Babylonian Journal of Machine Learning
Vol.2026, **pp**. 197–234
DOI: https://doi.org/10.58496/BJML/2026/014; ISSN: 3006-5429
https://mesopotamian.press/journals/index.php/BJML

Research Article

When GPUs Fail Quietly: Separating GPU Hardware Degradation from Monitoring Failure under Degraded Observability

Michael Bidollahkhani [1,*], Mohsen Seyedkazemi Ardebili [2], Andrea Bartolini [2], Rute C. Sofia [3], Julian Martin Kunkel [4]

[1] *Institute of Computer Science, University of Göttingen, Göttingen, Germany.*

[2] *Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi" (DEI), University of Bologna, Bologna, Italy.*

[3] *fortiss GmbH, Research Institute of the Free State of Bavaria for Software-Intensive Systems, Munich, Germany.*

[4] *Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG), Göttingen, Germany.*

ARTICLE INFO

Article History
Received 18 Jul 2026
Revised 22 Aug 2026
Accepted 25 Sep 2026
Published 31 Oct 2026

Keywords
causal diagnosis,
GPU reliability,
high-performance computing,
monitoring failure,
observability,
partial observability,
predictive maintenance,
structured missingness,
telemetry,
weak supervision.

CC BY

ABSTRACT

A missing GPU measurement in a production supercomputer is ambiguous: either the device has failed, or the pipeline that reports on it has. The two call for different responses – an engineer sent to the hardware, or a ticket raised against the collector – so the ambiguity has to be resolved before either one is right. We ask which half of that decision needs a model. Whether the monitoring path has degraded is close to directly readable: one lightweight statistic that is never fitted, the gap between the data-delivery rates on the out-of-band hardware-sensor path and the in-band host path, ranks monitoring incidents at AUPRC 0.237 where they are only 5.2 % of the units ranked, and no learned component we trained improves on it. Deciding whether already-degraded evidence has a physical cause is harder, and needs both workload-aware residuals on the hardware sensors and an explicit model of the observation process. PARALLAX supplies these, tracking device health and monitoring health as separate latent states over out-of-band IPMI measurements, in-band telemetry, workload context, and structured missingness; the configuration we recommend is the hybrid that takes its physical risk from PARALLAX and its monitoring risk from the lightweight statistic. On 29 months of a 988-node production cluster, restricted to a pre-specified subset whose evidence has already degraded so that nothing can score by detecting absence, it reaches macro-F1 0.455 against 0.349 for the strongest classical detector and AUPRC 0.603 against 0.184: the four-way verdict is right more often across all four states, and a worklist ranked by physical risk holds far more genuine faults near its top, which is what an operator with a few inspections a day actually spends. These are triage figures as absolute precision falls towards the class base rate beyond moderate recall, so AUPRC supports ranking near the top of a worklist rather than an unattended alarm, and both numbers are measured against constructed weak labels on one system. In practice that detector files 81 % of real hardware failures as monitoring problems, four in five sent to the wrong queue, and the hybrid cuts this to 20 % while escalating no healthy node. The coupled state, in which both causes act at once, is too rare in this corpus to be resolved and is left open. The contribution is a boundary rather than a system: use simple, interpretable indicators for monitoring health, and reserve explicit latent modelling for the regime in which observability itself has already degraded.

1. INTRODUCTION

A GPU-enabled HPC system is watched through a stack of software and hardware layers that interact with one another. What an operator knows about the state of a node comes from device counters, board temperatures, power readings, PCIe indicators, host utilization, network activity, the records of the workload manager, and periodic service checks, taken together. Failure detection, predictive maintenance, capacity planning, and triage all rest on these measurements. Reading them, however, relies on a premise that is seldom spelled out: the monitoring system itself has to stay healthy enough that what it reports still stands for the physical machine.

The premise does not hold reliably. When a GPU metric goes missing, the device may have dropped off the driver, Ganglia collection may have failed, the node may be unreachable for a while, the metric configuration may have changed, a message may be late, or a larger incident may have hit both the node and its collector at once. A flat temperature trace may mean

*Corresponding author. Email: michael.bkhani@uni-goettingen.de

stable hardware, but it may equally mean a frozen exporter or a value that is no longer being refreshed. A sudden drop in the number of metrics reported can follow from hardware detachment, from a fault on the in-band collection path, or from routine maintenance. Every observation is thus the joint product of two systems, the physical compute system and the one that watches it.

The classical literature on online failure prediction looks for temporal precursors in logs, counters, and environmental readings [1], [2], [3], [4]. Work specific to GPUs adds that how an error shows up depends on the architecture, the operating conditions, the workload, and the mechanism behind the error [5], [6]. More recently, operational studies have treated the disappearance of metrics and the degradation of the monitoring pipeline as evidence in their own right for GPU detachment and related faults [7]. Taken together, these results leave an attribution problem open. The structure of telemetry loss carries information, but as long as a single anomaly score stands for both a failing collector and a failing device, its cause cannot be assigned with any safety.

The M100 ExaData corpus is an unusually suitable setting in which to study this problem. It contains 934 days of telemetry from more than 980 compute nodes, 573 metrics, workload and job information, out-of-band baseboard management controller (BMC) measurements, in-band host measurements, and Nagios service states [8]. IPMI samples expose physical temperatures, including per-GPU core and memory temperatures and GPU card presence sensors, together with power, fan, PCIe, and power-supply measurements through an out-of-band path. Ganglia samples expose CPU, memory, load, process, disk, and network measurements through an in-band path driven by an agent that runs on the node itself. The job table and Slurm aggregates provide workload context. Nagios states, produced by an active network poller that is operationally separate from the passive telemetry publishers, provide weak but independent anchors for hardware, GPU, batch, network, and service conditions. Decisively for this study, the three collection systems fail in different ways and share only part of their infrastructure, which is what allows the monitoring pipeline to be modeled as an observable subsystem with its own operational state.

We select the research problem of *separating hardware failure from monitoring failure under explicit structural assumptions*, and the question we ask of it is where that separation needs machinery. We use the term causal deliberately and conservatively: we do not claim unrestricted causal identification from observational telemetry, we define an explicit structural causal model, state the exclusion and anchor assumptions required for attribution, and evaluate falsifiable consequences of those assumptions. Where a score is produced by substituting a learned latent rather than by an identified intervention, we say so and read it operationally (Section 5.7). The central idea is that out-of-band and in-band measurements respond differently to physical and monitoring states. If the hardware is degrading, at least one physical measurement or hardware-specific service anchor should move even once workload has been controlled for. If only the in-band monitoring path has failed, what changes is availability, cadence, or cardinality, while the out-of-band physical plane, conditionally on workload, stays where it was. When the two are coupled, both may change.

The proposed framework, PARALLAX, learns separate latent representations for physical health and observability health. The name is borrowed from optics: parallax recovers a depth that no single line of sight contains, using only the disagreement between two vantage points looking at the same object. Here the two vantage points are the out-of-band and in-band measurement paths, and the quantity recovered from their disagreement is the cause of an observation that either path alone would leave ambiguous. It combines five mechanisms: workload-conditioned physical residuals, explicit missingness and cadence features, cross-layer graph fusion, weak-label supervision from service-specific Nagios states, and reference-substitution scoring. Its output contains two calibrated risks: physical risk and monitoring risk. This output supports four operational states and allows nodes to be ranked by physical risk even when the model abstains from exact failure-time prediction. Of the four states, the coupled one is too rare in this corpus to be evaluated, and it is treated as unresolved throughout (Section 7.3). As Section 7 shows, only part of that machinery earns its place: the monitoring risk is better taken from a single unfitted coverage statistic, and it is the physical branch, under degraded observability, that the structural model is needed for. The configuration we report is that composition.

The question this paper answers is therefore not whether a four-state causal attribution system can be built, but *where along this problem an explicit model of the observation process earns its cost*. Our contributions are:

- **An empirical boundary.** We show that the two directions of the attribution problem are not of the same kind in this setting. Monitoring degradation is close to directly observable: one unfitted cross-path coverage statistic ranks monitoring incidents at AUPRC 0.237 against a prevalence of 0.052, and no learned observability head in the study beats it. Deciding whether already-degraded evidence has a physical cause is not directly observable, and there explicit workload-conditioned out-of-band modelling is what separates the methods.
- **A hybrid configuration that follows from the boundary** and is reported as the study's main result: physical risk from PARALLAX, monitoring risk from the unfitted asymmetry statistic, calibrated and thresholded through the same protocol as every other method.

- **An evaluation protocol built around the evidence-degraded control subset**, on which no method can succeed by detecting absence alone, together with event-level attribution, calibration, false maintenance escalation, false monitoring dismissal, cross-node and cross-era generalization, and controlled sensor-disappearance tests. Each deep model is given the same hyper-parameter-search budget and its own threshold grid, so that no comparison depends on an operating point that was unfair to one side.
- **An architectural failure, diagnosed and repaired.** In the first version of this study [7], the jointly trained dual-latent network hardly ever emitted the monitoring class. We trace the cause to three concrete defects, fix them, and report the repaired network next to the original, so that whether a dual-latent model can separate both causes is settled on evidence, one way or the other, rather than taken for granted.
- **An independent check of the constructed monitoring labels**, using three witnesses that play no part in building the anchors: continuity of Slurm workload, out-of-band physical residuals, and the drain actions operators took afterwards, each compared with a matched null.
- **A structural formulation of the four-state problem, together with a weak-label construction that can be reproduced**: identification assumptions stated explicitly, Nagios anchor families weighted by confidence, interval-censored events with guard bands, and a single reconciliation of every population count the study quotes. The four-state space is a design choice whose coupled corner this corpus cannot validate; the evidence supports claims about the hardware and monitoring directions only.

The rest of the paper runs as follows. Section 2 places the study against prior work on HPC failure prediction, GPU reliability, telemetry infrastructure, and learning under structured missingness. Section 3 states the causal attribution problem and the assumptions behind it, and Section 4 describes the M100 data planes and how events are constructed from them. PARALLAX itself is presented in Section 5, the experimental protocol in Section 6, and the empirical findings in Section 7. Section 8 closes with interpretation, the limits within which the method can be deployed, and threats to validity.

2. RELATED WORK and RESEARCH GAP

2.1 Failure Prediction in HPC Systems

Online failure prediction takes observations made at run time and estimates from them the probability, location, type, or timing of a failure still to come [1]. In HPC, that programme has been pursued on system logs, environmental sensors, management interfaces, and event sequences drawn from large production machines [2], [3], [4]. FTB-IPMI wired hardware monitoring into a fault-tolerance backplane and showed what management-interface data can do for proactive mechanisms [9]. The lesson of this body of work is that failure evidence is heterogeneous, dependent over time, and heavily imbalanced.

Nearly all of these predictors take the monitoring stream at face value, as a faithful picture of the system under study. Records that are missing are usually dropped, imputed, folded into an aggregate, or read as a nonspecific anomaly. None of this does harm so long as collection failures are independent of the state being predicted. It stops being safe once the observation process has failure modes of its own, or once a physical failure alters whether the device can be observed at all. In that regime a predictor is free to learn the collection topology, data-retention artifacts, or maintenance policy instead of physical degradation, and to hide the latter behind the former.

2.2 GPU Reliability and Workload Dependence

Studies of GPUs at scale have catalogued memory errors, vulnerabilities tied to particular architectures, spatial clustering of faults, and links between temperature, power, workload, and soft-error behaviour [5], [6]. One consequence is that a high temperature or power reading means nothing on its own; it has to be read against utilization, clock state, where the job was placed, and the node's own baseline. Telling an expected operating response from genuine degradation therefore calls for models conditioned on workload.

A different family of GPU incidents gives little or no numeric warning. A driver or interconnect failure can take a device out of the software stack while its temperature, utilization, and power all read as normal right up to the event. For incidents of this kind the main evidence is that device metrics vanish and the telemetry payload thins out [7]. A monitoring failure, however, produces exactly the same structural signature. What has not been done is to attribute that signature to one cause or the other.

2.3 Relation to Our Earlier Work

Our earlier study [7] established that structural telemetry loss is predictive evidence: it built a detector for quiet GPU failures in which metric disappearance and payload degradation, rather than threshold crossings on values, carry the signal. It deliberately treated all disappearance as evidence *about the node*, and it neither modelled nor evaluated the possibility that the collection path itself had failed. Its unit of analysis is a single anomaly score, its label plane is the same one that supplies its features, and it reports detection performance, not attribution.

This paper picks up where that one left off. Its subject is precisely the ambiguity the earlier work set aside: evidence is missing, so decide *why*. Changing the question changes four things, none of which is present in the earlier study. The state space has four values instead of two. The observability state is supervised from a held-out collection plane that is kept out of the feature table. The primary comparison is confined to windows whose evidence is already degraded, so that noticing absence earns nothing. And what is reported at the end is a boundary between regimes, not a detector. There is also a practical disagreement between the two papers that only the second was in a position to see: in this corpus, when the earlier detector's main signal, disappearance, fires, the cause is more often a faulty collector than a faulty device, and that confusion is the very thing the present formulation is built to keep under control.

2.4 Telemetry Infrastructures as Fallible Systems

Monitoring frameworks for HPC such as Seastar and ExaMon gather heterogeneous operational telemetry into a form that can be analysed at scale [10], [11], and Mantis goes further, towards scalable integration and representation learning over very large telemetry volumes [12]. Without such systems, analytics across several planes would not be possible at all. Yet the analyses built on them usually leave the health of the monitoring layer implicit. Here, by contrast, monitoring health is a second dynamical state, and one that has to be inferred.

That failure detectors are unreliable, and that dependable decisions must nonetheless be reached from suspicions that are incomplete, late, or simply wrong, has been understood in distributed systems for a long time [13]. Maintenance driven by telemetry is subject to the same principle. An alarm is itself an observation, and it arrives through a path that can fail. Diagnosing anything operationally therefore needs two models, one of the component being observed and one of how good the observation is.

2.5 Structured Missingness and Multivariate Anomaly Detection

GRU-D brought explicit masks and inter-observation intervals into recurrent models so that informative missingness could be put to use [14]. The wider literature on structured missingness shows that which variables go missing can be correlated across modalities, acquisition processes, and populations, and that this correlation carries information the usual missing-at-random assumptions throw away [15]. SAITS is a strong self-attention method for imputing multivariate time series [16]. For HPC observability, though, imputation by itself can erase the very signal one is after: a model has to keep both its estimate of the physical value and the fact that the value stopped arriving.

Isolation Forest, GDN, and TranAD are useful multivariate baselines for value-based, graph-based, and transformer-based detection respectively [17], [18], [19]. Each of them ends in one anomaly score, and one score cannot by itself say whether it is the hardware or the measurement process that has degraded. Weak supervision offers a way to turn operational rules and imperfect service states into probabilistic training targets [20]; for that to be sound here, the label model has to keep its uncertainty and keep operator actions out of what counts as root-cause truth.

2.6 Research Gap

Mature components exist for each piece of the problem: failure prediction, GPU-error analysis, telemetry collection, modelling of missing data, and multivariate anomaly detection. Where they meet, a specific gap remains: *the system being monitored and the system doing the monitoring are rarely treated as two separate causal sources of the telemetry that is observed*. The present paper addresses that gap with a correspondingly specific attribution objective: to decide, under assumptions that are stated outright, whether the evidence points to a physical cause, an observability cause, both at once, or no actionable degradation at all.

3. PROBLEM FORMULATION

3.1 Operational State Space

Consider a node observed over discrete windows $t = 1, \dots, T$. Two latent states are attached to it, each either binary or continuous:

- H_t, the physical health of the GPU-enabled node, or the intensity of its degradation;
- O_t, the health of the monitoring and collection path, or the intensity of its degradation.

From these the operational attribution state is

$$S_t \in \{\text{normal,hardware,monitoring,coupled}\}.$$

When a binary decision is required, the four states correspond to $(H_t, O_t) \in \{(0,0), (1,0), (0,1), (1,1)\}$; when nodes are to be ranked by risk, H_t and O_t are kept continuous and calibrated instead.

Terminology. The paper uses the following terms with fixed meanings. *Physical degradation* and *hardware failure* refer to the state H_t of the GPU-enabled node itself, and *hardware* is the corresponding attribution class. *Monitoring failure*, *monitoring-health degradation*, and *observability degradation* are used interchangeably for the state O_t of the collection and reporting path, and *monitoring* is the corresponding class; none of these terms says anything about the node. A *coupled event* is a window in which both states are degraded, whether or not one caused the other. *Evidence-degraded* describes a window whose telemetry delivery is low relative to the rest of the machine (Section 6.7), whatever its class. Wherever the text says *causal attribution* or *causal separation*, it means attribution under Assumptions A1–A4 of Section 3.3; no unrestricted causal identification from observational telemetry is claimed anywhere in the paper, and the PARALLAX outputs are read as calibrated risks under those assumptions, not as identified causal effects. The observed data for node n and window t are

$$D_{n,t} = \{X^o_{n,t}, X^i_{n,t}, M_{n,t}, W_{n,t}, A_{n,t}\},$$

where X^o denotes out-of-band physical measurements, X^i denotes in-band measurements, M is the availability and missingness structure, W is workload and scheduling context, and A contains weak operational anchors from Nagios. The task is to estimate

$$(r^H_{n,t}, r^O_{n,t}, \hat{S}_{n,t}, u_{n,t}) = F_\theta(D_{n,t-L+1:t}),$$

where r^H is hardware risk, r^O is monitoring risk, $\hat{S}$ is the four-state attribution, and u is predictive uncertainty.

3.2 Structural Causal Model

We use the structural model in Fig. 1. Let $X^{i,*}_t$ denote the in-band physical values that would be measurable through a healthy collection path. The observed in-band values are filtered by M_t.

$$\begin{gathered}
W_t = f_W(W_{t-1}, U^W_t), \\
H_t = f_H(H_{t-1}, W_t, C_t, U^H_t), \\
O_t = f_O(O_{t-1}, H_t, C_t, U^O_t), \\
X^o_t = f_o(H_t, W_t, C_t, U^o_t), \\
X^{i,*}_t = f_i(H_t, W_t, C_t, U^i_t), M_t = f_M(O_t, H_t, C_t, U^M_t), \\
X^i_t = M_t \odot X^{i,*}_t, A^H_t = f_{AH}(H_t, U^{AH}_t), \\
A^O_t = f_{AO}(O_t, U^{AO}_t).
\end{gathered}$$

Here, C_t includes node identity, rack, calendar time, metric-configuration period, and facility context. The edge $H_t \to O_t$ allows physical degradation to damage observability. This edge is essential for coupled events. The absence of an edge $O_t \to X^o_t$ expresses the relative exclusion assumption for selected out-of-band BMC values.

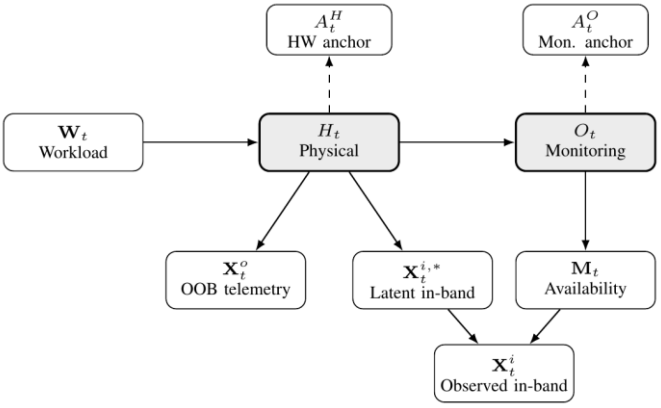


Fig. 1. Structural causal flow behind the observability separation. OOB signals (out-of-band, e.g. IPMI/BMC parameters) and the latent in-band signals (measurements collected by software running on the node itself, e.g. Ganglia or node-OS-derived parameters) are driven by the physical state; whether the in-band signals are available is governed by the monitoring state. Weak supervision comes from the hardware and monitoring anchors.

