CORD: LEARNING REUSABLE DEGRADATION REPRESENTATIONS ACROSS HETEROGENEOUS PHYSICAL SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Can heterogeneous physical degradation systems benefit from joint pretraining and move beyond system-specific prognostics toward reusable cross-system representation learning? CORD combines type-specific observation interfaces with a shared degradation backbone. Its two self-supervised objectives learn at complementary scales: Intra-Observation Structure Modeling (ISM) captures structure within observations, while Inter-Observation Dynamics Modeling (IDM) captures latent degradation evolution across observation histories. We evaluate CORD under two transfer boundaries: Pretraining-Included System Types, where downstream datasets and held-out units are unseen but their system types are represented during source pretraining, and Pretraining-Excluded System Types, where the entire turbofan-engine type is absent from pretraining. Across bearings, batteries, and cutting tools, CORD (Multi-domain) consistently improves over CORD (Single-domain) under Frozen adaptation, provides further gains under Full FT in most settings, and remains competitive with representative external baselines. Source-pretrained initialization also improves low-label adaptation to the pretraining-excluded engine type. Frozen-representation analysis further shows improved cross-unit lifecycle consistency after multi-domain pretraining. Joint pretraining across heterogeneous physical systems thus produces degradation representations reusable across devices, datasets, and system types.

1 INTRODUCTION

Physical degradation is observed through system-dependent measurements: bearings through vibration, batteries through electrochemical cycling, cutting tools through machining signals, and turbofan engines through multivariate flight histories. These observations differ in channel semantics, physical units, sampling cadence, and operating context. Industrial prognostics is therefore commonly developed in a system-specific manner, with each model learning degradation patterns from the measurements available for one asset type. Transfer-learning approaches can reduce this isolation by adapting knowledge across operating conditions or related prognostic domains (da Costa et al., 2020; Wang et al., 2026); however, they do not directly answer whether degradation knowledge can be learned jointly across physically different systems whose observations are not semantically aligned.

Large-scale pretraining has produced reusable temporal representations across diverse time-series datasets. MOMENT studies general-purpose representations and limited-supervision adaptation, while Moirai and Time-MoE address heterogeneous forecasting through universal models and sparse mixture-of-experts pretraining (Goswami et al., 2024; Woo et al., 2024; Shi et al., 2025). Tabular foundation models offer another route to data-efficient PHM prediction (Theiler et al., 2026). FeDaL addresses dataset-level heterogeneity in time-series pretraining, and FORMED adapts a shared backbone to medical datasets with different channel structures and tasks (Chen et al., 2026; Huang et al., 2026). Existing work has not established whether physically distinct degradation systems can contribute to a shared representation learner when their sensing semantics are not aligned. Section 2 reviews these directions alongside prognostics transfer.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

CORD addresses this problem by keeping observation interfaces type-specific while sharing the degradation backbone across systems. Each interface preserves its system's sensing semantics while mapping observations into the shared backbone. Two self-supervised objectives train the encoder. **Intra-Observation Structure Modeling (ISM)** captures structural relationships within a health-state observation, whereas **Inter-Observation Dynamics Modeling (IDM)** captures how latent health states evolve across observation histories. Lightweight target-aware prediction heads reuse the resulting representation, allowing the encoder to be evaluated independently of any single downstream RUL architecture.

We evaluate CORD under two transfer boundaries. **Level I: Generalization under Pretraining-Included System Types** evaluates downstream datasets that are entirely excluded from source pretraining while other datasets from the same bearing, battery, or cutting-tool type are available upstream. **Level II: Generalization under Pretraining-Excluded System Types** removes the complete target system type from source pretraining and introduces it only during downstream adaptation. XJTU-SY bearings (Wang et al., 2020), CALCE CS2 batteries (Center for Advanced Life Cycle Engineering, n.d.), PHM2010 cutting tools (PHM Society, 2010), and N-CMAPSS turbofan engines (Arias Chao et al., 2021) instantiate these two boundaries. Across Level I, CORD (Multi-domain) improves over CORD (Single-domain) in all nine Frozen settings and eight of nine Full-FT settings, while frozen-representation retrieval error decreases by 13.9%, 17.9%, and 27.3% for bearings, batteries, and cutting tools, respectively. At Level II, source pretraining also improves low-label adaptation to the previously unseen turbofan-engine type. These results indicate that the learned degradation representation is reusable beyond individual datasets and devices.

Our contributions are:

1. **Cross-system degradation representation architecture.** CORD combines type-specific observation interfaces with a shared degradation backbone, preserving native sensing semantics across physically heterogeneous systems while learning a shared degradation representation.
2. **Complementary structure–dynamics pretraining.** CORD learns degradation information at two complementary scales: ISM models structural dependencies within an observation, while IDM models latent state evolution across observation histories.
3. **Reusable multi-domain pretraining across system boundaries.** Multi-domain pretraining improves representation reuse on unseen datasets and supports low-label adaptation to a physical system type entirely excluded from source pretraining.

2 Related Work

Time-series foundation models. Large-scale pretraining has produced reusable models across heterogeneous time-series corpora. Forecasting models address variation in frequency, variate structure, and distribution through universal, decoder-based, or generative pretraining (Woo et al., 2024; Das et al., 2024; Ansari et al., 2024; Liu et al., 2024b; Shi et al., 2025). General-purpose and representation-oriented models also learn from temporal structure beyond forecasting targets, including limited-supervision adaptation, time-aware representations, subseries dependencies, and joint reconstruction and autoregression (Goswami et al., 2024; Fraikin et al., 2024; Dong et al., 2024; He et al., 2026). CORD focuses on physical degradation systems whose sensors and measured variables may have different meanings. It retains type-specific observation interfaces and shares representation learning above them.

Transfer and representation learning for prognostics. RUL transfer methods address operating-condition shifts and heterogeneous feature spaces (da Costa et al., 2020; Wang et al., 2026); tabular foundation models provide another interface for heterogeneous PHM tasks (Theiler et al., 2026). CORD learns source representations jointly from multiple physical system types and tests their reuse on held-out datasets and on a target type absent from source pretraining.

Learning under heterogeneous observations. Residual and transfer adapters combine shared transformations with specialized responses (Rebuffi et al., 2017; Houlsby et al., 2019). FeDaL handles dataset-specific heterogeneity in federated pretraining, while FORMED adapts foundation models across medical observation interfaces (Chen et al., 2026; Huang et al., 2026). In CORD,

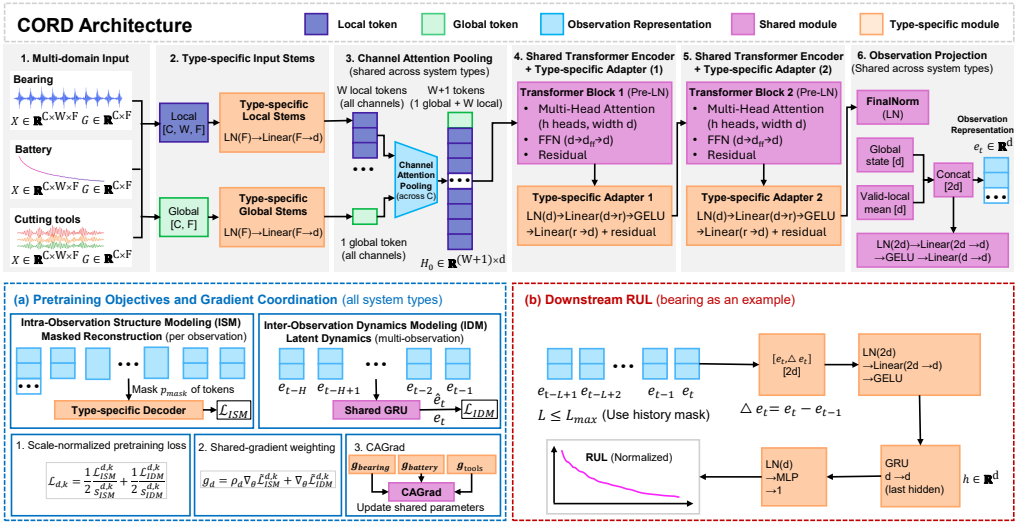

Figure 1: CORD framework. Type-specific observation interfaces preserve heterogeneous measurement semantics, while a shared Transformer backbone learns reusable degradation representations. ISM models structure within an observation and IDM models evolution across observation histories; target-aware heads reuse the encoder for downstream prognostics. Symbolic dimensions are instantiated in Appendices A.1 and B.

type-specific interfaces preserve the native measurements of each degradation system during centralized self-supervised pretraining of a shared backbone.

3 CORD: Reusable Degradation Representation Learning

3.1 Problem Setup and Transfer Boundaries

Let d denote a physical system type, k a dataset, i an individual unit, and t an ordered observation. A physical system type is a broad category of degrading assets, such as bearings, batteries, cutting tools, or turbofan engines. Each type can contain multiple datasets collected from different units, operating conditions, sensors, and acquisition protocols. The hierarchy is therefore system type $\rightarrow$ dataset $\rightarrow$ unit $\rightarrow$ observation. In the names of source-pretraining regimes, *domain* denotes a physical system type. A time-indexed sample is represented as a health-state observation $O_{i,t}^{d,k}$. Source pretraining uses system-type set $\mathcal{D}_{\text{pre}}$ and dataset collection $\mathcal{P}$. Shared parameters θ and type-specific parameters ϕ_d produce $e_{i,t} = E_{\theta,\phi_d}(O_{i,t}^{d,k})$. A target-aware readout maps an observation history to scalar remaining useful life (RUL). The primary learning objective is a reusable degradation representation; RUL prediction provides downstream tests of its reuse.

Level I: Generalization under Pretraining-Included System Types. Other datasets from the target system type belong to the source-pretraining collection $\mathcal{P}$, but the entire downstream target dataset is excluded; training and test units in that dataset are disjoint. **Level II: Generalization under Pretraining-Excluded System Types.** The target system type $d^\star$ and all its datasets are excluded from source pretraining. A new interface and prediction model may subsequently be trained using labeled observations from training units; test units do not enter model fitting or protocol selection. Both levels evaluate reuse from the same source-pretrained CORD model under different target boundaries. Frozen adaptation tests what can be read out without changing the encoder; Full FT tests the value of its initialization.

Figure 1 summarizes the complete CORD framework, including type-specific observation interfaces, shared structure–dynamics representation learning, and target-aware downstream adaptation.

