FactoryNet: A Large-Scale Dataset toward Industrial Time-Series Foundation Models

Karim Othman [*1,2] **Jonas Petersen** [*2,3] **Matei Ignuta-Ciuncanu** [4,5] **Camilla Mazzoleni** [2]
Federico Martelli [2,3] **Alessandro Lombardi** [2] **Riccardo Maggioni** [2] **Philipp Petersen** [6]

github.com/Forgis-Labs/FactoryNet huggingface.co/datasets/Forgis/FactoryNet

arXiv:2605.09081v4 [cs.LG] 3 Jun 2026

Abstract

We introduce the first universal pretraining corpus for industrial time-series data: **FactoryNet**. 51M datapoints across 23k end-to-end task executions (13.3k real, 9.8k synthetic) on six embodiments, unified by a shared schema that enables robust zero-shot cross-embodiment transfer and highly parameter-efficient anomaly detection. We introduce a novel schema: Setpoint, Effort, Feedback, Context (S-E-F-C) underlying the whole pipeline that maps any actuated system into a common representational frame. The corpus spans 27 annotated anomaly types alongside healthy baselines and counterfactual pairs across robotic manipulation and machining domains. Cross-embodiment transfer experiments yield positive results: under bias-aware metrics our model demonstrates fair cross-embodiment transfer capabilities on the evaluated source-target pair, while 24 schema-aligned signals achieves competitive anomaly detection performance compared to high-dimensional baselines. We release FactoryNet as a growing, multi-embodiment dataset to drive progress toward industrial foundation models.

1. Introduction

The manufacturing sector accounts for approximately 15% of global GDP, relying heavily on the continuous operation of complex actuated machinery (World Bank, 2026). While predictive maintenance and process optimization present significant opportunities for machine learning, industrial AI remains largely confined to single-machine, bespoke deployments. Foundation models have transformed vision and language by pretraining on large, structurally coherent corpora (Brown et al., 2020; Dosovitskiy et al., 2021), yet no analogous substrate exists for industrial time-series. The gap is not merely volume: existing anomaly detection and forecasting datasets (Brockmann et al., 2023; Leporowski et al., 2021) record sensor outcomes without separating *commanded intent* from *measured response*. For actuated systems, learning transferable dynamics requires observing the full control loop from target trajectory through actuation effort to the resulting physical state. While partial solutions exist for specific machines (as discussed in Section 2), no unified open dataset provides this explicit decomposition across multiple embodiments.

Every signal is mapped into the Setpoint-Effort-Feedback-Context (S-E-F-C) schema: a machine-agnostic signal taxonomy grounded in IEC 81346 functional classification that separates intent from outcome across arbitrary actuated systems. For physics-oriented modeling, S–E–F–C plays the role of a compact, structured interface for short rollouts: Setpoint and Context fix commanded intent and boundary conditions, while Effort and Feedback make *commanded versus realized* dynamics comparable as explicit prediction residuals rather than opaque reconstruction scores. Paired real and Isaac episodes under the same schema let sim-to-real mismatch be read as *forward-model error* under matched inputs, in line with the bias-aware transfer and phase-aware gap analyses.

2. Related Work

2.1. Industrial Fault-Detection Datasets

The availability of open-source data in the manufacturing domain lags significantly behind other modalities. Most existing datasets focus on single-machine, run-to-failure scenarios or specific rotating machinery components. Canonical examples include the NASA C-MAPSS turbofan degradation benchmark (Saxena et al., 2008) and recent robotic datasets: voraus-AD (Brockmann et al., 2023) provides 2,122 episodes of industrial robot recordings and AURSAD (Leporowski et al., 2021) offers 2,045 episodes. However, these datasets are bounded in scale and lack an explicit

[*]Equal contribution [1]Cairo University [2]Forgis [3]ETH Zurich [4]Imperial College London [5]UC Berkeley [6]University of Vienna. Correspondence to: Jonas Petersen <jep79@cantab.ac.uk>.

Preprint. June 4, 2026.

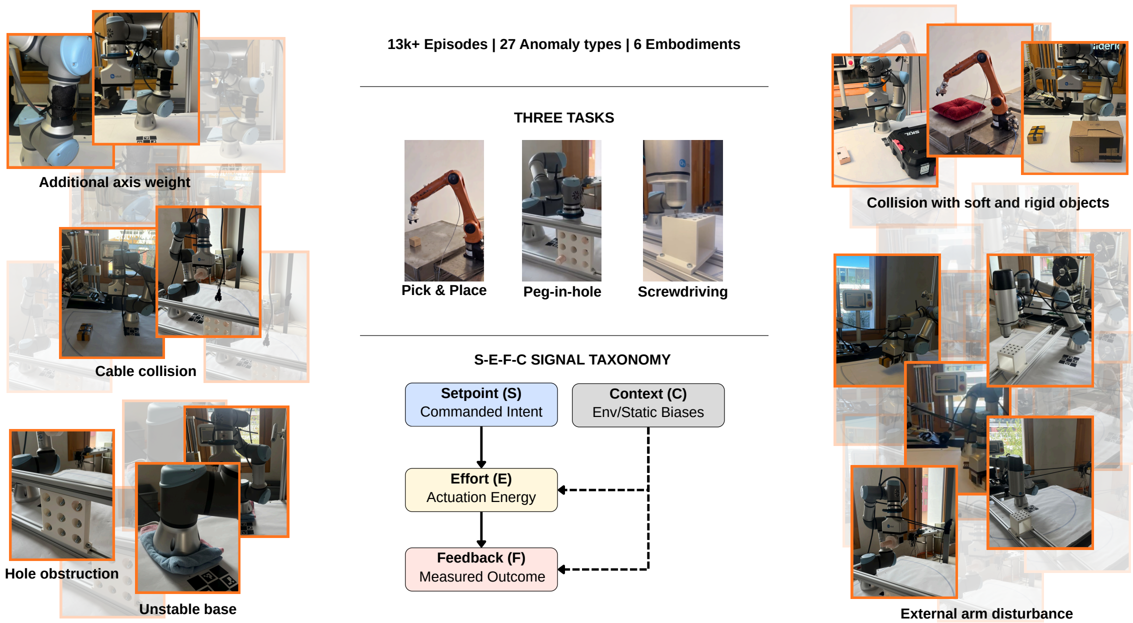

Figure 1. **FactoryNet overview.** A large-scale, multi-embodiment industrial time-series corpus spanning 13 k+ real episodes, 27 anomaly types, and 3 manipulation tasks. Every signal is mapped into the Setpoint-Effort-Feedback-Context (S-E-F-C) taxonomy, a control-theoretic decomposition that enables cross-embodiment learning. Representative fault types are shown alongside the corresponding laboratory setups.

control-loop structure mapping commands to outcomes. By unifying these sources alongside novel laboratory data under a standard control-theoretic decomposition, FactoryNet provides the largest open-source, fault-injected industrial robot dataset to date (see Table 1 for a comprehensive comparison with existing datasets).

Beyond single-component datasets, broader industrial anomaly detection benchmarks such as SKAB (Katser & Kozitsin, 2020), MetroPT (Veloso et al., 2022), WADI (Ahmed et al., 2017) and the Tennessee-Eastman process (Downs & Vogel, 1993) provide system-level multivariate time-series. Similarly, general-purpose time-series anomaly detection benchmarks like TSB-AD (Liu & Paparrizos, 2024) and TODS (Lai et al., 2025) have driven algorithmic progress (e.g., Anomaly-Transformer (Xu et al., 2022)). However, these datasets record holistic system states without the explicit commanded-versus-measured structural decomposition necessary for learning robotic dynamics.

2.2. Foundation Models for Time-Series and Robotics

Current foundation model research diverges into two distinct tracks. In robotics, efforts such as Open X-Embodiment (O'Neill et al., 2024) and DROID (Khazatsky et al., 2024) pool data across diverse kinematics to train generalizable, cross-embodiment behavioral policies. Simultaneously, in the structured time-series domain, models such as Chronos (Ansari et al., 2024), TimesFM (Das et al., 2024), Moirai (Woo et al., 2024), MOMENT (Goswami et al., 2024), Timer (Liu et al., 2024), Lag-Llama (Rasul et al., 2023), and TabPFN-TS (Hoo et al., 2025) leverage pretraining corpora to yield powerful zero-shot forecasters. However, because these TS-FMs are predominantly trained on web-scraped, financial, or ecological data, we hypothesize that they lack grounding in physical actuation and may struggle to disentangle static payload biases from actual dynamic behavior in industrial settings. FactoryNet is designed as a bridge between these two tracks: an industrial-scale, control-theoretically structured corpus for training and evaluating TS-FMs on complex dynamics across embodiments.