Fig. 1 distinguishes the latent physical and monitoring states from the measurements generated through the out-of-band and in-band paths. Selected OOB BMC measurements are not descendants of the monitoring state, expressing the relative exclusion assumption used for attribution. Shared context C_t is omitted from the diagram for readability, and the causal interpretation remains conditional on the assumptions stated below.

3.3 Identification Assumptions

The following assumptions define the boundary of causal interpretation.

A1: Relative out-of-band exclusion. After conditioning on workload and measured context, an isolated failure of the selected in-band collection path does not directly change the underlying BMC temperature, power, fan, PCIe-temperature, or power-supply values. Shared network, BMC, facility, and maintenance incidents can violate this assumption and must be modeled as ambiguous or coupled.

A2: Anchor specificity. Certain Nagios service descriptions, although imperfect, point in a definite direction. nvidia::memory::retirement, nvidia::memory::replace, bmc::events, and sys::rvitals, for instance, are read as physical anchors. monitoring::health is treated as an observability anchor. Generic reachability and batch-client states are not treated as physical ground truth.

A3: Measured-context sufficiency. The model conditions on the major observed common causes available in ExaData, including workload, node, rack, time, and metric-configuration period. Unrecorded maintenance actions and facility incidents remain possible confounders.

A4: Temporal consistency and positivity. Each combination of anchor-defined states occurs often enough in the data to be learned from, and the context relevant to a cause comes before, or overlaps, the observations used to attribute it. Windows that follow an operator drain are kept out of the predictive features except where an experiment is explicitly about diagnosis after action.

Stated this way, the assumptions leave the uncertainty of the weak operational labels intact and turn the attribution claim into something that can be tested. The evaluation accordingly contains sensitivity analyses that drop the out-of-band plane, vary anchor confidence, inject collection failures, and single out periods during which the metric configuration was stable.

3.4 Research Questions and Hypotheses

Four research questions are asked.

RQ1: Attribution. Do separate physical and monitoring risk heads tell hardware, monitoring, coupled, and normal windows apart more accurately than a unified anomaly detector does?

RQ2: Context and anchors. How much of the attribution is owed to out-of-band measurements, to workload conditioning, to structured missingness, and to service-specific anchors?

RQ3: Robustness. Is the causal separation stable when points go missing, when gaps form blocks, when a metric family or a whole plugin disappears, and when the model is moved to other nodes?

RQ4: Operational utility. Does separating the two causes lower false hardware escalation and false monitoring dismissal without giving up event-level recall or a calibrated risk ranking?

The corresponding hypotheses are:

- **H1:** Cross-path asymmetry between conditionally stable OOB measurements and degraded in-band availability improves monitoring-failure identification.
- **H2:** Workload-conditioned residuals reduce false physical alarms caused by expected thermal and power response.
- **H3:** Explicit dual-state modeling reduces hardware-versus-monitoring confusion compared with value-only and single-score baselines.
- **H4:** A coupled state is required; forcing all events into hardware or monitoring classes increases both operational error rates.

4. M100 EXADATA and EVENT CONSTRUCTION

4.1 Dataset Scope

M100 ExaData records the operation of the CINECA Marconi100 Tier-0 system from 9 March 2020 through 28 September 2022. The public corpus covers 934 days, more than 980 compute nodes, 573 metrics, and 49.9 TB before compression [8]. The data are partitioned by month, plugin, and metric. Monitoring coverage and the available metric set evolve over time, which makes configuration-era control necessary.

The complete public release is distributed across twelve Zenodo records [21]. The experimental artifact should record every source DOI, extracted month, plugin, metric, node set, row count, checksum, and transformation. This requirement prevents silent changes in the study population when only a subset of the 49.9 TB corpus is processed.

4.2 Data Planes

Table I maps the available data to its causal role. The IPMI plane is particularly important because it uses the BMC out-of-band interface and samples 104 node-level physical variables at approximately 20 s, including per-device GPU core and memory temperatures and GPU card presence sensors. The Ganglia plane contains 33 in-band host, memory, process, load, disk, and network measurements with variable sampling periods.

An audit of the public release, reported in Section 6, establishes one fact that shapes the whole design: the released Ganglia plane does *not* contain a GPU metric group. Device-level ECC, XID, retired-page, and NVLink counters are not exposed as in-band time series. GPU-specific evidence therefore reaches the model through two other routes: out-of-band GPU thermal and presence sensors on the IPMI plane, and GPU service checks on the Nagios plane. We regard this as a fact about the corpus rather than a weakness of the formulation: what the separation argument needs is two collection paths that fail in different ways, not any particular list of sensors. Wherever a claim would rest on device counters the release does not hold, we say so explicitly rather than stand in a proxy for them.

In the job table each job carries its node allocation, the resources it requested, its state, its start and end times, its exit code, and its trackable resources, with the allocated node set stored as a list of anonymised node identifiers. Nagios reports a state every 15 min for each node and service description, with a flag for hard versus soft state and a flag recording whether an operator drained the node.

TABLE I. EXADATA PLANES USED BY PARALLAX AND THEIR CAUSAL ROLE, AS VERIFIED BY THE RELEASE AUDIT OF SECTION 6.1. THE FINAL COLUMN GIVES THE NUMBER OF METRICS SELECTED FOR THIS STUDY OUT OF THE NUMBER PUBLISHED ON THAT PLANE. NAGIOS IS USED EXCLUSIVELY FOR LABELS AND NEVER ENTERS THE FEATURE TABLE.

Plane	Source and collection path	Variables used	Role in separation	Nominal cadence	Metrics used
Out-of-band physical	IPMI/BMC, polled over the management network independently of the node operating system	Per-GPU core and memory temperature, GPU card presence sensors, CPU core-voltage temperature, CPU/memory/IO power, DIMM temperature, PCIe temperature, fans, PSU input and output values, total node power, ambient temperature	Relative physical anchor; expected to remain independent of an isolated in-band collection failure once workload and context are controlled	About 20 s	56 of 104
In-band host	Ganglia gmond agent running on the node itself	CPU user/system/idle/wait/steal, load, free/cached/buffered/shar	Physical and activity evidence observed through a failure-prone in-band path; its	20 s to 5 min, variable	22 of 33

Plane	Source and collection path	Variables used	Role in separation	Nominal cadence	Metrics used
		ed memory, swap, bytes and packets in and out, running and total processes, disk and partition use, boot time, clock speed	availability is the primary observability signal		
Workload	Slurm job table and cluster aggregates	Allocated node set, allocated CPUs and GPUs per node, job age, job starts and terminations, job state and exit code	Confounder control and operating-regime description	Event records and aggregate series	job table + 20
Operational anchors (labels only)	Nagios, an active network poller separate from the passive publishers	GPU memory retirement and replacement, BMC events, remote vitals, GPU configuration, InfiniBand status, physical-memory total, reachability, batch client, filesystem states, drain flag	Confidence-weighted weak labels; observability anchors from the poller's own delivery structure; coupled-event construction; post-hoc action indicator	15 min	16 compute services

4.3 Temporal Alignment and Configuration Eras

Because the native sampling cadence differs from one plugin to another and drifts over time, one global resampling rule would throw away information that matters. Instead, for every node n, metric family f, and configuration era e we first estimate an expected cadence $\hat{\delta}_{n,f,e}$ as the robust median of the spacing between consecutive timestamps in the training data. A new era begins wherever the expected cardinality of a family, its sampling cadence, or its metric schema changes persistently across the whole system. Windows that span an era boundary are left out of the primary experiment and kept aside for a separate analysis of domain shift.

Every timestamp is then mapped onto fixed analysis windows of length L_w and stride L_s. Raw values keep their source cadence up to the point where window features are computed. Counter metrics become nonnegative increments, with resets detected along the way: a counter that goes down is read as a reset or a device reinitialization, never as a negative error count. Both the aggregated values and the timing structure reach the model.

4.4 Nagios Anchor Taxonomy

Nagios provides host and service states with values OK/UP, WARNING, CRITICAL/DOWN, and UNKNOWN/ UNREACHABLE, together with a soft or hard state indicator. These states are operational checks, not maintenance-confirmed root causes. We therefore divide service descriptions into high-specificity physical anchors, observability anchors, boundary or ambiguous anchors, and excluded services (Table II).

The service inventory of the released corpus determines what this taxonomy can contain, and the audit produced one result that had to change the design. Sixteen service descriptions are attached to the compute host group (a seventeenth, batchs::manager::state, appears in a single month and is excluded), and they include the physical checks the formulation needs: nvidia::memory::retirement, nvidia::memory::replace, bmc::events, sys::rvitals, nvidia::configuration, net::ib::status, and memory::phys::total. The monitoring-pipeline check monitoring::health, however, is attached only to two management hosts. It does not exist per compute node and therefore cannot serve as a node-level observability anchor.

4.5 Observability Anchors from a Held-Out Collection Plane

We resolve this by using the structure of the Nagios plane itself. Marconi100 is instrumented by three collection systems with different failure modes: the BMC path, which is polled out of band; the Ganglia agent, which runs in band on the node; and Nagios, which actively probes the node from a separate poller and writes one record per service every 15 min. The first two are given to the model as features. The third is reserved for labels and never enters the feature table.

The key observation is that Nagios reports a node that is down: an unreachable node produces a CRITICAL alive::ping record, not an absent record. The *absence* of every Nagios record for a node therefore indicates that the observation path failed, not that the node failed. We use this as the node-level observability anchor, with two safeguards against confusing it with decommissioning. First, the node must be demonstrably alive on both sides of the gap: the last record before and the first record after must show OK reachability. Second, the gap must exceed a per-month cut of $max(4, \lceil q_{99.5} \rceil)$ windows,

where $q_{99.5}$ is the 99.5th percentile of the poller's own per-node gaps in that month – the realised cut was 4 windows (1 h) in every month – and last at most 96 windows, so that permanent removals are excluded. A gap is attributed to the node rather than to the poller only when at least 50% of the population is reporting throughout it; a run of more than 4 windows in which fewer than 50% report is instead marked as a cluster-wide monitoring outage. A third, node-level component is added at assembly time: a node whose relative in-band coverage falls below 0.20 of the concurrent cluster median for at least two consecutive windows, while that median is itself at least 0.10, the liveness probe still answers, and no physical anchor fires within Δ_c, is marked alive-but-unobservable with weight 0.9. The management-host monitoring::health check is carried on the label side but never used as an anchor, because it is chronically non-OK; since it is operator-visible, it is used only in Section 7.2 to test whether the cluster-wide anchor identifies incidents that operators recorded independently. This construction keeps the observability label outside the model's input space. The model must infer that the monitoring layer is failing from the IPMI, Ganglia, and workload planes alone, which is the question the paper asks.

Two kinds of evidence must be kept apart on the monitoring side. The observability anchor just described is a *constructed weak label*: the corpus contains no per-node monitoring-health ground truth, so the label is derived from the delivery structure of a held-out plane under stated rules, and it can be wrong. The witnesses of Section 7.2 – workload continuity, out-of-band residuals, and operator drain actions – are *independently observed evidence* used only to validate that label indirectly; they never enter its construction and are never themselves treated as ground truth. Every monitoring-side metric in the paper is therefore measured against the constructed label, and the validation says how far that label can be trusted, not that it is exact.

TABLE II. THE FROZEN, CONFIDENCE-WEIGHTED ANCHOR TAXONOMY. SERVICE COVERAGE COUNTS THE MONTHS IN WHICH A DESCRIPTION APPEARS ON THE COMPUTE HOST GROUP, OUT OF THE 29 MONTHS FOR WHICH THE NAGIOS PLANE EXISTS; EVERY SERVICE, ONCE IT APPEARS, IS PRESENT WITHOUT INTERRUPTION UNTIL THE END OF THE CORPUS. A WEIGHT OF ZERO MEANS THE EVIDENCE IS KEPT ON RECORD BUT NEVER SERVES AS GROUND TRUTH.

Class	Description and treatment	Weight	Mon.
Physical, high	nvidia::memory::retirement, nvidia::memory::replace	1.0	26
	bmc::events, sys::rvitals	1.0	21
Physical, medium	nvidia::configuration	0.8	29
	memory::phys::total	0.7	23
	net::ib::status	0.6	24
Observability	Node-specific loss of the Nagios plane with confirmed liveness on both sides of the gap; qualifies in no month of this corpus (Section 7.2)	1.0	all
	Cluster-wide loss of the Nagios plane	1.0	all
	Alive-but-unobservable: relative in-band coverage below 0.20 for at least two windows while the liveness probe answers and no physical anchor is within Δ_c	0.9	all
	monitoring::health on management hosts, carried for the validation of Section 7.2 only	–	29
Boundary	alive::ping, ssh::daemon	0.4	29
	batchs::client, batchs::client::state	0.2–0.3	29
	sys::gpfs::status, filesys::*	0.2	29
Action only	nagiosdrained; operator response, never physical ground truth	–	29
Excluded	Login, service, switch, storage, and cluster-summary checks	–	–

Let $q_t^H, q_t^O \in [0,1]$ be physical and observability anchor confidence. A hard non-OK state receives its class weight. A soft state receives a reduced weight $\eta_s < 1$. UNKNOWN states are retained as observability evidence only when the description and surrounding coverage support that interpretation. Multiple anchors within a window are combined using a noisy-OR rule,

$$q_t^k = 1 - \prod_{j \in A_k(t)} (1 - w_j s_j), k \in \{H, O\},$$

where w_j is source confidence and s_j is state severity.

4.6 Interval-Censored Event Construction

The 15 min Nagios cadence means that event time is interval-censored. For each transition from OK to a non-OK hard state, the event interval is bounded by the last healthy sample and the first unhealthy sample. We define a pre-event context region, an attribution region centered on the interval, and a post-action exclusion region. The exact values of these intervals are fixed before any model comparison and listed at the end of this section.

A window is labeled *hardware* when $q_t^H \geq \kappa_H$ and $q_t^O < \kappa_O$. It is labeled *monitoring* when $q_t^O \geq \kappa_O$ and $q_t^H < \kappa_H$. When both exceed their thresholds within a coupling tolerance Δ_c, the window is labeled *coupled*. (The anchor-confidence thresholds κ_H, κ_O act on the label plane and are distinct from the decision thresholds τ_H, τ_O that Section 5.7 applies to calibrated risks.) A normal window must be separated from every non-OK anchor, schema transition, and maintenance/drain action by a guard interval Δ_g. Whatever windows remain are unlabeled: they may take part in reconstruction training but are left out of every supervised attribution metric. In the reported experiments $\kappa_H = \kappa_O = 0.5$, $\Delta_c = 4$ windows (1 h), $\Delta_g = 8$ windows (2 h), and non-normal windows separated by no more than four quiet windows belong to one event. These values are fixed before any model comparison.

For reproducibility, the complete label-construction procedure is the following. (i) Aggregate every plane to the 15 min Nagios grid, with analysis windows of $L_w = 15$ min and stride $L_s = 15$ min, so that one node-window is one grid cell. (ii) For each node-window compute q_t^H and q_t^O by the noisy-OR rule above over the frozen taxonomy of Table II, with state severity $s_j = 1.0$ for CRITICAL/DOWN, 0.5 for WARNING and 0.3 for UNKNOWN/UNREACHABLE, hard states at full class weight and soft states discounted by $\eta_s = 0.4$. (iii) Construct the observability anchors from Nagios delivery as in Section 4.5: a node-specific gap longer than $max(4, \lceil q_{99.5} \rceil)$ windows (realised cut: 4 windows in every month) and at most 96 windows, bracketed by OK alive::ping and ssh::daemon records and measured only while at least 50 % of nodes report, at weight 1.0; a cluster-wide outage for any run of more than 4 windows in which fewer than 50 % of nodes report, at weight 1.0; and the alive-but-unobservable rule (relative in-band coverage below 0.20 for at least two windows, cluster median at least 0.10, liveness probe answering, no physical anchor within Δ_c) at weight 0.9. (iv) Assign hardware, monitoring, coupled, normal, or unlabelled status with $\kappa_H = \kappa_O = 0.5$, coupling tolerance $\Delta_c = 4$ windows, and guard interval $\Delta_g = 8$ windows, leaving windows without any label-plane record unlabelled. (v) Merge consecutive non-normal windows separated by at most four quiet windows into one interval-censored event, and give all windows inside a cluster-wide outage one event identifier. For the modelling set, keep 5 % of the node-windows inside each cluster-wide outage, cap each non-normal class at 60 000 node-windows per month, and draw three normal windows per non-normal window. (vi) The pre-event context region is the $L = 8$ preceding windows (2 h) on the same node that form each sample; the attribution region is the merged event interval itself, within which windows at most Δ_c from both anchor types are eligible for the coupled state; the post-action exclusion is enforced by keeping the drain flag and every other post-action variable out of the feature table and by requiring every normal window to lie at least Δ_g from any drain action. No further region length is tuned. (vii) Mark a window as evidence-degraded when its mean out-of-band or in-band coverage is below 0.60 of the concurrent cluster median (Section 6.7). The artifact (Section 6.10) ships the configuration that fixes every one of these values together with the event table it produces. If the label plane produced no record at all for a window, that window is unlabeled, not normal. To put it in the negative class would be to fold the very evidence the observability head has to explain into the definition of "nothing is wrong".

Loss of reachability on its own is assigned to neither hardware nor monitoring. A node can become unreachable because it failed, because the network failed, because the management path failed, because it was under maintenance, or because of a monitoring fault. Such events are either kept out of the primary label set or, once the evidence from the other planes has been weighed, placed in the coupled/unknown class.

Finally, when collection fails across the whole cluster, that is one incident and not one incident per node. Counting every affected node as its own event would turn a single blackout into hundreds of correlated observations, swell the monitoring class, and shrink the confidence intervals far below what the evidence warrants. All windows inside such an outage therefore carry one event identifier, and on top of that each non-normal class is capped per month, so that a single long incident cannot take over the comparison.

4.7 Leakage Prevention

What operators do can make the classification deceptively easy. The nagiosdrained flag, jobs terminated after an event, and changes to scheduler allocation all happen once an operator or an automated policy has already reacted. None of these variables may appear in an attribution window that precedes the action. They are held back as secondary outcomes, for asking afterwards whether the predicted class matches how the incident was actually handled.

Splitting windows at random is ruled out, since neighbouring windows from one event share both measurements and labels. The primary split is chronological, and every window of a given event stays in a single partition. Tests across nodes and across racks are reported on their own. Scaling parameters, cadence estimates, workload baselines, the choice of metrics, and the imputation models are all fitted on training data and nothing else.

5 PROPOSED METHOD: PARALLAX

5.1 Overview

Figure 2 gives an overview. A temporal encoder specific to its source handles each data plane. Into the physical branch go the workload-conditioned OOB and in-band residuals. The observability branch receives coverage, gaps, cardinality, cadence, and mask sequences. A prior-constrained graph exchanges information across system layers. The model then factorizes the representation into a physical latent state z_t^H and an observability latent state z_t^O. Two weakly supervised heads estimate physical and monitoring risk. Reconstruction and mask-generation heads constrain the latents to explain the observed data-generating process.

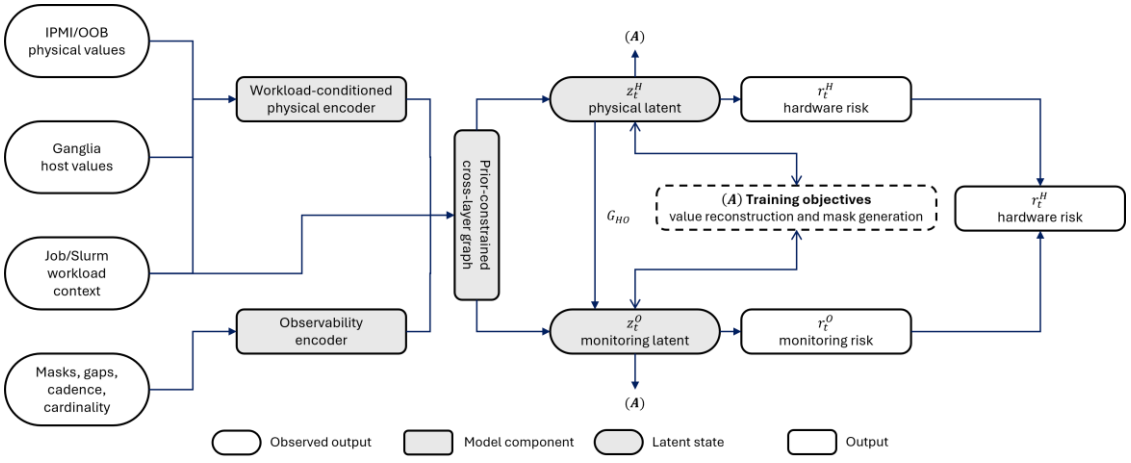


Fig. 2. PARALLAX architecture for hardware-versus-monitoring attribution. The repairs of Section 5.8 add a direct connection from the observability encoder to z^O and to the monitoring head, alongside the fused representation shown here.

