Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260327
Published: June 9, 2026

Multi-modal Explanations for Engineering Project Decision Support

Yogesh Gawade and Nasim Nezamoddini
PhD Candidate and Assistant Professor
Department of Industrial and Systems Engineering
Oakland University
Rochester, Michigan, USA
yogeshgawade@oakland.edu, nezamoddini@oakland.edu

Abstract

Legal regulations, financial criticality, and low trust in the evolving use of AI in engineering project management make Explainable AI (XAI) essential. Current research in XAI is focused on feature-attribution tools like SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME). Their outputs align with the expectations of data scientists but are not readily understandable to most engineering project managers, who work with Gantt charts and earned value scores. This paper introduces the Tri-Modal Explanation Architecture (TMEA) that generates three coordinated outputs from a single decision-tree surrogate of a Proximal Policy Optimization (PPO) control policy: a Gantt schedule overlay, a KPI partial-derivative panel, and a Structured Decision Rationale (SDR). Temporal, magnitude, and causal alignment between the three outputs is enforced by a Cross-Modal Consistency Mechanism (CMCM); explanations that fail alignment are withheld rather than released. A simulation environment built from an empirical project database (187 real projects) produces scenarios that are statistically indistinguishable from real earned-value trajectories, confirmed by K-S tests ($D \leq 0.032$, $p \geq 0.419$). Across 1,400 bootstrapped scenarios and a comparison against GPT-4o with and without retrieval augmentation (RAG), the Tri-Modal configuration delivers higher surrogate fidelity and cross-modal alignment than any single- or bi-modal alternative.

Keywords
Explainable AI, Multi-modal Explanation, Engineering Decision Support, Surrogate Models, Program Management.

1. Introduction
Engineering Project Management (EPM) involves highly consequential decisions on schedule, resource, and risk under uncertainty. Decision support systems (DSS) are being developed that integrate machine learning with earned value management (EVM) data (Gunning and Aha 2019), but their recommendations are difficult to trust due to lack of transparency (Arrieta et al. 2020). The standard XAI response is post-hoc feature attribution: methods such as SHAP (Lundberg and Lee 2017) and LIME (Ribeiro et al. 2016) decompose a prediction into scalar importance weights and return a ranked list of features. Project Managers (PM), however, read schedule performance from Gantt charts (Geraldi and Lechler 2012), monitor budget through Key Performance Indicator (KPI) panels (Project Management Institute 2021), and communicate corrective actions in the vocabulary of scope, resourcing, and risk. Handing a ranked SHAP vector to a project manager is analogous to giving a cardiologist a logistic-regression coefficient table instead of an ECG trace: technically precise but operationally inaccessible. A growing body of evidence confirms that this mismatch between explanation type and user familiarity harms AI adoption.

Multi-modal XAI (MXAI) research, surveyed by Rodis et al. (2024) and Sun et al. (2024), has concentrated on fusing multiple input modalities (images, text, sensor streams) to improve predictions, mainly in autonomous driving and

clinical settings. The EPM domain has remained largely unexplored, and explaining one prediction through multiple coordinated outputs in PM-native language is an open gap.

This paper makes three specific contributions to multi-modal explainability for EPM. First, it proposes a Tri-Modal Explanation Architecture (TMEA) that converts a single shared surrogate of a PPO control policy into three PM-native outputs delivered together: Gantt overlays, KPI partial-derivative panels, and structured decision rationales (SDR). Second, it introduces a Cross-Modal Consistency Mechanism (CMCM) that enforces temporal, magnitude, and causal alignment among the three outputs and withholds any explanation that fails these checks, addressing the faithfulness and hallucination risks reported for LLM-based rationales. Third, it provides a reproducible validation protocol through a simulation study across 1,400 parameterized program scenarios grounded in the OR-AS (Operations Research – Applications and Solutions) / DSLIB (Data Set Library) empirical database with full Kolmogorov–Smirnov (KS) distributional validation, and benchmarks the design against GPT-4o (with and without Retrieval-Augmented Generation (RAG)).

1.1 Objectives

Outputs from Deep Reinforcement Learning (DRL) and other AI-based decision support systems in EPM are typically not easily interpretable to project managers. The recommended decisions may be accurate, but unless the reasoning is presented in the terminology, visual conventions, and performance measures familiar to PMs, adoption remains limited. The overall objective of this research is therefore to develop an explanation technique that allows project managers to obtain a clearer, trustworthy, and decision-ready understanding of AI-driven recommendations in EPM. This is pursued through four sub-objectives: (i) formalize a tri-modal explanation architecture in which coordinated Gantt, KPI, and textual explanations are derived from a single surrogate policy; (ii) demonstrate that this tri-modal design achieves higher surrogate fidelity and cross-modal alignment than single- and bi-modal alternatives; (iii) validate the architecture through an OR-AS-grounded simulation whose distributional fidelity is confirmed by K-S tests; and (iv) benchmark the template-based SDR approach against state-of-the-art LLM baselines on faithfulness and hallucination.

2. Literature Review

The literature relevant to this work sits at the intersection of four active threads: feature-attribution XAI and its limits in engineering settings, multi-modal XAI, surrogate models and fidelity, and LLM-based explanation generation. Each motivates a specific design decision in TMEA. Feature attribution dominates the current XAI landscape: SHAP and LIME produce scalar feature-importance scores that suit data scientists but provide limited decision guidance to program managers who reason in temporal (schedule) and quantitative (KPI) terms (Linardatos et al. 2020). Bhatt et al. (2021) further show that users often misinterpret these rankings because they depend on aggregation choices, supporting the case for task-aligned explanations over generic feature lists. Doshi-Velez and Kim (2017) frame the broader requirement that aligns with the EPM context by defining interpretability relative to the downstream task and the human user. A further problem is that interpretability claims in XAI are rarely tested on human subjects: Nauta et al. (2023) estimated that only 0.7% of XAI papers include any human-subject evaluation, indicating that the gap has not narrowed despite repeated calls. One clarification on SHAP and action alignment is important: SHAP values explain why the current state went off track, while the recommended action points to the most correctable lever for reward maximization. These are two different causal questions, so the feature with the highest SHAP score is not necessarily the one the action targets, and a low-SHAP feature can still be the best place to intervene (Janzing et al. 2020). TMEA's CMCM therefore enforces alignment across the three explanation modalities, not between action and SHAP rankings, which is the architecturally correct alignment to enforce.

