

CausalPulse: An Industrial-Grade Neurosymbolic Multi-Agent Copilot for Causal Diagnostics in Smart Manufacturing

Chathurangi Shyalika¹, Utkarshani Jaimini², Cory Henson³, Amit Sheth^{1,4}

¹Artificial Intelligence Institute, University of South Carolina

²Stride Lab, University of Michigan, Dearborn

³Bosch Center for Artificial Intelligence, Pittsburgh

⁴Indian AI Research Organization

Abstract

Modern manufacturing environments demand real-time, trustworthy, and interpretable root-cause insights to sustain productivity and quality. Traditional analytics pipelines often treat anomaly detection, causal inference, and root-cause analysis as isolated stages, limiting scalability and explainability. In this work, we present *CausalPulse*, an industry-grade multi-agent copilot that automates causal diagnostics in smart manufacturing. It unifies anomaly detection, causal discovery, and reasoning through a neurosymbolic architecture built on standardized agentic protocols. CausalPulse is being deployed in a Robert Bosch manufacturing plant, integrating seamlessly with existing monitoring workflows and supporting real-time operation at production scale. Evaluations on both public (*Future Factories*) and proprietary (*Planar Sensor Element*) datasets show high reliability, achieving overall success rates of 98.0% and 98.73%. Per-criterion success rates reached 98.75% for planning and tool use, 97.3% for self-reflection, and 99.2% for collaboration. Runtime experiments report end-to-end latency of 50–60 s per diagnostic workflow with near-linear scalability ($R^2=0.97$), confirming real-time readiness. Comparison with existing industrial copilots highlights distinct advantages in modularity, extensibility, and deployment maturity. These results demonstrate how CausalPulse’s modular, human-in-the-loop design enables reliable, interpretable, and production-ready automation for next-generation manufacturing.

Demo Video —

<https://www.youtube.com/watch?v=bh1XHHvqZos>

Extended Version of the Paper Available at —

<https://arxiv.org/pdf/2603.29755>

Introduction

Manufacturing systems increasingly rely on dense, heterogeneous sensors and automated production lines that must maintain continuous operation under strict quality and yield constraints. Real-time detection of anomalies and accurate identification of their root causes are essential to prevent costly scrap, downtime, and latent defects. However, traditional analytic pipelines treat anomaly detection, causal discovery, and decision-making as isolated stages. Sensors feed black-box models, experts interpret alerts, and interventions occur in an ad hoc manner, leading to delayed diagnosis,

limited traceability, and reduced operator trust. Reliable root cause analysis (RCA) remains difficult due to multimodal and incomplete production data (Kuo et al. 2021; Pietsch et al. 2024), partially known and context-dependent causal relationships (Ko et al. 2023; Pearl 2019), and the need for explainable, auditable, operator-in-the-loop systems that ensure safety and accountability (Sofianidis et al. 2021; Trivedi et al. 2024). Large-scale deployment requires modular, interoperable, and composable architectures to support robust integration in industrial pipelines (Ribeiro and Gomes 2021).

Classical research in causal analysis and multi-agent systems provides a foundation for addressing these challenges. Structure-learning algorithms characterize conditional independence and score-based approaches to infer causal graphs from observational data (Spirtes, Glymour, and Scheines 2000; Chickering 2002). Multi-agent systems and service-oriented agent frameworks have demonstrated how distributed reasoning and standardized communication protocols enable modular, scalable automation in industrial settings (Bellifemine, Poggi, and Rimassa 2000; Wooldridge 2009). Recently, large language models have enabled practical methods that combine reasoning and tool use, integrating model-based planning with external tools to extend their capabilities (Yao et al. 2022; Schick et al. 2023). These developments motivate the synthesis of principled causal inference, neurosymbolic domain priors, and agentic orchestration to produce interpretable, operational RCA systems. Despite promising advances, current industrial copilots and diagnostic platforms lack a unified, standards-based, multi-agent framework that (i) integrates preprocessing and multi-modal data, (ii) constrains causal analysis with domain rules, (iii) produces transparent and explainable results, (iv) supports operator-driven selection, review, and replanning at runtime, and (v) enables easy extensibility to new use cases and manufacturing tasks.

To address these gaps, we present *CausalPulse*, an industrial-level neurosymbolic multi-agent copilot for *causal diagnostics* in smart manufacturing. Here, *causal diagnostics* refers to the unified causal analysis process that encompasses both structure learning and RCA. As a copilot, CausalPulse serves as a collaborative intelligence layer that guides human–AI interaction, enabling human-in-the-loop decision-making. CausalPulse combines (1) a protocol- and framework-first agentic substrate for discoverable, composable services; (2) hybrid neurosymbolic causal discovery that

blends structure learning with expert-defined rules; (3) a path-based RCA engine that scores and ranks candidate failure paths for operator usage; and (4) an adaptable workflow engine that supports operator-in-the-loop verification. We evaluated CausalPulse on two real-world use cases: a grinding process from an industrial production line and an assembly process from a research and development environment.

Related Work

Agentic AI Frameworks and Orchestration Protocols.

The rise of *Agentic AI* marks a shift toward systems that couple language-model reasoning with modular orchestration and communication protocols. Frameworks such as LangGraph (LangChain AI 2024), AutoGen (Wu et al. 2024), CrewAI (CrewAI 2025), and OpenDevin (Wang et al. 2024) integrate standardized interfaces for tool invocation, planning, and memory management (Bandi et al. 2025; Bouse-touane 2025). Protocols like the Model Context Protocol (MCP) (Anthropic 2024), Agent Communication Protocol (age 2025), Google A2A (Google A2A 2025) and Agora (Marro et al. 2024) have formalized structured message passing and task routing, allowing autonomous agents to coordinate across services in production workflows (Gort, Kibalya, and Antonopoulos 2025; Sawant 2025; Derouiche, Brahmi, and Mazeni 2025). These developments have renewed the practicality of deploying multi-agent reasoning systems in complex industrial environments.

Industrial and Diagnostic Copilots. Several domain-specific copilots adopt multi-agent and modular architectures for industrial monitoring and diagnostics. The UCSD Causal Copilot (Wang et al. 2025) integrates causal discovery and reasoning through an LLM-based planner. MATMCD (Shen et al. 2024) and IBM AssetOpsBench (Patel et al. 2025) employ multi-agent pipelines that combine IoT data processing, anomaly detection, and maintenance scheduling. SmartPilot (Shyalika et al. 2025a,b, 2026) extends this paradigm by coupling predictive agents for process forecasting, anomaly diagnostics, and causal analysis. These systems demonstrate the feasibility of integrating multi-agent coordination and reasoning in industrial settings but remain limited in extensibility and in incorporating semantic context, domain priors, and neurosymbolic reasoning.

