# Causal-Aware LLM Agents for PHM Co-Pilots: Health Monitoring and Intervention Planning

Rajarajan Kirubanandan[1]

[1] *Independent Researcher, New York, NY, 10705, USA*
*rajarajankirubanandan@gmail.com*

**ABSTRACT**

Large language models (LLMs) can generate plausible diagnostic plans and corrective actions from sensor data, but they lack intrinsic causal reasoning and cannot assess whether their recommendations will lead to successful issue remediation. This limitation is especially critical in high stakes domains like Prognostics and Health Management (PHM), where decisions must be both interpretable and causally grounded. While LLMs excel at generating contextually relevant outputs, they cannot be relied upon to evaluate their own plans. To address this gap, we propose a hybrid framework that integrates LLM based planning with structured causal inference. The system retrieves top-k similar historical traces based on the current sensor context and operational settings and constructs a localized causal structure from these matches to simulate and evaluate the impact of potential actions. Recommendations are ranked based on their estimated effect on resolution likelihood and are causally validated within the same agentic workflow. Our results demonstrate that augmenting LLM generated plans with external causal inference significantly improves relevance, consistency, and safety, offering a deployable blueprint for PHM scenarios where LLMs alone cannot be trusted to reason reliably.

## 1. INTRODUCTION

PHM plays a pivotal role in modern industrial systems by enabling early detection of faults, estimation of remaining useful life, and recommendation of corrective actions to prevent failures. Traditional PHM systems rely heavily on statistical modelling, signal processing, and machine learning techniques trained on historical sensor data. While these approaches have demonstrated success in detecting anomalies and predicting failures, they often fall short in providing interpretable reasoning, especially when tasked with recommending intervention strategies under uncertainty.



With the rise of LLMs, there has been growing interest in leveraging their reasoning capabilities and domain adaptability for industrial AI applications. LLMs, pretrained on vast corpora of technical documentation and maintenance logs, offer a new paradigm for building intelligent copilots that can understand system behaviour, suggest diagnostic paths, and generate maintenance plans. In high stakes maintenance environments, reliable decision making requires more than just identifying similar past cases it demands the ability to estimate what would happen under alternative intervention strategies. This involves answering counterfactual questions (e.g., What if a different diagnostic step had been taken?") and isolating true causal effects from confounding influences. Traditional similarity based retrieval systems, commonly employed in recent LLM based PHM copilots (Lukens, McCabe, Gen, & Ali, 2024), are not equipped to handle such tasks.

While these systems offer rapid access to semantically relevant historical cases, they operate on surface level correlations and provide no guarantees about intervention effectiveness. Without causal modelling, they cannot adjust for latent factors that may influence both failures and recommended actions, nor can they simulate the outcomes of untested alternatives (Pearl, 2009). As we emphasize, LLMs trained on observational text can mimic causal language but lack the structural foundation to perform genuine causal inference (Zečević, Willig, Dhami, & Kersting, 2023). These limitations motivate the need for a causally aware PHM framework one that augments LLM based reasoning. To bridge this gap, we propose a proof of concept causal aware LLM agent architecture that integrates the reasoning power of LLMs with structured causal inference mechanism.

The agent is designed to assist in real-time PHM tasks by:

1. Inferring sensor-level triggers from observed data,
2. Retrieving similar historical cases based on operating conditions and trigger profiles,
3. Constructing a localized causal graph from the retrieved cases, and
4. Applying do-interventions on candidate diagnostic or corrective actions to simulate causal effects, followed by

counterfactual reasoning to evaluate the likelihood of issue resolution under alternative actions.

This framework creates a hybrid pipeline where the LLM proposes diagnostic or repair plans, and a causal inference engine evaluates, and ranks them based on estimated treatment effects and causal metrics. The result is an interpretable, adaptive, and causally grounded PHM co-pilot that can recommend context-sensitive interventions. We evaluate our approach on synthetic extensions of the NASA CMAPSS dataset (Saxena, Goebel, Simon, & Eklund, 2008), demonstrating the agent's ability to simulate interventions and recommend high-impact corrective actions. Through a series of experiments, we highlight the benefits of local causal reasoning and counterfactual simulation in improving both accuracy and transparency in PHM decision making.

## 2. BACKGROUND AND RELATED WORK

### 2.1. Large Language Models in Industrial AI

(Lukens et al., 2024) demonstrated the feasibility of using LLM agents to support maintenance troubleshooting workflows in industrial Prognostics and Health Management (PHM) systems. Their framework introduced a modular copilot design where a Recommender agent generates structured diagnostic steps in response to sensor anomalies, and an Evaluator agent assesses whether each step would successfully uncover a known failure mode. The system leveraged historical PHM case data and evaluated agent behaviour through subject matter expert review. Notably, incorporating Retrieval Augmented Generation (RAG) by providing the Recommender agent with semantically similar historical cases improved failure detection rates and reduced the number of diagnostic steps. Their results showed that LLMs could automate parts of the PHM process with high accuracy and suggested improvements when using agent based decomposition, RAG grounding, and structured prompting. This dual agent architecture offers a foundation for integrating LLMs into operational maintenance settings by augmenting, rather than replacing, human expertise. (Lukens et al., 2024) proposed a multi-agent PHM Copilot architecture where large language model agents are orchestrated to automate tasks such as alert summarization, failure explanation, and prescriptive step generation. Their system demonstrated the utility of structured prompting and function calling APIs, with domain grounding provided by case data retrieval. This work established the feasibility of applying LLMs to industrial diagnostics and inspired extensions to causal modelling and intervention planning, as explored in our study.

### 2.2. Short Comings of the Similarity Based Searches

Similarity based retrieval systems, such as those employed in LLM driven PHM frameworks, identify and reuse past cases that appear semantically or operationally similar to the current system state. While this approach provides intuitive and fast access to historical patterns, it suffers from fundamental limitations. First, similarity does not imply causality; actions that resolved failures in prior cases may have done so under different latent conditions and may not generalize to the current context. Second, such systems offer no guarantees about intervention effectiveness, as they fail to estimate counterfactual outcomes what would have happened if a different action were taken. Third, without modelling confounding variables, similarity search may reinforce existing biases in the data (e.g., commonly chosen actions may appear repeatedly not because they are optimal, but because they were frequently applied). Consequently, decisions based solely on similarity can lead to misleading or suboptimal recommendations, particularly in high-stakes or dynamically evolving environments. To overcome these limitations, our framework integrates causal inference to simulate interventions, estimate treatment effects, and ground recommendations in counterfactual reasoning enabling decisions that are both data driven and causally sound.

### 2.3. Causal Inference in Industrial AI

Causal inference is increasingly being explored across various domains of industrial AI, particularly in scenarios where traditional correlation based methods fall short in guiding effective interventions. One such example is the work of (Diehl & Ramirez-Amaro, 2021), who present a causal based framework for predicting and preventing failures in robotic manipulation tasks. Their method leverages Bayesian networks learned from simulation data to model the causal relationships between action parameters and task outcomes. The system predicts failure likelihood given the current state and, when necessary, identifies corrective actions by searching for alternative parameter configurations using contrastive reasoning. A key contribution of their work is the handling of temporally shifted failure cases where an early action affects the outcome of later steps by capturing causal dependencies across the full sequence of actions. Their results show significant reductions in failure rates for both single and multi-step stacking tasks, highlighting the value of causal modelling for error prevention in robotics.

