Disentangling Prognostic Observability from Policy Stochasticity in PHM-Aware Reinforcement Learning for Semiconductor Fab Dispatch

Sean Mondesire[1], Tori Wright[1], and Bulent Soykan[1]

[1] *School of Modeling, Simulation, and Training, University of Central Florida, Orlando, FL, USA*
sean.mondesire@ucf.edu
tori.wright@ucf.edu
bulent.soykan@ucf.edu

ABSTRACT

High-mix low-volume (HMLV) semiconductor manufacturing is a demanding proving ground for prognostics and health management (PHM): frequent changeovers, re-entrant routes, and tool-qualification overhead make unexpected failures costly, raising the payoff of health-aware dispatch. We test that promise in a reliability-aware HMLV fab simulator with a 2×2 ablation of two deep reinforcement learning proximal policy optimization (PPO) dispatchers (PHM-aware vs. dispatch-only) and two deployment modes (deterministic argmax vs. stochastic sampling), all sharing reward, training budget, and seeds. Across 10 seeds and 10 held-out HMLV order sets, deployment mode dominated: stochastic PHM-aware PPO cut mean makespan by roughly half, from about 48,900 to 24,400 minutes, and lifted overall equipment effectiveness (OEE) from 0.71 to 0.83, with all differences significant after false discovery rate (FDR) correction, while PHM-aware and dispatch-only PPO were statistically indistinguishable under matched deployment. The result sharpens the PHM-awareness case for HMLV fabs: prognostic value is real but contingent on how learned policies turn information into action.

1. INTRODUCTION

Semiconductor fabs are high-value, high-variability production systems. Dispatch decisions interact with re-entrant process flows, queueing, preventive maintenance, unscheduled downtime, tool qualification, rework, scrap, and highly asymmetric bottlenecks. The prognostic health management (PHM) motivation is straightforward: if a dispatcher knows that a tool is likely to fail, it should be able to route work away from risky equipment, protect lots from interruption, and improve overall fab performance. Modern digital twin and cyber-physical manufacturing work argues for exactly this kind of closed-loop link between sensing, prediction, simulation, and decision support (Lee, Bagheri, & Kao, 2015; Kritzinger, Karner, Traar, Henjes, & Sihn, 2018; Jones, Snider, Nassehi, Yon, & Hicks, 2020; Fuller, Fan, Day, & Barlow, 2020; Tao, Xiao, Qi, Cheng, & Ji, 2022; Sabri et al., 2025; Azevedo et al., 2024; Wang, Zhou, Yang, Li, & Yang, 2022; Soykan, Mondesire, & Rabadi, 2026b).

However, the operational value of prognostics is not guaranteed by accurate health estimation alone. Prognostic models produce information; dispatchers convert information into action. The conversion layer itself can introduce confounds that PHM value-of-information studies often leave uncontrolled. In deep reinforcement learning (RL), a trained stochastic policy can be deployed either by selecting the highest-probability valid action or by sampling from the masked action distribution. PPO is explicitly trained as a stochastic policy-gradient method (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017), and prior RL reproducibility studies show that evaluation choices materially change conclusions (Henderson et al., 2018; Engstrom et al., 2020; Andrychowicz et al., 2021; Huang, Dossa, Raffin, Kanervisto, & Wang, 2022). The deterministic-versus-stochastic deployment gap for entropy-regularized on-policy methods is thus a known RL evaluation pitfall; our aim is not to rediscover it but to quantify its severity in a PHM value-of-information setting, where it can eclipse the operational signal contributed by prognostic observability. What is missing from the PHM literature is exactly this controlled demonstration of how strongly the deployment-mode choice can interact with, and even hide, that signal.

This paper therefore asks a simple question:

In a reliability-aware HMLV semiconductor fab, does giving

a learned dispatcher prognostic equipment-health information improve operations more than the choice of how the policy is deployed?

The contributions are: (1) a controlled 2×2 ablation of prognostic observability and PPO deployment mode in a reliability-aware fab simulator that builds on prior micro-discrete-event manufacturing digital twin work (Nsiye, Wright, Tse, & Mondesire, 2024; Mondesire, Brown, & Soykan, 2025), semiconductor remaining useful life (RUL) research (Mondesire & Wright, 2025), scalable simulation experience (Mondesire, Angelopoulou, Sirigampola, & Goldiez, 2019), and multi-objective RL scheduling work (Soykan, Mondesire, Rabadi, & Bochenek, 2025); (2) statistical evidence that PHM-aware and dispatch-only PPO are indistinguishable under matched deployment, while the within-policy effect of deployment mode is large and significant after false discovery rate (FDR) correction; (3) a methodological recommendation that PHM value-of-information papers using stochastic RL policies report all four cells of the 2×2, because deployment mode can mask or invert the apparent benefit of prognostic observability. The reward is held identical across the two observation conditions and penalizes operational consequences of failures, not health-state features themselves, so that PHM-aware observations have no direct reward advantage over dispatch-only observations.

The paper's main findings are that within-policy improvements from stochastic deployment are larger and more consistent than between-condition improvements from prognostic observability and that PHM-aware and dispatch-only PPO are statistically similar under matched deployment.

2. BACKGROUND AND PROBLEM DEFINITION

The flexible job shop scheduling problem (FJSP) generalizes the classical job shop by letting each operation be assigned to any one of several eligible machines, subject to the per-job operation routing, with non-identical processing times across the eligible machines (Brandimarte, 1993; Pinedo, 2016). High-mix low-volume (HMLV) semiconductor manufacturing is naturally an instance of FJSP at scale: every product follows a fixed re-entrant route of dozens to hundreds of operations (for example lithography, etch, implant, deposition, chemical-mechanical planarization (CMP), metrology), but most of those operations can be processed on any of several qualified tools inside a tool group, with non-identical setups, recipes, and qualification states (Kopp, Hassoun, Kalir, & Mönch, 2020; Stöckermann et al., 2023). The dispatcher must repeatedly decide, in real time, which feasible (job/wafer lot, machine/tool) pair to commit next, while tool failures, planned maintenance, qualification expiry, and degraded process states shift the feasible set. FJSP is NP-hard (Brandimarte, 1993); in HMLV semiconductor fabs, the additional uncertainty introduced by equipment health makes purely offline planning brittle and motivates dispatch policies that react to PHM signals at decision time.

We formalize the resulting sequential FJSP with equipment health information as follows. Let J denote the set of jobs, E the set of equipment, and O_j the ordered operation route for job $j \in J$. At each decision epoch t, the simulator contains queue, route, equipment, maintenance, failure, and sensor information. We summarize that complete simulator state as

$$s_t = (Q_t, B_t, M_t, H_t, R_t, X_t), \tag{1}$$

where Q_t is the waiting work-in-process (WIP) and queue state, B_t is the state of busy equipment and remaining processing time, M_t is preventive-maintenance, repair, and qualification state, H_t is latent or estimated equipment health, R_t is route, operation, recipe, and due-date context, and X_t is recent sensor, alarm, utilization, and reliability history. An action $a_t = (j, e)$ dispatches the next feasible operation of job j to compatible equipment e. The configuration used here has up to 100 lots and 20 tools, giving a fixed discrete action space of $100 \times 20 = 2000$ candidate (j, e) pairs. At each epoch a binary mask exposes only the feasible subset $\mathcal{A}_t$, and actions outside $\mathcal{A}_t$ are invalid because the job, operation, equipment, qualification, or equipment state does not permit the assignment; the mask therefore defines the parameter space over which the reward is exercised, and $|\mathcal{A}_t|$ is typically far smaller than 2000.

The central PHM distinction is not whether the simulator has hidden health, but what the policy is allowed to observe. The dispatch-only observation o_t^D is produced by the map $g_D(s_t)$ and includes scheduling and equipment-availability information. The PHM-aware observation o_t^P is produced by $g_P(s_t)$ and appends prognostic fields z_t^{PHM} to the dispatch state:

$$o_t^D = g_D(s_t), \qquad o_t^P = g_P(s_t) = (g_D(s_t), z_t^{PHM}). \tag{2}$$

In Eq. (2), z_t^{PHM} contains health estimates, time since maintenance, recent alarms, sensor summaries, failure-risk indicators, and RUL-like quantities. In semiconductor practice these correspond to concrete prognostic signals such as ion-mill etch chamber RUL from flow, pressure, and RF-sensor histories (PHM Society, 2018; Singh, Selvanathan, Zope, Nistala, & Runkana, 2018), CMP pad-wear and material-removal-rate indicators (PHM Society, 2016), alarm rates and qualification age on lithography and deposition tools, and tool-group RUL aggregations (Mondesire & Wright, 2025; Wright, Soykan, & Mondesire, 2025). The true latent degradation state is not directly rewarded and is not treated as an oracle decision target. RUL is used as a decision input whose operational value must appear indirectly through downstream

fab KPIs such as makespan, overall equipment effectiveness (OEE), unscheduled downtime, and scrap.