Fig. 2 shows the full inference path. Observed physical, workload, and observability inputs are processed by source-specific encoders and fused through the prior-constrained cross-layer graph. The representation is separated into physical and monitoring latent states, z_t^H and z_t^O, with directed coupling through G_{HO}. These states produce the hardware and monitoring risks used for four-state attribution and uncertainty-based abstention. The markers (A) denote the reconstruction and mask-generation objectives applied during training. Section 5.8 adds one path this diagram does not show: the observability encoder also feeds the monitoring head directly, so that head is not separated from its own evidence by the whole fusion stack.

5.2 Workload-Conditioned Physical Residuals

GPU temperature, power, memory use, clocks, and host load respond to workload. We estimate the expected value of each physical metric j from workload and stable context:

$$\hat{x}_{j,t} = g_j(W_t, \phi_n, \psi_{e(t)}),$$

where ϕ_n is a node embedding or node-specific baseline and $\psi_{e(t)}$ describes the configuration era. The workload model g_j is fitted on anchor-clean training windows. The normalized residual is

$$r_{j,t} = \frac{x_{j,t} - \hat{x}_{j,t}}{MAD_j + \epsilon}.$$

The physical encoder receives the observed value, expected value, residual, slope, robust dispersion, and counter increments. This representation separates a high workload-consistent temperature from a high unexplained temperature. It also exposes disagreement between OOB and in-band measurements of related physical quantities.

A single global workload regressor can hide node-specific behavior. We therefore compare three implementations: per-metric gradient-boosted regression, a shared temporal encoder with metric heads, and a hierarchical model with node and era effects. The selected implementation is fixed using validation residual quality on healthy windows; failure labels are excluded from this selection.

5.3 Observability Feature Plane

For each node, family f, and analysis window, let $N_{f,t}^{obs}$ be the number of received samples and $N_{f,t}^{exp}$ the expected number under the estimated cadence. Coverage is

$$\rho_{f,t} = min\left(1, \frac{N_{f,t}^{obs}}{N_{f,t}^{exp} + \epsilon}\right).$$

The estimate N^{exp} must be measured, not assumed. On Marconi100 the BMC polls at roughly 20 s while Ganglia metrics range from 20 s to 5 min and a few are near-static, so a single nominal cadence would make a slow but perfectly healthy metric look permanently degraded. We take N^{exp} for each metric group to be the median count over the windows in which that group reports at all, which is the robust cadence estimate of Section 4 applied per configuration era.
Absolute coverage is also not, by itself, a property of a node. Section 6.1 shows in-band delivery moving between 0.04 and 0.95 from month to month across the whole machine while out-of-band delivery stays between 0.55 and 0.99. A node is therefore compared with what the plane is delivering to the rest of the machine at the same moment,

$$\tilde{\rho}_{f,t} = \frac{\rho_{f,t}}{max\left(median_n \rho_{f,t}^{(n)}, \epsilon\right)},$$

and it is this relative coverage, not its absolute level, that carries the observability signal. We also compute the maximum internal gap $g_{f,t}^{max}$, cadence jitter

$$J_{f,t} = median_k \left|\Delta t_{f,k} - \hat{\delta}_{n,f,e}\right|,$$

active channel cardinality $c_{f,t}$, longest repeated-value run, fraction of stale timestamps, and family-level disappearance indicators. Cross-path asymmetry is

$$a_{f,t} = \rho_{f,t}^{o} - \rho_{f,t}^{i},$$

for physically related OOB and in-band families. A positive asymmetry with stable OOB residuals supports isolated in-band monitoring degradation. Simultaneous OOB and in-band degradation supports a physical, shared-path, or coupled explanation.
The full mask sequence is retained and is not reduced to a single missing ratio. Point gaps, contiguous blocks, device-selective loss, and plugin-wide loss have different operational meanings. The observability encoder therefore consumes channel masks and time-since-last-observation sequences in addition to window summaries.

5.4 Prior-Constrained Cross-Layer Graph

We define a graph $G = (V, E)$ whose nodes represent metric families. The individual metrics stay attached to their family node as attributes. The main families are GPU device, GPU memory, NVLink/interconnect, PCIe/BMC, CPU, host memory, node power and cooling, network, workload/scheduler, and the monitoring pipeline. Which edges may exist is fixed by a prior adjacency mask that admits only couplings the system could plausibly exhibit: workload to GPU, workload to CPU, GPU to power, GPU to temperature, PCIe to device availability, network to monitoring, and physical health to observability. The mask prevents a fully connected graph from learning spurious shortcuts across unrelated service families.

For layer ℓ, graph messages are

$$h_v^{(\ell+1)} = \sigma\left(W_s^{(\ell)} h_v^{(\ell)} + \sum_{u \in N(v)} \alpha_{uv}^{(\ell)} W_m^{(\ell)} h_u^{(\ell)} \right),$$

where α_{uv} is a learned attention value multiplied by the prior mask. Edge stability across seeds and time partitions is reported as part of interpretability analysis. Attention values count as evidence about the model, and no causal-effect reading is placed on them.

5.5 Dual Latent States and Directed Coupling

From the graph representation h_t two latents are formed:

$$z_t^H = e_H(h_t),$$
$$z_t^O = e_O(h_t) + G_{HO} z_t^H.$$

Through the matrix G_{HO} the physical state is allowed to act on the observability state. So that the two latents do not simply encode the same thing, we introduce the residual observability representation

$$\epsilon_t^O = z_t^O - G_{HO} z_t^H$$

and penalize statistical dependence between z^H and ϵ^O with the Hilbert-Schmidt independence criterion (HSIC) [22], after first residualizing both representations against workload and measured context. The directed path from physical state to observability is thereby kept, while redundant latent factors are discouraged.

5.6 Training Objective

The anchor loss, weighted by confidence, reads

$$L_{anchor} = \sum_t \omega_t^H \, BCE(q_t^H, \hat{q}_t^H) + \omega_t^O BCE(q_t^O, \hat{q}_t^O),$$

where ω_t^k reflects anchor specificity, hard/soft state, and temporal distance from the interval-censored event. Class-balanced focal weighting is evaluated for rare coupled events.

The physical reconstruction loss is calculated only on observed values:

$$L_{value} = \sum_{t,j} M_{t,j} \, \ell\left(X_{t,j}, \hat{X}_{t,j}\right).$$

The mask-generation loss explicitly models availability:

$$L_{mask} = -\sum_{t,j} log \, p_\theta(M_{t,j} | z_t^H, z_t^O, C_t).$$

Temporal regularization penalizes implausible latent jumps outside event and configuration boundaries. The full objective is

$$L = L_{anchor} + \lambda_v L_{value} + \lambda_m L_{mask} + \lambda_d HSIC(z^H, \epsilon^O) + \lambda_t L_{temporal} + \lambda_g L_{graph}.$$

Hyperparameters are selected on event-level validation metrics with a penalty for hardware-versus-monitoring confusion. Unlabeled windows contribute to reconstruction, mask, temporal, and graph losses without receiving forced attribution labels.

5.7 Reference-Substitution Risks

Let $\acute{z}^H$ and $\acute{z}^O$ be robust healthy reference states estimated from anchor-clean training windows. We define a monitoring-normalized hardware risk

$$r_t^H = p_\theta(A^H = 1 | z_t^H, z^O \leftarrow \acute{z}^O, W_t)$$

and a physically normalized monitoring risk

$$r_t^O = p_\theta(A^O = 1 | z^H \leftarrow \acute{z}^H, z_t^O, W_t),$$

where ← denotes substitution of a learned latent by its healthy reference, propagated through G_{HO} where the coupling applies.

We deliberately do not write these with a do operator and do not call them counterfactuals. The substitution is performed on a *learned* latent coordinate, not on a variable of the structural model of Section 3; the map from (z^H, z^O) to (H, O) is neither observed nor identified, so the resulting quantity has no interventional interpretation in the sense of Pearl [23] and we claim none. What the substitution does have is a precise operational reading, and that is the whole of our claim for it: r^H asks whether the physical evidence would still be scored as concerning if the observability representation looked healthy, and r^O asks the mirror question. Section 5.8 shows that this reading only holds if the heads are also *trained* under substitution, which the first version of this work did not do.

After calibration, thresholds τ_H and τ_O produce the four states. Continuous risks remain available for node ranking. An uncertainty threshold τ_u creates an abstention state when the ensemble disagreement or predictive entropy is excessive. Abstention is preferable to forced hardware attribution during unseen configuration changes or ambiguous shared-path incidents.

5.8 Repairing the Observability Head

The architecture described so far is the one we evaluated first, and its observability head did not train. At its validation-selected operating point it almost never emitted the monitoring class, and its monitoring ranking (AUPRC-O 0.054) was far below the unfitted asymmetry statistic over its own inputs. We report the diagnosis and the repair rather than the repaired network alone, because the three defects are generic to jointly trained dual-head models and the failure is not visible in an aggregate metric.

R1: no private path to the head. Both heads were fed from the pooled output of the shared family graph. The physical head thereby sat right beside the residuals it needed, whereas the observability head was cut off from its coverage, cadence, and cardinality evidence by the whole of the fusion stack. The repair connects the observability encoder directly to z^O and to the observability head, together with the cross-path coverage columns; the fused representation becomes one input among others instead of the only one. Nothing in the physical branch changes.

R2: train/score regime mismatch. Training used the observed latents, but scoring, as Section 5.7 prescribes, swapped one latent for a healthy reference. At test time the observability head was thus looking at an input distribution it had never encountered, and r^O flattened into a near-constant. The repair computes both regimes within one forward pass and supervises both, so that the score used at test time is the one that was actually trained. This is a condition for reference-substitution scoring to be correct at all, not a tuning decision.

R3: unbalanced head losses. The two anchor classes differ in prevalence by roughly a factor of four and in loss scale by more, so a plain sum of the two binary cross-entropies is minimised faster by serving the physical head alone. Each head's loss is divided by its own running magnitude before the two are added, and the observability branch is given its own parameter group and learning-rate multiple.

Section 7.5 reports the repair ablation, which removes one of R1–R3 at a time and is what attributes any recovery to a specific change.

5.9 The Hybrid Configuration

The configuration the paper reports, PARALLAX-H, is a composition and not a new model, and its two halves are combined by one rule. The physical risk r_t^H is the reference-substituted physical score of Section 5.7, taken from the trained PARALLAX network with its checkpoint frozen; nothing in that head is refitted for the hybrid. The monitoring risk r_t^O is the unfitted cross-path statistic of Section 5.3, the difference $\rho_t^o - \rho_t^i$ between relative out-of-band and in-band coverage, passed through a fixed monotone squashing function onto $[0,1]$; it has no trainable parameter. Each risk is then isotonic-calibrated on the earlier half of the validation partition, and the threshold pair (τ_H, τ_O) is selected jointly on the later half

under the same alert budget as every other method (Section 6.8). The four-state verdict follows the decision rule shared by all methods: hardware when $r^H \geq \tau_H$ and $r^O < \tau_O$, monitoring when $r^O \geq \tau_O$ and $r^H < \tau_H$, coupled when both thresholds are exceeded, and normal otherwise. Nothing is fitted on the test partition at any stage.

The composition is expected to help precisely under degraded observability. When in-band evidence is collapsing, a unified detector sees only absence and tends to read it as a monitoring problem. The asymmetry statistic instead reads the monitoring state directly from the difference between the two collection paths, and the physical head reads the physical state from workload-conditioned out-of-band residuals that survive in-band collapse. Each half therefore supplies the direction in which Section 7 shows it to be reliable, and neither is asked to supply the other.

5.10 Interpretation Output

Each alert includes: hardware risk, monitoring risk, uncertainty, state, top contributing metric families, OOB/in-band agreement, dominant missingness pattern, current workload regime, and the nearest supporting anchor. Explanations use deterministic reason codes. Free-form natural-language generation has no place in it. Example reason codes include OOB_PHYSICAL_DRIFT,INBAND_FAMILY_LOSS,DEVICE_SELECTIVE_DISAPPEARANCE,WORKLOAD_CONSISTENT_THERMAL, BMC_EVENT_ANCHOR, and COUPLED_PHYSICAL_OBSERVABILITY.

6 EXPERIMENTAL METHODOLOGY

6.1 Reproducible Data Slice

Rather than a convenient subset, the whole public release was processed. All 31 monthly archives distributed across the 12 Zenodo records [21] were downloaded, giving 429.8 GB of compressed Parquet covering 9 March 2020 to 28 September 2022. Before use, each archive was checked against the MD5 checksum published in its Zenodo record; the manifest that results, listing source DOI, archive name, byte count, and checksum for every archive, ships with the artifact. The 49.9 TB usually quoted for the corpus is the uncompressed measurement volume, whereas what we consume is the released Parquet encoding of the same data.

The release is laid out as year_month/plugin/metric. Five plugins are present: ipmi_pub (104 metrics), ganglia_pub (33 metrics), nagios_pub, job_table, and slurm_pub (20 cluster aggregates). Nodes are identified by anonymised integers that agree across planes, so IPMI, Ganglia, job-table, and Nagios records line up without any name-resolution step. We checked that alignment directly: the IPMI plane shows 980 nodes, the Ganglia plane 979, and the compute host group of the Nagios plane 980.

Of the 573 published metrics, 78 carry the evidence the formulation calls for and were kept, arranged into 39 metric groups and 9 observability families as Table I lists. Per-core CPU temperatures and per-DIMM sensors were not discarded but aggregated to group level, so what the reduction removes is redundancy, not evidence. Aggregating the raw samples to the 15 min Nagios grid gives 83 712 480 node-windows over 28 months and 988 nodes, with 449 features per window.

Several properties of the release constrain the analysis and are reported rather than smoothed over.

First, the release does not begin fully instrumented. March and April 2020 contain the IPMI plane and nothing else: no Ganglia, no job table, and no Nagios. Neither month can be labelled, so both are dropped from the modelling set, which leaves 29 labelled months. Both lie within the training period, and the test partition is untouched.

Second, monitoring coverage does not stay constant. The inventory of compute services grows from nine in early 2020 to sixteen by January 2021, which means the hardware anchors the formulation leans on – bmc::events and sys::rvitals above all – are only present from 2021 onward: GPU memory retirement and replacement checks enter the configuration in August 2020, InfiniBand status in October, physical-memory total in November, and the BMC anchors in January 2021, so the two highest-specificity physical anchors are unavailable for the first three labellable months and the BMC anchors for the first eight. Fig. 3 shows the month-by-family coverage matrix, and the configuration eras used for the held-out-era experiment are read from it. March 2021 is materially incomplete, containing roughly one tenth of the Nagios volume of neighbouring months.

Third, and most consequential for the method, the two observation paths do not merely differ in reliability: they differ in *stationarity*. Out-of-band delivery stays between 0.55 and 0.99 of expected samples per window in every modelled month. In-band delivery moves between 0.04 and 0.95, and it moves *uniformly across the machine* – in June 2020 essentially every node receives about a third of its expected Ganglia samples, and by January 2022 essentially every node receives 94%. This is a property of the collection and publication path, not of individual nodes, and it is why Section 4.5 references each node against the concurrent cluster median instead of an absolute coverage level. An absolute threshold calibrated on any one month would classify most of a neighbouring month as a monitoring failure.

Fourth, the Slurm TRES encoding changes mid-corpus, and the change is easy to miss because it fails silently. Until October 2020 the allocated GPU count appears in tres_alloc_str as gres/gpu=N for the whole job. From November 2020 onward

that field is empty and only tres_per_node carries the count, as gpu:N or gres:gpu:N. Reading tres_alloc_str alone – the natural choice, and the one the earlier months reward – yields exactly zero allocated GPUs for 26 of the 29 usable months, including the entire validation and test period, while leaving CPU and job-count features intact and therefore raising no obvious alarm. That would disable GPU workload conditioning across almost the whole study. We parse both encodings and verify per month that the recovered allocation is non-degenerate.

6.2 Data Splits

The proposed primary split is chronological:

- training: 9 March 2020 to 30 June 2021;
- validation: 1 July 2021 to 31 December 2021;
- testing: 1 January 2022 to 28 September 2022.

This split must be adjusted only when an essential class is absent from a partition, and any adjustment must be defined before final model comparison. A second experiment holds out complete nodes or racks from training to test transfer. A third experiment holds out one configuration era. All windows from one event remain in the same partition.

6.3 Baselines

Table III defines the minimum baseline suite. All baselines use the same training, validation, test windows, event definitions, and alert budget. Imputation baselines receive an explicit mask variant so that the benefit of missingness preservation is evaluated fairly.

Two provisions make the comparison with the deep baselines fair, and both were added after the first version of this study [7], in which the baselines ran at their default configurations. First, every learned model – the proposed network and each deep baseline alike – draws 4 configurations from the same search space over learning rate, hidden width, weight decay, dropout, and positive-class weighting, trained for a reduced number of epochs on a subsample and selected on the mean of validation AUPRC-H and AUPRC-O, so that no configuration can win by serving one head only; the winning configuration is then trained in full. Across 7 learned methods this is 28 search runs, and the selected configurations are part of the artifact. Ablations inherit the configuration selected for the full model, so removing a component is not confounded with changing a hyper-parameter. Second, decision thresholds are searched over the empirical quantiles of each method's *own* score in addition to a fixed grid on $[0.05,0.95]$. A fixed grid silently denies an operating point to any method whose risk never approaches 0.5, which turns a poorly scaled score into an apparently dead class; this provision helps the weaker baselines at least as much as it helps the proposed model.

TABLE III. REQUIRED BASELINE AND ABLATION SUITE.

Method	Input	Purpose	Required output
Rule hierarchy	Coverage, OOB residuals, anchors	Transparent operational baseline based on fixed asymmetry and threshold rules	Four-state class and reason code
Coverage asymmetry	$\rho^o - \rho^i$ only	The unfitted cross-path statistic, with no parameter estimated on any partition; the reference every learned observability head must beat	Monitoring risk; physical risk from the residual rule
Isolation Forest-V	Physical values/residuals	Classical value-only anomaly detector [17]	Unified score mapped to classes by anchor-calibrated rules
Isolation Forest-VM	Values plus masks/gaps	Quantifies the incremental value of structural telemetry features	Unified score and mapped class
GRU-D	Values, masks, time gaps	Missingness-aware recurrent baseline [14]	Physical and monitoring heads, multi-task
SAITS + classifier	Imputed sequence plus mask	Strong attention-based imputation baseline [16]	Class probabilities, multi-task
GDN	Multivariate families	Learned inter-variable graph baseline [18]	Anomaly score with root-family ranking
TranAD	Multivariate sequence	Transformer reconstruction baseline [19]	Anomaly score, mapped to a class
Multi-task temporal model	All planes	Same encoders and two heads without structural factorization, graph prior, or reference-substitution scoring	Two risks; four-state class
PARALLAX v1	All planes	The architecture of the first version of this study, retained so that the observability-head failure is reproduced rather than described	Two risks; four-state class

Method	Input	Purpose	Required output
PARALLAX	All planes	Proposed network with repairs R1–R3 of Section 5.8	Calibrated risks with class, uncertainty, and reason codes
PARALLAX-H (hybrid)	All planes	**Main configuration.** Physical risk from PARALLAX, monitoring risk from the unfitted asymmetry statistic	Calibrated risks with class and reason codes

6.4 Ablation Studies

Two ablation families are run. The *structural* ablations remove one evidence source or mechanism at a time: the OOB plane; workload conditioning; explicit masks and timing structure; the prior graph mask; directed physical-to-observability coupling; HSIC separation; reference-substitution normalization; and the observability feature plane. The *repair* ablation removes one of R1–R3 of Section 5.8 at a time, and is what attributes any recovery of the observability head to a specific change rather than to retraining. Both families report degraded-subset macro-F1 first, then full-partition macro-F1, hardware false-escalation rate, monitoring false-dismissal rate, and coupled recall. All ablations inherit the hyper-parameters selected for the full model.

