Act, Validate, Adapt: Closing the Causal Discovery-Control Loop

Taqiya Ehsan **Shuren Xia** **Jorge Ortiz**
Department of Electrical and Computer Engineering
Rutgers University, New Brunswick, NJ, USA
{taqiya.ehsan, shuren.xia, jorge.ortiz}@rutgers.edu

Abstract

Every control action on a physical system is already a do-operator intervention, yet causal discovery and control are typically treated as separate stages. This separation leaves deployed controllers unable to test whether the causal graph they rely on remains valid under changing operating conditions. We present PolicyGRID, a closed-loop framework that uses actuator interventions to validate candidate causal edges, fits a structural causal model over the validated graph, and reuses that graph for both policy optimization and latent-regime monitoring. In a multi-actuator physical simulation with latent operating regimes (building energy management), interventional graph validation reduces closed-loop energy use by 21% at the same comfort target, with satisfaction 1.4 points lower, relative to an otherwise identical observation-only graph, while improving four-way latent-context identification from 61.2% to 80.8%. A targeted ablation shows that the gains originate in graph pruning. Validation removes 16 of 47 candidate edges; with coefficient training held fixed, the validated graph reduces closed-loop energy relative to the unvalidated candidate set, while refitting coefficients on the same graph does not. On a live physical testbed, the same pipeline runs without code changes, performs 21 physical interventions, discovers 16 edges, and produces a target-responsive control policy where the observation-only baseline is insensitive to the comfort-energy target. These results show that physical controllers can use their own actions to validate causal structure and improve downstream control under latent regime shifts. Code: `https://github.com/taqiyaehsan/act-validate-adapt`.

1 Introduction

Causal discovery and control are usually studied separately. Discovery methods estimate structure from observational and interventional data [Hauser and Bühlmann, 2012, Toth et al., 2022, Annadani et al., 2024] but evaluate the inferred graph as an endpoint. Control methods use causal structure to choose actions [Lattimore et al., 2016, Zeng et al., 2025] but commonly assume the graph is given and stable. Recent methods bridge part of this separation [Cai et al., 2026, Caron et al., 2025], but neither physically validates discovered edges nor detects latent distributional shifts (Appendix Table 35).

In physical systems, this separation is poorly matched to deployment. A controller changes the data-generating process whenever it writes to an actuator. Those actions can also test causal hypotheses about the system. A graph learned from passive data may contain edges that look useful offline but become harmful when the controller acts on them. Under latent regime shifts (changes in unobserved operating context), prediction errors change meaning, and an unvalidated graph can amplify control errors through the controller's own actions. In building energy management, for example, an HVAC controller that learned a spurious edge between heating power and occupancy will misallocate energy whenever the occupancy pattern changes.

PolicyGRID connects causal discovery, graph validation, and deployed control in a closed-loop. Observational inference proposes candidate edges. Actuator interventions test whether those effects

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

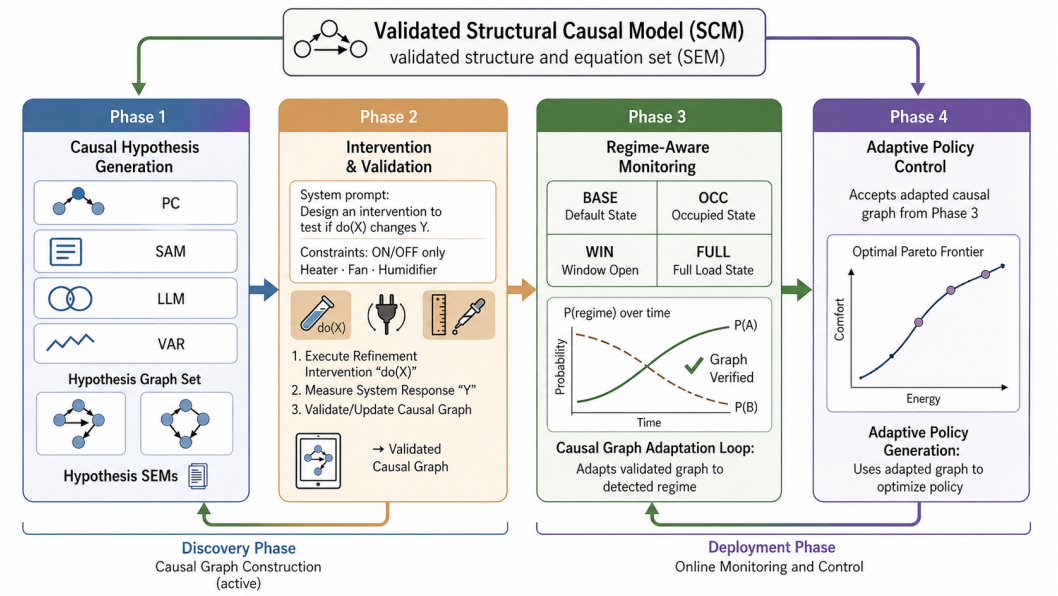


Figure 1: **The PolicyGRID closed loop** (system details in Section 4). Four hypothesis generators (PC, SAM, VARLiNGAM, LLM) propose candidate edges from observational data and context; candidates with actuator sources are tested by targeted *do*-operator interventions and the rest by an observational rescue test; the validated graph is fitted as a Ridge SCM that drives both policy optimization and regime monitoring.

appear under controlled changes to the system. Edges that pass validation are retained in a structural causal model (SCM). At deployment, this model is used to optimize the control policy and to monitor latent operating regimes through prediction error. Both deployment tasks use the same validated graph, but they contribute differently. The control gain comes from validation pruning false-positive edges from the candidate set. The monitor contributes detection and model-based attribution of regime changes: switching control coefficients on the inferred regime yields no measurable control benefit, and a supervised classifier identifies regimes more accurately than the monitor does (Section 4.3). We evaluate PolicyGRID on smart-building simulations, real building data, and a live physical testbed; transfer to robotics, industrial process control, and multi-zone commercial buildings is untested (Section 7).

Contributions.

1. **Interventional graph validation in the control loop.** The controller's own actuators test candidate edges by do-operator interventions, and the surviving graph drives deployed control. The validated candidate edge-set uses 21% less closed-loop energy with satisfaction 1.4 points lower (6.3–6.9 kWh less on each of three weather days, in all 15 seed-by-weather runs; Section 5.2), and a targeted ablation localizes the effect to candidate-set selection (Table 1).
2. **One validated graph for control and regime monitoring.** The same graph drives 4-way latent regime monitoring at 80.8% vs. 61.2% for the observation-only baseline (frozen consensus graph, 5 sim-replicate seeds; paired t=32.5, $p<10^{-5}$, 5/5 favor; Section 5.1). The monitor's role is detection and attribution, not control: regime-conditioned coefficient switching does not change closed-loop performance (Appendix E.10).
3. **Simulation-to-physical deployment.** On a live office testbed the same pipeline code discovers 16 edges, 10 confirmed by 21 physical interventions and 6 retained by the observational rescue test (F1=0.400 against an expert-proposed diagnostic ground truth, best graph utility of 11 baselines), and produces a control policy that responds to the comfort target; the testbed does not measure energy savings (Section 5.3).

2 Related Work

Causal discovery and regime change. Causal discovery has matured from constraint-based methods [Spirtes et al., 2000, Colombo et al., 2012] through score-based and continuous-optimization

approaches [Kalainathan et al., 2022, Zheng et al., 2020] to LLM-augmented frameworks [Kıcıman et al., 2023, Khatibi et al., 2024], all treating the data distribution as fixed. Active causal discovery selects interventions to resolve structural ambiguity [Hauser and Bühlmann, 2012, Toth et al., 2022, Brouillard et al., 2020, Zhang et al., 2023, Annadani et al., 2024]; DODO [Gregorini et al., 2025] is the closest structural parallel to our Phase 2 pipeline (edge t-test screening + Fisher z-pruning), extended here with multi-source hypothesis fusion, iterative discovery re-fit on intervention-augmented data, and physical deployment (DODO terminates at synthetic Erdős–Rényi recovery). Intervention-based verification [Choo et al., 2022] characterizes the interventions needed to confirm a proposed causal DAG; PolicyGRID instead tests individual candidate edges with the controller's own actuators and reuses the surviving graph for monitoring and control. A separate thread considers regime change [Zhao et al., 2025, Cheng et al., 2025, Rahmani and Frossard, 2025] on the identifiability foundation of Huang et al. [2020]. None physically validates discovered edges or uses the validated structure for multi-objective control.

Causal structure for control. Existing causal-control methods assume the graph is given [Peters et al., 2016, Lattimore et al., 2016, Buesing et al., 2019, Ding et al., 2022, Aglietti et al., 2020, Zeng et al., 2025]. Closest to our work, Cai et al. [2026] interleave structure learning and policy optimization but assume a single static structure, and C-MBPO [Caron et al., 2025] integrates causal learning into model-based RL without physical edge validation. We use an explicit, variable-level SCM rather than a latent-dynamics model [Ha and Schmidhuber, 2018, Hafner et al., 2020, Yao et al., 2024], trading representational generality for do-calculus reasoning [Ceriscioli and Mohan, 2025]. Only 23% of RL-for-HVAC studies deploy on real buildings [Kathirgamanathan et al., 2021, Al Sayed et al., 2024], and prior causal methods have appeared as offline analysis tools rather than closed-loop validators. To our knowledge, PolicyGRID is the only method combining interventional discovery, regime detection, edge validation, multi-objective control, and physical evaluation in a single closed loop. Thread-by-thread comparison and positioning matrix in Appendix K (Appendix Table 35).

3 Problem Formulation

3.1 Variables and Structural Causal Model

The state vector of a physical system, $\mathbf{V} = \mathbf{A} \cup \mathbf{S} \cup \mathbf{O} \cup \mathbf{X} \cup \mathbf{R}$ partitions by role: actions $\mathbf{A}$ targeted by the do-operator (HVAC, lighting), sensors $\mathbf{S}$ measured but not set externally (temperature, humidity), derived outcomes $\mathbf{O}$ (satisfaction, energy), exogenous variables $\mathbf{X}$ outside the control surface (outdoor weather), and latent context $\mathbf{R}$ unobserved at deployment whose changes induce distributional shifts (occupancy, window state). The partition is system-spec metadata, not learned; only edges with sources in $\mathbf{A}$ admit direct do-operator validation, while $\mathbf{X}$ and $\mathbf{R}$ sources require observational rescue (Section 4.1). This variable-level state abstraction is an assumption of the method: the system must expose named, semantically meaningful sensor and actuator variables. Cyber-physical deployments provide them through their sensor and actuator schemas; raw high-dimensional perception does not.

The structural causal model (SCM) has structural equations $V_i := f_i(\mathbf{Pa}(V_i), \epsilon_i)$ with exogenous noise ϵ_i and ground truth $\mathcal{G}^* = \{(V_j, V_i) : V_j \in \mathbf{Pa}(V_i)\}$ defined by simulator source code (or domain-expert proposal on the physical testbed; Appendix A.2); $\mathcal{G}^*$ is used only for diagnostic scoring and never enters decision-making. The discovered $(\hat{\mathcal{G}}, \boldsymbol{\beta})$ pair — the validated graph $\hat{\mathcal{G}}$ and its Ridge-fitted SCM parameter vector $\boldsymbol{\beta}$ (Section 4.2) — determines both policy gradients and regime prediction-error signals. We model at the variable level rather than as latent dynamics [Hafner et al., 2020, Ha and Schmidhuber, 2018]: each edge names a physical relationship, enabling do-calculus [Pearl, 2009] and cross-checking at deployment. $\hat{\mathcal{G}}$ may contain cycles (heater warms room, thermostat throttles heater); equilibrium semantics for cyclic SCMs [Bongers et al., 2021] accommodate this, with the linear $V^* = (I - B)^{-1}\epsilon$ [Haavelmo, 1944] as canonical illustration. The deployed SCM is quadratic (Section 4.2); policy optimization handles cycles via chained per-equation partials with single-pass recursion (Section 4.2).

3.2 The Closed-Loop Problem

PolicyGRID couples two normally separated problems. The first is causal structure recovery under a finite intervention budget B: choose which action to intervene on, at what value, in what order, so $\hat{\mathcal{G}}$ approaches $\mathcal{G}^*$ under SHD and directed-edge F1. The second is multi-objective control on $\hat{\mathcal{G}}$: with $\varepsilon \in [0, \varepsilon_{\max}]$ selecting a point on the satisfaction–energy frontier via $w_s(\varepsilon) = \max(w_{\min}, 1 - \alpha\varepsilon)$ and $w_e = 1 - w_s$ ($\alpha = 0.35$, $w_{\min} = 0.3$, $\varepsilon_{\max} = 2.0$; sensitivity in Appendix F), pick $\mathbf{a}^* =$

$\arg\max_{\mathbf{a}\in[0,1]^{|\mathbf{A}|}} w_s(\varepsilon)\,\mathrm{Sat}(\mathbf{a};\hat{\mathcal{G}},\boldsymbol{\beta}) + w_e(\varepsilon)(1-\mathrm{Eng}(\mathbf{a};\hat{\mathcal{G}},\boldsymbol{\beta}))$ (1). Eq. 1 is solved online at each step with current state (incl. exogenous $\mathbf{X}$), so $\mathbf{a}^*$ tracks ambient conditions, not a static setpoint. The coupling is the central claim. The do-operator interventions validating $\hat{\mathcal{G}}$ at discovery target the *same action variables* the policy optimizer later writes to and the regime monitor reads from.

3.3 Effect of Interventional Validation

Interventional validation affects the learned SCM through coefficients, topology, and regime discrimination. Full statements and proofs are in Appendix L. For a single outcome

$$Y = \beta\, A + \gamma\, C + \epsilon, \qquad \mathbb{E}[\epsilon \mid A, C] = 0,$$

where A is an action, C is an unobserved context, and β, γ are the causal coefficients of A and C on Y. With C omitted, the OLS coefficient is

$$\beta_{\mathrm{obs}} = \beta + \gamma\, \frac{\mathrm{Cov}(A, C)}{\mathrm{Var}(A)}.$$

Randomized intervention on A removes the covariance term. In our deployed Ridge structural equation model (SEM), this coefficient-level benefit is absorbed by shrinkage regularization. Section 5.2 (Row A) shows that the main empirical gain comes from candidate-edge pruning.

Pruning matters because graph errors contribute to SCM error. Let F_ε and $\hat{F}_\varepsilon$ be the true and SCM-predicted scalarized objectives over a compact action set. If

$$\sup_{\mathbf{a}} |F_\varepsilon(\mathbf{a}) - \hat{F}_\varepsilon(\mathbf{a})| \le \delta,$$

then an η-approximate maximizer of $\hat{F}_\varepsilon$ has regret at most $2\delta + \eta$. This bound ties graph topology to downstream control through the uniform error of the SCM objective.

The same SCM also determines regime discrimination. The Bayesian monitor (Eqs. 8–9) accumulates log-odds between regime-specific models at a rate proportional to their prediction-error difference. Reducing the correct-regime prediction error relative to alternatives improves discrimination. Section 5.1 evaluates this effect empirically.