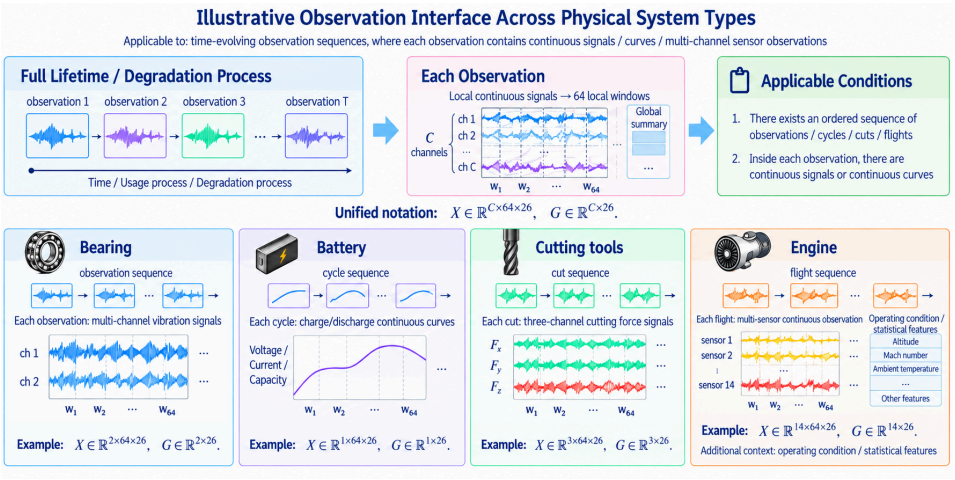

Figure 2: Observation interfaces across physical system types. Bearing, battery, cutting-tool, and engine observations retain different native signals and channel semantics, while type-specific interfaces expose a common local/global descriptor structure to the shared degradation backbone.

3.2 Type-specific observation interfaces and shared backbone

For observation t in dataset k and system type d, preprocessing constructs local descriptors $X_t^{d,k} \in \mathbb{R}^{C_{d,k}\times W\times F}$, global descriptors $G_t^{d,k} \in \mathbb{R}^{C_{d,k}\times F}$, and validity masks, where W is the number of local windows and F the descriptor dimension. The channel count $C_{d,k}$ may vary across datasets; padding is masked and excluded from representation computation. Descriptors need not share physical semantics across system types. Separate type-specific local and global stems map them to a common latent width d. The instantiated descriptor definitions and architecture dimensions are reported in Appendices A.1 and B.

Figure 2 illustrates how different physical system types retain their native observation sequences and channel semantics while exposing a common local/global descriptor structure to the shared backbone. Validity-aware channel attention forms local and global tokens. Shared Transformer blocks then process the tokens, while a type-specific residual adapter follows each block:

$$A_d(h) = W_{2,d}\,\mathrm{GELU}(W_{1,d}\,\mathrm{LN}(h)), \qquad h \leftarrow h + A_d(h). \tag{1}$$

The global state and valid-local mean feed a shared observation projector. Layer widths, adapter initialization, and readout dimensions are specified in Appendix B.

3.3 Structure–dynamics self-supervision

CORD shapes the shared degradation representation at two complementary scales: structure within the current observation and evolution across observation histories.

Intra-Observation Structure Modeling (ISM). ISM masks a fraction p_{mask} of valid local tokens ($p_{\text{mask}} = 0.30$ in our experiments) and reconstructs observed descriptor components. The reconstruction decoder is type-specific. If $\Omega_{\text{mask,obs}}$ indexes masked, observed entries, the objective is

$$L_{\text{ISM}}^{d,k} = \frac{1}{|\Omega_{\text{mask,obs}}|} \sum_{(c,j,f)\in\Omega_{\text{mask,obs}}} (\widehat{X}_{c,j,f} - X_{c,j,f})^2. \tag{2}$$

Only observed entries contribute, encouraging contextual descriptor structure within an observation.

Inter-Observation Dynamics Modeling (IDM). IDM predicts the next observation embedding from an H-step history of the same unit using a shared GRU and projection; the reported implementation uses $H = 6$:

$$\widehat{e}_t = Q_\psi(\mathrm{GRU}_\psi(e_{t-H}, \ldots, e_{t-1})), \qquad L_{\text{IDM}}^{d,k} = \mathrm{MSE}(\widehat{e}_t, \mathrm{sg}(e_t)). \tag{3}$$

Gradients are stopped through the target embedding. ISM and IDM update the same encoding system at different observation scales, coupling state structure and state evolution in the shared encoder.

3.4 Multi-domain pretraining and target-aware adaptation

Coordinated source pretraining. Fixed initial, dataset-specific scales normalize both objectives. Type-specific parameters receive both; shared parameters θ receive per-type gradients

$$g_d = \rho_d \nabla_\theta \widetilde{L}^{d,k}_{\text{ISM}} + \nabla_\theta \widetilde{L}^{d,k}_{\text{IDM}}, \tag{4}$$

Conflict-aware coordination aggregates shared gradients across system types (Liu et al., 2021); implementation details are provided in Appendix B. One selected source checkpoint initializes the three represented physical system types.

Target-aware adaptation and evaluation regimes. The downstream prediction heads are target-aware to exploit the reusable representation under each system's temporal structure. Bearings use a short observation-history GRU, batteries use cycle history and trend, cutting tools use hidden-state fusion and causal temporal modeling, and engines use flight history and operating context. We evaluate three initialization regimes under the transfer boundaries defined in Section 3.1: CORD (Scratch) starts from random initialization, CORD (Single-domain) uses pretraining from the target physical system type only, and CORD (Multi-domain) uses joint pretraining across all represented system types. All three use the same downstream architecture. During target adaptation, Frozen updates only the head, Partial FT updates selected encoder components and the head, and Full FT updates all active encoder parameters and the head. For a pretraining-excluded system type, a new type-specific input interface is randomly initialized, while the shared encoder is initialized from source pretraining.

4 Experimental Setup

4.1 Data and source-pretraining boundaries

Source pretraining uses 19 datasets across three physical system types: seven bearing, seven battery, and five cutting-tool datasets. Table 1 lists the source pools, downstream boundaries, and held-out units. Level-I target datasets are excluded in their entirety while other datasets of the same type are used upstream; Level II excludes the complete turbofan-engine type. Individual source-dataset citations are consolidated in Appendix A.

Table 1: Source-pretraining pools and transfer boundaries. Level-I targets are excluded upstream; Level II excludes the entire turbofan-engine system type.

Level / type	Source-pretraining pool	Target boundary: adaptation → test
I / Bearings	CWRU; FEMTO; Ferrara; IMS; KAIST; SEU; UNSW	XJTU-SY (Wang et al., 2020): Bearing2_2–2_5 → Bearing2_1
I / Batteries	HUST; Michigan; NASA; Oxford; KIT; SDU; XJTU	CALCE CS2 (Center for Advanced Life Cycle Engineering, n.d.): CS2_35–37 → CS2_38
I / Cutting tools	LUH; MATWI; Nonastreda; QIT-CEMC; HMoTP	PHM2010 (PHM Society, 2010): C1,C4 → C6
II / Turbofan engines	None	N-CMAPSS (Arias Chao et al., 2021): U2/U5/U10/U16/U18/U20 → U11

4.2 Training regimes, baselines, and reporting

The same architecture is evaluated as CORD (Scratch), CORD (Single-domain), or CORD (Multi-domain). External baselines include MLP (Rumelhart et al., 1986), random forest (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), TCN (Bai et al., 2018), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024a), and MOMENT (Goswami et al., 2024) with model-appropriate inputs

and readouts. We report mean ± SD over five downstream seeds at 10%, 20%, and 100% label budgets. Level-I runs use validation-selected checkpoints. For N-CMAPSS, a fixed 400-update supervised adaptation budget is chosen by grouped cross-validation on the six training engines and then applied to held-out U11 without validation-based early stopping.

5 RESULTS

5.1 GENERALIZATION UNDER PRETRAINING-INCLUDED SYSTEM TYPES

Table 2 compares CORD regimes on datasets never used in source pretraining. CORD (Multi-domain) improves over CORD (Single-domain) in all nine Frozen settings; the cutting-tool RMSE reductions are 30.6%, 30.9%, and 41.9%. Under Full FT, CORD (Multi-domain) improves over CORD (Single-domain) in eight of nine settings. These results show that multi-domain pretraining improves reuse of the learned representation, with consistent gains even when the pretrained encoder is kept fixed.

Table 2: Level-I mean RMSE. **Best** and second-best values are ranked within each system–budget column; complete metrics are in Appendix C.

		Bearings			Batteries			Cutting tools		
Source	Adaptation	10%	20%	100%	10%	20%	100%	10%	20%	100%
Scratch	—	.1463	.1400	.1122	.0711	.0680	**.0597**	.1169	.1007	.0782
Single-domain	Frozen	.1229	.1395	.1219	.0739	.0780	.0657	.1245	.1131	.1133
Multi-domain	Frozen	**.1087**	**.1245**	.0991	.0704	**.0676**	.0636	.0864	.0782	.0658
Single-domain	Partial FT	.1211	.1418	.1034	.0714	.0818	.0693	.1194	.1088	.0920
Multi-domain	Partial FT	.1175	.1351	.1063	.0726	.0732	.0739	**.0791**	.0660	**.0491**
Single-domain	Full FT	.1121	.1464	.0988	.0689	.0735	.0652	.1082	.0930	.0800
Multi-domain	Full FT	.1240	.1329	**.0972**	**.0673**	.0683	.0632	.0871	**.0609**	.0530

Figure 3 summarizes the relative effect of Multi-domain versus Single-domain pretraining across Frozen, Partial FT, and Full FT. Multi-domain pretraining yields lower RMSE in all Frozen settings, providing the clearest evidence that the jointly learned representation is more reusable without encoder updates. It retains an advantage in most Partial FT and Full FT settings, although the gains are less uniform after fine-tuning. Cutting tools show the largest and most consistent benefits across all three adaptation strategies.

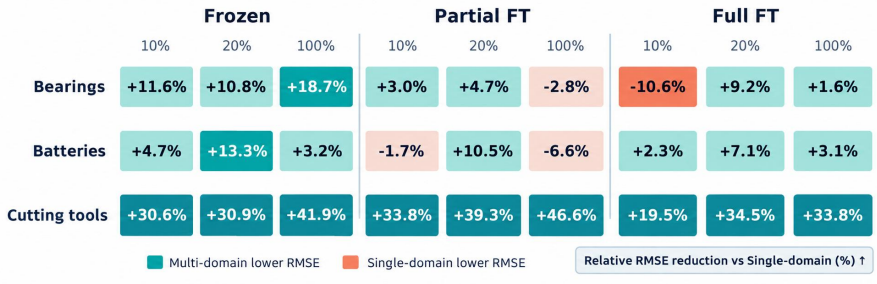

Figure 3: Multi-domain pretraining across adaptation regimes. Cells show the relative RMSE reduction of CORD (Multi-domain) with respect to CORD (Single-domain); the in-figure key identifies which source regime has lower RMSE.