Despite growing interest in industrial multi-agent systems, existing copilots primarily focus on task automation, fault prediction, or isolated RCA components. They often lack an integrated framework that unifies data preprocessing, anomaly detection, causal discovery, and RCA within a cohesive agentic architecture. Current systems provide limited support for neurosymbolic reasoning and operator-driven validation. Addressing these gaps, CausalPulse introduces an interpretable, standards-based multi-agent framework that supports dynamic orchestration, knowledge-infused causal diagnostics, and operator-in-the-loop decision intelligence for smart manufacturing.

CausalPulse System Overview

System Architecture and Four-layer Framework

Figures 1 and 2 illustrate complementary perspectives of the CausalPulse platform. The high-level system architecture (Figure 1) details the operational workflow, how data flows from the production line through multi-agent reasoning to generate diagnostics and recommendations. In contrast, the four-layer framework (Figure 2) formalizes the structural organization that enables this workflow, comprising the *user-facing*, *agent*, *utility*, and *data* layers. Both representations overlap conceptually; the multi-agent copilot in the system architecture maps to the agent and utility layers in the framework but serves distinct principles. The former emphasizes functional interactions, while the latter abstracts the system’s architectural hierarchy and interoperability mechanisms. The four layers of the framework are detailed below.

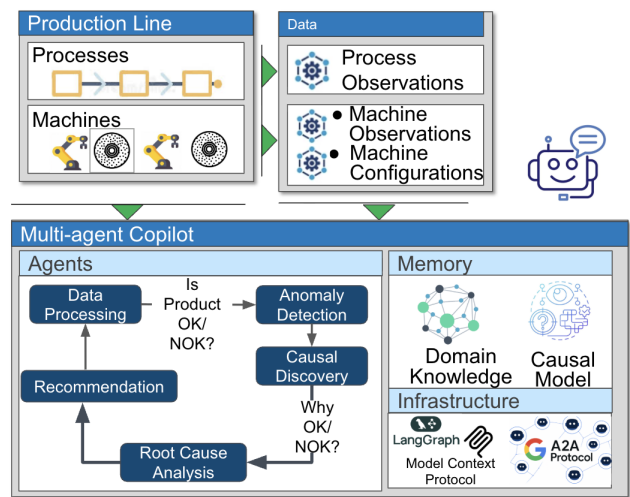


Figure 1: High-level overview of CausalPulse system

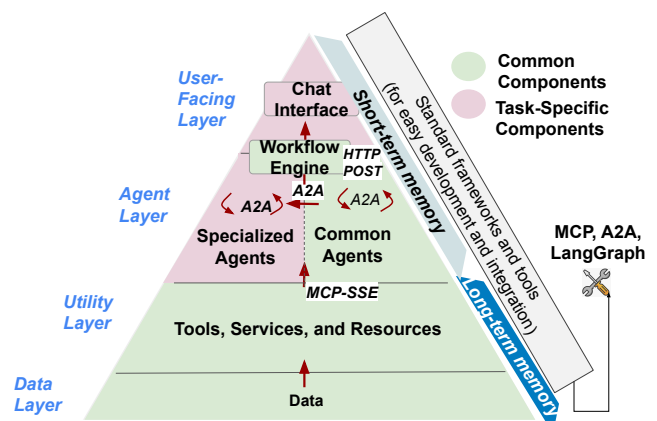


Figure 2: CausalPulse four-layer framework: user-facing, agent, utility, and data, connected through standardized protocols (MCP, A2A, and LangGraph).

User-Facing Layer: Provides a browser-based user interface (UI) through which operators interact with the copilot. It enables human-in-the-loop decision-making, allowing users to view copilot outputs and refine, validate, and adjust them based on domain expertise. The LangGraph-based Workflow Engine is positioned between the UI and Agent layers.

Agent Layer: Comprises two categories of agents: common agents, which are reusable across domains and applications, and specialized agents, which are tailored for specific causal diagnostic tasks. All agents are implemented as FastAPI microservices, described by agent cards, and coordinated through A2A communication with registry-based orchestration.

Utility Layer Provides MCP tools, services, and resources that underpin agent operations. Services integrate domain-informed services such as process ontologies, LLMs, knowledge graphs, curated technical manuals, and rule bases. Resources are artifacts, including agent cards, preprocessed documents, stored causal graphs, RCA results, and precomputed causal knowledge.

Data Layer Includes data collected from cyber-physical systems. It encompasses sensor and process logs, camera images, event and cycle metadata, process stages, and anomalies, which may take the form of time-series, images, or textual data. These heterogeneous inputs collectively serve as the basis for higher-level agentic operations.

Rationale and Design Principles. CausalPulse framework is conceived as a living ecosystem of *data*, *utilities*, *agents*, and *user-facing layer* to achieve modularity, explainability, and safe autonomy in complex industrial environments. Each layer in this vertical stack performs a distinct yet complementary role within the reasoning pipeline, establishing clean interfaces across human–AI–machine collaboration. The layered design bridges neural adaptability and symbolic control, enabling agents to learn from data while remaining constrained by domain knowledge. It supports plug-and-play extensibility, allowing new data sources, agents, and tools to be integrated without reconfiguring the overall workflow. The architecture ensures end-to-end transparency, allowing every decision to be traced from user query to raw data through clearly defined layer boundaries.

Agents and Their Capabilities

Common agents. The *Client Process Agent* (CPA) serves as the primary orchestrator between UI, and downstream agents. It first normalizes heterogeneous client inputs into canonical payloads that downstream tools can consume. Then, based on high-level task tags (e.g., *preprocessing*, *anomaly*, *causal*), it dispatches these payloads to the appropriate tools while managing asynchronous MCP sessions and routing execution requests. After tool execution, the CPA postprocesses results using metadata-defined rules, optionally triggering follow-up tools to support multi-step workflows.

The *Preprocessing Agent* performs deterministic data cleaning and normalization. It cleans the data by dropping empty rows and columns, standardizing types, imputing missing values, and label-encoding categorical variables into a

normalized CSV with stored encoder mappings. It queries the Background Info Agent to retrieve variable-level descriptions, thereby enriching the returned summary with domain semantics and improving the interpretability of subsequent diagnostics.

