, Cesare Alippi, Steven X. Ding, Yuri Shardt, Tao Peng, and Chunhua Yang. Graph neural network-based fault diagnosis: a review, 2021. Version Number: 1.

[4] Zixu Chen, Youcheng Zeng, Wennian Yu, and Jinchen Ji. A Spatial-Temporal Graph Convolutional Network for Small Sample Fault Diagnosis of Rotating Machinery. In *2024 Global Reliability and Prognostics and Health Management Conference (PHM-Beijing)*, pages 1–6, Beijing, China, October 2024. IEEE.

[5] Guimin Dong, Mingyue Tang, Zhiyuan Wang, Jiechao Gao, Sikun Guo, Lihua Cai, Robert Gutierrez, Bradford Campbell, Laura E. Barnes, and Mehdi Boukhechba. Graph Neural Networks in IoT: A Survey, 2022. Version Number: 2.

[6] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs, September 2018. arXiv:1706.02216 [cs].

[7] Ruochen Jiao, Juyang Bai, Xiangguo Liu, Takami Sato, Xiaowei Yuan, Qi Alfred Chen, and Qi Zhu. Learning Representation for Anomaly Detection of Vehicle Trajectories, March 2023. arXiv:2303.05000 [cs].

[8] Gamze Naz Kiprit, Andreas Koch, Michael Petry, and Martin Werner. Graph neural networks for anomaly detection in spacecraft. In *Proceedings of the First Joint European Space Agency / IAA Conference on AI in and for Space (SPAICE 2024)*, October 2024.

[9] Yi Lai, Ye Zhu, Li Li, Qing Lan, and Yizheng Zuo. STGLR: A Spacecraft Anomaly Detection Method Based on Spatio-Temporal Graph Learning. *Sensors*, 25(2):310, January 2025.

[10] Xinming Li, Meng Li, Jiawei Gu, Yanxue Wang, Jiachi Yao, and Jianbo Feng. Energy-Propagation Graph Neural Networks for Enhanced Out-of-Distribution Fault Analysis in Intelligent Construction Machinery Systems. *IEEE Internet of Things Journal*, pages 1–1, 2024.

[11] Pengfei Liang, Xiangfeng Wang, Chao Ai, Dongming Hou, and Siyuan Liu. SRSGCN: A novel multi-sensor fault diagnosis method for hydraulic axial piston pump with limited data. *Reliability Engineering & System Safety*, 253:110563, January 2025.

[12] Yu Liu, Lianzhen Zhang, Jiyu Xin, and Sijie Peng. TPS-GNN: Predictive model for bridge health monitoring data based on temporal periodicity and global spatial distribution characteristics. *Structural Health Monitoring*, page 14759217251330971, April 2025. Publisher: SAGE Publications.

[13] Shanshan Song, Shuqing Zhang, and Xiang Wu. Multi-view graph convolutional networks based on multi-source information fusion for mechanical fault diagnosis. *Structural Health Monitoring*, 24(6):3728–3752, November 2025.

[14] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks, February 2018. arXiv:1710.10903 [stat].

[15] Jiamin Xu, Haobin Ke, Zhaohui Jiang, Siwen Mo, Zhiwen Chen, and Weihua Gui. OHCA-GCN: A novel graph convolutional network-based fault diagnosis method for complex systems via supervised graph construction and optimization. *Advanced Engineering Informatics*, 61:102548, August 2024.

[16] Longda Zhang, Fengyu Zhou, Peng Duan, and Xianfeng Yuan. Fault diagnosis of mobile robot based on dual-graph convolutional network with prior fault knowledge. *Advanced Engineering Informatics*, 62:102865, October 2024.

