

Cross-Tool Wear Prediction with Limited Labeled Tools: Baseline Calibration and Validation-Set-Free Model Selection

Yu-yuan Ting

lbq588@gmail.com

National Yunlin University of Science and Technology <https://orcid.org/0009-0009-5566-6890>

Jenn-Yih Chen

National Formosa University

Yung-Cheng Wang

National Yunlin University of Science and Technology

Bean-Yin Lee

National Formosa University

Research Article

Keywords: tool wear prediction, cross-tool generalization, leave-one-tool-out validation, model selection, polynomial induction network, GMDH, interpretable machine learning, PHM 2010

Posted Date: October 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11151884/v2>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Cross-Tool Wear Prediction with Limited Labeled Tools: Baseline Calibration and Validation-Set-Free Model Selection

Yu-Yuan Ting^{1*}, Jenn-Yih Chen², Yung-Cheng Wang¹,
Bean-Yin Lee²

^{1*}Department of Mechanical Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan.

²Department of Mechanical and Computer-Aided Engineering, National Formosa University, Yunlin, Taiwan.

*Corresponding author(s). E-mail(s): d11211003@yuntech.edu.tw;

Abstract

Replacing a worn cutting tool with a nominally identical one in an unchanged process is a common cross-tool situation on a production line, yet a wear model trained on a few labeled tools often fails on the next one, whose signal baseline and initial wear differ. We remove these offsets in the data by initial-baseline normalization of the sensor features, a relative wear target ΔVB , and intersection feature selection across the training tools, so that a newly mounted tool needs only a short calibration window and no adaptation training. With two training tools, tool-grouped cross-validation has only two folds. On the PHM 2010 milling benchmark such tuning selected an almost unregularized degree-2 Lasso that collapsed on the held-out tool ($R^2 = -27$), whereas, within the same model family, the universal threshold, a rule that needs no held-out data, selected a five-term model that generalized (**0.885**); a polynomial induction network whose structure is chosen by the predicted-squared-error criterion is reported alongside as an auditable nonlinear candidate. Under complete leave-one-tool-out validation the polynomial network reaches $R^2 = \mathbf{0.8935}$ and **0.8743** on two folds and fails on the third ($-\mathbf{0.607}$); a five-coefficient Lasso and an untuned tree ensemble match it on the primary fold, so we claim no accuracy advantage, but its explicit terms make a joint-range extrapolation hypothesis for the failure inspectable and consistent with the error pattern. A no-signal baseline that replays the mean training wear curve is strong on average but reaches 38–41 μm RMSE in the accelerated-wear stage, where the sensor-based model holds 9–10 μm .

Keywords: tool wear prediction, cross-tool generalization, leave-one-tool-out validation, model selection, polynomial induction network, GMDH, interpretable machine learning, PHM 2010

1 Introduction

Tool wear directly governs machining quality, throughput, and cost. Tooling accounts for roughly 3–12% of total machining cost, but the indirect losses from unplanned downtime, scrapped workpieces, and line restarts caused by tool failure are typically several times larger [1, 2]. Most production sites still replace tools on fixed schedules or operator judgment; the former leaves 50–70% of theoretical tool life unused, the latter resists standardization. Sensor-based tool condition monitoring, which maps vibration, force, or acoustic-emission signals to flank wear (VB), is the widely pursued alternative [3–5], and deep learning models now routinely report high accuracy on single-tool benchmarks.

The obstacle between such results and a deployable system is *cross-tool generalization*. Even nominally identical tools cut under identical conditions produce systematically shifted signals, driven by edge-geometry microvariation, coating uniformity, and mounting offsets; in the PHM 2010 benchmark, the initial wear of three same-specification tools already spans 23 μm . A model trained on some tools and applied verbatim to a new one extrapolates badly and fails. Relabeling and retraining for every new tool is not economically viable, so the practical question is how a trained model can absorb a new tool’s individual offset without retraining.

We argue this problem is routinely conflated with a harder one. *Cross-condition* generalization (different speeds, feeds, or depths of cut) changes the signal physics, and domain-adaptation networks that learn domain-invariant representations are a reasonable answer there. *Cross-tool* generalization under fixed conditions is different. Analytical wear models express the flank-wear rate as a function of the normal stress, sliding velocity, and temperature at the tool–workpiece interface [6]; with the tool specification, workpiece material, and cutting parameters held fixed, these quantities remain in the same operating regime for every tool, so the dominant wear mechanism is expected to be shared, and the tool-to-tool differences are expected to appear mainly as systematic *baseline offsets* in the signals and a *starting-point offset* in the target. In PHM 2010 this expectation is borne out by two of the three tools, whose total wear increments differ by 0.31 μm despite a 23- μm difference in initial wear (Section 5). It is an assumption of the method rather than a guarantee: a tool whose degradation departs from the shared regime violates it (Section 7). Offsets of this kind do not require adversarial training to remove; they can be eliminated directly in the data. Much of the literature does not separate the two settings [7, 8], which obscures both what existing methods actually solve and how simple the fixed- condition problem can be made.

This paper studies cross-tool wear prediction in a situation that is common on a production line: a worn tool is replaced by a nominally identical one while the process is unchanged, and only a few wear-labeled tool histories exist. It is a deliberately

bounded benchmark case study on PHM 2010: a deployment protocol for that situation, a controlled comparison of complexity-selection rules with two training tools, and a counterexample analysis on the same dataset, organized as three claims that the experiments are designed to test.

1. **Setting and calibration protocol.** We define fixed-condition cross-tool prediction (Section 3) and treat the tool-to-tool differences as offsets that are removed in the data: initial-baseline normalization of the input features (each tool is centered on its own first passes), a relative wear target ΔVB (removing the target’s individual offset), and intersection feature selection (IFS; retaining only features that rank highly on every training tool). A newly mounted tool needs only a short calibration window (signal statistics plus one initial wear measurement) and no adaptation training on the new tool. The protocol is evaluated on PHM 2010 with two training tools per fold, and its claims are made for that setting.
2. **Model selection with two training tools.** With two labeled training tools, a validation split leaves one tool on each side and cross-validation over tools has two folds. We show on PHM 2010 that tuning a degree-2 Lasso by such grouped cross-validation selects an almost unregularized model that collapses on the held-out tool (R^2 down to -27), whereas the universal threshold, a rule that needs no held-out data, selects a five-term model that generalizes (0.885); on the other two folds the two rules are within 0.05 of each other (Section 6.5). This is a controlled comparison within one model family on one benchmark. The polynomial network, whose structure is chosen by the PSE criterion [9], is reported alongside as an auditable nonlinear candidate, not as further evidence for the rule, and we make no general claim about cross-validation in small-sample problems.
3. **An auditable model with a characterized failure.** The regressor is a Polynomial Induction Network (PIN), a GMDH-descended self-organizing polynomial network whose depth, node types, and coefficients are chosen by PSE and whose output is an explicit polynomial. Under complete leave-one-tool-out validation on the six public force and vibration channels it reaches $R^2 = 0.8935$ and 0.8743 on two folds and fails on the third (-0.607). Validation-free sparse linear and tree baselines under the identical pipeline match it on the primary fold, so we claim no accuracy advantage for PIN; what it adds is a nonlinear model whose every term can be inspected and recomputed, and whose explicit terms make a hypothesis for the failure on the third tool (evaluation of the interaction terms outside the joint feature range of the training tools) inspectable and consistent with the error pattern (Section 7), without establishing a sufficient failure criterion. PIN is thus a feasible candidate among printable nonlinear models, not a model the results show should be preferred. A no-signal baseline that replays the training tools’ mean wear curve calibrates how much of the accuracy is attributable to sensing at all.

An exploratory hyperparameter search on the primary fold (Section 6.2) reaches a higher fit, but because the same tool guided the search it is reported as an exploratory result and is not used in any comparison. We do not position the work as a leaderboard entry: published figures on this dataset are not protocol-comparable (Section 2), and all baselines here are run under the identical pipeline.

The remainder of the paper is organized as follows. Section 2 reviews cross-tool strategies, machine-learning wear prediction, polynomial networks, and model selection without held-out data, and maps the identified gaps to the three claims. Section 3 formalizes the problem and its working assumptions. Sections 4 and 5 describe PIN and the data pipeline. Section 6 reports the diagnostic evolution, hyperparameter search, full LOTO validation, model selection with two training tools, and the baselines. Section 7 reads the induced polynomial, analyzes the failed fold, and states the limitations; Section 8 concludes.

2 Related Work

2.1 Cross-tool generalization strategies

Even under identical cutting conditions, nominally identical tools produce systematically shifted signals (edge-geometry microvariation, coating uniformity, mounting offsets), so a model trained on some tools routinely fails on a new one. The dominant response is transfer learning and domain adaptation: adversarial feature alignment for tool condition monitoring [10], deep adversarial domain confusion with MMD alignment across machining processes [7], and attention-based adaptation of transferable frequency components [8]. These methods share a structural deployment premise: the new tool must first supply enough unlabeled signal for a domain alignment or attention-calibration *training procedure* before prediction is possible. The pipeline proposed here also requires a short initial window on the new tool, but uses it only to record baseline statistics, and no adaptation training runs on the target tool. The adaptation architectures are also MB-scale, multi-subnetwork, and effectively black-box. A second line applies stochastic degradation models: multi-stage Wiener processes with Bayesian online updating [11, 12] and interacting-multiple-model Kalman filtering [13]. These provide principled uncertainty quantification for remaining-life estimates but are not designed for high-dimensional vibration feature spaces. Recent work continues along both routes: feature decoupling with domain-adaptive representation learning for cross-condition prediction [14], adversarial domain generalization that needs no signals from the unseen condition but treats wear as a classification task [15], and meta-learning that warms up on historical data and then fine-tunes on one to five labeled trial cuts under the new condition [16]. On PHM 2010, Redjaimia et al [17] name cross-tool variability as a central difficulty and address it with a physics-informed health-indicator label and a multi-branch deep network. Closest in spirit to the present work, Lim and Kim [18] argue that the healthy baseline shifts from one tool and mounting to the next and rebuild a per-tool baseline from a short initial healthy segment, without fault labels, for wear diagnosis. The approach taken here is of that kind: remove the cross-tool shift in the data itself (baseline normalization, ΔVB target, intersection screening) and fit a single explicit polynomial, so that a new tool needs only a short calibration window: signal statistics for feature normalization, plus a one-time initial wear measurement to anchor absolute VB, with no adaptation training.

Gap 1. Most of this literature either targets cross-condition transfer, which needs signals or labels from the new condition, or does not separate the cross-individual case

under fixed conditions from it; the calibration-only case (same process, new tool) is rarely defined and treated on its own.

2.2 Machine-learning tool-wear prediction

Sensor-based wear prediction has moved from hand-crafted features with shallow regressors [3] to deep architectures on PHM 2010 and similar benchmarks [4, 5], and more recently to multi-sensor fusion with graph neural networks and Transformers that jointly regress remaining life and classify health state [19]. Two responses to the cost of labeled data have emerged. One embeds physics: a thermo-mechanical force model that includes wear reduces the dependence on large experimental datasets [20]; a fitted wear-evolution law fused with a data-driven model through a particle filter enforces monotonic wear with few sensors [21]; and symbolic regression with recursive updating yields interpretable closed-form expressions for wear prognostics [22]. The other designs for few samples directly, for example stacked recurrent networks trained on a few tens of samples [23]. Interpretability is pursued through explicit-formula architectures: Kolmogorov–Arnold networks [24] applied to wear monitoring [25] and, on PHM 2010, a BiLSTM–KAN model with SHAP-based feature refinement [26].

Gap 2. These studies report accuracy on single-tool or random splits far more often than on held-out tools, and, whether the model is a deep network, a symbolic expression, or a hybrid, they seldom state how model complexity was chosen when only two or three labeled tools exist; the reliability of that choice is, to our knowledge, not examined on PHM 2010.

2.3 Polynomial networks

GMDH [27, 28] introduced self-organizing polynomial modeling: layered low-order polynomials whose structure, depth, and complexity are chosen by an external criterion rather than by the analyst [29]. AIM [30] engineered this into richer node types, the validation-set-free PSE criterion [9], and statistical coefficient pruning; machining applications exist for drilling and milling tool life [31, 32], all in single-tool settings (Section 4). The recently proposed KAN [24] shares the goal of accurate models with explicit symbolic output via learnable spline activations, and has been applied to tool wear monitoring [25], but published KAN studies remain single-tool; cross-tool generalization of explicit-formula models is, to our knowledge, untested ground.

Gap 3. Explicit-formula models have not been evaluated under leave-one-tool-out protocols, and when one of them fails on a tool, the mechanism of the failure is rarely analyzed at the level of the fitted terms.

2.4 Model selection without held-out data

GMDH chooses network structure by an *external* criterion computed on a subsample withheld from coefficient fitting [33], which is a validation split in all but name. The alternative is to estimate prediction error from the training data with a complexity penalty: Mallows’ C_p [34], AIC [35], and BIC [36] charge each parameter a fixed cost; PRESS [37] and generalized cross-validation [38] approximate leave-one-out error in closed form without refitting; and the predicted squared error criterion [9] is the

member of this family used in GMDH-type networks, with an explicit prior-variance term (Section 4.2). Efron [39] shows that such covariance penalties and cross-validation [40, 41] estimate the same quantity, and that the penalties can have lower variance. Two properties of cross-validation matter at small sample sizes: its estimates carry large error bars when n is small [42], and using the same folds both to tune and to assess a model gives optimistic estimates unless the tuning is nested [43], which requires more groups than three tools provide. In sparse regression, the universal threshold of Donoho and Johnstone [44] sets the Lasso penalty from a noise-level estimate alone, with no resampling. One caveat applies to all penalty criteria: an adaptive structure search inflates the effective number of parameters beyond the raw count [45], so the penalty is a lower bound on the complexity actually explored.