The *Background Info Agent* acts as a lightweight semantic lookup service that exposes curated metadata for dataset variables. Implemented as an HTTP-only microservice optimized for low-latency responses, it accepts lists of variable names and returns concise, human-readable descriptions that capture sensor semantics, measurement locations, and other domain-specific context. They help the Preprocessing and RCA agents turn numeric results into domain-aware explanations, improving operator interpretability and providing a consistent semantic layer across the diagnostic pipeline.

The *Recommender Agent* provides context-aware, next-step guidance to both human operators and the automated planner as diagnostic workflows progress. It uses the current tool and outputs to suggest the next analyses, evidence to inspect, or causes to prioritize, which the user can confirm or the planner can execute automatically. Its behavior is governed by deterministic, rule-based heuristics rather than opaque learned policies, ensuring predictable responses and low inference latency. Recommendations are exposed via the MCP SSE interface, enabling seamless discovery and consumption by the CPA and other A2A callers.

Specialized agents. The *Anomaly Detection Agent* encapsulates a multi-strategy pipeline for detecting abnormal sensor and process behavior. Based on the query content and available data modalities, the agent selects among several anomaly detection backends: a cross-modal fusion model combining time-series and image-based signals (Shyalika et al. 2025d), deterministic threshold rules derived from domain-specific measurement limits, and an isolation-forest model for unsupervised detection on tabular features. Its outputs comprise categorical labels (e.g., *Normal* vs. specific anomaly types), anomaly scores, and explicit identification of features that violate expected bounds, and traceability metadata. When an anomaly is detected, the agent automatically constructs a structured payload summarizing the event and triggers the RCA Agent via A2A communication.

The *Causal Discovery Agent* induces candidate causal structures from preprocessed data in a schema-aware, knowledge-infused manner. Based on the uploaded data, it selects an appropriate discovery routine—using Peter–Clark (PC) for gripper-sensor datasets and Greedy Equivalence Search (GES) for process-variable datasets to construct causal graphs. During structure learning, both algorithms are instantiated via the *causal-learn* (CausalLearn 2024) and *pgmpy* (pgmpy 2024) and are constrained by domain-specific rules on control/observation types and process order (Figure 3), ensuring that only physically plausible edges are considered. The resulting directed graphs and edge-level statistics, filtered by these domain constraints, can further be inspected and refined by factory-floor experts.

The *RCA Agent* is an MCP- and HTTP-capable component that performs causal attribution on anomalous observations. It

Event sequence: $G_4 \prec G_{10} \prec \dots$

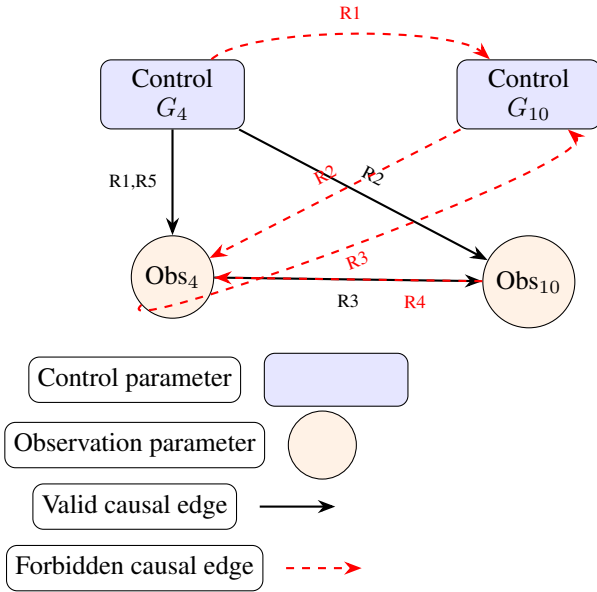


Figure 3: Illustration of domain-specific rules for a grinding process. Control parameters (rectangles) may cause observation parameters (circles) at the same or later grinding stage (Rules R1, R2, R5), observation parameters may only cause later observations (Rules R3, R4), and edges violating type or temporal order are explicitly disallowed (dashed red arrows).

accepts either in-memory records (reusing precomputed RCA results if available), an uploaded dataset, or a file path, and returns structured, ranked root-cause explanations suitable for operator interpretation and downstream tooling. The agent wraps the *ProRCA* analysis pipeline—a domain-informed SCM builder and path-based root-cause scorer (Dawoud and Talupula 2025). It outputs ranked candidate causes, per-node scores, and enumerated causal paths. Scoring combines structural and noise-based measures to balance causal plausibility with observed perturbation strength, enhancing operator trust and guiding follow-up diagnostics (Figure 5).

Agentic Task Flow Orchestration and Workflows

The workflow of CausalPulse includes three stages.

Stage 1: Agent card registration (Figure 4:b-I): Every agent registers its agent card on the central server. This registration process allows the central server to maintain a live directory of all available agents, enabling dynamic discovery and orchestration across the system

Stage 2: Agent interactions and data flow (Figure 4:b-II): Once the agents are registered, the workflow begins when a user query is submitted through the interface. Within the workflow engine (Figure 4a), the *planner node* receives the user’s query together with the current workflow state. It queries an LLM (llama-3.1-8b-instant) to determine the minimal sequence of agents required to address the query, such as preprocessing, anomaly detection, causal discovery, and RCA. This process constitutes dynamic task

decomposition, ensuring that redundant steps are avoided, such as reapplying preprocessing when it has already been completed. The node then updates the system state with the planned steps and returns this structured plan to the workflow.

The *executor node* is responsible for carrying out the planned steps. For every stage the executor sends a structured request to the CPA. CPA dispatches the request to the correct remote agent via the MCP protocol and SSE connection. It returns the agent’s output back to the executor, which merges results into the shared system state before moving on to the next stage. For efficiency, previously completed stages are skipped, such as preprocessing when it has already been performed. The executor also ensures proper data handling by forwarding file names and data paths, which is particularly important for consecutive agents. After execution, results are appended into the appropriate state slot (e.g., `state["anomaly"]`, `state["RCA"]`), and the workflow continues to the next step.

The *replanner node* is invoked after each execution to evaluate if the plan requires modification. It queries the LLM to assess whether additional stages are needed based on the updated state and prior results. For instance, if anomaly detection indicates a “normal” status, the system excludes unnecessary stages such as RCA or causal discovery. The replanner then returns an updated list of remaining steps, ensuring that the workflow adapts dynamically to the context.