Table 3 compares CORD with representative RUL prediction baselines. The pretrained CORD rows use Full FT as a common adaptation protocol, Scratch starts from random initialization, and MOMENT is evaluated with a frozen encoder. Other methods use their model-appropriate protocols; complete uncertainty and metrics are in Appendix C.4.

Table 3: RUL prediction comparison: mean RMSE. CORD (Single-domain) and CORD (Multi-domain) use Full FT; Scratch trains from random initialization. Each system–budget column ranks all displayed rows.

	Bearings			Batteries			Cutting tools		
Model	10%	20%	100%	10%	20%	100%	10%	20%	100%
CORD (Scratch)	.1463	**.1400**	.1122	.0711	**.0680**	**.0597**	.1169	.1007	.0782
CORD (Single-domain)	**.1121**	.1464	**.0988**	.0689	.0735	.0652	.1082	**.0930**	.0800
CORD (Multi-domain)	**.1240**	**.1329**	**.0972**	**.0673**	**.0683**	.0632	**.0871**	**.0609**	**.0530**
MLP	.2185	.2307	.2160	.1526	.1408	.1005	**.0991**	.1085	.1177
Random forest	.1840	.1910	.1853	.1085	.1110	.1054	.1117	.1202	.1230
XGBoost	.1494	.1824	.1846	.1036	.1052	.0992	.1258	.1204	.1214
TCN	.2681	.2930	.2287	.0807	.0813	.0788	.1409	.1266	**.0722**
PatchTST	.3019	.3338	.2663	**.0676**	.0722	.0792	.2258	.2103	.1810
iTransformer	.3138	.3157	.3071	.0904	.0893	.0813	.2728	.2582	.1890
MOMENT	.3070	.3176	.2711	.0730	.0695	**.0620**	.2452	.2431	.2421

A CORD regime achieves the lowest displayed RMSE in all nine system–budget columns, with CORD (Multi-domain) obtaining the best result in six of nine and outperforming the representative external baselines across the table. The gains are most pronounced in low-label settings, particularly for cutting tools. As target supervision increases, CORD (Scratch) becomes competitive on some targets, consistent with pretraining providing the largest benefit when labeled target data are limited.

5.2 Generalization under Pretraining-Excluded System Types

N-CMAPSS turbofan engines are entirely absent from source pretraining. Level II uses a fixed 400-update training budget because labeled target data are more limited, avoiding an additional per-seed validation split. The 400-update budget is selected using grouped cross-validation on the six training engines only and then fixed for evaluation on the held-out U11 engine. Table 4 and Figure 4 compare Scratch and source-pretrained initialization under the same 400-update budget. Source-pretrained initialization reduces U11 RMSE from .1463 to .1374 at 10% labels (6.1%) and from .1163 to .1122 at 20% (3.4%); MAE and R^2 show the same trend. At 100% labels, Scratch achieves the lower RMSE (.0768 versus .0841). Overall, the benefit of source pretraining is strongest in the low-label regime.

Table 4: Level-II N-CMAPSS U11 results under matched 400-update adaptation (mean ± SD over five seeds).

	RMSE ↓		MAE ↓		R^2 ↑	
Labels	Scratch	Source-pretrained	Scratch	Source-pretrained	Scratch	Source-pretrained
10%	**.1463 ± .0134**	**.1374 ± .0117**	**.1151 ± .0103**	**.1098 ± .0091**	**.6518 ± .0611**	**.6932 ± .0510**
20%	**.1163 ± .0131**	**.1122 ± .0069**	**.0903 ± .0090**	**.0883 ± .0070**	**.7795 ± .0482**	**.7959 ± .0252**
100%	**.0768 ± .0154**	**.0841 ± .0099**	**.0566 ± .0112**	**.0582 ± .0055**	**.9018 ± .0390**	**.8844 ± .0274**

5.3 Multi-domain pretraining improves the frozen representation

Before fitting the RUL heads, we evaluate whether the frozen representation captures lifecycle structure consistently across units. Figure 5 provides a quantitative measure using 5-NN lifecycle error, defined as the normalized-RUL discrepancy between a held-out query and its five nearest reference-unit states. CORD (Multi-domain) reduces this error relative to CORD (Single-domain) by 13.9% for bearings, 17.9% for batteries, and 27.3% for cutting tools, indicating more lifecycle-consistent local neighborhoods. Figure 6 provides a complementary qualitative view of the same representation space. Under multi-domain pretraining, samples within each system type show a clearer and more continuous progression with normalized RUL, consistent with the quantitative reductions in 5-NN lifecycle error.

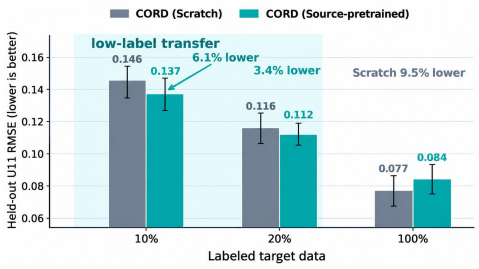

Figure 4: Pretraining-excluded engine transfer. Held-out U11 RMSE under matched 400-update adaptation; whiskers show one sample SD.

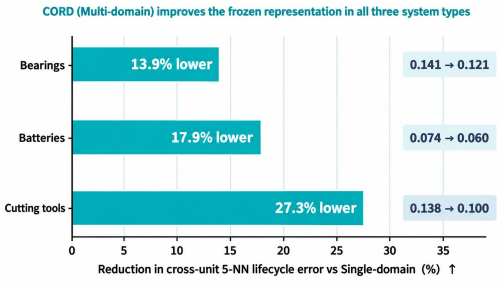

Figure 5: Multi-domain pretraining improves frozen representation reuse. Bars show relative reductions; callouts show absolute changes.

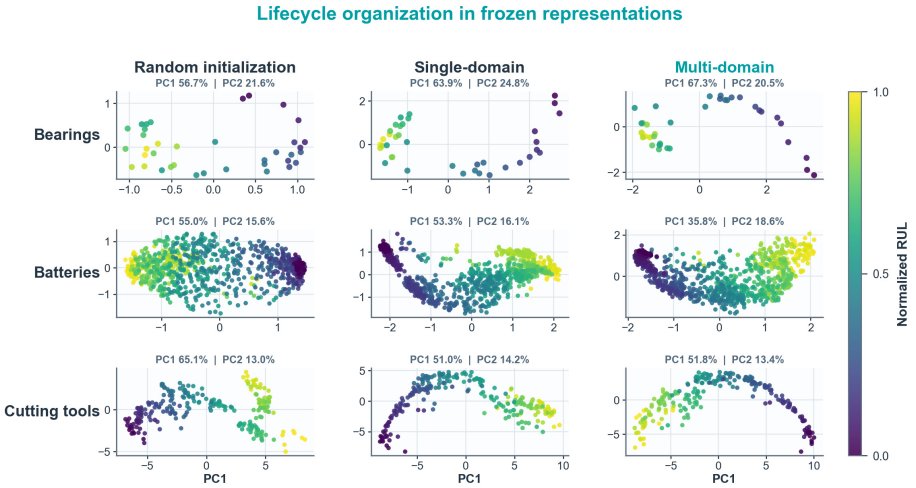

Figure 6: Lifecycle organization in frozen representations. Points are colored by normalized RUL. PCA is fitted independently within each encoder condition and is used only to visualize within-panel lifecycle organization.

5.4 Ablations on Observation Interfaces and Pretraining Objectives

Observation interface. Table 5 compares the proposed **Structured Descriptors** with **Raw-Resampled Inputs** under the same Scratch protocol. Structured Descriptors achieve lower RMSE in all nine system–budget settings, with Raw-Resampled Inputs yielding 2.16–3.00× larger errors. This result shows that the performance gain is mainly attributed to the structured observation interface rather than input dimensionality.

Figure 2 illustrates how the same local/global observation structure accommodates native bearing, battery, cutting-tool, and engine measurements before they enter the shared representation backbone.

Table 5: Observation-interface ablation under the matched Scratch protocol. Structured Descriptors are compared with parameter-free Raw-Resampled Inputs using the same input dimensionality and downstream architecture. Entries report mean RMSE over five downstream seeds; lower is better.

	Bearings			Batteries			Cutting tools		
Input construction	10%	20%	100%	10%	20%	100%	10%	20%	100%
Structured Descriptors	**.1463**	**.1400**	**.1122**	**.0711**	**.0680**	**.0597**	**.1169**	**.1007**	**.0782**
Raw-Resampled Inputs	.3590	.3157	.3091	.2131	.1715	.1724	.2845	.2178	.1894

Inter-observation dynamics (IDM). Table 6 evaluates the contribution of IDM to transfer on the pretraining-excluded turbofan-engine type using N-CMAPSS U11.

Table 6: Pretraining-objective ablation on N-CMAPSS U11. ISM-only and ISM+IDM source-pretrained encoders use the same fixed 400-update adaptation protocol. Entries are mean RMSE ± sample SD over five seeds; lower is better.

Pretraining objective	10% labels	20% labels	100% labels
ISM only	**0.1725 ± 0.0156**	**0.1448 ± 0.0059**	**0.0944 ± 0.0189**
ISM + IDM	**0.1374 ± 0.0117**	**0.1122 ± 0.0069**	**0.0841 ± 0.0099**
RMSE reduction	**20.3%**	**22.5%**	**10.9%**

Adding IDM reduces mean RMSE by 20.3%, 22.5%, and 10.9% at 10%, 20%, and 100% labels, respectively. These results show that inter-observation dynamics provide complementary information beyond within-observation structure, with the largest gains under low-label cross-type adaptation. Complete MAE and R^2 results are reported in Appendix D.

5.5 Efficiency and Reusable Deployment

CORD Full FT predictors contain only 0.28–0.38M parameters across the three Level-I systems. With one frozen shared encoder and three heads, deployment uses 657,235 parameters versus 1,022,899 for three extracted copies, reducing parameter storage by 35.75%. These results show that representation reuse can be achieved with a compact shared backbone. Detailed timing and memory results are reported in Appendix G.

6 Discussion: What Makes Degradation Representations Reusable?

Type-specific interfaces enable cross-system representation learning. CORD does not require physical system types to share sensors or variables. Type-specific interfaces preserve system-dependent sensing semantics while mapping observations into a shared degradation backbone. The observation-interface ablation shows that the structured observation interface consistently outperforms matched Raw-Resampled Inputs across the three represented system types.