2.1. PHM, Digital Twins, and Fab Dispatch

PHM usually begins with sensing, fault diagnosis, degradation assessment, and remaining useful life estimation (Si, Wang, Hu, & Zhou, 2011; Lei et al., 2018). Predictive maintenance (PdM) reviews show that machine-learning PdM systems depend on reliable operating histories, failure labels, and deployment context (Carvalho et al., 2019; Zonta et al., 2020). Semiconductor PHM data are especially difficult because complete tool failure, chamber degradation, repair, and production-disposition histories are rarely public. Public anchors include the PHM 2018 ion mill etch challenge, which provides tool, recipe, sensor, fault, and RUL labels (PHM Society, 2018; Singh et al., 2018), and the PHM 2016 CMP data challenge (PHM Society, 2016). These sources support tool-level prognostic modeling, but they do not by themselves answer whether RUL estimates improve closed-loop fab dispatch.

Fab scheduling research, meanwhile, has mature deterministic and stochastic simulation traditions. SMT2020 provides a public semiconductor manufacturing testbed with preventive maintenance and unscheduled downtime (Kopp et al., 2020). Recent semiconductor dispatch studies examine RL scalability and validation on increasingly realistic fab simulations (Stöckermann, Südfeld, et al., 2025; Stöckermann et al., 2023; Tassel et al., 2023), and recent reviews emphasize deployment and validation as open concerns (Stöckermann, Feudel, et al., 2025). Explainable RL scheduling work also highlights the need to understand what learned semiconductor dispatchers actually use when they make decisions (Immordino et al., 2025).

2.2. Similar Approaches and the Gap

Three research streams motivate this work, but each leaves the PHM value-realization question open. Ion mill RUL studies and the PHM data challenges supply rich sensor, fault, and time-to-failure data for tool-level prognostics (PHM Society, 2018; Singh et al., 2018; PHM Society, 2016) but stop short of closed-loop fab dispatch, so they cannot show whether an RUL estimate changes queueing, OEE, downtime, or rework. Semiconductor fab simulation and dispatch work, including SMT2020 (Kopp et al., 2020) and recent RL studies in front-end fabs (Stöckermann, Südfeld, et al., 2025; Stöckermann et al., 2023; Tassel et al., 2023; Immordino et al., 2025; Stöckermann, Feudel, et al., 2025), gives realistic re-entrant routes, preventive maintenance, and breakdowns, but does not isolate whether a learned dispatcher's gains come from prognostic variables or from how a stochastic policy is executed at deployment.

A third stream learns dispatching policies for job-shop problems using graph representations and reinforcement learning (Zhang et al., 2020; Park, Chun, Kim, Kim, & Park, 2021; Tassel, Gebser, & Schekotihin, 2021), often abstracting away chamber health, repair workflows, and lot-level failure consequences. Recent manufacturing digital-twin work contributes micro-discrete-event simulation, semiconductor RUL modeling, a Gymnasium-compliant micro-DES fab simulator, graph and distributed RL dispatchers for wafer-lot routing, and multi-objective RL scheduling foundations (Nsiye et al., 2024; Mondesire & Wright, 2025; Mondesire et al., 2025; Soykan et al., 2025; Soykan, Mondesire, & Rabadi, 2026a; Hatfield & Mondesire, 2026). The present paper connects those pieces around a sharper gap: PHM-aware RL dispatch studies should not attribute improvement to prognostic observability unless deployment stochasticity has also been controlled.

2.3. Learned Dispatch with PPO

The dispatch task defined in Eq. (1) is a sequential decision problem under uncertainty: at every epoch the controller observes a high-dimensional fab state, optionally including the prognostic fields of Eq. (2), and must commit a single feasible (j, e) action while failures, maintenance, and qualification continually reshape the feasible set $\mathcal{A}_t$. This is exactly the regime where deep reinforcement learning (RL) is attractive for PHM. A static priority rule cannot weigh a tool's predicted remaining useful life against current queue pressure, and an offline optimizer must re-solve whenever health state shifts; an RL dispatcher instead learns a direct mapping from raw health and scheduling observations to actions, reacts to degradation signals at decision time, and amortizes expensive planning into a policy that is cheap to evaluate online. These properties make RL a natural mechanism for closing the loop between tool-level prognostics and shop-floor dispatch in HMLV fabs, the closed-loop link that the digital-twin literature calls for but rarely demonstrates end to end (Lee et al., 2015; Kritzinger et al., 2018).

Among RL methods we adopt Proximal Policy Optimization (PPO) (Schulman et al., 2017) because it is comparatively stable to train, sample-efficient relative to earlier policy-gradient methods, and well suited to the large discrete, masked action space of fab dispatch. PPO optimizes a stochastic policy $\pi_\theta(a|o_t)$, where θ is the vector of learnable actor-critic network parameters, from sampled trajectories using the clipped surrogate objective

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\big[\min\big(\rho_t(\theta)\hat{A}_t, \operatorname{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\big] \tag{3}$$

Here $\rho_t(\theta) = \pi_\theta(a_t|o_t)/\pi_{\theta_{old}}(a_t|o_t)$ is the probability ratio for the selected action, $\hat{A}_t$ is the estimated advantage, and ϵ is

the clipping parameter that limits destructive policy updates. Stochasticity is therefore intrinsic to PPO's exploration and gradient estimation, not an implementation artifact (Sutton & Barto, 2018; Mnih et al., 2015; Haarnoja, Zhou, Abbeel, & Levine, 2018).

This intrinsic stochasticity is harmless during training, but it forces a choice at deployment that is easy to overlook. A trained PPO policy outputs a distribution $\pi_\theta(a|o_t)$ over masked dispatch actions, where o_t is either the dispatch-only or PHM-aware observation. The same trained policy can be converted into actions in two ways:

$$a_t^{det} = \arg\max_{a \in \mathcal{A}_t} \pi_\theta(a|o_t), \qquad a_t^{stoch} \sim \pi_\theta(\cdot|o_t, \mathcal{A}_t). \quad (4)$$

Equation (4) defines the second experimental lever. The deterministic rule a_t^{det} always commits the highest-probability feasible action, whereas the stochastic rule a_t^{stoch} samples from the masked distribution. Deterministic execution is repeatable and easier to audit, while stochastic execution preserves policy diversity when actions have similar estimated value. Modern RL toolkits make the toggle trivial (Raffin et al., 2021; Towers et al., 2024), so it is often treated as an incidental reporting detail. Because prior reproducibility studies show that such evaluation choices can materially change conclusions (Henderson et al., 2018; Engstrom et al., 2020; Andrychowicz et al., 2021; Huang et al., 2022), we instead treat it as a controlled variable. The paper's question is whether differences in operational performance are better explained by the observation map in Eq. (2) or by the deployment rule in Eq. (4).

Figure 1 summarizes the closed-loop problem: prognostic observability modifies the policy input before action probabilities are produced, while deployment stochasticity modifies how those probabilities become a production action. The two levers are usually entangled in RL dispatch studies; here they are deliberately separated.

Both observation conditions are trained against an identical reward with two parts: a per-step term applied at every dispatch decision and a terminal term applied once all lots complete. The per-step term grants a small bonus w_{op} for each completed operation and penalizes operation flow time F_t, queue wait Wq_t, and equipment idle I_t (each normalized by the simulation horizon $T_{\max}$), plus event penalties when the dispatched lot fails, is scrapped, or is reworked:

$$r_t = w_{op}\mathbf{1}[\text{done}] - w_F F_t - w_{Wq} Wq_t - w_I I_t - w_{fl}\mathbf{1}[\text{fail}] - w_{sc}\mathbf{1}[\text{scrap}] - w_{rw}\mathbf{1}[\text{rework}] \quad (5)$$

with w_{op}=0.05, w_F=0.10, w_{Wq}=0.05, w_I=0.02, w_{fl}=0.05, w_{sc}=0.25, and w_{rw}=0.05. The terminal term scores the finished schedule on makespan C, total tardiness Ta, on-time delivery rate ρ_{OT}, OEE, unscheduled downtime U (normalized by capacity time K), and scrap rate ρ_{SR}:

$$r^T = -w_C \tfrac{C}{T_{\max}} - w_{Ta} \tfrac{Ta}{T_{\max}} + w_{OT}\rho_{OT} + w_{OEE}\,\text{OEE} - w_U \tfrac{U}{K} - w_{SR}\rho_{SR} \quad (6)$$

with w_C=1.0, w_{Ta}=1.0, w_{OT}=2.0, w_{OEE}=1.0, w_U=1.5, and w_{SR}=2.0. All weights are fixed and identical for the PHM-aware and dispatch-only conditions. Operation flow time and terminal makespan capture throughput pressure, whereas the tardiness term penalizes due-date lateness, so the two are complementary rather than redundant. Crucially, the reward penalizes only the operational consequences of equipment-health events (failure, downtime, scrap, rework), not the prognostic features themselves. The simulator supports an optional PHM reward-shaping set (failure-risk, short-RUL, and PM-due penalties), but it was disabled for every experiment reported here, so none of the 14 prognostic features in Section 3.2 enters the reward. PHM-aware observations therefore receive no privileged numerical bonus, and any benefit must manifest through better routing.