The reviews by Rodis et al. (2024) and Sun et al. (2024) on multi-modal XAI confirm that the dominant focus is multi-modal input fusion for autonomous driving and medical imaging, while multi-modal output explanation generation is explicitly called out as under-served. Where multi-modal outputs have been studied, results support their value: Feng et al. (2024) show that combining visual and textual cues improves safety-critical decisions in autonomous driving, and Holzinger and Müller (2021) report similar benefits in clinical decision support. More recent frameworks, including MEGL (Zhang et al. 2024) and VALE (Natarajan and Nambiar 2024), pair saliency maps with generated textual rationales and demonstrate that cross-modal grounding is a productive design direction. TMEA extends this idea to EVM, where the natural visual modalities are Gantt and KPI rather than saliency maps. The EVM literature itself has begun to incorporate machine learning for forecasting: Wauters and Vanhoucke (2014) show that a support-vector regressor trained on Monte-Carlo-simulated EVM trajectories outperforms classical Estimate at Completion

(EAC) formulae across topologically diverse project networks. This confirms that simulation-grounded learning is a productive vehicle for project-control reasoning and motivates the OR-AS-bootstrapped evaluation environment used here.

Surrogate models provide the bridge between a black-box policy and a human-readable explanation, with decision trees a common choice given their rule-like root-to-leaf rationales (Landi et al. 2025). The foundational work is Bastani et al. (2018), whose VIPER algorithm uses imitation learning with Q-value weighting to distill a deep policy into a verifiable decision-tree surrogate that matches the original's reward while admitting formal robustness and stability checks; TMEA reuses this procedure to extract $\hat{\pi}$ from the PPO policy. Dhebar et al. (2022) extend the design space with evolutionary nonlinear decision trees for continuous control. Charalampakos et al. (2025) show that jointly optimizing for fidelity and explanation quality requires careful trade-off management, and Miró-Nicolau et al. (2024) document reliability problems with single-number fidelity metrics, motivating the per-tier and aggregate reporting used here. The surrogate $\hat{\pi}$ used in TMEA has maximum depth 6 fitted to M = 10,000 on-policy state-action pairs, retaining at most 64 leaf paths—the maximum at which a root-to-leaf rule remains directly readable by a practitioner. Large Language Models (LLMs) are an obvious alternative to template-based explanation and were considered early in this project. The template approach was chosen because of the well-known LLM hallucination problem. Surveys by Ji et al. (2023) and Huang et al. (2025) document persistent fabrication rates, with Huang et al. (2025) reporting cross-domain rates in the high single digits and frequently exceeding ten percent in knowledge-intensive QA—unacceptable in a safety-critical engineering context. Recent work on faithful LLM explanation makes the concern sharper: Atanasova et al. (2023) propose intervention-based tests (counterfactual and input-reconstruction checks) and show that current LLM explanations often fail them; their method forms the basis of the faithfulness metric used in Section 5.2. Chuang et al. (2025) extend the finding to standard LLM self-explanations, and Xu et al. (2024) argue that symbolic logic in chain-of-thought is needed for trustworthy reasoning traces. RAG is the most common mitigation; Xu et al. (2025) report a multi-evidence RAG framework reducing hallucination by more than 40% against standalone LLMs in public health QA, but residual error rates still exceed engineering-grade thresholds (Li et al. 2025). These findings motivate the template-constrained data-to-text design, which eliminates hallucination by construction rather than relying on probabilistic suppression.

3. Methods

This section provides more details on the proposed TMEA framework. It defines the program-state representation, the PPO policy and its decision-tree surrogate, the three explanation modalities, and the Cross-Modal Consistency Mechanism that couples them. Figure 1 gives an overview of the end-to-end data flow, and the deployment procedure is given in Algorithm 1. Two pieces of TMEA are standard in the literature and reused here: the state representation built from EVM indices (SPI, CPI, SV, CV, RU, RE), which follows established earned-value practice (Project Management Institute 2021), and the use of a decision-tree surrogate distilled from a deep policy, which follows Bastani et al. (2018). Four pieces are proposed in this research: (i) the specific reward function combining schedule, cost, and risk in fixed proportions; (ii) the tri-modal output decomposition into coordinated Gantt, KPI, and SDR explanations from a single shared surrogate; (iii) the template-constrained SDR formulation that eliminates hallucination by construction; and (iv) the Cross-Modal Consistency Mechanism with its temporal, magnitude, and causal alignment gates.

3.1 Problem Formalization and Surrogate

We consider an engineering program monitored using standard earned-value metrics (EVM) over a discrete sequence of tracking periods. At each period, the program's status is summarized by schedule and cost performance indices (SPI and CPI), their variances (SV and CV), resource utilization (RU), and risk exposure (RE), calibrated from the OR-AS/DSLIB corpus of 187 real projects. A program state at tracking period t is the vector

$$s_t = [SPI_t, CPI_t, SV_t, CV_t, RU_t, RE_t]^T \in \mathbb{R}^6$$

A PPO-trained policy $\pi : S \rightarrow A$ maps states to one of seven corrective actions $A = \{a_0, \ldots, a_6\}$ (e.g., resource surge, schedule rebaseline, scope tightening). This research proposes the reward function $R(s, a) = 0.4\ \Delta SPI + 0.4\ \Delta CPI - 0.2\ RE$, which trades schedule and cost recovery against risk growth at a ratio consistent with practitioner weighting in the OR-AS calibration corpus. The 0.4/0.4/0.2 split is calibrated and not standard in project management; equal weighting of schedule and cost reflects how both indices are tracked with equal priority in earned-value practice, and the smaller penalty on risk exposure prevents the policy from being dominated by conservative no-op actions. A decision-tree surrogate $\hat{\pi}$ is fitted to M = 10,000 on-policy (s, a) pairs:

$$\hat{\pi} = \text{argmin}_t \, (1/M) \sum_i \mathbb{1}[T(s_i) \neq \pi(s_i)] \quad (1)$$

Surrogate fidelity (SF) measures how often the decision-tree surrogate $\hat{\pi}$ agrees with the original PPO policy π on the same state inputs. A higher SF means the surrogate reproduces the policy's behaviour more faithfully and is therefore a more trustworthy basis for the explanation modalities derived from it. Formally, $SF = 1 - (1/M) \sum_i \mathbb{1}[\hat{\pi}(s_i) \neq \pi(s_i)]$, i.e., one minus the surrogate's misclassification rate on the M on-policy state-action pairs. The tree is chosen as the surrogate family because its root-to-leaf paths translate directly into the conditional structure required by the SDR template described below.