4 The PolicyGRID Framework

PolicyGRID is a closed loop (Algorithm 1, Figure 1). An ensemble of discovery methods proposes candidate edges from observational data and context; each candidate with an actuator source is tested by a targeted do-operator intervention that checks whether the predicted effect appears, and candidates from exogenous or latent sources, which cannot be intervened on, are retained only if they pass an observational rescue test (Section 4.1). Retained edges populate $\hat{\mathcal{G}}$; failed edges are dropped permanently. A Ridge SCM is fit on $\hat{\mathcal{G}}$, producing $\boldsymbol{\beta}$. The same $\hat{\mathcal{G}}$ drives two downstream tasks: a gradient-based policy optimizer, and a regime monitor using regime-specific coefficients $(\hat{\mathcal{G}}, \boldsymbol{\beta}^{(r)})$ fitted on partitioned training data. The optimizer runs in two configurations. The stationary open-loop sweep uses one global $(\hat{\mathcal{G}}, \boldsymbol{\beta})$. Closed-loop deployment keeps the monitor in the loop: at each step the most probable regime selects which coefficient set $\boldsymbol{\beta}^{(r)}$ the optimizer uses. Sharing $\hat{\mathcal{G}}$ across tasks ensures that every actuator-sourced edge influencing control or shift detection has passed a do-operator test, and that every other edge has passed the rescue test.

4.1 Hypothesis Generation and Validation

Four discovery methods run in parallel on the observational data: PC [Spirtes et al., 2000] (constraint-based), SAM [Kalainathan et al., 2022] (GAN-structural), VARLiNGAM [Hyvärinen et al., 2010] (VAR + LiNGAM residuals), and an LLM (GPT-3.5-turbo) prompted with variable names and building-science context, no data access. This hypothesis generation stage extends GRID [Ehsan et al., 2025], which merged PC, SAM, and LLM proposals and accepted an edge when subsequent interventions moved the target past a fixed threshold; PolicyGRID adds VARLiNGAM, updates that acceptance rule with the composite test below, and reuses the validated graph for control and monitoring. The union $\mathcal{G}_\cup$ is ranked by inter-method agreement $C_{ij} = |\{m : m \text{ proposes } i \to j\}|/|\mathcal{M}|$ (3), with $\mathcal{M}$ the set of four methods. The budget is spent disagreement-first: lowest-C_{ij} edges first, since interventional testing is most informative where methods disagree [Eberhardt and Scheines, 2007].

Algorithm 1 PolicyGRID closed loop. DISCOVER = ensemble of PC, SAM, VARLiNGAM, LLM (Section 4.1); TEST = relevance test (Section 4.1); PROBE = LLM-proposed diagnostic do-operation (Section 4.3).

Require: obs. data $\mathcal{D}$, methods $\mathcal{M}$, intervention budget B, comfort target ε
Ensure: validated graph $\hat{\mathcal{G}}$, SCM coefficients $\boldsymbol{\beta}$, policy $\mathbf{a}^*$

$\mathcal{G}_\cup \leftarrow \bigcup_m \text{DISCOVER}_m(\mathcal{D})$; rank by inter-method disagreement (Eq. 3)
for $i = 1, \ldots, I_{\max}$ **do**
    Test untested edges; update $\hat{\mathcal{G}}$ (validated $\rightsquigarrow$ in, failed out); $\mathcal{D} \mathrel{+}= \mathcal{D}_{\text{int}}$; re-run $\mathcal{M} \setminus \{\text{LLM}\}$ on augmented data
end for
Rescue non-intervenable edges (Eq. 4); prune indirect effects (Eq. 5); fit $\boldsymbol{\beta} \leftarrow \text{Ridge}(\hat{\mathcal{G}}, \mathcal{D})$ with direction constraints (Eq. 7) and $\boldsymbol{\beta}^{(r)}$ on each regime slice of $\mathcal{D}$
$\mathbf{a}^* \leftarrow \arg\max_{\mathbf{a}} w_s(\varepsilon)\,\text{Sat}(\mathbf{a}) + w_e(\varepsilon)\,(1 - \text{Eng}(\mathbf{a}))$
while deployed **do**
    Observe $\mathbf{y}_t$; update joint posteriors $w_t^{(r)}$ (Eqs. 8–10)
    if $\max_r w_t^{(r)} > \tau_{\text{conf}}$ $(=0.7)$ for ≥ 120 steps (Section 4.3) **then** PROBE; refold posterior.
    end if
    Apply $\mathbf{a}^*$ re-optimized under $\boldsymbol{\beta}^{(\hat{r}_t)}$, $\hat{r}_t = \arg\max_r w_t^{(r)}$ (closed loop; the open-loop sweep applies the global $\mathbf{a}^*$).
end while

Interventional testing. Each candidate $i \to j$ with action source $i \in \mathbf{A}$ receives $K{=}10$ trials: a 3-trial screen at $\text{do}(V_i{=}v_{\max})$ that rejects effects below the sensor noise floor, then 7 confirmatory trials mixing deterministic, paired-counterfactual ($\text{do}(v_{\max})$ vs. $\text{do}(0)$ from the same state), and LLM-designed intermediate-value interventions (Appendix B). Each trial yields a mean effect Δ_{ij} over $H{=}30$ steps. An edge is validated if a two-sided t-test on the K values of Δ_{ij} rejects zero mean at $\alpha_{\text{test}} = 0.05$, $|\bar{\Delta}_{ij}| > \tau_{\text{eff}} = 0.1$ in normalized units, and all K values share a sign; Benjamini–Hochberg FDR at $\alpha_{\text{FDR}} = 0.10$ [Benjamini and Hochberg, 1995] applies when $\geq$20 candidates are tested in a round. The sign condition binds: it bounds per-edge false validation by $2^{-(K-1)} \approx 0.2\%$, so the implementation's secondary acceptance at $0.05 \leq p < 0.10$ changes nothing, and the test reaches about 80% power at a per-trial standardized effect of $d \approx 2$ (Appendix B.4). The pooled t-test assumes that an edge's K effects share one distribution, so stationarity is required across each edge's trials, not globally. Discovery is then re-run on $\mathcal{D} \cup \mathcal{D}_{\text{int}}$ for fresh candidates, for $I_{\max} = 10$ rounds [Hauser and Bühlmann, 2012].

The t-test measures the *total* effect of i on j; we write $i \rightsquigarrow j$ for a validated relevance and $i \to j$ for a direct edge that survives pruning.

Non-intervenable rescue. Edges from exogenous ($\mathbf{X}$) or latent ($\mathbf{R}$) sources cannot be tested by intervention. We retain such candidates when two observational signals agree: a weighted majority of discovery methods proposes the edge, $\sum_{m: i \to j \in \mathcal{G}_m} w_m \geq \frac{1}{2} \sum_m w_m$ (uniform w_m by default), *and* the back-door-adjusted partial correlation $|r_{ij \cdot \mathbf{Z}}| > \tau_r = 0.5$ on the adjustment set $\mathbf{Z} = \phi(t) \cup \text{NonDesc}(j)$, where $\phi(t)$ are time-of-day sin/cos features and $\text{NonDesc}(j)$ the non-descendants of j [Pearl, 2009] (4). Rescued candidates enter $\hat{\mathcal{G}}$ as relevances $i \rightsquigarrow j$ and are pruned uniformly with the validated set.

Indirect-effect pruning. An effect that flows entirely through a mediator k still passes the t-test, and keeping both $i \to k \to j$ and $i \to j$ would double-count one pathway in the policy gradient and the monitor. We prune $i \to j$ if a Fisher z-test finds $i \perp\!\!\!\perp j \mid \mathbf{Pa}(j) \setminus \{i\}$ at $\alpha_{\text{prune}} = 0.01$ (5), so every surviving edge is a direct mechanism.

4.2 SCM Fitting and Policy Optimization

Each variable V_j with parents $\mathbf{Pa}(V_j)$ is modelled by a quadratic SEM with pairwise interaction terms,

$$V_j = \alpha_j + \sum_{p \in \mathbf{Pa}(V_j)} \left(\beta_{jp}\, p + \gamma_{jp}\, p^2\right) + \sum_{p<q \in \mathbf{Pa}(V_j)} \delta_{j,pq}\, p\, q + \epsilon_j, \tag{6}$$

with linear (β), quadratic (γ), and interaction (δ) coefficients; the second-order terms capture saturation and multiplicative actuator coupling while keeping closed-form gradients.

Coefficients are estimated by Ridge regression ($\alpha_{\text{ridge}} = 0.5$, 5-fold CV over $\{0.1, 0.5, 1.0, 5.0\}$): the polynomial features are correlated by construction, and OLS can inflate coefficients through suppressor effects [Bühlmann et al., 2014] enough to invert the policy gradient (Appendix I). When a Ridge linear term contradicts validated interventional evidence ($\text{sign}(\hat{\beta}_{jp}) \neq \text{sign}(\Delta_{p \to j})$ for $p \in \mathbf{A}$), we override it with a sign-correct floor $\hat{\beta}_{jp} \leftarrow \text{sign}(\Delta_{p \to j}) \cdot \beta_{\min}$, $\beta_{\min} = 10^{-3}$ (7).

Policy optimization runs gradient descent on Eq. 1, with $\partial\text{Sat}/\partial\mathbf{a}$ and $\partial\text{Eng}/\partial\mathbf{a}$ from per-equation partials of Eq. 6. Direct actuator-to-outcome edges keep the gradient well-defined on cyclic graphs. Optimization starts from a fixed midpoint, so $\mathbf{a}^*$ is a deterministic function of $(\hat{\mathcal{G}}, \boldsymbol{\beta}, \varepsilon)$ and the current state; in closed-loop deployment, $\boldsymbol{\beta}$ is the set $\boldsymbol{\beta}^{(\hat{r}_t)}$ selected by the monitor (Section 4.3).

4.3 Regime Monitoring

The monitor identifies which combination of latent variables in $\mathbf{R}$ is active without observing them. We treat the $|\mathcal{R}| = 2^{|\mathbf{R}|}$ joint settings (e.g., binary occupancy $\times$ binary window) as a finite hypothesis space and maintain a Bayesian posterior $w_t^{(r)}$. The graph $\hat{\mathcal{G}}$ is fixed across regimes; only $\boldsymbol{\beta}^{(r)}$ changes, obtained by refitting Ridge on the regime-r slice of training data, so the same mechanisms are captured at regime-specific strengths.

At each step the monitor predicts $\hat{\mathbf{y}}_t^{(r)}$ under each regime and compares to the observed $\mathbf{y}_t$. The likelihood decays exponentially with L_1 prediction error, $\ell_t^{(r)} = \exp(-\sigma \|\mathbf{y}_t - \hat{\mathbf{y}}_t^{(r)}\|_1)$ (8), where σ controls discrimination sharpness. The posterior updates multiplicatively, $w_t^{(r)} \propto \ell_t^{(r)} w_{t-1}^{(r)}$ (9), then a forgetting step pulls toward uniform at rate η, $w_t^{(r)} \leftarrow (1-\eta) w_t^{(r)} + \eta/|\mathcal{R}|$ (10). Forgetting prevents accumulated evidence from locking the posterior after a regime shift. The monitor never observes the true label R_t; inputs are only sensor measurements and SCM regime predictions.

When the posterior has settled ($\max_r w_t^{(r)} > 0.7$ for ≥ 120 steps), the system issues an active probe: the LLM proposes a brief diagnostic actuator pattern, each regime model predicts the response, and the best-matching regime's weight is boosted (Appendix B). The LLM only proposes; the comparison is deterministic from $\hat{\mathcal{G}}$ and the regime-specific coefficients.

Role in the loop. In closed-loop deployment the posterior selects the coefficient set the optimizer uses, but this switching is not the source of the control gain. Under common random numbers, static, regime-aware, and true-regime-oracle policies on the same validated graph are indistinguishable in closed-loop performance (Appendix E.10); the gain comes from the validated candidate set (Section 5.2). The monitor contributes detection and model-based attribution: each regime model predicts every sensor, so per-variable prediction errors show which mechanisms changed, which a regime label alone does not. The monitor uses labeled regime partitions at commissioning, as a supervised classifier would, and for pure regime classification a feature-matched classifier is more accurate (logistic regression 90.0% vs. 67.1% for the monitor; Appendix E.10).

5 Experiments

The experiments evaluate three claims. (i) Interventional validation produces a graph supporting more accurate latent-regime detection than observational discovery alone. (ii) The same validated graph, fed to the same gradient-based optimizer, produces measurably more efficient closed-loop control. (iii) The pipeline transfers from simulation to real hardware without code changes.

Discovery runs on six environments: five EnergyPlus-calibrated [Crawley et al., 2001] JavaScript simulators (5–15 variables, 0–2 latent regimes) and the real-world ASHRAE Great Energy Predictor III dataset. Shift detection and closed-loop control run on `smart_building_rich`, the only environment with a sufficiently rich observable surface for latent-regime monitoring (15 variables, two continuous actuators (HVAC 5 kW, Lighting 0.5 kW), a 29-edge GT graph, and two latent regimes (occupancy 8-level, window binary)). The control problem is the ε-constrained satisfaction–energy tradeoff of Eq. 1, evaluated at $\varepsilon \in \{0.5, 0.8, 1.0, 1.5, 2.0\}$ across 1440-step episodes in cold/warm/mild weather. Every method receives identical training data, simulator access, and five seeds,

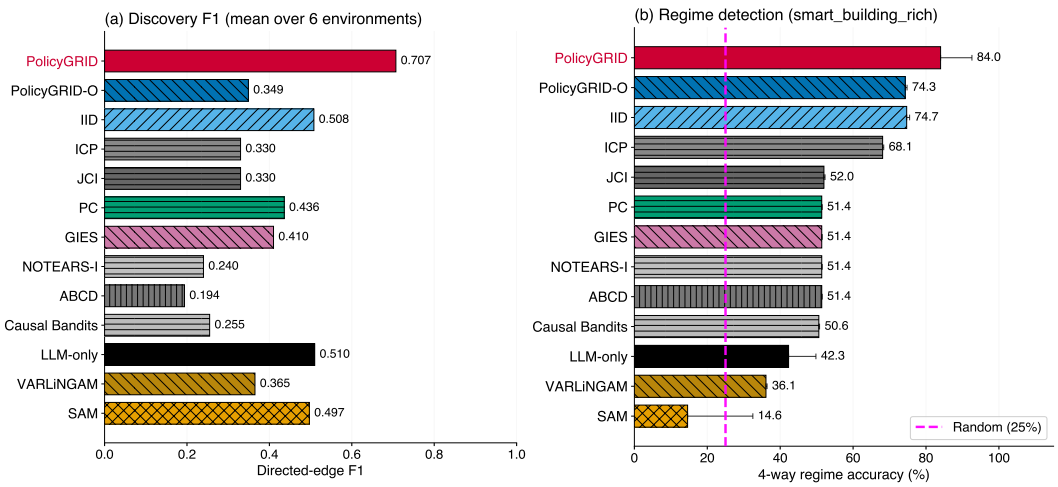


Figure 2: **Discovery and regime detection summary** (PolicyGRID, PolicyGRID-O, and the 11 baseline methods). (a) Directed-edge F1 averaged across all 6 environments (5 seeds): PolicyGRID leads at 0.707. (b) 4-way joint regime classification accuracy on `smart_building_rich` (per-discovery-seed, 5 graphs/method): PolicyGRID 84.0% vs. PolicyGRID-O 74.3% vs. IID 74.7%; most baselines stuck near the 50–52% chance floor. *Validation gives up a small per-environment F1 to win decisively on the downstream task that matters, with a 9.7 pp lead over the same pipeline running on the PolicyGRID-O candidate set.*

yielding 15 paired seed-by-weather conditions for closed-loop comparisons (each method uses one fixed graph across these runs, and the seed varies only simulator noise, so the 15 runs show consistency rather than independent replication); C-MBPO [Caron et al., 2025] additionally receives PolicyGRID's validated graph, isolating whether the difference traces to the graph or the optimizer. Full specs in Appendix Table 2; protocols in Appendix B.