For hypothesis tests, paired observations are indexed by held-out scenario and reliability seed k. The measured KPI value is denoted by y_k, with superscripts $cand$ and ref indicating the candidate method and reference method. For lower-is-better KPIs such as makespan, tardiness, failures, downtime, and WIP, the candidate advantage is

$$\Delta_k = y_k^{ref} - y_k^{cand}, \quad (7)$$

where positive Δ_k means the candidate has a lower, better value than the reference. For higher-is-better KPIs such as OEE, utilization, quality rate, and moves over inventory, the advantage is

$$\Delta_k = y_k^{cand} - y_k^{ref}. \quad (8)$$

In Eq. (8), positive Δ_k means the candidate has a higher, better value than the reference. We report mean percent advantage, paired t tests, Wilcoxon tests, bootstrap confidence intervals, and Benjamini-Hochberg false-discovery-rate adjusted q values.

2.4. Summary and Research Gap

Tool-level prognostics and realistic fab-dispatch simulators exist separately, but the literature rarely closes the loop to show that a health estimate changes operational KPIs, and when a learned dispatcher does close it, prognostic observability is confounded with the rule used to convert a stochas-

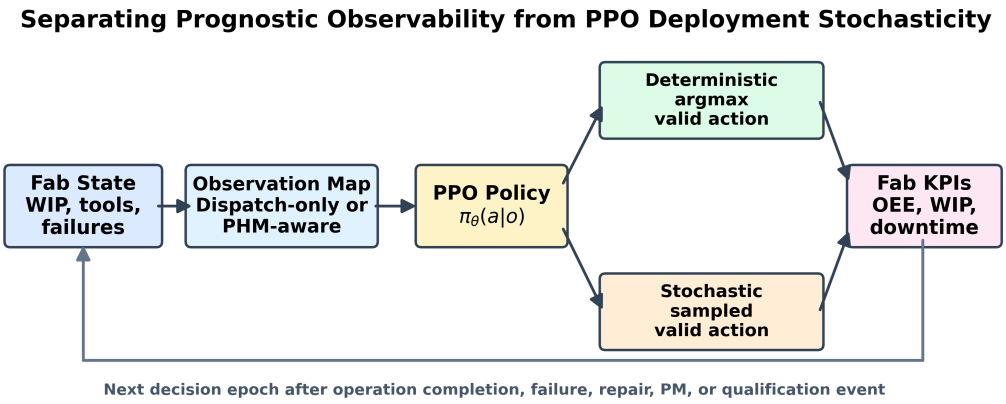

Figure 1. The two experimental levers: prognostic observability controls which health variables enter the policy observation, while deployment stochasticity controls how the trained PPO action distribution is converted into a fab dispatch action.

tic PPO policy into actions. This work isolates the two by varying prognostic observability (Eq. 2) and PPO deployment mode (Eq. 4) independently under a fixed reward (Eq. 5), training budget, and seeds. We test two hypotheses: (H1) PHM-aware PPO outperforms dispatch-only PPO under matched deployment, and (H2) deployment mode is only a secondary effect relative to prognostic observability.

3. METHODOLOGY

The methodology operationalizes the research question as a 2×2 factorial experiment designed to isolate cause rather than simply report the best-performing dispatcher. Two factors are crossed. The first is prognostic observability, with levels dispatch-only and PHM-aware as defined by the observation maps in Eq. (2). The second is PPO deployment mode, with levels deterministic and stochastic as defined in Eq. (4). Crossing the two factors yields four policy cells that are evaluated on a common testbed, and every quantity outside the two factors is held fixed: the simulator, the consequence-penalizing reward of Eq. (5), the training budget, the random seeds, the held-out order sets, and the evaluation metrics.

This design lets each hypothesis be tested by an isolated contrast. Hypothesis H1, that prognostic observability improves dispatch, is the between-condition contrast of PHM-aware against dispatch-only PPO under a matched deployment mode. Hypothesis H2, that deployment mode is only a secondary effect, is tested by comparing the magnitude of the within-policy deterministic-versus-stochastic contrast against that of the H1 contrast. Because no factor other than the two of interest varies across cells, any KPI difference is attributable to observability, to deployment mode, or to their interaction rather than to a confound. The remainder of this section describes the simulator and reliability layer, the four PPO cells and their architecture, the evaluation protocol and operational baselines, and the metrics and statistical tests used to evaluate the contrasts.

3.1. Simulator and Reliability Layer

FabSim (Mondesire et al., 2025), the training environment, is a HMLV semiconductor job-shop simulator that conforms to the Gymnasium API (Towers et al., 2024), the de facto standard reinforcement-learning interface in which an environment exposes `step` and `reset` functions together with declared observation and action spaces; each `step` commits one wafer lot-to-tool assignment and returns a reward. Compliance lets any standard RL library train and evaluate against the simulator without custom integration code, so the same MaskablePPO implementation drives both observation conditions unchanged. The fab configuration is data driven through CSV files for floor configuration, process flows, failure modes, repair profiles, maintenance policies, and sensor profiles. Equipment can be productive, idle, scheduled down, unscheduled down, in repair, in preventive maintenance, in inspection, or in qualification. A technician is assumed to be available for wafer front-opening unified pod (FOUP) movement, consistent with a fully automated 24/7 fab movement assumption. Tool failures can interrupt lots; lot disposition can include success, failure, rework, scrap, restart, or hold. The simulator logs operation histories, equipment status histories, failure events, repair events, preventive-maintenance events, sensor samples, RUL predictions, and Gantt charts.

The reliability model is synthetic but PHM-structured. Failure modes are configured with hazard and degradation parameters: each tool family can carry multiple failure modes, rather than a single one, each with its own Weibull hazard and degradation rate, so a tool can fail in several distinct ways and at different rates of progression (Wright, Nsiye, & Mondesire, 2024) (the present configuration uses 13 modes across 9 tool families, several families having two, with Weibull shape parameters from 1.3 to 3.0). Repair profiles include diagnosis, parts retrieval, active repair, and qualification time, and sensor/RUL traces expose observable prognostic indicators. This is aligned with the PHM premise that health information

Table 1. Observation inputs used by the PPO conditions.

Input group	Dispatch-only	PHM-aware
Job state	Job availability time; next operation one-hot vectors; next operation durations.	Same.
Equipment dispatch state	Available time; productive, standby, down, repair, qualification, and degraded states; time in state; qualification flag; capacity fraction.	Same.
Prognostic equipment state	Not visible.	Runtime, idle time, wafers, cycles/time since PM, time since failure, predicted RUL, failure risk, health index, alarm rate, downtime ratio, PM-due flag, sensor mean/std.

should be contextualized across manufacturing levels (Sharp & Weiss, 2018), while acknowledging that public semiconductor failure data are incomplete.

3.2. PPO Policy Conditions and Architecture

The two levels of the observability factor are realized as two MaskablePPO policy families, each trained for 10 million steps across the same 10 seeds. Table 1 summarizes their observation inputs. Dispatch-only PPO receives ordinary dispatch state, while PHM-aware PPO receives the same dispatch state plus the prognostic equipment fields; there are no inputs exclusive to dispatch-only PPO, so the two conditions differ only by the addition of health information.

Both observation conditions use the same MaskablePPO actor-critic from `sb3-contrib` (Raffin et al., 2021) on top of the Stable-Baselines3 Python library. The actor and critic are independent two-layer multilayer perceptrons (MLPs) of width 256 with `tanh` activations; the actor outputs an unnormalized logit per discrete (job, equipment) pair, and an action mask zeroes the probability of infeasible dispatch actions before sampling or argmax. The only difference between the PHM-aware and dispatch-only conditions is the input dimensionality of the actor and critic networks, which grows by the 14 prognostic equipment features in the PHM-aware condition. Holding the architecture, reward, and hyperparameters identical across conditions ensures that any difference between them reflects the added health information rather than a modeling change. Table 2 reports the PPO hyperparameters used for all 10 seeds in both conditions; the learning rate decays linearly with the remaining training progress.