Gap 4. Tool-wear studies that tune model complexity almost always do so by cross-validation, over passes or over tools; with two labeled training tools the former leaks within-tool autocorrelation and the latter has two folds, and how either behaves in this regime has not, to our knowledge, been tested against validation-free criteria on a public wear benchmark.

2.5 PHM 2010 and comparability

On PHM 2010 itself, Martinez-Arellano et al [1] gave the first systematic account of cross-tool heterogeneity: C1 wears fastest in the first ~ 20 passes yet ends with the smallest total increment, and models trained on it transfer poorly. Our measurements agree (Section 5), and the pipeline proposed here is a direct response to that observation. Subsequent work spans CNN and hybrid deep architectures [4, 5] and feature-selection-plus- optimization ensembles, e.g. XGBoost with PSO-optimized components reporting $R^2 \geq 0.96$ [46]. We deliberately do not rank ourselves against such figures: sensor channels (studies differ in whether they exploit the force, vibration, or acoustic-emission channels), feature pipelines, and train/test protocols differ across papers, and the evaluation protocol of Lin et al [46] is not specified at the level needed for a matched comparison. All results in this paper use the six public force and vibration channels (excluding acoustic emission), and all baselines are run under the identical pipeline (Section 6).

Gap 5. Published PHM 2010 cross-tool validations almost universally report only the C1+C4→C6 split, so the behavior of a method on the remaining folds, including its failures, is left unreported.

Table 1 maps these gaps to the three claims of Section 1 and to the sections that deliver the evidence.

3 Problem Formulation

Flank wear follows the classical three-stage pattern of break-in, steady-state, and accelerated wear, with the steady-state region covering roughly 70% of tool life and carrying the practical value of prediction: tracking wear accurately enough to schedule tool change before the accelerated stage. Cross-tool prediction is formulated here as follows: given complete wear-labeled histories of a set of training tools, predict the wear of an unseen tool of the same specification under the same cutting conditions,

Table 1 Research gaps identified in Section 2 and where this paper addresses them.

Gap	Response in this paper	Where
1: calibration-only cross-tool case not treated on its own	Claim 1: fixed-condition setting defined; offsets removed in the data with a calibration window and no adaptation training	Sections 3, 5
2, 4: complexity selection with two or three labeled tools not examined	Claim 2: grouped cross-validation vs. a validation-free rule compared within one model family on all three folds	Sections 4.2, 6.5
3, 5: explicit-formula models not evaluated leave-one-tool-out; failures unreported and unanalyzed	Claim 3: complete LOTO for PIN and all baselines; the failed fold analyzed at the level of the fitted terms	Sections 6.3, 7

using no wear labels from that tool beyond a short calibration window after mounting. Throughout this paper, *cross-tool* therefore refers to different *individual cutters of one nominal specification* (the specification reported for the dataset, here a 6-mm three-flute tungsten-carbide ball-nose cutter), mounted on the same machine and cutting the same workpiece material with the same parameters. It does not refer to tools of different nominal material, coating, geometry, or dimensions; what varies between tools is unit-to-unit variation from manufacturing and mounting, which is not measured directly here and is observed only through its effect on the signals and the wear trajectories (Section 5).

3.1 Degradation decomposition and the ΔVB target

For tool j at pass t we decompose flank wear as

$$VB_j(t) = VB_{0,j} + \Delta VB_j(t), \quad (1)$$

where $VB_{0,j}$, the mean wear over the first W_{INIT} passes, is an individual constant reflecting the as-manufactured edge state, and $\Delta VB_j(t)$ is the accumulated wear increment. The initial offsets differ substantially between individuals (44.15 to 67.64 μm across the PHM 2010 tools, Table 2), but these differences stem from initial edge condition, not from different wear mechanisms. Predicting ΔVB instead of absolute VB therefore has three consequences. First, the model no longer has to learn each tool’s starting point along with its wear progression; the individual offset is removed at the source. Second, the target moves into the training domain: in VB space the test tool’s upper range exceeds the training tools’ by tens of microns and the model must extrapolate, whereas in ΔVB space the test tool’s total wear increment exceeds the larger of the two training tools’ by only about 0.3 μm , so the range is essentially fully covered. Third, the transformation is causally deployable: $VB_{0,j}$ is obtained by measuring wear over the first W_{INIT} passes of a new tool, requiring no knowledge of its future. The cost is a one-time initial wear measurement per tool, not a relabeled training set. The matching input-side correction, initial-baseline normalization of the features, is described in Section 5.

3.2 Physical basis and working assumptions

The signal-side premise is the force–wear chain: as the edge dulls, contact area and normal force grow (in the sense of Archard’s wear relation). This increased cutting force is measured directly by the three-axis dynamometer, and also appears in the vibration channels through the tool–spindle transfer dynamics [47]. Under fixed cutting conditions this response is monotone for most amplitude-type features, which is what licenses Pearson correlation as a lightweight, small-sample-stable ranking statistic for feature screening even where the true relation is nonlinear. It also yields a testable physical prediction that the induced models can confirm or refute: the feed direction carries the largest dynamic load, so feed-direction (x -axis) force features should be the most wear-sensitive.

Three working assumptions underlie the pipeline, each falsifiable by the experiments of Section 6. (i) Tools of the same specification, free of early abnormal degradation, share the same wear dynamics; individual differences are confined to offsets, not mechanisms. This is a population-level assumption about quality-conforming tools, and Section 6.3 shows both where it holds and where it breaks. (ii) Features that correlate strongly with wear on *all* training tools are more likely to carry a transferable wear signature than features that excel on one tool only, which is the premise of intersection screening. (iii) The ΔVB transformation geometrically removes target out-of-distribution risk, provided the test tool’s total wear increment does not exceed the training tools’ coverage. None of these is asserted a priori; the staged experiments are designed so that each assumption’s failure would be visible as a specific performance signature.

4 Polynomial Induction Network

The Polynomial Induction Network (PIN) follows the self-organizing polynomial modeling lineage of GMDH [27, 28] and its commercial successor AIM [30]. Abductive polynomial networks have prior applications in machining: tool-life prediction in high-speed milling [32] and in drilling [31]. These studies established the feasibility of polynomial networks for tool wear modeling, but all operate on a single tool or under fixed cutting conditions; cross-tool generalization was not addressed. This work extends the approach to the cross-tool setting by combining PIN with the target transformation and normalization strategy of Section 3 and the feature screening of Section 5.

4.1 Kolmogorov–Gabor polynomial and network structure

PIN approximates the Kolmogorov–Gabor polynomial

$$y = a_0 + \sum_i a_i x_i + \sum_i \sum_j a_{ij} x_i x_j + \sum_i \sum_j \sum_k a_{ijk} x_i x_j x_k + \cdots, \quad (2)$$

whose complete expansion is intractable in practice: a full second-order expansion of the 306-dimensional feature set in this study already exceeds 47,000 terms, far beyond the 630 available training samples. GMDH resolves this by a layered approximation:

each layer fits only low-order polynomials of two or three variables, and the outputs of one layer become candidate inputs of the next, so that high-order interactions are built up incrementally under a fixed sample budget [27].

A PIN network is composed of typed polynomial nodes. *Normalizer* nodes linearly rescale each raw input, $z = b + wx$, with the scale set from the feature–target correlation; *Unitizer* nodes invert the scaling at the output. Between them, four functional node types are searched:

- *Single* (layer L1): a cubic polynomial of one input, $\hat{y} = a_0 + a_1x + a_2x^2 + a_3x^3$ (4 coefficients). L1 identifies the single most predictive feature and its best nonlinear mapping, and its selection is directly interpretable: if L1 selects the peak-to-peak amplitude of the x -axis cutting force, the network is asserting that feed-direction cutting-force amplitude is the strongest single indicator of wear.
- *Double* (L2+): a bivariate polynomial with cross term, $\hat{y} = a_0 + a_1x_i + a_2x_j + a_3x_ix_j + a_4x_i^2 + a_5x_j^2 + a_6x_i^3 + a_7x_j^3$ (8 coefficients), extending the classical GMDH quadratic element with pure cubic terms.
- *Triple* (L2+): the analogous trivariate polynomial with all first- and second-order terms plus the trivariate product and pure cubic terms (14 terms in total); more expressive, but requiring more samples.
- *White* (L2+): a purely linear combination of m intermediate variables, $\hat{y} = a_0 + \sum_{l=1}^m a_l z_l$, which fuses signals from different physical channels in the manner of multi-sensor fusion.

4.2 Validation-set-free model selection: PSE

Candidate nodes are ranked by the Predicted Squared Error criterion [9],

$$\text{PSE} = \text{FSE} + \text{KP}, \quad \text{FSE} = \frac{1}{N} \sum_{t=1}^N (\hat{y}_t - y_t)^2, \quad \text{KP} = \text{CPM} \cdot \frac{2K}{N} \sigma_p^2, \quad (3)$$

where K is the number of polynomial coefficients in the model, N the number of fitting samples, σ_p^2 an estimate of the model error variance that the implementation derives from the training data by a fixed rule, so that no prior variance has to be supplied by the user (the rule belongs to the proprietary implementation; see Code availability), and CPM (complexity penalty multiplier) a hyperparameter controlling the strength of the complexity penalty. FSE rewards fit; KP grows linearly with K and penalizes over-parameterization, so minimizing PSE trades expressiveness against generalization automatically.

The property of PSE that matters most in this application is that it requires *no validation set*: the penalty term is computed on the training data itself. Classical GMDH must split the data into a training set and a selection set [29], which is costly when each tool contributes only 315 samples; PSE-based selection uses all training data [30]. With only two training tools available, this is a structural advantage rather than a convenience, and it is one of the three pillars of the proposed approach.

4.3 Coefficient carving and layered induction

Not every term of a candidate polynomial earns its place. PIN prunes coefficients by a t -statistic-guided rule (*carving*): the candidate is first fitted by ordinary least squares; each coefficient’s two-sided p -value is computed with $N - K$ degrees of freedom; terms with confidence below the carving level CL (i.e., $p_j > 1 - \text{CL}$) are removed; and the surviving terms are refitted. Two caveats keep this honest: successive passes of the same tool are serially correlated, so the i.i.d. assumption behind these p -values does not strictly hold, and the operating threshold found by the hyperparameter search (CL = 0.75, i.e. retaining terms with $p \leq 0.25$) is far more lenient than conventional significance levels. We therefore use the t -statistics as a *confidence-guided pruning heuristic*, a principled way to rank and discard weak terms, rather than as formal statistical inference. Every coefficient in the final polynomial corresponds to a term that survived this screen, which keeps the model both compact and readable.

Network construction is a layer-by-layer greedy search (Fig. 1). Initialization computes a floor PSE from the best single linear feature; any candidate must beat it to survive. L1 fits a Single node to every input feature and retains the best survivors. Each subsequent layer takes the original features plus all surviving nodes of previous layers as candidate inputs, searches Double, Triple, and White combinations, and keeps nodes that improve the incumbent PSE. Induction stops when a preset maximum depth is reached or when no candidate improves PSE, which is the criterion’s judgment that added complexity would no longer pay for itself.

PIN exposes four model hyperparameters (CPM; CL; maximum depth; L1 survivor count), each with a direct statistical meaning, plus two preprocessing parameters introduced in this study (W_{INIT} , the initial-baseline window, and T , the intersection screening threshold; Section 5). Training reduces to a sequence of OLS fits, $O(K^2N + K^3)$: the full induction runs in about 300 s of wall-clock time on a laptop CPU (Intel Core Ultra 5 125H, single-process Python implementation) with no GPU or deep-learning framework. Inference is a single polynomial evaluation (under 1 ms per sample with a sub-kilobyte parameter footprint), so the trained model can run on edge hardware at the machine.

5 Data and Feature Engineering

5.1 PHM 2010 dataset

Experiments use the PHM 2010 milling dataset [48]: full life-cycle cutting on a Rödgers Tech RFM760 high-speed CNC mill (down milling, 6-mm three-flute tungsten-carbide ball-nose cutter, HRC52 stainless steel, 10,400 rpm, 1,555 mm/min feed, $a_p = 0.2$ mm, $a_e = 0.125$ mm, 315 passes per tool). The dataset reports this single nominal specification for all six cutters and provides no per-tool manufacturing, substrate, or coating measurements; C1, C4, and C6 are therefore treated as three individual units of one nominal specification, in the sense of cross-tool defined in Section 3, without any claim that their substrates or coatings were verified to be identical. Three of the six tools (C1, C4, C6) carry flank wear labels: after each pass, flank wear was measured optically on each of the three flutes, and we take the maximum across flutes as VB. The

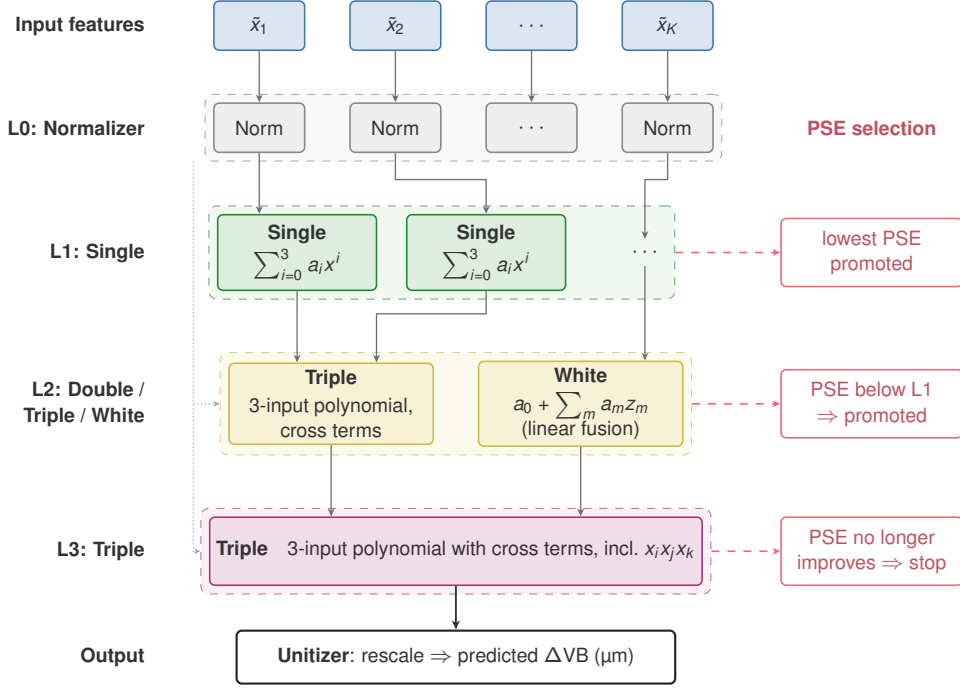


Fig. 1 Layered induction in PIN. The K screened features ($K = 21$ in the exploratorily selected model) are normalized (L0); L1 fits a cubic Single node per feature and retains the top- s survivors ($s = 10$ here); from L2 on, candidate inputs are the raw features (dotted arrows, schematic) plus all earlier survivors, and PSE selects the node type (Double, Triple, or White). Each layer promotes its lowest-PSE nodes (red dashed arrows); induction stops automatically once PSE no longer improves, and the Unitizer rescales the output to ΔVB in μm .

public release contains seven channels: three-axis cutting force, three-axis vibration ($x/y/z$), and acoustic-emission RMS. All modeling here uses the six mechanical channels (force and vibration); the acoustic-emission channel, whose frequency band and noise characteristics differ markedly from the mechanical channels, is not used.