The LangGraph workflow construction specifies the orchestration of these nodes. The workflow begins with the planner, proceeds to the executor, and transitions to the replanner as long as steps remain. If additional steps are identified, the cycle continues with replanning and execution. If no steps remain, the workflow terminates.

Stage 3: Postprocess chaining (Figure 4:b-III): After the primary agents complete their tasks, the Recommender Agent is automatically activated. It reviews the aggregated outputs and the system’s current state, then suggests the next best actions. Upon receiving these recommendations, the CPA validates and merges them into the shared workflow state, exposes the suggested actions through the UI, and evaluates any postprocess rules defined in the agent cards. When auto-chaining is permitted, the CPA resolves mapped inputs and triggers the corresponding downstream agents via standardized MCP calls. Returned results are integrated into the global state and recorded as entries for traceability. If new outputs alter the workflow context, the CPA invokes the replanner node to dynamically update the execution plan, ensuring adaptive, continuous reasoning.

Knowledge Representation and Semantics

CausalPulse grounds its agentic workflows in semantic artifacts capturing variable meaning, process structure, and causal constraints.

Variable-Level Semantics. CausalPulse maintains curated metadata for each sensor or process variable, including textual descriptions, units, physical location, subsystem membership, and high-level type (e.g., control vs. observation, quality vs. condition). This metadata is stored in as JSON and accessed through the Background Info Agent, which exposes a

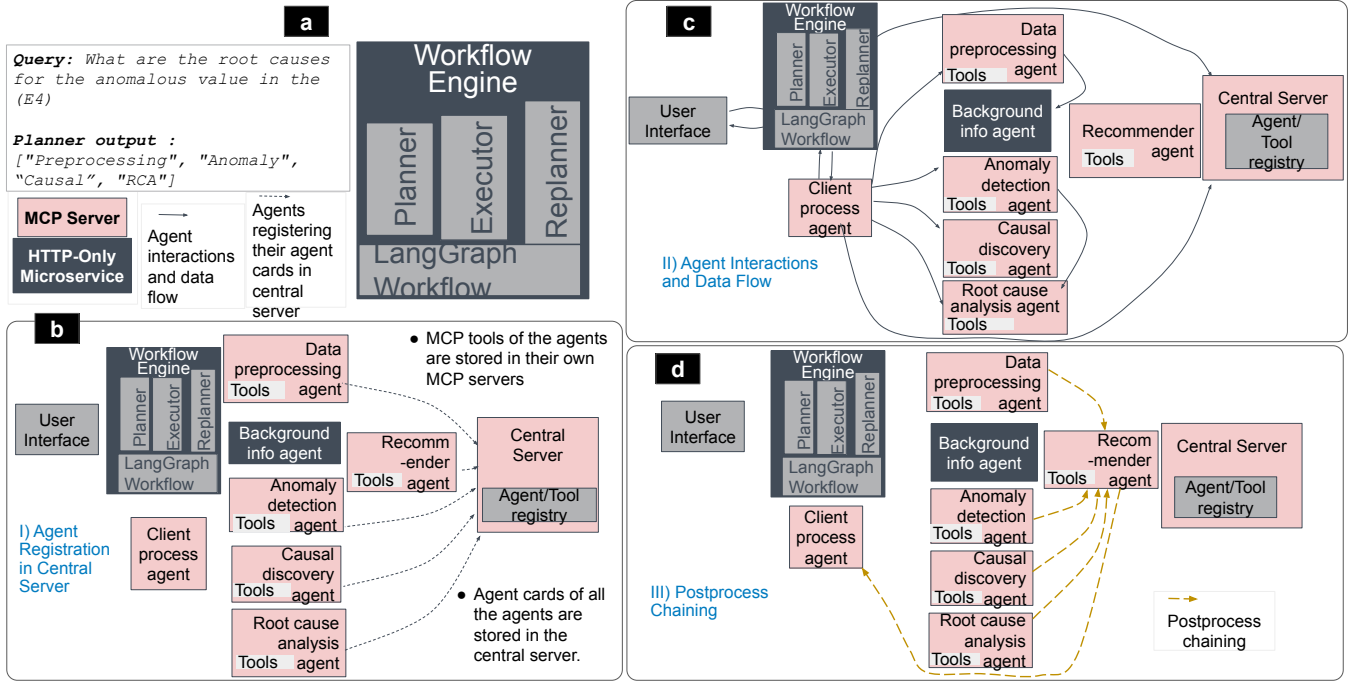


Figure 4: Agentic task orchestration and workflow execution in CausalPulse: (a) Query interpretation and planning by the LangGraph-based workflow engine, (b-I) Agent card registration and central server coordination, (b-II) Agent interactions and data flow during execution, and (b-III) Postprocess chaining with the Recommender Agent.