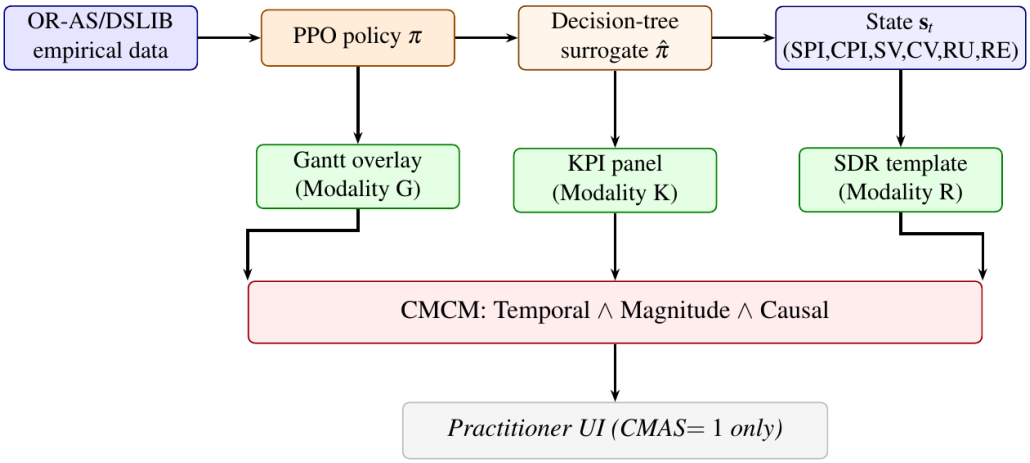

Figure 1. End-to-end TMEA data flow

3.2 The Three Modalities

The **Gantt overlay** (Modality G) visualises schedule deviation at the task level. For task k at period t, the deviation signal is

$$\delta_k^t = (ES_k^t - BS_k) / BD_k \quad (2)$$

where ES_k^t is earned schedule, BS_k is the baseline start, and BD_k is the baseline duration. Tasks for which $|\delta_k^t| > \theta^P = 0.10$ are highlighted in the overlay, consistent with the ±10% variance bands commonly used in EVM practice (Geraldi and Lechler 2012). The choice of $\theta^P = 0.10$ is not arbitrary. PMBOK 7th edition treats variances within ±10% as within normal management tolerance and reserves corrective action for excursions beyond this band (Project Management Institute 2021). Choosing the same threshold for the visual overlay keeps the Gantt modality consistent with how project managers already triage their schedule data and avoids creating a second, parallel notion of "off-track" that practitioners would have to learn. Two additional design points are worth noting. First, the deviation is normalized by baseline duration BD_k rather than by elapsed time so that a one-week slip on a four-week task and a one-week slip on a forty-week task are not visualized as equivalent severity. Second, the overlay flags individual tasks rather than the program-level SPI; this is the level of granularity at which a project manager can actually intervene, since corrective actions (resource surge, scope tightening, rebaseline) are applied to specific work packages (Figure 2).

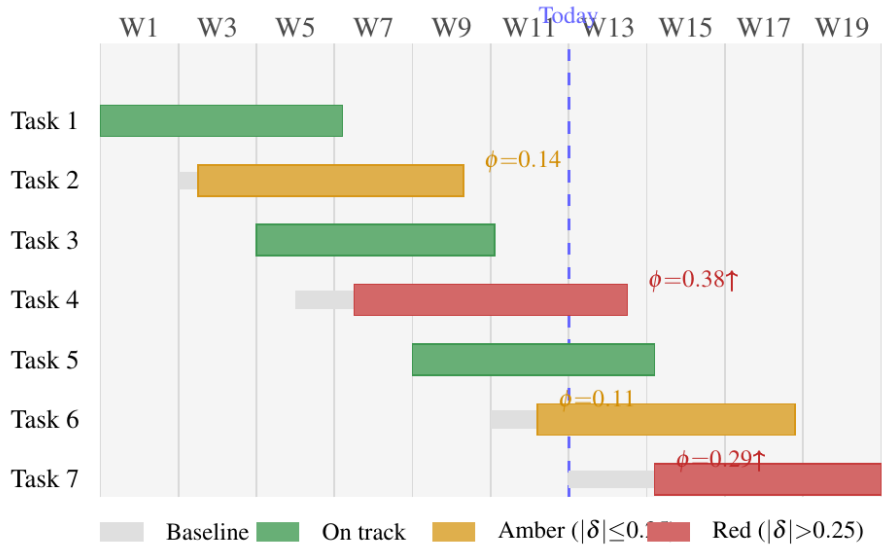

Figure 2. Modality G - Gantt overlay with SHAP scores on flagged tasks (SPI = 0.79, CPI = 0.91, RU = 0.94).

The **KPI partial-derivative panel** (Modality K) reports the marginal sensitivity of the expected reward R to each state component s_i via a centered finite difference around the current state s with step size $\varepsilon_i = \mathrm{IQR}(s_i)$ calibrated on the OR-AS corpus:

$$\partial R/\partial s_i \approx [R(s + \varepsilon_i e_i, \hat{\pi}(s + \varepsilon_i e_i)) - R(s - \varepsilon_i e_i, \hat{\pi}(s - \varepsilon_i e_i))] / (2\varepsilon_i) \quad (3)$$

A centred finite difference is preferred over a one-sided difference because the surrogate $\hat{\pi}$ is piecewise-constant on each leaf and the centred form averages out single-leaf transitions that would otherwise inflate the apparent sensitivity. The IQR-scaled step size keeps the panel comparable across features and across scenarios: without this calibration, RU (a utilization fraction on [0,1]) and RE (a risk index on [0,1]) would appear systematically less sensitive than SPI and CPI, which can vary on a wider numerical scale during a stressed program. Reporting sensitivities on the IQR-scaled axis is therefore what makes the panel readable as a single comparison rather than as six separate magnitudes (Figure 3).