The measurement system on this machine is described by Li et al [49] (Fig. 2): a Kistler three-component platform dynamometer mounted between the workpiece and the machine table measures the cutting forces, three Kistler piezoelectric accelerometers and a Kistler acoustic-emission sensor are mounted on the workpiece, the channels are digitized by a National Instruments data-acquisition system, and flank wear is measured off-line with a LEICA MZ12 microscope. The PHM 2010 release itself contains no photographs of the tools or of the worn cutting edges; the wear information it provides is the per-flute VB value of every pass.

Table 2 shows the cross-tool heterogeneity that motivates the target transformation of Section 3: the initial wear VB_0 spans a $23\text{-}\mu m$ range across tools, while the total wear increments $\Delta VB_{\max}$ of C4 and C6 (the closer pair) differ by only $0.31\text{ }\mu m$. In VB space, by contrast, C6 exceeds C1’s maximum by $62\text{ }\mu m$, forcing polynomial extrapolation exactly where it is least reliable. Validation follows a leave-one-tool-out (LOTO) protocol (630 training / 315 test samples in every split): the test tool

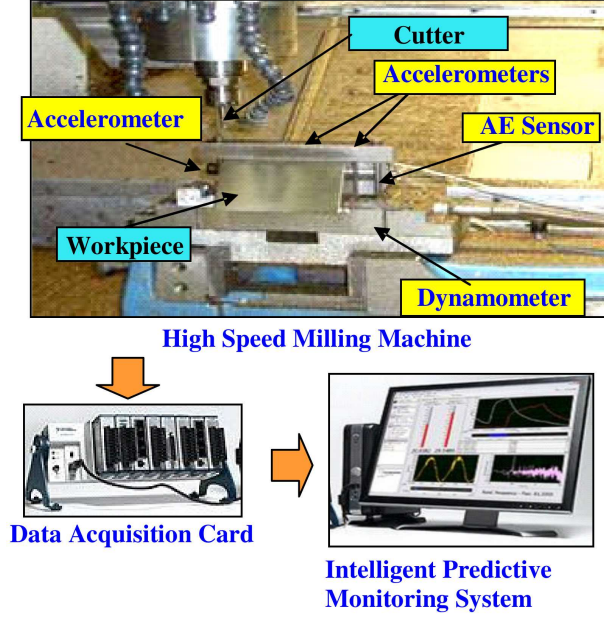


Fig. 2 Experimental set-up of the source study on the same Rödgers RFM760 machine and sensor system: cutter, workpiece, Kistler dynamometer, accelerometers and acoustic-emission sensor, and the data-acquisition chain. Reproduced from Li et al [49] (their Fig. 2) under the Creative Commons Attribution 3.0 United States License. The photograph documents the 2009 preparatory study of the same group; its cutting parameters (23,600 rpm, two-flute cutter) differ from those of the PHM 2010 runs listed above.

Table 2 Wear statistics of the three labeled tools (units: μm).

Tool	VB_0	$VB_{\max}$	$\Delta VB_{\max}$
C1	56.74	172.69	115.95
C4	44.15	210.92	166.77
C6	67.64	234.72	167.08

contributes nothing to model training, feature screening, or the global normalization statistics; its own first W_{INIT} passes supply only its per-tool baseline (feature means from signal, VB_0 from wear measurement), mimicking the calibration a newly mounted tool would receive in deployment.

5.2 Feature extraction and normalization

Each pass yields six channels of $\sim 220,000$ samples at 50 kHz. Per channel, 51 statistical features are extracted in three domains: 13 time-domain (mean, standard deviation, RMS, peak-to-peak, shape / crest / impulse factors, skewness, kurtosis, etc.), 8 frequency-domain (spectral centroid, RMSF, dominant frequency and amplitude, spectral entropy, etc.), and 30 time-frequency features from a level-4 db4 wavelet

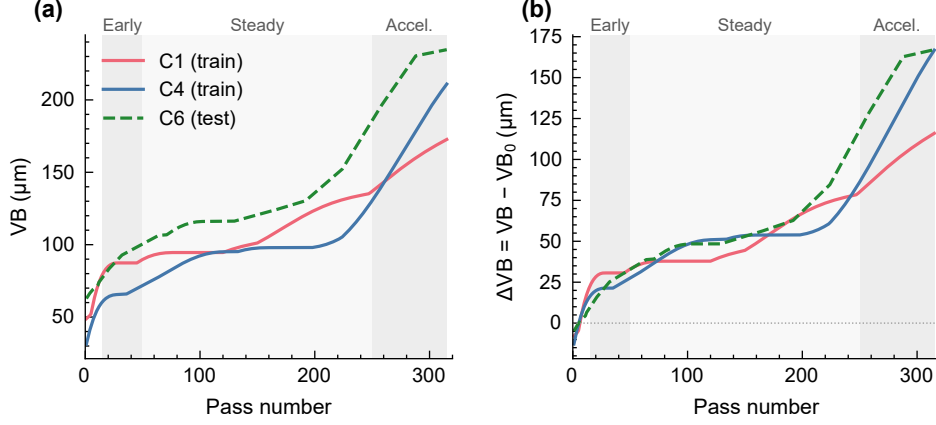


Fig. 3 Per-pass wear trajectories of the three labeled tools. (a) Absolute flank wear VB : C1’s break-in behavior (steep rise over the first ~ 25 passes, then a long plateau) is visually distinct from C4’s and C6’s steadier increase. (b) Relative wear increment ΔVB : the three curves converge far more closely, illustrating the target-transformation rationale of Section 3. Shaded bands mark the wear stages used for the stage-wise error analysis of Table 9: early (passes 15–50), steady (50–250), and accelerated (250–315); passes 1–14 contain the calibration window.

packet decomposition (16 sub-band energies, 8 sub-band RMS values, 4 sub-band kurtoses, 2 packet entropies), for a total of $51 \times 6 = 306$ features. The full list is given in the supplementary material (S1). At 10,400 rpm the tooth passing frequency is 520 Hz, so each 1.56-kHz WPD sub-band spans roughly three TPF harmonics; the sub-band RMS features that recur in the final models correspond to high-order harmonic bands.

Two normalization schemes are compared. Static z -scoring uses the global mean and standard deviation of the training tools; if the test tool’s signal distribution is shifted, as it is here, that shift passes straight into the model input. *Initial-baseline normalization* instead centers each tool on itself:

$$\tilde{x}^{(C_i)}(t) = \frac{x^{(C_i)}(t) - \mu_{\text{init}}^{(C_i)}}{\sigma_{\text{global}}}, \quad (4)$$

where $\mu_{\text{init}}^{(C_i)}$ is the feature mean over the first W_{INIT} passes of tool C_i and σ_{global} is fixed from the training tools. Operationally this amounts to a short calibration period after tool change (run W_{INIT} passes, record the baseline, and predict from deviations thereafter), which is feasible on a production line and requires no prior data from the new tool. The same “model the increment from an individual baseline” logic has a precedent in battery cycle-life prediction [50].

5.3 Intersection feature screening

All 306 features track the same wear trend, so the feature set is highly collinear and the covariance matrix is close to singular; screening is a necessity, not a refinement. The obvious approach, ranking features by $|r|$ (Pearson) against ΔVB on the pooled training set, has a failure mode: a feature that ranks first on one tool and hundredth

on the other can still enter the top 50 on pooled statistics, yet its high rank reflects one tool’s idiosyncrasies and transfers unpredictably.

The intersection feature selection (IFS) used here keeps only features that rank highly on *both* training tools: rank features by $|r|$ separately on C1 and on C4, take the top T of each, and retain the intersection. With $T = 75$ this yields 37 features; at the final operating point $T = 60$ it yields 21. The rank correlation between the two tools’ feature orderings (Spearman $\rho = 0.7150$) is high enough for a stable intersection to exist, yet low enough that intersection and pooled ranking select materially different sets. The resulting features are cross-tool stable by construction: they are retained because they correlate strongly with wear on two tools independently, which makes them more likely to reflect a shared wear signature than single-tool accidents. Section 6 quantifies the effect (R^2 rising from 0.589 with pooled top-50 screening to 0.8935 with intersection screening, all else equal).

The ranking statistic is a design choice, and a linear one. Section 6.3 (Table 6) repeats the entire pipeline with Spearman’s ρ , mutual information, and the maximal information coefficient (MIC) in place of $|r|$. None of them surpasses Pearson on more than one fold, and MIC is uninformative on this benchmark: because the released wear labels form a smooth, monotone series, any feature that is itself monotone in pass count attains $\text{MIC} \approx 1$, which happens for 107–177 of the 306 features per tool, so the top-75 ranking is decided by ties. Pearson $|r|$ is retained.

6 Experiments and Results

Unless noted otherwise, models are trained on C1+C4 (630 samples) and evaluated on C6 (315 samples), reporting R^2 , RMSE, and MAE on the test tool only:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad \text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad \text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (5)$$

where y_i and $\hat{y}_i$ are the measured and predicted wear of pass i , $\bar{y}$ is the mean measured wear of the test tool, and n is the number of test passes. R^2 is computed against the test tool’s own mean, so $R^2 < 0$ means the model is worse than predicting that constant. This split mimics the deployment case of predicting for a newly mounted tool from existing tool histories; the remaining LOTO folds are reported in Section 6.3. Because VB_0 is a measured per-tool constant, error metrics are identical in ΔVB and reconstructed-VB space, and each is reported once. The test tool’s first W_{INIT} passes serve as its calibration window; headline metrics include them for comparability with prior work, and Section 6.2 gives the excluded-window figures, which differ negligibly.

6.1 Diagnostic evolution

Rather than presenting the final configuration in isolation, we trace the sequence of failures that produced it, since each step isolates one cause of cross-tool failure (Table 3).

Table 3 Diagnostic evolution on the C1+C4 $\rightarrow$ C6 split. ^eExploratory hyperparameter search (Section 6.2); not used for model comparison.

Stage	Normalization	Target	Features	R^2	RMSE (μm)
1	static z-score	VB	pooled top-50	-0.616	59.38
2	initial baseline	VB	pooled top-50	-2.354	85.54
3	initial baseline	ΔVB	pooled top-50	+0.589	29.95
4	initial baseline	ΔVB	intersection ($T=75, 37$)	0.8935	15.25
5	initial baseline	ΔVB	physical groups (10)	0.1083	44.11
Optimized ^e	initial baseline	ΔVB	intersection ($T=60, 21$)	0.9398	11.46

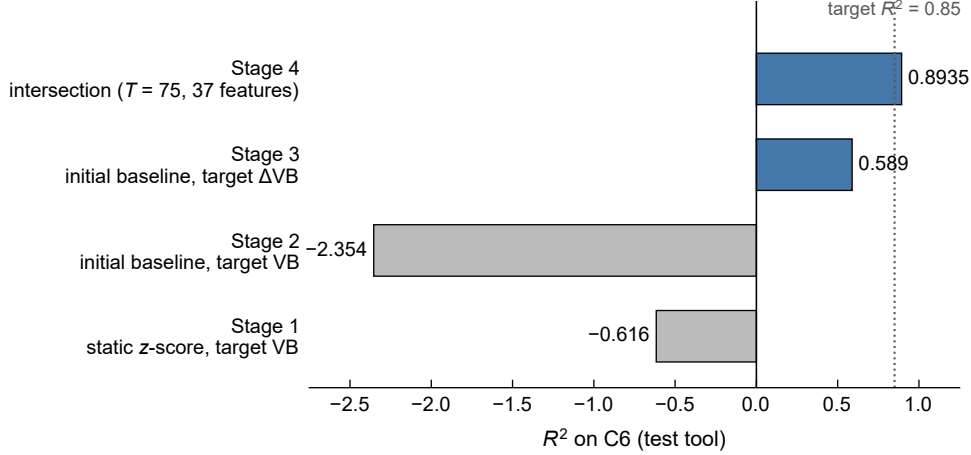


Fig. 4 R^2 progression on the C1+C4 $\rightarrow$ C6 split across the diagnostic stages of Table 3 (Stage 5, a failed side branch, and the exploratory configuration of Section 6.2 are omitted here).

The static baseline (stage 1) fails outright: C6’s initial wear and signal level differ from the training tools, so statically normalized features carry a systematic offset and all models (PIN, linear regression, and SVR alike) score negative R^2 . Switching to initial-baseline normalization alone (stage 2) makes things *worse* ($R^2 = -2.354$): with the signal offset removed, the target offset is exposed, and predicting absolute VB forces the polynomial to extrapolate beyond the training range. Only when both corrections are applied together, baseline normalization *and* the ΔVB target (stage 3), does R^2 turn positive (0.589). The two distribution shifts must be treated jointly; fixing either one alone is not merely insufficient but can be counterproductive.