(Vanderschueren, Boute, Verdonck, Baesens, & Verbeke, 2022) introduced a compelling shift in preventive maintenance strategy by proposing a prescriptive framework rooted in causal machine learning. Traditional approaches often treat maintenance effects as uniform, assuming every machine responds similarly to scheduled interventions. In contrast, their method embraces the individuality of machines, recognizing that maintenance outcomes can vary based on each machine's unique operational context. Their framework models overhauls and failures as potential outcomes influenced by the frequency of preventive maintenance (PM). To make truly individualized decisions, they optimize a cost function that

balances the expenses of performing PM, dealing with failures, and executing full overhauls. At the heart of their approach lies SCIGAN, a generative adversarial network designed for continuous treatments. This model enables accurate counterfactual outcome prediction that is, estimating what would have happened under a different maintenance schedule and also corrects for selection bias inherent in observational datasets.

Tested on a dataset of more than 4,000 industrial maintenance contracts, their causal approach significantly outperformed traditional supervised predictive models and generalized average-effect policies. The results underscore the value of learning individualized treatment effects. Policies tailored to the causal behaviour of each machine led to better predictive accuracy and greater cost efficiency, redefining what optimal maintenance can look like in industrial operations.

(Yu & Smith, 2017) propose the use of Causal Chain Event Graphs (CEGs) for modelling remedial maintenance processes, offering a structured graphical formalism to represent event driven system deterioration and recovery. Unlike traditional Bayesian networks, CEGs are designed to capture asymmetric, sequential event progressions and context-specific dependencies commonly seen in real world engineering systems. The authors define formal rules for modelling different types of maintenance intervention perfect, imperfect, and uncertain, and extend Pearl's do-calculus and the back-door criterion to the CEG framework. Their approach enables causal reasoning over maintenance strategies using path specific intervention logic, particularly suited for discrete event systems where temporal progression and conditional branching are central. Causality aware smart troubleshooting framework that leverages LLMs and Causal Bayesian Networks to extract root causes and solutions from textual maintenance records (RoX).

Their system integrates technical ontologies (FMMEA) with probabilistic graphical models, enabling causal inference at both observational and interventional levels. While conceptually aligned with our work, their approach is based on static, text driven causal graphs. In contrast, our PHM Co-pilot constructs localized causal models on the fly using structured sensor and operational data, supports sequential intervention simulation, and incorporates quantitative evaluation metrics such as ATE and Counterfactual Success Rate (CSR). Our system is designed for dynamic, real time decision making with direct integration of effect estimation modules (e.g., DR Learner), extending the capabilities of static textual diagnosis frameworks (Trilla, Yiboe, Mijatovic, & Vitrià, 2024).

### 2.4. The Epistemic Gap Between LLMs and Interventional Causal Inference

LLMs exhibit remarkable proficiency in generating maintenance plans and diagnostic suggestions from sensor inputs, their reasoning is fundamentally associative rather than causal. Trained on vast corpora of text, LLMs rely on semantic similarity and co-occurrence patterns to produce plausible outputs. However, this generative ability does not equate to an understanding of intervention effects or counterfactual reasoning. In the context of Prognostics and Health Management (PHM), where safety critical decisions depend on knowing whether an action will resolve an issue under specific conditions, LLMs fall short. As noted by (Zečević et al., 2023), LLMs may "talk causality" but lack the structural machinery to reason causally. They cannot simulate alternative outcomes (e.g., "what would have happened if a different action were taken") or account for confounding variables, both of which are central to robust treatment effect estimation. Therefore, while we employ LLMs to generate context aware and human-like maintenance suggestions, we pair them with a dedicated causal inference engine that estimates individualized treatment effects (e.g., ATE, ITE) over top k matched historical episodes.

This hybrid approach ensures that recommended actions are not just semantically plausible but also statistically grounded in causal evidence. To discuss further LLMs often appear to demonstrate causal reasoning, (Zečević et al., 2023) argue that this behaviour is largely illusory. In their paper Causal Parrots: Large Language Models May Talk Causality But Are Not Causal, the authors introduce the concept of meta- Structural Causal Models (meta-SCMs) models that encode correlations over causal facts found in natural language, rather than performing genuine causal inference. They propose the Correlation of Causal Facts (CCF) hypothesis, which posits that LLMs correctly answer causal questions only when those answers were already embedded in the training corpus as correlated causal statements, such as those found in Wikipedia. Their experiments show that LLMs like GPT-3 can handle simple causal chains and intuitive physical reasoning only up to a point, but performance drops on longer, randomized, or abstract chains indicating a lack of generalizable causal reasoning. Even with Chain-of-Thought (CoT) prompting, LLMs often rely on memorized reasoning templates rather than dynamic inference. When tasked with reconstructing known causal graphs, LLMs demonstrated only moderate accuracy, with results highly sensitive to prompt phrasing. Embedding-based retrieval using ConceptNet yielded marginal improvements but again highlighted that LLMs recall rather than reason. Thus, (Zečević et al., 2023), conclude that LLMs behave as "causal parrots" repeating causal patterns seen during training without modelling underlying mechanisms. This distinction is critical when designing systems that rely on trustworthy, generalizable causal reasoning. (Shrestha, Malberg, & Groh, 2025) investigate whether LLM possess genuine causal reasoning abilities or simply replicate memorized correlations, ultimately proposing prompt-based strategies to enhance their performance in interventional and

counterfactual tasks. The authors construct a new benchmark spanning 170 causal questions across three levels of reasoning: (1) direct cause-effect associations, (2) mediated causation, and (3) counterfactual interventions. They evaluate popular models including GPT-3.5, GPT-4, Claude, PaLM, and LLaMA under both zero-shot and Chain-of-Thought (CoT) prompting conditions.

Their findings reveal that while LLMs perform relatively well on direct causal queries, performance sharply declines on questions requiring mediated or counterfactual reasoning. Even with CoT prompting, improvements appear to stem from formatting regularity rather than genuine causal inference. The authors conclude that current LLMs still operate largely as "causal parrots", retrieving and reorganizing known facts without modelling causal mechanisms. (Zhang, Wang, Liu, & Agarwal, 2024) explore the kinds of reasoning errors LLMs make when faced with causal questions. Their controlled experiments show that models often fall prey to cognitive biases, such as inferring causation from temporal order (post hoc fallacy) or failing to recognize confounding variables. Even under structured prompting, LLMs struggle to self-correct these fallacies and instead rationalize incorrect inferences. The authors argue that such failure modes reveal the absence of a grounded causal model within the LLMs, cautioning against their use in any setting that demands rigorous counterfactual or interventional reasoning. To address these limitations, our framework augments LLM based case retrieval with explicit causal inference modules in both the recommender and evaluator components. While LLMs are capable of inferring high-level triggers and generating candidate plans, they lack principled causal reasoning capabilities and are prone to fallacies and memorization artifacts. We therefore integrate a lightweight, dynamically trained causal model to simulate do-interventions and estimate treatment effects, ensuring that the final recommendations are not only relevant but also causally grounded.

## 3. SYSTEM DESIGN

Our Recommender–Evaluator agent structure draws inspiration from the dual agent design proposed by (Lukens et al., 2024), where LLM are used to generate structured diagnostic plans and independently assess their ability to identify true failure modes. In their setup, a Recommender agent produces a sequence of troubleshooting steps, and an Evaluator agent verifies whether each step would reveal the actual fault. Our overall system architecture follows a similar dual-agent paradigm, as illustrated in Figure 1, we build upon this idea but extend it in two key ways:

1. Recommender generates causally sensitive actions by estimating treatment effects (e.g., ATE) based on structured inference input, and
2. Evaluator simulates counterfactual outcomes using causal inference models to assess the likely resolution impact of each action.