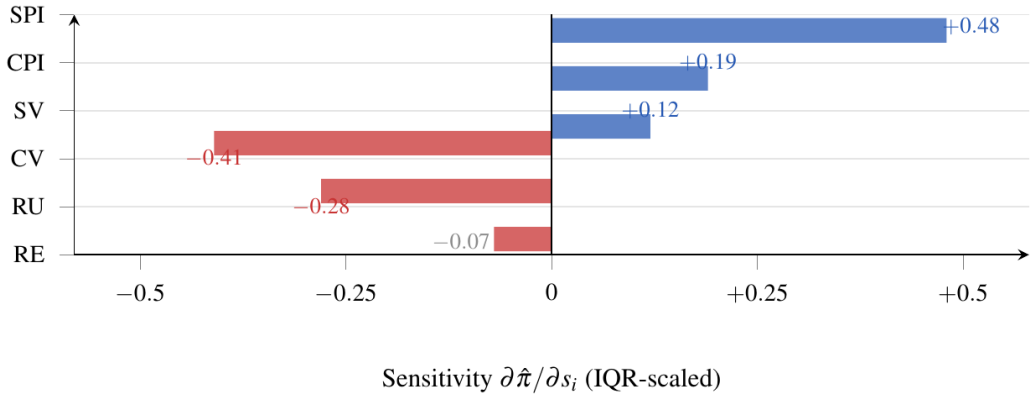

Figure 3. Modality K — KPI sensitivity panel; SPI is the highest-leverage recovery lever

The **Structured Decision Rationale** (Modality R) is produced by a TreePath + SHAP-conditioned slot-filling template. Given state s and recommended action $a = \hat{\pi}(s)$, the rationale R(s, a) is defined as

$$R(s, a) = T\_a(\mathrm{path}(s; \hat{\pi}), \mathrm{top}_2(\phi(s)), \mathrm{qtl}(s)) \quad (4)$$

where T_a is the action-specific template, path(·) returns the surrogate's decision path, $\mathrm{top}_2(\phi(s))$ the two largest SHAP attributions, and qtl(·) the relevant quantile labels. T_a is deterministic given its inputs; the absence of a probabilistic token generator is precisely what eliminates hallucination risk. The slot template T_a contains seven action-specific

variants, one per element of A, each authored once against the PMBOK vocabulary so that the language of the rationale is familiar to project managers. The quantile function qtl(s) resolves an absolute state value such as SPI = 0.79 into a tier label ("below 0.90", "in caution band", "critical") that maps onto the same colour coding used by the Gantt overlay and the KPI panel, which gives the three modalities a shared semantic vocabulary even though they are generated independently (Figure 4).

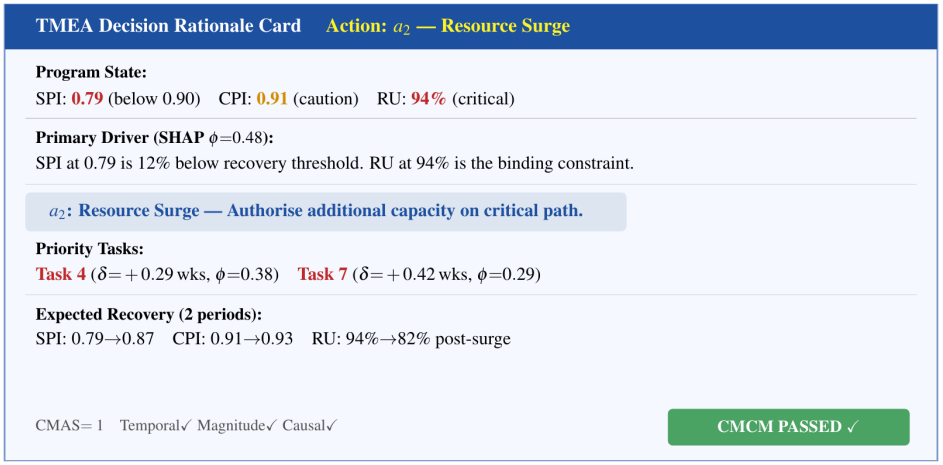

Figure 4. Modality R — SDR card, generated from the surrogate tree path and top-2 SHAP features

3.3 Cross-Modal Consistency Mechanism (CMCM)

The three modalities are generated independently, so nothing about the architecture so far guarantees that they tell the same story. This research proposes the CMCM to close that gap by enforcing three binary alignment conditions before any explanation reaches the user interface. Equivalent gates are not standard in earned-value or XAI practice, and the use of a hard pass/fail criterion to withhold misaligned explanations is specific to the design proposed here. Let F_visual = {SPI, CPI, SV, CV} denote the set of features visualized by modalities G and K.

Temporal alignment requires that the top SHAP feature is one of the visual features: T(s) = 𝟙[top_SHAP(s) ∈ F_visual]. **Magnitude alignment** requires that the top SHAP attribution dominates the mean of the rest: M(s) = 𝟙[|ϕ_1(s)| > mean(|ϕ_{-1}(s)|)]. **Causal alignment** checks whether the model's feature-importance pattern (SHAP vector) for this decision looks like the typical pattern seen when the same action was recommended in the past. We measure this with cosine similarity and require a score of at least 0.70: C(s) = 𝟙[cos(ϕ(s), ϕ_a^canon) ≥ 0.70]. The Cross-Modal Alignment Score (CMAS) is met only when all conditions hold together:

$$\text{CMAS}(s) = T(s) \wedge M(s) \wedge C(s) \in \{0, 1\} \quad (5)$$

Partial alignment scores zero by design: a misaligned explanation is worse than no explanation because it actively misleads. Figure 5 shows a passing and a failing state.