5.1 Graph quality and shift detection

The discovery comparator field comprises 11 baseline methods (five observational and six interventional) plus the observation-only variant PolicyGRID-O — the unvalidated union of PC + SAM + VARLiNGAM + LLM, identical to PolicyGRID in every step except the interventional validation. Citations and implementations in Appendix C. PolicyGRID achieves the highest discovery F1 on four of six environments.

On `smart_building_rich`, two methods exceed PolicyGRID's F1 (LLM-only 0.441, IID 0.433 vs. 0.376) but fall behind on the regime detection the graph supports (42.3% and 74.7% vs. 84.0%, each method on its own discovered graphs; Appendix D.2). F1 weights inclusion and exclusion errors symmetrically, but a retained false positive becomes a coefficient the policy gradient actively exploits, while a missed true positive merely fails to contribute. Validation removes 16 of 47 PolicyGRID-O edges (12 GT false positives, 4 sub-threshold true positives). Results under sensor noise and a latent confounder are in Appendix Table 9.

The most direct structural-sharpness test is latent regime detection: all 13 methods drive the same Bayesian monitoring pipeline (Section 4.3) over five sim-replicate seeds with frozen edge sets, varying only the input graph, and accuracy is the 4-way joint classification over (Occupancy, Window). PolicyGRID reaches $80.8 \pm 1.1\%$ against PolicyGRID-O's $61.2 \pm 0.5\%$ (paired $+19.5$ pp, $5/5$ seeds) and IID's $57.9 \pm 1.1\%$. The advantage concentrates in window detection (81.4% vs. 62.7%) through thermal-coupling edges that interventional testing tightens; occupancy saturates at 94–99% for every non-trivial graph. A per-discovery-seed evaluation gives $84.0 \pm 8.6\%$ (Appendix D.2).

5.2 Closed-loop control

The closed-loop evaluation adds seven control baselines (Appendix C) and four candidate-set conditions with the optimizer fixed: PolicyGRID's 31-edge graph, PolicyGRID-O's 47-edge union, a count-matched random graph, and an empty graph. Under stationary open-loop evaluation, Policy-

Table 1: **Graph topology vs. coefficient values** (`smart_building_rich`, closed-loop @ ε=1.0, 15 paired conditions, mean $\pm$ std). Same optimizer/monitor/regime-switching across rows; rows differ only in (graph, training data). A vs. B isolates the *coefficient* effect (same graph, augmented training data); C vs. B isolates the *graph* effect (different graph, same training data). Lower kWh is better; Sat is 1.4 points lower for B than for C. All 15 conditions share one graph per row and the seeds vary only simulator noise, so the 15/15 counts show consistency across weather days and seeds, not independent replication. *0/15 paired conditions favor intervention-augmented coefficient training, while 15/15 favor PolicyGRID. Candidate-set selection accounts for the closed-loop performance difference.*

	Graph	**Ridge SEM training data**	**kWh**	**Sat**
A	PolicyGRID (31 edges)	obs. + do-operator probes	26.19±0.14	82.2
B	PolicyGRID (31 edges)	observational only	24.56±0.16	81.9
C	PolicyGRID-O (47 edges)	observational only	31.07±0.18	83.3
Paired Δ, A $-$ B (*coefficient effect*, same graph)		15/15 favor B	$+1.63 \pm 0.04$	$+0.3$
Paired Δ, C $-$ B (*graph effect*, same training data)		15/15 favor B	$+6.51 \pm 0.32$	$+1.4$

GRID and PolicyGRID-O are nearly indistinguishable (Appendix Tables 15 and 17); closed-loop deployment tells a different story.

With latent regime shifts active, PolicyGRID uses 24.56 kWh against 31.07 kWh for PolicyGRID-O at $\varepsilon = 1.0$, a paired 6.51 kWh (21%) difference favouring PolicyGRID in all 15 seed-by-weather conditions (6.3–6.9 kWh on each weather day), with satisfaction 1.4 points lower (81.9 vs. 83.3; Table 1). The gap narrows to 7% at $\varepsilon = 0.5$ and closes at $\varepsilon = 2.0$, where energy reaches its floor; control baselines without causal structure stay at 60–66 kWh regardless of ε (Appendix Table 14). Stationary metrics under-report the deployment cost of unvalidated edges.

Pareto coverage separates graph from optimizer: PolicyGRID alone navigates the energy–satisfaction frontier above 79% satisfaction (hypervolume 0.772; Appendix Fig. 8), while C-MBPO, given the same validated graph with a different optimizer, stays at one operating point (HV 0.513). Table 1 locates the gain: with training data fixed, replacing the 47-edge candidate set with the validated graph (Row C vs. B) accounts for the full 6.51 kWh, while refitting coefficients on intervention-augmented data with the graph fixed (Row A vs. B) degrades energy by 1.63 kWh ($0/15$ favour augmentation; Section 6).

What drives the gain. Further ablations support this reading. Regime-conditioned coefficient switching gives no detectable control benefit, even with the true regime (Appendix E.10). Every hypothesis generator contributes ground-truth candidates that no other proposes, and removing any one lowers discovery F1 on the primary environment (Appendix E.9). The energy reduction from validation replicates on two further single-zone simulators in 49 of 49 paired cells (Appendix E.11).

5.3 Physical deployment

On a live office testbed (Appendix A.2), the same pipeline code runs end-to-end, with the validation gate adapted to hardware time cost: 7 trials per actuator instead of 10, $\alpha_{\text{test}} = 0.10$, and no sign-consistency stage (Appendix B.4). Discovery performs 21 do-operator interventions over about 15.5 hours, about 9 of them actuator holds, and yields 16 edges: 10 confirmed by intervention and 6 retained by the observational rescue test (F1$= 0.400$ against an expert-proposed diagnostic graph; Appendix D.1). As in simulation, the LLM-only prior leads on raw F1 but trails on monitoring. On a separate 9-hour episode with 7 induced regime transitions, PolicyGRID leads in-distribution (82% occupancy, 91% window; Fig. 3); when regime training data are heavily imbalanced, SAM leads instead (Appendix D.2). The validated SCM yields a policy that responds to the comfort target, while PolicyGRID-O's gradient stays flat (Appendix D.3). The testbed shows that the pipeline runs end-to-end on hardware and that its policy responds to the target; it does not measure energy savings. This deployment lies inside the boundary the method currently supports: few actuators, reversible and bounded actions, effects observable within one washout window, and no safety-critical exploration. Validating actions are deliberate experiments at maximum actuator settings within equipment ratings; no safety or cost term enters the intervention ordering, and safety-aware ordering is future work.

Robustness checks (Appendices E and F) show that disagreement-first ordering beats random ordering at a fixed intervention budget (36% relative F1 gain), the monitor re-identifies the regime after

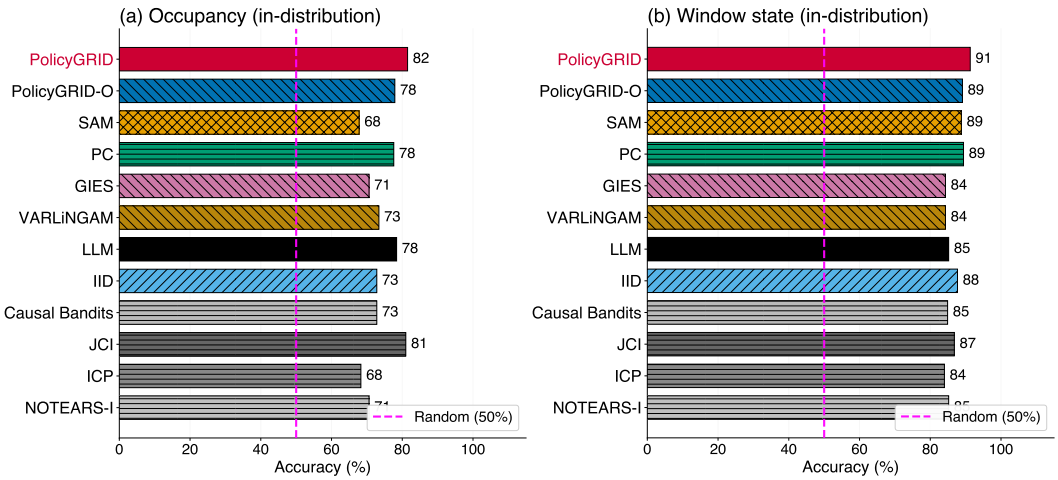


Figure 3: **Physical testbed monitoring** (PolicyGRID, PolicyGRID-O, and the 11 baseline methods evaluated on the physical testbed; in-distribution per-dimension accuracy on a 9-hour episode with interleaved temporal-block train/test split). All methods use the same Bayesian monitoring pipeline (Section 4.3); only the discovered edge set differs. OOD evaluation in Appendix Fig. 7. *PolicyGRID achieves the highest occupancy accuracy (82%) and window accuracy (91%) on real hardware. The simulation pattern survives sim-to-real transfer.*

an induced shift, all 5 deliberately false LLM edges are rejected before reaching the policy, and monitoring and discovery vary by under 5 pp across the swept hyperparameters.

6 Discussion

The classical do-operator premise [Pearl, 1995] is that interventions decorrelate the action from latent context and remove the omitted-context bias described in Section 3.3. Row A in Table 1 contradicts this empirically. With PolicyGRID's graph fixed, refitting Ridge on intervention-augmented data *degrades* closed-loop kWh by 1.63 ($0/15$ runs favouring augmentation). The reconciliation is regularization. Ridge shrinkage absorbs the bias the do-operator would correct (Appendix E.2). Candidate-set selection accounts for the performance difference. False-positive edges produce SCM error that Ridge cannot recover, and the regret bound (Section 3.3) ties this directly to policy cost. F1 treats inclusion and exclusion errors symmetrically, which misranks methods downstream. A retained false-positive becomes a coefficient the policy actively writes against, while a missed true-positive merely fails to contribute.

This has a methodological consequence. The 21% difference is invisible under stationary open-loop evaluation (Appendix Table 15) and surfaces only when the controller must respond to latent shifts. A discovery method evaluated on its training distribution alone looks stronger than it should, since the deployment cost of unvalidated edges accumulates through the controller's own actions in regimes the training data did not represent.

7 Limitations

The current implementation tests $O(|\mathbf{A}|\cdot|\mathbf{V}|)$ candidate edges per seed, each with at most K trials (1,576 interventions in a measured seed of the primary environment; Appendix B.4), which limits deployment to systems with roughly fifty observable variables. Larger systems require hierarchical or modular discovery. The quadratic SEM (Eq. 6) captures dominant second-order couplings but does not model higher-order mechanisms. The regime monitor detects coefficient shifts within a fixed topology, so newly appearing causal pathways require re-running discovery. Its posterior compares known regimes against each other: a pathway absent from $\hat{\mathcal{G}}$ raises the error of every regime model but can still leave high confidence on the least-bad one. The system has no none-of-the-above detector or automatic re-discovery trigger, and the regime-shift experiment covers transitions among known regimes only; absolute-residual thresholds are the natural extension.

The evidence is replication within one problem family: six building environments, a live office testbed, and two further single-zone simulators for closed-loop control (Appendix E.11). Transfer to high-dimensional robotics, industrial process control, and multi-zone commercial buildings is

untested; those domains require safety screening of actions, faster dynamics, and high-dimensional state, which the method does not yet handle. Validating actions are not screened for safety or cost.

The LLM (GPT-3.5-turbo) is used as an advisory component. It proposes candidate edges, designs intermediate-value interventions, and suggests diagnostic probes at deployment. Every candidate passes the same interventional validation procedure, and probe classification is computed from the SCM. A passive-only fallback matches full accuracy at steady state (Appendix D.2), and the wrong-prior experiment rejects all five injected false edges (Appendix E). Broader adversarial stress-testing of the LLM channel remains open.

Appendix B.4 characterizes the per-edge test's error rates under a Gaussian working model; a distribution-free sample-complexity bound relating per-edge trials to residual SHD remains open. The physical testbed F1 is computed against a 14-edge diagnostic graph derived from building physics rather than a system-spec graph. Under heavily imbalanced out-of-distribution training (81% base regime), SAM leads on monitoring; PolicyGRID's advantage returns under an in-distribution split. This identifies regime imbalance as a practical data-collection constraint for deployment.

8 Conclusion

Every control action on a physical system is already a do-operator intervention. PolicyGRID connects this observation to deployed control. The graph these interventions validate at discovery is the graph the policy optimizer and the regime monitor use at deployment. On the smart-building simulation, the validated graph cuts closed-loop energy by 21% over its observation-only counterpart, with satisfaction 1.4 points lower, and identifies four latent contexts at 80.8% vs. 61.2%. Regime-conditioned coefficient switching adds no control benefit; the monitor's value is detection and attribution. The targeted ablation in Table 1 attributes the performance difference to candidate-set selection, and the same pipeline transfers to a live physical testbed without code changes. The broader lesson is that closed-loop controllers whose actions are do-operators can validate and prune their causal models using existing actuator commands, and that doing so changes which graph performs best downstream regardless of how it ranks on stationary structural metrics.

Acknowledgments and Disclosure of Funding

Emma Zafrir helped with collecting data and running experiments in the live testbed. This work was supported by the National Science Foundation (NSF) and Center for Smart Streetscapes (CS3) under NSF Cooperative Agreement No. EEC-2133516. The authors declare no competing interests.

References

Virginia Aglietti, Xiaoyu Lu, Andrei Paleyes, and Javier González. Causal Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3155–3164. PMLR, 2020.

Khalil Al Sayed, Abhinandana Boodi, Roozbeh Sadeghian Broujeny, and Karim Beddiar. Reinforcement learning for hvac control in intelligent buildings: A technical and conceptual review. *Journal of Building Engineering*, 95:110085, 2024.

Yashas Annadani, Panagiotis Tigas, Stefan Bauer, and Adam Foster. Amortized active causal induction with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 37:44216–44239, 2024.

ASHRAE. Great energy predictor III: Kaggle competition dataset. `https://www.kaggle.com/c/ashrae-energy-prediction`, 2019. Building energy consumption data from 1449 buildings.

ASHRAE. Standard 55-2023: Thermal environmental conditions for human occupancy. Technical report, American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2023.

Karl Johan Åström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America, Research Triangle Park, NC, 2nd edition, 1995. Canonical reference for proportional-integral-derivative control, including Ziegler–Nichols-style tuning.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Stephan Bongers, Patrick Forré, Jonas Peters, and Joris M. Mooij. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 49(5):2885–2915, 2021. doi: 10.1214/21-AOS2064.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. In *Advances in Neural Information Processing Systems*, volume 33, pages 21865–21877, 2020.

Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=BJG0voC9YQ`.

Peter Bühlmann, Jonas Peters, and Jan Ernest. CAM: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014. doi: 10.1214/14-AOS1260. Explicitly two-stage: linear models for causal ordering, spline regression for nonlinear mechanisms. Proves causal order can be consistently estimated even if Stage 1 is misspecified (linear when truth is nonlinear).

Ruichu Cai, Siyang Huang, Jie Qiao, Wei Chen, Yan Zeng, Keli Zhang, Fuchun Sun, Yang Yu, and Zhifeng Hao. Learning by doing: an online causal reinforcement learning framework with causal-aware policy. *Science China Information Sciences*, 69(2):122104, 2026.

Alberto Caron, Vasilios Mavroudis, and Chris Hicks. Towards causal model-based policy optimization. *arXiv preprint arXiv:2503.09719*, 2025.

Matteo Ceriscioli and Karthika Mohan. Agents robust to distribution shifts learn causal world models even under mediation. In *Advances in Neural Information Processing Systems*, 2025.