2.3. Physics-Informed and Structured Dynamics

In parallel to data-driven forecasting, Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) and grey-box models incorporate known governing equations into the learning process. While these approaches excel in scenarios with well-defined partial differential equations, applying them to complex multi-joint robotic systems often requires residual-dynamics learning (Zeng et al., 2020) or structured priors. FactoryNet complements this literature by providing the empirical Setpoint-Effort-Feedback decomposition required to train and evaluate structured or physics-inspired models at scale.

Table 1. Comparison of open-source industrial time-series datasets. Expanding beyond single-domain recordings, FactoryNet is the only multi-machine corpus requiring a strict control-loop structure decoupling intended Setpoint from applied Effort.

Dataset	Year	Machine Type	Signals	Samples (Eps.)	Has Setpoint?	Has Effort?
CWRU (Li, 2019)	2000	Bearings	Vibration	480	No	No
PHM 2010 (PHM Society, 2010)	2010	CNC milling	Force, vibration	315	Partial	Yes
Paderborn (Lessmeier et al., 2016)	2016	Bearings	Vibration, current	2,000	No (RPM only)	Partial
MAFAULDA (Ribeiro et al., 2016)	2016	Rotating machinery	Vibration, audio	1,951	No	No
XJTU-SY (Wang et al., 2020)	2019	Bearings	Vibration	15	No	No
AURSAD (Leporowski et al., 2021)	2021	UR3e robot	Joint signals	2,045	Yes	Yes
voraus-AD (Brockmann et al., 2023)	2023	Collaborative robot	130 channels	2,122	Yes	Yes
FactoryNet (Ours)	**2026**	**Multi-machine**	**Multimodal**	**23k**	**Yes (Required)**	**Yes (Required)**

3. The FactoryNet Dataset

FactoryNet v1.0 is the largest open-source industrial-robot time-series dataset containing labelled anomalies and organized around a control-theoretic schema. It comprises three pillars: real-world laboratory recordings, standardized open-source adaptations, and a synthetic generation pipeline. Because industrial data is sampled at high frequencies across many joints and sensors, 23k episodes yield a high-density, continuous corpus of the kind required to learn complex physical dynamics.

3.1. Dataset Composition

As detailed in Table 2 and Figure 2, the corpus encompasses multiple data streams. The laboratory track consists of novel recordings we collected from UR3 and KUKA KR10 industrial robotic arms executing three complex tasks: Pick & Place, Screwdriving, and Peg-in-Hole. We ingested and restructured existing high-quality open-source datasets (voraus-AD, AURSAD, and UMich CNC (Sun, 2018)) into our unified schema. The dataset is structured using a hierarchical taxonomy grounded in the IEC 81346 standard for industrial systems, allowing the schema to seamlessly integrate diverse modalities such as CNC milling centres and rotating machinery. A parallel synthetic track generated via NVIDIA Isaac Sim provides procedurally scaled data for model pretraining.

Table 2. FactoryNet Composition. Datapoint counts are approximate.

Source	Machine	Tasks	Faults	Episodes	Datapoints
Our Lab (Real)	UR3	P&P, Screw, Peg	Yes	7,141	18M
Our Lab (Real)	KUKA KR10	P&P	Yes	1,973	4M
Open (Real)	voraus-AD (Yu-Cobot)	P&P	Yes	2,122	16M
Open (Real)	AURSAD (UR3e)	Screw	Yes	2,045	3M
Open (Real)	UMich CNC	Machining	Yes	18	18K
Synthetic	Isaac Sim (UR5)	P&P	No	9,799	10M

3.2. The S-E-F-C Signal Taxonomy: A Unified Vocabulary

The defining feature of FactoryNet is its control-theoretic structure. Most existing benchmarks simply log raw sensor streams, intrinsically tangling the controller's target variables with the machine's actual physical execution. This conflates cause and effect while suffering from vendor-specific naming conventions.

To resolve this, we employ embodiment-specific adapter scripts to programmatically map over 300 heterogeneous data columns into four standardized roles (machine-readable tables: Appendix B). These categories are: **Setpoint (S)** (commanded intent, e.g., target joint positions), **Effort (E)** (actuation energy expended, e.g., motor current/torque), **Feedback (F)** (measured physical outcome, e.g., actual positions), and **Context (C)** (environmental or static variables, e.g., payload mass).

This taxonomy allows a single dataloader to work across UR3, KUKA KR10, and CNC machinery: a 6-DOF rotational arm and a 4-axis CNC gantry expose the same four roles, only with different axis counts and units. By enforcing this taxonomy, S-E-F-C acts as a unified vocabulary for cross-embodiment models, providing a principled inductive bias for learning the difference between expected dynamics and external disturbances. Signal availability varies by embodiment: KUKA KR10 (KSS 8.3) does not expose joint velocities, commanded TCP pose, or TCP force/torque via its RSI interface; these channels are marked absent in the S-E-F-C mapping (Appendix B).

3.3. Synthetic Pipeline and Sim-to-Real

To reduce dependence on real-only data, FactoryNet includes a synthetic pipeline in NVIDIA Isaac Sim with procedural generation and domain randomization (mass, friction, controller gains), yielding 9799 Pick & Place episodes with aligned S-E-F-C metadata and matched healthy twins for controlled fault-deviation analysis. To ensure the synthetic pretraining corpus captures robust physical dynamics and helps bridge the sim-to-real gap, we employ extensive do-

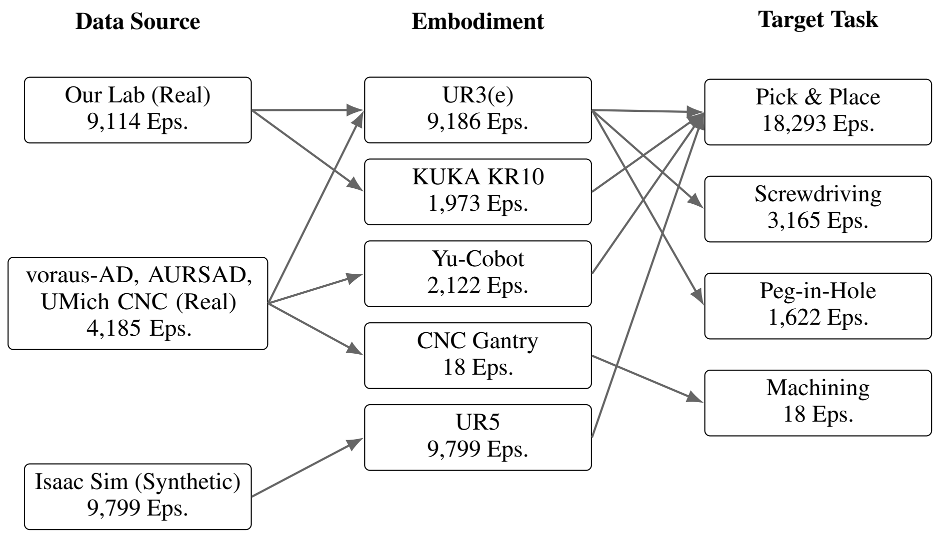

Figure 2. **The FactoryNet Dataset.** A structured overview of the corpus composition illustrating the mapping of 23k task executions. The dataset aggregates real-world laboratory recordings, standardized open-source subsets, and synthetic generations into a unified pretraining substrate covering four distinct actuation tasks.

main randomization across the procedurally generated UR5 Pick & Place episodes in NVIDIA Isaac Sim.