Stage 3’s residual gap traces to feature screening: pooled Pearson ranking admits features that rank highly on one training tool only. Replacing it with intersection screening (stage 4; $T = 75, 37$ features, three layers) lifts R^2 to 0.8935. Stage 5 probes the opposite strategy (one representative feature per physical mechanism group, 10 in total) and fails ($R^2 = 0.1083$): all ten representatives are amplitude-type statistics tracking the same wear trend, with pairwise $|r| > 0.927$, so the design matrix is nearly rank-one and OLS fitting degrades. High wear correlation is necessary but not sufficient; the retained set also needs enough statistical diversity to keep the covariance matrix well conditioned. Across all diagnostic stages on this split, the L1 seed is **Ax_ptp**,

Table 4 Layer structure of the exploratorily selected model (CPM=0.5, CL=0.75, $W_{\text{INIT}}=10$, $T=60$).

Layer	Node type	Inputs	Coefficients
L2	Triple	Ax_std , Ax_wpd_e12 , Ay_spec_mean	10
L3 (output)	Triple	Az_wpd_rms5 , L2 output, Vy_std	9
Total (final model)			19

The layer-1 winner **L1.S.Ax.ptp** (cubic, 4 coefficients, PSE 94.40) guides the induction but is not part of the final path; the full explicit model with all coefficients and scaling constants is given in Supplementary Material S2.

the feed-direction cutting-force peak-to-peak value, a first-order interpretability check that persists under every feature-engineering change (across all six main experimental settings, including the other LOTO folds, it is selected in five; Section 7).

6.2 Exploratory hyperparameter search

Starting from the stage-4 configuration, a staged full-factorial search (56 runs) optimizes the hyperparameters: Stage A sweeps $\text{CPM} \times \text{CL}$ (5×5 , giving $\text{CPM}^* = 0.5$, $\text{CL}^* = 0.75$, $R^2 = 0.9151$); Stage B sweeps $W_{\text{INIT}} \times T$ (4×7 , giving $W_{\text{INIT}}^* = 10$, $T^* = 60$, $R^2 = 0.9398$); Stage C confirms depth 3. The two stages contribute +0.022 and +0.025 in R^2 respectively. Tightening T from 75 to 60 shrinks the intersection from 37 to 21 features, discarding exactly those whose ranks disagree most between C1 and C4, that is, the least cross-tool-stable members of the set.

This configuration is an exploratory fit: the search used the C6 split for evaluation, so its numbers are reported for completeness and for the interpretability reading of Section 7, and are not used in any comparison with the baselines, all of which use the stage-4 settings. The exploratory model reaches $R^2 = 0.9398$, $\text{RMSE} = 11.46 \mu\text{m}$, $\text{MAE} = 9.53 \mu\text{m}$ on C6, a relative error of about 6.9% of C6’s ΔVB range. Excluding the test tool’s calibration window (passes 11–315), $R^2 = 0.9353$ and $\text{RMSE} = 11.59 \mu\text{m}$. Both are post-selection exploratory estimates: excluding the calibration window does not remove the selection bias introduced by evaluating the 56-run search on C6. The final model is a nesting of two Triple nodes over five of the 21 screened features, 19 coefficients in total (Table 4): an inner node combining the feed-direction force standard deviation (**Ax_std**), a feed-direction force wavelet sub-band energy (**Ax_wpd_e12**), and the y -axis force spectral mean (10 coefficients), and an output node combining that intermediate with a z -axis force wavelet RMS and the y -axis vibration standard deviation (9 coefficients). The layer-1 winner, the cubic seed on **Ax_ptp** that anchors the induction, is superseded by richer combinations in deeper layers and does not appear in the final composition; layer winners in GMDH-type induction are not necessarily nested. The full polynomial is compact enough to print, which Section 7 uses for physical interpretation.

Two sensitivity observations matter for practice. First, depth: at stage-4 settings, training PSE keeps falling through depth 4 and 5, but test R^2 collapses from 0.8935 (depth 3) to 0.0931 (depth 4). The PSE minimum at depth 3 correctly identifies the generalization optimum, which is the criterion behaving as designed. Second, the

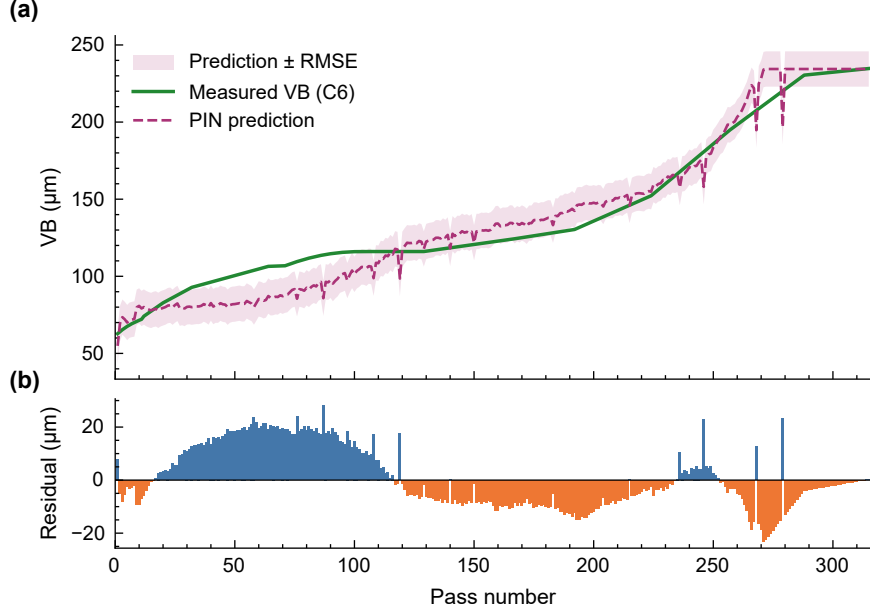


Fig. 5 Exploratory-fit prediction on the C6 test set ($CPM = 0.5$, $CL = 0.75$, $W_{INIT} = 10$, $T = 60$; metrics in Section 6.2; this fit is not used for model comparison). (a) Measured vs. predicted VB with a $\pm RMSE$ band; (b) per-pass residual $y_i - \hat{y}_i$ (blue: underestimate, orange: overestimate). The residual pattern is discussed in Section 7.

Table 5 Complete LOTO cross-validation (stage-4 settings; screening re-run per fold) and baseline models under matched conditions.

Train	Test	Features	R^2	RMSE (μm)
C1+C4	C6	37	0.8935	15.25
C1+C6	C4	43	0.8743	13.66
C4+C6	C1	37	-0.607	35.21
SVR (RBF, tuned; optimal feature set, C6)			0.3566	37.47
Linear regression (same conditions, C6)			-20.98	219.00

baseline window: $W_{INIT} \geq 20$ is broadly unstable, consistent with long calibration windows absorbing early wear trend into the baseline and erasing the signal. An isolated anomaly at ($CPM=0.5$, $CL=0.80$), where R^2 drops to -0.97 while both neighboring CL values are normal, is noted as a sensitivity of the criterion at specific penalty-threshold combinations (Section 7).

6.3 Leave-one-tool-out validation

Most published work on PHM 2010 reports only the $C1+C4 \rightarrow C6$ split. We report all three folds, each with independent intersection screening at the stage-4 settings (Table 5).

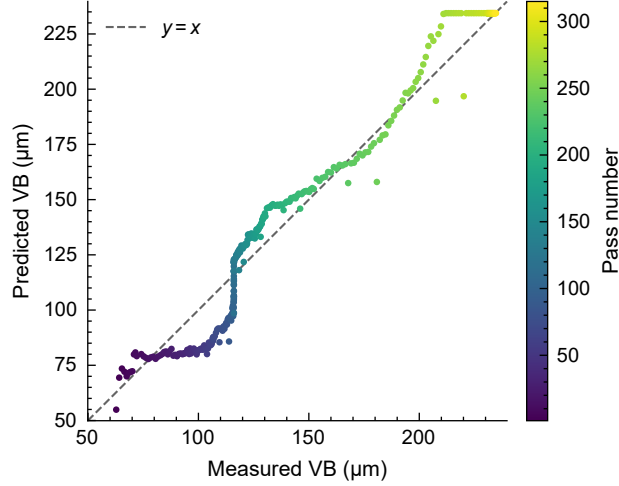


Fig. 6 Predicted vs. measured VB for all 315 C6 passes, colored by pass number; the dashed line marks perfect prediction ($y = x$).

Table 6 Screening-statistic ablation: PIN R^2 per LOTO fold when the per-tool ranking statistic of IFS is replaced, with everything else unchanged (stage-4 settings, $T = 75$). Intersection size in parentheses. MIC ties at 1.0 are broken at random; three seeds are shown. Criterion defined before execution (Supplementary Material S7): a statistic replaces Pearson if it exceeds it by at least 0.05 on at least two folds; none does. Full results, including RMSE and the calibration-window-excluded R^2 , are in Supplementary Material S5.

Ranking statistic	C1+C4→C6	C1+C6→C4	C4+C6→C1
Pearson $ r $ (used)	0.893 (37)	0.874 (43)	-0.607 (37)
Spearman $ \rho $	0.850 (58)	0.897 (43)	-0.219 (47)
Mutual information	0.560 (54)	0.928 (53)	-3.116 (52)
MIC, tie-break seed 0	0.700 (29)	0.797 (25)	-0.503 (29)
MIC, tie-break seed 1	0.906 (31)	0.807 (28)	0.362 (28)
MIC, tie-break seed 2	0.832 (30)	0.775 (24)	-1.030 (30)

Two folds generalize ($R^2 > 0.87$); one fails. The failure is specific to PIN rather than to the tool: under the identical pipeline the no-signal template reaches 0.410 on C1, a validation-free Lasso 0.488, and gradient boosting 0.720 (Table 7), so the wear of C1 is learnable from C4 and C6. C1 does differ from the other tools (a steep rise over the first ~ 25 passes, a long plateau, then late acceleration, with a total increment of $115.95 \mu\text{m}$ that is 69% of theirs), but the PIN error is not concentrated where the curve shape differs: 93% of the squared error falls after pass 200, and Section 7.3 examines a hypothesis consistent with it: that the polynomial’s interaction terms are being evaluated outside the joint range of the training features. The fold is also sensitive to the screening statistic (Table 6): replacing Pearson’s $|r|$ moves PIN’s C1 result between -3.1 and $+0.36$, while the other two folds stay within 0.56 – 0.93 . We regard reporting and analyzing this failure as part of the validation, not a caveat to it.

Table 6 answers whether a nonlinear ranking statistic would improve the intersection. Spearman’s ρ and mutual information each win one fold and lose the others; mutual information drops the primary fold to 0.560. MIC saturates: 166, 107, and 177 of the 306 features reach $\text{MIC} \geq 0.999$ on C1, C4, and C6 respectively, so its top-75 set is a random draw among tied features, and the three seeds span R^2 from -1.03 to $+0.36$ on C1. The spread across statistics is itself informative about the C1 fold, which responds to the feature set far more than the other two folds do.

Selection-bias disclosure.

The hyperparameter search of Section 6.2 used the C6 split for evaluation, and C6 is also the reported test set, so $R^2 = 0.9398$ carries selection bias. With only three labeled tools this protocol is common in the literature, but we state it explicitly and offer the C4 fold ($R^2 = 0.8743$), whose hyperparameters were *not* tuned for that tool, as the conservative estimate of cross-tool performance.

6.4 Baselines and negative results

Under matched conditions (optimal feature set and normalization), tuned SVR reaches $R^2 = 0.3566$ (RMSE $37.47 \mu\text{m}$, MAE $28.80 \mu\text{m}$) and linear regression fails at -20.98 (RMSE $219.00 \mu\text{m}$, MAE $145.34 \mu\text{m}$; Table 5). The SVR was tuned by grid search over (C, γ, ε) with 5-fold cross-validation on the training tools (best: $C=10$, $\gamma=0.1$, $\varepsilon=0.5$; CV $R^2 = 0.573$); the linear model is ordinary least squares with no regularization. The intersection-screened feature space is highly collinear (mean pairwise $|r| > 0.92$), which destabilizes plain OLS, while PIN’s layered induction with PSE-based stopping navigates it.

To rule out the possibility that the linear failure merely reflects missing regularization, and to answer whether the screening or the tuning protocol disadvantages conventional learners, we run ridge, elastic-net or Lasso, and gradient-boosted-tree baselines under the identical pipeline on all three LOTO folds in a 2×2 design: the features are either the intersection-screened set of each fold or all 306, and the hyperparameters are chosen either by two-fold leave-one-training-tool-out cross-validation (so no test-tool information enters the choice) or by a rule that needs no held-out data: generalized cross-validation for ridge, the universal threshold for the Lasso (Section 6.5), and library defaults for boosting. Table 7 reports the results together with the two no-signal references defined in Section 6.6. All stage-4 baselines of Table 7 are produced by one script under one protocol; the SVR and OLS results of Table 5 were obtained separately at the optimal 21-feature setting.

Three results follow. First, screening does not handicap the conventional learners: with cross-validated tuning, the all-feature variant beats the screened one by at least 0.05 on only one of the three folds for each of the three models. Second, the tuning protocol matters more than the feature set: the validation-free Lasso beats the cross-validated elastic net by at least 0.05 (the margin fixed in advance) on two of three folds with either feature set, and does so with five to six non-zero coefficients on the screened features. Third, and most important for how PIN should be read, on the primary fold a five-coefficient Lasso (0.895) and an untuned gradient-boosting model

Table 7 Reference baselines under the identical pipeline (stage-4 settings, Δ VB space, R^2 per LOTO fold, all 315 passes). Columns: test tool of each fold. “Screened” uses the intersection-selected features of the fold (37, 43, 37); “All” uses the 306 features. Grouped CV = two-fold leave-one-training-tool-out; GCV = generalized cross-validation; UT = universal threshold; the no-signal baselines use pass index only.