Xia Chen, Jimmy Abualdenien, Manav Mahan Singh, André Borrmann, and Philipp Geyer. Introducing causal inference in the energy-efficient building design process. *Energy and Buildings*, 277: 112583, 2022. doi: 10.1016/j.enbuild.2022.112583.

Yue Cheng, Bochen Lyu, Weiwei Xing, and Zhanxing Zhu. Dycast: Learning dynamic causal structure from time series. In *The Thirteenth International Conference on Learning Representations*, 2025.

Davin Choo, Kirankumar Shiragur, and Arnab Bhattacharyya. Verification and search algorithms for causal DAGs. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. PETS: bootstrap ensemble + TS-infinity uncertainty propagation + CEM. DDPC_PETS baseline.

Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control: In the shallows of the DeePC. In *2019 18th European Control Conference (ECC)*, pages 307–312, 2019. doi: 10.23919/ECC.2019.8796306. DeePC / Willems' Fundamental Lemma basis for DDPC_Behavioral.

Drury B Crawley, Linda K Lawrie, Frederick C Winkelmann, Walter F Buhl, Y Joe Huang, Curtis O Pedersen, Richard K Strand, Richard J Liesen, Daniel E Fisher, Michael J Witte, et al. Energyplus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4): 319–331, 2001.

Wenhao Ding, Haohong Lin, Bo Li, and Ding Zhao. Generalizing goal-conditioned reinforcement learning with variational causal reasoning. *Advances in Neural Information Processing Systems*, 35:26532–26548, 2022.

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007. doi: 10.1086/525638.

Taqiya Ehsan, Shuren Xia, and Jorge Ortiz. GRID: Graph-based reasoning for intervention and discovery in built environments. *arXiv preprint arXiv:2509.16397*, 2025. URL `https://arxiv.org/abs/2509.16397`.

Povl Ole Fanger. *Thermal Comfort*. Danish Technical Press, 1970. PMV/PPD model. Fundamental physical justification for quadratic comfort response.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:524, 2019.

Matteo Gregorini, Chiara Boldrini, and Lorenzo Valerio. DODO: Causal structure learning with budgeted interventions. *arXiv preprint arXiv:2510.08207*, 2025. URL `https://arxiv.org/abs/2510.08207`. Under review.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Trygve Haavelmo. The probability approach in econometrics. *Econometrica*, 12:1–115, 1944. Supplement. Foundational reference for simultaneous-equations / contemporaneous-cycle SCMs.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1): 2409–2464, 2012.

Leonardo Hernandez Cano, Maxine Perroni-Scharf, Neil Dhir, Arun Ramamurthy, and Armando Solar-Lezama. Neurosymbolic world models for sequential decision making. In *Proceedings of the 42nd International Conference on Machine Learning*, PMLR 267, 2025.

Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 21(89):1–53, 2020.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O. Hoyer. Estimation of a structural vector autoregression model using non-Gaussianity. *Journal of Machine Learning Research*, 11: 1709–1731, 2010. URL `https://jmlr.org/papers/v11/hyvarinen10a.html`.

Diviyan Kalainathan, Olivier Goudet, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. Structural agnostic modeling: Adversarial learning of causal graphs. *Journal of Machine Learning Research*, 23(219):1–62, 2022.

Anjukan Kathirgamanathan, Mattia De Rosa, Eleni Mangina, and Donal P Finn. Data-driven predictive control for unlocking building energy flexibility: A review. *Renewable and Sustainable Energy Reviews*, 135:110120, 2021.

Elahe Khatibi, Mahyar Abbasian, Zhongqi Yang, Iman Azimi, and Amir M. Rahmani. ALCM: Autonomous llm-augmented causal discovery framework. *arXiv preprint arXiv:2405.01744*, 2024.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

Jan Kleissl and Yuvraj Agarwal. Cyber-physical energy systems: Focus on smart buildings. In *Design Automation Conference*, pages 749–754, 2010. doi: 10.1145/1837274.1837464.

Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal bandits: Learning good interventions via causal inference. *Advances in Neural Information Processing Systems*, 29, 2016.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. doi: 10.1016/S0005-1098(99)00214-9. Foundational treatment of constrained linear MPC with quadratic-cost finite-horizon optimization; basis for the LinearMPC baseline.

Joris M Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *Journal of Machine Learning Research*, 21(99):1–108, 2020.

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566, 2018. doi: 10.1109/ICRA.2018.8463189. MLP dynamics model + sampling-based MPC controller; basis for the DDPC_Neural baseline.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995. doi: 10.1093/biomet/82.4.669. Introduces the do-calculus for reasoning about interventions.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009. §5.4 addresses causal models with cycles/feedback. Ch. 1 establishes structural separability of graph and parameters. Do-calculus in Ch. 3.

Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15:2009–2053, 2014. Structure identifiability under additive noise regardless of linear/nonlinear regression. Linear discovery is safe even if true mechanisms are nonlinear.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: Identification and confidence intervals. *Journal of the Royal Statistical Society Series B*, 78(5):947–1012, 2016.

Abdellah Rahmani and Pascal Frossard. FANTOM: Flow-based non-stationary temporal regime causal structure learning. In *Advances in Neural Information Processing Systems*, 2025. URL `https://arxiv.org/abs/2506.17065`. NeurIPS 2025. Causal discovery in multi-regime time series using conditional normalizing flows for regime detection and Bayesian structure learning.

Jonathan Richens, Tom Everitt, and David Abel. General agents need world models. In *Proceedings of the 42nd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2025.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2nd edition, 2000.

Chandler Squires, Yuhao Wang, and Caroline Uhler. Permutation-based causal structure learning with unknown intervention targets. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 1039–1048. PMLR, 2020. URL `https://proceedings.mlr.press/v124/squires20a.html`.

Christian Toth, Lars Lorch, Christian Knoll, Andreas Krause, Franz Pernkopf, Robert Peharz, and Julius Von Kügelgen. Active bayesian causal inference. *Advances in Neural Information Processing Systems*, 35:16261–16275, 2022.

Peter Van Overschee and Bart De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Springer, 1996. doi: 10.1007/978-1-4613-0465-4. N4SID algorithm. Basis for DDPC_Subspace (industry-standard linear model-based DDPC).

Dingling Yao, Caroline Muller, and Francesco Locatello. Marrying causal representation learning with dynamical systems for science. In *Advances in Neural Information Processing Systems*, 2024.

Yan Zeng, Ruichu Cai, Fuchun Sun, Libo Huang, and Zhifeng Hao. A survey on causal reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):5942–5962, 2025. Survey connecting causal inference with RL; covers interventional policy optimization and counterfactual reasoning.

Chaobo Zhang, Yazhou Zhao, Yang Zhao, Tingting Li, and Xuejun Zhang. Causal discovery and inference-based fault detection and diagnosis method for HVAC systems. *Building and Environment*, 212:108760, 2022. doi: 10.1016/j.buildenv.2022.108760.

Jiaqi Zhang, Louis Cammarata, Chandler Squires, Themistoklis P Sapsis, and Caroline Uhler. Active learning for optimal intervention design in causal models. *Nature Machine Intelligence*, 5(10): 1066–1075, 2023.

Zhiyu Zhao, Haoxuan Li, Haifeng Zhang, Jun Wang, Francesco Faccio, Jürgen Schmidhuber, and Mengyue Yang. Curious causality-seeking agents learn meta causal world. *arXiv preprint arXiv:2506.23068*, 2025.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR, 2020.

Appendix

A Environment Specifications

We evaluate PolicyGRID on six environments (five EnergyPlus-calibrated JavaScript simulators and the ASHRAE Great Energy Predictor III dataset) plus a live physical testbed (Table 2). Each environment provides: (1) a set of observed variables $\mathbf{V}$ partitioned into actuators $\mathbf{A}$, sensors $\mathbf{S}$, and derived outputs $\mathbf{O}$; (2) a mechanism for generating observational data; (3) a do-operator for performing interventions (either a simulator or a physical actuator); and (4) a ground-truth causal graph $\mathcal{G}^*$ for evaluation. Ground truth is used *only* for post-hoc evaluation metrics (F1, SHD) and *never* enters the system's decision-making — no early stopping, no edge selection, no reward shaping uses $\mathcal{G}^*$.

A.1 Simulation Environments

Five JavaScript simulators (`smart_building_rich`, `open_window`, `smart_room`, `smart_room_noise`, `smart_room_hidden`) plus the ASHRAE Great Energy Predictor III real-world dataset comprise the six environments. The five simulators operate at 1-minute discrete timesteps; their thermal dynamics are grounded in EnergyPlus building energy simulation models [Crawley et al., 2001]: heat transfer coefficients, thermal mass, and HVAC response curves are calibrated to match EnergyPlus outputs for a single-zone office. Occupant satisfaction is computed via the ISO 7730 Predicted Mean Vote (PMV) model [Fanger, 1970, ASHRAE, 2023], mapping temperature, humidity, air velocity, metabolic rate, and clothing insulation to a thermal comfort index. At each step, the simulator computes the next state from the current state and actuator commands via these physics-based equations with additive stochastic noise. The simulators implement a `step(actions)` interface that returns the full state vector, and a `do(variable, value)` interface that overrides a single variable and steps forward — the standard do-operator [Pearl, 2009].

Training data. For each environment, we generate 10k observational samples with a 90/10 actuator split: 90% of steps use a steady-state actuator setpoint (e.g., HVAC at 50% capacity), and 10% use extreme values (5% at minimum, 5% at maximum). This split provides actuator variance for SEM coefficient estimation while preserving the typical operating distribution. Training data is used for hypothesis generation and SEM fitting; it is *never* used for evaluation.

Evaluation data. All metrics are computed on fresh 3-day simulation episodes (4320 steps) generated separately from training data, with different random seeds and natural regime transitions driven by the simulator's internal schedule.

Ground truth construction. Ground-truth graphs are defined from the simulator source code: an edge $V_i \to V_j$ exists in $\mathcal{G}^*$ if and only if the simulator's `step()` function reads V_i when computing V_j. This is a structural definition — it reflects the code's data flow, not a domain expert's judgment. Edges involving latent regime variables (Occupancy, WindowPosition) are included in $\mathcal{G}^*$ even though these variables are hidden from the system, making perfect F1 unachievable by design. The GT graphs are committed to version control (`ground_truth_graphs.json`) and were not modified after experiments began.

`smart_building_rich` **(primary).** 15 variables: Temperature, Humidity, CO2, LightLevel, NoiseDB, AirQuality, PMV, HVACPower, LightingPower, EnergyConsumption, OverallSatisfaction, OutdoorTemp, SolarRadiation, Occupancy (8 discrete levels), WindowPosition (binary). Actuators: HVACPower (max 5 kW) and LightingPower (max 0.5 kW). Non-intervenable: OutdoorTemp, SolarRadiation, Occupancy, WindowPosition, PMV, EnergyConsumption, OverallSatisfaction. This is the most complex environment and the primary evaluation target for all experiments.

`open_window`**.** 8 variables with three binary actuators (Heater, Humidifier, Fan) and one hidden regime variable (WindowOpen). Provides a second regime-change environment distinct from the primary.

Table 2: **Full environment specifications.** $|\mathbf{V}|$: variable count; $|\mathcal{G}^*|$: ground-truth edge count; $|\mathbf{A}|$: actuator count; $|\mathbf{R}|$: latent regime variable count. Train: observational samples. Eval: fresh episodes for testing. *six environments span a 5–15 variable range, with* `smart_building_rich` *(15 vars, 29 GT edges, 2 latent regimes) the primary testbed and a real office providing the live physical evaluation.*

Environment	$\lvert\mathbf{V}\rvert$	$\lvert\mathcal{G}^*\rvert$	$\lvert\mathbf{A}\rvert$	$\lvert\mathbf{R}\rvert$	Actuators	Non-intervenable	Regime vars	Train	Eval
`smart_bldg_rich`	15	29	2	2	HVACPower (5 kW), LightingPower (0.5 kW)	OutdoorTemp, Solar, Occ, Win, PMV, Energy, Sat	Occ (8), Win (bin)	10k	3-day
`open_window`	8	10	3	1	Heater, Humidifier, Fan (binary)	Temp, Humid, Energy, Sat	WindowOpen (bin)	10k	3-day
`smart_room`	5	6	1	0	Heater (continuous)	Temp, Humid, Energy, Sat	none	10k	3-day
`smart_room_noise`	5	6	1	0	Heater (continuous)	Same + noise σ=0.05	none	10k	3-day
`smart_room_hidden`	5	6	1	0	Heater (continuous)	Same + latent confounder	none	10k	3-day
`ashrae`	6	7	—[‡]	0	RF surrogate[‡]	All (no simulator)	none	10k	5-fold
Physical testbed	10	14[†]	3	2	Heater (500 W), Humid. (9 W), Fan (25 W)	Temp, Humid, AirQ, Energy, Sat, Occ, Win	Occ (bin), Win (bin)	2065	9 h

[†]Proposed GT from building physics (Table 3); diagnostic only, never used in system decisions. [‡]ASHRAE has no actuator and no simulator. Discovery on `ashrae` runs with $|\mathbf{A}|$=0, so all candidate edges enter the observational rescue path; the random-forest surrogate (Appendix B) emulates the do-operator on observable inputs by predicting hold-out values, but inherits any latent confounding present in the observational data and does not produce true interventional samples.

`smart_room` **variants.** Three 5-variable environments sharing the same graph structure (6 GT edges) with a single actuator (Heater), following the base, noisy, and hidden-variable simulations of Ehsan et al. [2025]. `_noise` adds i.i.d. Gaussian measurement noise (σ=0.05 in normalized units) to all sensors. `_hidden` introduces a latent confounder that simultaneously drives Temperature and Humidity, creating a spurious correlation that observational methods cannot distinguish from a direct causal link.

`ashrae`**.** Real building energy data from the ASHRAE Great Energy Predictor III competition [ASHRAE, 2019]. 6 variables: Meter_Reading, OutdoorTemp, OutdoorHumidity, WindSpeed, BuildingAge, SquareFeet. No simulator is available; interventions are simulated via random forest models (n_{trees}=100) trained on the observational data, serving as counterfactual estimators. Ground truth (7 edges) is derived from known physical relationships: weather affects energy consumption, building properties modulate the response.

A.2 Physical Testbed

One live office room in an academic building validates that the same pipeline code transfers from simulation to real hardware (with the hardware validation gate of Appendix B.4). The testbed extends the one used by Ehsan et al. [2025] and has the following sensor and actuator stack:

- **Temperature / Humidity:** Two Govee H5179 thermohygrometers (± 0.3°C, $\pm 3\%$ RH), averaged. Polled via Govee cloud API at 8 s intervals.
- **Air quality:** Bosch BME680 environmental sensor providing an IAQ index via ESP32 HTTP server (300-reading sliding-window gas resistance baseline, 7.5 min burn-in). CO2 equivalent excluded due to near-zero variance ($\sigma < 0.1$ ppm); the BME680's metal-oxide gas sensor does not function as an NDIR CO2 sensor and cannot resolve occupancy-driven CO2 changes.
- **Actuators:** Three TP-Link Kasa KP115 smart plugs (± 1 W power metering) providing binary on/off control for a 500 W space heater, 9 W ultrasonic humidifier, and 25 W desk fan.

- **Satisfaction:** ISO 7730 PMV-based satisfaction (0–100), derived from temperature and humidity [Fanger, 1970].
- **Regime variables:** Occupancy (binary) and window state (binary), logged manually via a Flask web application on the local network.