6.5 Robustness Stress Tests

Monitoring failures are simulated only inside windows that are healthy or confidently physical and whose original data are present. Five kinds of corruption are injected at controlled severity:

- removal of individual points at random;
- gaps made of contiguous blocks;
- disappearance of an entire metric family;
- disappearance of the in-band plugin with OOB still delivering;
- loss of one GPU device family alone, with node-level metrics intact.

A sixth test delays timestamps and jitters the cadence without touching any value. Sensitivity to physical perturbation is assessed separately, through bounded shifts in workload-conditioned temperature, power, error-counter increments, and disagreement between planes. What synthetic perturbations test is whether decisions are consistent; the primary evidence still has to come from real failure events.

6.6 Evaluation Metrics

Evaluation is done at the level of the event; the overlapping windows inside one event are treated as dependent observations of it. Within an event interval, window scores are combined by taking the maximum calibrated risk or by applying a persistence rule fixed in advance. Concretely, unless a caption says otherwise, the statistical unit of every table in Section 7 is the *evaluation unit* of the test partition: each interval-censored non-normal event is one unit, scored by the maximum calibrated risk over its windows, and each normal window is one singleton unit. Macro-F1, AUPRC-H, AUPRC-O, coupled recall, HFER, MFDR, and expected calibration error are all computed over these units. The alert rate alone is a window-level quantity, the number of alerted windows divided by the number of node-days in the partition. The robustness sweep and the operational examples of Table XII are reported per node-window, and the label validation of Table VI per labelled event. The quantities reported are:

- macro-F1 and balanced accuracy across the four states;
- the area under the precision-recall curve (AUPRC), per class;
- recall at event level, and false alerts per node-day;
- Brier score and expected calibration error of r^H and of r^O;
- for node ranking, recall of physical risk in the top K and mean average precision;
- recall on coupled events;
- abstention coverage together with selective risk;
- the latency of attribution, measured from the interval-censored event boundary.

Two operational error measures carry particular weight. The first, the hardware false-escalation rate, is

$$HFER = \frac{N(\hat{S} = \text{hardware} | S = \text{monitoring})}{N(S = \text{monitoring})},$$

and the second, the monitoring false-dismissal rate, is

$$MFDR = \frac{N(\hat{S} = \text{monitoring} | S = \text{hardware})}{N(S = \text{hardware})}.$$

HFER captures physical escalations that a monitoring incident provoked unnecessarily; MFDR captures physical incidents that were waved away as data-quality problems. Both are computed over evaluation units, with $N(\cdot)$ counting units, and given with event-level bootstrap confidence intervals.

6.7 The Evidence-Degraded Subset: the Primary Comparison

An attribution benchmark can be gamed in one particular way: a method can look successful merely by noticing that *something* is absent. Since most normal windows come with complete evidence, a detector that did nothing but report missingness would already separate the classes, without attributing anything. We therefore designate as the *primary* comparison the subset of events in which the evidence is actually degraded, defined as mean out-of-band or in-band coverage below 0.60 of the concurrent cluster median in the target window. Within this subset the hardware, monitoring, and coupled classes all exhibit missing data, and the only way to score is to decide *why* it is missing.

The restriction removes the shortcut in a specific way. The subset is selected by coverage alone, before any class label is consulted, so the normal units inside it have degraded evidence just as the hardware, monitoring, and coupled units do; the amount of missing data therefore no longer separates normal from non-normal, and a score that rises with missingness gains nothing. What remains available is the *pattern* of what was delivered – which path degraded, whether the out-of-band residuals that did arrive are workload-consistent, how the gaps are structured – and that pattern is what distinguishes a physical cause from a collection failure. A method that merely counts absence is reduced to the base rate on this subset, which is the behaviour Table VII shows for the rule hierarchy and the raw statistic on AUPRC-H. This is a change of emphasis from the first version of this study, in which the full-partition aggregate was primary and the degraded subset a supplementary control, and it is a change we argue for rather than one made for convenience. In this corpus a large part of the observability anchor is *defined* by in-band collapse, and the feature that wins the full-partition aggregate *measures* in-band collapse; the aggregate therefore rewards a detector for rediscovering the label construction. The degraded subset does not depend on the observability anchor being right, only on the hardware anchors, which come from an independently polled service plane. It is also the regime an operator actually faces when the decision is difficult. Full-partition results are reported throughout alongside it, and where the two views disagree we say so and give both.

6.8 Calibration and Threshold Protocol

Every method is scored through one protocol. The validation partition is split chronologically in two. Its earlier half fits an isotonic calibrator for each risk head; its later half, after calibration, selects the decision threshold pair (τ_H, τ_O) under the fixed alert budget. No method is calibrated and thresholded on the same windows, thresholds live on the probability scale that the test scores are reported on, and nothing is fitted on test. The threshold grid is the union of a fixed grid and the empirical quantiles of the method's own score, as described above.

6.9 Statistical Analysis

Confidence intervals are generated by hierarchical bootstrap sampling at the event and node levels. Pairwise model differences use a paired permutation test over events. The number of random seeds is fixed before results are examined, and all seeds are reported. Multiple comparisons in the ablation table are controlled using the Holm procedure. Thresholds are selected on validation data under a fixed false-alert budget and remain unchanged on the test set.

6.10 Implementation Details

Pipeline. Extraction streams each monthly archive once and aggregates every selected metric to the 15 min grid inside that pass, so the release is read but never duplicated on disk. Aggregation is a sorted-segment kernel that returns, per node-window and per metric, the sample count, mean, standard deviation, minimum, maximum, least-squares slope from raw moments, the internal maximum inter-sample gap, and the fraction of consecutive identical values. The gap and flatness statistics are what separate a uniformly thinned stream from a contiguous outage and a live stream from a frozen exporter, and they are computed rather than approximated. Intermediate tables are stored as Zstandard-compressed Parquet.

Feature layout. Each sample is a target window with the seven preceding windows on the same node, giving a 8×449 tensor with a parallel binary availability mask. The canonical column set is identical in every month; a metric group that an era does not contain is emitted as all-NaN with a plane-level presence flag, which keeps era-driven absence distinguishable from within-era loss.

Workload baselines. The expected value of each workload-driven physical group is estimated by histogram gradient-boosted regression on allocated jobs, allocated CPUs and GPUs, job age, job starts and terminations, time of day, node identity as a categorical variable, and configuration era. The regressors are fitted only on training-period windows that carry no anchor of any kind, and residuals are normalised by the training median absolute deviation.
Models. All neural models share one gated recurrent encoder architecture and the same optimiser, and all draw their width, learning rate, weight decay, dropout, and positive-class weighting from the shared search of Section 6, so differences reflect architecture and objective rather than budget or tuning effort. PARALLAX has a workload-conditioned physical branch and an observability branch; their concatenation is projected onto ten metric-family nodes, over which a single graph-attention layer, masked by the prior adjacency of Section 5, is applied. In addition, the observability branch feeds its own head directly (repair R1). Because attention is multiplied by the prior mask, no forbidden edge can arise. The optimiser is AdamW on a one-cycle schedule, with batch size 512, gradient norms clipped at 5, no more than 12 epochs, and early stopping on validation area under the precision-recall curve with a patience of 3. The observability parameters form a group of their own whose learning-rate multiple is part of the search. Each anchor loss is divided by its own running magnitude before the two are summed, which is repair R3. Loss coefficients are $\lambda_v = 0.1$, $\lambda_m = 0.3$, and $\lambda_d = 0.5$; unlabelled windows contribute to the reconstruction, mask, and graph terms only. Healthy reference latents $\acute{z}^H$ and $\acute{z}^O$ are the coordinatewise medians of the latents on anchor-clean training windows, refreshed during training so that the reference-substituted scores of Section 5.7 are supervised on the references they are later evaluated with, which is repair R2.
Baselines. Isolation Forest uses 200 trees fitted on anchor-clean training windows; its single score is split into the two risks by the out-of-band/in-band coverage asymmetry, which is the only structural information a unified detector can contribute to attribution, and this favours the baseline rather than the proposed method. GRU-D implements input decay towards the empirical mean together with hidden-state decay. The SAITS baseline uses two diagonally masked self-attention blocks with a joint imputation and classification objective. GDN learns an inter-variable graph from node embeddings and scores forecast deviation. TranAD uses a two-stage transformer reconstruction. The rule hierarchy is a fixed threshold cascade over coverage asymmetry and workload-conditioned residuals with no learned parameters.
Thresholds and reporting. Each risk head is calibrated by isotonic regression on the earlier half of validation; decision thresholds (τ_H, τ_O) are then selected on the later half by grid search under a fixed alert budget of 0.05 alerts per node-day, over the union of a fixed grid and the empirical quantiles of that method's own score, and are frozen for the test partition. Seeds are fixed before results are examined; PARALLAX, its v1 predecessor, and the closest learned baseline are run with three seeds and the remaining methods with one, and all seeds are reported.
Environment. Experiments ran on a Slurm cluster using AMD EPYC compute nodes with 128 cores and 512 GB of memory per node for extraction and NVIDIA A100 GPUs for training, under Python 3.11 with NumPy 2.4, pandas 2.3, PyArrow 25, scikit-learn 1.8, and PyTorch 2.6 with CUDA 12.4. Extraction of the full release is the dominant cost; model training and the complete ablation and robustness suite are comparatively small.
Measured cost. The full release is 429.8 GB compressed across 31 monthly archives from 12 Zenodo records. Windowed feature extraction is the dominant cost and is embarrassingly parallel over months: 29 monthly jobs at eight cores each consumed 119 core-hours in total, roughly half an hour of wall-clock per month, producing 27 GB of features and a further 27 GB of model tensors. Everything downstream is small by comparison. The complete experimental suite is 38 scored runs – the proposed model and its predecessor at three seeds each, the baselines, the composed configurations, and both ablation families – preceded by 28 hyper-parameter-search runs over 7 learned methods, and fits in single-digit GPU-hours on one A100. The sensor-disappearance sweep of six corruption modes at six severities takes minutes. The one stage that is not cheap is statistical testing: paired permutation over 441 837 evaluation units, at one thousand resamples per comparison and now run twice per comparison for the full and degraded views, dominates the end-to-end time and should be budgeted for accordingly.
Artifact availability. The implementation, experiment configuration, preprocessing and evaluation scripts, and figure-generation code are available in the public repository at https://gitlab-ce.gwdg.de/bidollahkhan/parallax-gpu-degradation.

7 RESULTS

All numbers below are reported at event level on the chronologically held-out test partition, with window-level diagnostics confined to the supplementary material. Each number in the text is written out by the analysis scripts straight from the result files, which keeps the prose from drifting away from the runs behind it.
The order of this section follows what the evidence can bear, not the order in which the method was proposed. Everything on the monitoring side hinges on whether the constructed monitoring labels hold up against evidence that played no part in building them, so Section 7.2 asks that first. The primary comparison, on the evidence-degraded subset, comes in

Section 7.3. Section 7.4 then turns to the full partition, where the picture points the other way, and explains what that metric is in fact measuring. The architectural question left open by the first version of this study is closed in Section 7.5.

7.1 Dataset Audit and Label Distribution

The assembled corpus spans 29 monthly partitions from May 2020 to September 2022, giving 83 712 480 node-windows on the full in-service grid at a fifteen-minute resolution. Anchor construction yields 119 169 interval-censored non-normal events. The four-state window distribution is strongly imbalanced: 84.65 % normal, 2.07 % hardware, 8.82 % monitoring, and 0.06 % coupled, with a further 4.40 % left unlabelled because no anchor plane was delivering. Coupled windows are two orders of magnitude rarer than either single-cause class, which sets a hard ceiling on what can be established about the coupled state and is the reason we report coupled recall separately rather than folding it into an average.

The first version of this study quoted node, month, window, and event figures that each described a different population and so appeared to contradict one another. Table IV reconciles them in one place, and every count in this paper is drawn from it. Three distinctions do the work. *Nodes*: 980 identifiers appear on the IPMI plane, 979 on Ganglia and 980 in the Nagios compute host group in any given month, but the node set changes over the corpus, so the union across all planes and months is 988 identifiers, of which 982 are present in the leanest month; wherever the text says “nodes” without qualification it means the union. *Months*: 31 monthly partitions are published and 29 of them carry both telemetry planes and can therefore be labelled, since March and April 2020 carry the IPMI plane only, and 28 enter the modelling set after March 2021 is excluded as materially incomplete. *Events versus evaluation units*: the corpus contains 119 169 interval-censored non-normal events, whereas the event-level evaluation additionally treats each normal window as its own singleton unit, giving 441 837 evaluation units on the test partition of which 36 338 are non-normal events. These are different quantities and are never given the same name again.

TABLE IV. COUNT RECONCILIATION. EVERY POPULATION FIGURE QUOTED IN THE PAPER IS GENERATED FROM THIS TABLE, AND THE THREE DISTINCTIONS THAT CAUSED THE EARLIER INCONSISTENCY – PER-PLANE VERSUS UNION NODE COUNTS, PUBLISHED VERSUS ASSEMBLED VERSUS MODELLED MONTHS, AND EVENTS VERSUS EVALUATION UNITS – ARE SEPARATED HERE.

Quantity	Value
Monthly partitions published	31
Monthly partitions assembled (both planes present)	29
Monthly partitions in the modelling set	28
Compute nodes, union over all planes and months	988
Compute nodes present in the leanest month	982
Compute nodes in the Nagios compute host group	980
Node-windows on the in-service grid	83 712 480
Node-windows in the modelling set	6 727 451
Interval-censored non-normal events	119 169
Test-partition windows	540 000
Test-partition evaluation units	441 837
Test-partition non-normal events	36 338

Three properties of the release govern the experimental design and are visible in Fig. 3. First, in-band delivery is non-stationary at cluster scale: the concurrent cluster-median in-band coverage ranges from 0.00 to 1.00 across months, and in 7 of 29 months the in-band plane is effectively not delivering at all. Absolute coverage thresholds are therefore meaningless, and every anchor and feature in this study is defined relative to the concurrent cluster median rather than to a fixed constant. Second, the out-of-band plane is markedly more stable than the in-band plane, which is the asymmetry the formulation depends upon. Third, March 2021 is materially incomplete and is excluded from the modelling set, leaving 28 months and 6 727 451 windows after event-aware sampling; Table V gives the split-level composition.

The observability anchor is dominated by two mechanisms. Cluster-wide loss of the Nagios plane accounts for the long, wide incidents visible as vertical bands in Fig. 4; the alive-but-unobservable component, in which a node’s in-band plane collapses while the active liveness probe still answers, accounts for 479 646 further windows and supplies all of the node-local monitoring events, since the node-specific Nagios-silence component qualifies in no month (Section 7.2). The per-node monitoring-health check that would have provided a direct label does not exist for compute nodes in this release, a limitation examined in Section 7.2 and again in Section 8.

Workload conditioning is effective but partial. Across 13 physical targets the workload model attains held-out R^2 between 0.21 and 0.71, highest for I/O power and ambient temperature and lowest for fan speed. The residuals therefore remove a substantial but not complete share of the thermal and power variation that workload alone explains.

Finally, 30.8 % of node-windows meet the evidence-degraded criterion of Section 6.7. This subset is large enough to support the separate analysis that Section 7.3 reports, and, as it turns out, it is where the methods differ most.

TABLE V. FOUR-STATE WINDOW DISTRIBUTION OF THE MODELLING SET AFTER EVENT-AWARE SAMPLING. NORMAL WINDOWS ARE SUBSAMPLED AT THREE PER NON-NORMAL WINDOW; EVERY NON-NORMAL WINDOW IS RETAINED. UNIT: NODE-WINDOWS, ALL THREE PARTITIONS.

Split	Normal	Hardware	Monitoring	Coupled
Train	2 265 223	377 067	369 983	7 464
Validation	1 158 521	304 848	81 580	208
Test	1 621 565	383 734	147 282	9 976

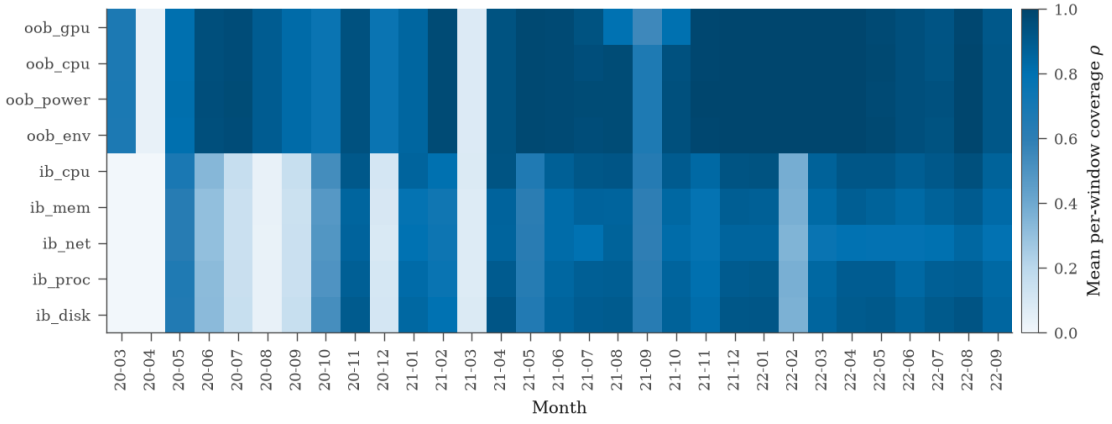


Fig. 3. Mean per-window coverage ρ by month and observability family

Fig. 3 exposes the configuration changes that motivate the era-aware analysis. The in-band plane is absent in April 2020, the compute service inventory grows through 2020, and March 2021 is materially incomplete. The resulting vertical coverage transitions define the configuration-era boundaries used in the held-out-era experiment.

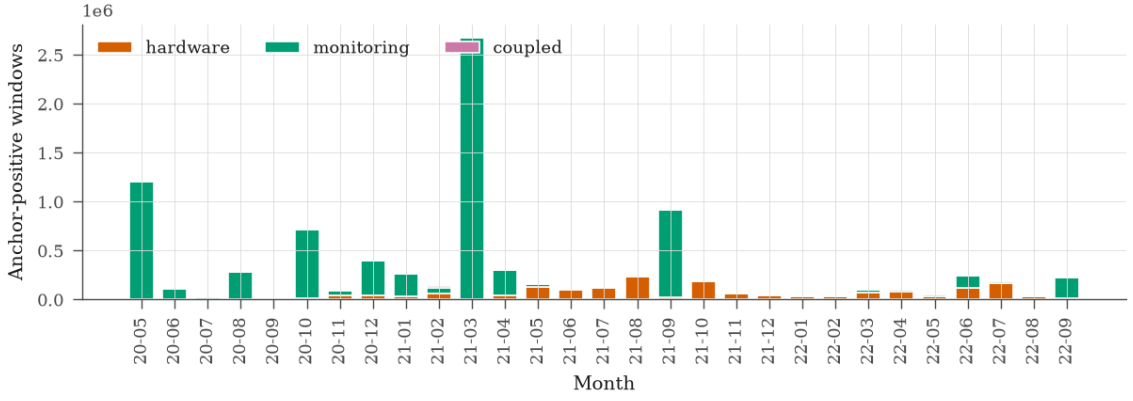


Fig. 4. Anchor-positive windows per month by attribution state

Fig. 4 shows that hardware-anchor coverage follows the growth of the Nagios compute-service inventory, whereas observability anchors are concentrated in a small number of long cluster-wide collection outages. This concentration is important when interpreting the observability class and motivates treating a cluster-wide outage as one incident rather than as independent node events.