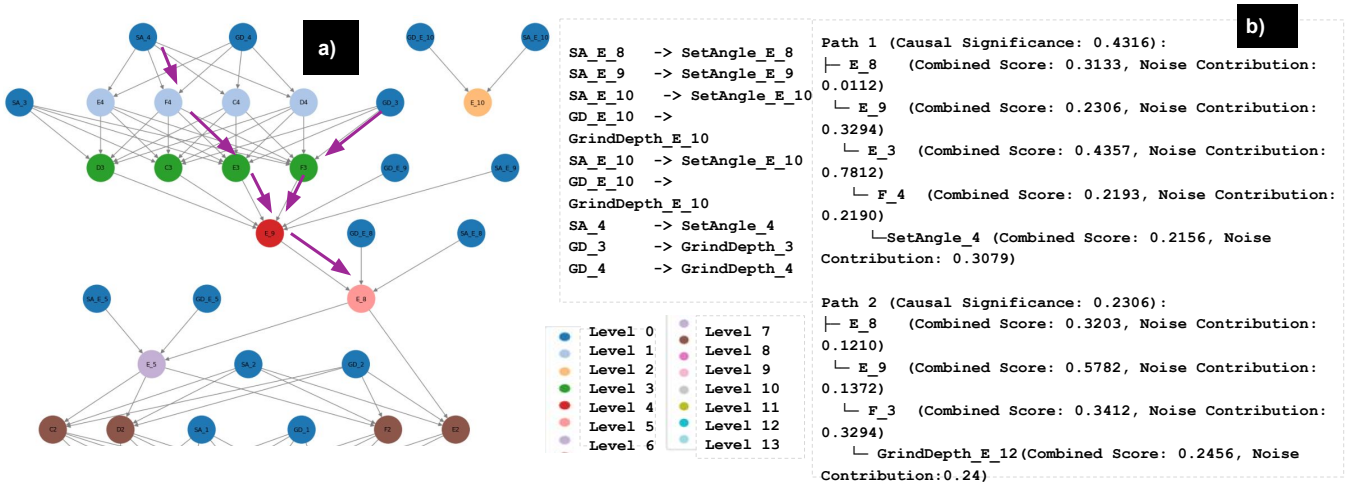


Figure 5: Root cause analysis results: (a) ranked root-cause paths ordered by causal significance for the anomaly in E.8 in the PSE dataset (refer to "Datasets Descriptions" Section), and (b) visual representation of the corresponding causal graph, highlighting identified root causes and propagation paths. The legend on the right lists variable names and their hierarchical levels, where levels represent the sequential order of grinding steps in the PSE use case.

simple lookup API from variable names to human-readable descriptions. The Preprocessing Agent uses this catalog to enrich preprocessing summaries with semantic context, while the RCA and Causal Discovery Agents incorporate it into their explanations.

Process and Asset Structure. Process-level semantics are encoded as directed graphs covering process stages and asset hierarchies (machine → subsystem → sensor) defining the temporal sequence and stage-specific control/observation variables. They help agents filter relevant variables, group them by stage, and give the planner a coarse sense of earlier vs. later process regions when building workflows.

Causal and Operational Rules. Domain experts define rule sets that restrict which causal relations between variables are admissible. Example rules for PSE grinding process are visualized in Figure 3. These rules are encoded in a compact declarative schema over variable types and stage indices and enforced as hard constraints during causal structure learning. The Causal Discovery Agent uses them to filter candidate edges from PC and GES, retaining only rule-consistent relationships, which reduces spurious correlations and yields causal graphs that are physically plausible and aligned with operators’ mental models of the line.

Generalization to New Lines. Onboarding a new manufacturing line requires (i) ingesting its process topology and stage ordering into the process graph schema, (ii) populating the variable catalog with descriptions and type labels, and (iii) eliciting a small number of control/observation and order rules from domain experts. Once registered, these artifacts are consumable by agents and planners without architectural changes, allowing semantic grounding and knowledge-infused causal discovery to carry over across lines and sites.

Results and Evaluation

Datasets Descriptions

Planar Sensor Element (PSE) dataset. The PSE dataset is collected from Bosch’s Anderson Plant (AdP) in South Carolina, USA, where planar oxygen sensors are manufactured. It captures the grinding process, a precision operation critical to product quality and yield. This process spans 18 robotic stations with dual-arm configurations performing 12 sequential grind types (G_1 – G_{12}). Each grind is defined by parameters such as angle, depth, duration, and speed, with repetitions possible upon anomaly detection or operator intervention. For each grind, distances (mm) and angles (degrees) are measured at multiple surface positions, producing 26 variables through optical inspection and machine sensors. Each feature includes defined tolerance limits that adjust dynamically for regrinds to avoid over-polishing or material stress.

Future Factories (FF) Dataset. We use the publicly available manufacturing dataset generated by the Future Factories (FF) Lab at the McNair Aerospace Research Center, University of South Carolina (Harik et al. 2024). This dataset captures synchronized sensor and image data from a prototype rocket assembly pipeline designed to emulate industrial-grade manufacturing processes. Each cycle is segmented into 21 distinct operational states. The dataset includes time-series measurements from potentiometers, load cells, drive temperatures, robot kinematics and synchronized image data captured from two cameras. These anomalies indicate missing rocket components and include six types, such as NoNose, NoBody2, and combined cases like NoBody2, NoBody1.

CausalPulse delivers interpretable results, reducing manual adjustments in high-cost manufacturing, with a 4% scrap rate (\approx \$ 15M annually) in the PSE dataset and a \approx 40% anomaly rate in the FF dataset.

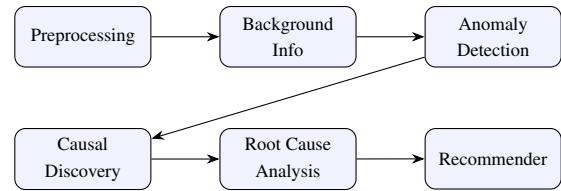


Figure 6: Canonical agent pipeline in CausalPulse. Predefined workflows correspond to subpaths of this pipeline with optional reuse of cached intermediate results.

Evaluation Setup

We evaluated CausalPulse’s core agentic capabilities using predefined workflows for diagnostic queries (Figure 6, Table 1). The responses were analyzed to determine the number of expected and predicted reasoning steps and agents, along with successful and unsuccessful attempts. We computed three quantitative metrics, *Criterion success (%)*, *Aggregate criterion success (%) per stage*, and the *Aggregate success rate per agentic criterion*. The *Criterion success (%)* represents the proportion of successful executions for each agentic criterion, such as planning, tool use, self-reflection, and multi-agent collaboration, calculated as the ratio of successful to total queries. The *Aggregate criterion success (%) per stage* summarizes the overall success rate of a criterion type across all diagnostic stages within each dataset, computed as the ratio of total successful attempts to total queries for that criterion within a workflow stage. *Aggregate success rate per agentic criterion* computes the combined reliability of each criterion type aggregated across all workflow stages. A high *Criterion success (%)* indicates close alignment between CausalPulse’s predicted reasoning trajectory and the expected expert-designed plan, while high *Aggregate criterion success (%)* and *Aggregate success rate per agentic criterion* reflect robust and reliable multi-agent orchestration across workflows (Figure 7).

Evaluating Planning Capability: We assessed the ability of CausalPulse’s workflow engine to correctly decompose user intents into coherent subtasks following canonical workflow patterns. Across both datasets, the system achieved a planning success rate of 98.75% (PSE) and 98.12% (FF). The planner generated the correct sequence of reasoning steps and agents, with minor discrepancies in cases where intermediate causal diagnostics steps were omitted due to incomplete contextual cues.

Evaluating Tool Use: Tool invocation success rate was evaluated by comparing the tools executed per query against ground-truth workflows. CausalPulse achieved tool-use success rates of 98.75% (PSE) and 98.1% (FF), reflecting reliable orchestration of MCP tools through LangGraph and A2A protocols.