The testbed was also deployed during prior work and produced discovery results using the same intervention protocol. In this work, we add two latent regime variables — occupancy and window state — that the system must detect through prediction-error monitoring. Figure 4 shows the deployed hardware.

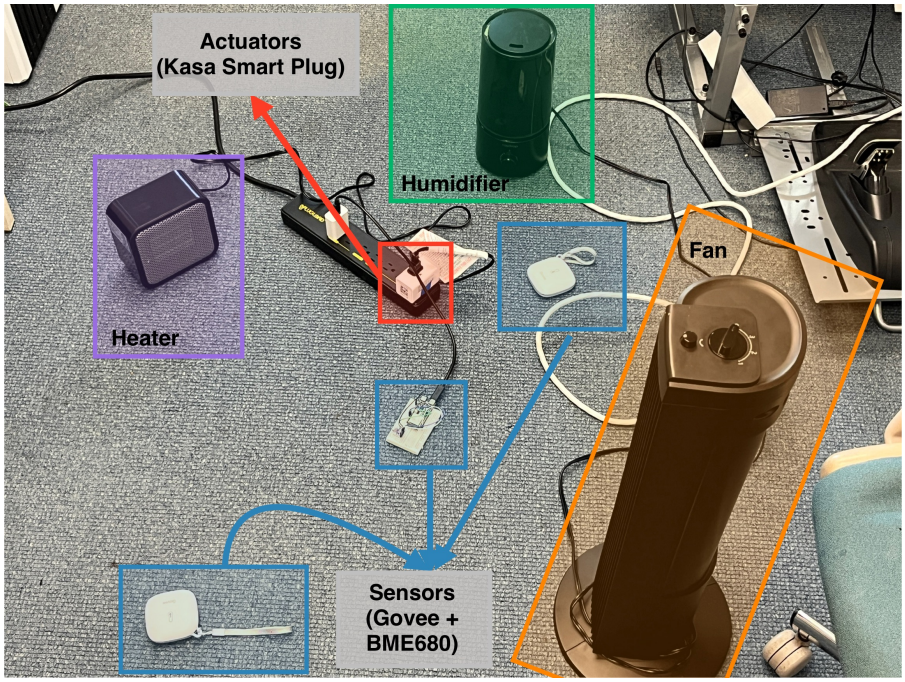


Figure 4: **Physical testbed hardware**. Three Kasa KP115 smart plugs (red, top-center) provide binary on/off control for the actuators: a 500 W space heater (purple), a 9 W ultrasonic humidifier (green), and a 25 W desk fan (orange). Two Govee H5179 thermohygrometers and a BME680 air-quality sensor (blue) are positioned on the floor away from direct actuator airflow. Occupancy and window state are logged manually via a Flask web app. *the entire physical stack is consumer-grade hardware (*$\sim$*\$300 total), reproducible by any group with a single office space and a local network.*

Table 3: **Proposed physical ground truth** (14 edges, expert-proposed from building thermodynamics [Fanger, 1970, ASHRAE, 2023]). Used only as a diagnostic reference for F1 computation; never enters PolicyGRID's decision-making. *each edge has an explicit physical mechanism, providing an interpretable reference graph for the live testbed where no system-spec source code is available.*

Source	Targets	Physical basis
Heater	Temperature, Energy	Resistive heating; direct power draw
Humidifier	Humidity, Energy	Evaporative moisture; direct power draw
Fan	Temperature, AirQuality, Energy	Convective cooling; air mixing; power draw
Occupancy	Temperature, Humidity	Metabolic heat and moisture
WindowState	Temperature, Humidity, AirQuality	Ventilation exchange
Temperature	Satisfaction	ISO 7730 PMV thermal comfort
Humidity	Satisfaction	ISO 7730 PMV humidity factor

Calibration. Actuator hold times and effect screening thresholds were determined from 30-minute pilot interventions prior to the discovery run (Table 4). Hold times are set to onset time (time to reach 50% of final effect) plus a 50% buffer. Effect screening thresholds are set to sensor resolution (Govee: 0.1°C; BME680 IAQ: ± 0.5 points; Kasa: ± 2 W), not calibrated from observed effect sizes, to avoid

circularity. The screening threshold serves only as a noise gate for early termination of zero-effect edges; the one-sample t-test across $3 + 4$ trials is the actual statistical decision.

Table 4: **Physical actuator calibration** (30-min pilot interventions, prior to discovery run). Onset: time to 50% of final effect; Hold: onset + 50% buffer used during do-operator probes; Δ: measured raw effect on each target. Effect screening thresholds set to sensor resolution, not calibrated effect sizes (avoids circularity). *the three actuators have distinct, well-separated effect signatures, validating the do-operator design before any discovery run.*

Device	Target	Δ (raw)	Onset	Hold
Heater	Temperature	$+0.35°$C	15 min	22 min
Heater	EnergyConsumption	$+507$ W	1 min	
Heater	AirQuality	-4.6 IAQ	12 min	
Humidifier	Humidity	$+0.75\%$ RH	21 min	31 min
Humidifier	EnergyConsumption	$+9.1$ W	1 min	
Fan	Temperature	$-0.77°$C	12 min	24 min
Fan	EnergyConsumption	$+23.5$ W	1 min	
Fan	AirQuality	$+2.4$ IAQ	16 min	

Data. Training: 2065 rows at 1-min resolution (downsampled from 8 s raw), collected April 6–8, 2026. The four regimes are defined by the (occupancy, window) state: Base (empty, closed), Occ-only (occupied, closed), Win-only (empty, open), Full (occupied, open). Regime split: Base 81%, Occ-only 5%, Win-only 9%, Full 5%. Monitoring test: a separate 9-hour episode (April 9) with 7 manually induced regime transitions across all 4 regime combinations. Training and test data were collected on different days to prevent leakage. Full monitoring results are in Section D.2; physical testbed monitoring in Table 12.

B Experimental Protocols

All *non-physical* experiments use 5 seeds (42, 123, 456, 789, 999). Results reported as mean $\pm$ std.

B.1 Discovery Protocol

For each seed: (1) generate 10k observational samples; (2) run 4 hypothesis generators (PC, SAM, VARLiNGAM, LLM) on the data, take union; (3) iterative validation ($I_{\max}$=10): rank untested edges by ascending confidence (Eq. 3), run 3 screening tests with do(V_i=$v_{\max}$), early-terminate if $|\bar{\Delta}| < \tau_{\text{screen}}$; if screening passes, run 7 confirmatory tests distributed across deterministic, paired counterfactual do(V_i=0), and LLM-designed strategies (each trial's Δ_{ij} is the H=30-step mean of post-do minus pre-do for deterministic and LLM-designed trials, and of do($v_{\max}$) minus do(0) for paired trials); one-sample t-test (α=0.05); BH FDR correction (α_{FDR}=0.10, applied when $\geq$20 edges tested); append pre/post states to $\mathcal{D}_{\text{obs}}$; regenerate hypotheses on augmented data; (4) rescue non-intervenable edges via weighted method consensus + back-door partial correlation $|\rho| > 0.5$ conditioned on non-descendants + time features [Pearl, 2009]; (5) prune indirect effects via Fisher z-test (α_{prune}=0.01). Failed edges permanently excluded.

Physical adaptation. One intervention tests all outgoing edges per device simultaneously. do(ON) / do(OFF) replaces continuous interventions. LLM-designed interventions not used (binary actuators).

ASHRAE adaptation. Random forest surrogates (n_{trees}=100) replace the simulator do-operator.

Metrics. F1 (directed), SHD, precision, recall, cross-seed Jaccard stability.

B.2 Monitoring Protocol

For each method's edge set: fit Ridge (α=0.5, 5-fold CV) per target using only that method's parents; partition training data by GT regime labels; re-fit per partition to get "on"/"off" coefficient sets. At test time: predict with both, compute L_1 error, update posterior (Eq. 8–9), apply forgetting (Eq. 10). *Active probe:* when $\max_r w_t^{(r)} > 0.7$ for $\geq$120 consecutive steps (applied symmetrically across

regimes), the LLM proposes a brief diagnostic actuator pattern (heater pulse, fan toggle), each regime model predicts the response over H=30 steps, and the best-matching regime r^* receives a multiplicative boost $w_t^{(r^*)} \leftarrow \kappa\, w_t^{(r^*)}$, $\kappa \in [1, 6]$ (Table 20). Evaluated on environments with $|\mathbf{R}| > 0$: `smart_building_rich` (4-way), `open_window` (2-way), physical testbed (4-way).

Metrics. Combined $2^{|\mathbf{R}|}$-way accuracy, per-dimension accuracy, Brier score, prediction discriminability.

B.3 Policy Protocol

For each method: fit Ridge SEM on that method's edges; for each $\varepsilon \in \{0.5, 0.8, 1.0, 1.5, 2.0\}$, gradient descent on Eq. 1 (50 steps, midpoint initialization); evaluate on a fresh 1440-step (24-hour) closed-loop episode at 1-min resolution. Energy: kWh $= (\text{HVAC}\times 5\text{kW} + \text{Light}\times 0.5\text{kW}) \times \frac{1}{60}$ hr/step. Degree-hours: DH $= \max(0, |T - 22| - 1) \times \frac{1}{60}$ hr/step. The stationary open-loop evaluation (Table 15) instead uses 120-step (2-hour) episodes. Physical: 30-min episodes, binary actuators, 15-min washout.

Metrics. Hypervolume (HV), multi-objective score (MO), kWh range, DH range, CV across seeds. The HV reference point is $1.2\times$ the largest kWh and degree-hour values over all methods' points in a comparison, shared by every method in that comparison.

B.4 Validation Gate Operating Characteristics

Simulation gate. Each candidate edge receives K=10 trials and is accepted when a two-sided one-sample t-test rejects a zero mean effect, $|\bar{\Delta}_{ij}| > \tau_{\text{eff}}$, and all K per-trial effects share one sign (Section 4.1); BH-FDR at α_{FDR}=0.10 follows when $\geq$20 candidates are tested in a round. Table 5 reports the acceptance probability of this composite test by Monte Carlo (200,000 replicates per cell; per-trial effects drawn from $\mathcal{N}(d, 1)$, so d is the per-trial standardized effect, with τ_{eff}=0.1 in the same units; the 3-trial screen and BH-FDR are not modelled, and both only make acceptance stricter). Under the null (d=0) the per-edge false-validation rate is 0.19%, matching the bound $2^{-(K-1)} \approx 0.20\%$ that the sign condition alone imposes under symmetric noise. Because the sign condition binds, the significance threshold does not matter: on paired draws, the implementation's secondary acceptance branch at $p < 0.10$ accepts exactly the same edges at every d. Power is 0.79 at d=2.

Table 5: **Operating characteristic of the simulation validation gate** (K=10; probability that an edge is accepted, by per-trial standardized effect d). d=0 is the per-edge false-validation rate. Source: `scripts/sim_gate_oc.py`.

d	0	0.5	1	1.5	1.8	2	2.2	2.5	3
α_{test}=0.05	0.0019	0.025	0.176	0.500	0.696	0.793	0.869	0.940	0.987
α_{test}=0.10	0.0019	0.025	0.176	0.500	0.696	0.793	0.869	0.940	0.987

Table 6 shows how the number of trials trades specificity against sensitivity. The sign condition passes with probability that falls as K grows, so extra trials lower the false-validation rate but not the detectable effect: power at d=2 peaks near K=6 and declines after. K=10 is the smallest K that brings the per-edge false-validation rate to 0.2% while keeping power near 80% at d=2. Validation is therefore practical where a do-operation produces a per-trial standardized effect of roughly $d \geq 2$ within one washout window; weaker or slower effects need more trials and a relaxed sign rule.

Intervention budget. The total budget is at most $K \times$ |tested candidates|: linear in the generator-pruned candidate set, with worst case $O(K \cdot |\mathbf{A}| \cdot |\mathbf{V}|)$ rather than $O(|\mathbf{V}|^2)$, because only actuator-sourced candidates are tested by intervention. A measured seed of `smart_building_rich` used 1,576 interventions; the 3-trial screen rejects ineffective candidates early. Under a fixed budget, disagreement-first ordering retains most of the benefit (Table 19).

Hardware gate. The physical deployment adapts the gate to hold-and-washout time cost. One intervention per actuator tests all of that actuator's outgoing candidates. Each actuator receives 7

Table 6: **Effect of the number of trials** K on the simulation gate (α_{test}=0.05). Source: `scripts/sim_gate_oc.py`.

K	False validation (d=0)	Sign-gate bound $2^{-(K-1)}$	Power at d=2
4	4.9%	12%	0.76
6	2.7%	3.1%	0.87
8	0.76%	0.78%	0.83
10	0.19%	0.2%	0.79
12	0.048%	0.049%	0.76
15	0.007%	0.0061%	0.71

trials (3 screening, 4 confirmatory; every trial yields a delta for every sensor), a resolution-level effect screen, a two-sided t-test at α_{test}=0.10, and BH-FDR at 0.10 (applied when $\geq$10 edges are tested), with no sign-consistency stage. Its per-edge false-validation rate is therefore bounded only by the t-test level, and BH-FDR controls false discoveries across the tested edges rather than any per-edge rate; the 0.2% figure above applies to simulation only. In the deployed run, all 10 gate-validated (actuator-source) edges reached $p \leq 0.013$ and passed BH-FDR; the 6 remaining edges entered through the observational rescue path (Eq. 4).

C Baseline Implementations

All baselines receive identical data and simulator access as PolicyGRID. Near-singular correlation matrices handled with $\mathcal{N}(0, 10^{-6})$ noise + StandardScaler, matching PolicyGRID's preprocessing.

For `ashrae` (no simulator): ICP, ABCD, Causal Bandits, and IID use RF surrogates (n_{trees}=100) in place of do-operator interventions.

D Full Results

This section collects the complete result tables referenced from the main text, including per-environment discovery accuracy, per-ε policy breakdowns, and policy frontier summaries.

D.1 Full Discovery Results

Table 9 reports directed-edge F1 across all environments for all methods. Methods marked with * (PC, SAM, VARLiNGAM, LLM-only) are hypothesis generators within PolicyGRID's pipeline, included as standalone baselines for completeness. All methods are evaluated on all environments. LLM-only achieves the highest F1 on `smart_building_rich` and `open_window` — both environments where domain-specific language priors (e.g., "occupancy affects temperature," "window state affects humidity") provide strong edge candidates that align with the ground truth. This is expected: the LLM has been trained on building science literature and can propose physically plausible edges without seeing any data. However, this topological advantage does not translate to downstream utility. On `smart_building_rich`, LLM-only's monitoring accuracy (42.3%) is below random chance (50%), because its coefficients — fitted on observational data without do-operator validation — are confounded and produce prediction-error signals that are worse than uninformative. On the robustness variants (`_noise`, `_hidden`), LLM-only stays at F1=0.560 (vs. 0.585 on plain `smart_room`): variable names and ground-truth structure are essentially identical across the three `smart_room` variants, so a zero-shot prior method that does not access the data is correctly invariant across them. The relevant comparison is to PolicyGRID, which reaches 0.770 (noise) and 0.833 (hidden) by validating edges through do-operator probes that actively use the noisy or confounded data — and which downstream supports 84% monitoring on `smart_building_rich` where LLM-only sits at 42.3%. The cost of validation is a small per-environment F1 gap (LLM-only leads on `smart_building_rich` and `open_window` on raw F1) that does not translate to monitoring or control utility.