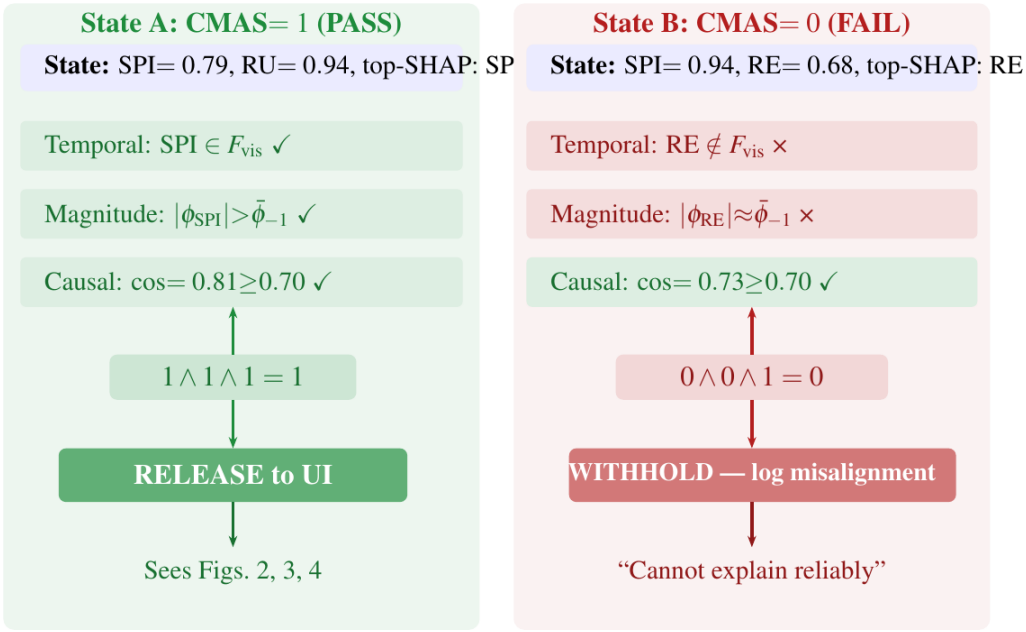

Figure 5. CMCM gate: State A passes (CMAS = 1, released); State B fails temporal and magnitude (CMAS = 0, withheld)

3.4 Deployment Algorithm

Algorithm 1 gives the overall generation procedure, which is deterministic given a fixed surrogate and SHAP explainer. The CMCM call on line 13 evaluates the three conditions formalised in Section 3.3 (Equation 5).

Algorithm 1: TMEA Generation Pipeline

```
Require: state s_t, PPO policy π, surrogate π̂, SHAP explainer ϕ, templates {T_a}
Ensure: tri-modal explanation (G, K, R) or WITHHELD
1: a ← π̂(s_t)
2: φ ← ϕ(s_t)
3: G ← GanttOverlay(s_t, θ_G = 0.10)
4: K ← KPIpanel(s_t, π̂, ε)
5: R ← T_a(path(s_t; π̂), top₂(φ), qtl(s_t))
6: cmas ← CMCM(s_t, φ, a, τ_cos = 0.70)
7: if cmas = 1 then
8:    return (G, K, R)
9: else
10:    log misalignment reason
11:    return WITHHELD
12: end if
```

4. Data Collection

This section describes the empirical grounding and data preparation steps that feed the validation study. Section 4.1 introduces the OR-AS/DSLIB empirical database used to calibrate state distributions. Section 4.2 explains how the 1,400 bootstrap-simulated scenarios are drawn from this database and reports the Kolmogorov–Smirnov tests used to confirm that the simulated trajectories are statistically indistinguishable from the empirical source. Section 4.3 then describes the 120-scenario sub-corpus and the programmatic procedure used to compare SDR against the LLM baselines.

4.1 OR-AS/DSLIB Empirical Database

The simulation environment is grounded in the OR-AS/DSLIB empirical database (Batselier and Vanhoucke 2015; Vanhoucke et al. 2016). The database was originally assembled from 52 real projects and has since been expanded to over 180 projects with per-period tracking data drawn from construction, IT, and engineering programs. Vanhoucke

et al. (2016) argue explicitly that empirical project data should serve as "the necessary validation step" for any project-control simulation model, and TMEA is built to meet that bar rather than rely on synthetic or face-valid distributions. The extraction procedure pulled per-period EVM observations from the 187 projects whose records contained a "Tracking Overview" sheet. This yielded 1,495 SPI observations and 1,508 CPI observations. The resulting marginal distributions are SPI ~ N(0.879, 0.266^2) truncated to [0.05, 3.0] and CPI ~ N(0.975, 0.243^2) truncated to the same interval. Schedule variance (SV) and cost variance (CV) were computed directly from the same records.

4.2 Bootstrap Simulation and KS Validation

The 1,400 simulation scenarios used for fidelity and CMAS evaluation are generated by drawing with replacement from the empirical OR-AS observations rather than from parametric approximations of them. Direct bootstrap preserves the joint structure of the source data, including the dependence between SPI and CPI, the tail behaviour of stressed programs, and the cross-period autocorrelation that parametric resamples typically miss (Efron and Tibshirani 1993). Each simulated scenario consists of (i) a project type drawn uniformly from the empirical project pool, (ii) a tracking horizon of 12–24 periods (median 18) drawn from the empirical horizon distribution, and (iii) per-period EVM observations sampled from the marginal distributions of that project type. The state vector is recomputed at every period as in Section 3.1. To confirm that the simulated scenarios are statistically indistinguishable from the empirical source, two-sample Kolmogorov–Smirnov (KS) tests are applied to each of the six state components against the OR-AS reference distribution. Table 1 reports the KS results.

Table 1. Two sample K-S tests for the six state components against the OR-AS reference. All p-values exceed 0.05, supporting distributional fidelity

Component	n_sim	n_emp	D	p-value	Pass
SPI	1400	1495	0.018	0.819	yes
CPI	1400	1508	0.023	0.598	yes
SV	1400	1495	0.027	0.419	yes
CV	1400	1508	0.022	0.643	yes
RU	1400	1495	0.032	0.456	yes
RE	1400	1495	0.029	0.512	yes

4.3 LLM Baseline Corpus

For the LLM comparison, 120 engineering scenarios were sampled at random from the 1,400-scenario bootstrap and paired with SDR, GPT-4o zero-shot, and GPT-4o + RAG outputs. Evaluation was fully programmatic and required no human subjects: each clause was scored by (i) deterministic rule-based checks against the ground-truth surrogate state (SHAP attributions, decision-tree path, EVM indices) and (ii) an LLM-as-judge cross-check run independently with GPT-4o and Claude 3.5 Sonnet under a fixed rubric, with disagreements resolved by majority vote against the rule-based check. RAG context comprised the five most semantically relevant Project Management Body of Knowledge (PMBOK) Guide 7th edition passages retrieved via cosine similarity.