### 3.1. Trigger Inference Module

To enable structured causal reasoning in our PHM Copilot, we propose a causally aware inference pipeline that integrates sensor condition snapshots with the generative reasoning capabilities of LLMs. At the heart of this pipeline is the Trigger Inference Module, which transforms structured sensor inputs into interpretable maintenance narratives encompassing anomaly detection, failure diagnosis, and corrective action planning. While the CMAPSS dataset provides rich time-series degradation data across multiple engines and operating conditions, it lacks explicit annotations for real-world failure modes, diagnostic steps, corrective actions, and resolution outcomes. This presents a significant barrier to building explainable PHM systems grounded in causal logic, as most supervised learning approaches rely on labelled intervention data. To overcome this limitation, we develop a prompting framework that extracts relevant sensor snapshots from CMAPSS typically corresponding to early signs of degradation and feeds them into an LLM with tailored instruction templates. The prompts are designed to elicit structured maintenance traces that include:

1. The sensor triggers that prompt investigation
2. The inferred root cause or failure mode
3. The suggested diagnostic steps
4. The corresponding corrective actions
5. The expected resolution outcome

These generated episodes serve a dual purpose. First, they enrich the original CMAPSS data with structured, interpretable maintenance narratives, which are essential for downstream interventional simulation. Second, they allow us to synthetically generate large volumes of diagnostic and corrective workflows without relying on proprietary logs or costly human annotation. By grounding each generated maintenance trace in real sensor behaviour and inferring plausible failure and recovery sequences, the Trigger Inference Module produces structured representations of industrial scenarios that are both operationally realistic and semantically rich. This module forms the backbone of our LLM-guided PHM system, enabling subsequent components retrieval, graph construction, and do-intervention analysis to operate on consistent and interpretable maintenance narratives derived from raw sensor inputs.

To generate structured maintenance traces from CMAPSS sensor data, we adopt a hybrid prompting strategy that combines Chain-of-Table (CoT-table) prompting (Zhang et al., 2024) with the ReAct (Reasoning + Acting) framework (Yao et al., 2023). CoT-table prompting allows us to present multivariate sensor and operational inputs as compact tabular snap-

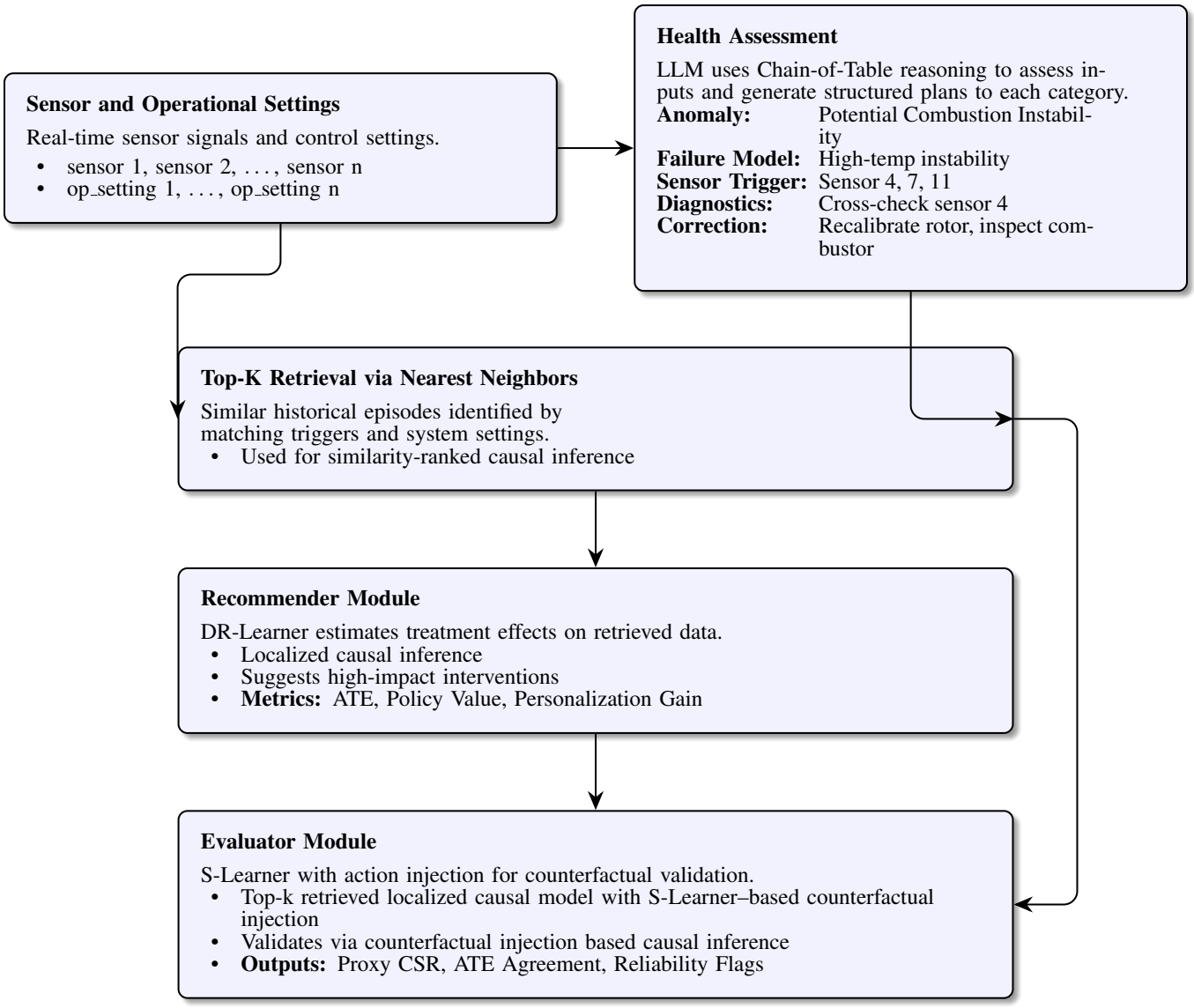


Figure 1. Causally informed PHM Co-Pilot architecture.

shots, enabling the language model to reason across correlated signals and detect emerging trends. By layering Re-Act on top, the model is guided through step by step reasoning identifying anomalies, inferring failure modes, selecting diagnostic steps, and recommending corrective actions in a transparent and interpretable manner. This combined approach mirrors how human technicians approach troubleshooting: observe, decide, and act. It also produces structured outputs that are easily transformable into elements of a causal knowledge graph. Additionally, the integration of Re-Act allows for mid step interventions, plan revisions, and fallback reasoning, making the framework suitable for scalable and interactive synthetic data generation. This approach enables the simulation of complete maintenance workflows in a manner that is both physically grounded and structurally interpretable. The resulting episodes preserve the physical realism of CMAPSS sensor dynamics while generating semantically rich maintenance traces that serve as inputs for downstream modules, including case-based retrieval, localized causal graph construction, and counterfactual intervention simulation. Unlike RAG-based frameworks that rely on retrieved textual knowledge, our approach constructs prompts entirely from CMAPSS sensor data. This ensures that all generated maintenance traces are grounded in real system behaviour, making the framework both interpretable and simulation ready.