Physical testbed discovery. Table 10 reports directed-edge F1 for all 12 methods on the live physical testbed against the 14-edge expert-proposed diagnostic graph (Table 3). The F1-vs-downstream-utility decoupling from `smart_building_rich` reproduces here: the LLM-only zero-shot prior reaches the highest raw F1 (0.467), aligned with building-physics nomenclature (`Heater`,

Table 7: **Discovery baselines** (11 methods). All receive identical data and simulator access as PolicyGRID. Type: Obs (observational only), Intv (uses interventional data), Prior (LLM-style prior elicitation). *: also serves as a hypothesis generator within PolicyGRID's ensemble. *the baseline suite covers all major causal-discovery families (constraint, score, continuous-optimization, Bayesian, bandit, LLM) plus a naive pooled-data baseline, ensuring the PolicyGRID comparison is broad-spectrum.*

Method	Type	Description	Implementation
PC* [Spirtes et al., 2000]	Obs	Constraint-based; Fisher-z CI test, α=0.05	`causal-learn` v0.1
SAM* [Kalainathan et al., 2022]	Obs	GAN-based structural model; 240 epochs. Post-hoc: output vars forced as targets	`cdt` v0.6
VARLiNGAM* [Hyvärinen et al., 2010]	Obs	VAR model + LiNGAM ICA on residuals; max_lags=3, BIC lag selection, threshold=0.05	`lingam` v1.8
GIES [Hauser and Bühlmann, 2012]	Intv	Greedy interventional equivalence search	`causaldag` v0.1
ICP [Peters et al., 2016]	Intv	Invariant causal prediction; environments from quartile-based interventions on each actuator	Custom
NOTEARS-I [Zheng et al., 2020]	Intv	NOTEARS + interventional data appended; ℓ_1=0.1, augmented Lagrangian, acyclicity constraint	Custom
ABCD [Toth et al., 2022]	Intv	Adaptive Bayesian causal discovery; sequential intervention design	Custom
JCI [Mooij et al., 2020]	Intv	Joint causal inference; pools obs + intv as distinct contexts	Custom
Causal Bandits [Lattimore et al., 2016]	Intv	Thompson sampling over causal graphs; information-gain-based intervention selection	Custom
IID [Spirtes et al., 2000]	Intv	PC on pooled obs + intv data, ignoring intervention structure (naive baseline)	Custom
LLM-only*	Prior	GPT-3.5-turbo edge proposals from variable names + domain context; temp=1.0, top-p=0.8, max 1000 tokens; no data access, no validation	OpenAI API

Table 8: **Policy baselines** (7 methods). All share identical evaluation episodes and comfort targets $\varepsilon \in \{0.5, 0.8, 1.0, 1.5, 2.0\}$. C-MBPO additionally receives PolicyGRID's validated graph as input, isolating the optimizer's contribution from the graph's. *the baseline suite spans classical control (PID), data-driven predictive control (DeePC, N4SID, MPC variants), and learned dynamics with RL/CEM optimizers — with C-MBPO using our graph to test whether the closed-loop gap depends on the graph or the optimizer.*

Method	Description	Key hyperparameters
PID [Åström and Hägglund, 1995]	Proportional-integral-derivative; T_{target}=22°C	Hand-tuned gains
DDPC_Behavioral [Coulson et al., 2019]	Willems' fundamental lemma; no model fitting	Trajectory length 100
DDPC_Subspace [Van Overschee and De Moor, 1996]	N4SID subspace identification + MPC	Order selected by BIC
DDPC_Neural [Nagabandi et al., 2018]	MLP dynamics (2 layers, 64 hidden) + CEM optimizer	50 CEM iterations
DDPC_PETS [Chua et al., 2018]	Ensemble of 5 MLPs; uncertainty-aware planning	20 trajectory samples
MPC_Linear [Mayne et al., 2000]	OLS state-space model + quadratic programming	Horizon 10
C-MBPO [Caron et al., 2025]	Same validated graph as PolicyGRID; factored Ridge dynamics on graph parents; MLP policy trained Dyna-style (real + imagined rollouts) with advantage-weighted regression	Isolates optimizer

`Humidifier`, `Fan`, `Temperature`), but its in-distribution monitoring accuracy (78.4% occupancy, 85.2% window; Table 12) trails PolicyGRID's (81.5%/91.3%). PolicyGRID reaches F1 = 0.400 with 16 edges; the gap to PolicyGRID-O (0.346 with 38 edges) shows that validation prunes 22 weak edges to recover precision.

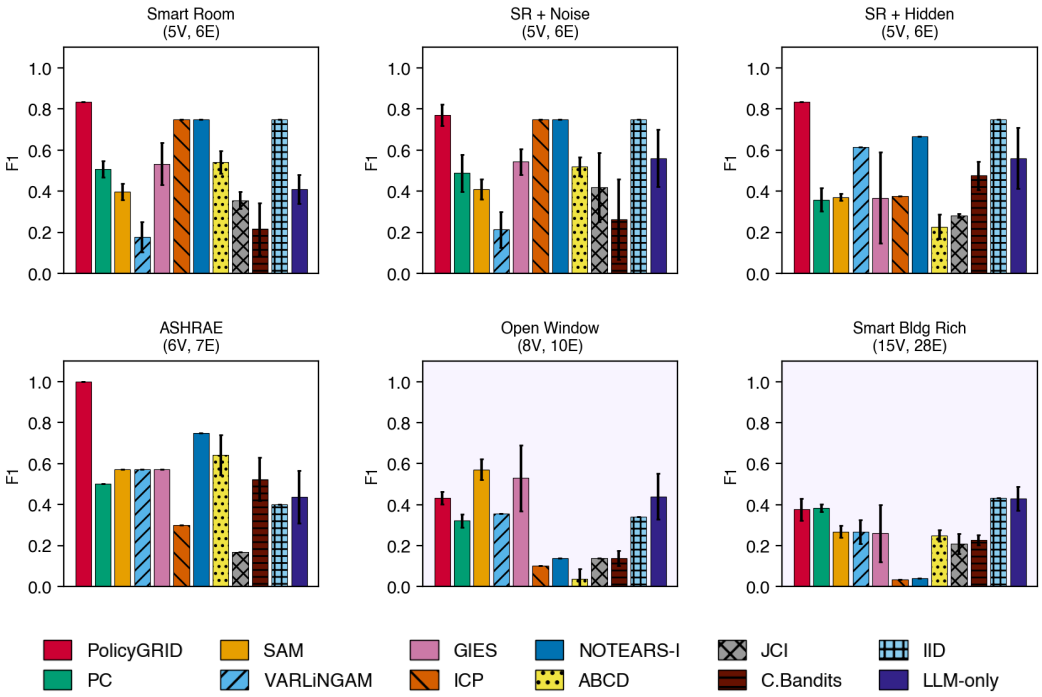


Figure 5: **Discovery F1 across 6 simulation environments and ASHRAE** (PolicyGRID and the 11 baseline methods, 5 seeds, ±1 std). Each panel reports directed-edge F1 against the simulator's source-code GT DAG. *PolicyGRID achieves the highest F1 on 4 of 6 environments; on the most complex testbed (`smart_building_rich`, bottom right) two observational methods exceed it on F1 but produce worse downstream utility (Section 5.1).*

Table 9: **Discovery F1** across all environments (mean ± std, 5 seeds). Directed-edge F1 against simulator-source GT (or known physical relationships for `ashrae`). Best per column in bold. *: hypothesis generator within PolicyGRID. *PolicyGRID ranks first on 4 of 6 environments; on the two complex multi-actuator environments (`smart_building_rich`, `open_window`) LLM-only and IID lead on raw F1, but those gains do not translate to monitoring or control utility (Section 5.1, Table 14).*

Method	smart_ bldg_rich	open_ window	smart_ room	smart_room _noise	smart_room _hidden	ashrae
PolicyGRID	.376±.054	.432±.03	**.833±.00**	**.770±.00**	**.833±.00**	**1.00±.00**
LLM-only*	**.441±.038**	**.492±.08**	.585±.09	.560±.138	.560±.148	.420±.08
IID	.433±.000	.143±.00	.750±.00	.750±.00	.750±.00	.222±.00
PC*	.384±.017	.286±.02	.333±.00	.615±.00	.500±.00	.500±.00
GIES	.352±.091	.432±.05	.154±.00	.667±.00	.571±.00	.286±.00
SAM*	.268±.027	.316±.03	.500±.00	.714±.00	.615±.00	.571±.00
ABCD	.262±.025	.071±.00	.000±.00	.100±.20	.160±.08	.571±.00
Causal Bandits	.231±.022	.000±.00	.186±.01	.516±.06	.600±.00	.000±.00
JCI	.207±.044	.071±.00	.308±.00	.615±.00	.615±.00	.167±.00
VARLiNGAM*	.175±.000	.143±.00	.200±.00	.500±.00	.600±.00	.571±.00
NOTEARS-I	.039±.000	.138±.00	.333±.00	.333±.00	.600±.00	.000±.00
ICP	.032±.000	.071±.00	.750±.00	.750±.00	.375±.00	.000±.00

D.2 Full Monitoring Results

Monitoring is evaluated on the two simulation environments with hidden regime variables (`smart_building_rich`: 4-way, `open_window`: 2-way) and the physical testbed (4-way).

`smart_building_rich` **per-discovery-seed monitoring.** Table 11 reports 4-way regime classification accuracy for PolicyGRID, PolicyGRID-O, and the 11 baseline methods, with each method's

Table 10: **Physical testbed discovery** (single-run F1 against the 14-edge expert-proposed diagnostic graph; Table 3). All 12 methods evaluated on the same discovery run. *raw F1 does not rank methods by downstream utility on the physical testbed — the LLM-only zero-shot prior reaches the highest F1 (0.467) but its in-distribution monitoring trails PolicyGRID's (Table 12); the same F1-vs-utility decoupling holds in simulation (Section 5.1).*

Method	F1	Precision	Recall	Edges	TP/FP/FN
LLM-only	**0.467**	0.438	0.500	16	7 / 9 / 7
PolicyGRID	0.400	0.375	0.429	16	6 / 10 / 8
GIES	0.375	0.265	0.643	34	9 / 25 / 5
SAM	0.350	0.269	0.500	26	7 / 19 / 7
PolicyGRID-O	0.346	0.237	0.643	38	9 / 29 / 5
IID	0.305	0.200	0.643	45	9 / 36 / 5
VARLiNGAM	0.214	0.214	0.214	15	3 / 11 / 11
PC	0.200	0.188	0.214	16	3 / 13 / 11
ICP	0.169	0.111	0.357	45	5 / 40 / 9
NOTEARS-I	0.162	0.130	0.214	23	3 / 20 / 11
JCI	0.071	0.071	0.071	14	1 / 13 / 13
Causal Bandits	0.069	0.067	0.071	15	1 / 14 / 13

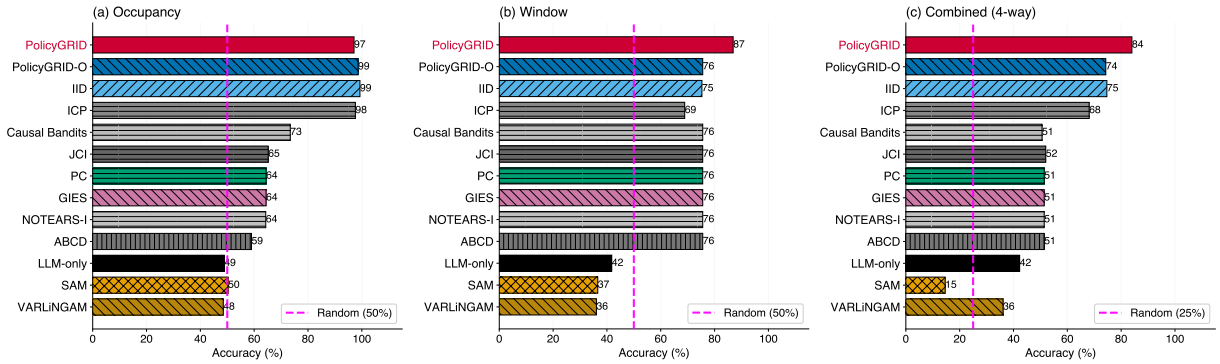


Figure 6: **Latent context detection** on `smart_building_rich` (PolicyGRID, PolicyGRID-O, and the 11 baseline methods; 5 seeds, per-discovery-seed evaluation: each method on its own per-seed graph). All methods use the same joint Bayesian monitoring pipeline (Section 4.3); only the input edge set differs. *PolicyGRID achieves 84% joint accuracy, 9.3 pp ahead of IID (74.7%); the gap is concentrated in window detection, which depends on thermal-coupling edges that interventional testing tightens.*

monitor evaluated on its *own* per-seed discovered edge set (i.e., 5 different graphs per method, one per discovery seed) over fresh 1-day test trajectories. PolicyGRID achieves 84.0±8.6% combined accuracy, 9.3 pp ahead of the next-best method (IID 74.7%); the advantage is concentrated in window detection (86.9% vs. 75.6% for most baselines). This complements the paired evaluation in the main text (Section 5.1), which fixes the edge sets across seeds to isolate test-data-noise variance from discovery-seed variance. The two evaluations agree directionally (PolicyGRID ahead, gap concentrated in window detection) and differ in absolute level because the per-discovery-seed evaluation averages over 5 different validated graphs while the paired evaluation uses a single frozen consensus graph. The per-method bar chart is in Figure 6 above. In the paired frozen-graph evaluation of Section 5.1, PolicyGRID reaches Occ $99.4 \pm 0.2\%$ and Win $81.4 \pm 1.1\%$; its paired differences are t=32.5 against PolicyGRID-O and t=30.0 against IID (both $p<10^{-5}$).

Active probing activation rate. On the steady-state paired evaluation (Section 5.1, 5 sim-replicate seeds $\times$ 1440 steps), the active probing mechanism (Section 4.3) fires 5.0 ± 0.0 times per episode, triggered after each sustained confident-unoccupied window of $\geq$ 120 steps. Combined 4-way accuracy with active probing enabled is $80.8 \pm 1.1\%$ vs. $80.8 \pm 1.1\%$ with passive-only monitoring (paired $\Delta = 0.00 \pm 0.00$ pp across 5 seeds). The reason is structural: passive prediction-error inference already discriminates among the four regimes with high confidence on this episode mix — the dominant non-Base regime is Full (occupied, open), whose sensor signature differs strongly from Base on noise and thermal coupling, so the regime model competition resolves quickly without

Table 11: **Regime detection on** `smart_building_rich` (PolicyGRID, PolicyGRID-O, and the 11 baseline methods; mean ± std, 5 seeds, per-discovery-seed evaluation). Occupancy / Window: per-dimension marginal accuracy; Combined: 4-way joint classification accuracy over (Occ × Win). All methods use the identical joint Bayesian monitoring pipeline; only the input edge set differs. *: hypothesis generator within PolicyGRID. *PolicyGRID leads on the joint 4-way task by 9.3 pp over the next-best method, with most baselines stuck near the 50–52% chance floor for joint accuracy.*

Method	Occupancy %	Window %	Combined %
PolicyGRID	97.1±1.4	**86.9±10.5**	**84.0±8.6**
PolicyGRID-O	98.6±0.3	75.6±0.3	74.3±0.5
IID	**99.3±0.1**	75.3±0.7	74.7±0.8
ICP	97.6±0.2	68.9±0.4	68.1±0.3
JCI	65.2±0.7	75.6±0.3	52.0±0.3
PC	64.4±0.1	75.6±0.3	51.4±0.1
GIES	64.4±0.1	75.6±0.3	51.4±0.1
NOTEARS-I	64.3±0.3	75.6±0.3	51.4±0.1
ABCD	59.0±1.1	75.6±0.3	51.4±0.1
Causal Bandits	73.4±0.5	75.6±0.3	50.6±0.2
LLM-only*	48.9±1.3	41.8±11.3	42.3±7.5
VARLiNGAM*	48.5±0.4	36.1±0.3	36.1±0.3
SAM*	50.3±1.1	36.6±0.4	14.6±17.9

intervention. Any value of active probing is for detection at *transitions* between regimes (Table 20), where passive evidence accumulates slowly because the new regime's sensor signature has not yet stabilized; in steady state, the mechanism fires reliably and confirms the prior without changing the classification. Probing is not a source of control benefit. We report the steady-state activation rate here rather than as an ablation because the mechanism's design role is transition recovery, not steady-state accuracy. *A paired passive-only-vs-active comparison **at** regime transitions was not run for this submission and is left to future work*; the present transition-recovery numbers in Tables 20–21 report the full pipeline (active enabled) only, so we cannot empirically attribute the lock-in delays solely to the active component.