- **Mass Randomization:** The payload (cube) mass is uniformly sampled per episode between 0.10 and 0.30 kg (with broader exploratory configurations allowing up to 0.80 kg). Robot link masses remain fixed to isolate payload-driven dynamic variations and ensure baseline kinematic stability.
- **Surface Friction:** We randomize the general Coulomb friction coefficient of the target object between 0.30 and 0.50 per episode. The end-effector gripper pad maintains a fixed, high-friction coefficient of 1.2 to ensure stable grasps once contact is successfully established.
- **Controller Gains:** To simulate variations in actuation force and mechanical compliance at the end-effector, the proportional gain (K_p) of the gripper is randomized uniformly in the range of $[5000.0, 12000.0]$. The UR5 arm's main joint PID controllers remain fixed to nominal values.
- **Sensor Noise Model:** We inject artificial Gaussian noise into the simulated telemetry to mimic real-world sensor degradation, encoder quantization, and signal noise. Using a base standard deviation of $\sigma_{\text{base}} = 0.002$, noise is scaled across modalities: joint positions ($\sigma = 0.002$ rad), joint velocities ($\sigma = 0.02$ rad/s), and joint efforts/torques ($\sigma = 0.1$). Additionally, spatial perception noise is applied to the object's tracked state ($\sigma_{xy} = 0.002$ m, $\sigma_z = 0.001$ m).
- **Task and Geometric Variation:** The task features procedural geometric and spatial variations to prevent policy overfitting. The target cube's physical dimensions (width, depth, and height) are independently randomized within specified bounds for every episode. Furthermore, the object's initial spawn position on the conveyor is continuously randomized within an 8×8 cm (0.08 m) bounding box relative to the nominal pick center.
- **Simulation Dynamics:** The internal physics simulation engine operates at a fixed temporal step size of $\Delta t \approx 0.016667$ s (60 Hz). To align this synthetic track with the 100 Hz standard utilized by the physical laboratory recordings (see Section 3), the raw 60 Hz simulation telemetry undergoes temporal interpolation during the data ingestion pipeline.

Batch sim-to-real validation (pairing real and simulated episodes under the same RTDE-shaped schema, phase-aware gap metrics) is reported in Section 3.4.

3.4. Batch sim-to-real validation

We run sim2real rollouts in Isaac Sim 4.5.0 (headless Docker, PhysX) using the built-in UR3 USD asset and a position-control stack (`ArticulationAction`) with phase-consistent waypoint replay from real `target_joint_*` signals. The task is UR3 pick-and-place with a 10-phase structure (`above_pick` to `return`). Simulation steps at 240 Hz and logs at 10 Hz. Real episodes are exported from FactoryWave parquet to per-episode RTDE-shaped CSVs, converted to per-episode simulation configs, replayed in Isaac, and paired by episode ID/filename. Simulated logs preserve FactoryWave-compatible fields (`joint_*`, `target_joint_*`, `joint_current_*`, `tcp_*`, `task_phase`, and status/context fields). Gap metrics are computed phase-wise with time normalization/interpolation. Results of the gap analysis are shown in Table 3.

Table 3. Batch sim-to-real gap over 1,155 paired episodes (pooled per-episode metrics).

Metric	Mean	Median	P10	P90
Joint RMSE (deg)	3.65	2.83	1.93	5.12
TCP position RMSE (mm)	13.16	13.26	8.87	16.92
EE L2 RMS (mm)	25.11	25.17	16.96	32.58
TCP rotvec RMSE (mrad)	1877	2605	8.75	3136
W1 effort mean (A)	0.83	0.81	0.77	0.91

An important source of residual pose discrepancy is end-effector mismatch: the Isaac setup used a Robotiq 2FG85-style gripper configuration, whereas the real FactoryWave episodes used an OnRobot 2FG14 gripper. Differences in tool geometry/TCP definition and mounting can bias absolute TCP and orientation metrics. We therefore interpret remaining TCP rotation spread conservatively and treat gripper-accurate tool calibration as future work.

3.5. Faults and Anomalies

To support anomaly detection and robust control research, of the 9,114 lab episodes, approximately 40% are healthy and 60% contain injected faults across 27 anomaly types spanning three tasks: **Pick & Place**, **Screwdriving**, and **Peg-in-Hole**.

3.6. Data Accessibility and Licensing

Novel laboratory and synthetic data are released under the MIT license; adapted open-source subsets retain their original licenses (CC-BY 4.0 or equivalent). The repository provides S-E-F-C Parquet files, metadata, and framework-native dataloaders at `https://huggingface.co/datasets/Forgis/FactoryNet`.

4. Dataset Utility & Validation

To demonstrate that FactoryNet provides a viable substrate for both single-machine modelling and foundation model pretraining, we evaluate the dataset across standard industrial baselines and establish the open challenge of cross-embodiment transfer.

Evaluation Protocol. Evaluation protocols are task-specific. For voraus-AD anomaly detection, we follow the official protocol of Brockmann et al. (2023): training on 948 healthy episodes only, and testing on the 1,174-episode labelled set (419 healthy + 755 anomalous). In contrast, the TCN-Transformer forecasting experiments utilize a separate pre-training split (1,093 training / 137 validation, randomly sampled from all healthy episodes) to maximize observed dynamics. Confidence intervals for our S-E-F-C MLP are 95% bootstrap CIs computed over 1,000 resamplings of episode-level anomaly scores; CIs for unstructured baselines are reported as standard deviation across fault categories as published in Brockmann et al. (2023).

4.1. Model architectures and training

Anomaly Detection: MLP Architecture. The S-E-F-C MLP is a supervised regressor trained to predict motor torque from setpoint signals. **Inputs:** 18 Setpoint signals (`setpoint_pos_0...5`, `setpoint_vel_0...5`, `setpoint_acc_0...5`). **Outputs:** 6 Effort signals (`effort_motor_torque_0...5`). **Anomaly score:** per-episode mean absolute error (MAE) between predicted and true motor torque—higher error indicates anomaly. To maintain parity with standard anomaly detection protocols, the model is trained on healthy episodes only (948 episodes from voraus-AD).

Architecture: The network is constructed with three hidden layers consisting of 512, 256, and 128 units, respectively. We apply the Rectified Linear Unit (ReLU) activation function after each hidden layer. Dropout is not utilized in this architecture.

Training Details: The model is optimized using Adam (`torch.optim.Adam`) with an initial learning rate of 5×10^{-4}, a weight decay (L_2 penalty) of 1×10^{-5}, and a batch size of 4,096. The learning rate is decayed following a cosine annealing schedule. Models are trained for a maximum of 500 epochs, utilizing an early stopping criterion that halts training if the validation loss fails to improve for 30 consecutive epochs.

TCN-Transformer Architecture and Training. The TCN-Transformer comprises a 3-layer dilated Temporal Convolutional Network (kernel size 3) for local feature extraction, followed by a 2-layer Transformer encoder (4 attention heads, hidden dimension 64, feedforward dimension 128) for sequence modelling. The total parameter count is approximately 105,000. Training was conducted using the AdamW optimizer (learning rate 1×10^{-4}) for 100 epochs. *(Note: Due to the reduced parameter count, training is highly efficient on standard hardware.)*

The model predicts joint acceleration; Euler integration ($\Delta t = 0.01$ s) yields position and velocity. Survival steps are computed as the first step at which per-joint position error exceeds 0.01 rad, averaged across all test episodes and joints.

For the anomaly detection evaluation (Table 4), all unstructured baselines (1-NN, PCA, GANF, CAE, LSTM-VAE, HMM, and MVT-Flow) utilize the exact architectures and hyperparameters established in the original voraus-AD benchmark (Brockmann et al., 2023).

For the multi-step forecasting and zero-shot transfer evaluations (Tables 15 and 6), we evaluate our model against four baseline forward-dynamics predictors. All trainable baselines utilize the identical 10-step context window to predict 1-step-ahead joint accelerations, which are subsequently integrated.

Training Details: All trainable forecasting baselines (Linear, Flat MLP, TCN) were trained on the exact same 1,093-episode pretraining split as the main TCN-Transformer model. They were trained to minimize Mean Squared Error (MSE) on the predicted joint accelerations using the Adam optimizer.