Structure and dynamics provide complementary pretraining signals. ISM models structural dependencies within individual observations, while IDM models latent state evolution across observation histories. The Level II ablation on the pretraining-excluded engine type shows that adding IDM reduces prediction error across all label budgets, indicating that inter-observation dynamics provide complementary information beyond within-observation structure.

Multi-domain pretraining improves representation reuse across system boundaries. At Level I, CORD (Multi-domain) outperforms CORD (Single-domain) in all nine Frozen settings, while cross-unit 5-NN lifecycle error also decreases for bearings, batteries, and cutting tools. At Level II, source-pretrained initialization improves adaptation to the pretraining-excluded engine type at 10% and 20% labels, while its advantage diminishes at 100%. These results show that multi-domain pretraining supports representation reuse both within represented system types and beyond the source-pretraining system boundary.

7 Conclusion

CORD separates heterogeneous observation modeling from shared degradation representation learning. Type-specific interfaces preserve system-dependent measurement semantics, while ISM and IDM capture within-observation structure and inter-observation dynamics in a shared backbone. At Level I, multi-domain pretraining improves frozen representation reuse on held-out bearing, battery, and cutting-tool datasets. At Level II, source pretraining improves low-label adaptation to a pretraining-excluded engine type. Overall, the results demonstrate the feasibility of reusable degradation representation learning across heterogeneous physical systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with literature search, research methodology and experiment design, code implementation and debugging, result analysis and interpretation, figure preparation, and manuscript organization, drafting, and language refinement. We did not use generative AI tools to generate experimental data or to replace the execution and evaluation of the reported experiments. We reviewed all AI-assisted work: literature suggestions were checked against the original sources, generated code was manually inspected and tested, numerical claims were verified against experimental outputs, and AI-assisted text and figures were reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims, code, analyses, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The experiment packages retain preprocessing implementations, protocol files, checkpoints, per-seed metrics, and prediction arrays. Appendices A and B document the data construction, architecture, optimization, and model-selection protocols, while Appendices C–G provide complete results and computational details. The implementations, experiment configurations, and retained experiment records are available in the anonymous code repository.

ETHICS STATEMENT

The intended application is research on equipment health monitoring. Prediction errors can affect maintenance decisions; these experiments do not establish safety for autonomous operational deployment. Any release of derived data or model artifacts will comply with dataset licenses and redistribution permissions.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2403.07815.

Manuel Arias Chao, Chetan Kulkarni, Kai Goebel, and Olga Fink. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, 6(1):5, 2021. doi: 10.3390/data6010005. URL https://doi.org/10.3390/data6010005.

Luca Arpa, Alberto Gabrielli, Mattia Battarra, and Emiliano Mucchi. University of Ferrara run-to-failure vibration dataset of self-aligning double-row ball bearings. *Data in Brief*, 55:110620, 2024. doi: 10.1016/j.dib.2024.110620. URL https://doi.org/10.1016/j.dib.2024.110620.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018. URL https://arxiv.org/abs/1803.01271.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Case Western Reserve University Bearing Data Center. CWRU bearing data center. Case School of Engineering, n.d. URL https://engineering.case.edu/bearingdatacenter/welcome.

Center for Advanced Life Cycle Engineering. CALCE battery data: CS2 prismatic cells. University of Maryland battery data archive, n.d. URL https://calce.umd.edu/data.

Shengchao Chen, Guodong Long, Michael Blumenstein, and Jing Jiang. FeDaL: Federated dataset learning for general time series foundation models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a0a53fefef4c2ad72d5ab79703ba70cb-Abstract-Conference.html.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Paulo Roberto de Oliveira da Costa, Alp Akçay, Yingqian Zhang, and Uzay Kaymak. Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering & System Safety*, 195:106682, 2020. doi: 10.1016/j.ress.2019.106682. URL https://doi.org/10.1016/j.ress.2019.106682.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 10148–10167. PMLR, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Lars De Pauw, Tom Jacobs, and Toon Goedemé. MATWI: A multimodal automatic tool wear inspection dataset and baseline algorithms. In *International Conference on Computer Vision Systems*, pp. 255–269, 2023. doi: 10.1007/978-3-031-44137-0_22. URL https://doi.org/10.1007/978-3-031-44137-0_22.

Berend Denkena, Heinrich Klemme, and Tobias H. Stiehl. Multivariate time series data of milling processes with varying tool wear and machine tools. *Data in Brief*, 50:109574, 2023. doi: 10.1016/j.dib.2023.109574. URL https://doi.org/10.1016/j.dib.2023.109574.

Jiaxiang Dong, Haixu Wu, Yuxuan Wang, Yun-Zhong Qiu, Li Zhang, Jianmin Wang, and Mingsheng Long. TimeSiam: A pre-training framework for siamese time-series modeling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 11412–11436. PMLR, 2024.

Archibald Fraikin, Adrien Bennetot, and Stephanie Allassonniere. T-Rep: Representation learning for time series using time-embeddings. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/84b946c4c4162b29464fc7aa57e480e1-Paper-Conference.pdf.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 16115–16152, 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Cheng He, Xu Huang, Gangwei Jiang, Zhaoyi Li, Defu Lian, Hong Xie, Enhong Chen, Xijie Liang, Zengrong Zheng, and Patrick P. C. Lee. GTM: A general time-series model for enhanced representation learning of time-series data. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0c6639f49f01a8578675303ce0030233-Abstract-Conference.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2790–2799, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

David Howey and Christoph Birkl. Oxford battery degradation dataset 1. University of Oxford Research Archive, 2017.

Nan Huang, Haishuai Wang, Zihuai He, Marinka Zitnik, and Xiang Zhang. Repurposing foundation model for generalizable medical time series classification. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8707924df5e207fa496f729f49069446-Paper-Conference.pdf.

Wonho Jung, Sung-Hyun Yun, and Yong-Hwa Park. Vibration and temperature run-to-failure dataset of ball bearing for prognostics. *Data in Brief*, 54:110403, 2024. doi: 10.1016/j.dib.2024.110403. URL https://doi.org/10.1016/j.dib.2024.110403.

J. Lee, H. Qiu, G. Yu, J. Lin, and Rexnord Technical Services. Bearing data set, IMS, university of cincinnati. NASA Ames Prognostics Data Repository, 2007. URL https://data.nasa.gov/dataset/ims-bearings.

Na Li, Xiao Wang, Wanzhen Wang, Miaomiao Xin, Dongfeng Yuan, and Mingqiang Zhang. A multi-feature dataset of coated end milling cutter tool wear whole life cycle. *Scientific Data*, 12: 16, 2025.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9d27fdf2477ffbff837d73ef7ae23db9-Abstract.html.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024a. URL https://arxiv.org/abs/2310.06625.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32369–32399. PMLR, 2024b. URL https://proceedings.mlr.press/v235/liu24cb.html.

Matthias Luh and Thomas Blank. Comprehensive battery aging dataset: Capacity and impedance fade measurements of a lithium-ion NMC/C-SiO cell. *Scientific Data*, 11:1004, 2024. doi: 10.1038/s41597-024-03831-x. URL https://doi.org/10.1038/s41597-024-03831-x.

Guijun Ma, Songpei Xu, Benben Jiang, et al. Real-time personalized health status prediction of lithium-ion batteries using deep transfer learning. *Energy & Environmental Science*, 15:4083–4094, 2022.

Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Noureddine Zerhouni, and Christophe Varnier. PRONOSTIA: An experimental platform for bearings accelerated degradation tests. In *IEEE Conference on Prognostics and Health Management*, pp. 1–8, 2012.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.14730.

PHM Society. 2010 PHM society conference data challenge, 2010. URL https://phmsociety.org/phm_competition/2010-phm-society-conference-data-challenge/.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/e7b24b112a44fdd9ee93bdf998c6ca0e-Abstract.html.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0. URL https://doi.org/10.1038/323533a0.

Bhaskar Saha and Kai Goebel. Battery data set. NASA Ames Prognostics Data Repository, 2007.

Siyu Shao, Stephen McAleer, Ruqiang Yan, and Pierre Baldi. Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, 15(4):2446–2455, 2019.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/558d48c1f08675daa636e09bfe94a89e-Abstract-Conference.html.

Wade A. Smith and Robert B. Randall. Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical Systems and Signal Processing*, 64–65: 100–131, 2015. doi: 10.1016/j.ymssp.2015.04.021. URL https://doi.org/10.1016/j.ymssp.2015.04.021.

Raffael Theiler, Lev Telyatnikov, Leandro von Krannichfeldt, and Olga Fink. Towards unified and data-efficient prognostics and health management with tabular foundation models, 2026. URL https://arxiv.org/abs/2606.05481.

Hubert Truchan and Zahra Ahmadi. Nonastreda multimodal dataset for efficient tool wear state monitoring. *Data in Brief*, 62:111905, 2025.

Biao Wang, Yaguo Lei, Naipeng Li, and Ningbo Li. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, 69(1):401–412, 2020. doi: 10.1109/TR.2018.2882682. URL https://doi.org/10.1109/TR.2018.2882682.

Bingsen Wang, Piero Baraldi, Ahmed Shokry, and Enrico Zio. A novel two-stage heterogeneous transfer learning framework for the estimation of the remaining useful life of industrial components. *Reliability Engineering & System Safety*, 2026. doi: 10.1016/j.ress.2025.111968. URL https://doi.org/10.1016/j.ress.2025.111968.

Fujin Wang, Z. Zhai, Z. Zhao, et al. Physics-informed neural network for lithium-ion battery degradation stable modeling and prognosis. *Nature Communications*, 15:4332, 2024a.

Runqiong Wang, Qinghua Song, Y. Peng, et al. Toward digital twins for high-performance manufacturing: Tool wear monitoring in high-speed milling of thin-walled parts using domain knowledge. *Robotics and Computer-Integrated Manufacturing*, 88:102723, 2024b.

Shuquan Wang, Feng Gao, and Hao Tian. Deep sorting of reused batteries for enabling long-term consistency grouping with unknown prior conditions. *Cell Reports Physical Science*, 6 (7):102657, 2025.

Andrew Weng, Peyman Mohtat, Peter M. Attia, et al. Predicting the impact of formation protocols on battery lifetime immediately after manufacturing. *Joule*, 5(11):2971–2992, 2021.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 53140–53164, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Hengcheng Zhang, Pietro Borghesani, Robert B. Randall, and Zhongxiao Peng. A benchmark of measurement approaches to track the natural evolution of spall severity in rolling element bearings. *Mechanical Systems and Signal Processing*, 166:108466, 2022.