Appendix

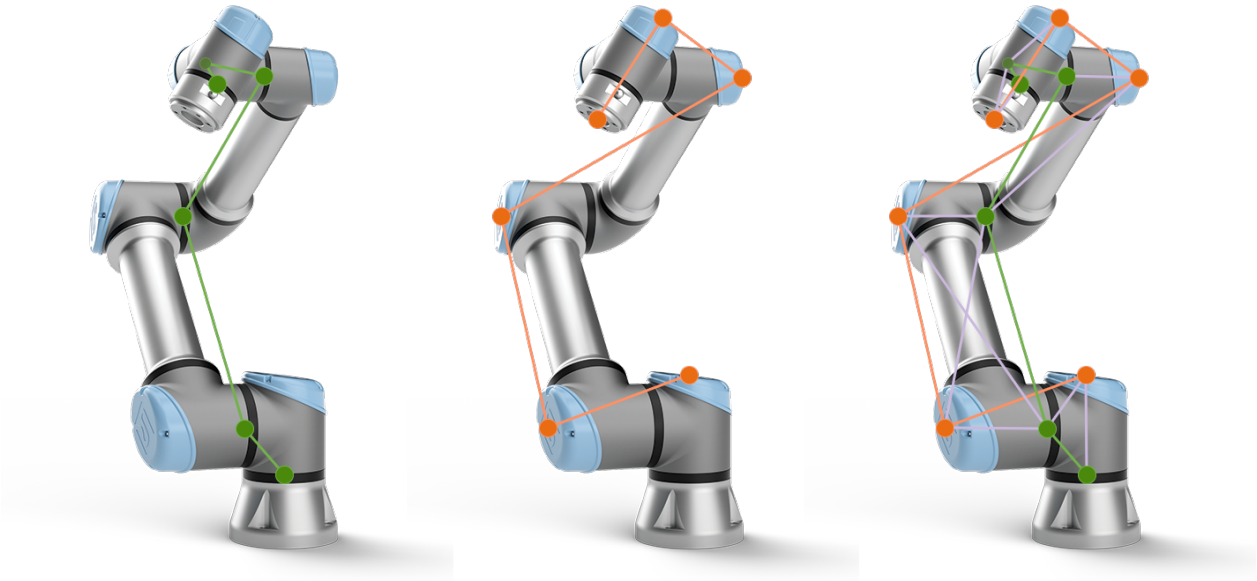

Figure 15: Sample graph construction on UR5e robotic arm; the green graph represents joints, orange represents sensor nodes, purple represents the auxiliary connections between them.

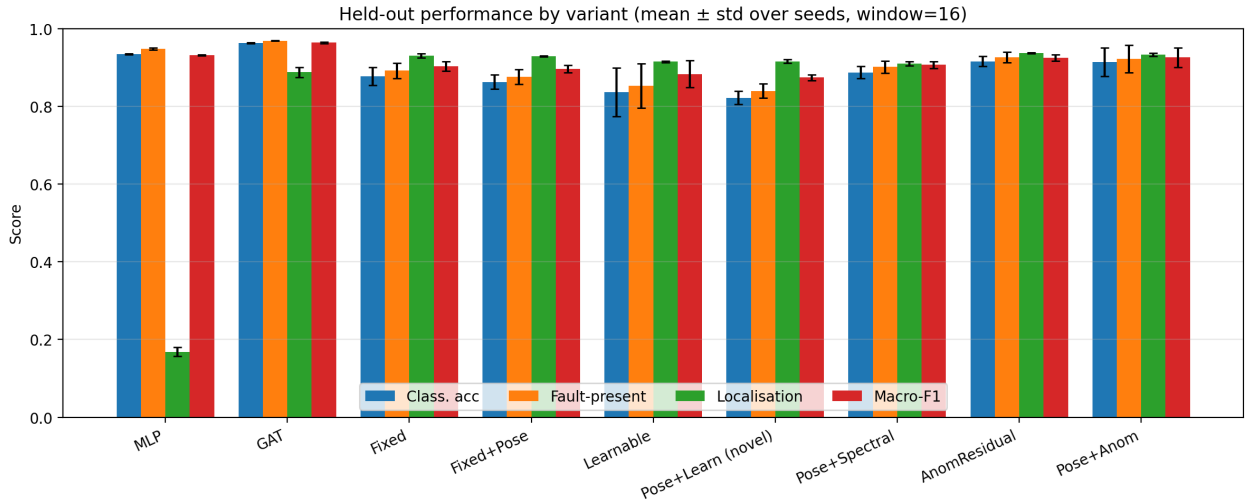

Figure 16: Held-out cls / fault-present / localisation / macro-F1 by variant (mean ± std, w=16).