Evaluating Self-Reflection: Self-reflection is evaluated as the system’s ability to introspect on prior actions and reuse intermediate results instead of recomputing them. CausalPulse exhibited three reflective behaviors: (i) dynamically pruning unnecessary tasks when upstream outputs render them obsolete (e.g., skipping RCA when the anomaly detector returns “normal”), (ii) avoiding redundant preprocessing and causal

ID	Workflow path (agents in order)
W1	Preprocessing → Recommender
W2	Preprocessing → Background Info → Recommender
W3	Background Info → Recommender
W4	Preprocessing* → Anomaly Detection → Recommender
W5	Preprocessing* → Background Info → Causal Discovery → Recommender
W6	Preprocessing* → Anomaly Detection → Causal Discovery* → RCA → Recommender
W7	Preprocessing* → Anomaly Detection → Causal Discovery → Recommender
W8	Preprocessing → Anomaly Detection → Background Info → Causal Discovery → RCA → Recommender
W9	Preprocessing* → Anomaly Detection → Causal Discovery* → RCA → Recommender

Table 1: Predefined workflows in CausalPulse as paths through the canonical agent pipeline in Figure 6. Steps marked * may reuse cached results if available.

discovery steps when prior results are available, and (iii) reusing cached intermediate results for recurrent subqueries. Empirical analysis shows a reflective success rate of 97.3% for both datasets, while redundant computation was reduced by approximately 22% across repeated workflow executions.

Evaluating Multi-Agent Collaboration: Multi-agent collaboration was assessed based on the system’s ability to invoke the correct subset of agents and exchange intermediate results through standardized A2A communication. The collaboration success rate reached 99.2% (PSE) and 98.3% (FF), demonstrating robust coordination and information sharing across agents. Occasional collaboration lapses were limited to ambiguous user intents where the recommender agent prematurely finalized responses before RCA completion.

System achieved high criterion-level performance across both datasets, with per-criterion success rates ranging from 95–100%. The aggregate criterion success reached 98.73% for PSE and 98.0% for FF, indicating stable and reliable multi-agent execution across workflow stages.

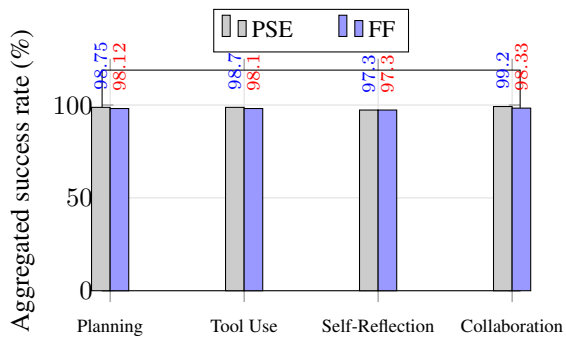


Figure 7: Aggregated success rate per agentic criterion for PSE and FF datasets.

Runtime and Scalability Analysis: Runtime experiments were performed for the same set of diagnostic queries on a MacBook Pro (Apple M1 Pro, 8-core CPU, 16 GB RAM, macOS Monterey 12.7.5). On the PSE dataset ($M = 5$ variables, $N \approx 1.0$ M samples), data preprocessing averaged 15s, anomaly detection 5s, causal discovery with the knowledge-infused GES algorithm 12s, and RCA ≈ 21 s. End-to-end latency from query submission to RCA was ≈ 53 s per request. On the FF dataset ($M = 5$ variables, $N \approx 1.0$ M samples), preprocessing averaged 16s, anomaly detection 6s, causal discovery 15s, and RCA ≈ 23 s, with a total latency of ≈ 60 s. End-to-end latency represents total wall-clock time across all pipeline stages. To assess scalability, M was increased from 5 to the full dimensionality of each dataset. Runtime grew approximately linearly with M (coefficient of determination $R^2 = 0.97$), confirming near-linear scaling of the microservice architecture and asynchronous task queue.

Root-Cause Attribution Verification and Quantitative Evaluation To verify the quality of PRORCA’s root-cause attribution, we manually inspected anomalous events and annotated the set of “immediate variables changed” at the anomaly timestamp as ground-truth causes. For each event, we compared the ranked variable list produced by PRORCA against these annotations using two families of metrics: (i) rank-based $hits@k$, which checks whether any true cause appears within the top- k positions of the PRORCA ranking, and (ii) set-based precision, recall, and F1 between the predicted causes and the annotated set. Across all anomaly instances in the PSE dataset, PRORCA attains Hits@1 = 0.44 and Hits@2 = 0.65, rising to 0.73 for $k \geq 3$. It achieves an average precision of 0.46, recall of 0.40, and F1 of 0.39 aggregating all anomalies in PSE dataset. Representative qualitative outcomes for three anomalies are shown in Table 2. PRORCA frequently surfaces at least one correct causal variable near the top of its ranking, but still misses a non-trivial fraction of annotated causes, leaving room for improving recall while preserving its relatively sharp rankings.

Qualitative Comparison with Existing Copilots: To situate CausalPulse within the landscape of industrial copilots and multi-agent systems, we qualitatively compare it against four representative systems: UCSD Causal Copilot (U) (Wang et al. 2025), MATMCD (Shen et al. 2024) (M), IBM AssetOpsBench (Patel et al. 2025)(A), and SmartPilot (S) (Shyalika et al. 2025c) (Table 3). UCSD and MATMCD primarily address offline causal graph refinement or simulated fault analysis with single or small sets of agents and fixed pipelines, while AssetOpsBench focuses on multi-agent simulation for asset-management scenarios rather than operator-facing decision support. SmartPilot offers a neurosymbolic multi-agent copilot, but its interaction patterns are mostly predefined and not grounded in standardized agentic protocols. CausalPulse is designed as an industry-grade multi-agent copilot that (i) implements a full end-to-end causal diagnostics pipeline (from preprocessing and anomaly detection through causal discovery, RCA, and recommendation), (ii) leverages standardized protocols for dynamic agent discovery, context-aware replanning, and real-time inter-agent communication, and (iii) is currently undergoing deployment on a live manufacturing line with a dashboard.