Model	Tuning	Screened			All 306		
		C6	C4	C1	C6	C4	C1
PIN	PSE	0.894	0.874	−0.607	—	—	—
No-signal template	—	0.785	0.938	0.410	—	—	—
No-signal cubic	—	0.780	0.949	0.433	—	—	—
Ridge	grouped CV	0.032	0.839	0.315	−149.5	0.804	0.601
Ridge	GCV	−16.1	0.777	−1414	−145.3	0.717	0.817
Elastic net	grouped CV	0.701	0.813	0.304	−2.28	0.808	0.653
Lasso	UT	0.895	0.827	0.488	0.805	0.784	0.728
Grad. boosting	grouped CV	0.758	0.828	0.720	0.882	0.803	0.509
Grad. boosting	defaults	0.682	0.850	0.712	0.897	0.810	0.472

on all 306 features (0.897) match PIN (0.894); on C4, PIN is the highest of the sensor-based learners (0.874), although the no-signal references exceed it there (0.938–0.949, Section 6.6); and on C1 every conventional baseline except the degenerate ridge–GCV fit on the screened features exceeds PIN. We therefore claim no accuracy advantage for PIN over conventional learners on this benchmark. Ridge with GCV on the screened features is a cautionary case rather than a bug: GCV selects an almost unregularized penalty on 630 highly collinear samples and the fit collapses (−16.1 and −1414); we report it rather than tune it away.

We also record a negative extension: appending 23 cross-channel ratio features (306 $\rightarrow$ 329 dimensions) and re-running the identical pipeline collapsed R^2 to 0.2985, as a single admitted ratio feature displaced the L1 seed and re-routed the induction path; the extension was discarded. No results from other datasets are reported in this work.

6.5 Model selection with two training tools

Claim 2 of Section 1 concerns the protocol by which model complexity is chosen, not the model. With two training tools a tool-grouped cross-validation has exactly two folds, each of which trains on a single tool. We test what this does to a model family that is directly comparable to PIN: penalized regression on the degree-2 polynomial expansion of the screened features (37, 43, and 37 features expand to 740, 989, and 740 terms; $n = 630$), fitted in three ways: Lasso and elastic net with the penalty chosen by two-fold grouped cross-validation over a 25-point grid, and Lasso with the penalty fixed by the universal threshold $\alpha = \hat{\sigma}\sqrt{2\ln p/n}$ [44], where $\hat{\sigma}$ is the residual standard deviation of a ridge fit on the training data. Table 8 reports all three folds.

On the primary fold, cross-validation selects $\alpha \approx 0.027$ for the Lasso and 0.003 for the elastic net (38 and 469 non-zero terms), and both fits collapse on C6 ($R^2 = -27.2$ and -46.4); the universal threshold selects five terms and reaches 0.885. On C4 the protocols agree (0.828–0.853), and on C1 the validation-free Lasso is again the better

Table 8 Model selection with two training tools: R^2 per LOTO fold (all 315 passes) for penalized regression on the degree-2 expansion of the screened features, with the penalty chosen by grouped cross-validation or by a validation-free rule, alongside PIN (PSE) and the no-signal template. Columns C6, C4, C1: test tool of the fold. Last column: non-zero terms on the C6 / C4 / C1 folds. Selected penalties, noise estimates, and the tuning grids of all baselines are in Supplementary Material S6.

Model	Penalty chosen by	C6	C4	C1	Terms
Lasso, degree 2	2-fold grouped CV	-27.2	0.853	0.379	38 / 34 / 5
Elastic net, degree 2	2-fold grouped CV	-46.4	0.853	-1.73	469 / 34 / 315
Lasso, degree 2	universal threshold	0.885	0.828	0.426	5 / 10 / 5
Ridge, linear ($\alpha = 1$)	fixed	-6.44	0.820	-35.7	37 / 43 / 37
PIN	PSE	0.894	0.874	-0.607	—
No-signal template	—	0.785	0.938	0.410	0

of the two Lasso fits (0.426 vs. 0.379). The two-fold estimate misleads not because cross-validation is wrong in principle but because each fold trains on one tool: the tuning objective measures transfer from one tool to one other tool, which is exactly the quantity that varies most between tools (Section 5), and small-sample cross-validation carries large error bars in any case [42]. The conclusion we draw is deliberately narrow. Within this controlled comparison of degree-2 Lasso fits, the universal threshold never produced a collapse, whereas grouped cross-validation did on the primary fold; on the other two folds the two rules are within 0.05 of each other. PSE, which selects the PIN network on the same 630 samples without any split, yields 0.894 on the primary fold and -0.607 on C1; because PIN is a different model family, its results are reported alongside as those of an auditable nonlinear candidate, not as further evidence about the selection rule. A five-term Lasso chosen by a fixed rule is as accurate as the polynomial network on the primary fold, and Table 7 shows the same for an untuned tree ensemble.

6.6 Where sensor-based prediction pays: no-signal baselines

Under strictly fixed cutting conditions, wear is strongly monotone in pass count, so a baseline that ignores the sensors entirely is a mandatory control. We evaluate two: a *mean wear-curve template* that predicts, at each pass, the average of the training tools’ ΔVB values at that pass, and a cubic polynomial in pass index fitted to the pooled training curves. Table 7 shows both are strong on average, stronger than every conventional learner on two of the three folds, which calibrates how little of a full-history R^2 on this benchmark is attributable to sensing per se.

Full-history R^2 , however, hides where the approaches differ. Table 9 breaks the primary fold’s error into wear stages: the template is more accurate over the first ~ 150 passes, where every tool tracks the population mean, but its error grows to $\sim 41 \mu\text{m}$ RMSE in the accelerated-wear stage, because C6’s late-life wear exceeds the training tools’ average and a pass-index replay cannot notice. The sensor-based PIN holds 9–10 μm in exactly this stage, the region where tool-change decisions are made and under-prediction is most costly. (The tuned configuration behaves the same: 9.97 vs. 37.87 μm .) The same analysis on the C4 fold gives the opposite outcome (template

Table 9 RMSE (μm) by wear stage on the primary fold (C1+C4→C6, stage-4 settings, calibration window excluded).

Stage (passes)	PIN	No-signal template
15–50 (early)	10.94	4.65
50–150 (steady, first half)	22.11	7.01
150–250 (steady, second half)	11.78	18.16
250–315 (accelerated)	9.15	40.94
All out-of-sample (15–315)	15.54	22.18

11.8 vs. PIN 23.9 μm in the accelerated stage): C4’s trajectory happens to lie close to the mean of C1 and C6, so the template wins by geometric coincidence. The distinction matters for deployment: a template’s accuracy depends on how closely the next tool happens to track the historical mean, a condition unobservable in advance, whereas a sensor-based model reads the individual tool’s actual state. With three labeled tools we cannot quantify how often each case arises in a production population; we report both.

7 Discussion

7.1 Reading the induced polynomial

By interpretability we mean that the complete fitted model can be inspected, recomputed by hand from its printed coefficients, and traced back to the sensor features it uses; we do not claim that every term carries its own physical meaning. Within this scope we distinguish two levels of interpretability. At the *first order*, the L1 seed feature is directly readable against cutting mechanics, and we treat this as a genuine explanatory claim. At *higher order*, the L2/L3 Triple nodes are fully visible polynomials, but attributing a specific physical coupling to a given cross term would require controlled experiments (e.g., modal analysis across cutting speeds) that this study does not run; we report these as observations, not confirmed mechanisms.

At the first order, the L1 seed is `Ax_ptp` in five of six main experimental settings, the one exception being a run where the training pair’s dominant axis shifted; this is the peak-to-peak cutting force in the feed direction, and its dominance is consistent with the feed direction carrying the largest dynamic cutting-force component as the edge dulls (Section 3). We anchor this first-order claim on the layer-1 selection itself, the strongest *single* feature the induction finds, since the final composition supersedes it: the deployed model consists of two nested Triple nodes over five features, four from the force channels (`Ax_std`, `Ax_wpd_e12`, `Ay_spec_mean`, `Az_wpd_rms5`) and one from vibration (`Vy_std`), with `Ax_ptp` absent from the final path (Section 6.2). Feed-direction force features still dominate the composition. That both retained nodes resolve to three-variable rather than two-variable forms indicates the wear-related information is not captured by any single-channel or pairwise relationship alone. We note this as an observation about model structure; we do not decompose the specific cross-terms into individual physical couplings.

Reading `Ax.ptp`, the directly measured feed-direction cutting force, as an indicator of the normal contact force in Archard’s relation is a qualitative, not calibrated, correspondence: the dynamometer records the total feed-direction force, of which the wear-driven normal component is one contributor, and edge geometry and spindle dynamics also shape it; these factors are held constant under the fixed cutting conditions used here, which is what licenses the interpretation in this study without asserting it more generally.

7.2 Observed sensitivities

A handful of empirical patterns are worth recording as observations, without claiming a complete mechanistic account. The optimal baseline window ($W_{\text{INIT}} = 10$) is small enough to sit inside the break-in region rather than straddle it; windows that straddle the break-in transition (e.g., $W_{\text{INIT}} = 20$) show markedly degraded and unstable performance, consistent with the baseline mean mixing break-in and steady-state signal character. The complexity penalty and carving threshold interact non-monotonically at one point in the search grid (an isolated collapse to $R^2 \approx -0.97$ at one combination, with both neighboring settings normal), which we attribute to the discrete, threshold-driven nature of coefficient carving rather than to any flaw specific to this dataset. Reducing ten highly wear-correlated features to a physically motivated representative set (Section 6.1) failed because those features’ pairwise correlations exceed 0.9 and the resulting design matrix has an effective rank close to one; high correlation with the target does not imply the statistical diversity a stable fit needs.

7.3 The C1 boundary

The one LOTO failure (Section 6.3) is a useful negative result rather than an anomaly to be explained away, and because the model is explicit, a mechanism hypothesis can be examined in the fitted terms. Three candidate explanations were tested on the stage-4 network of the $C4+C6 \rightarrow C1$ fold, retrained once so that its nodes could be instrumented; the criteria were defined before execution (the analysis plan is reproduced in Supplementary Material S7), and the details are given in Supplementary Material S4.

Curve shape. C1’s wear curve does differ from C4’s and C6’s (a steep rise over roughly the first 25 passes, a long near-flat plateau, then late acceleration), but the error does not follow that shape. Over passes 16–200 the RMSE of the PIN prediction is 6–19 μm and these passes contribute 6% of the squared error, whereas passes 201–250 alone contribute 59% (RMSE 68 μm , mean overestimate 39 μm) and passes 251–315 a further 34%. A shape mismatch in the break-in region cannot explain an error that appears 180 passes later.

Single-input effects. Three of the four inputs on the network’s path respond 1.3–2.6 times more steeply to wear on C1 than on the training tools (the fourth, slightly less), so one might expect the model to read C1’s signals as more wear than has occurred. Rescaling each input by the ratio of training to C1 slopes, an oracle correction that uses C1’s labels and is not deployable, does not help ($R^2 = -0.74$), and neither does removing the input clamp (−1.21): the clamp, which limits each input to its training range before the polynomial is evaluated, protects the model rather than causing the

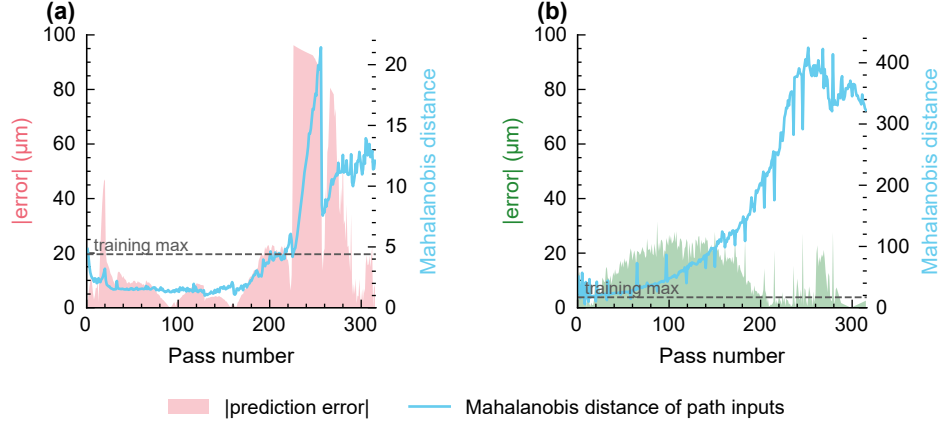


Fig. 7 Absolute prediction error (shaded, left axis) and Mahalanobis distance of the network’s path inputs from the training tools (line, right axis; dashed: maximum over the training passes) for (a) the failed fold C4+C6→C1 and (b) the primary fold C1+C4→C6, stage-4 settings. On C1 the error rises where the input combination leaves the training range; on C6 the distance exceeds the training maximum on almost every pass, yet the prediction is accurate, so distance alone does not predict failure.

failure. PIN’s polynomial therefore does not extrapolate freely: inputs and output are clipped to their training ranges by design, and on this fold 36 of the 315 predictions sit at the output clamp.

Joint-range extrapolation. What changes after pass 200 is the *combination* of inputs. The Mahalanobis distance of the four-input vector from the training tools’ distribution stays below the training maximum on every pass up to 200, exceeds it on 74% of passes 201–250 and on all of passes 251–315, and correlates with the absolute error (Spearman $\rho = 0.69$ over passes 16–315; Fig. 7a). The leverage of each pass in the term space of the two Triple nodes tells the same story ($\rho = 0.70$ and 0.80). Individually in range, the inputs jointly reach a region the training data never covered; the interaction terms of the Triple nodes are evaluated there, and the prediction saturates at the output clamp ($\Delta VB = 164.6 \mu\text{m}$, the training maximum at $W_{\text{INIT}} = 15$; Table 2 uses $W_{\text{INIT}} = 10$) while the measured increment is 74–110 μm .