7.2 Validating the Constructed Monitoring Labels

Because the observability anchor is constructed from the delivery structure of the Nagios plane rather than taken from a labelled service, it has to be validated before it is used. We first report two checks internal to the label plane, both of which fail and for instructive reasons, and then three external witnesses that do not.

The first internal check is agreement with an operator-visible signal: cluster-wide silences of the Nagios plane are compared against non-OK monitoring::health states on the management hosts, which are produced by a different check and were never used to define the silence. The second is a specificity check: node-specific silences are compared against the hardware anchors, since a construction that merely rediscovered node failures would show strong overlap with them.

The first check as originally specified is void. Asking whether a window carrying a cluster-wide Nagios silence also carries a non-OK monitoring::health state cannot return a positive answer, because a silence means the Nagios plane delivered nothing for that window and monitoring::health is itself a Nagios service. The measured overlap is exactly zero over all 18 months, and that zero is forced by construction rather than informative. We record the error rather than removing the check, because the same trap is easy to fall into whenever a held-out plane is validated against a service carried on that plane.

The corrected form of the question is temporal. An outage an operator would recognise should leave a trace when the plane returns, so we ask whether a non-OK monitoring::health appears within 2 hours on either side of each silence run, and compare against a null in which the same runs are slid to random offsets within the same month. Of 36 cluster-wide silence runs, median duration 8.2 hours, 16 are corroborated, a rate of 44.4 % against a null rate of 40.5 % and a lift of 1.10. This is no evidence in either direction. The reason is visible in the null itself: the management-host health service is non-OK in a median of 38 % of windows and in 4 months the random baseline is above 0.9, so almost any interval contains a non-OK state and the signal cannot discriminate.

The second internal check is empty for a different reason. The node-specific silence component contributed no anchor windows at all in any month, so there is nothing whose overlap with hardware anchors could be measured. In practice the observability anchor is the union of cluster-wide Nagios silence and the alive-but-unobservable rule, and the specificity of the latter cannot be tested against the hardware anchors either, since it is defined to exclude them.

The management-host route is therefore closed. It does not follow that the anchor is untestable, only that it cannot be tested against another Nagios service, and the first version of this study stopped one step too early. What a constructed label can be tested against is evidence from planes that did not build it, and three such witnesses exist in this release.

W1, workload continuity. The Slurm job table is a third collection system, independent of both the Nagios poller and the in-band agent. A node whose collector has failed keeps running its jobs; a node whose hardware is failing tends not to. We count job terminations recorded inside each labelled event.

W2, out-of-band physical stability. Assumption A1 says an isolated in-band collection failure leaves the BMC values conditionally unchanged. We take the maximum workload-conditioned out-of-band residual inside each event. The out-of-band plane never enters the anchor construction.

W3, operator action. The nagiosdrained flag is a post-hoc operator decision, excluded from the feature table by the leakage rules of Section 4. If the labels track what operators believed at the time, hardware events should be followed by a drain and monitoring events should not.

Each witness is measured against a matched null in which pseudo-events of the same length are placed at random offsets on the same node in the same month, so a witness that is merely common cannot pass. Table VI gives the result over 154 413 events. Two of the three witnesses separate the classes in the predicted direction and one does not.

The out-of-band residual is the clearest. Hardware events carry a maximum residual of 10.55 against a null of 5.89, while monitoring events carry 4.15, which is *below* their own null: windows the anchor calls monitoring failures look physically more ordinary than randomly chosen windows on the same node. The difference between the two classes is significant at $p = 0.001$ under a permutation test. Operator action tells the same story: 0.06 of hardware events are followed by a drain, against 0.02 of monitoring events, and since the monitoring figure sits on its null, the operators evidently did not take these incidents for hardware faults either. Job terminations agree, with monitoring events coming in below their null.

Only one witness fails to discriminate: whether an allocation that was present at onset outlives the event, which happens for 0.56 of monitoring events and 0.52 of hardware events. With hindsight this is unsurprising, because the hardware anchors that dominate here, GPU memory retirement and BMC event records, are logged without the running job being killed.

Two limits are left standing, and nothing above removes either. The witnesses validate the *labels*, not the difficulty of the task, and W1 and W2 draw on planes that are also in the feature table, so a model can in principle use the same evidence; this is why the degraded subset, not the headline metric, carries the claims of Section 7.3. And validation on a second machine, with a genuine per-node collector-health signal, remains the check we cannot run here and the one we would run first with access to another installation.

TABLE VI. INDEPENDENT VALIDATION OF THE CONSTRUCTED LABELS. EACH CELL IS THE WITNESS MEAN OVER REAL EVENTS, WITH THE MATCHED-NULL MEAN IN PARENTHESES. W1 COUNTS JOB TERMINATIONS INSIDE THE EVENT, W2 IS THE MAXIMUM WORKLOAD-CONDITIONED OUT-OF-BAND RESIDUAL, W3 IS THE FRACTION FOLLOWED BY AN OPERATOR DRAIN WITHIN 24 H, W4 IS THE FRACTION OF WINDOWS IN WHICH THE ACTIVE LIVENESS PROBE ANSWERED. UNIT: LABELLED EVENTS AS CONSTRUCTED IN SECTION 4, ONE ROW PER CLASS WITH n THE NUMBER OF EVENTS; THE MATCHED NULL PLACES PSEUDO-EVENTS OF THE SAME LENGTH AT RANDOM OFFSETS ON THE SAME NODE IN THE SAME MONTH. EVENTS ARE ENUMERATED PER NODE, SO A CLUSTER-WIDE OUTAGE CONTRIBUTES ONE EVENT PER AFFECTED NODE AND n EXCEEDS THE EVENT COUNT OF TABLE IV.

Class	n	W1 alloc.	W2 OOB resid.	W3 drain	W4 alive
Hardware	61 102	0.52 (0.60)	10.55 (6.22)	0.06 (0.03)	0.91 (0.97)
Monitoring	83 538	0.56 (0.55)	4.15 (5.89)	0.02 (0.02)	0.63 (0.95)
Coupled	9 773	0.57 (0.54)	5.14 (5.22)	0.04 (0.02)	0.56 (0.98)

7.3 Primary Result: Attribution when Evidence Is Already Degraded

The primary comparison is the evidence-degraded subset of Section 6.7: 72 760 of the 441 837 test evaluation units, covering 30.8 % of node-windows, in which every window already has degraded evidence and no method can separate the classes by noticing that data is absent. Its base rates differ from the full partition and the AUPRC columns must be read against them: hardware-or-coupled events are 4.52 % of the subset against 3.44 % overall, and monitoring-or-coupled events 29.21 % against 5.20 %. The subset is therefore not simply a harder version of the same task; it is the regime in which the question the paper asks actually has to be answered.

Table VII gives the result. The hybrid configuration – physical risk from PARALLAX, monitoring risk from the unfitted cross-path statistic – attains macro-F1 0.455 and AUPRC-H 0.603 against 0.349 and 0.184 for Isolation Forest-VM, the strongest classical detector, and 0.211 and 0.045 for the rule hierarchy. Its physical ranking is more than an order of magnitude above the 4.52 % base rate, which is what a practitioner ranking nodes for inspection is buying.

What these two numbers mean operationally should be stated before they are compared further. Macro-F1 of 0.455 is the mean over the four states of the per-state F_1 at the frozen operating point, computed over degraded-subset evaluation units: it says that the four-way verdict is right materially more often than for the detectors below it, averaged so that the rare classes count as much as the common one, and it is held down by a coupled-class F_1 that is near zero for every method. AUPRC-H of 0.603 is a ranking quantity: against a hardware-or-coupled base rate of 4.52 % it says that a worklist ordered by physical risk concentrates genuine hardware events near its top roughly thirteen times more densely than random ordering would. It does not say that an alarm raised at a fixed threshold is right six times in ten; Fig. 7 shows precision falling towards prevalence beyond moderate recall. Both figures are measured against constructed weak labels, on one machine, on the 30.8 % of node-windows that meet the degradation criterion, and they should be read as the performance of a triage ranking under those conditions rather than as unrestricted real-world diagnostic accuracy. It is in the operational error rates that the difference counts for most. False monitoring dismissal – a genuine hardware event filed as a data-quality problem, the error that delays physical intervention – reaches 0.897 for Isolation Forest-V, 0.811 for Isolation Forest-VM and 0.435 for the rule hierarchy: once evidence is degraded everywhere, the missingness-aware detectors dismiss most genuine hardware events as monitoring problems, which is the specific failure the formulation predicts for them. The hybrid records 0.203 at a false hardware escalation of 0.000, the lowest HFER in the table.

That MFDR figure deserves its own sentence, because it is not the lowest in the table and we do not present it as such. PARALLAX alone records 0.007, and several deep baselines are lower still, for a reason Section 7.5 makes plain: a method that rarely emits the monitoring class rarely misuses it, so a low MFDR earns credit only when it is paid for with monitoring detections. The hybrid emits that class 50 486 times on the test partition, and its 0.203 is the price of doing so; among the five methods that emit it at a comparable rate – the two Isolation Forest variants, the rule hierarchy, the asymmetry detector and the hybrid – it is the lowest, ahead of 0.264 for the next best. An installation that cared only about never dismissing a hardware event would deploy the physical branch alone and accept that it names an order of magnitude fewer monitoring incidents.

This is the substantive claim of the paper, and it is narrow by construction: it holds on 30.8 % of node-windows, it says nothing about the coupled state, and it rests on labels this release cannot verify directly (Section 7.2).

Both halves are computed on the validation partition, the isotonic calibrators are fitted on its earlier half and the threshold pair is selected on its later half under the same 0.05 alerts per node-day budget applied to every other method, and only then is the frozen pair applied to test; the physical half reuses the seed-0 PARALLAX checkpoint unchanged, so no component is refitted and nothing is tuned on test. On the evidence-degraded subset the configuration reaches macro-F1 0.455, AUPRC-H 0.603, AUPRC-O 0.407, HFER 0.000 and MFDR 0.203; on the full partition it reaches macro-F1 0.442, AUPRC-H 0.192, AUPRC-O 0.378, HFER 0.000 and MFDR 0.046 at 0.011 alerts per node-day ($\tau_H = 0.522, \tau_O = 0.455$). This is the highest degraded-subset macro-F1 of any configuration in the study, +0.084 against the strongest single model, TranAD, and +0.105 against the strongest classical detector, Isolation Forest-VM; a paired permutation test over degraded-

subset events, each method held at its own validated operating point, gives Holm-adjusted $p = 0.037$ (significant at the 0.05 level).

For every method in the table, coupled recall is zero or close to it. Coupled windows make up 0.06 % of the corpus and the coupled events cluster in a handful of incidents, so the experiment as built is in no position to establish H4. We leave the coupled state unresolved rather than call it refuted; Section 8 says what a corpus would have to contain for the question to be settled.

TABLE VII. **PRIMARY COMPARISON.** TEST-SET RESULTS RESTRICTED TO THE EVIDENCE-DEGRADED SUBSET, ON WHICH EVERY WINDOW ALREADY HAS DEGRADED EVIDENCE AND NO METHOD CAN SEPARATE THE CLASSES BY DETECTING ABSENCE ALONE. THE SUBSET COVERS 30.8 % OF NODE-WINDOWS. COMPARE WITH TABLE VIII: THE MACRO-F1 ORDERING CLOSES AND THE FALSE-MONITORING-DISMISSAL ORDERING REVERSES. ROWS ARE ORDERED BY INCREASING MODELLING MACHINERY – RULE-BASED AND UNFITTED, THEN CLASSICAL UNSUPERVISED, THEN DEEP BASELINES, THEN THIS WORK – AND NOT BY SCORE; THE SAME ORDER IS USED IN EVERY RESULTS TABLE. VALUES ARE GIVEN TO TWO DECIMALS, WITH 95 % HIERARCHICAL-BOOTSTRAP INTERVALS IN BRACKETS. UNIT: TEST-PARTITION EVALUATION UNITS (ONE PER NON-NORMAL EVENT PLUS ONE PER NORMAL WINDOW) RESTRICTED TO THE EVIDENCE-DEGRADED SUBSET, 72 760 UNITS IN ALL; HFER AND MFDR ARE FRACTIONS OF MONITORING AND HARDWARE UNITS RESPECTIVELY; THE HYBRID REUSES THE FROZEN SEED-0 PARALLAX PHYSICAL HEAD, AND NOTHING IS FITTED ON TEST.

Method	Macro- F1	AUPRC- H	AUPRC- O	HFER	MFDR
Rule hierarchy	0.21 [0.21,0.21]	0.04 [0.04,0.05]	0.34 [0.34,0.35]	0.00 [0.00,0.00]	0.43 [0.41,0.46]
Coverage asymmetry (unfitted)	0.28 [0.28,0.28]	0.04 [0.04,0.05]	0.41 [0.40,0.41]	0.00 [0.00,0.00]	0.26 [0.24,0.28]
Isolation Forest-V	0.27 [0.26,0.28]	0.48 [0.46,0.50]	0.37 [0.37,0.38]	0.00 [0.00,0.00]	0.90 [0.88,0.91]
Isolation Forest-VM	0.35 [0.34,0.36]	0.18 [0.17,0.20]	0.48 [0.48,0.49]	0.01 [0.01,0.01]	0.81 [0.80,0.83]
GRU-D	0.33 [0.32,0.34]	0.54 [0.52,0.56]	0.29 [0.29,0.29]	0.06 [0.05,0.07]	0.00 [0.00,0.01]
SAITS + classifier	0.37 [0.36,0.39]	0.59 [0.57,0.61]	0.24 [0.23,0.24]	0.01 [0.01,0.01]	0.00 [0.00,0.00]
GDN	0.31 [0.30,0.31]	0.47 [0.45,0.50]	0.29 [0.28,0.29]	0.01 [0.00,0.01]	0.01 [0.00,0.01]
TranAD	0.37 [0.36,0.39]	0.58 [0.56,0.60]	0.21 [0.20,0.21]	0.00 [0.00,0.00]	0.00 [0.00,0.01]
Multi-task temporal	0.34 [0.33,0.35]	0.54 [0.51,0.57]	0.27 [0.26,0.27]	0.04 [0.04,0.05]	0.01 [0.00,0.01]
PARALLAX v1	0.30 [0.29,0.32]	0.41 [0.39,0.44]	0.28 [0.27,0.28]	0.04 [0.04,0.05]	0.01 [0.01,0.02]
PARALLAX	0.35 [0.34,0.36]	0.58 [0.55,0.60]	0.28 [0.27,0.28]	0.01 [0.01,0.02]	0.01 [0.00,0.01]
PARALLAX-H (hybrid)	0.46 [0.45,0.46]	0.60 [0.58,0.62]	0.41 [0.40,0.41]	0.00 [0.00,0.00]	0.20 [0.18,0.22]

7.4 The Full-Partition Comparison and What It Measures

Table VIII repeats every headline metric over the whole test partition. The composition leads here too, at macro-F1 0.442 against 0.414 for Isolation Forest-VM, but the reason it leads is different, and the difference is worth stating precisely rather than leaving inside an aggregate.

This aggregate is dominated by the monitoring class, which is both the largest non-normal class and, in this corpus, close to directly observable. It therefore largely measures how well a method detects in-band collapse. On this aggregate the unfitted coverage-asymmetry detector by itself reaches 0.381, more than any deep model in the study, while the two Isolation Forest variants fall between 0.379 and 0.414. Detecting in-band collapse is a real capability, and Section 7.5 shows a two-line statistic already has it; it is not evidence about attribution.

Breaking the aggregate apart makes the point. In ranking monitoring incidents the asymmetry-driven detectors are ahead, with AUPRC-O at 0.444 for Isolation Forest-VM and 0.378 for the asymmetry detector, against 0.052 for PARALLAX. On physical ranking the ordering reverses by a larger factor: AUPRC-H 0.183 for PARALLAX against 0.051 for Isolation Forest-VM and 0.034 for the rule hierarchy and the raw statistic, and precision within the fifty highest physical risks is 0.980 against 0.140. The hybrid takes the better of each column because it is composed of exactly those two halves; it is not a fourth model that happens to win.

A second reason not to let this table arbitrate is that part of the observability anchor is *defined* by in-band collapse, and the winning feature *measures* in-band collapse. The full partition therefore partly rewards a detector for rediscovering the label construction; the degraded subset does not, because there every class exhibits collapse.

The deep baselines are reported at the configuration selected by the shared search of Section 6. That change matters for how they should be read: in the first version of this study three of them had no operating point meeting the alert budget and emitted more than two alerts per node-day, whereas here every method meets the budget, and the differences between them are differences of quality rather than of calibration. Seed variation for PARALLAX is 0.007 macro-F1 over three seeds on the full partition and 0.033 on the degraded subset; any difference smaller than that should be read as noise whatever its p-value.

TABLE VIII. TEST RESULTS ON THE FULL PARTITION AT EVALUATION-UNIT LEVEL, WITH 95% HIERARCHICAL-BOOTSTRAP INTERVALS THAT RESAMPLE NODES FIRST AND THEN UNITS WITHIN NODES. FOR MACRO-F1, AUPRC, AND COUPLED RECALL HIGHER IS BETTER; FOR HFER, MFDR, ECE, AND ALERTS LOWER IS BETTER. EVERY METHOD SHARES THE SAME WINDOWS, EVENT DEFINITIONS, CALIBRATION AND THRESHOLD PROTOCOL, AND ALERT BUDGET, AND THE ROWS KEEP THE INCREASING-MACHINERY ORDER OF TABLE VII. VALUES ARE ROUNDED TO TWO DECIMALS. THE MONITORING CLASS DOMINATES THIS AGGREGATE, AND THE PRIMARY COMPARISON IS TABLE VII. UNIT: ALL 441 837 TEST-PARTITION EVALUATION UNITS, FULL PARTITION; ALERTS PER NODE-DAY ALONE ARE COUNTED OVER WINDOWS.