Anom. GT	RCA pred.	Hits@k	Prec.	Rec.	F1	
F4	{SA_3, GD_3, GD_4}	{SA_4, GD_4}	hits@1=0, hits@2=1	0.50	0.33	0.40
E12	{GD_E_9, GD_E_12}	{SA_3, GD_E_12, GD_E_9, SA_4, SA_2}	hits@1=0, hits@2=1, hits@3=1, hits@4=1, hits@5=1	0.40	1.00	0.57
E3	{SA_3, GD_3}	{GD_3, SA_3, SA_4, GD_4}	hits@1=1, hits@2=1, hits@3=1, hits@4=1	0.50	1.00	0.67

Table 2: Representative PRORCA root-cause attribution (RCA pred.) outcomes on the PSE dataset, illustrating (top) a partial match where only one of three ground-truth causes appears in the top-2 positions, (middle) a high-recall but noisy case where all ground-truth causes are recovered along with several false positives, and (bottom) a strong match where both ground-truth causes are ranked in the top-2 positions. GT: Ground Truth, SA:SetAngle, GD: GrindDepth

	(U)	(M)	(A)	(S)	CausalPulse
1. System Type	SAG	MAD	MAS	MAC	MAC
2. End-to-end causal analysis	✗	✗	✗	✗	✓
3. Dynamic multi-agent orchestration (A2A, replanning)	✗	✗	✗	✗	✓
4. Knowledge-grounded / Neurosymbolic reasoning	✗	✗	✗	✓	✓
5. Live /in-factory deployment underway	✗	✗	✗	✓	✓
6. Operator-facing dashboard	✗	✗	✗	✓	✓

Table 3: Summary comparison of CausalPulse with representative industrial copilots. SAG: Single-Agent, MAD: Multi-Agent Discovery, MAS: Multi-Agent Simulation, MAC: Multi-Agent Copilot

Pathway to Deployment

CausalPulse is currently being deployed on a Bosch production line, with the full deployment pipeline shown in Figure 8. At the factory level, an *OPC UA Server (Open Platform Communications Unified Architecture)* aggregates live data from heterogeneous sources. The *OPC UA Client* of CausalPulse receives and processes these streams, standardizing and structuring signals for agent-level consumption. The data are then integrated into the *utility layer*, which hosts and retrieves domain knowledge, rule-based constraints, MCP tools, agent

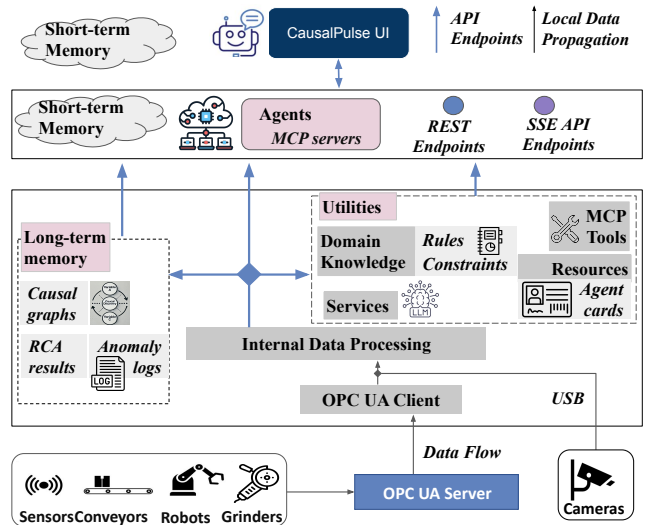


Figure 8: End-to-end deployment pipeline of CausalPulse

cards, and long-term memory. These are persistently accessed through standardized APIs and invoked by agents for real-time analytics and coordination. The *agent layer* hosts agents as MCP servers and exposes REST and SSE API endpoints, enabling both human operators and other digital agents to interact with services. The short-term memory resides between the UI and agent layers, caching transient session-level data such as current workflow states, variable contexts, and intermediate results to enable fast context sharing during live interactions. *CausalPulse UI* is a browser-based interface that connects to the backend through FastAPI and LangGraph orchestration. The architecture supports local data propagation with minimal latency, while maintaining traceability and transparency across all layers.

From the ongoing deployment experience, we identify key challenges in integrating agentic reasoning into safety-critical manufacturing pipelines. These include ensuring protocol interoperability across heterogeneous systems, calibrating operator trust through explainable outputs, and maintaining latency control under real-time production constraints. These challenges have directly informed the design of CausalPulse’s modular architecture and protocol-first orchestration strategy, which together enable safe and reliable industrial deployment. Ongoing work focuses on extending these capabilities to distributed, edge-enabled configurations for even greater scalability and responsiveness.

Conclusion and Future Work

CausalPulse is an industry-grade neurosymbolic multi-agent copilot for causal diagnostics, with ongoing extensions to include an Optimization Agent that enables automated, explainable process interventions. Future work focuses on edge deployment for low-latency distributed operations and reinforcement learning-based adaptation for autonomous calibration of thresholds and causal priors.