A Data Boundaries and Observation Construction

Source pretraining uses 19 datasets across the three represented physical system types. The downstream datasets are excluded in their entirety from source pretraining. Historical three-channel cutting-tool results are excluded from this study; PHM2010 uses its seven recorded channels. Observation descriptors use 64 local slots and 26 feature dimensions with validity masks. Target construction and dataset-specific preprocessing are recorded in the released manifests. Main-text Table 1 reports the source pools, transfer boundaries, and downstream units in one place.

Table 7: Source-pretraining and downstream target datasets with their direct data-source citations. References are consolidated here to keep the main-text data description readable.

System type	Dataset	Reference
Bearings	CWRU	(Case Western Reserve University Bearing Data Center, n.d.; Smith & Randall, 2015)
	FEMTO / PRONOSTIA	(Nectoux et al., 2012)
	Ferrara	(Arpa et al., 2024)
	IMS	(Lee et al., 2007)
	KAIST	(Jung et al., 2024)
	SEU	(Shao et al., 2019)
	UNSW	(Zhang et al., 2022)
Batteries	HUST	(Ma et al., 2022)
	Michigan	(Weng et al., 2021)
	NASA	(Saha & Goebel, 2007)
	Oxford	(Howey & Birkl, 2017)
	KIT NMC/C-SiO	(Luh & Blank, 2024)
	SDU	(Wang et al., 2025)
	XJTU	(Wang et al., 2024a)
Cutting tools	LUH	(Denkena et al., 2023)
	MATWI	(De Pauw et al., 2023)
	Nonastreda	(Truchan & Ahmadi, 2025)
	QIT-CEMC	(Li et al., 2025)
	HMoTP	(Wang et al., 2024b)
Downstream targets	XJTU-SY bearings	(Wang et al., 2020)
	CALCE CS2 batteries	(Center for Advanced Life Cycle Engineering, n.d.)
	PHM2010 cutting tools	(PHM Society, 2010)
	N-CMAPSS turbofan engines	(Arias Chao et al., 2021)

A.1 Structured Descriptor Definitions

For the reported implementation, $F = 26$; Table 8 records the exact feature order used by the three represented system-type interfaces. Identical names in the bearing and cutting-tool columns denote the same statistic applied to different native sensor channels; they do not impose shared channel semantics across system types.

Table 8: Ordered 26-dimensional Structured Descriptors exposed by each source-system observation interface. Bearing and cutting-tool descriptors are computed independently for every valid native sensor channel and local signal window. Battery descriptors are computed over valid local discharge windows; unavailable temperature entries are masked rather than imputed as observations.

Index	Bearings	Batteries	Cutting tools
1	Mean	Mean voltage	Mean
2	Mean absolute value	Voltage standard deviation	Mean absolute value
3	Standard deviation	Minimum voltage	Standard deviation
4	Variance	Maximum voltage	Variance
5	Root mean square	Voltage peak-to-peak range	Root mean square
6	Mean-square energy	Voltage skewness	Mean-square energy
7	Absolute peak amplitude	Voltage kurtosis	Absolute peak amplitude

Continued on next page.

Table 8 continued.

Index	Bearings	Batteries	Cutting tools
8	Peak-to-peak range	Voltage slope	Peak-to-peak range
9	Minimum	Mean absolute voltage derivative	Minimum
10	Maximum	Mean absolute voltage curvature	Maximum
11	Skewness	Mean C-rate	Skewness
12	Kurtosis	C-rate standard deviation	Kurtosis
13	Crest factor	Minimum C-rate	Crest factor
14	Shape factor	Maximum C-rate	Shape factor
15	Impulse factor	Mean absolute C-rate	Impulse factor
16	Clearance factor	C-rate slope	Clearance factor
17	Root amplitude	Mean temperature	Root amplitude
18	Zero-crossing rate	Temperature standard deviation	Zero-crossing rate
19	Spectral centroid	Minimum temperature	Spectral centroid
20	Spectral bandwidth	Maximum temperature	Spectral bandwidth
21	Spectral entropy	Temperature change	Spectral entropy
22	Dominant frequency	Temperature slope	Dominant frequency
23	Low-band normalized energy	Local duration	Low-band normalized energy
24	Mid-band normalized energy	Local capacity	Mid-band normalized energy
25	High-band normalized energy	Local energy	High-band normalized energy
26	High/low-band energy ratio	Mean $\mathrm{d}V/\mathrm{d}Q$	High/low-band energy ratio

B Architecture, Optimization, and Model Selection

The reported implementation uses $W = 64$, $F = 26$, $d = 96$, $h = 4$, $d_{\mathrm{ff}} = 192$, and $r = 24$. Type-specific local and global stems map the descriptors to width d. Validity-aware channel attention yields W local tokens plus one global token. Two shared pre-normalized Transformer blocks use h attention heads and feed-forward width d_{ff}; a residual adapter after each block has bottleneck width r and a zero-initialized final map. The global state and valid-local mean form a $2d$-dimensional concatenation, which is projected back to d to produce the observation representation. Cutting-tool prediction may also use the $2d$-dimensional final-normalized token readout.

The selected CORD checkpoint uses a bearing-gradient directional projection following CAGrad. For dataset k in system type d, initial source-training batches fix scales $s^{d,k}_{\mathrm{ISM}}$ and $s^{d,k}_{\mathrm{IDM}}$:

$$\widetilde{L}^{d,k}_{\mathrm{ISM}} = L^{d,k}_{\mathrm{ISM}}/(2s^{d,k}_{\mathrm{ISM}}), \qquad \widetilde{L}^{d,k}_{\mathrm{IDM}} = L^{d,k}_{\mathrm{IDM}}/(2s^{d,k}_{\mathrm{IDM}}). \tag{5}$$

System-type-specific stems, adapters, channel embeddings, and decoders receive both normalized components. For shared parameters, Eq. 4 uses fixed ISM coefficients $(0.3, 0.3, 1)$ for bearings, batteries, and cutting tools; IDM is not multiplied by $1-\rho_d$. CAGrad uses $\alpha = 0.4$ and its rescale = 1 convention. Given its output v and bearing gradient g_b, the implementation then applies

$$v' = v + \frac{[\|g_b\|^2/D - g_b^{\top}v]_+}{\max(\|g_b\|^2, 10^{-20})} g_b, \qquad D = 3. \tag{6}$$

This minimum Euclidean correction enforces the specified bearing-gradient directional floor before clipping and AdamW. Single-domain training uses the corresponding interface, scales, and coefficient without multi-domain gradient aggregation.

Each epoch has 20 joint updates, each with batch 32 per active system type and microbatch eight. Source datasets rotate within system types. AdamW uses learning rate and weight decay 10^{-4}, with a 2,000-epoch ceiling, patience 30, and minimum validation improvement 10^{-4}. Source selection averages dataset loss ratios within system types, then averages system types. The validation total is reconstruction plus 0.2 times dynamics relative to its initial value; it differs from the separately normalized training objective. One joint checkpoint is selected for all represented system types.

The target-aware RUL readouts use nonlinear temporal heads. Bearings use up to six observations and a GRU; batteries use a 20-cycle history/trend head; cutting tools use up to 20 observations, hidden-state fusion, a causal TCN, and a GRU. Engines use seven-flight histories and operating context. Represented-type fine-tuning uses encoder learning rate 3×10^{-4} and head rate 10^{-3}; CORD (Scratch) uses 10^{-3}. Partial FT updates only designated final encoder components.

C Generalization under Pretraining-Included System Types: Complete Results

Tables report mean ± SD. **Red bold** marks the lowest displayed mean and **blue bold underlined** the second-lowest within each physical system type and label budget among the listed adaptation procedures.

C.1 Bearing

Table 9: Bearing adaptation on XJTU-SY: RMSE, MAE, and R^2 at all three label budgets (five seeds).

Labels	Treatment	RMSE	MAE	R^2
10%	CORD (Scratch)	0.1463 ± 0.0393	0.1172 ± 0.0292	0.7428 ± 0.1406
10%	CORD (Single-domain) Frozen	0.1229 ± 0.0176	0.0984 ± 0.0207	0.8256 ± 0.0519
10%	CORD (Multi-domain) Frozen	**0.1087 ± 0.0153**	**0.0859 ± 0.0114**	**0.8635 ± 0.0379**
10%	CORD (Single-domain) Partial FT	0.1211 ± 0.0318	0.0944 ± 0.0261	0.8242 ± 0.0959
10%	CORD (Multi-domain) Partial FT	0.1175 ± 0.0190	0.0977 ± 0.0155	0.8397 ± 0.0523
10%	CORD (Single-domain) Full FT	**0.1121 ± 0.0224**	**0.0833 ± 0.0142**	**0.8526 ± 0.0623**
10%	CORD (Multi-domain) Full FT	0.1240 ± 0.0289	0.0979 ± 0.0239	0.8177 ± 0.0859
20%	CORD (Scratch)	0.1400 ± 0.0145	0.1081 ± 0.0112	0.7752 ± 0.0453
20%	CORD (Single-domain) Frozen	0.1395 ± 0.0185	0.1141 ± 0.0086	0.7758 ± 0.0581
20%	CORD (Multi-domain) Frozen	**0.1245 ± 0.0097**	**0.0994 ± 0.0095**	**0.8229 ± 0.0283**
20%	CORD (Single-domain) Partial FT	0.1418 ± 0.0252	0.1127 ± 0.0210	0.7656 ± 0.0879
20%	CORD (Multi-domain) Partial FT	0.1351 ± 0.0171	0.1108 ± 0.0120	0.7899 ± 0.0516
20%	CORD (Single-domain) Full FT	0.1464 ± 0.0142	0.1100 ± 0.0132	0.7546 ± 0.0474
20%	CORD (Multi-domain) Full FT	**0.1329 ± 0.0214**	**0.1044 ± 0.0142**	**0.7951 ± 0.0697**
100%	CORD (Scratch)	0.1122 ± 0.0233	0.0854 ± 0.0143	0.8519 ± 0.0637
100%	CORD (Single-domain) Frozen	0.1219 ± 0.0107	0.0940 ± 0.0086	0.8301 ± 0.0308
100%	CORD (Multi-domain) Frozen	0.0991 ± 0.0166	0.0805 ± 0.0121	0.8857 ± 0.0371
100%	CORD (Single-domain) Partial FT	0.1034 ± 0.0182	0.0802 ± 0.0131	0.8753 ± 0.0455
100%	CORD (Multi-domain) Partial FT	0.1063 ± 0.0190	0.0842 ± 0.0137	0.8682 ± 0.0488
100%	CORD (Single-domain) Full FT	**0.0988 ± 0.0184**	**0.0718 ± 0.0162**	**0.8859 ± 0.0436**
100%	CORD (Multi-domain) Full FT	**0.0972 ± 0.0071**	**0.0744 ± 0.0076**	**0.8921 ± 0.0160**

C.2 Battery

Table 10: Battery adaptation on CALCE CS2: RMSE, MAE, and R^2 at all three label budgets (five seeds).