We state this as a correlation within the failed fold, not as a sufficient condition for failure: on the primary fold the C6 inputs also lie outside the training range by the same measure on 99% of passes (the C6 network uses 37 inputs, in which space a new tool is almost always far from the training cloud), yet the prediction is accurate (Fig. 7b). Distance from the training support therefore does not by itself predict failure, and a label-free alarm built on it did not meet the false-alarm rate we had set in advance (Supplementary Material S4). What the analysis does show is that PIN’s error on C1 is concentrated where its inputs jointly leave the training range, which is consistent with the extrapolation hypothesis without establishing it as a sufficient criterion. On this fold, gradient boosting and the five-term Lasso remain at 0.49–0.72 (Table 7); that difference is compatible with their not depending on high-order interactions of inputs that have left the training region together, but the present study does not establish this as the cause. The practical boundary is therefore a joint property of the

model class and the data: a polynomial with interaction terms is safest where the joint distribution of its inputs was sampled during training, and with two training tools that region is narrower than the range of any single input suggests.

7.4 Limitations

The training set is 630 samples from two tools, small relative to typical deep-learning benchmarks; all reported numbers are single-fold estimates, since three labeled tools do not support a confidence interval on cross-tool generalization. All experiments use one machine, one tool geometry, one workpiece material, and fixed cutting parameters. Transfer across cutting conditions is excluded by design rather than overlooked: replacing a worn tool within an unchanged process is a practically common deployment case, the one targeted here, and it is the case in which the shared-mechanism assumption of Section 1 is defensible. A joint cross-condition, cross-tool study would require wear-labeled data over several cutting conditions and is left to future work. This study uses the force and vibration channels but not the acoustic-emission signal that a fuller sensor suite might exploit. PIN has not been shown to be more accurate than sparse linear or tree-ensemble learners under validation-free tuning (Section 6.4); its case rests on an auditable nonlinear structure and on validation-free structure selection, not on accuracy. The five-coefficient Lasso is at least as auditable and did not collapse on C1, so PIN is presented here as a feasible candidate among printable nonlinear models, not as the model these results show should be preferred. Finally, the method has no built-in mechanism to signal, without labels, that a new tool has entered a region where its prediction should not be trusted: the distance-based alarm tested in Supplementary Material S4 was accurate on the failed fold but raised alarms on almost every pass of a fold on which the prediction was correct, so a usable version remains future work.

A property of the benchmark itself further bounds how the absolute errors should be read. The wear annotations released with PHM 2010 are a smoothed series rather than raw per-pass microscope readings: across the nine per-flute sequences of the three labeled tools, none of the 2,826 successive transitions decreases, and the residual of a seven-point moving quadratic fit has a standard deviation of $0.003\text{--}0.24\,\mu\text{m}$, that is, $0.002\text{--}0.1\%$ of the wear range and two to three orders of magnitude below the repeatability of optical flank-wear measurement; two of the three flutes of C6 are reproduced exactly by a piecewise-linear function with fifteen knots over 315 passes. Checking against an independent public compilation of the same dataset confirms that this smoothing is present in the released annotations and is not introduced by our preprocessing. Because every study built on this benchmark inherits the same targets, comparisons between methods are unaffected; the absolute errors reported here, however, are measured against a smoothed target and would likely be larger against raw per-pass readings.

8 Conclusion

We studied cross-tool flank-wear prediction in the case where the process is fixed and only a few labeled tool histories exist, and organized the work around three claims.

First, the tool-to-tool differences that defeat a model trained on other tools can be treated as offsets and removed in the data. Initial-baseline normalization of the features, a relative wear target ΔVB , and intersection feature selection raised R^2 on the primary PHM 2010 fold from -0.616 under static normalization to 0.8935 , and a newly mounted tool needs only a calibration window of signal statistics and one initial wear measurement, with no adaptation training on that tool.

Second, with two training tools the choice of model complexity is the binding constraint. In a controlled comparison within one model family, tool-grouped cross-validation over its two possible folds selected a degree-2 Lasso with 38 terms (and an elastic net with 469) that reached $R^2 = -27$ and -46 on the held-out tool, while the universal threshold, which uses no held-out data, selected a five-term Lasso that reached 0.885 on the same fold; on the other two folds the two rules were within 0.05 of each other. This is a finding on one benchmark, not a general claim about cross-validation. The polynomial network selected by PSE (0.894 on that fold) is reported alongside as an auditable nonlinear candidate.

Third, the polynomial induction network is an auditable model, not a more accurate one. Under complete leave-one-tool-out validation it reached 0.8935 and 0.8743 on two folds, and a five-coefficient Lasso and an untuned gradient-boosting model matched it on the primary fold. Its explicit structure made a hypothesis for the third fold’s failure (-0.607) inspectable: the error is concentrated not where that tool’s wear curve differs in shape, but where the combination of its inputs leaves the region the training tools covered, consistent with extrapolation of the interaction terms; the same departure occurs on the primary fold without failure, so it is not a sufficient criterion. On that fold the bounded learners kept smaller errors, a difference compatible with, but not shown to be caused by, their lack of high-order interaction terms. The no-signal baseline calibrates all of this: replaying the mean training wear curve is competitive on average, but its error grows to $38\text{--}41\ \mu\text{m}$ RMSE in the accelerated-wear stage of a tool that wears faster than the training mean, where the sensor-based model holds $9\text{--}10\ \mu\text{m}$, the regime in which tool-change decisions carry the most risk.

The limitations (one dataset, three labeled tools, fixed cutting conditions, no acoustic-emission channel, and a smoothed wear label) define the next steps: larger multi-tool benchmarks, a joint cross-condition and cross-tool study, a reliable label-free indicator of when a tool has left the training support (the distance-based alarm tested here was not one), and prediction uncertainty rather than point estimates. No results from datasets other than PHM 2010 are reported in this work.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used Claude (Anthropic) to assist in expediting the drafting of manuscript text, organizing content across sections, and refining the manuscript’s language and structure, based on experimental results, analysis, and content decisions made by the author(s). After using this tool, the author(s) reviewed and edited all content as needed and take full responsibility for the accuracy and integrity of the content of the published article.

Statements and Declarations

Funding Not applicable.

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Availability of data and materials The PHM 2010 dataset analyzed in this study is publicly available from the PHM Society (https://phmsociety.org/phm_competition/2010-phm-society-conference-data-challenge/) and via IEEE DataPort (<https://doi.org/10.21227/jdxd-yy51>). The per-pass feature matrices of the three labeled tools (315×306 each, with the wear labels), the screened feature lists of the three folds, and a script that regenerates every conventional-baseline row of Tables 7 and 8 and checks them against the printed values are provided as Electronic Supplementary Material 2, so that the baseline comparison can be rerun with open-source software alone.

Code availability The Polynomial Induction Network implementation is proprietary research software and is not publicly released; its inputs, settings, and outputs are documented in Supplementary Material S6 so that its results can be audited. The analysis script for the conventional baselines is released with the data (Electronic Supplementary Material 2).

Authors' contributions All authors contributed to the study conception and design. Experimental design and methodology development were performed by Jenn-Yih Chen and Yu-Yuan Ting. Data analysis and software implementation were performed by Yu-Yuan Ting. The first draft of the manuscript was written by Yu-Yuan Ting, and all authors discussed the methodology and commented on previous versions of the manuscript. Yung-Cheng Wang and Bean-Yin Lee supervised the study. All authors read and approved the final manuscript.

References

- [1] Martinez-Arellano G, Terrazas G, Ratchev S (2019) Tool wear classification using time series imaging and deep learning. The International Journal of Advanced Manufacturing Technology 104:3647–3662. <https://doi.org/10.1007/s00170-019-04090-6>
- [2] Serin G, Sener B, Ozbayoglu AM, et al (2020) Review of tool condition monitoring in machining and opportunities for deep learning. The International Journal of Advanced Manufacturing Technology 109:953–974. <https://doi.org/10.1007/s00170-020-05449-w>
- [3] Teti R, Jemielniak K, O'Donnell G, et al (2010) Advanced monitoring of machining operations. CIRP Annals 59(2):717–739. <https://doi.org/10.1016/j.cirp.2010.05.010>
- [4] Li R, He Z, Zhao M (2023) Convlstm-att: An attention-based composite deep neural network for tool wear prediction. Machines 11(2):297. <https://doi.org/10.3390/machines11020297>

- [5] Xu X, Wang J, Zhong B, et al (2021) Deep learning-based tool wear prediction and its application for machining process using multi-scale feature fusion and channel attention mechanism. *Measurement* 177:109254. <https://doi.org/10.1016/j.measurement.2021.109254>
- [6] Usui E, Shirakashi T, Kitagawa T (1984) Analytical prediction of cutting tool wear. *Wear* 100(1–3):129–151. [https://doi.org/10.1016/0043-1648\(84\)90010-3](https://doi.org/10.1016/0043-1648(84)90010-3)
- [7] Huang Z, Shao J, Zhu J, et al (2024) Tool wear condition monitoring across machining processes based on feature transfer by deep adversarial domain confusion network. *Journal of Intelligent Manufacturing* 35(3):1079–1105. <https://doi.org/10.1007/s10845-023-02088-2>
- [8] He J, Sun Y, Yin C, et al (2023) Cross-domain adaptation network based on attention mechanism for tool wear prediction. *Journal of Intelligent Manufacturing* 34(8):3365–3387. <https://doi.org/10.1007/s10845-022-02005-z>
- [9] Barron AR (1984) Predicted squared error: A criterion for automatic model selection. In: Farlow SJ (ed) *Self-Organizing Methods in Modeling: GMDH Type Algorithms*. Marcel Dekker, New York, p 87–103
- [10] Li K, Chen M, Lin Y, et al (2022) A novel adversarial domain adaptation transfer learning method for tool wear state prediction. *Knowledge-Based Systems* 254:109537. <https://doi.org/10.1016/j.knosys.2022.109537>
- [11] Liu C, Wang Y, Chen Z (2022) Three-stage wiener-process-based model for remaining useful life prediction of a cutting tool. *Sensors* 22(13):4763. <https://doi.org/10.3390/s22134763>
- [12] Sun H, Pan J, Zhang J, et al (2020) Non-linear wiener process-based cutting tool remaining useful life prediction considering measurement variability. *The International Journal of Advanced Manufacturing Technology* 107(11–12):4493–4502. <https://doi.org/10.1007/s00170-020-05264-3>
- [13] Yang Q, Mishra D, Awasthi U, et al (2024) Tool wear and remaining useful life estimation in precision machining using interacting multiple model. *Journal of Manufacturing Systems* 74:367–386. <https://doi.org/10.1016/j.jmsy.2024.04.001>
- [14] He J, Meng X, Guo J, et al (2026) Cross-condition tool wear prediction via feature decoupling and domain-adaptive representation learning. *Mechanical Systems and Signal Processing* 259:114783. <https://doi.org/10.1016/j.ymssp.2026.114783>
- [15] Cheng M, Liang G, Bai X, et al (2026) Rational invariance constrained weighted adversarial domain generalization model for tool wear monitoring under unseen machining conditions. *Measurement Science and Technology* 37(29):296003.

<https://doi.org/10.1088/1361-6501/ae8928>

- [16] Zhou J, Wang L, Wang T (2026) Tool wear prediction under varying cutting conditions: A few-shot warm-start framework based on model-agnostic meta-learning. *Machines* 14(5):471. <https://doi.org/10.3390/machines14050471>
- [17] Redjaimia O, Habbouche H, Benkedjouh T, et al (2026) Wavelet-based multi-branch deep learning for tool wear monitoring with physics-informed label construction. *The International Journal of Advanced Manufacturing Technology* 143(1-2):751–772. <https://doi.org/10.1007/s00170-026-17636-2>
- [18] Lim SH, Kim JM (2026) A self-referencing framework for milling tool wear diagnosis under tool-to-tool variability using physics-informed order-tracked features. *Machines* 14(9):966. <https://doi.org/10.3390/machines14090966>
- [19] Chen X, Cheng K (2025) Cutting tool remaining useful life prediction using multi-sensor data fusion through graph neural networks and transformers. *Machines* 13(11):1027. <https://doi.org/10.3390/machines13111027>
- [20] Pashmforoush F, Ebrahimi Araghizad A, Budak E (2025) Tool wear prediction in milling process using physics-informed machine learning and thermo-mechanical force model with monitoring applications. *Journal of Manufacturing Systems* 82:1192–1212. <https://doi.org/10.1016/j.jmsy.2025.08.014>
- [21] Mao Z, Luo M, Zhang D (2025) Tool wear prediction through a physics-assisted method using few-sensor monitoring data. *The International Journal of Advanced Manufacturing Technology* 139(3-4):1645–1660. <https://doi.org/10.1007/s00170-025-15965-2>
- [22] Han S, Awasthi U, Bolas GM (2025) Physics-informed symbolic regression for tool wear and remaining useful life predictions in manufacturing. *Journal of Manufacturing Systems* 80:734–748. <https://doi.org/10.1016/j.jmsy.2025.03.023>
- [23] Zhang H, Yang W, Xu W, et al (2026) Tool wear prediction with few samples based on stacked BiLSTM and parallel residuals with multi-head self-attention. *Measurement* 258:119078. <https://doi.org/10.1016/j.measurement.2025.119078>
- [24] Liu Z, Wang Y, Vaidya S, et al (2024) KAN: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756* <https://doi.org/10.48550/arXiv.2404.19756>
- [25] Bao S, Chao B, Zhang C, et al (2025) Real-time tool wear monitoring and multi-step forward prediction based on multi-information fusion. *Measurement Science and Technology* 36(2):025104. <https://doi.org/10.1088/1361-6501/ada4cf>
- [26] Yu N, Meng J, Zhang X (2026) Interpretable BiLSTM–KAN modeling of tool wear through multi-source signal fusion. *Wear* 600:206809. <https://doi.org/10.1016/j.wear.2026.206809>