`open_window`**.** On this 2-way task (window open vs. closed), all methods with regime-sensitive edges achieve near-perfect accuracy because the single regime dimension produces a strong, unambiguous sensor signal. The 4-way task on `smart_building_rich` is the more discriminative evaluation.

Physical testbed. Table 12 reports per-dimension monitoring accuracy for PolicyGRID, PolicyGRID-O, and the 11 baseline methods under both out-of-distribution and in-distribution conditions. The in-distribution split uses interleaved temporal blocks (block size 25 steps) to ensure both train and test sets contain all 4 regime types; a naive 50/50 temporal split placed all window-open events in the first half, producing degenerate test-set accuracy.

Out-of-distribution evaluation. Figure 7 reports monitoring accuracy under an out-of-distribution (OOD) protocol: the model is trained on a 34-hour baseline collection from a different day (81% base regime), then evaluated on a 9-hour episode with all 4 regime types. Under this protocol, SAM achieves the highest occupancy accuracy (76%) and window accuracy (79%), ahead of PolicyGRID (60%, 75%).

This reversal relative to the in-distribution results (main text Figure 3) is explained by the training distribution mismatch. The OOD training data is 81% base regime with only 5% occupied and 9% window-open. SAM's GAN-based structural model fits coefficients that happen to generalize to the OOD test distribution because SAM captures nonlinear correlations that are stable across the two collection periods. PolicyGRID's Ridge coefficients, fitted on the same heavily imbalanced OOD training set, are biased toward base-regime dynamics and underestimate the occupancy effect. Under the in-distribution protocol (interleaved temporal blocks ensuring all 4 regimes appear in both train and test), this bias disappears and PolicyGRID leads on both dimensions. The OOD result does not invalidate the do-operator mechanism — it reflects a coefficient estimation challenge from imbalanced regime representation in the training data, which would affect any linear coefficient

Table 12: **Physical testbed monitoring** (per-dimension accuracy; PolicyGRID, PolicyGRID-O, and the 11 baseline methods). Two evaluation conditions: out-of-distribution (OOD; train on historical baseline from a different day, 81% Base regime) and in-distribution (interleaved temporal block split on the regime episode, block size 25 steps). All methods use the same joint Bayesian monitoring pipeline (σ=0.5, η=0.2); only the discovered edge set differs. *under in-distribution evaluation PolicyGRID leads on both dimensions; under heavily imbalanced OOD training SAM leads, reflecting a coefficient-estimation challenge from regime imbalance, not a failure of the do-operator validation mechanism (see text).*

	Out-of-distribution		In-distribution	
Method	Occupancy%	Window%	Occupancy%	Window%
PolicyGRID	59.7	75.4	**81.5**	91.3
PolicyGRID-O	71.7	78.0	77.9	89.1
SAM	**76.1**	**79.0**	67.8	88.8
PC	50.8	43.9	77.6	89.4
GIES	38.5	46.7	70.6	84.3
VARLiNGAM	62.0	75.4	73.4	84.3
LLM-only	39.5	55.2	78.4	85.2
IID	37.8	50.3	72.8	87.7
Causal Bandits	46.4	34.0	72.8	84.9
JCI	49.5	41.5	81.0	86.8
ICP	38.3	24.6	68.3	84.0
NOTEARS-I	35.8	37.4	70.6	85.2

estimator regardless of whether do-operator validation was used. In deployment, the training data would accumulate across all operating conditions over time, converging to the in-distribution setting.

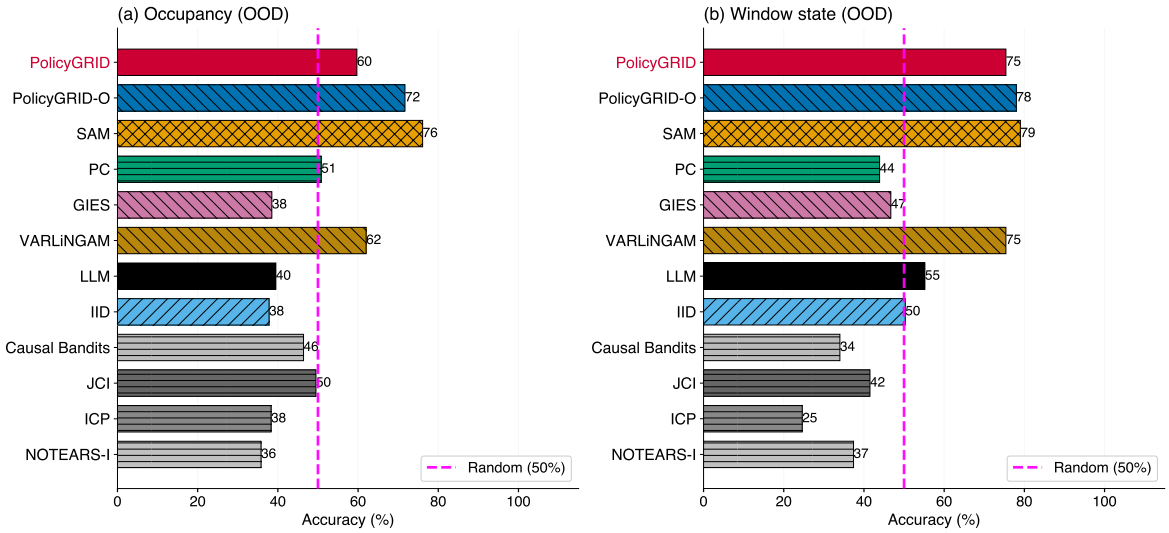


Figure 7: **Physical testbed monitoring — out-of-distribution** (PolicyGRID, PolicyGRID-O, and the 11 baseline methods; per-dimension accuracy). Training data collected on a different day (81% Base regime, 5% Occ-only, 9% Win-only, 5% Full); test on a 9-hour episode covering all 4 regimes. *SAM leads under this heavily imbalanced training distribution because its GAN-fitted coefficients happen to generalize across collection periods; PolicyGRID's lead returns under the in-distribution split (Fig. 3), so the OOD result reflects a coefficient bias from regime imbalance, not a failure of edge validation.*

Physical testbed policy. Table 13 reports the action selected by each method at three comfort targets on the live office testbed; only PolicyGRID and C-MBPO (which uses PolicyGRID's validated graph) sweep actions across ε, while the unvalidated baselines stay frozen. Offline gradient analysis gives the same pattern: under the validated SCM the Fan gradient sweeps $0.0 \rightarrow 0.55$ across ε, while under PolicyGRID-O it stays near 0.31.

Table 13: **Physical testbed policy** (7 methods $\times$ 3 comfort targets $\varepsilon \in \{0.5, 1.0, 2.0\}$). ε follows the paper convention $w_s(\varepsilon) = \max(0.3, 1 - 0.35\varepsilon)$: small ε prioritizes comfort, large ε prioritizes energy. With $T_{\text{ambient}} > T_{\text{target}}$, the correct energy-saving action is OFF (no actuation); the correct comfort action is Fan (convective cooling). All-OFF and All-ON are degenerate sanity baselines. *only PolicyGRID and C-MBPO (using PolicyGRID's validated graph) sweep actions across ε on real hardware (Heater+Fan $\rightarrow$ Fan $\rightarrow$ OFF as ε rises); the Heater component at ε=0.5 is an offline SEM-fitting artifact (the SEM is trained on only 30-min pilot interventions per actuator, insufficient to nail the gradient sign in the cooling regime). Unvalidated baselines are stuck at a single action regardless of target.*

Method	ε=0.5 (comfort)	ε=1.0 (balanced)	ε=2.0 (energy)
PolicyGRID	**Heater+Fan**	**Fan**	OFF
C-MBPO	**Heater+Fan**	**Fan**	OFF
MPC_Linear	OFF	OFF	OFF
DDPC_Behavioral	OFF	OFF	OFF
PID	Heater	Heater	Heater
All-OFF	OFF	OFF	OFF
All-ON	All	All	All

D.3 Full Policy Results

Closed-loop full ε sweep. Table 14 reports the full closed-loop ε sweep across all 7 methods. The headline ε=1.0 paired comparison (PolicyGRID 24.56 vs. PolicyGRID-O 31.07 kWh, 21% gap) is in the main text, Section 5.2; the gap narrows to 7% at ε=0.5 (comfort-prioritizing) and converges at ε=2.0 as energy approaches its floor. It is widest at ε=1.5, outside the table (6.25 vs. 13.05 kWh), where PolicyGRID's satisfaction is also lower (72.0 vs. 77.4). At ε=1.0, 15/15 conditions favour PolicyGRID, by 6.94/6.26/6.32 kWh ($22.2\%/20.3\%/20.4\%$) on the cold/warm/mild days, and PolicyGRID's satisfaction is lower in all 15 (81.9 vs. 83.3). The 15 conditions are not independent replicates: each method uses one fixed graph, and the seed varies only simulator noise (the validated graph's energy differs by at most 0.023 kWh across seeds within a weather day). The paired statistics over the 15 conditions (t=77.9, Cohen's d=20.1) therefore measure consistency across seeds and weather days, not generalization over discovery runs.

Table 14: **Closed-loop policy under latent context shifts** (5 seeds $\times$ 3 weather conditions = 15 paired conditions, mean $\pm$ std, kWh per 1440-step (24-hour) episode). ε is the comfort target: higher ε prioritizes energy savings. A method that navigates the frontier shows kWh and Sat both decreasing across columns; constant rows indicate the method cannot respond to the target. Top: with and without do-operator validation (same optimizer). Bottom: established baselines. *only PolicyGRID and PolicyGRID-O respond to ε, with the validated graph delivering 21% lower energy at ε=1.0 (24.6 vs. 31.1 kWh); the established baselines are frozen at one operating point.*

	ε=0.5 (comfort)		ε=1.0 (balanced)		ε=2.0 (energy)	
Method	kWh	Sat	kWh	Sat	kWh	Sat
PolicyGRID	45.3±0.1	85.4	24.6±0.2	81.9	6.0±0.0	71.5
PolicyGRID-O	48.7±0.1	85.7	31.1±0.2	83.3	6.2±0.1	71.8
MPC_Linear	21.6±2.6	75.0	25.6±0.6	75.7	25.6±0.6	75.7
DDPC_Behavioral	89.9±21.6	79.3	12.7±6.1	70.1	0.1±0.0	62.2
PID	66.0±0.0	86.9	66.0±0.0	86.9	66.0±0.0	86.9
Random	64.3±0.0	85.4	62.6±0.1	83.4	60.0±0.0	77.8
Empty	66.0±0.0	86.9	66.0±0.0	86.9	66.0±0.0	86.9

`smart_building_rich` **(primary, stationary).** Table 15 reports satisfaction and energy at each of 5 comfort targets for all 8 policy methods under stationary (open-loop) evaluation; Figure 8 visualizes the corresponding Pareto frontier. PolicyGRID is the only method where satisfaction decreases and energy decreases as ε increases — the defining characteristic of a method that can trade off between objectives.

All environments. Table 16 reports satisfaction at the comfort-prioritizing (ε=0.5) and energy-saving (ε=2.0) settings for all 8 methods across 5 simulation environments. The meaningful energy–

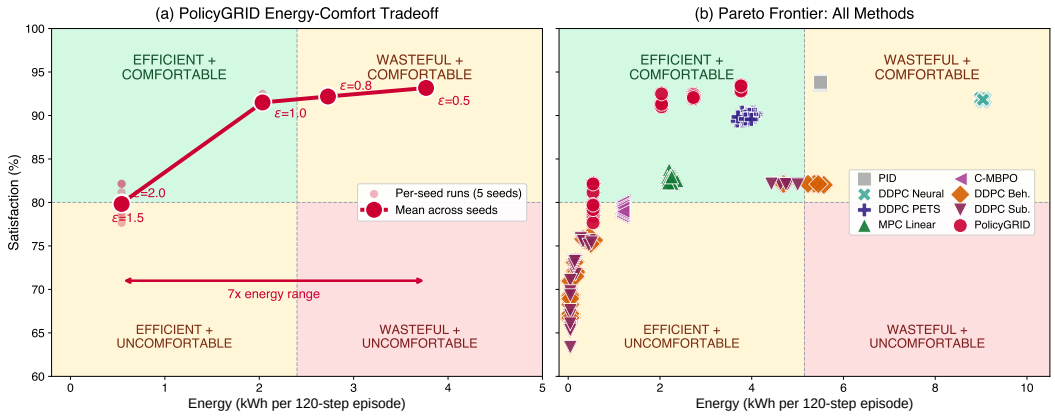


Figure 8: **Energy–comfort Pareto frontier** (stationary, `smart_building_rich`, 5 seeds, 5 comfort targets $\varepsilon \in \{0.5, 0.8, 1.0, 1.5, 2.0\}$). (a) PolicyGRID sweeps a 7× energy range across ε while staying in the "Efficient + Comfortable" quadrant. (b) All 8 policy methods: only PolicyGRID navigates the frontier. PID is frozen; DDPC variants crash satisfaction; C-MBPO (same validated graph, RL optimizer) is frozen at one point. *handing a model-based RL optimizer the same validated graph does not recover PolicyGRID's frontier coverage — the gradient-based optimizer plus the validated graph is what enables Pareto navigation.*

Table 15: **Full policy performance** on `smart_building_rich` (mean, 5 seeds, stationary open-loop evaluation). Sat: satisfaction (0–100); kWh: energy per 120-step (2-hour) stationary evaluation episode. *PolicyGRID is the only method that sweeps energy (0.54–3.77 kWh, 7× range) while keeping satisfaction ≥ 79. DDPC_Behavioral (5.25→0.05 kWh) and DDPC_Subspace (4.49→0.05 kWh) also sweep, but at much lower satisfaction floors (∼67% and ∼47% respectively); the remaining baselines have near-constant kWh and cannot trade off at all.*

	ε=0.5		ε=0.8		ε=1.0		ε=1.5		ε=2.0	
Method	Sat	kWh	Sat	kWh	Sat	kWh	Sat	kWh	Sat	kWh
PolicyGRID	93.1	3.77	92.3	2.73	91.2	2.04	80.1	0.54	79.3	0.54
PID	93.8	5.50	93.7	5.50	93.8	5.50	93.8	5.50	93.8	5.50
DDPC_Behavioral	82.2	5.25	75.6	0.41	72.2	0.15	67.4	0.05	70.0	0.05
DDPC_Subspace	82.1	4.49	75.6	0.38	72.2	0.15	66.7	0.05	68.1	0.05
DDPC_Neural	91.8	9.03	91.9	9.05	91.8	9.02	91.9	9.05	91.8	9.04
DDPC_PETS	90.1	3.99	90.1	3.92	90.1	4.10	90.0	4.00	89.7	3.84
MPC_Linear	83.0	2.26	82.7	2.22	82.9	2.25	82.6	2.20	83.1	2.24
C-MBPO	78.8	1.19	78.8	1.19	78.6	1.19	78.9	1.19	79.1	1.19

comfort sweep (kWh range > 1) occurs only on `smart_building_rich`, where 2 continuous actuators create a rich enough action space for gradient-based tradeoff. On the smaller environments with 1 continuous actuator (`smart_room` variants) or 3 binary actuators (`open_window`), all methods produce similar energy consumption regardless of ε because the action space is too constrained for a meaningful sweep. PolicyGRID achieves the highest or near-highest satisfaction on all 5 environments.