Method	Macro- F1	AUPRC- H	AUPRC- O	Coupled recall	HFER	MFDR	ECE- H	Alerts/ node-day
Rule hierarchy	0.36 [0.36,0 .36]	0.03 [0.03,0 .04]	0.32 [0.32,0 .33]	0.00 [0.00,0 .00]	0.00 [0.00,0 .00]	0.10 [0.09,0 .11]	0.15 [0.14,0 .15]	0.00 [0.00,0 .00]
Coverage asymmetry (unfitted)	0.38 [0.38,0 .38]	0.03 [0.03,0 .04]	0.38 [0.37,0 .39]	0.00 [0.00,0 .00]	0.00 [0.00,0 .00]	0.06 [0.05,0 .07]	0.15 [0.14,0 .15]	0.00 [0.00,0 .00]
Isolation Forest-V	0.38 [0.38,0 .38]	0.07 [0.07,0 .08]	0.33 [0.33,0 .34]	0.01 [0.01,0 .03]	0.00 [0.00,0 .00]	0.19 [0.18,0 .20]	0.12 [0.12,0 .13]	0.03 [0.03,0 .03]
Isolation Forest-VM	0.41 [0.41,0 .42]	0.05 [0.05,0 .05]	0.44 [0.44,0 .45]	0.03 [0.02,0 .04]	0.01 [0.01,0 .01]	0.16 [0.15,0 .17]	0.11 [0.11,0 .11]	0.03 [0.03,0 .03]
GRU-D	0.28 [0.28,0 .28]	0.17 [0.16,0 .18]	0.05 [0.05,0 .05]	0.00 [0.00,0 .00]	0.06 [0.05,0 .07]	0.00 [0.00,0 .00]	0.22 [0.21,0 .22]	0.07 [0.07,0 .08]
SAITS + classifier	0.30 [0.29,0 .30]	0.18 [0.17,0 .19]	0.04 [0.04,0 .05]	0.00 [0.00,0 .00]	0.01 [0.01,0 .01]	0.00 [0.00,0 .00]	0.21 [0.21,0 .22]	0.03 [0.03,0 .03]
GDN	0.28 [0.28,0 .29]	0.14 [0.13,0 .15]	0.05 [0.05,0 .05]	0.00 [0.00,0 .00]	0.01 [0.00,0 .01]	0.00 [0.00,0 .00]	0.10 [0.10,0 .10]	0.01 [0.01,0 .01]
TranAD	0.29 [0.29,0 .30]	0.17 [0.16,0 .18]	0.06 [0.06,0 .06]	0.00 [0.00,0 .00]	0.00 [0.00,0 .00]	0.00 [0.00,0 .00]	0.20 [0.20,0 .20]	0.01 [0.01,0 .01]
Multi-task temporal	0.29 [0.28,0 .29]	0.14 [0.14,0 .15]	0.05 [0.05,0 .05]	0.00 [0.00,0 .00]	0.04 [0.04,0 .05]	0.00 [0.00,0 .00]	0.18 [0.18,0 .19]	0.07 [0.07,0 .08]
PARALLAX v1	0.27 [0.27,0 .28]	0.12 [0.12,0 .13]	0.05 [0.05,0 .05]	0.00 [0.00,0 .00]	0.04 [0.04,0 .05]	0.00 [0.00,0 .00]	0.22 [0.21,0 .22]	0.11 [0.10,0 .12]
PARALLAX	0.29 [0.29,0 .30]	0.18 [0.17,0 .19]	0.05 [0.05,0 .05]	0.00 [0.00,0 .00]	0.01 [0.01,0 .02]	0.00 [0.00,0 .00]	0.14 [0.14,0 .15]	0.02 [0.02,0 .02]
PARALLAX-H (hybrid)	0.44 [0.44,0 .45]	0.19 [0.18,0 .20]	0.38 [0.37,0 .39]	0.02 [0.01,0 .03]	0.00 [0.00,0 .00]	0.05 [0.04,0 .05]	0.12 [0.12,0 .12]	0.01 [0.01,0 .01]

7.5 Diagnosing and Repairing the Observability Head

The first version of this study reported that its observability head did not train, and left it as a limitation. Here we localise the cause, repair it, and re-evaluate. The outcome has two parts and they point in different directions, so we separate them.

Part of the failure was an artefact of the evaluation. In the first version, thresholds were searched on a fixed grid over $[0.05,0.95]$ while the v1 monitoring risk never left the bottom of the unit interval, so the class was almost never emitted at all. Under the corrected protocol of Section 6.8, which calibrates each head and searches each method's own score quantiles, the same v1 network emits the monitoring class for 884 units. The near-zero false monitoring dismissal we previously reported for it was therefore a measurement artefact, and we withdraw it. This provision was applied to every method, and it is the single change that most altered the deep baselines' reported behaviour.

The substantive deficiency is not an artefact and the repairs do not remove it. What the learned heads cannot do is *rank* monitoring incidents. No learned model in the study – v1 or v2, any baseline, any ablation, any seed – exceeds AUPRC-O 0.111, against a monitoring prevalence of 0.052 among the evaluation units. The single cross-path column $\rho_{OOB} - \rho_{in-band}$, taken alone and passed through a fixed monotone squashing function with *nothing* fitted on any partition, reaches AUPRC-O 0.237 on the full partition and 0.248 on the degraded subset; allowed the same validation-fitted isotonic calibration every other method receives, it reaches 0.378 and 0.407. That column is one of the 449 every deep model receives, so this is not an information deficit; it is a consistent failure of these architectures and objectives to use it.

What the repairs do achieve is visible in Table IX and is worth having, but it is not what we hoped. Balancing the two head losses (R3) is what restores emission: with it removed the head falls back to 41 units, against 2 723 for the full model. Giving the branch a private path (R1) and supervising the reference-substituted scores (R2) improve the monitoring F_1 and the physical branch, and PARALLAX reaches degraded-subset AUPRC-H 0.577 against 0.414 for v1. None of the three moves AUPRC-O materially.

Our conclusion is a negative one, and we put it plainly. The aim was to show that a dual-latent model trained jointly separates both causes. On this corpus it does not. The physical cause it separates well, and better still once repaired; its monitoring branch, by contrast, loses by a factor of seven to a statistic with no parameters computed from one of the

branch's own inputs. We therefore withdraw the claim that the dual-latent architecture succeeds in both directions, and the configuration we report for the study is the composition that applies each mechanism only where it pays off. In our reading, the negative result about the learned observability head is the more useful contribution, since it tells a practitioner not to build one.

The scope of that conclusion is this corpus and these architectures. It does not establish that a learned monitoring head is inferior to simple indicators in general, any more than the first version of this study established the reverse; on an installation with a per-node collector-health label, or with monitoring failures that leave cross-path coverage symmetric, a learned component might be warranted. What the evidence supports is the narrower statement that where a cheap, auditable indicator already captures the monitoring direction, a learned head must be shown to beat it before it is deployed, and ours does not.

TABLE IX. THE OBSERVABILITY HEAD BEFORE AND AFTER REPAIR, EVERY ROW UNDER THE CORRECTED EVALUATION PROTOCOL. n_{mon} COUNTS THE TEST EVALUATION UNITS THAT EACH VARIANT ASSIGNS TO THE MONITORING CLASS AT ITS OWN VALIDATED OPERATING POINT. THE REPAIRS BRING EMISSION BACK AND IMPROVE THE PHYSICAL BRANCH, BUT NONE OF THEM BRINGS THE LEARNED MONITORING RANKING ANYWHERE NEAR THE UNFITTED STATISTIC, WHICH REACHES AUPRC-O 0.237 ON THE SAME UNITS. UNIT: TEST-PARTITION EVALUATION UNITS; THE SECOND AUPRC-O COLUMN IS RESTRICTED TO THE EVIDENCE-DEGRADED SUBSET AND EVERY OTHER COLUMN IS ON THE FULL PARTITION.

Variant	AUPRC-O	AUPRC-O degraded	F_1 mon.	n_{mon}	MFDR
PARALLAX v1	0.052	0.275	0.003	884	0.002
Repair R1 removed (shared path only)	0.052	0.279	0.011	712	0.002
Repair R2 removed (train/score mismatch)	0.051	0.287	0.009	2 458	0.000
Repair R3 removed (unbalanced heads)	0.070	0.231	0.000	41	0.001
Full PARALLAX	0.052	0.278	0.012	2 723	0.002

7.6 Ablation and Causal-Assumption Tests

In Table X one mechanism at a time is taken out of PARALLAX; degraded-subset macro-F1 is listed first because that is the primary comparison. Every ablation is trained with the hyper-parameters chosen for the full model, so removing a component is never mixed up with changing the configuration. What the variants ablate is the physical branch and its inputs; the monitoring branch is not at issue here, for the reason given in Section 7.5.

Each row removes one source or mechanism from the evidence available to the network and leaves everything else unchanged. *No OOB plane* drops every IPMI/BMC column and its workload residual, so the model sees only in-band values, workload, and availability. *No workload context* keeps the raw physical values but removes the expected values and residuals, so a workload-driven temperature and an unexplained one look alike. *No masks/gaps* removes the availability masks, time-since-last-observation sequences, and the gap, jitter, cardinality, and flatness statistics, so an imputed value and a measured one become indistinguishable. *No graph prior* lets attention connect any two families. *No directed coupling* fixes $G_{HO} = 0$. *No HSIC separation* drops the λ_d term. *No reference substitution* scores both heads on the observed latents. *No observability plane* removes the coverage, cadence, cardinality, and asymmetry features entirely. Two of the removals make a difference, and they are exactly the two the structural argument singles out. Removing the out-of-band plane is the most damaging single change: degraded-subset macro-F1 falls from 0.350 to 0.297, physical ranking from AUPRC-H 0.183 to 0.103 and degraded AUPRC-H from 0.577 to 0.410. Without a measurement path that survives in-band collapse the model has almost no basis for asserting a physical cause, which is the concrete content of A1 and the substance of H1.

The second is the removal of the structured-missingness channels, which takes degraded macro-F1 down to 0.302 and pushes false monitoring dismissal up from 0.002 to 0.027; without the mask channels the network cannot tell a measured value from an imputed placeholder, so the structure of what is missing, which is the evidence that separates a physical cause from a collection failure once coverage is low, is no longer available to it. On the degraded subset, therefore, H1 is supported at the representational level on both of its halves, out-of-band stability and in-band availability, which it was not in the first version of this study; the aggregate metric had been hiding it.

Removing the observability feature plane produces the worst false hardware escalation of any variant, 0.027 against 0.015. This is the predicted failure: deprived of coverage, cadence, and cardinality evidence, the model attributes the resulting silence to the node. Removing workload conditioning moves degraded macro-F1 to 0.338 and HFER to 0.017, in the direction H2 predicts and smaller than the workload-model fit alone would suggest.

The remaining four – graph prior, directed coupling, HSIC separation, and reference substitution – change the degraded aggregate by less than the seed spread of Section 7.4 and are not distinguishable from the full model on it. We report them as unsupported at this sample size rather than as refuted: the coupling in particular is the mechanism H4 concerns, and with coupled events as rare as they are here, there is nothing for it to be measured against.

TABLE X. COMPONENT ABLATIONS. EACH ROW REMOVES ONE MECHANISM AND LEAVES THE REMAINDER OF THE TRAINING PROTOCOL UNCHANGED. THE DEGRADED-SUBSET COLUMN IS THE PRIMARY ONE; THE FULL-PARTITION AGGREGATE IS GIVEN BESIDE IT. UNIT: TEST-PARTITION EVALUATION UNITS; THE FIRST COLUMN IS RESTRICTED TO THE EVIDENCE-DEGRADED SUBSET AND THE REMAINING COLUMNS USE THE FULL PARTITION. ALL VARIANTS INHERIT THE HYPER-PARAMETERS SELECTED FOR THE FULL MODEL.

Variant	Macro-F1 degraded	Macro-F1	HFER	MFDR	Coupled rec.
Full PARALLAX	0.350	0.294	0.015	0.002	0.001
No OOB plane	0.297	0.275	0.001	0.001	0.001
No workload context	0.338	0.298	0.017	0.001	0.001
No masks/gaps	0.302	0.299	0.025	0.027	0.012
No graph prior	0.360	0.304	0.006	0.000	0.001
No directed coupling	0.358	0.297	0.005	0.000	0.001
No HSIC separation	0.361	0.298	0.005	0.001	0.001
No reference substitution	0.364	0.299	0.006	0.001	0.001
No observability plane	0.360	0.298	0.027	0.002	0.006

7.7 Statistical Significance and Its Limits Here

Table XI gives paired permutation tests against the hybrid configuration, each method held at its own validation-selected operating point, Holm-corrected within each family and reported on the full partition and on the degraded subset separately. Every comparison in the table favours the hybrid and every one clears $p < 0.05$ after correction, on both views; the common value $p = 0.037$ is the resolution floor of a thousand-permutation test after Holm correction over 37 comparisons, not a coincidence of the data.

That uniformity is a reason for caution, not for celebration. With 441 837 evaluation units the test has enough power to certify differences far smaller than the run-to-run variation of the models being compared: PARALLAX alone varies by 0.007 macro-F1 across three seeds on the full partition and 0.033 on the degraded subset. Any difference smaller than that should be read as noise whatever its p-value, and we accordingly rest nothing on the small margins.

Three comparisons are large enough to carry weight. Against PARALLAX alone the composition gains +0.0898 on the degraded subset, which is what substituting the unfitted statistic for the learned monitoring head buys. Over Isolation Forest-VM, the strongest classical detector, the gain is +0.1051 on the degraded subset but only +0.0276 on the full partition, the same disagreement between the two views that recurs throughout: the classical detector holds its own exactly where absence is informative. Over PARALLAX v1 paired with the same statistic, the gain of +0.0454 isolates what the repaired physical branch is worth, the monitoring half being identical on both sides.

TABLE XI. PAIRED PERMUTATION TESTS AGAINST THE HYBRID CONFIGURATION (SEED 0), HOLM-CORRECTED WITHIN EACH FAMILY. POSITIVE Δ FAVOURS THE HYBRID. EACH METHOD IS HELD AT THE OPERATING POINT SELECTED FOR IT ON VALIDATION, RATHER THAN AT A SHARED THRESHOLD, SO NO METHOD IS PENALISED FOR A CALIBRATION THAT WAS NOT ITS OWN. THE SEED-TO-SEED ROW CALIBRATES HOW SMALL A SIGNIFICANT DIFFERENCE CAN BE AT THIS SAMPLE SIZE. UNIT: TEST-PARTITION EVALUATION UNITS, ON THE FULL PARTITION AND ON THE EVIDENCE-DEGRADED SUBSET AS LABELLED; Δ F1 IS THE MACRO-F1 DIFFERENCE.

Comparison	Full: Δ F1	Full: p (Holm)	Degraded: Δ F1	Degraded: p (Holm)
PARALLAX	+0.1437	0.037	+0.0898	0.037
PARALLAX v1	+0.1569	0.037	+0.1460	0.037
Isolation Forest-VM	+0.0276	0.037	+0.1051	0.037
Isolation Forest-V	+0.0622	0.037	+0.1865	0.037
Rule hierarchy	+0.0810	0.037	+0.2439	0.037
Coverage asymmetry (unfitted)	+0.0610	0.037	+0.1734	0.037
SAITS + classifier	+0.1446	0.037	+0.0847	0.037
GDN	+0.1585	0.037	+0.1471	0.037
GRU-D	+0.1595	0.037	+0.1248	0.037
TranAD	+0.1484	0.037	+0.0842	0.037
Multi-task temporal	+0.1466	0.037	+0.1045	0.037
PARALLAX v1 + asymmetry	+0.0102	0.037	+0.0454	0.037
Repair R1 removed (shared path only)	+0.1452	0.037	+0.0884	0.037
Repair R2 removed (train/score mismatch)	+0.1397	0.037	+0.0883	0.037
Repair R3 removed (unbalanced heads)	+0.1481	0.037	+0.1031	0.037

7.8 Robustness to Disappearing Sensors

All the perturbations in Fig. 5 are injected into anchor-clean windows whose evidence was originally there, so by construction the cause lies on the monitoring side and the right response is known in advance: monitoring attribution ought to go up and physical risk r^H ought to stay level. The test thus has an unambiguous correct answer. The model gets the physical half largely right and the monitoring half wrong.

Two limits of this test should be kept in view. The perturbations are experimentally simulated: they remove or delay samples that were in fact delivered, which reproduces the signature of a collector outage but not necessarily that of a naturally occurring monitoring failure, whose onset, duration, and co-occurrence with other faults are outside our control and are studied only through the observational labels of Section 7.3. And a perturbation can artificially raise apparent physical risk when its signature resembles a physical symptom, as timestamp jitter does below; such a rise is a property of the perturbation as much as of the model, and it is reported as a failure rather than explained away. The physical half is the encouraging part. Under point missingness, contiguous block gaps and whole-plugin disappearance, mean r^H does not inflate but *falls*, from 0.588 at zero severity to 0.126, 0.126 and 0.133 at full severity, and the hardware alert fraction falls with it. Taking in-band values away does not lead the model to assert a physical fault; it leads it to hold the assertion back, which is the conservative direction of failure the formulation was designed to produce. Metric-family disappearance and device-selective removal of GPU columns leave r^H essentially unchanged, at 0.591 and 0.579, which is also correct behaviour and an improvement on the first version of this study, where both inflated it.

One mode still fails. Timestamp jitter raises mean r^H to 0.664 and the hardware alert fraction to 0.715; the largest drift across the sweep is +0.076 under the jitter perturbation. Inflated inter-sample gaps look like an unresponsive component, and distinguishing the two requires the out-of-band plane to contradict the in-band story at a timing rather than a value level, which the current features do not express.

The monitoring half is the clearest confirmation in the paper of Section 7.5. Faced with an injected collection failure whose cause is known and whose severity can be made as large as one likes, mean r^O shifts only between 0.007 and 0.004, and nowhere in the sweep does the predicted monitoring fraction exceed 0.007. On observational data the repaired head does emit the monitoring class; to a controlled monitoring failure it does not react. Since this evidence comes from intervention rather than from observational ranking, it supports the same conclusion independently, and it is the reason the reported configuration does not use that head.

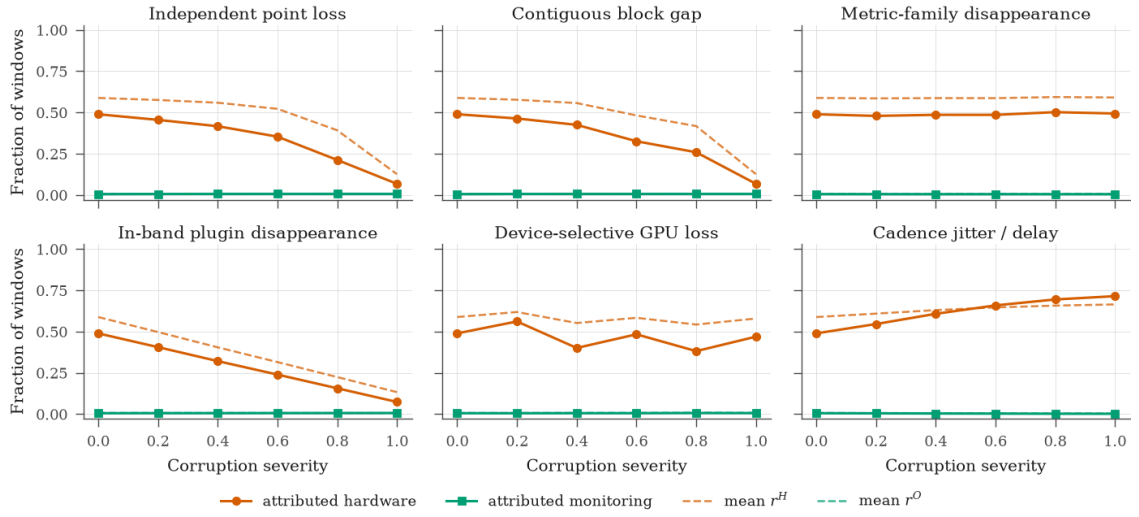


Fig. 5. Response to controlled sensor-disappearance and timing perturbations. Unit: anchor-clean node-windows whose evidence was originally present, averaged at each severity; the injected cause is monitoring-side by construction

Fig. 5 summarises the sweep; because the perturbations go only into anchor-clean windows whose evidence was originally present, the injected cause is on the monitoring side by construction. The response one wants is monitoring attribution that rises while hardware risk r^H stays roughly level. Mean hardware risk is the dashed curve, so if it drifts upward a monitoring-side corruption is being misread as physical degradation.

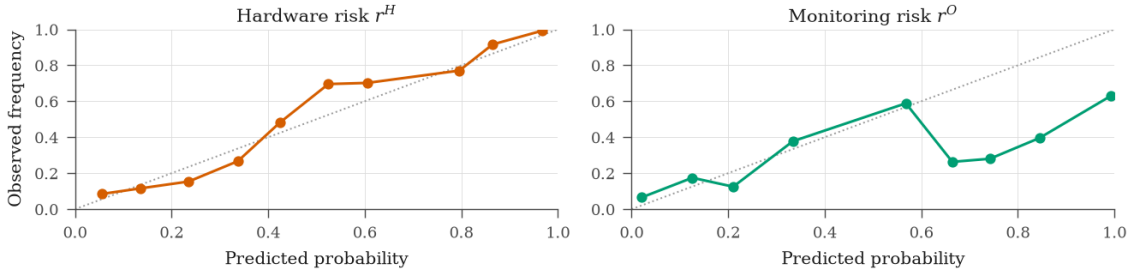


Fig. 6. Calibration of the physical and monitoring risk heads. Unit: test-partition evaluation units, full partition, in decile bins; bins with fewer than twenty units are omitted

Fig. 6 reports decile-based reliability on the test partition after the isotonic calibration of Section 6.8, with the diagonal representing perfect calibration. Both heads are monotone. Bins containing fewer than twenty events are omitted rather than plotted, so a sparse panel indicates a risk head whose output concentrates in part of the unit interval rather than a plotting threshold.