### 3.2. Localized Causal Inference

Once the Trigger Inference Module generates a structured corrective actions, the next step in our pipeline is to evalu-

ate which actions are causally effective in resolving the detected failure. To do this, we adopt a localized causal inference strategy that estimates treatment effects over a narrow, context aware subset of historical cases, selected based on similarity in operating conditions and sensor trigger profiles. This ensures that estimated effects are tailored to the current system state, improving relevance and reducing confounding from unrelated operating regimes. Prior work has shown that global causal models often fail to account for contextual heterogeneity and may introduce noise from unrelated operating regimes, leading to biased or diluted effect estimates (Guo, Gifford, & Fraser, 2010).

Instead, localized causal inference based on nearest neighbour matching or contextual subsets has shown to improve treatment effect estimation by conditioning on comparable prior experiences (Zhou & Kosorok, 2017). Guided by this principle, we avoid constructing a global causal model over the entire dataset. Rather, we perform inference over the top k most similar maintenance episodes, retrieved based on operational settings and sensor trigger profiles. These top-k episodes represent historical instances with similar degradation behaviour and environmental context enabling our system to estimate context sensitive treatment effects grounded in relevant past experiences.

For each candidate corrective action proposed by the LLM, such as water wash or shop visits are not randomly assigned; they are strongly influenced by engine operating settings and sensor triggers. For example, engines with high compressor pressure ratio or rising exhaust gas temperature were much more likely to receive a water wash intervention. This creates systematic confounding between treated and control groups, because the same covariates that trigger action assignment also directly accelerate degradation and shorten remaining useful life. In other words, engines that receive treatment typically begin in worse health states than those in the control group. If we were to compare outcomes naively, the treated group would appear to perform worse, not because the intervention was harmful, but because they started from a more degraded baseline. The DR-Learner is designed to address exactly this scenario by combining outcome regression and propensity modelling, yielding doubly robust treatment effect estimates that remain consistent even when one of the nuisance models is mis specified (Künzel, Sekhon, Bickel, & Yu, 2019). This makes it highly effective for learning individualized treatment effects in environments where intervention assignment is biased by system state.

At the same time, PHM data exhibits nonlinear interactions among operating variables, sensor states, and interventions. For instance, the combined effect of high fan speed and moderate altitude can lead to degradation patterns that are not well captured by linear models. To handle this complexity, the S-Learner was implemented with a flexible nonparametric regressor, which can capture such nonlinearities while providing straightforward aggregate effect measures (ATE and ATE-based metrics) for evaluating fleet level performance. For example, causal forests have been developed as a random-forest-based algorithm designed to estimate heterogeneous treatment effects with valid asymptotic guarantees (Athey & Imbens, 2016),(Wager & Athey, 2018). While our framework instead employs meta learners specifically the S-learner in the evaluator module and the DR-learner in the recommender module, motivation is similar both approaches leverage flexible learning algorithms to overcome the limitations of classical parametric estimators in high-dimensional, nonlinear settings. Thus, the two learners together address the core challenges of PHM data confounding in action assignment and nonlinear dependencies in system behaviour, ensuring robust individualized recommendations and reliable population level evaluation.

Corrective action suggested by the LLM, we estimate its causal impact on the resolution outcome using a localized subset of top k historical cases matched by operational settings and sensor triggers. This subset serves as the inference time dataset for training a lightweight causal model typically a DR-Learner in the recommender module or an S-Learner in the evaluator module, thus avoiding reliance on a static, global causal model. Operating within the potential outcomes framework, we compute the Average Treatment Effect (ATE) and, when needed, the Individual Treatment Effect (ITE) for each action. This localized causal reasoning approach enables simulation of do-interventions over relevant prior episodes, thereby grounding each treatment effect estimate in system states similar to the current context. In some cases, we further enhance robustness by injecting synthetic counterfactuals into the matched subset, allowing simulation of alternative intervention trajectories. Although our method introduces a per inference training step, the small size of the top k window (typically 20–100 rows) ensures computational feasibility and allows the system to generate personalized, context aware causal estimates aligned with each engine's real time state.

### 3.3. Recommender Module: Intuitive Plan Generation and Ranking

The Recommender module generates candidate corrective actions based on the current system issue, defined by its operational settings and sensor trigger profile. It initiates inference by passing the input features to a Large Language Model (LLM), which returns a set of natural language maintenance actions that reflect plausible diagnostic or repair strategies for the inferred condition. To assess their likely effectiveness, the system retrieves a localized subset of historical maintenance episodes specifically, the top-k rows most similar to the current case. Similarity is computed based on both operational settings and LLM-identified sensor triggers, ensuring that re-

trieved examples reflect comparable degradation patterns and environmental conditions.

We standardized these features using scikit-learn's Standard Scaler, fitting the scaler on the candidate knowledge base and applying the same transformation to the inference input. This normalization ensures consistent scaling during similarity comparison. Scikit-learn's NearestNeighbors with Euclidean distance was then applied to identify the top-k. To identify the most effective corrective action among those proposed by the LLM, we apply a causal re-ranking step based on treatment effect estimation. For each candidate intervention, a localized DR-Learner model is dynamically trained on the top-k retrieved historical episodes those most similar to the current system state based on operational settings and sensor triggers. This model estimates the Average Treatment Effect (ATE) or Individual Treatment Effect (ITE) under the current context, representing the predicted resolution likelihood if that specific action were taken. This follows the principle of treatment effect based ranking, as demonstrated in prior work (Xu, Mahajan, Manrao, Sharma, & Kiciman, 2020), where individualized causal estimates guide the prioritization of interventions in observational settings.

We implement the DR-Learner using the EconML library (Battocchi et al., 2019), which combines outcome regression and propensity modelling to achieve doubly robust estimates (Kennedy, 2020). To estimate the conditional treatment effects for each inference instance, we adopt the DR-Learner (Doubly Robust Learner) framework, which combines outcome modelling and inverse propensity weighting to generate stable, bias-resistant treatment effect estimates. Specifically, for each row i, we model the potential outcomes under treatment and control using separate regressors:

$$\hat{\mu}_1(x_i) = \mathbb{E}(Y \mid X = x_i,\, T = 1) \tag{1}$$

$$\hat{\mu}_0(x_i) = \mathbb{E}(Y \mid X = x_i,\, T = 0) \tag{2}$$

We also estimate the propensity score:

$$\hat{e}(x_i) = \mathbb{P}(T = 1 \mid X = x_i) \tag{3}$$

Using these, the doubly robust estimate of the individual treatment effect $\hat{\tau}_i$ is given by:

- $\hat{\tau}_i$ Estimated treatment effect for instance $i$
- $x_i$ Covariates / features for instance $i$
- $T_i \in \{0, 1\}$ Treatment indicator (1=treated, 0=control)
- $Y_i$ Observed outcome for instance $i$
- $\hat{e}(x_i)$ Estimated propensity score: probability of receiving treatment given features $x_i$
- $\hat{\mu}_1(x_i)$ Predicted outcome under treatment
- $\hat{\mu}_0(x_i)$ Predicted outcome under control

$$\begin{aligned}\hat{\tau}_i = &\left(\hat{\mu}_1(x_i) + \frac{T_i}{\hat{e}(x_i)}\left(Y_i - \hat{\mu}_1(x_i)\right)\right) \\ &- \left(\hat{\mu}_0(x_i) + \frac{1 - T_i}{1 - \hat{e}(x_i)}\left(Y_i - \hat{\mu}_0(x_i)\right)\right)\end{aligned} \tag{4}$$

Our configuration uses linear regression for outcome and final-stage treatment effect models, and logistic regression for the propensity score model (with `max_iter=1000` to ensure convergence). Given the small size and structured nature of the top-$k$ matched subset, we disable cross fitting (`cv=1`), as data splitting could reduce stability and model fit in such localized windows. This setup provides an efficient, interpretable foundation for personalized causal ranking of candidate actions within a context-aware decision support pipeline.