Labels	Treatment	RMSE	MAE	R^2
10%	CORD (Scratch)	0.0711 ± 0.0091	0.0516 ± 0.0065	0.9384 ± 0.0146
10%	CORD (Single-domain) Frozen	0.0739 ± 0.0059	0.0547 ± 0.0042	0.9339 ± 0.0106
10%	CORD (Multi-domain) Frozen	0.0704 ± 0.0028	0.0522 ± 0.0019	0.9402 ± 0.0048
10%	CORD (Single-domain) Partial FT	0.0714 ± 0.0030	0.0522 ± 0.0019	0.9385 ± 0.0051
10%	CORD (Multi-domain) Partial FT	0.0726 ± 0.0040	0.0529 ± 0.0031	0.9364 ± 0.0070
10%	CORD (Single-domain) Full FT	**0.0689 ± 0.0048**	**0.0492 ± 0.0035**	**0.9426 ± 0.0078**
10%	CORD (Multi-domain) Full FT	**0.0673 ± 0.0064**	**0.0487 ± 0.0044**	**0.9450 ± 0.0105**
20%	CORD (Scratch)	**0.0680 ± 0.0060**	0.0497 ± 0.0056	**0.9439 ± 0.0101**
20%	CORD (Single-domain) Frozen	0.0780 ± 0.0042	0.0561 ± 0.0038	0.9265 ± 0.0080
20%	CORD (Multi-domain) Frozen	**0.0676 ± 0.0012**	**0.0494 ± 0.0009**	**0.9450 ± 0.0019**

Continued on next page

Labels	Treatment	RMSE	MAE	R^2
20%	CORD (Single-domain) Partial FT	0.0818 ± 0.0057	0.0594 ± 0.0038	0.9191 ± 0.0109
20%	CORD (Multi-domain) Partial FT	0.0732 ± 0.0052	0.0524 ± 0.0040	0.9351 ± 0.0092
20%	CORD (Single-domain) Full FT	0.0735 ± 0.0050	0.0533 ± 0.0044	0.9346 ± 0.0089
20%	CORD (Multi-domain) Full FT	0.0683 ± 0.0043	**0.0484 ± 0.0035**	0.9436 ± 0.0072
100%	CORD (Scratch)	**0.0597 ± 0.0085**	**0.0435 ± 0.0073**	**0.9564 ± 0.0125**
100%	CORD (Single-domain) Frozen	0.0657 ± 0.0070	0.0480 ± 0.0055	0.9475 ± 0.0115
100%	CORD (Multi-domain) Frozen	0.0636 ± 0.0018	**0.0454 ± 0.0026**	0.9512 ± 0.0028
100%	CORD (Single-domain) Partial FT	0.0693 ± 0.0013	0.0504 ± 0.0005	0.9421 ± 0.0022
100%	CORD (Multi-domain) Partial FT	0.0739 ± 0.0085	0.0527 ± 0.0040	0.9334 ± 0.0159
100%	CORD (Single-domain) Full FT	0.0652 ± 0.0064	0.0464 ± 0.0054	0.9483 ± 0.0103
100%	CORD (Multi-domain) Full FT	**0.0632 ± 0.0056**	0.0458 ± 0.0050	**0.9515 ± 0.0086**

C.3 CUTTING TOOLS

Table 11: Cutting-tool adaptation on PHM2010: RMSE, MAE, and R^2 at all three label budgets (five seeds).

Labels	Treatment	RMSE	MAE	R^2
10%	CORD (Scratch)	0.1169 ± 0.0059	0.0959 ± 0.0048	0.8074 ± 0.0191
10%	CORD (Single-domain) Frozen	0.1245 ± 0.0162	0.1044 ± 0.0155	0.7793 ± 0.0592
10%	CORD (Multi-domain) Frozen	**0.0864 ± 0.0059**	**0.0671 ± 0.0060**	**0.8948 ± 0.0144**
10%	CORD (Single-domain) Partial FT	0.1194 ± 0.0105	0.0996 ± 0.0088	0.7986 ± 0.0348
10%	CORD (Multi-domain) Partial FT	**0.0791 ± 0.0116**	**0.0668 ± 0.0107**	**0.9105 ± 0.0267**
10%	CORD (Single-domain) Full FT	0.1082 ± 0.0110	0.0924 ± 0.0107	0.8340 ± 0.0338
10%	CORD (Multi-domain) Full FT	0.0871 ± 0.0245	0.0761 ± 0.0206	0.8867 ± 0.0638
20%	CORD (Scratch)	0.1007 ± 0.0113	0.0805 ± 0.0114	0.8561 ± 0.0318
20%	CORD (Single-domain) Frozen	0.1131 ± 0.0160	0.0937 ± 0.0150	0.8173 ± 0.0532
20%	CORD (Multi-domain) Frozen	0.0782 ± 0.0080	0.0607 ± 0.0074	0.9132 ± 0.0177
20%	CORD (Single-domain) Partial FT	0.1088 ± 0.0106	0.0903 ± 0.0109	0.8323 ± 0.0322
20%	CORD (Multi-domain) Partial FT	**0.0660 ± 0.0049**	**0.0514 ± 0.0027**	**0.9385 ± 0.0092**
20%	CORD (Single-domain) Full FT	0.0930 ± 0.0023	0.0782 ± 0.0046	0.8785 ± 0.0060
20%	CORD (Multi-domain) Full FT	**0.0609 ± 0.0096**	**0.0529 ± 0.0091**	**0.9469 ± 0.0152**
100%	CORD (Scratch)	0.0782 ± 0.0247	0.0608 ± 0.0197	0.9073 ± 0.0530
100%	CORD (Single-domain) Frozen	0.1133 ± 0.0056	0.0911 ± 0.0060	0.8191 ± 0.0180
100%	CORD (Multi-domain) Frozen	0.0658 ± 0.0109	0.0516 ± 0.0102	0.9378 ± 0.0186
100%	CORD (Single-domain) Partial FT	0.0920 ± 0.0077	0.0745 ± 0.0065	0.8805 ± 0.0199
100%	CORD (Multi-domain) Partial FT	**0.0491 ± 0.0079**	**0.0413 ± 0.0077**	**0.9654 ± 0.0111**
100%	CORD (Single-domain) Full FT	0.0800 ± 0.0216	0.0671 ± 0.0199	0.9048 ± 0.0436
100%	CORD (Multi-domain) Full FT	**0.0530 ± 0.0165**	**0.0436 ± 0.0136**	**0.9575 ± 0.0250**

C.4 RUL PREDICTION BASELINES

The baselines use model-appropriate observations and scalar RUL heads. MOMENT is evaluated with a frozen encoder; the remaining baselines follow their model-appropriate input processing and readout configurations.

C.4.1 BEARING

Table 12: Bearing RUL prediction comparison at 10% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	0.1463 ± 0.0393	0.1172 ± 0.0292	0.7428 ± 0.1406
CORD (Single-domain)	**0.1121 ± 0.0224**	**0.0833 ± 0.0142**	**0.8526 ± 0.0623**
CORD (Multi-domain)	**0.1240 ± 0.0289**	**0.0979 ± 0.0239**	**0.8177 ± 0.0859**
MLP	0.2185 ± 0.0179	0.1992 ± 0.0149	0.4543 ± 0.0919
Random Forest	0.1840 ± 0.0035	0.1698 ± 0.0032	0.6148 ± 0.0145
XGBoost	0.1494 ± 0.0039	0.1361 ± 0.0054	0.7462 ± 0.0131
TCN	0.2681 ± 0.0286	0.2186 ± 0.0175	0.1757 ± 0.1767
PatchTST	0.3019 ± 0.0367	0.2547 ± 0.0238	−0.0482 ± 0.2475
iTransformer	0.3138 ± 0.0207	0.2513 ± 0.0184	−0.1232 ± 0.1454
MOMENT	0.3070 ± 0.0018	0.2696 ± 0.0021	−0.0714 ± 0.0128

Table 13: Bearing RUL prediction comparison at 20% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	**0.1400 ± 0.0145**	**0.1081 ± 0.0112**	**0.7752 ± 0.0453**
CORD (Single-domain)	0.1464 ± 0.0142	0.1100 ± 0.0132	0.7546 ± 0.0474
CORD (Multi-domain)	**0.1329 ± 0.0214**	**0.1044 ± 0.0142**	**0.7951 ± 0.0697**
MLP	0.2307 ± 0.0124	0.2093 ± 0.0106	0.3937 ± 0.0644
Random Forest	0.1910 ± 0.0022	0.1664 ± 0.0017	0.5853 ± 0.0096
XGBoost	0.1824 ± 0.0043	0.1548 ± 0.0060	0.6218 ± 0.0176
TCN	0.2930 ± 0.0477	0.2323 ± 0.0324	0.0035 ± 0.3391
PatchTST	0.3338 ± 0.0080	0.2736 ± 0.0078	−0.2670 ± 0.0607
iTransformer	0.3157 ± 0.0280	0.2611 ± 0.0314	−0.1404 ± 0.2036
MOMENT	0.3176 ± 0.0017	0.2761 ± 0.0014	−0.1471 ± 0.0120

Table 14: Bearing RUL prediction comparison at 100% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	0.1122 ± 0.0233	0.0854 ± 0.0143	0.8519 ± 0.0637
CORD (Single-domain)	**0.0988 ± 0.0184**	**0.0718 ± 0.0162**	**0.8859 ± 0.0436**
CORD (Multi-domain)	**0.0972 ± 0.0071**	**0.0744 ± 0.0076**	**0.8921 ± 0.0160**
MLP	0.2160 ± 0.0114	0.1922 ± 0.0100	0.4685 ± 0.0553
Random Forest	0.1853 ± 0.0009	0.1570 ± 0.0009	0.6095 ± 0.0040
XGBoost	0.1846 ± 0.0031	0.1619 ± 0.0032	0.6125 ± 0.0130
TCN	0.2287 ± 0.0073	0.1893 ± 0.0058	0.4049 ± 0.0389
PatchTST	0.2663 ± 0.0243	0.2230 ± 0.0193	0.1882 ± 0.1497
iTransformer	0.3071 ± 0.0281	0.2480 ± 0.0200	−0.0796 ± 0.1996
MOMENT	0.2711 ± 0.0023	0.2238 ± 0.0020	0.1646 ± 0.0142