- [27] Ivakhnenko AG (1968) The group method of data handling—a rival of the method of stochastic approximation. *Soviet Automatic Control* 1(3):43–55
- [28] Ivakhnenko AG (1971) Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics* 1(4):364–378. <https://doi.org/10.1109/TSMC.1971.4308320>
- [29] Farlow SJ (ed) (1984) *Self-Organizing Methods in Modeling: GMDH Type Algorithms*. Marcel Dekker, New York
- [30] AbTech Corporation (1990) AIM user’s manual
- [31] Lee BY, Liu HS, Tarng YS (1999) An abductive network for predicting tool life in drilling. *IEEE Transactions on Industry Applications* 35(1):190–195. <https://doi.org/10.1109/28.740864>
- [32] Tsai MK, Lee BY, Yu SF (2005) A predicted modelling of tool life of high-speed milling for skd61 tool steel. *The International Journal of Advanced Manufacturing Technology* 26(7-8):711–717. <https://doi.org/10.1007/s00170-003-1596-z>
- [33] Ivakhnenko AG (1970) Heuristic self-organization in problems of engineering cybernetics. *Automatica* 6(2):207–219. [https://doi.org/10.1016/0005-1098\(70\)90092-0](https://doi.org/10.1016/0005-1098(70)90092-0)
- [34] Mallows CL (1973) Some comments on C_p . *Technometrics* 15(4):661–675. <https://doi.org/10.1080/00401706.1973.10489103>
- [35] Akaike H (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19(6):716–723. <https://doi.org/10.1109/TAC.1974.1100705>
- [36] Schwarz G (1978) Estimating the dimension of a model. *The Annals of Statistics* 6(2):461–464. <https://doi.org/10.1214/aos/1176344136>
- [37] Allen DM (1974) The relationship between variable selection and data augmentation and a method for prediction. *Technometrics* 16(1):125–127. <https://doi.org/10.1080/00401706.1974.10489157>
- [38] Craven P, Wahba G (1979) Smoothing noisy data with spline functions. *Numerische Mathematik* 31(4):377–403. <https://doi.org/10.1007/BF01404567>
- [39] Efron B (2004) The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association* 99(467):619–632. <https://doi.org/10.1198/016214504000000692>
- [40] Stone M (1974) Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)* 36(2):111–133. <https://doi.org/10.1111/j.2517-6161.1974.tb00994.x>

- [41] Arlot S, Celisse A (2010) A survey of cross-validation procedures for model selection. *Statistics Surveys* 4:40–79. <https://doi.org/10.1214/09-SS054>
- [42] Varoquaux G (2018) Cross-validation failure: Small sample sizes lead to large error bars. *NeuroImage* 180:68–77. <https://doi.org/10.1016/j.neuroimage.2017.06.061>
- [43] Varma S, Simon R (2006) Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics* 7:91. <https://doi.org/10.1186/1471-2105-7-91>
- [44] Donoho DL, Johnstone IM (1994) Ideal spatial adaptation by wavelet shrinkage. *Biometrika* 81(3):425–455. <https://doi.org/10.1093/biomet/81.3.425>
- [45] Ye J (1998) On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association* 93(441):120–131. <https://doi.org/10.1080/01621459.1998.10474094>
- [46] Lin Z, Fan Y, Tan J, et al (2025) Tool wear prediction based on XGBoost feature selection combined with PSO-BP network. *Scientific Reports* 15:3096. <https://doi.org/10.1038/s41598-025-85694-9>
- [47] Altintas Y (2012) *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*, 2nd edn. Cambridge University Press, <https://doi.org/10.1017/CBO9780511843723>
- [48] PHM Society (2010) 2010 phm society conference data challenge. IEEE DataPort, <https://doi.org/10.21227/jdxd-yy51>
- [49] Li X, Lim BS, Zhou JH, et al (2009) Fuzzy neural network modelling for tool wear estimation in dry milling operation. In: *Annual Conference of the Prognostics and Health Management Society*, URL <https://papers.phmsociety.org/index.php/phmconf/article/view/1403>
- [50] Severson KA, Attia PM, Jin N, et al (2019) Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy* 4:383–391. <https://doi.org/10.1038/s41560-019-0356-8>

Supplementary Material

S1 Feature definitions S2 Explicit model S3 Search space
S4 The failed fold S5 Screening ablation S6 Model selection and baselines S7
Analysis plan

Each cutting pass yields six mechanical channels: $\mathbf{Ax}$, $\mathbf{Ay}$, $\mathbf{Az}$ (three-axis cutting force, dynamometer) and $\mathbf{Vx}$, $\mathbf{Vy}$, $\mathbf{Vz}$ (three-axis vibration acceleration, accelerometers), sampled at $f_s = 50$ kHz. Per channel, 51 features are extracted in the order listed below (13 time-domain, 8 frequency-domain, 30 time-frequency), and channels are concatenated in the order $\mathbf{Ax} \rightarrow \mathbf{Ay} \rightarrow \mathbf{Az} \rightarrow \mathbf{Vx} \rightarrow \mathbf{Vy} \rightarrow \mathbf{Vz}$, giving $51 \times 6 = 306$ features. Feature names follow the pattern $\{\mathbf{channel}\}_{\mathbf{feature}}$, e.g. $\mathbf{Ax_ptp}$. Signals longer than 65,536 samples are uniformly downsampled before extraction. In the formulas, $x = (x_1, \dots, x_N)$ denotes the channel signal of one pass, $\bar{x}$ its mean, and σ its population standard deviation.

S1.1 Time-domain features (13 per channel)

#	Name	Definition	Note
T1	mean	$\bar{x} = \frac{1}{N} \sum_i x_i$	
T2	std	$\sigma = \sqrt{\frac{1}{N} \sum_i (x_i - \bar{x})^2}$	
T3	var	σ^2	
T4	rms	$\sqrt{\frac{1}{N} \sum_i x_i^2}$	
T5	max	$\max_i x_i$	
T6	min	$\min_i x_i$	
T7	ptp	$\max_i x_i - \min_i x_i$	peak-to-peak
T8	shape_factor	$\text{RMS} / \frac{1}{N} \sum_i x_i $	
T9	crest_factor	$\max_i x_i / \text{RMS}$	
T10	impulse_factor	$\max_i x_i / \frac{1}{N} \sum_i x_i $	
T11	skewness	$\frac{1}{N} \sum_i \left(\frac{x_i - \bar{x}}{\sigma} \right)^3$	
T12	kurtosis	$\frac{1}{N} \sum_i \left(\frac{x_i - \bar{x}}{\sigma} \right)^4 - 3$	excess kurtosis
T13	mean_square	$\frac{1}{N} \sum_i x_i^2 = \text{RMS}^2$	

S1.2 Frequency-domain features (8 per channel)

Power spectral density $P(f_k)$ is estimated by Welch's method ($f_s = 50$ kHz); $p_k = P(f_k) / \sum_k P(f_k)$ denotes the normalized spectrum.

#	Name	Definition	Note
F1	cog_freq	$f_c = \sum_k f_k P(f_k) / \sum_k P(f_k)$	spectral centroid
F2	rmsf	$\sqrt{\sum_k f_k^2 P(f_k) / \sum_k P(f_k)}$	RMS frequency
F3	vf	$\sqrt{\sum_k (f_k - f_c)^2 P(f_k) / \sum_k P(f_k)}$	frequency variance
F4	dom_freq	$\arg \max_{f_k} P(f_k)$	dominant frequency
F5	dom_amp	$\max_k P(f_k)$	dominant amplitude
F6	spec_mean	$\frac{1}{K} \sum_k P(f_k)$	
F7	spec_var	$\frac{1}{K} \sum_k (P(f_k) - \bar{P})^2$	
F8	spec_entropy	$-\sum_k p_k \log_2 p_k$	spectral entropy

S1.3 Time–frequency features (30 per channel)

A level-4 wavelet packet decomposition (WPD) with the **db4** wavelet and periodization boundary mode yields $2^4 = 16$ terminal sub-bands of width ≈ 1.5625 kHz each; $d_{j,k}$ denotes the k -th coefficient of sub-band j , $E_j = \sum_k d_{j,k}^2$ the sub-band energy, and N_j the number of coefficients in sub-band j .

#	Name	Definition	Note
W1–W16	wpd_e0–wpd_e15	$E_j, j = 0, \dots, 15$	16 sub-band energies
W17–W24	wpd_rms0–wpd_rms7	$\sqrt{E_j / N_j}, j = 0, \dots, 7$	RMS of first 8 sub-bands
W25–W28	wpd_kurt0–wpd_kurt3	excess kurtosis of $d_{j,\cdot}, j = 0, \dots, 3$	first 4 sub-bands
W29	wpe_l3	$-\sum_{j=0}^7 e_j \log_2 e_j, e_j = E_j^{(3)} / \sum E^{(3)}$	level-3 packet entropy
W30	wpe_l4	$-\sum_{j=0}^{15} e_j \log_2 e_j, e_j = E_j / \sum E$	level-4 packet entropy

At the 10,400-rpm spindle speed used throughout, the tooth passing frequency is 520 Hz, so each 1.5625-kHz sub-band spans roughly three TPF harmonics.

S2 Full explicit form of the exploratorily selected model

The exploratorily selected configuration (CPM=0.5, CL=0.75, $W_{\text{INIT}}=10$, $T=60$, three layers) was retrained from the cached features and reproduced the reported result exactly ($R^2 = 0.9398$, RMSE = 11.46 μm on C6; layer PSE trace 94.40 $\rightarrow$ 19.52 $\rightarrow$ 14.08); its accuracy is a post-selection exploratory estimate (main text, Section 6.2). The selected model uses five of the 21 screened features and 19 polynomial coefficients in two nested Triple nodes; the layer-1 winner (L1_S_Ax_ptp, cubic, 4 coefficients) guides the induction but is not part of the final path.

S2.1 Input scaling

Each feature is first baseline-normalized as in the main text (Eq. 4): $\tilde{x} = (x - \mu_{\text{init}})/\sigma_{\text{global}}$. The network then applies a per-feature linear rescaling $n_f = b_f + w_f \tilde{x}_f$ with the constants:

Feature f	b_f	w_f
Ax_std	-1.008481	0.999129
Ax_wpd_e12	-0.817185	0.993781
Ay_spec_mean	-1.104847	1.001210
Az_wpd_rms5	-1.513819	0.994826
Vy_std	-0.902566	1.000539

S2.2 Inner Triple node (10 coefficients)

With $u_1 = n_{\text{Ax_std}}$, $u_2 = n_{\text{Ax_wpd_e12}}$, $u_3 = n_{\text{Ay_spec_mean}}$:

$$z = 0.186936 - 1.25321 u_1 + 1.78528 u_3 + 1.50217 u_1^2 + 1.49784 u_3^2 \\ - 4.43522 u_1 u_3 + 0.989862 u_2 u_3 - 0.739038 u_1 u_2 u_3 + 1.00069 u_1^3 + 0.025515 u_2^3$$

S2.3 Output Triple node (9 coefficients)

With $v_1 = n_{\text{Az_wpd_rms5}}$, $v_3 = n_{\text{Vy_std}}$:

$$\widehat{\Delta\text{VB}} = -0.0271356 + 0.175184 v_1 + 0.888641 z - 0.0829042 v_3 - 0.285634 v_1^2 \\ + 0.117118 v_3^2 + 0.244492 v_1 z - 0.101924 v_1 z v_3 + 0.0700933 v_1^3$$

The output node is the identity, so $\widehat{\Delta\text{VB}}$ is in μm directly; absolute wear is recovered as $\widehat{\text{VB}}(t) = \text{VB}_0 + \widehat{\Delta\text{VB}}(t)$.

S3 Hyperparameter search space

Stage	Swept	Grid	Fixed
A	CPM $\times$ CL	$\{0.5, 0.8, 1.0, 1.2, 1.5\} \times \{0.70, 0.75, 0.80, 0.85, 0.90\}$	$T=75$, $W_{\text{INIT}}=15$, depth 3
B	$W_{\text{INIT}} \times T$	$\{10, 15, 20, 25\} \times \{60, 65, 70, 75, 80, 85, 90\}$	CPM=0.5, CL=0.75, depth 3
C	depth	$\{2, 3, 4\}$	all of the above at optimum

Final operating point: CPM=0.5, CL=0.75, $W_{\text{INIT}}=10$, $T=60$, depth 3; L1 survivor count fixed at 10 throughout.

S4 The C4+C6→C1 fold: diagnosis and an out-of-support alarm

All analyses in this section use the stage-4 network of the C4+C6→C1 fold ($W_{\text{INIT}}=15$, $T=75$, 37 screened features, CPM=1.0, CL=0.80, depth 3), retrained once with the same settings; the retrained network reproduces the reported $R^2 = -0.6071$ exactly, and an instrumented forward pass that records every node output reproduces the engine’s predictions to machine precision. The candidate mechanisms and the criteria for each test were written down before the tests were run.

S4.1 Where the error is

Table S4.1 gives the prediction error in ΔVB by pass range ($e = \hat{y} - y$). Passes 201–250 contribute 59.3% of the squared error and passes 251–315 a further 34.1%; passes 16–200, which contain the break-in region in which C1’s wear curve differs most in shape from C4’s and C6’s, contribute 6.2%. The engine clips its output to the training range of ΔVB ($[-17.5, 164.6] \mu\text{m}$); 36 of the 315 predictions sit at the upper clamp, all after pass 200.

Table S4.1: Error of the C1 prediction by pass range.