### 3.4. Evaluator Module: Critical Counterfactual Scoring and Validation

The Evaluator module provides the counterfactual reasoning component of the PHM Co-Pilot pipeline, validating the causal impact of candidate actions proposed by the Recommender. While the Recommender generates plausible maintenance plans using LLM based reasoning and localized DR-Learner estimates, the Evaluator independently assesses whether these actions are likely to be effective in the current system context defined by operational settings and sensor trigger patterns observed at inference time. To perform this evaluation, the Evaluator retrieves the top k most similar historical episodes based on the current context and trains a localized S-Learner causal model. The S-Learner is implemented using the EconML library (Microsoft Research; Battocchi et al., 2019) and fits a single predictive model to jointly learn outcomes as a function of covariates and treatment indicators. Treatment effects are then computed by comparing predicted outcomes under different interventions while holding system features fixed (Künzel et al., 2019). In our setup, the S-Learner uses a random forest regressor (RandomForestRegressor from scikit-learn) as the underlying outcome model. This choice supports nonlinear interactions and accommodates heterogeneous treatment effects in the top k matched subset, providing a robust and flexible modelling foundation that complements the DR-Learner's doubly robust estimates.

To improve generalization and causal coverage, the Evaluator applies synthetic counterfactual injection, where LLM generated actions are embedded into the retrieved historical rows to simulate "what-if" scenarios even for interventions not previously observed. This approach is consistent with recent work in Augmented Causal Effect Estimation (ACEE) (Chen, Shen, & Pan, 2025) and enhances the system's abil-

ity to evaluate underrepresented or novel actions by expanding the diversity of treatment outcome pairs. Beyond scoring, the Evaluator also supports transparent decision making by generating personalized treatment effect estimates (ITE) and surfacing supporting evidence such as similar past cases or interpretable causal summaries. These outputs ITE rankings, resolution probabilities, and counterfactual comparisons provide users with a grounded and explainable basis for action selection. Prior studies have shown that such counterfactual justifications significantly improve user trust and interpretability in AI driven recommendation systems (Warren, Keane, & Byrne, 2022).

## 4. EVALUATION AND DISCUSSION

As a proof of concept, we employed the aforementioned prompting strategy combining Chain-of-Table and ReAct-style reasoning to generate corrective actions from CMAPSS sensor data. These prompts enabled the LLM to infer structured maintenance traces, including failure context and candidate repair steps, based on early-stage degradation patterns observed in the sensor inputs. To simulate interventional and counterfactual outcomes for evaluating the causal validity of these LLM-generated maintenance plans, we deployed the Evaluator module on a structured dataset of synthetic diagnostic episodes derived from the CMAPSS benchmark. All prompts were processed using OpenAI's GPT-4-turbo model (gpt-4-0125-preview), selected for its strong reasoning capabilities and inference efficiency. The Evaluator was run on a test set of 850 distinct inference rows, each representing a simulated degradation scenario. For each instance, the LLM first generated a candidate corrective action plan. Subsequently, the top k=20 most similar historical maintenance episodes were retrieved from a knowledge base of 10,500 generated episodes using similarity in operational settings and sensor trigger profiles. A localized causal model using an S-Learner was then trained on this context-aware subset to estimate treatment effects.

To perform counterfactual simulation, synthetic interventions were injected by substituting the LLM-recommended action into retrieved episodes where it had not originally occurred. This enabled the estimation of unobserved outcomes under alternative decisions. Both Individual Treatment Effects (ITE) and Average Treatment Effects (ATE) were computed for each candidate action, simulating do-interventions and evaluating their expected impact on resolution outcomes. Final rankings of recommended actions were derived based on these simulated causal effects, specifically their likelihood of resolving the failure under hypothetical interventions. This experimental design combines structured prompt-based generation with localized causal inference and counterfactual reasoning, showcasing the feasibility of an LLM-driven, causally-grounded PHM Co-pilot. We evaluated our Causal-Aware LLM PHM Co-pilot framework across 2,046 synthetic inference cases, each representing a distinct sensor-derived system state. For a focused evaluation of the Evaluator module, a filtered subset of 850 cases was selected based on completeness and diversity of the retrieved historical context. These 850 cases were used to simulate counterfactual outcomes, compute causal metrics (ATE, ITE), and re-rank candidate corrective actions.

### 4.1. Recommender Effectiveness

We evaluated the causal impact of the LLM-generated corrective actions, produced by the recommender module using the DR-Learner framework, through average treatment effect (ATE) estimation. The results, summarized in the table below, show that the recommended actions were generally effective in improving resolution likelihood across diverse system states. While most cases benefited from positive causal influence, some instances exhibited variability and risk, particularly in low-confidence settings, as reflected in the observed range of ATE values. Although the DR-Learner estimates CATE the treatment effect conditional on each system's features we summarize these into a global ATE metric to quantify the general effectiveness of the LLM-recommended actions across all inference rows. The mathematical formulation underlying this causal estimation is provided in the section below.

$$\text{Mean ATE} = \frac{1}{n}\sum_{i=1}^{n}\hat{\tau}_i \tag{5}$$

$$\text{Median ATE} = \text{median}\left(\hat{\tau}_1, \hat{\tau}_2, \ldots, \hat{\tau}_n\right) \tag{6}$$

$$\text{Negative ATE Rate} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left(\hat{\tau}_i < 0\right) \tag{7}$$

### 4.2. ATE Lift Over Control

When comparing the LLM recommended corrective actions against the historical actions taken in similar situations by evaluating their relative causal effectiveness. Specifically, we assessed the ATE lift the improvement in average treatment effect of the LLM generated recommendation compared to the average effect of past (control) actions in the top-k retrieved rows. This comparison allows us to quantify whether the recommender system offers meaningful improvements over prior maintenance decisions made in similar operational contexts. Using ATE lift as a metric is particularly important in high stakes decision making systems, where simply estimating the effect of an action is not enough. We need to benchmark the LLM's proposed actions against historical baselines to ensure that the system is not just generating plausible interventions, but is actually recommending better-than-before decisions. By evaluating the direction and frequency

of ATE lift, we can identify how often the LLM's recommendations truly outperform historical decisions and also flag instances where they may introduce risk thereby supporting downstream validation and risk-aware deployment.

Let $\hat{\tau}_i^{\text{LLM}}$ be the estimated treatment effect (ATE) of the LLM-recommended action for inference row $i$, and let $\hat{\tau}_i^{\text{control}}$ be the mean ATE of the historical (control) actions taken in the top-$k$ retrieved rows for the same inference input.

$$\text{ATE Lift}_i = \hat{\tau}_i^{\text{LLM}} - \hat{\tau}_i^{\text{control}} \tag{8}$$

$$\text{Mean ATE Lift} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\tau}_i^{\text{LLM}} - \hat{\tau}_i^{\text{control}}\right) \tag{9}$$

$$\text{Positive Lift Rate} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left(\hat{\tau}_i^{\text{LLM}} > \hat{\tau}_i^{\text{control}}\right) \tag{10}$$

- $\mathbb{1}(\cdot)$ is the indicator function: it equals 1 if the condition is true, and 0 otherwise.
- $n$ is the number of inference rows.