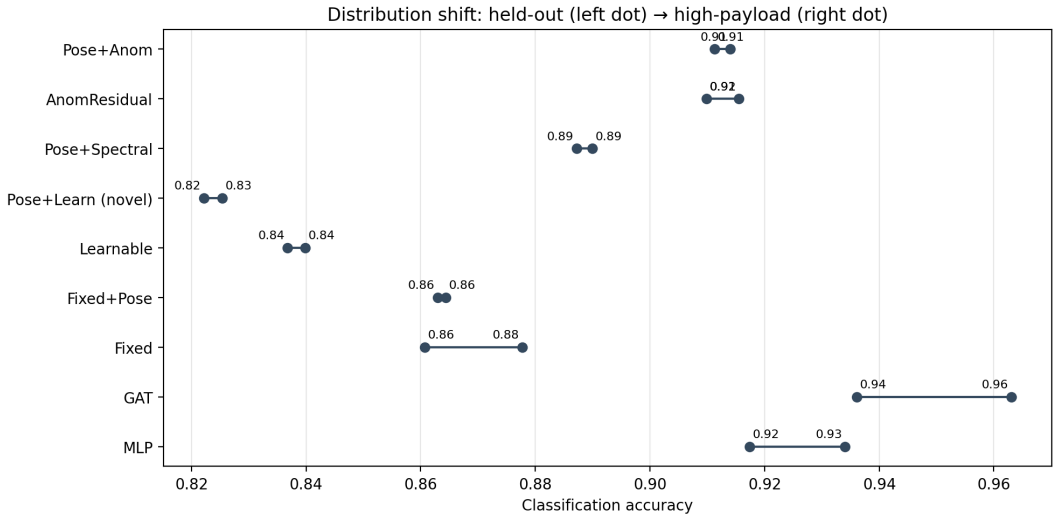

Figure 17: Held-out vs high-payload accuracy per variant.

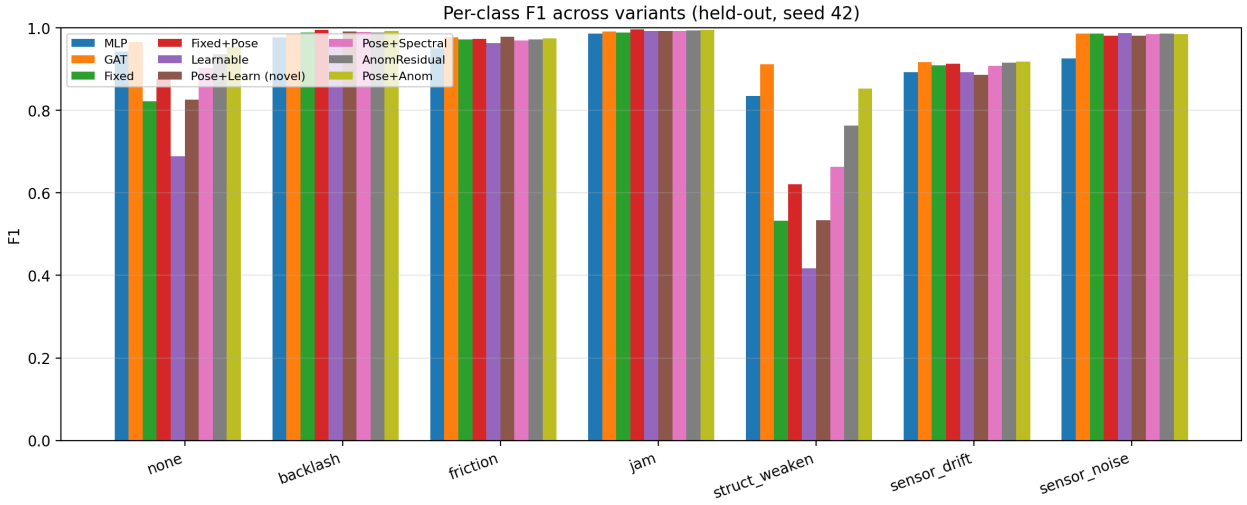

Figure 18: Per-class F1 across variants.