3.3. Evaluation Protocol and Baselines

For each family, we evaluate the final 10M checkpoint and the validation-selected checkpoint. The validation-selected checkpoint is selected using checkpoint order sets only; the held-out test order sets never influence training or model selection. Each selected policy is then evaluated under both levels of the deployment factor, deterministic and stochastic, which completes the four PPO cells of the 2×2. The same 10 held-out order sets and the same 10 reliability seeds are used across all PPO and baseline methods, so every method is scored on identical scenarios.

Table 2. MaskablePPO hyperparameters and architecture, identical for PHM-aware and dispatch-only conditions across all 10 seeds.

Setting	Value
Algorithm	MaskablePPO (sb3-contrib)
Actor / critic network	MLP, 2 hidden layers, width 256, tanh
Total training steps	10,000,000 per seed
Rollout horizon n_{steps}	1,024
Minibatch size	256
Optimization epochs	6
Discount γ	0.99
Generalized advantage estimation (GAE) λ	0.95
Entropy coefficient	0.01
PPO clip ϵ	0.2
Learning rate	linear decay from 3×10^{-4}
Action mask	infeasible (job,equipment) pairs zeroed pre-softmax
Seeds	0–9 (10 independent runs per condition)

Baselines are priority dispatching rule Most Remaining Work (MRW), representative of the greedy heuristics used for HMLV wafer dispatching (Mondesire & Soykan, 2026a), and constraint-programming satisfiability (CP-SAT) with 60 s and 600 s time limits using OR-Tools (Google, 2025). CP-SAT and MRW are treated as operational baselines, not PHM-aware agents. A 1 s CP-SAT setting was explored but produced no feasible solution within the time budget on any of the 100 held-out scenarios. Because the original run logic falls back to MRW when no feasible plan is available, the 1 s output is numerically identical to the MRW baseline; we therefore exclude it from the reported CP-SAT comparisons to avoid presenting MRW under two different names. CP-SAT 60 s and CP-SAT 600 s returned the FEASIBLE status (not OPTIMAL) on every held-out scenario, so the CP-SAT numbers in this paper are best-found feasible schedules within the time budget rather than provably optimal makespans. The comparison therefore separates three questions: whether PPO learned a nontrivial policy at all, whether prognostic observability helped under matched deployment (H1), and whether deployment mode altered PPO behavior more than observability did (H2).

3.4. Metrics and Statistical Evaluation

The main KPIs are makespan, total tardiness, overall equipment effectiveness (OEE), failure count, unscheduled downtime, average WIP, moves over inventory, productive utiliza-

tion, quality rate, rework count, and scrap count. These represent schedule performance, reliability burden, equipment productivity, flow, and quality. RUL is not scored directly. Instead, RUL-like information is available to the PHM-aware policy as a decision variable, and its value is judged by downstream operational KPIs. Consistent with the PHM emphasis of this study, the reliability and equipment KPIs (failures, unscheduled downtime, OEE, utilization, scrap, and rework) are the quantities through which prognostic value, if present, should appear.

For each contrast, KPI values are paired by held-out scenario and reliability seed and quantified with the direction-aware advantage of Eqs. (7) and (8). Statistical support follows the procedure defined in Section 2: paired t tests, Wilcoxon signed-rank tests, bootstrap confidence intervals, and Benjamini-Hochberg FDR-adjusted q values across the full KPI family. We report the paired t test and the Wilcoxon signed-rank test together deliberately: several KPIs (for example failure and scrap counts) are discrete or skewed, so rather than assume normality of the $n = 100$ paired differences we use the t test as a parametric summary and the distribution-free Wilcoxon test as a robustness check, and we only draw conclusions where the two agree. We treat a contrast as detectable only when its FDR-adjusted q value falls below 0.05, which is the criterion used to adjudicate H1 and H2.

3.5. Model Verification and Validation

All experiments of this work were performed through simulations. Therefore, the deployed simulator (FabSim) was evaluated through a standard test, verification, and validation process established for discrete-event and PHM simulation studies (Sargent, 2013). Conceptual-model and data validity come first: the re-entrant routing, tool-group eligibility, reliability hazards, repair and qualification workflows, and sensor or RUL signals of Section 3.1 are grounded in public PHM and semiconductor references rather than chosen freely (PHM Society, 2018, 2016; Kopp et al., 2020; Sharp & Weiss, 2018). Computerized-model verification then confirmed the implementation through smoke and integration tests that assert Gymnasium API conformance (Towers et al., 2024), deterministic reproducibility under the fixed seeds used throughout, trace consistency among the operation, equipment, failure, and repair event logs and the Gantt reconstruction, action masks that admit only the feasible (j, e) pairs of $\mathcal{A}_t$, and conservation of lots across success, rework, scrap, restart, and hold dispositions. Operational validation uses extreme-condition and face-validity checks: emergent KPIs such as makespan and OEE respond sensibly to offered load and are sanity-checked against the MRW and CP-SAT baselines (Google, 2025) on the held-out scenarios. Finally, simulated processing times, operation durations, failure rates, and repair durations were compared against a proprietary, closed-access HMLV production dataset and matched the observed plant statistics within one standard deviation of their means. The underlying records are confidential, but this calibrates the reliability layer against a real line rather than public proxies alone.

4. RESULTS

The results are organized around the research question. First, we verify that PPO training produced nontrivial learned policies. Second, we compare deterministic and stochastic deployment within each PPO condition. Third, we compare PHM-aware and dispatch-only observations under matched deployment modes. Finally, we place stochastic PPO in context against MRW and CP-SAT baselines.

4.1. Training Produced Nontrivial Policies

Figure 2 plots validation checkpoint scores during PPO training, computed on checkpoint order sets only. The aggregate validation score rises steeply early in training and then plateaus, and the validation-selected checkpoint is taken at that plateau. This improvement over the earliest checkpoints is the direct evidence that checkpoint selection acted on a genuine learning signal rather than noise. The study therefore compares learned policies under different observation and deployment modes, not baselines against randomly initialized policies.

4.2. Deployment Stochasticity Dominated PPO Performance

Figure 3 and Table 3 show the central result. Under validation-selected PHM-aware PPO, stochastic deployment reduced mean makespan from 48,939.9 to 24,387.2, reduced total tardiness from 2.515×10^6 to 1.292×10^6, improved OEE from 0.707 to 0.829, reduced failures from 722.5 to 508.0, and reduced unscheduled downtime from 10,025.2 to 6,467.8. The same pattern appeared for dispatch-only PPO. The cost was higher average WIP: deterministic PHM-aware PPO had mean WIP 53.8 versus 57.9 for stochastic PHM-aware PPO. Together these comparisons are the clearest evidence that action-selection mode is not a harmless evaluation detail. As Section 4.4 quantifies, this within-PPO contrast is driven substantially by an unusually weak deterministic policy, so "dominated" refers to the size of the deployment-mode effect relative to prognostic observability, not to stochastic PPO being strong in absolute terms.

4.3. PHM Observability Was Not Sufficient for Dominance

Table 4 compares PHM-aware PPO to dispatch-only PPO under the same deployment mode. Under deterministic validation-selected deployment, PHM-aware PPO slightly improved makespan and OEE, but not with strong FDR-