Passes	Mean e (μm)	RMSE (μm)	Share of SSE	Predictions at output clamp
1–15 (calibration)	+4.0	9.3	0.3%	0
16–50	−4.0	19.1	3.3%	0
51–150	+0.2	6.2	1.0%	0
151–200	−9.1	12.3	1.9%	0
201–250	+38.8	68.1	59.3%	25
251–315	+34.0	45.3	34.1%	11

S4.2 Single-input effects (oracle controls)

The network’s output depends on four inputs (Ax_{rms} , Ax_{var} , $\text{Ay}_{\text{dom.amp}}$, Vz_{ptp}) through two Triple nodes. The slope of each input against ΔVB (ordinary least squares) on C1, divided by the mean slope on C4 and C6, is 1.77, 1.32, 2.63, and 0.86 respectively. Two controls were run on the same trained network without retraining: (B) each input rescaled by the inverse of that ratio (an oracle correction that uses C1’s labels and is therefore not deployable), and (C) the input clamp removed, so that inputs outside the training range enter the polynomial unclipped. Criteria fixed in advance: (B) succeeds if $R^2 \geq 0.41$ (the level of the no-signal template on C1); (C) is significant if it exceeds the original by ≥ 0.3 .

Table S4.2: Oracle controls on the C1 fold (same network, no retraining).

Arm	R^2 (1–315)	R^2 (16–315)	RMSE 201–250	RMSE 251–315
A: original	−0.607	−0.940	68.1	45.3
B: gain-corrected inputs	−0.743	−1.108	58.9	53.6
C: input clamp removed	−1.212	−1.673	68.1	65.7
D: B and C combined	−0.743	−1.108	58.9	53.6

Neither criterion is met. Correcting the per-input gain makes the fit worse, and removing the clamp makes it worse still, so the clamp is protecting the model; after gain correction every input lies inside its training range (D equals B), yet the prediction still saturates at the clamp.

S4.3 Joint-range extrapolation

For every pass, two label-free quantities were computed from the same network: the Mahalanobis distance of the four-input vector from the distribution of the training tools’ inputs (both tools, all passes), and the leverage $h = z^T(Z^T Z)^+ z$ of the pass in the term space of each Triple node, where Z collects the node’s polynomial terms over the training passes. Criterion fixed in advance: the mechanism is a proximate cause if, in passes 201–250, at least half of the passes exceed the training maximum of the statistic and the Spearman correlation between the statistic and $|e|$ over passes 16–315 is at least 0.5.

Table S4.3: Distance of C1’s inputs from the training support.

Statistic	Training max	Share of passes above training max			Spearman with $ e $ (16–315)
		16–200	201–250	251–315	
Mahalanobis distance, 4 inputs	4.40	0%	74%	100%	+0.69
Leverage, L2 Triple node	0.192	0–3%	52%	100%	+0.70
Leverage, L3 Triple node	0.375	0%	68%	77%	+0.80

Both leverage statistics meet the criterion. We report this as a correlation within the failed fold; S4.4 shows it is not a sufficient condition for failure.

S4.4 A label-free out-of-support alarm: criteria defined before execution, negative result

Because the Mahalanobis distance needs no wear labels, it was tested as a deployment alarm on all three folds, with the thresholds, definitions, and criteria defined before execution (S7). For each fold the stage-4 network was trained once (the C6 and C4 networks reproduce the reported R^2 of 0.8935 and 0.8743) and the distance of its path inputs from the training tools was computed per pass. Alarm: distance above τ on three consecutive passes, with τ the maximum over the training passes (primary) or its 99th percentile (sensitivity). Large error: $|e| > 20 \mu\text{m}$; error onset: the first three consecutive large-error passes. False-alarm rate: share of non-large-error passes flagged; coverage: share of large-error passes flagged; lead: error onset minus first alarm (positive = earlier). Window: passes 16–315. Criterion for inclusion in the main text: false-alarm rate $\leq 10\%$ on both C6 and C4, C1 coverage $\geq 70\%$, and C1 lead ≥ 0 .

Table S4.4: Out-of-support alarm on the three LOTO folds (τ = training maximum; 99th-percentile values in parentheses).

	C1+C4→C6	C1+C6→C4	C4+C6→C1
Path inputs of the network	37	5	4
τ	16.96 (12.58)	5.98 (4.16)	4.40 (3.73)
Large-error passes (of 300)	84	11	87
False-alarm rate	98.6% (99.1%)	0% (4.8%)	18.3% (19.2%)
Coverage	98.8% (100%)	72.7% (81.8%)	72.4% (87.4%)
First alarm (pass)	16 (16)	308 (262)	217 (200)
Error onset (pass)	65	307	16
Lead (passes)	+49 (+49)	−1 (+45)	−201 (−184)

The criterion is not met: on the C6 fold the network uses 37 inputs, and in that space the new tool lies beyond the training maximum on almost every pass while the prediction is accurate; on

the C1 fold the error is already large in passes 16–50, before any alarm. The alarm was therefore not adopted, and no threshold was adjusted after the result.

S5 Screening-statistic ablation: full results

Table S5 extends the screening-statistic ablation table of the main text (Section 6.3). Mutual information: scikit-learn `mutual_info_regression` ($k=3$ nearest neighbours, random state 0). MIC: the `minepy` implementation with $\alpha=0.6$, $c=15$; because 166, 107, and 177 of the 306 features reach $\text{MIC} \geq 0.999$ on C1, C4, and C6, ties at the top are broken by a random permutation (seeds 0, 1, 2). Everything downstream of the ranking is identical to the main pipeline; the Pearson arm reproduces the reported LOTO results exactly.

Table S5: PIN per LOTO fold under each ranking statistic. R_{all}^2 : all 315 passes; R_{16+}^2 : passes 16–315; n : intersection size; J : Jaccard overlap with the Pearson set.

Statistic	C1+C4→C6				C1+C6→C4				C4+C6→C1			
	R_{all}^2	R_{16+}^2	RMSE	n (J)	R_{all}^2	R_{16+}^2	RMSE	n (J)	R_{all}^2	R_{16+}^2	RMSE	n (J)
Pearson $ r $	0.893	0.881	15.2	37 (1.00)	0.874	0.858	13.7	43 (1.00)	−0.607	−0.940	35.2	37 (1.00)
Spearman $ \rho $	0.850	0.832	18.1	58 (0.22)	0.897	0.884	12.4	43 (0.39)	−0.219	−0.475	30.7	47 (0.33)
Mutual information	0.560	0.507	31.0	54 (0.20)	0.928	0.919	10.3	53 (0.30)	−3.116	−3.978	56.3	52 (0.37)
MIC, seed 0	0.700	0.664	25.6	29 (0.22)	0.797	0.774	17.4	25 (0.11)	−0.503	−0.813	34.1	29 (0.22)
MIC, seed 1	0.906	0.896	14.3	31 (0.10)	0.807	0.783	16.9	28 (0.04)	0.362	0.234	22.2	28 (0.14)
MIC, seed 2	0.832	0.812	19.1	30 (0.08)	0.775	0.750	18.3	24 (0.14)	−1.030	−1.453	39.6	30 (0.12)

S6 Model selection and baselines: settings and full results

S6.1 Penalized regression on the degree-2 expansion (main text, Section 6.5)

The screened features of each fold (37, 43, 37) are expanded to all degree-2 terms (740, 989, 740 terms) and standardized; $n = 630$. Grouped cross-validation: `GroupKFold` with the two training tools as groups (two folds), α over 25 log-spaced values in $[10^{-2.5}, 10^2]$, and for the elastic net l_1 ratio in $\{0.5, 0.7, 0.9, 0.95, 1.0\}$. Universal threshold: $\alpha = \hat{\sigma} \sqrt{2 \ln p / n}$ with $\hat{\sigma}$ the residual standard deviation of a ridge fit ($\alpha=1$) on the standardized screened features of the training tools: $\hat{\sigma} = 5.69, 4.19, 5.86$ and $\alpha = 0.824, 0.620, 0.848$ on the C6, C4, C1 folds. Selected by cross-validation: Lasso $\alpha = 0.027, 0.027, 0.562$; elastic net $\alpha = 0.003, 0.027, 0.065$ (C6, C4, C1).

Table S6.1: R^2 over passes 16–315 (calibration window excluded), complementing the all-pass values of the model-selection table in Section 6.5 of the main text.

Model	Penalty chosen by	C1+C4→C6	C1+C6→C4	C4+C6→C1
Lasso, degree 2	2-fold grouped CV	−30.70	0.836	0.277
Elastic net, degree 2	2-fold grouped CV	−52.36	0.836	−2.291
Lasso, degree 2	universal threshold	0.879	0.814	0.335
Ridge, linear ($\alpha=1$)	fixed	−7.376	0.798	−43.45
PIN	PSE	0.881	0.858	−0.940
No-signal template	—	0.758	0.930	0.287

S6.2 Conventional baselines (main text, Section 6.4)

All models are fitted on standardized features in Δ VB space with $W_{\text{INIT}}=15$. Grouped cross-validation uses the same two-fold `GroupKFold` with mean squared error as the objective. Ridge: α over 33 log-spaced values in $[10^{-3}, 10^5]$; validation-free variant `RidgeCV` with its efficient leave-one-out (generalized cross-validation) solution over the same grid. Elastic net: α over 21 log-spaced values in $[10^{-3}, 10^2]$ and l_1 ratio in $\{0.1, 0.5, 0.9, 0.95, 1.0\}$; validation-free variant: Lasso at the universal threshold computed as in S6.1 on the respective feature set. Gradient boosting: `HistGradientBoostingRegressor` with learning rate $\{0.03, 0.1\}$, iterations $\{100, 300\}$, maximum leaf nodes $\{15, 31\}$, minimum samples per leaf $\{20, 50\}$, `random_state=0`; validation-free variant: library defaults. The no-signal template and the cubic pass polynomial are recomputed by the same script and reproduce the values of Table 7 of the main text to three decimals.

S6.3 Reproducibility of the comparisons

Everything needed to audit the protocol of Tables 7 and 8 of the main text, and to rerun every conventional-baseline row of those tables, is fixed as follows and released as Electronic Supplementary Material 2: the per-pass feature matrices, the screened feature lists of the three folds, and a self-contained script that regenerates the rows and checks them against the printed values. The PIN engine itself is proprietary (Code availability); its inputs, settings, and outputs are specified here so that its rows can be audited. Folds: defined by tool identity ($C1+C4 \rightarrow C6$, $C1+C6 \rightarrow C4$, $C4+C6 \rightarrow C1$), with no random splitting. Data: the cached per-pass feature matrices (315×306 per tool) and per-pass VB labels of the three labeled tools. Transformations: initial-baseline normalization uses each tool’s own first 15 passes as its baseline and the pooled standard deviation of the training tools as the scale; intersection screening, feature standardization, and the noise estimate $\hat{\sigma}$ are fitted on the training tools of the fold only; nothing is fitted on the test tool. Software: Python 3.13.5, NumPy 2.4.3, SciPy 1.17.1, scikit-learn 1.8.0, pandas 3.0.1; `minepy` 1.2.6 (Python 3.9 environment) for MIC only. Randomness: `random_state = 0` for gradient boosting and mutual information; MIC tie-breaking seeds 0, 1, 2 as stated; all other estimators are deterministic. Tuning rules and grids: as listed in S6.1 and S6.2, with the selected values reported there. The PIN engine and its settings are those of Section 4 of the main text (stage-4: CPM=1.0, CL=0.80, three layers, ten layer-1 survivors).

S7 Analysis plan defined before execution

Each additional experiment of the revision was entered in the project’s decision log on 2026-09-24 before it was run, with the hypotheses, the harness checks, and the decision criteria stated below; where an experiment had several steps, each step was entered before that step was run. They are reproduced here so that the criteria can be distinguished from the outcomes. The log entries and the time-stamped result files are available from the corresponding author on request. No criterion was changed after a result was seen; the one addendum (MIC tie-breaking) was entered before the affected runs started.

E1, baselines (Table 7). Hypotheses: H8a, screening disadvantages conventional learners if the all-feature variant beats the screened one by ≥ 0.05 on at least two of three folds; H8b, the tuning protocol disadvantages them if the validation-free variant beats the cross-validated one by ≥ 0.05 on at least two folds. Harness: the no-signal template must reproduce 0.784 / 0.938 / 0.410 within 0.005. Outcome: H8a not met (1/3 for each model); H8b met for sparse regression (validation-free Lasso vs. cross-validated elastic net, 2/3 with either feature set).

- E2, screening statistic (Table 6).** Hypothesis H9: a statistic replaces Pearson $|r|$ if it exceeds it by ≥ 0.05 on at least two folds. Harness: the Pearson arm must reproduce 0.8935 / 0.8743 / -0.6071 within 0.005. Addendum before the PIN runs: MIC ties at 1.0 are broken by random permutation with seeds 0, 1, 2, and all three are reported. Outcome: H9 not met for any statistic.
- E3, C1 fold diagnosis (Section 7.3, S4.1–S4.3).** Candidate mechanisms listed before the first, descriptive step: target range (C1’s smaller total increment), single-input extrapolation with the input clamp, and induction path; a fourth, joint-range extrapolation, was entered before the third step. Criteria entered before the second step: the oracle gain correction is “sufficient” if $R^2 \geq 0.41$; the input clamp is “significant” if removing it changes R^2 by ≥ 0.3 . Criterion entered before the third step: joint-range extrapolation is a “proximate cause” if $\geq 50\%$ of passes 201–250 exceed the training maximum of the statistic and its Spearman correlation with $|e|$ over passes 16–315 is ≥ 0.5 . Outcome: the second-step criteria not met; the third-step criterion met for both leverage statistics. E4 then showed the same departure on C6 without failure, which is why the main text reports the mechanism as a hypothesis, not a sufficient criterion.
- E4, out-of-support alarm (S4.4).** Thresholds, definitions, and the inclusion criterion (false-alarm rate $\leq 10\%$ on C6 and C4, C1 coverage $\geq 70\%$, C1 lead ≥ 0) as stated in S4.4. Outcome: not met; the alarm was not adopted and no threshold was adjusted.