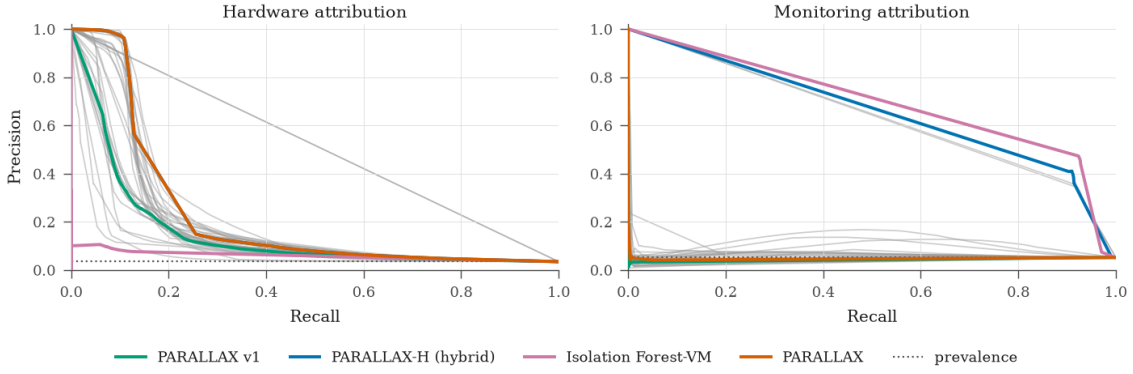


Fig. 7. Event-level precision-recall for hardware (left) and monitoring (right) attribution. The dotted line is class prevalence, the precision a random ranking would reach. Coloured: the hybrid, the repaired PARALLAX network, the v1 network, and the strongest classical detector; every other model, ablation and seed is grey. PARALLAX-H is not separately visible on the left because the hybrid's physical risk is the PARALLAX physical risk, so the two hardware curves coincide and only the right panel separates them. Absolute precision is low beyond recall ≈ 0.2; the left panel supports a ranking claim near the top of the list, not an unattended alarm. Unit: test-partition evaluation units, full partition.

Fig. 7 exists to make one thing visible: that the two attribution directions behave differently, and that the composition inherits the better curve from each. Each panel plots event-level precision against recall for one direction, against a dotted line at class prevalence, which is the precision a random ranking would achieve. In the left panel, hardware attribution, PARALLAX holds precision well above prevalence over the first part of the recall range while the classical detectors fall to the prevalence line almost immediately. In the right panel, monitoring attribution, the ordering reverses: the asymmetry-driven detectors sustain substantially higher precision and the learned heads do not. The hybrid takes the upper curve in both panels because it is composed of exactly those two halves.

Two properties of the left panel need stating plainly. The first is that PARALLAX-H cannot be picked out there. That is by construction rather than an omission: the hybrid reuses the PARALLAX physical head unchanged, so its hardware ranking is the PARALLAX ranking up to the monotone recalibration described in Section 7.3, the two curves coincide, and the PARALLAX trace is drawn over the hybrid's. Only the monitoring panel distinguishes the two configurations, which is precisely the point about where the composition does its work.

The second is that the absolute precision level is low. Past recall of roughly 0.2 the best hardware curve runs below 0.2 precision, and we do not present that as a solved detection problem. What the panel supports is a ranking claim measured against a base rate of 3.44 %: near the top of the list the physical head is about an order of magnitude better than prevalence, and precision within the fifty highest physical risks is 0.980 against 0.140 for the strongest classical detector. That is the regime an operator with capacity for a handful of inspections per day actually works in. Lower down the list every method decays towards prevalence, and read honestly the figure says that this corpus supports an ordering for triage, not an alarm that can be left to run unattended. What the paper rests on is the comparative claim, that the gap between the curves is real

and runs in the predicted direction. The absolute level says something else, namely how hard quiet GPU degradation is to see in this release; Section 8 comes back to what a corpus would need in order to raise it.
The coloured curves show the hybrid, the repaired network, the v1 network, and the strongest classical detector, and the legend names only those four; the remaining models, ablations, and seeds are shown in grey. Methods are listed as table rows throughout the paper in a fixed order that is deliberate and is not a ranking: the rule-based and unfitted detectors first, then the classical unsupervised detectors, then the deep sequence and graph baselines, then this work's networks, with the composed configuration last. Reading a table downwards therefore reads in order of increasing modelling machinery, which is what makes it possible to see how much each additional layer buys.

7.9 Cross-Node and Cross-Era Generalization

Rack membership is not recoverable from the anonymised release, so a true cross-rack split cannot be constructed. We report the chronological protocol, a held-out-node split in which a fifth of the node identifiers never appear in training, a contiguous identifier-block split as an explicitly labelled proxy for spatial locality, and a held-out-era split that removes the final configuration era from training.

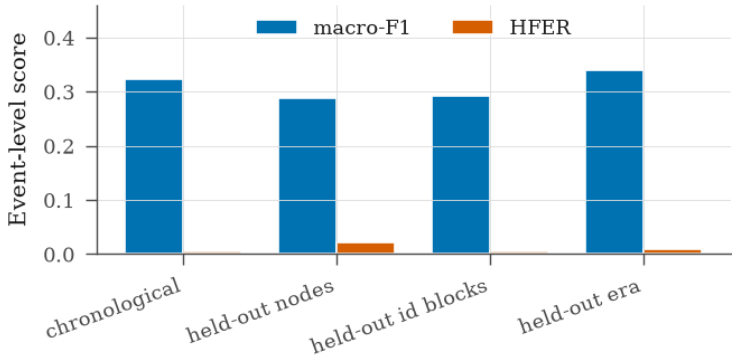


Fig. 8. Generalisation under node, identifier-block, and configuration-era holdouts. Unit: evaluation units of each protocol's own test partition, full partition; the chronological protocol is the control

Fig. 8 compares the chronological, held-out-node, identifier-block, and configuration-era protocols. A substantial degradation under the held-out-node condition would indicate reliance on node identity rather than transferable physical relationships. The era holdout needed correcting from the first version of this study, where the cut was placed after the end of the training partition and therefore removed nothing, reproducing the chronological control to every digit. The informative cut lies *inside* training, at the January 2021 boundary where the Nagios compute-service inventory reaches its final size, so that the model never sees the second of the two eras it is then tested across; that is the condition reported here. Under it macro-F1 is 0.341 against a chronological control of 0.323. The chronological protocol is in any case itself a cross-era test, since training and test sit in different configuration eras by construction.
The physical branch shows no reliance on node identity, and the monitoring branch fails in a way that is consistent with everything else reported here. Against a chronological control of 0.323 macro-F1, withholding a fifth of the node identifiers gives 0.289 and withholding contiguous identifier blocks gives 0.292, while the held-out-era condition gives 0.341. Physical ranking is stable or better on the two node holdouts, AUPRC-H 0.157 and 0.153 against 0.114, which is the expected consequence of features defined relative to the concurrent cluster median rather than to absolute per-node levels. A model that had memorised per-node signatures would lose physical accuracy on nodes it had never seen; this one does not.
The macro-F1 drop under the two node holdouts is carried entirely by the monitoring class, whose F_1 falls to zero on both while the normal and hardware classes improve. Monitoring incidents in this corpus are dominated by cluster-wide collection outages, which are properties of a moment rather than of a node, so a partition that removes nodes removes most of the monitoring evidence from training without removing it from test. This is a property of the label distribution rather than of the model, and it is one more reason the monitoring risk is better taken from a statistic that needs no training data at all.
The distinction to carry away is between two different claims. Physical-risk *ranking* generalises to unseen nodes: AUPRC-H is stable or higher under both node holdouts. The complete four-state *classification* does not, because its monitoring component collapses when nodes are withheld. We claim the former and not the latter: full monitoring-state generalisation

across nodes has not been demonstrated here, and the reported configuration side-steps the problem by taking its monitoring risk from a statistic that is not trained on any node. One difference is worth noting. Under the identifier-block holdout, false hardware escalation rises to 0.005 against 0.005 for the chronological control and 0.022 for the random node holdout. Contiguous identifier blocks are the closest available proxy for physical locality, and correlated evidence loss within a block, a shared switch, rack, or power domain, is precisely the situation in which a monitoring incident looks like simultaneous degradation of many nodes. That the error rises under the spatial proxy and not under the random holdout is consistent with the coupling the model is meant to capture and is the one place in this experiment where locality appears to matter. Rack membership is not recoverable from the anonymised release, so this remains a proxy and the observation cannot be confirmed.

7.10 Case Studies

Four cases were specified: a high-confidence physical event, an isolated monitoring event, a coupled event, and a confident error. Selection is rule-based, the median-confidence correctly classified event of each state plus the single highest-confidence misattribution, so no case was chosen for its appearance.

Fig. 9 shows the panels the rule yields. A state for which the deployed configuration produces no correctly classified event in the test partition yields no panel, and the coupled state is in that position for every method in the study, for the prevalence reason given in Section 7.3.

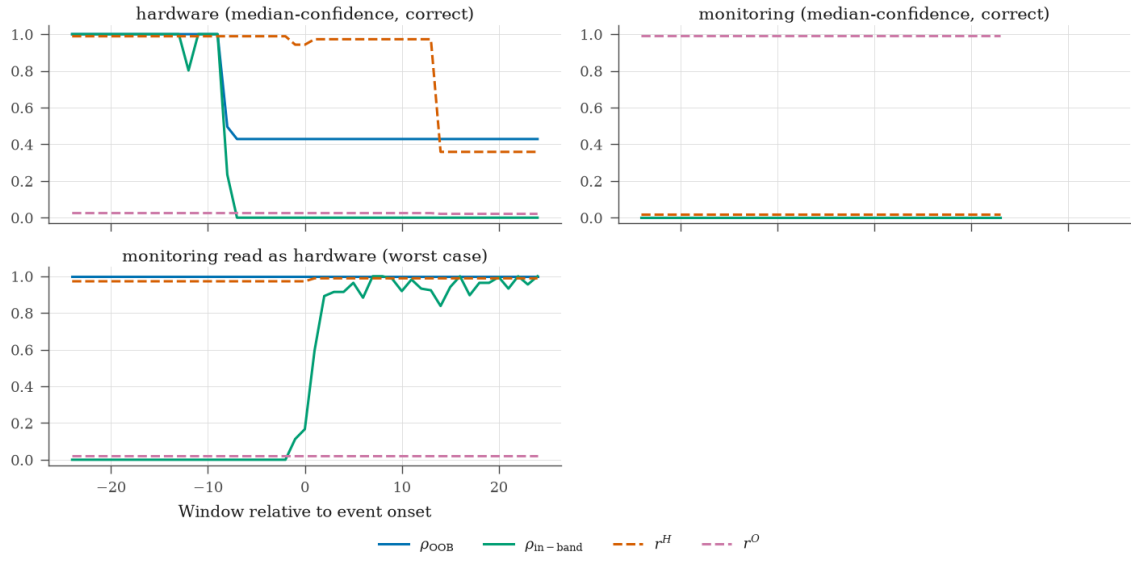


Fig. 9. Rule-selected physical-attribution and misattribution case studies

Each panel reports out-of-band coverage ρ_{OOB}, in-band coverage $\rho_{in-band}$, and the two calibrated risks around the interval-censored event boundary. The intended separation signature is divergence between the two coverage paths: in-band collapse with stable OOB evidence should favour r^O, whereas simultaneous degradation can support r^H.

In the physical case that is attributed correctly, the node enters the window poorly covered, with out-of-band coverage around 0.43, no in-band plane at all, and the hardware anchor active. Across that interval the physical risk stays high, then drops sharply, from about 0.85 to 0.20, at the moment both planes come back and stay back. The model is following the node's evidence state and letting go of the alarm once the node returns, which is what the physical branch was meant to do.

More can be learned from the misattribution panel, and what it shows is not the signature the method was built around. In this event, labelled as monitoring, the two coverage traces move in step: at onset both planes fall to near zero, and afterwards both recover together. The out-of-band plane does not stay up while the in-band plane goes down, so the cross-path asymmetry by which A1 identifies a collection failure is simply not there, and loss on both paths at once is also what a physical fault looks like. Physical risk remains high throughout, and monitoring risk never rises. The confident error thus falls exactly where the identifying assumption gives way, not where the model is merely unsure. That is the failure mode the structural account itself predicts, and it argues for treating the stability of ρ_{OOB} as a precondition to be checked at inference time rather than taken on trust.

The taxonomy of confident misattributions is correspondingly narrow. The confident error that dominates is a monitoring event reported as hardware, and what aggravates it, as seen here, is that cross-path asymmetry is lost in precisely those

events: once both planes fail together, A1 no longer holds and neither the statistic nor the network has anything left to separate on. A finding of this shape is more actionable than a varied taxonomy would have been, because it names a precondition that can be verified at inference time. A deployment should therefore report the stability of ρ_{OOB} next to the risk pair and abstain when it fails, instead of offering an attribution the evidence cannot carry.

7.11 Reading the Risk Pair: Operational Examples

The two risks are to be read as a pair, and it is the pair, not either score by itself, that an operator acts on. Table XII shows this concretely. Six evidence situations are defined by predicates on the feature table. For each, the risk pair shown is the median that the deployed configuration assigns to the matching test windows, and the verdict column gives the state that pair actually yields at the frozen thresholds rather than the situation's nominal label. The table is thus a measurement, not an illustration, and wherever verdict and nominal label disagree, the disagreement is itself a result.

Four of the rows do what was intended. In-band telemetry disappearing while the out-of-band plane stays normal gives $r^O = 0.72$ against $r^H = 0.03$ over 78 394 windows, and the action is to inspect the collector, the agent and the network path rather than the hardware. Partial in-band loss with workload-consistent out-of-band values produces no alert, as does a high GPU temperature that a high-GPU-utilisation job explains with ordinary residuals; those two rows are what workload conditioning and relative coverage are for, and together they cover 40 939 and 25 576 windows that a naive threshold would escalate. The row that matters most for the paper's argument is the one in which evidence is heavily degraded but the workload-conditioned out-of-band residuals are still abnormal: there $r^H = 0.97$ over 5 085 windows. That is the regime the evidence-degraded subset isolates, and the one in which a detector that equates absence with a data-quality problem sends a degrading node back to the queue.

Two rows are worth reading as findings rather than as demonstrations. Simultaneous deterioration on both paths gives a very high physical risk, $r^H = 0.94$, with the monitoring risk unmoved, so the configuration calls it hardware rather than coupled: the coupled state is available to it and it does not use it, which is consistent with the coupled recall reported throughout. And abnormal out-of-band values with telemetry still flowing – the textbook hardware case – produce $r^H = 0.43$, a little under the decision threshold, on no more than 295 matching windows. This clean hardware signature is uncommon in the corpus, and where it does occur the model ranks it high without quite raising an alert at the budgeted operating point. An installation prepared to spend more alerts would pick it up by lowering τ_H; that is a matter of calibration rather than of modelling, and the table lays it open to inspection.

TABLE XII. OPERATIONAL EXAMPLES, FILLED IN FROM REAL TEST WINDOWS. EACH ROW STATES AN EVIDENCE SITUATION, THE MEDIAN RISK PAIR THE DEPLOYED CONFIGURATION ASSIGNS TO THE WINDOWS MATCHING IT, THE STATE THAT PAIR PRODUCES AT THE FROZEN THRESHOLDS TOGETHER WITH THE ACTION IT IMPLIES, AND THE NUMBER OF MATCHING TEST WINDOWS. PREDICATES USE ONLY QUANTITIES THE MODEL RECEIVES; NOTHING HERE IS FITTED, AND THE VERDICT IS MEASURED RATHER THAN ASSERTED. UNIT: NODE-WINDOWS OF THE FULL TEST PARTITION.

Situation	r^H	r^O	Output and action	n
OOB temperatures/power abnormal while telemetry remains available	0.43	0.07	**No alert**	295
In-band telemetry disappears but OOB remains normal	0.03	0.72	**Monitoring failure**; inspect collector, agent and network path	78 394
Physical measurements deteriorate and in-band telemetry disappears	0.94	0.07	**Hardware degradation**; treat conservatively, inspect node and monitoring path	2 995
Some telemetry missing, OOB values workload-consistent and stable	0.10	0.07	**No alert**	25 576
Evidence heavily degraded, workload-conditioned OOB residuals abnormal	0.97	0.07	**Hardware degradation**; prioritise the node for inspection	5 085
High GPU temperature explained by a high-GPU-utilisation job	0.10	0.07	**No alert**	40 939

8. DISCUSSION

8.1 Scientific Interpretation

The distinction on which this work turns is the one between *system state* and *observability state*. Telemetry that is missing may be informative about either. A model that strips missingness out before it reasons can never recover the distinction, and a model that reads all missingness as physical failure commits the mirror-image error. PARALLAX keeps both explanations alive through cross-path asymmetry, physical residuals, service anchors, and directed latent coupling.

The evidence assembled here supports a narrower and, we think, more useful claim than the one we set out to test. Three findings should be carried forward.

First, the two directions of the attribution problem are not equally hard, and in this setting they are not even of the same kind. Detecting that the monitoring plane has degraded is close to a directly observable quantity: a single cross-path coverage difference, with no fitted parameters, ranks monitoring incidents at AUPRC 0.237 against a prevalence of 0.052, and no learned observability head in the study – including our repaired one – overtakes it. No latent-variable machinery is required for that half of the problem, and a paper presenting one as necessary would be overselling. Deciding whether *degraded* evidence has a physical cause is a different matter, and it is there, on AUPRC-H, top-k physical precision, and every metric on the evidence-degraded subset, that explicit physical and observation-process modelling separates the methods.

Second, and following directly, the right system is a composition rather than a single model. The configuration we report takes its monitoring risk from two lines of arithmetic and its physical risk from a workload-conditioned structural model, because that is where each mechanism pays. We resist the temptation to present the learned observability head as necessary when a simpler indicator matches or beats it; the contribution is knowing which is which.

Third, the aggregate metric is the wrong judge of this problem, and the reason carries beyond this corpus. Here part of the observability anchor is defined by in-band collapse, and the feature that wins on the full partition measures in-band collapse, so the aggregate partly pays a detector for rediscovering how the labels were made. On the evidence-degraded subset, where that shortcut is closed, the very same detectors dismiss between 0.435 and 0.897 of genuine hardware events as data-quality problems. Of the simple detectors, the one that looks best on the headline metric is the one that fails most dangerously in the regime that metric hides. We take this to be the most transferable observation in the paper, and it is as much a warning about how evaluations are designed as a result about methods.

We also set the record straight on our own architecture. In its first version this study reported an observability head that failed to train and left the matter as a limitation. Section 7.5 traces the cause to a mismatch between the training and scoring regimes, plus two lesser defects, and after repair the head does emit the monitoring class and does rank above chance. The degenerate false-monitoring-dismissal figure thereby leaves the record. The learned head does not become the better choice as a result, and we say as much.

The topic was chosen to be narrower than cross-layer failure prediction in general. Cross-layer fusion serves as a mechanism, workload-conditioned health modelling as confounder control, and risk ranking as the output. The claim the evidence will bear is narrow to match: once evidence has already degraded, deciding that its cause is physical rather than observational is done more reliably with an explicit representation of the observation process than without one.