C.4.2 BATTERY

Table 15: Battery RUL prediction comparison at 10% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	0.0711 ± 0.0091	0.0516 ± 0.0065	0.9384 ± 0.0146
CORD (Single-domain)	0.0689 ± 0.0048	**0.0492 ± 0.0035**	0.9426 ± 0.0078
CORD (Multi-domain)	**0.0673 ± 0.0064**	**0.0487 ± 0.0044**	**0.9450 ± 0.0105**
MLP	0.1526 ± 0.0130	0.1026 ± 0.0071	0.7178 ± 0.0469
Random Forest	0.1085 ± 0.0006	0.0723 ± 0.0003	0.8581 ± 0.0015
XGBoost	0.1036 ± 0.0030	0.0716 ± 0.0021	0.8705 ± 0.0074
TCN	0.0807 ± 0.0068	0.0623 ± 0.0062	0.9210 ± 0.0129
PatchTST	**0.0676 ± 0.0080**	0.0535 ± 0.0070	**0.9443 ± 0.0123**
iTransformer	0.0904 ± 0.0018	0.0684 ± 0.0011	0.9015 ± 0.0039
MOMENT	0.0730 ± 0.0021	0.0594 ± 0.0017	0.9358 ± 0.0038

Table 16: Battery RUL prediction comparison at 20% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	**0.0680 ± 0.0060**	**0.0497 ± 0.0056**	**0.9439 ± 0.0101**
CORD (Single-domain)	0.0735 ± 0.0050	0.0533 ± 0.0044	0.9346 ± 0.0089
CORD (Multi-domain)	**0.0683 ± 0.0043**	**0.0484 ± 0.0035**	**0.9436 ± 0.0072**
MLP	0.1408 ± 0.0081	0.0968 ± 0.0041	0.7605 ± 0.0275
Random Forest	0.1110 ± 0.0011	0.0730 ± 0.0006	0.8516 ± 0.0030
XGBoost	0.1052 ± 0.0009	0.0722 ± 0.0005	0.8668 ± 0.0023
TCN	0.0813 ± 0.0064	0.0634 ± 0.0055	0.9199 ± 0.0124
PatchTST	0.0722 ± 0.0062	0.0548 ± 0.0063	0.9368 ± 0.0105
iTransformer	0.0893 ± 0.0044	0.0685 ± 0.0043	0.9037 ± 0.0096
MOMENT	0.0695 ± 0.0013	0.0556 ± 0.0013	0.9418 ± 0.0022

Table 17: Battery RUL prediction comparison at 100% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	**0.0597 ± 0.0085**	**0.0435 ± 0.0073**	**0.9564 ± 0.0125**
CORD (Single-domain)	0.0652 ± 0.0064	0.0464 ± 0.0054	0.9483 ± 0.0103
CORD (Multi-domain)	0.0632 ± 0.0056	**0.0458 ± 0.0050**	0.9515 ± 0.0086
MLP	0.1005 ± 0.0052	0.0769 ± 0.0032	0.8780 ± 0.0126
Random Forest	0.1054 ± 0.0002	0.0712 ± 0.0001	0.8662 ± 0.0005
XGBoost	0.0992 ± 0.0013	0.0696 ± 0.0011	0.8814 ± 0.0031
TCN	0.0788 ± 0.0053	0.0622 ± 0.0043	0.9250 ± 0.0101
PatchTST	0.0792 ± 0.0049	0.0592 ± 0.0041	0.9242 ± 0.0094
iTransformer	0.0813 ± 0.0008	0.0615 ± 0.0008	0.9203 ± 0.0016
MOMENT	**0.0620 ± 0.0028**	0.0485 ± 0.0028	**0.9536 ± 0.0042**

C.4.3 CUTTING TOOLS

Table 18: Cutting-tool RUL prediction comparison at 10% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	0.1169 ± 0.0059	0.0959 ± 0.0048	0.8074 ± 0.0191
CORD (Single-domain)	0.1082 ± 0.0110	0.0924 ± 0.0107	0.8340 ± 0.0338
CORD (Multi-domain)	**0.0871 ± 0.0245**	**0.0761 ± 0.0206**	**0.8867 ± 0.0638**
MLP	**0.0991 ± 0.0161**	**0.0792 ± 0.0152**	**0.8591 ± 0.0475**
Random Forest	0.1117 ± 0.0034	0.0922 ± 0.0045	0.8244 ± 0.0107
XGBoost	0.1258 ± 0.0023	0.1021 ± 0.0021	0.7775 ± 0.0081
TCN	0.1409 ± 0.0270	0.1123 ± 0.0221	0.7128 ± 0.1080
PatchTST	0.2258 ± 0.0130	0.1833 ± 0.0090	0.2818 ± 0.0828
iTransformer	0.2728 ± 0.0085	0.2268 ± 0.0085	−0.0465 ± 0.0650
MOMENT	0.2452 ± 0.0052	0.2023 ± 0.0039	0.1548 ± 0.0358

Table 19: Cutting-tool RUL prediction comparison at 20% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	0.1007 ± 0.0113	0.0805 ± 0.0114	0.8561 ± 0.0318
CORD (Single-domain)	**0.0930 ± 0.0023**	**0.0782 ± 0.0046**	**0.8785 ± 0.0060**
CORD (Multi-domain)	**0.0609 ± 0.0096**	**0.0529 ± 0.0091**	**0.9469 ± 0.0152**
MLP	0.1085 ± 0.0132	0.0862 ± 0.0104	0.8325 ± 0.0414
Random Forest	0.1202 ± 0.0024	0.0999 ± 0.0022	0.7968 ± 0.0079
XGBoost	0.1204 ± 0.0012	0.1010 ± 0.0018	0.7962 ± 0.0040
TCN	0.1266 ± 0.0117	0.1017 ± 0.0090	0.7730 ± 0.0413
PatchTST	0.2103 ± 0.0086	0.1733 ± 0.0057	0.3775 ± 0.0507
iTransformer	0.2582 ± 0.0085	0.2137 ± 0.0080	0.0626 ± 0.0629
MOMENT	0.2431 ± 0.0080	0.1989 ± 0.0055	0.1689 ± 0.0552

Table 20: Cutting-tool RUL prediction comparison at 100% labels (five seeds).

Method	RMSE	MAE	R^2
CORD (Scratch)	0.0782 ± 0.0247	0.0608 ± 0.0197	0.9073 ± 0.0530
CORD (Single-domain)	0.0800 ± 0.0216	0.0671 ± 0.0199	0.9048 ± 0.0436
CORD (Multi-domain)	**0.0530 ± 0.0165**	**0.0436 ± 0.0136**	**0.9575 ± 0.0250**
MLP	0.1177 ± 0.0128	0.0794 ± 0.0077	0.8034 ± 0.0449
Random Forest	0.1230 ± 0.0014	0.1014 ± 0.0022	0.7874 ± 0.0050
XGBoost	0.1214 ± 0.0016	0.0992 ± 0.0011	0.7927 ± 0.0056
TCN	**0.0722 ± 0.0070**	**0.0581 ± 0.0057**	**0.9261 ± 0.0144**
PatchTST	0.1810 ± 0.0158	0.1485 ± 0.0103	0.5368 ± 0.0806
iTransformer	0.1890 ± 0.0038	0.1657 ± 0.0038	0.4977 ± 0.0204
MOMENT	0.2421 ± 0.0022	0.2121 ± 0.0036	0.1763 ± 0.0151

D GENERALIZATION UNDER PRETRAINING-EXCLUDED SYSTEM TYPES: ENGINE ADAPTATION

The engine comparison isolates initialization. Both conditions use the same observation-interface construction, engine interface, seven-flight context, MSE objective, optimizer, label-budget samples, and 400 supervised optimizer updates. Scratch starts all parameters randomly; Source-pretrained initializes the reusable encoder from the selected bearing–battery–cutting-tool checkpoint and adapts all trainable components. The 400-update adaptation budget is selected by grouped cross-validation on U2/U5/U10/U16/U18/U20 and then fixed for all final U11 runs. Tables report mean ± sample SD for seeds 42–46. Within each label budget, **red bold** marks the better mean and **blue bold underlined** marks the second-best mean.

Table 21: N-CMAPSS U11, 10% labels: matched Scratch and source-pretrained initialization (five seeds).

Initialization	Updates	RMSE	MAE	R^2
CORD (Scratch)	400	**0.1463 ± 0.0134**	**0.1151 ± 0.0103**	**0.6518 ± 0.0611**
CORD (Source-pretrained)	400	**0.1374 ± 0.0117**	**0.1098 ± 0.0091**	**0.6932 ± 0.0510**

Table 22: N-CMAPSS U11, 20% labels: matched Scratch and source-pretrained initialization (five seeds).

Initialization	Updates	RMSE	MAE	R^2
CORD (Scratch)	400	**0.1163 ± 0.0131**	**0.0903 ± 0.0090**	**0.7795 ± 0.0482**
CORD (Source-pretrained)	400	**0.1122 ± 0.0069**	**0.0883 ± 0.0070**	**0.7959 ± 0.0252**

Table 23: N-CMAPSS U11, 100% labels: matched Scratch and source-pretrained initialization (five seeds).

Initialization	Updates	RMSE	MAE	R^2
CORD (Scratch)	400	**0.0768 ± 0.0154**	**0.0566 ± 0.0112**	**0.9018 ± 0.0390**
CORD (Source-pretrained)	400	**0.0841 ± 0.0099**	**0.0582 ± 0.0055**	**0.8844 ± 0.0274**

Table 24 isolates IDM under the matched protocol of Section 5.4; architecture and downstream adaptation are fixed.

Table 24: IDM pretraining ablation on N-CMAPSS U11 (five seeds; 400 adaptation updates).