- **Linear Baseline:** A standard linear regression model consisting of a single `nn.Linear` layer that maps the flattened context window (10 steps $\times$ 36 features $=$ 360 inputs) directly to the target acceleration space.
- **Flat MLP:** A multi-layer perceptron utilizing two hidden layers (128 units and 64 units, respectively) with ReLU activation functions, mapping the flattened context window to the target predictions.
- **TCN Baseline:** A 2-layer Temporal Convolutional Network (`Conv1d`) utilizing a kernel size of 3 and a hidden dimension of 64. This serves as a representative pre-2023 benchmark for sequence modeling on industrial control data.
- **Kinematic Baseline (Zero-Predictor):** A non-learned, naive physics baseline that constantly predicts zero acceleration. It assumes the robot maintains constant velocity from the final observation step, generating its trajectory purely through the kinematic integrator.

4.2. Single-Machine Baselines: Validating the S-E-F-C Schema

Anomaly Detection. We use voraus-AD to test whether S-E-F-C enables competitive anomaly detection via supervised dynamics rather than holistic reconstruction. Unstructured baselines reconstruct all 130 channels end-to-end; our **S-E-F-C MLP** is a regressor on 24 signals, mapping 18 Setpoints (`setpoint_pos_0...5`, `setpoint_vel_0...5`, `setpoint_acc_0...5`) to 6 Efforts (`effort_motor_torque_0...5`). Per-episode MAE on Effort is the anomaly score. On 24 signals alone it reaches **83.2%** mean AUROC. Table 4 shows it beats weaker full-channel baselines (1-NN, GANF, PCA) but not the strongest ones (CAE, LSTM-VAE, MVT-Flow). Architectures follow Brockmann et al. (2023).

Table 4. Mean AUROC on voraus-AD. Baselines: 130 channels; ours: 24 (Setpoint→Effort). Values from Brockmann et al. (2023). ‡Std across 12 fault categories.

Method	Input Channels	Mean AUROC
MLP - All signals (Ours)	130	$69.1 \pm 3.1^{\ddagger}$
1-NN	130	77.5
GANF	130	$79.9 \pm 12.7^{\ddagger}$
PCA	130	80.0
S-E-F-C MLP (Ours)	**24**	$\mathbf{83.2 \pm 9.6}^{\ddagger}$
CAE	130	$85.2 \pm 9.2^{\ddagger}$
LSTM-VAE	130	$86.7 \pm 10.1^{\ddagger}$
HMM	130	$87.4 \pm 5.8^{\ddagger}$
MVT-Flow	130	$93.6 \pm 5.7^{\ddagger}$

Table 5 reports per-category AUROC on the voraus-AD subset for all seven methods compared in the main paper. The S-E-F-C MLP achieves the highest AUROC on mechanically distinctive faults (Miscommutation: 99.2, Additional Axis Weight: 95.8) where sustained Effort–Feedback divergence provides a strong discriminative signal, but struggles on transient or subtle gripping failures (Collision w/ Cables: 67.6, Losing Can: 71.8) where the anomaly window is brief and the single-step MLP lacks temporal modelling capacity.

Multi-Step Forecasting. To demonstrate support for high-fidelity dynamics modelling, we evaluate an autoregressive TCN-Transformer (105k parameters) on the voraus-AD Pick & Place data. Operating strictly on S-E-F-C inputs, the model acts as a forward-dynamics predictor: it forecasts 1-step-ahead joint accelerations via a 10-step context window, which are integrated (Euler, $\Delta t = 0.01$ s) to derive position and velocity. Models are trained on 1,093 normal episodes and validated on 137 held-out normal episodes. Full architecture and training hyperparameters are detailed in Section 4.1.

As shown in Figure 3 and Table 15 (Appendix D), the TCN-Transformer achieves an average of **156.7 steps** (78.4% of the 200-step horizon at 100 Hz) without exceeding a strict 0.01 rad per-joint position error threshold, substantially outperforming all baselines. At 200 steps the TCN-Transformer's MSE (0.11×10^{-4} rad^2) is more than four orders of magnitude below the next best baseline.

Table 5. Per-category anomaly detection AUROC on voraus-AD. Unstructured baselines use 130 signals; S-E-F-C MLP uses only 24 signals. N = number of anomalous episodes per category. † 95% bootstrap CI over episodes for S-E-F-C MLP; ‡ standard deviation across fault categories for other methods (from Brockmann et al. (2023)).

Fault Category	N	**1-NN**	**PCA**	**GANF**	**CAE**	**LSTM-VAE**	**HMM**	**MVT-Flow**	**S-E-F-C MLP (Ours)**
Additional friction	144	74.8	76.4	$88.5_{\pm 4.5}$	$89.4_{\pm 0.2}$	$88.7_{\pm 1.5}$	$88.0_{\pm 0.7}$	$96.6_{\pm 0.6}$	80.0
Miscommutation	89	80.8	87.0	$98.8_{\pm 1.1}$	$99.1_{\pm 0.0}$	$98.1_{\pm 0.5}$	$93.7_{\pm 1.3}$	$99.8_{\pm 0.3}$	99.2
Misgrip can	11	100.0	100.0	$47.6_{\pm 13.1}$	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$71.0_{\pm 3.3}$	$95.3_{\pm 3.3}$	83.8
Losing can	74	68.7	70.1	$72.1_{\pm 5.8}$	$72.6_{\pm 0.3}$	$70.4_{\pm 2.9}$	$88.8_{\pm 1.0}$	$96.2_{\pm 0.4}$	71.8
Add. axis weight	156	75.0	79.2	$93.2_{\pm 2.4}$	$93.5_{\pm 0.1}$	$82.7_{\pm 1.1}$	$89.6_{\pm 1.2}$	$94.1_{\pm 0.7}$	**95.8**
Collision w/ foam	72	69.6	73.9	$81.2_{\pm 6.2}$	$81.5_{\pm 0.2}$	$81.5_{\pm 2.1}$	$89.8_{\pm 1.5}$	$87.5_{\pm 1.2}$	74.7
Collision w/ cables	48	74.5	75.7	$82.7_{\pm 6.4}$	$79.6_{\pm 0.3}$	$77.3_{\pm 3.6}$	$91.8_{\pm 1.4}$	$84.7_{\pm 1.2}$	67.6
Collision w/ cardboard	22	82.8	83.6	$77.5_{\pm 5.8}$	$78.6_{\pm 0.3}$	$82.8_{\pm 4.8}$	$86.3_{\pm 1.2}$	$88.3_{\pm 1.2}$	76.7
Varying can weight	80	63.7	64.2	$68.9_{\pm 8.7}$	$72.3_{\pm 0.3}$	$71.4_{\pm 2.1}$	$90.9_{\pm 2.0}$	$85.1_{\pm 1.1}$	76.2
Cable at robot	10	63.0	71.6	$76.6_{\pm 8.1}$	$83.1_{\pm 0.3}$	$96.0_{\pm 1.4}$	$84.4_{\pm 1.1}$	$100.0_{\pm 0.0}$	91.3
Invalid gripping pos.	12	93.4	92.1	$86.6_{\pm 10.8}$	$88.8_{\pm 0.3}$	$97.7_{\pm 1.3}$	$91.8_{\pm 1.4}$	$100.0_{\pm 0.0}$	89.1
Unstable platform	37	83.6	85.9	$84.6_{\pm 3.8}$	$83.9_{\pm 0.2}$	$93.6_{\pm 1.9}$	$82.5_{\pm 1.2}$	$96.1_{\pm 0.7}$	92.8
Mean	755	77.5	80.0	$79.9_{\pm 12.7}{}^{\ddagger}$	$85.2_{\pm 9.2}{}^{\ddagger}$	$86.7_{\pm 10.1}{}^{\ddagger}$	$87.4_{\pm 5.8}{}^{\ddagger}$	$93.6_{\pm 5.7}{}^{\ddagger}$	**83.2 [81.5, 86.1]**†

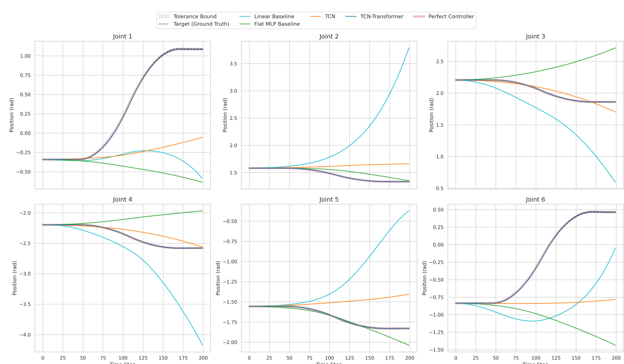

Figure 3. Multi-step forecasting on the voraus-AD (Yu-Cobot) Pick & Place task. The TCN-Transformer maintains predictive accuracy within the 0.01 rad threshold for an average of 156.7 steps (78.4% of the 200-step horizon at 100 Hz), substantially outperforming all baselines on in-domain dynamics prediction.