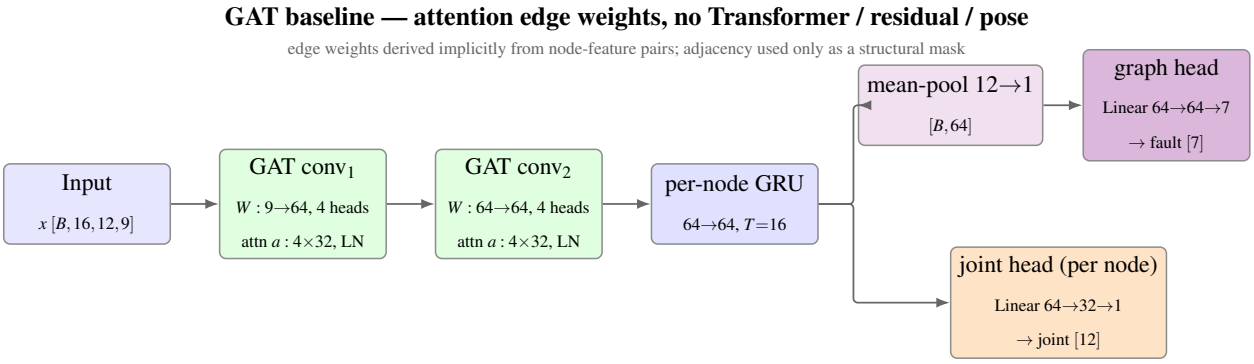

Figure 19: GAT attention (implicit, feature-derived edge weights).

Table 8: Window-size ablation for the two swept variants (mean ± std over 3 seeds, held-out set). Payload is accuracy under high-payload shift; γ is the converged pose-bandwidth. Bold = best window per column within each block. The headline `anomaly_residual` peaks at its operating window w=32 (best on every metric) then degrades at w=48 with rising variance; γ grows with window ($0.31 \rightarrow 0.59$ from w=8 to w=48), indicating stronger configuration-gating over longer temporal context.

Window	Cls. acc	Localisation	Macro-F1	Payload	γ
Headline model: `anomaly_residual`					
$w = 8$	0.891 ± 0.005	0.903 ± 0.004	0.905 ± 0.005	0.891 ± 0.002	–
$w = 16$	0.916 ± 0.013	0.937 ± 0.001	0.925 ± 0.008	0.910 ± 0.008	–
$w = 32$	$\mathbf{0.935 \pm 0.024}$	$\mathbf{0.950 \pm 0.002}$	$\mathbf{0.940 \pm 0.019}$	$\mathbf{0.927 \pm 0.021}$	–
$w = 48$	0.865 ± 0.077	0.947 ± 0.005	0.896 ± 0.039	0.875 ± 0.068	–
Pose-conditioned model (`pose-cond.`), for contrast					
$w = 8$	0.856 ± 0.013	0.878 ± 0.000	0.886 ± 0.005	0.863 ± 0.015	0.31 ± 0.09
$w = 16$	0.822 ± 0.017	$\mathbf{0.916 \pm 0.004}$	0.874 ± 0.007	0.825 ± 0.013	0.36 ± 0.09
$w = 32$	$\mathbf{0.879 \pm 0.025}$	0.912 ± 0.003	$\mathbf{0.906 \pm 0.012}$	$\mathbf{0.884 \pm 0.019}$	0.59 ± 0.17
$w = 48$	0.871 ± 0.047	0.915 ± 0.005	0.900 ± 0.030	0.882 ± 0.031	0.59 ± 0.14

Acknowledgments

Thank you to my supervisor, Dr. Ang Liu, for your initial guidance in the direction of this thesis. I also want to express my appreciation to my friends and mentors, Nadia Pandoulis, Mary Pillay, and Eben Steenekamp for their advice and encouragement throughout the development of this report.