### 4.3. Policy Value Estimation

To assess the effectiveness of the LLM-generated actions in a real world deployment scenario, we estimate the policy value defined as the expected outcome (e.g., resolution likelihood) if a given policy were applied consistently across all inference cases. This metric allows us to simulate and compare the performance of different decision making strategies without needing to deploy them in practice. By comparing the estimated policy value of the LLM-based policy against that of the historical control actions, we can quantify the potential benefit of adopting the recommender system over legacy or frequency based heuristics. A higher policy value for the LLM recommended actions suggests that they are more likely to lead to successful outcomes when broadly applied, offering evidence for replacing or augmenting current maintenance practices with learned acausal policies Dudík, M., Erhan, D., Langford, J., and Li, L. (2014).

Let $\pi$ be the policy being evaluated (e.g., the LLM-generated policy), $Y_i$ be the observed outcome for instance $i$, $\hat{Y}_i(\pi)$ be the estimated outcome had policy $\pi$ been applied, and $n$ be the total number of inference cases.

Then, the estimated policy value is:

$$\text{Policy Value}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\hat{Y}_i(\pi) \tag{11}$$

If we're comparing the LLM policy $\pi^{\text{LLM}}$ against a control policy $\pi^{\text{control}}$, the policy value improvement is:

$$\begin{aligned}\text{Policy Value}\\ \text{Improvement} = \text{Policy Value}(\pi^{\text{LLM}})\\ - \text{Policy Value}(\pi^{\text{control}})\end{aligned} \tag{12}$$

### 4.4. ATE Agreement with Injected Estimate

To assess internal consistency and validate the stability of our causal effect estimation pipeline, we compared the ATE values generated directly by the recommender module with those obtained through a separate process involving synthetic action injection. This comparison helps ensure that the model's estimated treatment effects are not artifacts of a particular estimation strategy, but instead remain consistent across different causal reasoning procedures. High agreement between these two approaches strengthens confidence in the reliability of the estimated action impacts, reinforcing the validity of decisions based on the LLM-generated recommendations (Narita, Yasui, & Yata, 2021). We define agreement between the direct ATE and the injected ATE when their absolute difference is within a small threshold $\epsilon$ typically set to 0.05:

$$\text{Agreement}_i = \mathbb{1}\left(\left|\hat{\tau}_i^{\text{direct}} - \hat{\tau}_i^{\text{injected}}\right| \le \epsilon\right) \tag{13}$$

$\hat{\tau}_i^{\text{direct}}$ is the ATE estimated by the DR-Learner on the original (non-injected) recommendation. $\hat{\tau}_i^{\text{injected}}$ is the ATE computed after injecting the same recommended action synthetically into the historical data for unit $i$. $\epsilon$ is the agreement threshold (e.g., 0.05). $\mathbb{1}(\cdot)$ is the indicator function returning 1 if the condition is true, 0 otherwise.

$$\text{ATE Agreement Rate} = \frac{1}{n}\sum_{i=1}^{n}\text{Agreement}_i \tag{14}$$

### 4.5. Personalization Gain

The recommender's general causal effect estimates with individualized treatment effects tailored to each inference case. This comparison helps evaluate how well the global policy aligns with row specific needs. A high alignment indicates that the recommender's default behaviour performs reasonably well across diverse scenarios. However, discrepancies between the global and individualized effects can reveal opportunities for fine grained personalization enabling the system to adapt recommendations more precisely based on the unique characteristics of each system state. By quantifying this alignment, we gain insight into how much additional value could be unlocked through personalization beyond the average-case policy. This is especially important in high-stakes domains like predictive maintenance, where a generic recommendation might not optimally address specific degra-

dation profiles. Personalization gain thus serves as a guiding metric to identify when and where more tailored interventions are warranted. Tu, Y., Basu, K., DiCiccio, C., Bansal, R., Nandy, P., Jaikumar, P., and Chatterjee, S. (2020).

Let $\hat{\tau}_i$ be the Individual Treatment Effect (ITE) for the $i$-th inference case, estimated by the DR-Learner. Let $\text{ATE}_{\text{global}}$ be the global Average Treatment Effect estimated by the recommender across all inference cases.

Then, for each row $i$, the Personalization Gain is computed as:

$$\text{PersonalizationGain}_i = \hat{\tau}_i - \text{ATE}_{\text{global}}$$

### 4.6. Negative ATE Flag Rate

In a decision-support system that recommends corrective actions, it is critical to identify instances where the proposed actions could inadvertently lead to worse outcomes than the status quo. Negative ATE (Average Treatment Effect) cases represent such failure points situations where the LLM-recommended intervention is estimated to reduce the probability of resolution compared to historically observed control actions. Flagging these cases is essential because it helps distinguish risky or potentially harmful recommendations from beneficial ones. This metric serves as a safeguard, highlighting the need for an evaluator module that can act as a filter rejecting or adjusting interventions that may backfire. By surfacing these adverse effect scenarios, the system ensures that causal recommendations undergo an additional layer of scrutiny before being deployed, enhancing both safety and trust in high-stakes operational settings like industrial maintenance or healthcare diagnostics.

Let $\hat{\tau}_i$ be the estimated individual treatment effect (ITE or CATE) for inference case $i$, i.e.,

$$\hat{\tau}_i = \hat{Y}_i(1) - \hat{Y}_i(0) \tag{15}$$

where:

- $\hat{Y}_i(1)$: predicted outcome if the LLM-recommended action is applied
- $\hat{Y}_i(0)$: predicted outcome under the control (historical) action

Define an indicator function: Let the negative flag for instance $i$ be defined as:

$$\text{Negative Flag}_i = \mathbb{1}\left(\hat{\tau}_i < 0\right) \tag{16}$$

Then, the Negative ATE Flag Rate is:

$$\begin{aligned} \text{Negative ATE Flag Rate} &= \frac{1}{n}\sum_{i=1}^{n} \text{Negative Flag}_i \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(\hat{\tau}_i < 0\right) \end{aligned} \tag{17}$$

### 4.7. Support-Based ATE Reliability

Many LLM-recommended actions are based on limited historical matches, making their ATE estimates potentially unreliable. Even if the estimated effect appears strong, low data support raises concerns about statistical validity. This motivates the need for an Evaluator module, which helps filter or flag low-support actions. By assessing the reliability of ATE estimates, the evaluator ensures that only well supported and trustworthy actions are considered for decision-making.

Let $S_i$ be the number of matched historical samples (support) for inference row $i$, and let $\hat{\tau}_i$ be the estimated ATE. We define an indicator for high-support reliability as:

$$\text{Reliable Flag}_i = \mathbb{1}\left(S_i \geq s_{\min}\right) \tag{18}$$

where:

- $S_i$ is the number of matched historical samples for inference case $i$
- $s_{\min}$ is the minimum support threshold for reliability

$\mathbb{1}(\cdot)$ is the indicator function that returns 1 if the condition is true, and 0 otherwise. Then, the Support-Based Reliability Rate across all $n$ inference rows is:

$$\text{Reliability Rate} = \frac{1}{n}\sum_{i=1}^{n} \text{Reliable Flag}_i \tag{19}$$

### 4.8. Evaluator Effectiveness (Proxy Analysis)

To gauge the evaluator's contribution without relying on expensive counterfactual simulations for every inference case, we introduced a proxy-based approach that estimates how well the evaluator aligns with the recommender's individualized causal estimates. By comparing the evaluator's assessment with personalized treatment effects generated by the recommender, we can approximate how consistently the evaluator filters or validates recommended actions. This proxy serves as a lightweight indicator of counterfactual agreement how likely it is that a recommended action truly leads to resolution when compared against plausible alternatives. Most cases fall within a moderate confidence zone, indicating that the evaluator plays a stabilizing role, offering a second opinion in uncertain decision regions. Moreover, a high rate of agreement between the recommender and evaluator modules suggests that the overall framework maintains causal coherence across components. This builds confidence in the system's internal alignment and helps identify edge cases where evaluator intervention can prevent incorrect or risky recommendations from being deployed.