References

2025. Agent Communication Protocol. <https://agentcommunicationprotocol.dev/introduction/welcome>. Accessed: 2025-10-17.
- Anthropic. 2024. Introducing the Model Context Protocol. <https://www.anthropic.com/news/model-context-protocol>. Accessed: 2025-10-15.
- Bandi, A.; Kongari, B.; Naguru, R.; Pasnoor, S.; and Vilipala, S. V. 2025. The Rise of Agentic AI: A Review of Definitions, Frameworks, Architectures, Applications, Evaluation Metrics, and Challenges. *Future Internet*, 17(9): 404.
- Bellifemine, F.; Poggi, A.; and Rimassa, G. 2000. Developing multi-agent systems with JADE. In *International workshop on agent theories, architectures, and languages*, 89–103. Springer.
- Bousetouane, F. 2025. Agentic systems: A guide to transforming industries with vertical ai agents. *arXiv preprint arXiv:2501.00881*.
- CausalLearn. 2024. CausalLearn Documentation. <https://causal-learn.readthedocs.io/en/latest/>. Accessed: 2025-10-17.
- Chickering, D. M. 2002. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov): 507–554.
- CrewAI. 2025. CrewAI Documentation. Accessed: 2025-10-28.
- Dawoud, A.; and Talupula, S. 2025. ProRCA: A Causal Python Package for Actionable Root Cause Analysis in Real-world Business Scenarios. *arXiv preprint arXiv:2503.01475*.
- Derouiche, H.; Brahmi, Z.; and Mazeni, H. 2025. Agentic AI Frameworks: Architectures, Protocols, and Design Challenges. *arXiv preprint arXiv:2508.10146*.
- Google A2A. 2025. A2A: A New Era of Agent Interoperability. <https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/>. Accessed: 2025-10-15.
- Gort, B.; Kibalya, G. M.; and Antonopoulos, A. 2025. Agent-Edge: Agentic AI for Service Orchestration in the Edge-Cloud Continuum. *Authorea Preprints*.
- Harik, R.; Kalach, F. E.; Samaha, J.; Clark, D.; Sander, D.; Samaha, P.; Burns, L.; Yousif, I.; Gadow, V.; Tarekegne, T.; and Saha, N. 2024. Analog and Multi-modal Manufacturing Datasets Acquired on the Future Factories Platform. *arXiv preprint arXiv:2401.15544*.
- Ko, H.; Lu, Y.; Yang, Z.; Ndiaye, N. Y.; and Witherell, P. 2023. A framework driven by physics-guided machine learning for process-structure-property causal analytics in additive manufacturing. *Journal of Manufacturing Systems*, 67: 213–228.
- Kuo, Y.-H.; Pilati, F.; Qu, T.; and Huang, G. Q. 2021. Digital twin-enabled smart industrial systems: Recent developments and future perspectives. *International Journal of Computer Integrated Manufacturing*, 34(7-8): 685–689.
- LangChain AI. 2024. LangGraph: A Framework for Building Agentic Applications. <https://www.langchain.com/langgraph>. Accessed: 2025-10-17.
- Marro, S.; La Malfa, E.; Wright, J.; Li, G.; Shadbolt, N.; Wooldridge, M.; and Torr, P. 2024. A scalable communication protocol for networks of large language models. *arXiv preprint arXiv:2410.11905*.
- Patel, D.; Lin, S.; Rayfield, J.; Zhou, N.; Vaculin, R.; Martinez, N.; O'donncha, F.; and Kalagnanam, J. 2025. AssetOps-Bench: Benchmarking AI Agents for Task Automation in Industrial Asset Operations and Maintenance. *arXiv preprint arXiv:2506.03828*.
- Pearl, J. 2019. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3): 54–60.
- pgmpy. 2024. pgmpy: Probabilistic Graphical Models using Python. <https://pgmpy.org/index.html>. Accessed: 2025-10-17.
- Pietsch, D.; Matthes, M.; Wieland, U.; Ihlenfeldt, S.; and Munkelt, T. 2024. Root cause analysis in industrial manufacturing: A scoping review of current research, challenges and the promises of AI-driven approaches. *Journal of Manufacturing and Materials Processing*, 8(6): 277.
- Ribeiro, L.; and Gomes, L. 2021. Describing structure and complex interactions in multi-agent-based industrial cyber-physical systems. *IEEE Access*, 9: 153126–153141.
- Sawant, P. D. 2025. Unified, Modular, Self-Learning, Real-Time Agentic AI Ecosystem: Integrating Multi-Modal Data, Automation, Feedback Loops, and Secure Agent Orchestration. *Recent Trends in Artificial Intelligence & Its Applications*, 4(2): 53–61. Founding Director, AI Research & Director, AI-Discovery Consultancy, Melbourne, Australia.
- Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36: 68539–68551.
- Shen, C.; Chen, Z.; Luo, D.; Xu, D.; Chen, H.; and Ni, J. 2024. Exploring multi-modal integration with tool-augmented llm agents for precise causal discovery. *arXiv preprint arXiv:2412.13667*, 1(3).
- Shyalika, C.; Prasad, R.; Al Ghazo, A.; Eswaramoorthi, D.; Kaur, H.; Muthuselvam, S. S.; and Sheth, A. 2025a. SmartPilot: A Multiagent CoPilot for Adaptive and Intelligent Manufacturing. In *2025 IEEE Conference on Artificial Intelligence (CAI)*, 1–8. IEEE.
- Shyalika, C.; Prasad, R.; Al Ghazo, A.; Eswaramoorthi, D. L.; Shree Muthuselvam, S.; and Sheth, A. 2025b. SmartPilot: Agent-Based CoPilot for Intelligent Manufacturing. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, 3053–3055.
- Shyalika, C.; Prasad, R.; Al Ghazo, A.; Eswaramoorthi, D. L.; Shree Muthuselvam, S.; and Sheth, A. 2025c. SmartPilot: Agent-Based CoPilot for Intelligent Manufacturing. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems*, 3053–3055.
- Shyalika, C.; Prasad, R.; El Kalach, F.; Venkataramanan, R.; Zand, R.; Harik, R.; and Sheth, A. 2025d. NSF-MAP: Neurosymbolic Multimodal Fusion for Robust and Interpretable

Anomaly Prediction in Assembly Pipelines. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI) 2025*. Special Track on AI4Tech: AI Enabling Critical Technologies; currently available as an arXiv preprint arXiv:2505.06333.

Shyalika, C.; Sharma, A.; Kalach, F. E.; Jaimini, U.; Henson, C.; Harik, R.; and Sheth, A. 2026. CausalTrace: A Neurosymbolic Causal Analysis Agent for Smart Manufacturing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40.

Sofianidis, G.; Rožanec, J. M.; Mladenec, D.; and Kyriazis, D. 2021. A review of explainable artificial intelligence in manufacturing. *Trusted Artificial Intelligence in Manufacturing: A Review of the Emerging Wave of Ethical and Human Centric AI Technologies for Smart Production*, 93–113.

Spirtes, P.; Glymour, C. N.; and Scheines, R. 2000. *Causation, prediction, and search*. MIT press.

Trivedi, C.; Bhattacharya, P.; Prasad, V. K.; Patel, V.; Singh, A.; Tanwar, S.; Sharma, R.; Aluvala, S.; Pau, G.; and Sharma, G. 2024. Explainable AI for industry 5.0: vision, architecture, and potential directions. *IEEE Open Journal of Industry Applications*, 5: 177–208.

Wang, X.; Li, B.; Song, Y.; Xu, F. F.; Tang, X.; Zhuge, M.; Pan, J.; Song, Y.; Li, B.; Singh, J.; et al. 2024. Opendevin: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*, 3.

Wang, X.; Zhou, K.; Wu, W.; Singh, H. S.; Nan, F.; Jin, S.; Philip, A.; Patnaik, S.; Zhu, H.; Singh, S.; et al. 2025. Causal-copilot: An autonomous causal analysis agent. *arXiv preprint arXiv:2504.13263*.

Wooldridge, M. 2009. *An introduction to multiagent systems*. John wiley & sons.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.