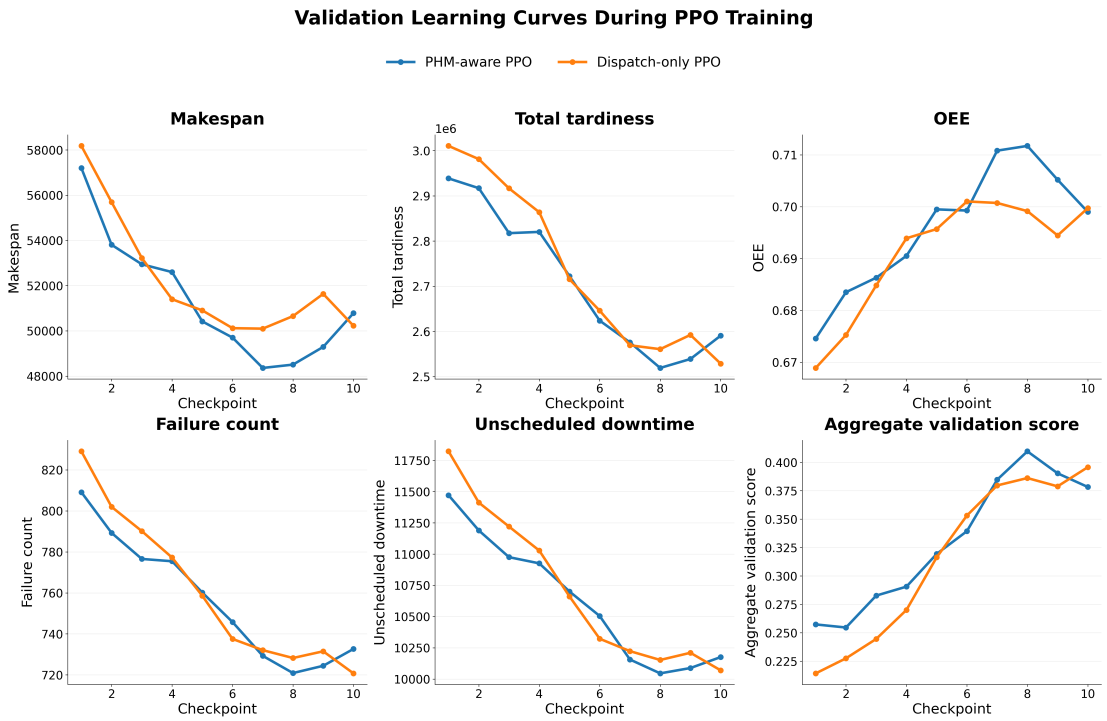

Figure 2. Validation learning curves for deterministic PPO checkpoints, including the aggregate validation score (final panel) used for checkpoint selection. The curves use checkpoint order sets only and never touch the held-out test data.

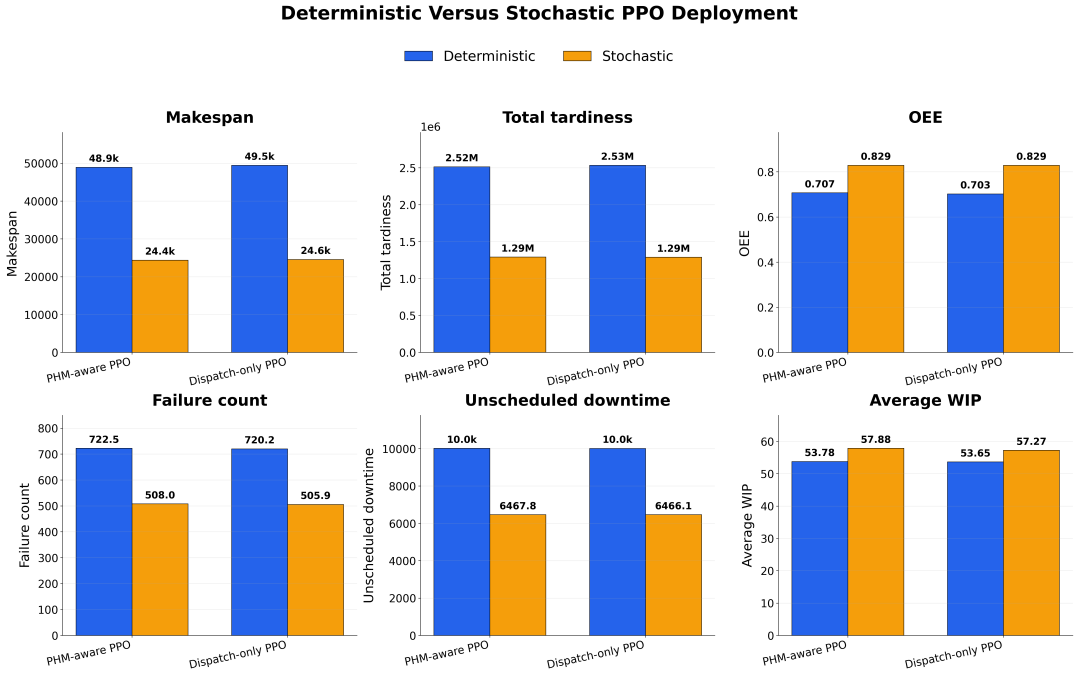

Figure 3. Deterministic versus stochastic PPO deployment for validation-selected policies. Bars are color-coded by action-selection mode, with value labels above bars.

corrected significance. Under stochastic deployment, PHM-aware and dispatch-only PPO were statistically similar across the main KPIs. For example, PHM-aware stochastic PPO had makespan 24,387.2 versus 24,576.8 for dispatch-only stochastic PPO, but the FDR-corrected q value was 0.499. OEE was essentially tied: 0.82915 versus 0.82920.

In this experimental setting, access to prognostic fields did not translate into a robust, statistically dominant dispatch

Table 3. Validation-selected PPO: deterministic versus stochastic deployment. Negative advantage means deterministic deployment was worse than stochastic for that KPI direction. All shown differences are FDR-significant. The d_z column is the paired Cohen's d_z effect size, and "Det. better" is the fraction of the $n = 100$ paired (scenario, seed) evaluations in which deterministic deployment gave the better value.

PPO condition	KPI	Det. mean	Stoch. mean	Det. adv. %	d_z	q	Det. better
PHM-aware	Makespan	48,939.9	**24,387.2**	-101.8	-6.04	$< 10^{-68}$	0.00
PHM-aware	Total tardiness	2,515,049	**1,292,305**	-95.4	-6.71	$< 10^{-68}$	0.00
PHM-aware	OEE	0.707	**0.829**	-14.6	-4.63	$< 10^{-68}$	0.00
PHM-aware	Failure count	722.5	**508.0**	-42.6	-5.38	$< 10^{-68}$	0.00
PHM-aware	Unscheduled downtime	10,025.2	**6,467.8**	-56.0	-4.65	$< 10^{-68}$	0.00
PHM-aware	Average WIP	**53.8**	57.9	6.9	1.04	2.17×10^{-17}	0.87
Dispatch-only	Makespan	49,460.6	**24,576.8**	-102.1	-5.94	$< 10^{-68}$	0.00
Dispatch-only	Total tardiness	2,534,706	**1,289,783**	-97.3	-7.75	$< 10^{-68}$	0.00
Dispatch-only	OEE	0.703	**0.829**	-15.3	-4.93	$< 10^{-68}$	0.00
Dispatch-only	Failure count	720.2	**505.9**	-42.7	-5.41	$< 10^{-68}$	0.00
Dispatch-only	Unscheduled downtime	10,018.9	**6,466.1**	-55.5	-5.43	$< 10^{-68}$	0.00
Dispatch-only	Average WIP	**53.6**	57.3	6.1	0.90	2.20×10^{-14}	0.84

Table 4. PHM-aware PPO versus dispatch-only PPO under matched deployment mode. Positive advantage means PHM-aware PPO is better. Bold indicates the numerically better value between PHM-aware and dispatch-only on each row; "PHM better" is the fraction of the $n = 100$ paired (scenario, seed) evaluations in which PHM-aware PPO gave the better value; the "tied" column marks rows where the FDR-corrected q value exceeds 0.05, indicating that no row reaches statistically detectable significance.

Deployment	KPI	PHM-aware	Dispatch-only	PHM adv. %	q	PHM better	Tied
Deterministic	Makespan	**48,939.9**	49,460.6	0.42	0.340	0.56	yes
Deterministic	OEE	**0.707**	0.703	0.81	0.112	0.59	yes
Deterministic	Failure count	722.5	**720.2**	-0.56	0.646	0.46	yes
Deterministic	Unscheduled downtime	10,025.2	**10,018.9**	-0.48	0.943	0.50	yes
Stochastic	Makespan	**24,387.2**	24,576.8	0.27	0.499	0.49	yes
Stochastic	OEE	0.82915	**0.82920**	0.03	0.985	0.50	yes
Stochastic	Failure count	508.0	**505.9**	-0.71	0.624	0.47	yes
Stochastic	Unscheduled downtime	6,467.8	**6,466.1**	-0.49	0.984	0.52	yes
Stochastic	Average WIP	57.9	**57.3**	-1.24	0.101	0.41	yes

policy under an equal non-PHM reward. Several mechanisms are plausible. Dispatch-only observations may already encode indirect reliability information through availability, state, and queue dynamics. The reward may not strongly reward preemptive health-aware routing, or the PPO policy class may need a representation that better binds tool health to lot risk. The result is valuable because it identifies where PHM value was lost: between health-state observability and action choice.