4.3. Cross-Embodiment Transfer: An Open Challenge

The S-E-F-C schema enables structured zero-shot transfer across machine types. To evaluate this, we define the *mean-centered MAE* (MC-MAE) metric. By subtracting the per-episode, per-joint mean from both the ground truth and predictions prior to computing the absolute error, MC-MAE explicitly isolates dynamic forces from static payload biases.

A TCN-Transformer trained solely on voraus-AD (Yu-Cobot) Pick & Place data achieves a mean-centered MAE of **0.339** $\pm$ **0.006** on 1,433 AURSAD (UR3e) Screwdriving episodes, outperforming every baseline including the kinematic baseline (0.373 $\pm$ 0.005) and all structureless learned models (Table 6). Raw effort-MAE remains high (1.74 vs. 1.51 for a zero-predictor) due to static payload and gravity-compensation differences between embodiments, but MC-MAE confirms that the *shape* of the dynamics transfers successfully across machines.

Table 6. Zero-shot cross-embodiment transfer (voraus-AD Yu-Cobot P&P $\rightarrow$ AURSAD UR3e Screwdriving, 1,433 episodes). MC-MAE removes per-joint trajectory bias; lower is better. 95% CI computed over episodes.

Model	**MC-MAE**	**95% CI** ($\pm$)
Linear	0.928	0.023
Flat MLP	0.792	0.019
TCN	0.770	0.017
Kinematic Baseline	0.373	0.005
TCN-Transformer (Ours)	**0.339**	**0.006**

5. Conclusion & Limitations

FactoryNet provides the largest open-source, fault-injected time-series corpus for industrial robotics, unified by the S-E-F-C taxonomy. Our experiments show that the schema enables competitive anomaly detection with 5$\times$ fewer signals, accurate multi-step dynamics forecasting, and positive zero-shot cross-embodiment transfer. The sim-to-real gaps we quantify are naturally interpreted as errors of learned *forward* dynamics under S–E–F–C-aligned inputs, not merely covariate shift in raw telemetry. Current limitations include synthetic data restricted to Pick & Place and cross-embodiment transfer evaluated on a single source-target pair; future work will expand to additional machine families, tasks, and transfer settings.

References

Ahmed, C. M., Palleti, V. R., and Mathur, A. P. Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, pp. 25–28, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349475. doi: 10.1145/3055366.3055375. URL https://doi.org/10.1145/3055366.3055375.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

Brockmann, J. T., Rudolph, M., Rosenhahn, B., and Wandt, B. The voraus-ad dataset for anomaly detection in robot applications. *IEEE Transactions on Robotics*, 40:438–451, 2023. doi: 10.1109/TRO.2023.3332224. URL https://ieeexplore.ieee.org/document/10315239.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

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

Dosovitskiy, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Downs, J. and Vogel, E. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245–255, 1993. doi: 10.1016/0098-1354(93)80018-I. URL https://doi.org/10.1016/0098-1354(93)80018-I.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., and Crawford, K. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. doi: 10.1145/3458723. URL https://doi.org/10.1145/3458723.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., and Dubrawski, A. MOMENT: A family of open time-series foundation models. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Hoo, S. B., Müller, S., Salinas, D., and Hutter, F. From tables to time: Extending tabpfn-v2 to time series forecasting. *arXiv preprint arXiv:2501.02945*, 2025.

Katser, I. D. and Kozitsin, V. O. Skoltech anomaly benchmark (skab). https://www.kaggle.com/dsv/1693952, 2020.

Khazatsky, A., Pertsch, K., Nair, S., et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. URL https://www.roboticsproceedings.org/rss20/p120.html.

Lai, K.-H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., Chen, Y., Zumkhawaka, P., Wan, M., Martinez, D., and Hu, X. Tods: An automated time series outlier detection system, 2025. URL https://arxiv.org/abs/2009.09822.

Leporowski, B., Tola, D., Hansen, C., and Iosifidis, A. Aursad: Universal robot screwdriving anomaly detection dataset. *arXiv*, 2021. URL https://arxiv.org/abs/2102.01409.

Lessmeier, C., Kimotho, J. K., Zimmer, D., and Sextro, W. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: a benchmark data set for data-driven classification. In *Proceedings of the European Conference of the Prognostics and Health Management Society*, 2016.

Li, Z. Cwru bearing dataset and gearbox dataset of IEEE phm challenge competition in 2009, 2019. URL https://dx.doi.org/10.21227/g8ts-zd15.

Liu, Q. and Paparrizos, J. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. In *Advances in Neural Information Processing Systems*, 2024. Datasets and Benchmarks Track.

Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., and Long, M. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32369–32399. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24cb.html.

O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://ieeexplore.ieee.org/document/10611477.

PHM Society. 2010 PHM society conference data challenge. https://www.phmsociety.org/competition/phm/10, 2010. Dataset and challenge page for remaining useful life estimation of high-speed CNC milling machine cutters; accessed 2026-05-08.

Raissi, M., Perdikaris, P., and Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045. URL https://doi.org/10.1016/j.jcp.2018.10.045.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., et al. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv*, 2023. URL https://arxiv.org/abs/2310.08278.

Ribeiro, F. M. L., Marins, M. A., Netto, S. L., and da Silva, E. A. B. Mafaulda – machinery fault database. http://www02.smt.ufrj.br/~offshore/mfs/page_01.html, 2016. Signals, Multimedia, and Telecommunications Laboratory, COPPE/Poli/UFRJ; contact: felipe.ribeiro@smt.ufrj.br.

Saxena, A., Goebel, K., Simon, D., and Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management*, pp. 1–9. IEEE, 2008. doi: 10.1109/PHM.2008.4711414. URL https://ieeexplore.ieee.org/document/4711414.

Sun, S. Cnc mill tool wear, 2018. URL https://www.kaggle.com/datasets/shasun/tool-wear-detection-in-cnc-mill. CC0 Public Domain; machining experiments at the System-level Manufacturing and Automation Research Testbed (SMART), University of Michigan; dataset metadata last modified 2018-04-06 on Kaggle.

Veloso, B., Gama, J., Ribeiro, R. P., and Pereira, P. M. A benchmark dataset for predictive maintenance, 2022. URL https://arxiv.org/abs/2207.05466.

Wang, B., Lei, Y., Li, N., and Li, N. 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.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

World Bank. Manufacturing, value added (% of gdp), 2026. URL https://data.worldbank.org/indicator/NV.IND.MANF.ZS. Accessed: 2026-04-30.

Xu, J., Wu, H., Wang, J., and Long, M. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Zeng, A., Song, S., Ju, J., Hsieh, A., Huang, I., Chen, H., Adelson, E., and Rodriguez, A. Tossingbot: Learning to throw arbitrary objects with envelope-based residual reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

A. Datasheet for Datasets (Gebru et al., 2021)

Following the recommendations of Gebru et al. (2021), we provide a structured datasheet for FactoryNet v1.0 (Table 7).

B. Signal-to-S-E-F-C Mapping

We provide one table per source mapping raw signal names to S-E-F-C roles. UR3, KUKA KR10 and Isaac Sim mappings are listed with raw signal names confirmed; CNC, AURSAD, and voraus-AD mappings are fully confirmed.

B.1 UR3 Laboratory Source

Signals of UR3 can be found at Table 8

B.2 KUKA KR10 Laboratory Source

Signals of KUKA KR10 can be found at Table 9

B.3 UMich CNC Source

Mapping of CNC columns (Table 10)

B.4 AURSAD Source (UR3e Screwdriving)

Per-joint signals follow a fixed pattern for joints $i \in \{0, \dots, 5\}$; one representative joint (joint 0) is shown below. (Table 11)

B.5 voraus-AD Source (Yu-Cobot)

Per-joint signals follow the same index pattern for joints $i \in \{1, \dots, 6\}$ (1-indexed in source); one representative joint is shown. (Table 12)

B.6 Isaac Sim UR5 (Synthetic)

Mapping of Isaac Sim UR5 columns (Table 13)

C. Full Fault Catalog

Table 14 lists all 27 fault types injected during laboratory data collection.