Let $\hat{\tau}_i$ be the individual treatment effect (ITE) for inference case $i$, estimated by the DR-Learner in the recommender. Let

$n$ be the total number of inference cases. Define the proxy CSR (Counterfactual Success Rate) for case $i$ as:

$$\text{CSR}_i = \mathbb{1}\left(\hat{\tau}_i > 0\right) \tag{20}$$

That is, we consider a case successful if the estimated ITE is positive, implying that the recommended action improves the resolution likelihood over the control. Then, the overall mean proxy CSR is:

$$\text{Mean Proxy CSR} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(\hat{\tau}_i > 0) \tag{21}$$

This gives the proportion of inference cases where the recommended action is expected to have a positive causal effect, serving as a lightweight proxy for full counterfactual simulation success. To improve decision-making under offline conditions, we rely on proxy evaluation strategies that estimate relative policy improvement rather than absolute value. The $\Delta$-OPE framework introduced by Jeunen and Ustimenko (2024) enables low-variance estimation of policy value differences by leveraging offline data from a logging policy, thereby functioning as an effective proxy for real-world performance evaluation in recommender systems Jeunen, O., & Ustimenko, A. (2024).As shown in Table 1, each evaluation metric is accompanied by observed values and a descriptive comment explaining its relevance. This ensures that the usability of all metrics both for recommender effectiveness and evaluator alignment is clearly conveyed.

## 5. LIMITATIONS

### 5.1. Lack of Ground Truth for Corrective Actions

The CMAPSS dataset does not include labelled ground-truth corrective actions for each degradation event. Consequently, the causal effectiveness of the LLM generated and evaluator-ranked interventions cannot be validated against real-world outcomes. While our evaluation leverages synthetic counterfactual injection and causal inference metrics (ATE, ITE), these remain approximations and cannot fully replace expert-annotated resolution logs or operator feedback.

### 5.2. Synthetic Data and Simulated Reasoning

Our inference and evaluation pipelines operate on structured synthetic episodes derived from CMAPSS sensor traces. While useful for proof-of-concept, these do not capture the full operational variability or uncertainty seen in real industrial workflows. This may affect generalizability and suggests caution when extrapolating our results to field-deployed PHM systems.

### 5.3. Generalizability and Domain Transfer

The framework was tested on a single benchmark (CMAPSS) and has not yet been validated across different machinery types, industries, or sensor ecosystems. The quality of LLM-generated plans and the robustness of causal estimators (like DR-Learner) may vary significantly depending on domain-specific failure modes and data sparsity.

### 5.4. Simplified Causal Modelling Assumptions

Our use of S-Learner and DR-Learner methods relies on assumptions of conditional ignorability and minimal confounding within the top-k retrieved cases. However, in real-world settings with latent confounders or time-varying effects, more advanced methods such as marginal structural models (MSMs) or instrumental variable techniques may be required for robust inference.

### 5.5. Lack of Multi-Step Causal Modelling

While our framework evaluates individual actions, it does not model the joint or sequential causal effect of multi-step maintenance plans. Specifically, we do not estimate the cumulative impact of executing sequences like Inspect $\rightarrow$ Calibrate $\rightarrow$ Replace, nor do we model chained do-calculus effects (Dawson & Lavori, 2007; Fan, 2022). Each DR-Learner/S-Learner is trained per action, ignoring interdependencies between sequential steps.

### 5.6. Causal Limits of LLMs

While our framework integrates LLMs for generating candidate actions, the actual causal reasoning is delegated to structured inference modules due to the LLMs' lack of explicit causal modelling. As emphasized in prior work (Zečević et al., 2023),(Pearl, 2009), LLMs cannot simulate counterfactuals or perform do-calculus. This reinforces the necessity of hybrid approaches that ground generative fluency with causal rigor (Trilla et al., 2024),(Warren et al., 2022)

## 6. FUTURE WORK

### 6.1. Integrating Evaluator Feedback for Causal Re-Ranking

Our current architecture maintains an open-loop separation between the Recommender and Evaluator modules, where the Evaluator passively computes causal metrics such as localized ATE, Proxy Counterfactual Success Rate (CSR), and support-based reliability, without influencing the Recommender's final action choice. As a future extension, we propose enabling a re-ranking mechanism where the Evaluator's causal assessments can be used to prioritize or filter candidate actions. By computing a composite causal quality score based on metrics like estimated treatment effect, policy value, and reliability flags the system could dynamically adjust the rank-

Table 1. Summary of Causal Evaluation Metrics for LLM-Generated Actions

| Metric | Observed Values | Comments |
|---|---|---|
| Recommender Effectiveness (Mean ATE) | Mean = 0.695, Median = 0.833, Max = +1.0, Min = -1.0 | LLM-generated actions were generally effective in improving outcomes, though variability in causal impact underscores the need for caution in low-confidence cases. |
| ATE Lift Over Control | Mean = -0.011, Median = 0.017, Positive Lift = 52.64% | LLM-generated actions outperformed historical ones in the majority of cases, but variability in causal lift highlights the need for validation to prevent suboptimal decisions. |
| Policy Value Estimation | LLM Policy Value = 0.689, Control = -0.011, Absolute Gain = +0.700 | LLM-generated actions outperformed historical controls in expected resolution outcomes, demonstrating their potential as effective, policy-level interventions when guided by causal evaluation. |
| ATE Agreement with Injection Estimate | Agreement Rate (within ±0.05) = 93% | LLM-generated actions demonstrated strong internal consistency, with causal estimates remaining stable across different evaluation methods, supporting the reliability of the recommender's outputs. |
| Personalization Gain | Mean Gain = -0.011, Median = 0.017, Positive Gain = 52.64% | LLM-generated actions show moderate alignment with personalized causal needs, but their effectiveness could be improved by tailoring recommendations more precisely to individual system states. |
| Negative ATE Rate | 14.31% of inference rows had negative ATE | LLM-generated actions occasionally reduced resolution likelihood, highlighting the risk of blindly accepting such outputs and underscoring the need for causal filtering to avoid harmful decisions. |
| Support-Based ATE Reliability | Low-Support ATE = 0.696, High-Support ATE = 0.654, High-Support Rows = 0.21% | Most LLM-generated actions were supported by few historical matches, making their ATE estimates less statistically reliable and highlighting the need for causal validation. |
| Evaluator Effectiveness (Proxy CSR) | Mean CSR = 0.702, Std Dev = 0.247, High Confidence ($> 0.8$) = 2.27%, Low Confidence ($< 0.5$) = 0.91%, Agreement = 93% | High agreement between recommendation and causal validation modules suggests that while LLM actions are often plausible, their reliability benefits from causal oversight. |

ing of LLM-generated actions prior to final selection. Such a feedback loop would strengthen the causal grounding of the overall recommendation pipeline, improving robustness in high-stakes or low-confidence environments. This direction aligns with broader trends in causal recommender systems, where treatment effect estimation is increasingly integrated into ranking decisions, though our approach is distinct in leveraging localized causal inference over retrieved top-k episodes with action injection. Incorporating such causal reranking as a downstream filtering or reordering layer remains a promising area for further development.