These two findings are not an artifact of validation-based checkpoint selection. Both replicate on the final 10M-step checkpoint. There, stochastic deployment again cut mean makespan from 49,649.7 to 25,072.9 for PHM-aware PPO and from 50,667.9 to 25,329.9 for dispatch-only PPO (FDR-significant), while PHM-aware and dispatch-only PPO remained statistically indistinguishable under matched deployment (final-checkpoint makespan 49,649.7 versus 50,667.9, $q = 0.099$; OEE $q = 0.561$). The dominance of deployment mode and the PHM-versus-dispatch null therefore hold across both checkpoint-selection rules.

4.4. Stochastic PPO in Baseline Context

The operational baselines remain important and they reveal an asymmetry that should not be hidden. Figure 4 places deterministic and stochastic PPO beside MRW and CP-SAT. The headline change inside PPO is partly an artifact of how poorly the deterministic policies dispatched: mean makespan of 48,939.9 for deterministic PHM-aware PPO is roughly 2.3× the MRW baseline of 21,276.9 on the same scenarios, indicating that argmax-on-mask converged to a poor local mode for the MaskablePPO architecture in this fab. The large within-PPO gap from deterministic to stochastic deployment therefore should not be read as evidence that stochastic sampling is unusually beneficial; it is partly that deterministic deployment was unusually weak. Mean makespan was 17,693.5 for CP-SAT 600 s (best-found feasible), 20,483.0 for CP-SAT 60 s, 21,276.9 for MRW, 24,387.2 for PHM-aware stochastic PPO, and 24,576.8 for dispatch-only stochastic PPO. OEE was 0.906 for CP-SAT 600 s, 0.888 for CP-SAT 60 s, and 0.829 for both stochastic PPO policies. Stochastic PPO never outperformed either CP-SAT setting on any of the schedule or reliability KPIs.

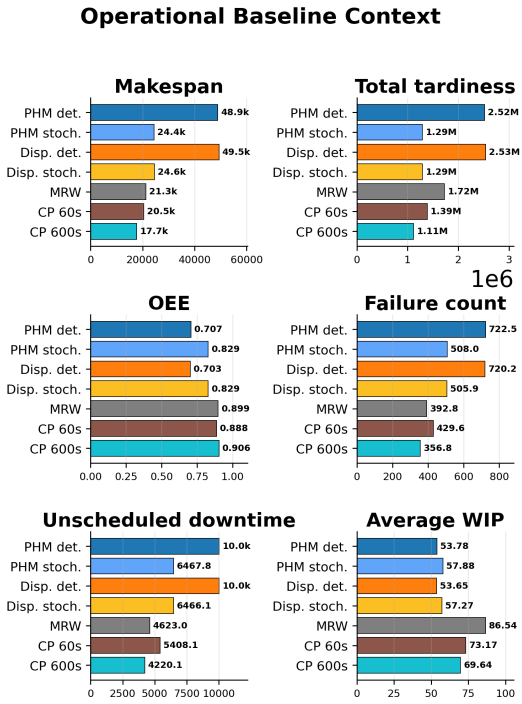

Figure 4. Stochastic and deterministic PPO in baseline context. Value labels above bars use a consistent color palette across PPO, MRW, and CP-SAT conditions.

Table 5 shows the tradeoff. Stochastic PPO improved over MRW and CP-SAT 60 s on total tardiness, and over all three baselines on average WIP, but it was worse than MRW and both CP-SAT settings on makespan, OEE, failures, and downtime. The careful conclusion is therefore not that stochastic PPO is a competitive dispatcher, but that stochastic deployment moves PPO from clearly inferior to MRW (under deterministic deployment) to mixed-but-still-inferior to CP-SAT 600 s on the schedule and reliability KPIs that matter most in a fab. The 2×2 ablation is meaningful inside the PPO family, not as a claim of operational dominance.

Table 5. Validation-selected stochastic PPO versus operational baselines. Positive advantage means the PPO candidate is better than the reference after KPI direction is applied.

Candidate	Reference	Mksp.%	Tard.%	OEE%	WIP%
PHM PPO (stoch.)	MRW	-14.6	**25.0**	-7.7	**33.1**
Disp. PPO (stoch.)	MRW	-15.6	**25.0**	-7.7	**33.8**
PHM PPO (stoch.)	CP-SAT 60 s	-20.2	**4.9**	-6.6	**20.6**
Disp. PPO (stoch.)	CP-SAT 60 s	-20.8	**5.3**	-6.6	**21.5**
PHM PPO (stoch.)	CP-SAT 600 s	-37.9	-16.6	-8.5	**16.5**
Disp. PPO (stoch.)	CP-SAT 600 s	-38.9	-16.5	-8.5	**17.4**

5. DISCUSSION

These results advance how the field can measure and realize the value of prognostics in closed-loop dispatch. By holding the simulator, reward, training budget, and seeds fixed and varying only prognostic observability and deployment mode, the study isolates a factor that prior PHM value-of-information work has largely left uncontrolled, and shows that it can rival or exceed the apparent contribution of health information itself. The constructive finding is that this work identifies a concrete, inexpensive lever for improving learned fab dispatch, and it gives the community a clear methodological standard for attributing operational gains to prognostics.

This work asked whether, in a reliability-aware HMLV fab, prognostic observability improves dispatch more than the choice of deployment mode, and framed the question as two hypotheses. The experiments answer the question and resolve both. Hypothesis H1, that PHM-aware PPO outperforms dispatch-only PPO under matched deployment, was not supported: the two conditions were statistically indistinguishable across the main KPIs under both deterministic and stochastic deployment, and this null held on the validation-selected and the final 10M checkpoints alike. Hypothesis H2, that deployment mode is only a secondary effect, was rejected in the opposite direction: switching a fixed policy from deterministic to stochastic execution moved makespan, tardiness, OEE, failures, and downtime far more than adding prognostic observability did, with very large paired effect sizes ($|d_z| > 4$) and negligible q values. The direct answer to the research question is therefore that, in this controlled setting, operational outcomes depended more on how the learned policy was executed than on whether it could observe equipment health.

The within-policy deployment effect is large partly because the deterministic policies were unusually weak, dispatching at roughly twice the makespan of the simple MRW rule, not because stochastic sampling is intrinsically powerful (Section 4.4). The claim that survives is the comparison of magnitudes: under an identical reward, architecture, and training budget, the deployment-mode contrast dwarfed the observability contrast, independent of the absolute quality of either deployment.

This is a useful negative result for PHM. It suggests that prognostic information does not automatically produce operational value merely because it is exposed to an RL state vector. Value realization depends on reward sensitivity, action constraints, learned representation, evaluation mode, and how policy probabilities are converted into production actions. The finding is consistent with broader evidence that RL conclusions are sensitive to implementation and evaluation details (Henderson et al., 2018; Engstrom et al., 2020; Andrychowicz et al., 2021; Huang et al., 2022), and it echoes automation research showing that how an automated decision aid is used can affect trust, workload, and performance beyond the nominal quality of the underlying information (Schreck et al., 2025).

The main implication of this work is that PHM-aware decision work using stochastic RL policies should report all four

cells of the deployment-by-observability design: PHM-aware deterministic, PHM-aware stochastic, dispatch-only deterministic, and dispatch-only stochastic. Without these four cells, it is difficult to know whether prognostic state, policy sampling, or their interaction produced the reported effect, which is precisely the confound this paper set out to isolate.

The second implication is for reward design. This study used a reward that penalizes the operational consequences of failures (downtime, scrap, rework) but does not see the 14 prognostic features directly. That choice is scientifically clean for the 2×2 ablation because both observation conditions are optimized against the same objective. It may, however, partially explain why PHM-aware observations were not strongly exploited: the consequence terms already give a learned policy an implicit pressure to avoid bad health states, which is much of what prognostic features would otherwise have added. Two caveats follow directly. First, the reward weights are fixed and not equal across terms (Section 2), so the reported balance between prognostic observability and deployment mode is conditional on this particular weighting; a reward that priced downtime or lost qualification more heavily could shift the balance toward PHM-aware observations. Second, the downtime cost here is static and uniform across lots and tools, which limits how much preemptive health-aware routing can pay off, so a criticality- or risk-weighted downtime cost is a natural way to make the comparison more discriminating.

The third implication concerns continual and multi-objective learning. The deterministic PPO policies showed weak deployment behavior despite long training. Catastrophic forgetting and representation drift are known challenges in learning systems (Mondesire & Wiegand, 2023); the checkpoint and deployment-mode results here suggest that multi-objective fab dispatch may need stronger model selection, ensembles, risk-aware action constraints, or continual validation.