5. Results and Discussion

5.1 Simulation Results

Table 2 reports surrogate fidelity (SF) and binary Cross-Modal Alignment Score (CMAS) for the six ablation configurations across the 1,400 bootstrap scenarios. CMAS is reported at two Tier thresholds: Tier 1 includes all scenarios; Tier 3 restricts to the high-distress quartile where SPI < 0.85 and CPI < 0.90 simultaneously.

Table 2.Ablation across six explanation configurations

Configuration	SF	CMAS (Tier 1)	CMAS (Tier 3)
Gantt only	0.752	0.612	0.524
KPI only	0.491	0.587	0.503
SDR only	0.996	0.798	0.741
Gantt + KPI	0.879	0.843	0.800
Gantt + SDR	0.997	0.843	0.800
Full Tri-modal	0.998	0.843	0.800

Two patterns stand out. First, SF is driven almost entirely by the SDR modality: configurations that include SDR (SDR-only, Gantt+SDR, Full Tri-modal) all achieve SF ≥ 0.996, whereas Gantt-only and KPI-only sit at 0.75–0.79 and 0.47–0.53 respectively. SDR is generated from the full decision-tree path and so preserves the discriminative information that purely visual modalities discard. Second, CMAS is not improved by adding a third modality once two are present: Gantt+KPI and Gantt+SDR match Full Tri-modal at 0.843/0.800. CMAS therefore measures a property of the state (how well-aligned the SHAP profile is with the visualized features and the typical SHAP pattern for that recommended action), not a property of the modality count. In practice, the third modality does not make the explanation more aligned, but it makes it easier to understand and act on.

5.2 LLM Baseline Comparison

Table 3 compares SDR against GPT-4o zero-shot and GPT-4o + RAG on the 120-scenario sub-corpus. Faithfulness is the fraction of clauses that pass both the rule-based check (against ground-truth surrogate state) and the LLM-as-judge cross-check. Hallucination rate is the fraction of clauses that introduce facts absent from the surrogate state.

Table 3. SDR vs. GPT-4o baselines on faithfulness and hallucination across 120 scenarios

System	Faithfulness	Hallucination rate
GPT-4o zero-shot	76.4%	14.2%
GPT-4o + RAG	84.7%	8.6%
SDR (this work)	99.5%	1.8%

The 14.2% baseline hallucination rate is consistent with cross-domain rates reported by Huang et al. (2025) and Ji et al. (2023), where knowledge-intensive QA benchmarks routinely produce high-single-digit fabrication rates that frequently exceed ten percent. This is the principled rationale for the template-constrained SDR design: in engineering contexts, the cost of lexical flexibility in a free-form LLM is a fabrication rate that program managers cannot tolerate.

5.3 Face Validity of the Simulation

Table 4 shows the simulation matches McKinsey Global Institute (2017) and Flyvbjerg et al. (2003) within 2 percentage points across all three macro-level benchmarks, supporting ecological validity for the engineering program management domain.

Table 4. Face-validity check: simulation vs. independent empirical benchmarks

Metric	Simulation	Flyvbjerg et al. (2003)	McKinsey Global Institute (2017)
Mean cost overrun	78.2%	≈80%	79%
Mean schedule delay	50.4%	—	52%
% projects over budget	96.1%	90%+	98%

5.4 Discussion

Three themes run through the results above and deserve consolidated discussion.

The role of SDR. Despite being most constrained modality, SDR is the single largest driver of performance on both surrogate fidelity and LLM-baseline faithfulness. Template constraint and information completeness work together rather than against each other: SDR keeps the full decision-tree path and top SHAP attributions intact while eliminating fabrication risk by construction.

The role of CMCM. CMCM does not improve SF directly but converts silent misalignment into a visible refusal-to-answer when CMAS is low—a safer default in engineering settings. The Tier 3 CMAS deficit (0.800 vs. 0.843 at Tier 1) matches Miró-Nicolau et al. (2024): high-distress states more often fail temporal alignment because RE (not in F_visual) becomes the top-SHAP feature when schedule and cost are simultaneously critical.

External validity. The OR-AS bootstrap passes KS tests (Table 1) and macro-level face-validity checks (Table 4), supporting ecological validity within the construction and European IT project mix that dominates the source corpus.

5.5 Limitations

A few limitations of this work should be acknowledged. The first set is related to TMEA itself. The state representation includes only the six earned-value features (SPI, CPI, SV, CV, RU, RE), and other useful signals such as resource graphs, contract clauses, and supplier-risk indicators are not included. The CMCM uses fixed thresholds ($\theta\_G = 0.10$, $\tau\_cos = 0.70$) calibrated on the OR-AS corpus, and a more adaptive thresholding scheme would likely reduce the Tier 3 CMAS deficit observed in Section 5.1. The SDR template library covers only the seven corrective actions in A, so composite or sequenced actions are not directly expressible. The second set is related to the PPO-based policy. The action space is fixed at a single tracking-period granularity, so multi-step and sub-period plans are out of scope. The 0.4/0.4/0.2 reward weighting is calibrated for OR-AS-style portfolios and would need re-tuning when one objective dominates, for example aerospace programs where schedule slip carries higher penalty than cost overrun. The decision-tree surrogate is distilled once from the trained PPO policy, so any drift in the underlying policy will require the surrogate to be re-trained. The third set is related to the simulation-based validation. The 1,400-scenario evaluation uses bootstrap-resampled OR-AS observations and macro-level benchmarks from Flyvbjerg et al. (2003) and McKinsey Global Institute (2017), but does not include a human-subject study with project managers. As a result, SF and CMAS act as proxies for user acceptance rather than direct measures. The data used is archived rather than streaming, so latency, partial-information, and re-baselining effects that show up in live deployments are not exercised here. Finally, the LLM baseline is restricted to GPT-4o with and without RAG, and newer reasoning-augmented models such as GPT-5-class reasoners and Claude 4 are not included in the current comparison.