Labels	Pretraining objective	RMSE ↓	MAE ↓	R^2 ↑
10%	ISM only	0.17251 ± 0.01562	0.14087 ± 0.01404	0.51609 ± 0.08495
10%	ISM + IDM	0.13742 ± 0.01169	0.10979 ± 0.00914	0.69319 ± 0.05103
20%	ISM only	0.14477 ± 0.00588	0.11872 ± 0.00390	0.66099 ± 0.02775
20%	ISM + IDM	0.11225 ± 0.00686	0.08833 ± 0.00698	0.79586 ± 0.02515
100%	ISM only	0.09439 ± 0.01887	0.07094 ± 0.01156	0.85148 ± 0.06445
100%	ISM + IDM	0.08415 ± 0.00987	0.05819 ± 0.00546	0.88436 ± 0.02740

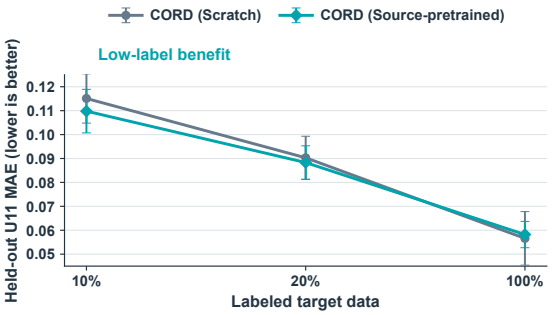

Figure 7: Engine MAE under the same matched 400-update protocol as Figure 4. Points and error bars show mean ± one sample SD over five seeds.

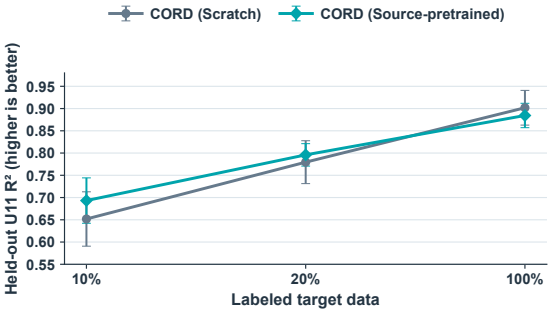

Figure 8: Engine R^2 under the same matched 400-update protocol as Figure 4. Points and error bars show mean ± sample SD over five seeds.

E Frozen-Representation Reuse

Frozen-encoder analysis uses one selected checkpoint per condition. Held-out unit queries retrieve five nearest reference-unit states, and normalized-RUL discrepancy measures lifecycle-neighborhood quality. Bearings and batteries use the 96-dimensional projector and cutting tools the 192-dimensional final-layer normalized hidden representation.

Table 25: Cross-unit 5-NN normalized-RUL error on held-out units. Lower is better. Within each system type, **red bold** marks the best encoder and **blue bold underlined** the second-best. Multi-domain pretraining improves over Single-domain for all three system types.

System type	Encoder	5-NN normalized-RUL error
Bearings	CORD (Random init.)	0.2585
Bearings	CORD (Single-domain)	**0.1407**
Bearings	CORD (Multi-domain)	**0.1212**
Batteries	CORD (Random init.)	0.0793
Batteries	CORD (Single-domain)	**0.0736**
Batteries	CORD (Multi-domain)	**0.0604**
Cutting tools	CORD (Random init.)	**0.1054**
Cutting tools	CORD (Single-domain)	0.1376
Cutting tools	CORD (Multi-domain)	**0.1000**

Figure 6 visualizes lifecycle organization within the same selected frozen encoder conditions.

F Observation Interface: Structured Descriptors versus Raw-Resampled Inputs

This scratch-only comparison uses the same FP32 downstream split and fitting protocol at 10%, 20%, and 100% labels, with seeds 42–46. The Structured-Descriptor condition uses the same Scratch results reported above, whereas the Raw-Resampled condition replaces descriptor extraction with parameter-free resampling of native signal windows to 26 values while retaining the same downstream split and training protocol. Both use 65-token, width-96, two-layer encoders and the corresponding type-specific RUL readouts. The displayed uncertainty is sample SD over downstream seeds on one fixed test unit per system type. In all nine settings, Structured Descriptors are better on all three metrics; they also win all 45 matched-seed comparisons for each metric. The advantage of Structured Descriptors persists at the 100% label budget.

Table 26: Structured Descriptors versus Raw-Resampled Inputs under the matched Scratch protocol: mean ± SD over five seeds. Red bold marks the better input within each system type and label budget; blue underlined marks the other.

System type	Labels	Input representation	RMSE ↓	MAE ↓	R^2 ↑
Bearings	10%	Structured Descriptors	**0.1463 ± 0.0393**	**0.1172 ± 0.0292**	**0.7428 ± 0.1406**
		Raw-Resampled Inputs	0.3590 ± 0.0237	0.3016 ± 0.0212	−0.4704 ± 0.1947
Bearings	20%	Structured Descriptors	**0.1400 ± 0.0145**	**0.1081 ± 0.0112**	**0.7752 ± 0.0453**
		Raw-Resampled Inputs	0.3157 ± 0.0195	0.2691 ± 0.0131	−0.1362 ± 0.1428
Bearings	100%	Structured Descriptors	**0.1122 ± 0.0233**	**0.0854 ± 0.0143**	**0.8519 ± 0.0637**
		Raw-Resampled Inputs	0.3091 ± 0.0338	0.2619 ± 0.0283	−0.0966 ± 0.2323
Batteries	10%	Structured Descriptors	**0.0711 ± 0.0091**	**0.0516 ± 0.0065**	**0.9384 ± 0.0146**
		Raw-Resampled Inputs	0.2131 ± 0.0112	0.1664 ± 0.0133	0.4519 ± 0.0579
Batteries	20%	Structured Descriptors	**0.0680 ± 0.0060**	**0.0497 ± 0.0056**	**0.9439 ± 0.0101**
		Raw-Resampled Inputs	0.1715 ± 0.0065	0.1213 ± 0.0092	0.6454 ± 0.0269
Batteries	100%	Structured Descriptors	**0.0597 ± 0.0085**	**0.0435 ± 0.0073**	**0.9564 ± 0.0125**
		Raw-Resampled Inputs	0.1724 ± 0.0113	0.1097 ± 0.0070	0.6408 ± 0.0463
Cutting tools	10%	Structured Descriptors	**0.1169 ± 0.0059**	**0.0959 ± 0.0048**	**0.8074 ± 0.0191**
		Raw-Resampled Inputs	0.2845 ± 0.0223	0.2363 ± 0.0152	−0.1435 ± 0.1828
Cutting tools	20%	Structured Descriptors	**0.1007 ± 0.0113**	**0.0805 ± 0.0114**	**0.8561 ± 0.0318**
		Raw-Resampled Inputs	0.2178 ± 0.0421	0.1755 ± 0.0318	0.3132 ± 0.2656
Cutting tools	100%	Structured Descriptors	**0.0782 ± 0.0247**	**0.0608 ± 0.0197**	**0.9073 ± 0.0530**
		Raw-Resampled Inputs	0.1894 ± 0.0254	0.1531 ± 0.0235	0.4885 ± 0.1421

The Structured-Descriptor condition uses the proposed local and global observation descriptors. Bearings and cutting tools use 26 statistical and spectral coordinates: mean, absolute mean, standard deviation, variance, RMS, energy, peak, peak-to-peak range, minimum, maximum, skewness, kurtosis, crest, shape, impulse, and clearance factors, root amplitude, zero-crossing rate, spectral

centroid, spectral bandwidth, spectral entropy, dominant frequency, low-, mid-, and high-band energy, and the high-to-low-band energy ratio. Batteries use voltage mean, standard deviation, minimum, maximum, range, skewness, kurtosis, slope, mean absolute derivative, and mean absolute curvature; C-rate mean, standard deviation, minimum, maximum, mean absolute value, and slope; temperature mean, standard deviation, minimum, maximum, change, and slope; local duration, capacity, and energy; and mean dV/dQ. Global descriptors use the corresponding coordinates over the complete observation extent, with full-cycle duration and capacity used for batteries when available.

G Computational Cost and Shared Deployment

G.1 Frozen shared deployment

Table 27: Parameter storage for one frozen shared encoder with three task heads versus three separate extracted encoders.

Deployment	Parameters	FP32 MiB	Relative
One frozen shared encoder + three heads	657,235	2.507	35.75% reduction
Three extracted encoders + three heads	1,022,899	3.902	Reference

The parameter reduction applies when one frozen encoding system serves three specialized heads. Independently fine-tuned encoders diverge and require separate copies. The 35.75% reduction refers to parameter storage under frozen shared deployment; latency is reported separately in Table 28.

G.2 A100 benchmark

The A100-SXM4-40GB benchmark uses BF16 autocast with FP32 parameters, resident model/input tensors, five warm-up calls, and five measurement rounds. These rounds are timing repetitions, not accuracy seeds. CPU tree-model measurements are excluded from the GPU comparison. Raw-signal preprocessing and total source-pretraining GPU-hours are not included. These costs have different directions and scopes, so no single best/second-best ranking is assigned to the efficiency table. Table 28 reports latency, throughput, memory, and update time as separate efficiency dimensions.

Table 28: A100 inference and training efficiency by system type and predictor. B1 denotes batch-one latency; B32 denotes batch-32 throughput and peak memory.

System type	Model	Ch.	Parameters	Trainable	B1 ms	B32 samples/s	B32 MiB	Update s
Bearings	CORD Full	2	279,209	279,209	6.275	3,855	73	0.338
Bearings	CORD Partial	2	279,209	161,145	6.332	3,821	73	0.261
Bearings	CORD Frozen	2	279,209	81,249	6.349	3,828	73	0.198
Bearings	TCN	2	85,505	85,505	0.779	41,662	59	0.052
Bearings	PatchTST	2	231,649	231,649	2.804	11,029	38	0.166
Bearings	iTransformer	2	245,953	245,953	2.876	10,643	38	0.172
Bearings	MOMENT	2	109,656,961	21,505	19.554	1,648	634	0.350
Batteries	CORD Full	1	376,137	376,137	6.614	4,702	160	0.356
Batteries	CORD Partial	1	376,137	258,073	6.699	4,623	160	0.271
Batteries	CORD Frozen	1	376,137	178,177	6.815	4,568	160	0.210
Batteries	TCN	1	85,057	85,057	0.761	42,697	118	0.053
Batteries	PatchTST	1	234,529	234,529	2.738	10,891	41	0.168
Batteries	iTransformer	1	289,153	289,153	2.840	10,856	39	0.168
Batteries	MOMENT	1	109,658,497	23,041	19.275	1,637	557	0.354
Cutting tools	CORD Full	7	367,553	367,553	5.627	4,990	224	0.291
Cutting tools	CORD Partial	7	367,553	249,489	5.588	4,914	223	0.211
Cutting tools	CORD Frozen	7	367,553	169,593	5.919	4,853	223	0.168
Cutting tools	TCN	7	88,385	88,385	0.766	34,605	143	0.052
Cutting tools	PatchTST	7	234,337	234,337	2.983	9,088	91	0.171
Cutting tools	iTransformer	7	288,961	288,961	2.902	9,848	92	0.172
Cutting tools	MOMENT	7	109,656,961	21,505	19.484	537	1068	0.351