Taken together, these contributions strengthen the foundation for trustworthy closed-loop PHM in semiconductor manufacturing. The controlled 2×2 ablation gives the field a reusable template for separating what a prognostic signal contributes from how a learned policy is executed, so that future claims of PHM value rest on evidence rather than on an uncontrolled evaluation choice. The deployment-mode result hands practitioners a low-cost, immediately actionable lever for improving fab dispatch from already-trained policies, with no additional sensing investment. And by pinpointing where prognostic value was not yet realized, between health-state observability and action choice, the study directs reward design, risk-aware action constraints, and richer policy architectures toward the place they will matter most. For high-mix low-volume fabs, where the operational cost of unexpected failures is greatest, these are concrete and encouraging steps toward turning equipment-health information into dependable production gains.

6. CONCLUSION

This paper studied a single PHM value-realization question in a reliability-aware HMLV fab simulator: whether giving a learned dispatcher prognostic equipment-health information improves operations more than the choice of how the trained policy is deployed. A controlled 2×2 ablation crossed prognostic observability (PHM-aware versus dispatch-only) with deployment mode (deterministic versus stochastic) under an identical reward, architecture, training budget, and seeds, evaluated across 10 seeds and 10 held-out order sets. The answer is clear. Deployment mode was the larger driver: stochastic execution cut mean makespan from 48,939.9 to 24,387.2 minutes and lifted OEE from 0.707 to 0.829 with very large, FDR-significant effects, while PHM-aware and dispatch-only PPO were statistically indistinguishable under matched deployment, a null that held on both the validation-selected and final checkpoints. Hypothesis H1 was therefore not supported and H2 was reversed, although the size of the deployment effect partly reflects an unusually weak deterministic policy rather than any intrinsic superiority of sampling.

These findings sharpen the case for PHM-aware modeling within flexible job shop-related solutions. They show that health estimates create operational value only when the decision policy can use them reliably and under a deployment rule that does not hide or distort their contribution. The study's lasting contributions are the controlled ablation that disentangles prognostic observability from policy execution, the statistical evidence that the two are easily confounded, and the practical recommendation that closed-loop PHM studies report all four cells of the deployment-by-observability design before attributing operational gains to RUL or health-state information. For high-mix low-volume semiconductor manufacturing, this offers both a rigorous evaluation standard and an inexpensive, immediately usable lever for improving learned fab dispatch.

Future work. The central next step is to make prognostic information change dispatch actions when it should, rather than simply adding more prognostic variables. This points to PHM-shaped rewards that explicitly price failure risk, lost qualification, and RUL uncertainty rather than only their downstream consequences, and to policy architectures that bind jobs, chambers, tools, recipes, and degradation modes as connected entities rather than a flat feature vector, with generative and diffusion-based schedulers as a further alternative (Mondesire & Soykan, 2026b). Deployment should be treated as a first-class design choice: risk-constrained stochastic execution, gated by action-risk thresholds, RUL confidence bounds, or human supervision, could retain the gains of sampling while remaining auditable for production

use. Finally, calibrating failure, repair, and qualification distributions against real fab logs and adding online CP-SAT replanning baselines would strengthen external validity and tighten the comparison between planning and dispatch.

REFERENCES

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., ... Bachem, O. (2021). *What matters in on-policy reinforcement learning? a large-scale empirical study.* Retrieved from https://arxiv.org/abs/2006.05990 doi: 10.48550/arXiv.2006.05990

Azevedo, R., Amon, M. J., Anderson, M., Mondesire, S., Sanz, F.-G., Sottilare, R., & Wiedbusch, M. (2024). Human digital twins to support nurse practitioner's clinical decision-making using multimodal data: A theoretical, methodological, and analytical framework. In *Digital twin* (pp. 149–172). Springer Nature. Retrieved from https://doi.org/10.1007/978-3-031-67778-6_7 doi: 10.1007/978-3-031-67778-6_7

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, *41*(3), 157–183. Retrieved from https://doi.org/10.1007/BF02023073 doi: 10.1007/BF02023073

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. d. P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024. Retrieved from https://doi.org/10.1016/j.cie.2019.106024 doi: 10.1016/j.cie.2019.106024

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., & Madry, A. (2020). Implementation matters in deep policy gradients: A case study on PPO and TRPO. In *International conference on learning representations.* Retrieved from https://openreview.net/forum?id=r1etN1rtPB

Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, *8*, 108952–108971. Retrieved from https://doi.org/10.1109/ACCESS.2020.2998358 doi: 10.1109/ACCESS.2020.2998358

Google. (2025). *OR-Tools: CP-SAT solver.* Retrieved from https://developers.google.com/optimization/cp/cp_solver

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 1861–1870). Retrieved from https://proceedings.mlr.press/v80/haarnoja18b.html

Hatfield, C., & Mondesire, S. C. (2026). Distributed proximal policy optimization for wafer-lot dispatching in semiconductor manufacturing digital twins with 2D3PO. In *Proceedings of the winter simulation conference (wsc).*

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep reinforcement learning that matters. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32). Retrieved from https://doi.org/10.1609/aaai.v32i1.11694 doi: 10.1609/aaai.v32i1.11694

Huang, S., Dossa, R. F. J., Raffin, A., Kanervisto, A., & Wang, W. (2022). *The 37 implementation details of proximal policy optimization.* Retrieved from https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/

Immordino, A., Stöckermann, P., Hayen, N., Altenmüller, T., Susto, G. A., Gebser, M., ... Seidel, G. (2025). Explainable AI for reinforcement learning based dynamic scheduling solutions in semiconductor manufacturing. *Journal of Intelligent Manufacturing*. Retrieved from https://doi.org/10.1007/s10845-025-02631-3 doi: 10.1007/s10845-025-02631-3

Jones, D., Snider, C., Nassehi, A., Yon, J., & Hicks, B. (2020). Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, *29*, 36–52. Retrieved from https://doi.org/10.1016/j.cirpj.2020.02.002 doi: 10.1016/j.cirpj.2020.02.002

Kopp, D., Hassoun, M., Kalir, A., & Mönch, L. (2020). SMT2020: A semiconductor manufacturing testbed. *IEEE Transactions on Semiconductor Manufacturing*, *33*(4), 522–531. Retrieved from https://doi.org/10.1109/TSM.2020.3001933 doi: 10.1109/TSM.2020.3001933

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. In *Ifac-papersonline* (Vol. 51, pp. 1016–1022). Retrieved from https://doi.org/10.1016/j.ifacol.2018.08.474 doi: 10.1016/j.ifacol.2018.08.474

Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23. Retrieved from https://doi.org/10.1016/j.mfglet.2014.12.001 doi: 10.1016/j.mfglet.2014.12.001

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018).

Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834. Retrieved from https://doi.org/10.1016/j.ymssp.2017.11.016 doi: 10.1016/j.ymssp.2017.11.016

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*, 529–533. Retrieved from https://doi.org/10.1038/nature14236 doi: 10.1038/nature14236

Mondesire, S. C., Angelopoulou, A., Sirigampola, S., & Goldiez, B. (2019). Combining virtualization and containerization to support interactive games and simulations on the cloud. *Simulation Modelling Practice and Theory*, *93*, 233–244. Retrieved from https://doi.org/10.1016/j.simpat.2018.08.005 doi: 10.1016/j.simpat.2018.08.005

Mondesire, S. C., Brown, M., & Soykan, B. (2025). Fabsim: A micro-discrete event simulator for machine learning dynamic job shop scheduling optimizers. In *Proceedings of the winter simulation conference.* Retrieved from https://www.informs-sim.org/wsc25papers/inv186.pdf

Mondesire, S. C., & Soykan, B. (2026a). Automated wafer dispatching through greedy multi-heuristic scheduling in high-mix low-volume wafer fabs. In *Advanced semiconductor manufacturing conference (asmc).*

Mondesire, S. C., & Soykan, B. (2026b). Real-time diffusion scheduling for job shop optimization. In *Proceedings of the winter simulation conference (wsc).*

Mondesire, S. C., & Wiegand, R. P. (2023). Mitigating catastrophic forgetting with complementary layered learning. *Electronics*, *12*(3), 706. Retrieved from https://doi.org/10.3390/electronics12030706 doi: 10.3390/electronics12030706

Mondesire, S. C., & Wright, T. (2025). A tool grouping-based approach to remaining useful life prediction for predictive maintenance in semiconductor manufacturing. In *Advanced semiconductor manufacturing conference (asmc): Big data management and machine learning.*

Nsiye, E., Wright, T., Tse, B., & Mondesire, S. C. (2024). A micro-discrete event simulation environment for production scheduling in manufacturing digital twins. In *Modsim world.* Retrieved from https://www.modsimworld.org/papers/2024/MODSIM_2024_paper_27.pdf