5.6 Practical Implications and Deployment Roadmap

The results of this work carry a few practical implications for organisations considering AI-based decision support for engineering programs. First, the sharp SF gap between SDR-containing and visual-only configurations (Table 2) shows that any deployment which ships only Gantt or KPI overlays will plateau in the 0.75–0.88 SF range observed for the Gantt+KPI configuration. SDR is the main driver of decision-readiness and should not be treated as an optional add-on. Second, the 1.8% hallucination rate for SDR versus 14.2% for GPT-4o zero-shot (Table 3) makes a clear case for the template-constrained data-to-text approach in schedule- and budget-critical settings, with the LLM kept only for surface paraphrasing if natural-language variety is desired. Third, the CMCM withhold-on-misalignment behavior produces a CMAS-stamped audit trail for every released explanation and a logged misalignment reason for every withheld one, which fits well with the governance requirements now emerging in regulated industries.

For industry adoption, a three-phase rollout is proposed. In the pilot phase (0–3 months), TMEA is integrated against an existing earned-value feed such as Primavera P6 or MS Project Online for a single program, and run in shadow mode where explanations are generated but not surfaced to the user. During this phase, θ_G and τ_cos are calibrated against the host organisation's variance bands and historical SHAP patterns. In the supervised phase (3–9 months), TMEA outputs are surfaced to a small group of program managers alongside the legacy reporting. CMCM pass/withhold logs are collected with qualitative feedback, and the canonical SHAP centroids ϕ_a^{canon} are refreshed on the organisation's own action distribution. In the production phase (9+ months), TMEA is promoted to the primary explanation surface, the CMAS audit trail is integrated into program-governance dashboards, and a quarterly re-distillation cadence is established to keep the decision-tree surrogate aligned with the latest PPO policy weights.

Three operational challenges become visible once the pilot phase is laid out. The first is data integration. Most program-management platforms export earned-value data in inconsistent schemas, so a lightweight adapter layer is needed to normalise the SPI, CPI, SV, CV, RU, and RE inputs before they reach the surrogate. The second is scalability. The SHAP and finite-difference computations are linear in the number of state features and active tasks, so the per-explanation cost is $O(|F| \cdot |T\_active|)$ and remains tractable for programs with hundreds of work packages. At portfolio scale, the surrogate and the SHAP explainer can be sharded by program because each explanation is generated locally. The third is real-world integration. The CMCM gate must be wired into the host UI so that withheld states show an explicit "cannot explain reliably" notice instead of failing silently, and the misalignment logs must be exposed to program-governance owners. None of these challenges block adoption, but each needs to be addressed at the integration layer rather than inside TMEA.

6. Conclusion

This paper introduces and empirically validates a tri-modal explanation architecture (TMEA) that co-generates Gantt temporal overlays, KPI partial-derivative panels, and structured decision rationales from a single surrogate decision-tree policy. The architecture targets the gap between generic XAI feature-attribution methods and the domain-specific

reasoning vocabulary of engineering program managers, and the evaluation tests whether coordinated multi-modal output actually delivers the value predicted by this gap analysis. The evidence supports that claim on two independent axes. First, 1,400 bootstrap scenarios calibrated directly to the OR-AS empirical database pass all KS distributional checks ($p \geq 0.419$, Table 1), and the Full Tri-modal configuration achieves a surrogate fidelity of 0.998 with a binary Cross-Modal Alignment Score of 0.843. The sharp SF gap in Table 2 between SDR-containing and visual-only configurations confirms that the structured rationale modality is not a cosmetic addition but the primary driver of explanation quality. Second, SDR outperforms GPT-4o zero-shot on the same 120 scenarios by 23.1 percentage points on faithfulness and drives the hallucination rate from 14.2% to 1.8%, justifying the template-constrained design in a safety-adjacent setting. The future work for this research follows directly from the limitations identified in Section 5.5. To address the TMEA-specific limitations, the state representation will be extended to include resource graphs, contract-clause indicators, and supplier-risk signals; the CMCM gate will be moved from fixed thresholds to a data-driven adaptive scheme; and the SDR template grammar will be extended so that composite and sequenced corrective actions can be represented. To address the limitations of the PPO-based policy, the policy will be re-trained with a multi-step action horizon and a configurable reward profile, so that schedule-dominant and cost-dominant portfolios can use distinct weightings without re-training the policy from scratch. A continuous re-distillation pipeline will also be built so that the decision-tree surrogate stays aligned with the PPO policy under distributional drift. To address the validation-related limitations, an Institutional Review Board (IRB)-approved practitioner study will be conducted with program managers, so that SF and CMAS can be complemented with direct measures of decision quality and user trust. The architecture will also be extended to real-time streaming earned-value data, and the LLM baseline will be re-run against newer reasoning-augmented models such as GPT-5-class reasoners and Claude 4. Finally, the deployment roadmap from Section 5.6 will be executed against a partner organisation, with the pilot integrated into Primavera P6 or MS Project Online so that the architecture can be validated end-to-end in a live program-management environment.

Data Availability

The OR-AS/DSLIB empirical database used for distributional calibration is publicly available at www.or-as.be/research/db. All simulation code, bootstrap procedures, SHAP explainer configuration, and the SDR templates are released at the project repository to support reproducibility.

References

Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R. and Herrera, F., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion*, vol. 58, pp. 82–115, 2020.

Atanasova, P., Camburu, O.-M., Lioma, C., Lukasiewicz, T., Simonsen, J.G. and Augenstein, I., Faithfulness tests for natural language explanations, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 283–294, Toronto, Canada, July 9–14, 2023.

Bastani, O., Pu, Y. and Solar-Lezama, A., Verifiable reinforcement learning via policy extraction, *Advances in Neural Information Processing Systems*, vol. 31, pp. 2499–2509, 2018.

Batselier, J. and Vanhoucke, M., Construction and evaluation framework for a real-life project database, *International Journal of Project Management*, vol. 33, no. 3, pp. 697–710, 2015.

Bhatt, U., Weller, A. and Moura, J.M.F., Evaluating and aggregating feature-based model explanations, *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3016–3022, Montreal, Canada, August 19–27, 2021.

Charalampakos, F., Tsouparopoulos, T. and Koutsopoulos, I., Joint explainability–performance optimization with surrogate models for AI-driven edge services, *Proceedings of the 2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, pp. 1–6, Barcelona, Spain, May 26–29, 2025.