D. Multi-Step Forecasting Metrics

Table 15. Multi-step forecasting on voraus-AD (Yu-Cobot) Pick & Place, held-out normal episodes. MSE ($\times 10^{-4}$ rad^2) and MAE ($\times 10^{-2}$ rad) $\pm$ std over test episodes. Lower is better. Values below 0.005 in the chosen unit are rounded to 0.00.

	H = 50		**H = 100**		**H = 200**	
Model	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
Linear	$28.17_{\pm 0.28}$	$3.25_{\pm 0.02}$	$541.22_{\pm 36.57}$	$12.99_{\pm 0.27}$	$6930.06_{\pm 67.51}$	$51.31_{\pm 0.84}$
Flat MLP	$1.69_{\pm 0.05}$	$0.80_{\pm 0.02}$	$67.02_{\pm 17.12}$	$4.15_{\pm 0.46}$	$1939.57_{\pm 160.39}$	$24.43_{\pm 1.33}$
TCN	$0.46_{\pm 0.03}$	$0.46_{\pm 0.02}$	$102.63_{\pm 30.85}$	$4.68_{\pm 0.78}$	$2940.49_{\pm 218.03}$	$30.20_{\pm 1.73}$
TCN-Transformer	$\mathbf{0.00_{\pm 0.00}}$	$\mathbf{0.01_{\pm 0.00}}$	$\mathbf{0.00_{\pm 0.00}}$	$\mathbf{0.03_{\pm 0.00}}$	$\mathbf{0.11_{\pm 0.01}}$	$\mathbf{0.18_{\pm 0.01}}$

Table 7. Datasheet for FactoryNet v1.0.

Question	**Answer**
Motivation	
Purpose	Pretraining substrate for industrial time-series foundation models, anomaly detection benchmarking, and cross-embodiment transfer research.
Creators	Karim Othman (Forgis), Jonas Petersen (ETH Zurich / Forgis), Matei Ignuta-Ciuncanu (Imperial College / UC Berkeley), Camilla Mazzoleni (Forgis), Federico Martelli (ETH Zurich / Forgis), Alessandro Lombardi (Forgis), Riccardo Maggioni (Forgis), Philipp Petersen (University of Vienna).
Funding	Fully funded by Forgis AG.
Composition	
Instances	Episodes of robotic task execution, each containing multi-channel time-series at 100 Hz with S-E-F-C role annotations.
Count	23,098 episodes total: 7,141 real lab (UR3), 1,973 real lab (KUKA KR10), 4,185 open-source (voraus-AD 2,122; AURSAD (UR3e) 2,045; UMich CNC 18), 9,799 synthetic (Isaac Sim UR5).
Sampling	Lab data: programmatic task execution with configurable fault injection (40% healthy, 60% anomalous). Open-source: full datasets ingested. Synthetic: procedural generation with domain randomization.
Missing data	Sensor dropouts $<0.1\%$ of time steps; handled via linear interpolation. No episodes removed due to missing data.
Confidentiality	No PII. Industrial parameters only (joint angles, torques, velocities).
Collection Process	
Acquisition	Lab episodes recorded by the authors via robot controller APIs (RTDE for UR3, RSI for KUKA KR10) at 100 Hz. Open-source subsets downloaded and re-processed into S-E-F-C format via adapter scripts.
Mechanisms	Automated scripts executing task programs with fault injection at configurable rates. 27 anomaly types injected programmatically (payload variation, misgrips, collisions, stripped threads, jamming, friction changes, etc.).
Who collected	Karim Othman, Jonas Petersen, Camilla Mazzoleni, Federico Martelli, Alessandro Lombardi, Riccardo Maggioni (Forgis / ETH Zurich).
Time frame	March 1 – April 20, 2026.
Ethical review	N/A — no human subjects.
Preprocessing, Cleaning, and Labeling	
Preprocessing	Resampling to 100 Hz, NaN interpolation, S-E-F-C column mapping per source via adapter scripts. No per-channel normalisation applied (raw physical units preserved).
Raw data	Raw recordings available on request. Distributed format: S-E-F-C Parquet files with metadata.
Labeling	Programmatic fault injection with known ground truth; no manual annotation required for lab data. Open-source subsets retain original labels, mapped to unified anomaly taxonomy.
Uses	
Prior uses	Anomaly detection baselines (this paper), cross-embodiment transfer experiments (this paper).
Other uses	Forecasting, remaining useful life estimation, control policy learning, sim-to-real benchmarking, counterfactual analysis.
Misuse potential	Dataset reflects specific robot models and lab conditions; generalisation claims to unseen embodiments or industrial deployments should be scoped accordingly.
Distribution	
Distribution	Available on HuggingFace at `https://huggingface.co/datasets/Forgis/FactoryNet`.
License	Novel lab and synthetic data under MIT. Open-source subsets retain original licenses (CC-BY 4.0 or equivalent).
Restrictions	None. No export controls apply.
Maintenance	
Maintainer	Forgis AG. Issue tracker: GitHub repository.
Updates	Semantic versioning (v1.0 at submission). New embodiments and tasks increment minor version; schema changes increment major version. Prior versions remain available.
Retention	All versions permanently hosted on HuggingFace.
Errata	Reported and tracked via GitHub issues; corrections published as patch versions.

Table 8. S-E-F-C mapping for the UR3 laboratory source.

Raw Signal Name	S-E-F-C Role	Unit	Notes
`machine_id`	Metadata	—	Episode metadata
`ur3_robot_target_joint_0...5`	Setpoint	rad	6 joints
`ur3_robot_target_joint_vel_0...5`	Setpoint	rad/s	6 joints
`ur3_robot_joint_0...5`	Feedback	rad	Actual positions, 6 joints
`ur3_robot_joint_vel_0...5`	Feedback	rad/s	Actual velocities, 6 joints
`ur3_robot_joint_current_0...5`	Effort	A	6 joints
`ur3_robot_target_joint_current_0...5`	Setpoint	A	6 joints
`ur3_robot_joint_temp_0...5`	Context	°C	6 joints
`ur3_robot_joint_control_output_0...5`	Effort	—	6 joints
`ur3_robot_joint_mode_0...5`	Context	enum	6 joints
`ur3_robot_tcp_x/y/z`	Feedback	m	Actual TCP position
`ur3_robot_tcp_rx/ry/rz`	Feedback	rad	Actual TCP orientation
`ur3_robot_target_tcp_x/y/z`	Setpoint	m	Target TCP position
`ur3_robot_target_tcp_rx/ry/rz`	Setpoint	rad	Target TCP orientation
`ur3_robot_tcp_speed_x/y/z/rx/ry/rz`	Feedback	m/s & rad/s	Actual TCP speed (6 axes)
`ur3_robot_target_tcp_speed_x/y/z/rx/ry/rz`	Setpoint	m/s & rad/s	Target TCP speed (6 axes)
`ur3_robot_tcp_force_x/y/z`	Effort	N	TCP force
`ur3_robot_tcp_torque_x/y/z`	Effort	Nm	TCP torque
`ur3_robot_robot_mode`	Context	enum	Enum
`ur3_robot_safety_mode`	Context	enum	Enum
`ur3_robot_digital_inputs`	Context	bitmask	Bitmask
`ur3_robot_digital_outputs`	Context	bitmask	Bitmask
`ur3_robot_robot_status`	Context	enum	
`ur3_robot_safety_status`	Context	enum	
`ur3_robot_runtime_state`	Context	enum	
`ur3_robot_main_voltage`	Context	V	V
`ur3_robot_robot_voltage`	Context	V	V
`ur3_robot_robot_current`	Context	A	A
`ur3_robot_analog_input_0/1`	Context	V/A	
`ur3_robot_analog_output_0/1`	Context	V/A	
`ur3_robot_speed_scaling`	Context	%	
`ur3_robot_target_speed_fraction`	Context	%	
`ur3_robot_momentum`	Context	kg·m/s	
`realsense_camera_frame_count`	Context	count	
`realsense_camera_connected`	Context	bool	
`realsense_camera_streaming`	Context	bool	

Table 9. S-E-F-C mapping for the KUKA KR10 laboratory source. Note hardware limitations of KSS 8.3.