Park, J., Chun, J., Kim, S. H., Kim, Y., & Park, J. (2021). *Learning to schedule job-shop problems: Representation and policy learning using graph neural network and reinforcement learning.* Retrieved from https://arxiv.org/abs/2106.01086 doi: 10.48550/arXiv.2106.01086

PHM Society. (2016). *2016 PHM data challenge: Chemical mechanical polishing data set.* Retrieved from https://phmsociety.org/conference/annual-conference-of-the-phm-society/annual-conference-of-the-prognostics-and-health-management-society-2016/phm-data-challenge-4/

PHM Society. (2018). *2018 PHM data challenge: Ion mill etch tool fault and remaining useful life data.* Retrieved from https://c3.ndc.nasa.gov/dashlink/resources/1009/

Pinedo, M. L. (2016). *Scheduling: Theory, algorithms, and systems* (5th ed.). Springer. Retrieved from https://doi.org/10.1007/978-3-319-26580-3 doi: 10.1007/978-3-319-26580-3

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, *22*(268), 1–8. Retrieved from https://www.jmlr.org/papers/v22/20-1364.html

Sabri, S., Aghaabbasi, M., Reay Atkinson, S., Amon, M. J., Hancock, P., Azevedo, R., ... Rabadi, G. (2025). *Integrating human-machine systems and digital twin technologies: Navigating trust, interoperability, and ethical challenges.* Retrieved from https://doi.org/10.2139/ssrn.5189676 doi: 10.2139/ssrn.5189676

Sargent, R. G. (2013). Verification and validation of simulation models. *Journal of Simulation*, *7*(1), 12–24. Retrieved from https://doi.org/10.1057/jos.2012.20 doi: 10.1057/jos.2012.20

Schreck, J., Matthews, G., Lin, J., Mondesire, S., Metcalf, D., Dickerson, K., & Grasso, J. (2025). Levels of automation for a computer-based procedure for simulated nuclear power plant operation: Impacts on workload and trust. *Safety*, *11*(1), 22. Retrieved from https://doi.org/10.3390/safety11010022 doi: 10.3390/safety11010022

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal policy optimization algorithms.* Retrieved from https://arxiv.org/abs/1707.06347 doi: 10.48550/arXiv.1707.06347

Sharp, M., & Weiss, B. A. (2018). Hierarchical modeling of a manufacturing work cell to promote contextualized PHM information across multiple levels. *Manufacturing Letters*, *15*, 46–49. Retrieved from https://doi.org/10.1016/j.mfglet.2018.02.003 doi: 10.1016/j.mfglet.2018.02.003

Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation: A review on

the statistical data driven approaches. *European Journal of Operational Research*, *213*(1), 1–14. Retrieved from https://doi.org/10.1016/j.ejor.2010.11.018 doi: 10.1016/j.ejor.2010.11.018

Singh, K., Selvanathan, B., Zope, K., Nistala, S. H., & Runkana, V. (2018). Concurrent estimation of remaining useful life for multiple faults in an ion etch mill: A data-driven approach. In *Annual conference of the phm society* (Vol. 10). Retrieved from https://doi.org/10.36001/phmconf.2018.v10i1.591 doi: 10.36001/phmconf.2018.v10i1.591

Soykan, B., Mondesire, S. C., & Rabadi, G. (2026a). Real-time dispatching in semiconductor manufacturing: A hybrid graph deep reinforcement learning and discrete-event simulation approach. In *Proceedings of the winter simulation conference (wsc).*

Soykan, B., Mondesire, S. C., & Rabadi, G. (2026b). A supply chain digital twin architecture for semiconductor industry. In *Optimizing supply chains through digital twins: Navigating the modern landscape.* Springer Nature.

Soykan, B., Mondesire, S. C., Rabadi, G., & Bochenek, G. (2025). Graph-enhanced deep reinforcement learning for multi-objective unrelated parallel machine scheduling. In *Proceedings of the winter simulation conference* (pp. 2515–2526). Retrieved from https://doi.org/10.1109/WSC68292.2025.11338986 doi: 10.1109/WSC68292.2025.11338986

Stöckermann, P., Feudel, S., Immordino, A., Hayen, N., Altenmüller, T., Gebser, M., ... Higgins, F. (2025). Reinforcement learning based dispatching solutions in semiconductor manufacturing: A literature review on validation and deployment. *Production and Manufacturing Research*, *13*(1). Retrieved from https://doi.org/10.1080/21693277.2025.2582472 doi: 10.1080/21693277.2025.2582472

Stöckermann, P., Immordino, A., Altenmüller, T., Seidel, G., Gebser, M., Tassel, P., ... Zhang, F. (2023). Dispatching in real frontend fabs with industrial grade discrete-event simulations by deep reinforcement learning with evolution strategies. In *Proceedings of the winter simulation conference* (pp. 3047–3058). Retrieved from https://doi.org/10.1109/WSC60868.2023.10408625 doi: 10.1109/WSC60868.2023.10408625

Stöckermann, P., Südfeld, H., Immordino, A., Altenmüller, T., Wegmann, M., Gebser, M., ... Zhang, F. F. (2025). Scalability of reinforcement learning methods for dispatching in semiconductor frontend fabs: A comparison of open-source models with real industry datasets. *The International Journal of Advanced Manufacturing Technology*, *139*, 4395–4415. Retrieved from https://doi.org/10.1007/s00170-025-16117-2 doi: 10.1007/s00170-025-16117-2

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press. Retrieved from http://incompleteideas.net/book/the-book-2nd.html

Tao, F., Xiao, B., Qi, Q., Cheng, J., & Ji, P. (2022). Digital twin modeling. *Journal of Manufacturing Systems*, *64*, 372–389. Retrieved from https://doi.org/10.1016/j.jmsy.2022.06.015 doi: 10.1016/j.jmsy.2022.06.015

Tassel, P., Gebser, M., & Schekotihin, K. (2021). *A reinforcement learning environment for job-shop scheduling.* Retrieved from https://arxiv.org/abs/2104.03760 doi: 10.48550/arXiv.2104.03760

Tassel, P., Kovács, B., Gebser, M., Schekotihin, K., Stöckermann, P., & Seidel, G. (2023). Semiconductor fab scheduling with self-supervised and reinforcement learning. In *Proceedings of the winter simulation conference* (pp. 1924–1935). Retrieved from https://doi.org/10.1109/WSC60868.2023.10407747 doi: 10.1109/WSC60868.2023.10407747

Towers, M., Kwiatkowski, A., Terry, J. K., Balis, J. U., De Cola, G., Deleu, T., ... Younis, O. G. (2024). *Gymnasium: A standard interface for reinforcement learning environments.* Retrieved from https://arxiv.org/abs/2407.17032 doi: 10.48550/arXiv.2407.17032

Wang, B., Zhou, H., Yang, G., Li, X., & Yang, H. (2022). Human digital twin driven human-cyber-physical systems: Key technologies and applications. *Chinese Journal of Mechanical Engineering*, *35*(11). Retrieved from https://doi.org/10.1186/s10033-022-00680-w doi: 10.1186/s10033-022-00680-w

Wright, T., Nsiye, E., & Mondesire, S. C. (2024). Generalized vs. individualized kaplan-meier time-to-failure predictive models for preventative maintenance in semiconductor manufacturing. In *International symposium on semiconductor manufacturing (issm).*

Wright, T., Soykan, B., & Mondesire, S. C. (2025). Real-time remaining useful life prediction with LSTM priority resampling. In *Ieee international conference on machine learning and applications (icmla).*

Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P. S., & Xu, C. (2020). Learning to dispatch for job shop scheduling via deep reinforcement learning. In *Advances in neural information processing systems.* Retrieved from https://proceedings.neurips.cc/paper/2020/hash/

11958dfee29b6709f48a9ba0387a2431
-Abstract.html

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, 106889. Retrieved from https://doi.org/10.1016/j.cie.2020.106889 doi: 10.1016/j.cie.2020.106889

BIOGRAPHIES

SEAN MONDESIRE is an Assistant Professor in the School of Modeling, Simulation, and Training at the University of Central Florida. His research interests include real-time decision-making through machine learning, digital twins, and simulation.

TORI WRIGHT is a Research Assistant in the High-performance Artificial Intelligence Laboratory (HAIL) within the School of Modeling, Simulation, and Training at the University of Central Florida. Her research interests include data analytics and machine learning for real-time systems and digital twins.

BULENT SOYKAN is an Associate Research Scientist at the University of Central Florida. His research interests include real-time decision-making under uncertainty, combinatorial optimization, digital twins, and reinforcement learning.