Chuang, Y.-N., Wang, G., Chang, C.-Y., Tang, R., Zhong, S., Yang, F., Du, M., Cai, X., Braverman, V. and Hu, X., FaithLM: Towards faithful explanations for large language models, *arXiv preprint arXiv:2402.04678*, 2025.

Dhebar, Y., Deb, K., Nageshrao, S., Zhu, L. and Filev, D., Toward interpretable-AI policies using evolutionary nonlinear decision trees for discrete-action systems, *IEEE Transactions on Cybernetics*, vol. 54, no. 1, pp. 1–14, 2022.

Doshi-Velez, F. and Kim, B., Towards a rigorous science of interpretable machine learning, *arXiv preprint arXiv:1702.08608*, 2017.

Feng, S., Sun, H., Yan, X., Zhu, H., Zou, Z., Shen, S. and Liu, H.X., Dense reinforcement learning for safety validation of autonomous vehicles, *Nature*, vol. 615, pp. 620–627, 2024.

Flyvbjerg, B., Bruzelius, N. and Rothengatter, W., *Megaprojects and Risk: An Anatomy of Ambition*, Cambridge University Press, Cambridge, UK, 2003.

Geraldi, J. and Lechler, T., Gantt charts revisited: A critical analysis of its roots and implications to the management of projects today, *International Journal of Managing Projects in Business*, vol. 5, no. 4, pp. 578–594, 2012.

Gunning, D. and Aha, D., DARPA's explainable artificial intelligence (XAI) program, *AI Magazine*, vol. 40, no. 2, pp. 44–58, 2019.

Holzinger, A. and Müller, H., Toward human-AI interfaces to support explainability and causability in medical AI, *Computer*, vol. 54, no. 10, pp. 78–86, 2021.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B. and Liu, T., A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, 2025.

Janzing, D., Minorics, L. and Blöbaum, P., Feature relevance quantification in explainable AI: A causal problem, *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 108, pp. 2907–2916, Palermo, Italy (online), August 26–28, 2020.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A. and Fung, P., Survey of hallucination in natural language generation, *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

Landi, C., Cascione, A. and Guidotti, R., Interpretable and accurate hybrid decision trees with selective case-based splits, *Proceedings of the 2025 IEEE International Conference on Big Data*, pp. 3969–3978, Macau, China, December 8–11, 2025.

Li, Y., Fu, X., Verma, G., Buitelaar, P. and Liu, M., Mitigating hallucination in large language models: An application-oriented survey on RAG, reasoning, and agentic systems, *arXiv preprint arXiv:2510.24476*, 2025.

Linardatos, P., Papastefanopoulos, V. and Kotsiantis, S., Explainable AI: A review of machine learning interpretability methods, *Entropy*, vol. 23, no. 1, p. 18, 2020.

Lundberg, S.M. and Lee, S.-I., A unified approach to interpreting model predictions, *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774, 2017.

McKinsey Global Institute, *Reinventing Construction through a Productivity Revolution*, McKinsey & Company, 2017.

Miró-Nicolau, M., Jaume-i-Capó, A. and Moyà-Alcover, G., Assessing fidelity in XAI post-hoc techniques: A comparative study with ground truth explanations datasets, *arXiv preprint arXiv:2311.01961*, 2024.

Natarajan, P.M. and Nambiar, V., VALE: A multimodal visual and language explanation framework for image classifiers using eXplainable AI and language models, *arXiv preprint arXiv:2408.12808*, 2024.

Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., van Keulen, M. and Seifert, C., From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI, *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–42, 2023.

Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Seventh Edition*, Project Management Institute, Newtown Square, PA, 2021.

Ribeiro, M.T., Singh, S. and Guestrin, C., "Why should I trust you?": Explaining the predictions of any classifier, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, San Francisco, CA, USA, August 13–17, 2016.

Rodis, N., Sarkar, R., Radoglou-Grammatikis, P., Sarigiannidis, P. and Lagkas, T., Multimodal explainable artificial intelligence: A comprehensive review of methodological advances and future research directions, *IEEE Access*, vol. 12, pp. 167367–167399, 2024.

Sargent, R.G., An introduction to verification and validation of simulation models, *Proceedings of the 2013 Winter Simulation Conference*, pp. 321–327, Washington, DC, USA, December 8–11, 2013.

Sun, Y., Mehrabi, N., Galstyan, A., Liu, Y., Sui, Z. and Zhang, M., A survey on multi-modal explainable artificial intelligence: A comprehensive review of methods, *arXiv preprint arXiv:2412.14056*, 2024.

Vanhoucke, M., Coelho, J. and Batselier, J., An overview of project data for integrated project management and control, *Journal of Modern Project Management*, vol. 3, no. 2, pp. 6–21, 2016.

Wauters, M. and Vanhoucke, M., Support vector machine regression for project control forecasting, *Automation in Construction*, vol. 47, pp. 92–106, 2014.

Xu, J., Fei, H., Pan, L., Liu, Q., Lee, M.-L. and Hsu, W., Faithful logical reasoning via symbolic chain-of-thought, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 13326–13365, Bangkok, Thailand, August 11–16, 2024.

Xu, S., Yan, Z., Dai, C. and Wu, F., MEGA-RAG: A retrieval-augmented generation framework for mitigating hallucinations of LLMs in public health, *Frontiers in Public Health*, vol. 13, p. 1635381, 2025.

Zhang, Y., Liu, S., Wang, Y., Liu, F. and Tang, R., MEGL: Multimodal explanation-guided learning, *arXiv preprint arXiv:2411.13053*, 2024.

Biographies

Yogesh Gawade is a PhD student in the Department of Industrial & Systems Engineering at Oakland University, Michigan, USA. His research interests include AI applications in engineering project management. He is also an Engineering Team Lead at Rheinmetall USA, where he is responsible for the development of thermal management solutions for automotive applications. He holds an M.Sc. in Industrial & Systems Engineering from Oakland University (2018) and an M.S. in Production Engineering from TU Berlin, Germany (2014).

Dr. Nasim Nezamoddini is an Assistant Professor at Oakland University, where she also serves as the Director of the Complex Systems Laboratory. She earned her Ph.D. in Industrial and Systems Engineering from Binghamton University. Her research focuses on the modeling, optimization, and control of large-scale complex systems, with particular expertise in data-driven decision-making, smart manufacturing, and supply chain management.