Signal Group	Described Fields	S-E-F-C Role	Unit
Position (actual)	Actual joint positions (×6)	Feedback	degrees
Position (commanded)	Commanded joint positions (×6)	Setpoint	degrees
TCP pose (actual)	Actual TCP pose (×6)	Feedback	mm / degrees
Motor current	Motor current per axis (×6)	Effort	% of max
Motor torque	Motor torque per axis (×6)	Effort	Nm
Motor temperature	Motor temperature per axis (×6)	Context	Kelvin
Cartesian accel.	Accel. X, Y, Z, abs (×4)	Feedback	m/s^2
Speed override	Speed override (scalar)	Context	%
Process state	State enum (FREE/ACTIVE/STOP/END)	Context	enum
Digital I/O	Digital inputs bitmask	Context	bitmask
Digital I/O	Digital outputs bitmask	Context	bitmask

Not available on KSS 8.3: joint velocities, commanded TCP pose, TCP force/torque, joint voltage, tool accelerometer, joint control output.

Table 10. S-E-F-C mapping for the UMich CNC milling source. Axes: X1, Y1, Z1 (linear), S1 (spindle).

Raw Signal Name	S-E-F-C Name	Role	Unit	Axis
`X1_CommandPosition`	`setpoint_pos_0`	Setpoint	mm	X
`X1_CommandVelocity`	`setpoint_vel_0`	Setpoint	mm/s	X
`X1_CommandAcceleration`	`setpoint_acc_0`	Setpoint	mm/s^2	X
`X1_ActualPosition`	`feedback_pos_0`	Feedback	mm	X
`X1_ActualVelocity`	`feedback_vel_0`	Feedback	mm/s	X
`X1_ActualAcceleration`	`feedback_acc_0`	Feedback	mm/s^2	X
`X1_CurrentFeedback`	`feedback_current_0`	Feedback	A	X
`X1_OutputCurrent`	`effort_current_0`	Effort	A	X
`X1_OutputVoltage`	`effort_voltage_0`	Effort	V	X
`X1_OutputPower`	`effort_power_0`	Effort	W	X
`X1_DCBusVoltage`	`ctx_busvoltage_0`	Context	V	X
(Pattern repeats for Y1 $\rightarrow$ axis 1, Z1 $\rightarrow$ axis 2)				
`S1_CommandPosition`	`setpoint_pos_3`	Setpoint	deg	Spindle
`S1_CommandVelocity`	`setpoint_vel_3`	Setpoint	rpm	Spindle
`S1_CommandAcceleration`	`setpoint_acc_3`	Setpoint	rpm/s	Spindle
`S1_ActualPosition`	`feedback_pos_3`	Feedback	deg	Spindle
`S1_ActualVelocity`	`feedback_vel_3`	Feedback	rpm	Spindle
`S1_ActualAcceleration`	`feedback_acc_3`	Feedback	rpm/s	Spindle
`S1_CurrentFeedback`	`feedback_current_3`	Feedback	A	Spindle
`S1_OutputCurrent`	`effort_current_3`	Effort	A	Spindle
`S1_OutputVoltage`	`effort_voltage_3`	Effort	V	Spindle
`S1_OutputPower`	`effort_power_3`	Effort	W	Spindle
`S1_DCBusVoltage`	`ctx_busvoltage_3`	Context	V	Spindle
`S1_SystemInertia`	`ctx_inertia_3`	Context	—	Spindle
`M1_CURRENT_PROGRAM_NUMBER`	`ctx_program_number`	Context	—	—
`M1_sequence_number`	`ctx_sequence_number`	Context	—	—
`M1_CURRENT_FEEDRATE`	`ctx_feedrate`	Context	mm/min	—
`Machining_Process`	`ctx_process_phase`	Context	enum	—

Table 11. S-E-F-C mapping for the AURSAD source (representative joint 0 pattern; repeats for joints 1–5 with incremented index).

Raw Signal Name	**S-E-F-C Name**	**Role**	**Unit**	**Notes**
`target_q_0`	`setpoint_pos_0`	Setpoint	rad	Joint 0
`target_qd_0`	`setpoint_vel_0`	Setpoint	rad/s	
`target_qdd_0`	`setpoint_acc_0`	Setpoint	rad/s^2	
`target_current_0`	`setpoint_current_0`	Setpoint	A	
`target_moment_0`	`setpoint_torque_0`	Setpoint	Nm	
`actual_q_0`	`feedback_pos_0`	Feedback	rad	
`actual_qd_0`	`feedback_vel_0`	Feedback	rad/s	
`actual_current_0`	`effort_current_0`	Effort	A	
`actual_control_output_0`	`effort_control_0`	Effort	—	
`actual_joint_voltage_0`	`effort_voltage_0`	Effort	V	
`joint_temperatures_0`	`ctx_temp_0`	Context	°C	
`joint_mode_0`	`ctx_joint_mode_0`	Context	enum	
`target_TCP_pose_0`	`setpoint_pos_cartesian_0`	Setpoint	m/rad	
`target_TCP_speed_0`	`setpoint_vel_cartesian_0`	Setpoint	m/s	
`actual_TCP_pose_0`	`feedback_pos_cartesian_0`	Feedback	m/rad	
`actual_TCP_speed_0`	`feedback_vel_cartesian_0`	Feedback	m/s	
`actual_TCP_force_0`	`effort_force_cartesian_0`	Effort	N	
`actual_tool_accelerometer_0/1/2`	`auxiliary_accel_tool_0/1/2`	—	m/s^2	Tool IMU
`output_double_register_24`	`feedback_torque_tool`	Feedback	Nm	Tool torque
`output_double_register_25`	`effort_torque_tool`	Effort	Nm	
`output_double_register_26`	`setpoint_torque_tool`	Setpoint	Nm	
`output_double_register_27`	`setpoint_torque_gradient_tool`	Setpoint	Nm/s	
`actual_main_voltage`	`ctx_main_voltage`	Context	V	
`actual_robot_voltage`	`ctx_robot_voltage`	Context	V	
`actual_robot_current`	`ctx_robot_current`	Context	A	
`speed_scaling`	`ctx_speed_scaling`	Context	—	
`target_speed_fraction`	`ctx_target_speed_fraction`	Context	—	
`actual_momentum`	`ctx_momentum`	Context	kg·m/s	
`robot_mode`	`ctx_robot_mode`	Context	enum	
`safety_mode`	`ctx_safety_mode`	Context	enum	
`runtime_state`	`ctx_runtime_state`	Context	enum	
`label`	`raw_label`	—	—	0=Normal, 1=Damaged screw, 2=Extra part, 3=Missing screw, 4=Damaged thread

Table 12. S-E-F-C mapping for the voraus-AD source (representative joint 1 pattern; repeats for joints 2–6).

Raw Signal Name	S-E-F-C Name	Role	Unit	Notes
target_position_1	setpoint_pos_0	Setpoint	rad	Joint 1→idx 0
target_velocity_1	setpoint_vel_0	Setpoint	rad/s	
target_acceleration_1	setpoint_acc_0	Setpoint	rad/s^2	
target_torque_1	setpoint_torque_0	Setpoint	Nm	
joint_position_1	feedback_pos_0	Feedback	rad	
joint_velocity_1	feedback_vel_0	Feedback	rad/s	
motor_position_1	feedback_motor_pos_0	Feedback	rad	
motor_velocity_1	feedback_motor_vel_0	Feedback	rad/s	
torque_sensor_a_1	feedback_torque_a_0	Feedback	Nm	
torque_sensor_b_1	feedback_torque_b_0	Feedback	Nm	
motor_torque_1	effort_motor_torque_0	Effort	Nm	
motor_iq_1	effort_current_iq_0	Effort	A	
motor_id_1	effort_current_id_0	Effort	A	
power_motor_el_1	effort_power_el_0	Effort	W	
power_motor_mech_1	effort_power_mech_0	Effort	W	
power_load_mech_1	effort_power_load_0	Effort	W	
motor_voltage_1	effort_voltage_0	Effort	V	
computed_inertia_1	ctx_inertia_0	Context	—	
computed_torque_1	ctx_computed_torque_0	Context	Nm	
supply_voltage_1	ctx_busvoltage_0	Context	V	
brake_voltage_1	ctx_brake_voltage_0	Context	V	
robot_voltage	ctx_robot_voltage	Context	V	Global
robot_current	ctx_robot_current	Context	A	Global
io_current	ctx_io_current	Context	A	Global
system_current	ctx_system_current	Context	A	Global
anomaly	ctx_is_anomaly	—	bool	Label
category	ctx_anomaly_category	—	str	Fault type
setting	ctx_setting	—	str	

Table 13. S-E-F-C mapping for the Isaac Sim (UR5) synthetic source. Signals are procedurally generated and logged at a base 60 Hz simulation step before interpolation.