### 6.2. Toward Production-Scale Causal Co-Pilots

To move beyond the current proof-of-concept, future work will focus on enhancing both the realism and scalability of the framework. In production settings, the computational overhead of causal estimation can be mitigated through techniques such as batching, caching, and retrieval-aware model reuse. To improve the quality of LLM-generated plans, future iterations may incorporate fine-tuning on real-world maintenance logs and operator annotations. Expanding the system's causal reasoning capabilities will involve modelling multi-step action chains using Structural Causal Models (SCMs) or Marginal Structural Models (MSMs) to capture temporal dependencies and treatment interactions. Additionally, replacing localized retrieval with FAISS style global embedding search could enable more scalable and diverse access to historical knowledge. Finally, integrating symbolic causal modules or structured reasoning scaffolds into the prompting strategy may allow future LLMs to generate plans with stronger causal coherence, bridging the gap between generative fluency and interventional validity.

## 7. CONCLUSION

This paper presents a hybrid causally aware PHM Co-pilot framework that integrates large language models with structured causal inference to generate and evaluate maintenance recommendations. By combining generative prompting strategies with localized treatment effect estimation, the system is capable of producing interpretable and context-sensitive action plans grounded in real-time sensor data. Through extensive simulation over the CMAPSS benchmark, we demonstrate that LLM generated corrective actions when evaluated using causal models such as DR-Learner and S-Learner can be ranked and validated effectively using synthetic counterfactual injections. This dual-module design en-

sures both adaptability and accountability; while the Recommender suggests human-like interventions, the Evaluator verifies their potential effectiveness under realistic "what-if" scenarios.

The framework shows strong alignment between individual and average treatment effects, and exhibits promising behaviour in terms of personalization and causal reliability. While certain limitations such as lack of multi-step causal modelling and ground-truth labels remain, this work establishes a robust proof of concept for the use of causality driven LLM agents in industrial prognostics and health management. Future extensions may explore multi-step treatment trajectories, symbolic causal model fusion, and real world deployment in safety critical environments. Together, these advancements can help transform LLM-based co-pilots from generative tools into trustworthy, decision-critical partners in industrial maintenance.

## NOMENCLATURE

| | |
|---|---|
| LLM | Large Language Model |
| PHM | Prognostics and Health Management |
| ATE | Average Treatment Effect |
| ITE | Individual Treatment Effect |
| CATE | Conditional Average Treatment Effect |
| DR-Learner | Doubly Robust Learner for causal effect estimation |
| S-Learner | Single-model learner estimating treatment effects in a unified model |
| SCM | Structural Causal Model |
| MSM | Marginal Structural Model |
| CSR | Counterfactual Success Rate |
| CCF | Correlation of Causal Facts (hypothesis about LLM causal behavior) |
| CMAPSS | Commercial Modular Aero-Propulsion System Simulation dataset |
| Do-Intervention | Causal intervention operator from do-calculus |

## REFERENCES


Athey, S., & Imbens, G. W. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, *113*(27), 7353–7360. doi: 10.1073/pnas.1510489113

Chen, L., Shen, X., & Pan, W. (2025). Enhancing causal effect estimation with diffusion-generated data. *arXiv preprint arXiv:2504.03630*. (https://arxiv.org/abs/2504.03630)

Diehl, M., & Ramirez-Amaro, K. (2021). A causal-based approach to explain, predict and prevent failures in robotic tasks. *IEEE Robotics and Automation Letters*, *6*(4), 7157–7164. doi: 10.1109/LRA.2021.3095800

Guo, S., Gifford, A., & Fraser, M. W. (2010). Matching methods for causal inference: A review and a look forward. *Statistical Science*, *25*(1), 1–20. doi: 10.1214/10-STS342

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, *116*(10), 4156–4165. doi: 10.1073/pnas.1804597116

Lukens, S., McCabe, L. H., Gen, J., & Ali, A. (2024). Large language model agents as prognostics and health management copilots. In *Annual conference of the phm society* (Vol. 16). Retrieved from \url{https://doi.org/10.36001/phmconf.2024.v16i1.3906} doi: 10.36001/phmconf.2024.v16i1.3906

Narita, Y., Yasui, S., & Yata, K. (2021, September). Debiased off-policy evaluation for recommendation systems. In *Proceedings of the 15th acm conference on recommender systems (recsys '21)*. Amsterdam, Netherlands. (September 27–October 1) doi: 10.1145/3460231.3474231

Pearl, J. (2009). *Causality: Models, reasoning and inference* (2nd ed.). Cambridge University Press.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine prognostics. In *Proceedings of the international conference on prognostics and health management (phm08)* (pp. 1–9). IEEE. (https://www.phmsociety.org/events/conference/phm/08/datachallenge/)

Shrestha, R. B., Malberg, H., & Groh, G. (2025). Do large language models reason causally? evaluating interventional, counterfactual, and contrastive capabilities. *arXiv preprint arXiv:2403.06234*. (https://arxiv.org/abs/2403.06234)

Trilla, A., Yiboe, O., Mijatovic, N., & Vitrià, J. (2024). Industrial-grade smart troubleshooting through causal technical language processing: A proof of concept. In *Kdd 2024 workshop on causal inference and machine learning in practice*. (arXiv:2407.20700, https://arxiv.org/abs/2407.20700)

Vanderschueren, T., Boute, R., Verdonck, T., Baesens, B., & Verbeke, W. (2022). Prescriptive maintenance with causal machine learning. *European Journal of Operational Research*, *299*(1), 252–267. doi: 10.1016/j.ejor.2021.06.040

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242. doi: 10.1080/01621459

.2017.1319839

Warren, G., Keane, M. T., & Byrne, R. M. J. (2022). Features of explainability: How users understand counterfactual and causal explanations in ai decisions. *arXiv preprint arXiv:2204.10152*. (https://arxiv.org/abs/2204.10152)

Xu, Y., Mahajan, D., Manrao, L., Sharma, A., & Kiciman, E. (2020). Split treatment analysis to rank heterogeneous causal effects for prospective interventions. In *Proceedings of the 14th acm international conference on web search and data mining (wsdm '21)*. doi: 10.1145/3437963.3441821

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). React: Synergizing reasoning and acting in language models. In *Proceedings of the 37th conference on neural information processing systems (neurips)*. (https://arxiv.org/abs/2210.03629)

Yu, X., & Smith, J. Q. (2017). Causal chain event graphs for remedial maintenance. *IEEE Transactions on Reliability*, *66*(1), 62–75. doi: 10.1109/TR.2016.2598561

Zečević, M., Willig, M., Dhami, D. S., & Kersting, K. (2023). Causal parrots: Large language models may talk causality but are not causal. *Transactions on Machine Learning Research*. (https://openreview.net/forum?id=tv46tCzs83)

Zhang, Q., Wang, Y., Liu, S., & Agarwal, A. (2024). Do large language models understand causality? a study on reasoning fallacies. *arXiv preprint arXiv:2403.12345*. (https://arxiv.org/abs/2403.12345)

Zhou, X., & Kosorok, M. R. (2017). Causal nearest neighbor rules for optimal treatment regimes. *arXiv preprint arXiv:1711.08451*. (https://arxiv.org/abs/1711.08451)

## BIOGRAPHIES

**Rajarajan Kirubanandan** is an NLP Engineer and independent researcher with over eight years of experience in developing and deploying machine learning solutions across the legal and healthcare domains. Raj holds a B.E. in Electronics and Communication Engineering, as well as an M.E. in Applied Electronics from Anna University, India. He is passionate about applying recent AI and LLM developments to assist in complex processes in healthcare and legal sectors. Rajarajan has led applied AI teams focusing on natural language processing, predictive modelling, and causally informed decision systems.