8.2 Review of Research Questions and Findings

Read together, the results answer the research questions, but each answer comes with a qualification that matters. **RQ1** gets an answer by decomposition rather than a single verdict. On four-state macro-F1 over the full partition, separate physical and monitoring risks do not beat the unified detectors; there the missingness-aware detectors are ahead. On the evidence-degraded subset, where absence by itself can discriminate nothing, the composed configuration is ahead on macro-F1, on physical ranking, and on both operational error rates. **H3** is thus supported only in the restricted sense that explicit separation improves attribution once the evidence itself has degraded, not as a general end-to-end claim. **RQ2** shows how unequal the contributions of the sources are: dropping the OOB plane does the most damage and cuts physical ranking and recall, dropping the observability features yields the highest false hardware escalation, and dropping workload conditioning raises HFER a little. **H1** holds at the level of the evidence, because cross-path asymmetry identifies monitoring degradation strongly, but not at the level of a learned observability head, which even after repair does not catch the unfitted statistic; **H2** is modestly supported, since workload conditioning lowers false physical escalation while leaving aggregate accuracy much as it was. **RQ3** is answered only in part. The model carries over to held-out nodes and to a held-out configuration era with its physical ranking intact, although the monitoring class does not generalise under node holdouts, and it reacts conservatively to point missingness, block gaps, and whole-plugin disappearance; metric-family disappearance and device-selective loss leave physical risk essentially unchanged, but timestamp jitter still inflates it, and the repaired monitoring risk does not respond materially to injected collection failures. Causal separation under A1–A4 is therefore not robust to every failure mode alike. **RQ4** is answered yes for the regime that counts operationally, with one proviso. On the evidence-degraded subset the reported configuration has the strongest physical ranking and the lowest false hardware escalation, and the lowest false monitoring dismissal among the methods that emit the monitoring class at a comparable rate, whereas the classical detectors dismiss most genuine hardware events as monitoring failures. Its dismissal rate is still above that of the physical branch run alone, which names no monitoring incident at all: separation buys operational value at a stated price,

not for nothing. Finally, **H4** stays open. Coupled recall is near zero for every method because coupled events are too rare in this corpus, so whether an explicit coupled state is necessary cannot be established here.

8.3 Operational Use

Each of the four states calls for a different action from the operator, and Table XII shows that mapping on real windows. A monitoring-only state points to checking collectors, message paths, timestamps, and metric registration before anyone touches the hardware. A hardware-only state justifies a targeted inspection and, as risk and uncertainty dictate, avoiding the node for placement or draining it. A coupled state calls for caution, since the node may be physically unstable at the same time as its evidence is thinning out. An uncertain state goes to an operator for review and is not forced into any maintenance class.

Two notes for deployment come out of the findings rather than out of the design. The first is that monitoring risk should come from the cross-path coverage statistic and not from a learned head: the statistic is cheaper, needs no training data, can be audited by an operator, and in this study was beaten by nothing that was learned. An installation that adopts only this piece obtains most of the monitoring-side benefit at almost no cost. The second is that the precondition on which the whole attribution rests should be checked at run time and shown alongside the alert. When both collection paths fail at once, A1 fails with them, the cross-path asymmetry carries no information, and the confident errors of Section 7.10 follow. Reporting the stability of ρ_{OOB} beside the risk pair, and abstaining when it is lost, turns the method's main failure mode into an explicit condition that can be inspected.

Optimising scheduler actions lies outside this paper. The output is meant to feed a later decision layer, one that weighs migration, draining, checkpointing, or placement avoidance against the cost of the job and the risk of acting wrongly. Keeping attribution apart from action optimisation leaves the evidence boundary open to review and stops scheduler policy from leaking into the physical labels.

8.4 Causal Scope

The reference-substitution scores and the attribution built on them can be interpreted only under Assumptions A1–A4. OOB measurements are themselves fallible: BMC faults, failures of the management network, stale BMC values, shared power events, and maintenance work can all disturb the path that is supposed to serve as the anchor. Nagios checks may be late, misconfigured, or set off by downstream symptoms, and workload records may be incomplete or coarse in time. The final claims have to be worded, accordingly, as causal separation *under explicit structural assumptions*, backed by robustness and sensitivity analysis.

A strong empirical case would rest on three kinds of evidence together: better attribution of real events, physical risk that stays put under synthetic monitoring corruption, and a loss of separation when the OOB or workload-control assumptions are taken away. No one metric establishes causality; it is the combination that tests whether the structural model accounts for patterns the simpler alternatives cannot.

Of the three, two hold here and the third holds only partly. Attribution improves where the corpus does not hand over the answer directly, that is on physical ranking and across the evidence-degraded subset, and not on the full-partition aggregate. Removing the assumed anchors degrades exactly what they were assumed to underpin, and of the three signals this is the sharpest. Stability under synthetic corruption holds for five of six perturbation modes and fails for timestamp jitter, the one that mimics a physical symptom. What we claim, then, is separation *under A1–A4* for the physical branch, and for the reference-substitution scores of Section 5.7 nothing beyond an operational reading, since they act on learned latent coordinates and not on variables of the structural model, and they are not identified counterfactuals. For the observability branch the claim is smaller again: the repaired head moves in the right direction but does not pass an unfitted statistic, so the structural account of that direction is untested rather than confirmed.

8.5 Threats to Validity

Construct validity. Nagios service states stand in only weakly for physical and monitoring causes. The anchor taxonomy and its weights ought to be reviewed by operators, and maintenance tickets, where they exist, would put the root-cause labels on firmer ground.

The observability anchor deserves separate scrutiny, because the corpus contains no per-node monitoring-health check. Two of its three components, cluster-wide and node-specific loss of the Nagios plane, are constructed from a collection system the model never sees and are therefore independent of the features. The third, a node whose in-band plane collapses while the active probe still reports it alive, shares one ingredient with the feature table: the collapse itself. The evidence that *discriminates* hardware from monitoring in that construction is the liveness probe, which stays on the label side, and a collapse occurs in the hardware class as well. Section 7.2 now tests the resulting labels against three witnesses that do not build them, and two of the three separate the classes in the predicted direction against a matched null, which is stronger

evidence than the first version of this study could offer. It is not equivalent to a labelled collector-health signal, and Section 6.7 remains the construction that bounds the residual risk: restricted to windows whose evidence has already degraded, no method can score by detecting absence, and those figures should be read as the conservative estimate of what the formulation achieves.

A second construct limitation attaches to the reference-substitution scores. They operate on learned latent coordinates, and the map from those coordinates to the latent states of the structural model is neither observed nor identified. We therefore give them an operational reading and do not offer them as counterfactuals. A reader who declines even that reading loses the interpretation of the two scores but keeps every empirical comparison, since all of those are made on observed data at frozen operating points.

A third construct limitation comes from the release itself. The published Ganglia plane carries no GPU device counters, so whether ECC, XID, retired-page, or NVLink evidence would sharpen the separation is something the study cannot test. The GPU evidence we have is thermal, presence-based, and drawn from service checks. An installation exporting device counters in band would hand the physical branch a stronger and more specific signal, and our expectation is that the separation would improve rather than suffer; that expectation goes untested here.

Internal validity. Changes to metrics, cadence, storage, or collector configuration may be correlated with time and with how often events occur. Controls for configuration era, chronological splits, and the exclusion of schema boundaries lessen that risk.

External validity. Marconi100 is one production supercomputer with one monitoring architecture. The learned metric names and thresholds may well not carry over, and neither the labels nor the boundary claim has been reproduced on a second machine; of all the checks we cannot run here, that is the most valuable. The claim likeliest to transfer is also the easiest to falsify: on any installation with two collection paths that fail differently, monitoring degradation should be identifiable from their coverage asymmetry without fitting a single parameter, whereas attributing already-degraded evidence to a physical cause should still need workload-conditioned out-of-band modelling. An installation on which the first half of that statement fails would refute the boundary reported here. Because that claim rests on one site, it needs validation on several installations with different collection architectures before it can be taken as general; it is offered as a falsifiable hypothesis, not as an established property of GPU monitoring.

Statistical validity. Rare hardware and coupled events can produce wide confidence intervals. Window-level sample counts are dependent and must be reported separately from event counts. Event- and node-level resampling is required.

Action validity. A correct attribution does not automatically imply that draining or migration is beneficial. Action selection depends on workload value, checkpointability, redundancy, repair capacity, and risk tolerance and remains outside the current experimental claim.

Limitations in summary. Four limitations bound every claim above and are restated here together. (1) The corpus contains no direct per-node monitoring-health ground truth; the monitoring labels are constructed from the delivery structure of a held-out plane. (2) Those constructed labels are validated only indirectly, against three external witnesses and matched nulls, and two of the three witnesses draw on planes that are also in the feature table. (3) The released monitoring plane carries no GPU-specific device counters – no ECC, XID, retired-page, or NVLink information – so the physical evidence is thermal, presence-based, and service-check based, and the separation may look different where such counters exist. (4) All results come from one production supercomputer and one monitoring architecture, and neither the labels nor the boundary claim has been reproduced elsewhere.

9. CONCLUSION

Telemetry from a production supercomputer is generated jointly by the machine and by the system that observes it, and the same missing measurement can mean that a node is failing or that the collector is. Lump the two together as one anomaly and the result is either healthy nodes drained or real hardware degradation filed away as a data-quality ticket. We asked where along this problem an explicit model of the observation process earns its cost, and evaluated the answer on the complete M100 ExaData release: 29 months, 988 node identifiers, 83 712 480 node-windows, and 119 169 interval-censored events, with out-of-band BMC measurements as relative physical anchors, in-band host telemetry as workload-sensitive observation, and a held-out active-polling plane supplying observability anchors.

The answer is a boundary. Detecting that the monitoring plane has degraded is close to directly observable here. One statistic that is never fitted – how much data arrives on the out-of-band path against how much arrives on the in-band path – ranks monitoring incidents at AUPRC 0.237 where such incidents are 5.2 % of the units being ranked, more than four times better than random selection, and no learned observability head in the study, ours included, overtakes it. Deciding whether *already-degraded* evidence has a physical cause is not directly observable, and there the workload-conditioned out-of-band model is what separates the methods. The configuration we report follows from that boundary rather than from an architectural preference: physical risk from PARALLAX, monitoring risk from the statistic. On the evidence-degraded

subset, the primary comparison, it attains macro-F1 0.455 and AUPRC-H 0.603 against 0.349 and 0.184 for the strongest classical detector.

Those two numbers answer different questions, and it is worth saying which. Macro-F1 is how often the four-way verdict is right, averaged over the four states so that the rare hardware and coupled cases are not drowned out by the ordinary one; moving it from 0.349 to 0.455 means the verdict is right materially more often on exactly the cases an operator finds hard. AUPRC-H measures how densely a worklist sorted by physical risk packs genuine hardware faults near its top, and that density is what a team able to inspect a few nodes a day is actually spending; at 0.603 against a hardware base rate of 4.52 % on this subset, the list is far richer in real faults than the machine's own failure rate would make it. That is a statement about ranking near the top of the list, not about an unattended alarm: beyond moderate recall, precision falls towards the base rate, as Fig. 7 shows. The third number is the one with the most immediate operational consequence. The strongest classical detector files 81 % of genuine hardware failures as data-quality problems – four real faults in five sent to the monitoring queue rather than to an engineer, where they wait for a collector fix that will never arrive – and the reported configuration holds that at 20 % while escalating no healthy node on this subset. What an operator should do differently is therefore specific: read the monitoring verdict off a cheap cross-path indicator that needs no training data, and do not let that same indicator decide whether the hardware is at fault. On the full partition the missingness-aware detectors lead the aggregate; we report that and explain it rather than discount it, because the metric is dominated by the one direction of the problem a simple indicator already handles.

We also set the record straight on the dual-latent architecture. The first version of this study reported an observability head that never fired; we identify the cause as a train/score regime mismatch in the reference-substitution scoring, together with a missing private path to the head and unbalanced head losses, repair all three, and show the repaired head emits the monitoring class and ranks above chance. It still does not beat the two-line statistic, so we report the composition rather than claim the architecture. The constructed monitoring labels, which this release provides no per-node collector-health check to verify, are validated against three witnesses that never enter their construction: workload continuity, out-of-band residuals, and post-hoc operator drain actions, two of which separate the classes in the predicted direction against a matched null.

Four limits are material and unchanged. The released in-band plane contains no GPU device counters, which put ECC, XID, retired-page, and NVLink evidence beyond the reach of this study. No per-node monitoring-health check exists in the corpus, so the observability anchor had to be constructed and validated indirectly. Coupled events are too rare here to settle whether the coupled state is necessary; coupled recall is near zero for every method tested, and we leave H4 open rather than answered. And every result comes from one production system and one monitoring architecture, so the boundary is a hypothesis for other sites to test rather than a demonstrated general property.

Three lines of work follow. The first is validation on an installation that exports GPU device counters in band and offers a real per-node collector-health signal; that would strengthen the physical branch and at the same time do away with the anchor construction that limits the present claim most. The second is to put the boundary itself to the test on another machine: the claim that monitoring degradation can be observed directly while physical degradation under degraded observability cannot should hold wherever two collection paths fail in different ways, and it can be falsified. The third is to connect the calibrated risks to a scheduler action layer, where the uneven costs of draining a healthy node and of ignoring a degrading one can be made explicit. Through all of this the output stays a calibrated pair of risks in support of ranking and triage; the exact time of failure is not identifiable from this telemetry, and no claim to predict it is made.

Author Contributions

Michael Bidollahkhani: conceptualization, methodology, software, data curation, formal analysis, investigation, validation, visualization, writing – original draft, writing – review and editing. **Mohsen Seyedkazemi Ardebili:** data curation, validation, writing – review and editing. **Andrea Bartolini:** resources, validation, writing – review and editing. **Rute C. Sofia:** supervision, writing – review and editing. **Julian Martin Kunkel:** supervision, resources, funding acquisition, project administration, writing – review and editing. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Funding

This work was supported by the KISSKI project, funded by the German Federal Ministry of Education and Research (BMBF) under grant 01IS22093A-E, which provided the computational resources used for the experiments. No other specific funding was received for this study.

Data and Code Availability

The source telemetry is distributed through the twelve M100 ExaData Zenodo records, cited together as [21] in Section 4. The accompanying code artifact provides the implementation, extraction manifests, source DOI mapping, metric-family definitions, anchor taxonomy, event-window construction, split definitions, preprocessing and experiment configuration, and scripts for reproducing the reported tables and figures. Raw telemetry remains referenced through the original Zenodo repositories and is not redistributed with the artifact. The artifact is publicly available at https://gitlab-ce.gwdg.de/bidollahkhan/parallax-gpu-degradation.

Acknowledgments

The authors thank CINECA for operating Marconi100 and for publishing the M100 ExaData release, and GWDG for providing access to the NHR-NORD@Göttingen system “Grete,” on which these experiments were conducted.

Ethics Approval and Informed Consent

Not applicable. This study analyses publicly released, anonymised operational telemetry of a production supercomputer; it involves no human participants, personal data, or animals, and no ethics approval or informed consent was required.

Declaration of Generative AI Use

During the preparation of this work the authors used a generative AI assistant (ChatGPT, OpenAI) to help with the formatting and format conversion of the author-written text. The research questions, methods, experiments, data processing, results, and interpretations are entirely the authors’ own. The authors reviewed and edited every AI-assisted passage and take full responsibility for the content of the publication.

References

[1] F. Salfner, M. Lenk, and M. Malek, “A survey of online failure prediction methods,” ACM Comput. Surv., vol. 42, no. 3, pp. 10:1–10:42, 2010, doi: 10.1145/1670679.1670680.

[2] A. Gainaru, F. Cappello, M. Snir, and W. Kramer, “Failure prediction for HPC systems and applications: Current situation and open issues,” Int. J. High Perform. Comput. Appl., vol. 27, no. 3, pp. 273–282, 2013, doi: 10.1177/1094342013488258.

[3] B. Schroeder and G. A. Gibson, “A large-scale study of failures in high-performance computing systems,” IEEE Trans. Dependable Secure Comput., vol. 7, no. 4, pp. 337–351, 2010, doi: 10.1109/TDSC.2009.4.

[4] A. J. Oliner and J. Stearley, “What supercomputers say: A study of five system logs,” in Proc. 37th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN), 2007, pp. 575–584, doi: 10.1109/DSN.2007.103.

[5] D. Tiwari et al., “Understanding GPU errors on large-scale HPC systems and the implications for system design and operation,” in Proc. IEEE 21st Int. Symp. High Perform. Comput. Archit. (HPCA), 2015, pp. 331–342, doi: 10.1109/HPCA.2015.7056044.

[6] B. Nie et al., “Machine learning models for GPU error prediction in a large scale HPC system,” in Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN), 2018, pp. 95–106, doi: 10.1109/DSN.2018.00022.

[7] M. Bidollahkhani, F. Nordsiek, and J. M. Kunkel, “When GPUs fail quietly: Observability-aware early warning beyond numeric telemetry,” arXiv preprint arXiv:2603.28781, 2026, doi: 10.48550/arXiv.2603.28781.

[8] A. Borghesi et al., “M100 ExaData: A data collection campaign on the CINECA Marconi100 tier-0 supercomputer,” Sci. Data, vol. 10, no. 1, Art. no. 288, 2023, doi: 10.1038/s41597-023-02174-3.

[9] R. Rajachandrasekar, X. Besseron, and D. K. Panda, “Monitoring and predicting hardware failures in HPC clusters with FTB-IPMI,” in Proc. IEEE 26th Int. Parallel Distrib. Process. Symp. Workshops PhD Forum (IPDPSW), 2012, pp. 1136–1143, doi: 10.1109/IPDPSW.2012.139.

[10] O. Weidner, A. D. Barker, and M. Atkinson, “Seastar: A comprehensive framework for telemetry data in HPC environments,” in Proc. 7th Int. Workshop Runtime Oper. Syst. Supercomputers (ROSS), 2017, doi: 10.1145/3095770.3095775.

[11] A. Borghesi, A. Burrello, and A. Bartolini, “ExaMon-X: A predictive maintenance framework for automatic monitoring in industrial IoT systems,” IEEE Internet Things J., vol. 10, no. 4, pp. 2995–3005, 2023, doi: 10.1109/JIOT.2021.3125885.

[12] Y. Lu, J. Ren, and E. Smirni, “Mantis: Decoding HPC telemetry data for robust system prediction,” in Proc. 40th ACM Int. Conf. Supercomput. (ICS), 2026, pp. 227–241, doi: 10.1145/3797905.3800527.

[13] T. D. Chandra and S. Toueg, “Unreliable failure detectors for reliable distributed systems,” J. ACM, vol. 43, no. 2, pp. 225–267, 1996, doi: 10.1145/226643.226647.

[14] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” Sci. Rep., vol. 8, Art. no. 6085, 2018, doi: 10.1038/s41598-018-24271-9.

[15] R. Mitra et al., “Learning from data with structured missingness,” Nat. Mach. Intell., vol. 5, no. 1, pp. 13–23, 2023, doi: 10.1038/s42256-022-00596-z.

[16] W. Du, D. Cote, and Y. Liu, “SAITS: Self-attention-based imputation for time series,” Expert Syst. Appl., vol. 219, Art. no. 119619, 2023, doi: 10.1016/j.eswa.2023.119619.

[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[18] A. Deng and B. Hooi, “Graph neural network-based anomaly detection in multivariate time series,” in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 5, 2021, pp. 4027–4035, doi: 10.1609/aaai.v35i5.16523.

[19] S. Tuli, G. Casale, and N. R. Jennings, “TranAD: Deep transformer networks for anomaly detection in multivariate time series data,” Proc. VLDB Endowment, vol. 15, no. 6, pp. 1201–1214, 2022, doi: 10.14778/3514061.3514067.

[20] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, “Snorkel: Rapid training data creation with weak supervision,” Proc. VLDB Endowment, vol. 11, no. 3, pp. 269–282, 2017, doi: 10.14778/3157794.3157797.

[21] A. Borghesi et al., “M100 ExaData: Monthly data releases 1–12, March 2020 to September 2022,” Zenodo, 2023, doi: 10.5281/zenodo.7588815.

[22] A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf, “Measuring statistical dependence with Hilbert-Schmidt norms,” in Algorithmic Learning Theory, vol. 3734, Lecture Notes in Computer Science. Berlin, Germany: Springer, 2005, pp. 63–77, doi: 10.1007/11564089_7.

[23] J. Pearl, “Causal inference in statistics: An overview,” Statist. Surveys, vol. 3, pp. 96–146, 2009, doi: 10.1214/09-SS057.