Signal Group (Raw Prefix)	**Described Fields**	**S-E-F-C Role**	**Unit**
`episode`, `step`, `time`	Global/episode steps, sim/wall time	Metadata	count / s
`state_machine`, `phase`	Task, controller phases, event IDs	Context	enum / str
`joint_cmd_pos_rad_*`	Commanded joint positions (×6)	Setpoint	rad
`joint_cmd_vel_radps_*`	Commanded joint velocities (×6)	Setpoint	rad/s
`joint_pos_rad_*`	Actual joint positions (×6)	Feedback	rad
`joint_vel_radps_*`	Actual joint velocities (×6)	Feedback	rad/s
`joint_accel_radps2_*`	Actual joint accelerations (×6)	Feedback	rad/s^2
`joint_torque_nm_*`	Joint applied torques (×6)	Effort	Nm
`joint_pos_error_rad_*`	Tracking error per joint (×6)	Context	rad
`ee_cmd_pos_*`, `quat_*`	Commanded TCP pose (pos, quat)	Setpoint	m / —
`ee_pos_*`, `quat_*`	Actual TCP pose (pos, quat)	Feedback	m / —
`ee_euler_*_rad`	Actual TCP orientation (RPY)	Feedback	rad
`ee_linvel_*`, `angvel_*`	TCP velocities (linear, angular)	Feedback	m/s / rad/s
`ee_linacc_*`, `angacc_*`	TCP accelerations (linear, angular)	Feedback	m/s^2 / rad/s^2
`gripper_cmd_rad`	Commanded gripper position	Setpoint	rad
`gripper_pos_rad`	Actual gripper position	Feedback	rad
`gripper_attached`	Gripper logical contact state	Context	bool
`contact_force_*_n`	Physical EE contact force (XYZ, mag)	Effort	N
`contact_torque_*_nm`	Physical EE contact torque (XYZ, mag)	Effort	Nm
`cube_pos_*`, `quat_*`	Target object pose (pos, quat)	Context	m / —
`cube_linvel_*`, `angvel_*`	Target object velocities	Context	m/s / rad/s
`ee_cube_offset_*`	Relative distance to target	Context	m
`cube_mass_kg`	Domain rand: payload mass	Context	kg
`cube_friction_coeff`	Domain rand: target object friction	Context	—
`cube_width/depth/height_m`	Domain rand: target dimensions	Context	m

Table 14. Complete Fault Catalog. Checkmarks (✓) indicate whether the fault is included in the Pick-and-Place (PP), Screwing (Scr), and Peg-in-Hole (PiH) task recordings. Crosses (×) indicate absence.

Fault	**Explanation**	**Injection Procedure**	**PP**	**Scr**	**PiH**
Damaged screw thread	Screw thread physically damaged, preventing proper engagement during tightening.	Pre-damaged the screw thread with sandpaper.	×	✓	×
Missing screw	Tightening is attempted with no screw present.	Removed the screw from screwdriver before the cycle.	×	✓	×
Damaged plate thread	The threaded hole in the plate is damaged, preventing engagement.	Pre-damaged the plate hole with a metal screwdriver.	×	✓	×
Loosening phase	A counterclockwise loosening rotation replaces tightening; a normal phase with a distinct signal. Counted as an anomaly to match AURSAD schema.	Reversed the rotation direction in the controller program to execute a counterclockwise loosening pass.	×	✓	×
Gripper activation failure	The vacuum gripper fails to activate and never picks up the box.	Disabled the gripper activation command in the robot program.	✓	×	×
Gripper release during motion	The gripper releases the payload mid-trajectory, causing an abrupt payload loss.	Triggered a programmatic gripper-release command at a scripted timestep mid-trajectory.	✓	×	×
Additional axis payload	A dead weight attached to one link increases inertia and gravity loading on all joints.	Bolted calibrated weights of varying mass to one robot link.	✓	✓	×
Collision with foam object	Contact with a soft foam block produces a brief TCP force spike without a protective stop.	Placed a foam cube directly in the programmed TCP trajectory.	✓	✓	✓
Unexpected payload weight	The transported box has a weight that deviates from the nominal payload configuration.	Swapped the nominal box for a known heavier or lighter one.	✓	×	×
Invalid gripping position	Timing error: the gripper closes after the arm has begun lifting, gripping the object off-centre.	Added a sleep() call in the controller program so gripper closure lags the lift motion.	✓	×	×
Unstable mounting platform	Base instability introduces low-frequency vibrations into the arm.	Placed 8 layers of towels under the base plate.	✓	×	×
Joint position limit violation	A joint moves outside its configured soft or hard position limits, triggering a safety stop.	Random waypoint slightly beyond the configured soft joint limit.	✓	×	×
TCP frame misconfiguration	TCP frame or mounting orientation misconfigured at the controller; gravity torques deviate.	Entered an incorrect TCP offset or mounting angle in the controller settings.	✓	✓	✓
Payload weight misconfiguration	Payload mass misconfigured while a tool or workpiece is physically attached.	Left configured payload weight at multiple wrong mass configurations.	✓	✓	×
External arm disturbance	A continuous external force pulls or pushes the TCP during motion.	Anchored an elastic rope between the bench frame and the tool flange.	✓	✓	✓
Mild payload CoG misconfig.	Payload CoG specified at an incorrect offset from the tool flange; gravity compensation is wrong.	Entered a CoG offset in the payload configuration that differs from physical CoG.	✓	✓	✓
Collision with hanging cable	A loose cable drags along a robot link or the tool flange during motion.	Suspended a loose cable across the trajectory at arm height.	✓	✓	✓
Collision with cardboard	A cardboard carton in the trajectory provides moderate resistance; may trigger a protective stop.	Placed a free-standing or braced cardboard box in the programmed trajectory.	✓	✓	✓
Collision with rigid object	A rigid plastic block in the TCP path triggers an immediate protective stop.	Clamped a rigid plastic block in the programmed TCP path.	✓	✓	✓
Peg insertion misalignment	Peg approaches the hole at an angular or lateral offset and jams against the rim.	Added a recorded lateral offset at the insertion approach waypoint.	×	×	✓
Hole obstruction	Foreign object or debris in the hole prevents full insertion.	Placed a small object (metal screw) inside the hole before insertion.	×	×	✓
Incorrect insertion depth	Insertion terminates at an incorrect Z depth due to a misconfigured waypoint.	Shifted the insertion endpoint waypoint Z by recorded offset from nominal.	×	×	✓
Peg surface contamination	Contamination or roughness on the peg raises insertion friction and produces stick-slip.	Applied dry chalk powder to the peg surface.	×	×	✓
Fixture displacement	Insertion fixture shifted laterally, breaking the match between programmed approach and actual hole.	Physically shifted the hole fixture by recorded offset without updating program.	×	×	✓
Self-collision	Arm configuration causes a link to contact the robot body or mounting fixture.	Random waypoint that forces a self-colliding configuration.	✓	×	×
Missing box	Box absent from the pick position; the gripper closes on empty air.	Removed the box from the pick position before the episode started.	✓	×	×
Missing peg	Arm descends into the hole empty-handed; insertion produces no contact force.	Removed the peg from the gripper before the episode started.	×	×	✓