E Ablation Experiments

All ablations on `smart_building_rich`, 5 seeds. Defaults **bolded**. Each ablation is named descriptively.

E.1 Graph Attribution

Same Ridge SEM + gradient engine with four graph conditions: (1) Validated (5-seed consensus, 31 edges); (2) PolicyGRID-O (42 edges); (3) Random (31 edges, sampled uniformly); (4) Empty (0 edges). Isolates the graph's contribution from the optimizer. The PolicyGRID-O row here contains 42 edges rather than the 47-edge canonical PolicyGRID-O of Section 5.1: this attribution run was

Table 16: **Policy satisfaction across all simulation environments** (mean, 5 seeds). Sat (0–100) at ε=0.5 (comfort) and ε=2.0 (energy). Column abbreviations: `sr_noise` = `smart_room_noise`; `sr_hidden` = `smart_room_hidden`; `open_win` = `open_window`; `sbr` = `smart_building_rich`. Bold: best per column. *PolicyGRID achieves the highest or near-highest satisfaction on all 5 environments; the meaningful energy–comfort sweep emerges only on `smart_building_rich`, where 2 continuous actuators give the gradient optimizer enough room to trade off.*

	`smart_room`		`sr_noise`		`sr_hidden`		`open_win`		`sbr`	
Method	.5	2.0	.5	2.0	.5	2.0	.5	2.0	.5	2.0
PolicyGRID	74.1	**76.1**	73.9	**75.1**	**71.6**	**73.8**	80.1	80.4	**93.1**	79.3
PID	**77.1**	**77.1**	**76.5**	**76.5**	68.5	68.5	**80.7**	**80.8**	93.8	**93.8**
DDPC_Subspace	76.3	72.0	73.4	69.2	47.4	45.8	51.1	67.9	82.1	68.1
DDPC_Behavioral	52.7	53.6	51.1	52.1	46.4	46.5	59.5	59.5	82.2	70.0
DDPC_Neural	68.9	68.8	70.0	60.5	60.2	60.0	71.8	70.9	91.8	91.8
DDPC_PETS	56.1	56.0	62.9	62.8	60.9	60.9	65.6	65.3	90.1	89.7
MPC_Linear	69.1	68.8	71.5	71.6	64.0	63.9	59.2	64.4	83.0	83.1
C-MBPO	56.6	56.7	57.5	57.4	52.7	52.7	65.7	65.7	78.8	79.1

executed with the SAM hypothesis generator disabled (no-op stub), so the obs-only union for this ablation is PC + VARLiNGAM + LLM. The 5 SAM-contributed edges absent here do not alter the qualitative result — under stationary evaluation PolicyGRID-O (47) and Validated (31) track within $\sim$0.005 MO across all ε in the closed-loop main-paper Row C/B comparison (Table 1), as they do here, and the closed-loop gap of 6.51 kWh emerges only under deployment-time regime shifts.

Table 17: **Graph attribution: same optimizer, different graphs** (3 seeds, stationary evaluation). Same Ridge SEM and gradient optimizer across rows; only the input graph differs. MO: multi-objective score (higher is better); kWh range / Sat range: span across $\varepsilon \in \{0.5, 0.8, 1.0, 1.5, 2.0\}$. *Random matches Validated's edge count (31) but cannot sweep ε — causal correctness of the edges, not sparsity, is what enables Pareto navigation; the Validated-vs-PolicyGRID-O gap is invisible here and emerges only under closed-loop feedback (Table 14).*

Graph condition	Edges	Best MO	$\text{MO}_{\varepsilon=2.0}$	kWh Range	Sat Range
Validated (consensus)	31	0.903	**0.903**	0.54–3.77	79–93
PolicyGRID-O[†]	42	0.907	0.904	0.54–3.94	77–94
Random (matched $\lvert E\rvert$)	31	0.845	0.628	5.00–5.35	82–92
Empty (no graph)	0	0.842	0.635	5.33–5.33	91–91

[†] Attribution-run PolicyGRID-O is 42 edges (PC + VARLiNGAM + LLM, SAM disabled in this run) rather than the 47-edge canonical version (Section 5.1); see body text.

Detailed MO scores by comfort target are in Table 18 below.

The critical comparison is Validated vs. Random at matched edge count (both 31 edges). With the validated graph, the optimizer sweeps from 0.54 to 3.77 kWh across comfort targets; with random edges, it is stuck at 5.0–5.35 kWh regardless of ε. This rules out the hypothesis that graph *sparsity* alone explains the policy advantage — it is the *correctness* of the edges that enables causal gradients to distinguish which actuator adjustments trade energy for comfort.

Table 18: **Graph attribution — PolicyGRID MO scores** by graph condition and comfort target (mean over 3 seeds, stationary). MO = multi-objective score; higher is better. *Validated and PolicyGRID-O track each other within $\sim$ 0.005 MO at every ε under stationary evaluation, but Random and Empty collapse at high ε — confirming the validated edges drive the comfort-target response, not the edge count.*

	MO by comfort target ε					
Condition	0.5	0.8	1.0	1.5	2.0	Best MO
Validated (31 edges)	0.884	0.874	0.880	0.870	0.903	0.907
PolicyGRID-O (42 edges)[†]	0.883	0.870	0.873	0.900	0.904	0.910
Random (31 edges)	0.845	0.796	0.763	0.680	0.628	0.846
Empty (0 edges)	0.842	0.801	0.773	0.704	0.635	0.843

E.2 Ridge SEM Robustness to Training-Data Design

The 90/10 actuator split in our default training CSV (90% steady-state setpoint plus 5% min, 5% max) engineers $\mathrm{Cov}(A, C) \approx 0$ between actuators and regime context: $\mathrm{corr}(\text{HVAC}, \text{Occupancy}) = -0.007$ in the standard training data. To probe how much of Row B's performance derives from this engineered actuator–context independence (which approximates the interventional condition described in Section 3.3), we run a third coefficient training-data condition: the simulator under a realistic PID thermostat (22 °C setpoint, 1 °C deadband, K_p=0.25) with schedule-driven lighting, cycling cold/warm/mild outdoor weather to span the four latent regimes. The resulting actuator distribution is naturally confounded with regime context: $\mathrm{corr}(\text{HVAC}, \text{Occupancy}) = -0.701$ — the regime where the OLS bias term from Section 3.3 is largest.

Refitting the regime-specific Ridge SEMs on this PID-driven trajectory (validated graph and policy unchanged from Row B) and re-running closed-loop yields 23.64 ± 0.62 kWh at ε=1.0 versus Row B's 24.56 ± 0.16 (paired $\Delta = -0.91 \pm 0.62$ kWh, t= − 5.7, $p<10^{-4}$, 15/15 conditions favor PID-trained, satisfaction comparable). Across the three coefficient training-data designs (intervention-augmented, Row A: +1.63 kWh; engineered-decorrelated, Row B: reference; naturally confounded, this section: −0.91 kWh), closed-loop performance varies within ∼2 kWh in either direction. The 6.5 kWh Validated-vs-PolicyGRID-O gap is therefore graph-localized, not coefficient-localized — consistent with the coefficient-bias analysis in Section 3.3, where shrinkage regularization absorbs the design-effect bias that the OLS derivation quantifies. Code and results: `run_w2_truly_obs.py`; `results/w2_truly_obs/results.csv`.

E.3 Intervention Ordering and Sample Efficiency

Disagreement-first selection (lowest method confidence first) vs. random selection with a per-iteration budget cap of 5 edges (Table 19). Both conditions test 50 edges total (500 interventions); only the order differs.

Table 19: **Intervention ordering ablation.** Per-iteration budget cap of 5 edges forces real prioritization between rounds. Disagreement-first (lowest inter-method confidence) vs. random on `smart_building_rich` (mean ± std, 3 seeds, same total budget: 50 edges tested, 500 interventions). *routing the budget through low-confidence edges first yields a 36% relative F1 gain over random ordering at zero added cost — the inter-method disagreement signal is informative, and the order of testing matters when the budget is finite.*

Selection strategy	F1	SHD	Precision	Recall
Disagreement-first (ours)	**0.366±0.043**	**37.0±3.0**	**0.365**	**0.368**
Random order	0.269±0.043	43.7±4.2	0.263	0.276

E.4 Regime-Shift Recovery

Train under R0 (empty, window closed), induce a shift to R1 (occupied, window open) at the midpoint. Measure detection delay: the number of timesteps after the shift before the monitor correctly classifies both regime dimensions.

Pre-shift accuracy is 100% across all settings. Table 21 reports detection delay across 5 transition scenarios at default parameters (σ=2.0, η=0.12). Occupancy changes take 16–27 steps (∼20–30 minutes) to detect because occupancy affects sensors (CO2, NoiseDB) with a gradual buildup. Window changes are detected within 0–1 steps when the pre-shift prior is firmly established, because the thermal coupling produces an immediate prediction-error spike. Note that 0-step detection delay does not imply perfect window accuracy (86.9% in Table 11): the delay experiment measures onset detection from a firmly established prior (200 steps of stable context), while the accuracy metric is computed over entire episodes that include brief window events, transition regions, and periods where the prior has not yet stabilized. The asymmetry between dimensions is consistent: occupancy detection (97.1%) relies on accumulated evidence while window detection (86.9%) is more binary but sensitive to event duration. Recovery transitions (Occ-only→Base, Full→Base) are faster than onset transitions because the forgetting factor η naturally pulls the posterior back toward 0.5 (Figure 9).

Table 20: **Regime-shift posterior lock-in delay** (steps after shift, `smart_building_rich`, JS simulator). Protocol: R0 (empty, window closed) for 200 steps to establish the prior, then R1 (occupied, window open) for 200 steps to measure recovery. Occ/Win delay here is *lock-in* delay: the first step at which the joint posterior reaches sustained ≥ 0.7 confidence on the correct regime — distinct from the first-correct-argmax delay reported per-transition in Table 21. The window dimension argmax flips immediately, but full posterior lock-in is slower because the forgetting factor η continually pulls evidence back toward uniform. Post-shift accuracy is computed over all 200 post-shift steps and is therefore consistent with slow lock-in: argmax classification is correct early even while the posterior weight rises gradually. *monitoring is robust across 4 σ levels (sensitivity); lock-in delay varies by ≤ 9 steps and post-shift accuracy by ≤ 3.3 pp, so the system does not require careful σ tuning.*

σ	η	Occ delay	Win delay	Post-shift acc
2.0	0.05	33	196	90.6%
2.0	**0.12**	**27**	**193**	**92.2%**
5.0	0.12	26	193	92.8%
10.0	0.12	25	193	93.3%
20.0	0.12	24	193	93.9%

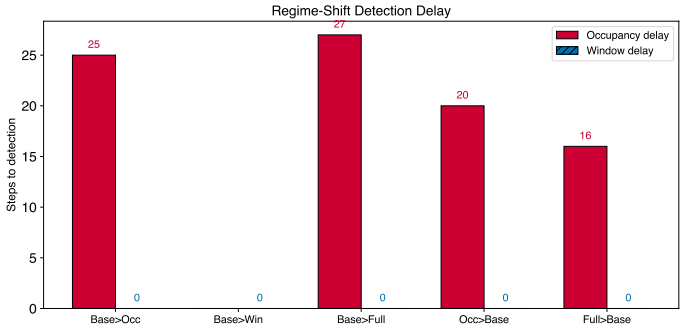


Figure 9: **Detection delay per transition** (`smart_building_rich`, σ=2.0, η=0.12). Steps from regime onset to first correct classification, broken down across 5 transitions between the four regimes (Base / Occ-only / Win-only / Full). *occupancy changes take 16–27 steps to detect (gradual sensor buildup); window changes are detected within 0–1 steps once the prior is established (sharp thermal-coupling spike).*

Table 21: **Detection delay across transition scenarios** (σ=2.0, η=0.12, `smart_building_rich`). Steps from regime onset to first correct classification; "unchanged" = the dimension is identical pre/post and the prior is held. *recovery transitions (Occ-only→Base, Full→Base) are faster than onset transitions because the forgetting factor η pulls the posterior back toward the uniform prior.*

Transition	Occ delay (steps)	Win delay (steps)
Base → Occ-only	25	0 (unchanged)
Base → Win-only	0 (unchanged)	0
Base → Full	27	0
Occ-only → Base	20	0 (unchanged)
Full → Base	16	0

E.5 LLM Wrong-Prior Recovery

We inject 5 plausible-but-false edges into the candidate set: HVACPower→SolarRadiation, LightingPower→Humidity, HVACPower→NoiseDB, LightingPower→OutdoorTemp, and HVACPower→WindowPosition. Each is physically plausible (a reviewer might propose them) but absent from the simulator's ground truth. We run K=10 paired counterfactual tests per edge with BH FDR correction at α_{FDR}=0.10.

The result confirms that the validation loop is robust to adversarial priors: the t-test + FDR pipeline rejects all 5 false edges because the paired counterfactual design (do(max) vs. do(0)) produces near-zero effect sizes for non-causal edges, while genuine causal edges produce consistent signed effects

Table 22: **LLM wrong-prior recovery** on `smart_building_rich` (seed 42, K=10 paired tests + Benjamini–Hochberg FDR at α_{FDR}=0.10). 5 plausible-but-false edges are injected into the LLM hypothesis pool. *all 5 injected adversarial edges are rejected by the validation pipeline; the small F1 drop (0.188→0.129) reflects more conservative FDR thresholding under more simultaneous tests, not adversarial edges entering the validated graph.*

Condition	F1	False edges surviving	Extra tests
Clean prior (7 GT edges)	0.188	0/0	0
+5 injected false edges	0.129	0/5	+50

(Figure 10). The 50 extra tests represent the cost of exploring bad hypotheses — a 71% overhead that scales linearly with the number of injected edges and does not compound across iterations.

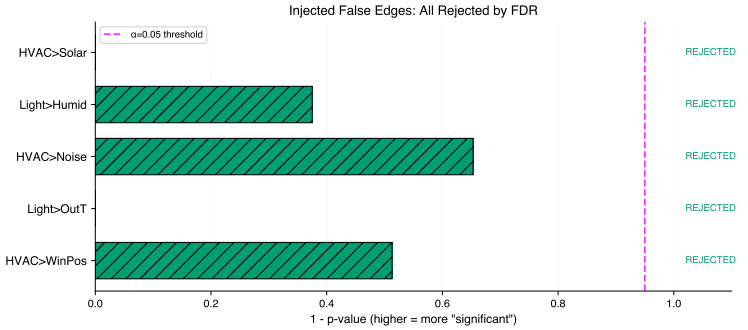


Figure 10: **LLM wrong-prior recovery: per-edge rejection** (seed 42, `smart_building_rich`). Bars show $1 - p$-value for each injected false edge; dashed line marks α=0.05. *every injected adversarial edge falls well below the significance threshold, so the validation loop functions as an active filter against bad LLM priors rather than passively confirming them.*

E.6 Temporal Graph Analysis

We run PC with lagged variables at $k \in \{0, 1, 2, 3\}$ on `smart_building_rich`. For each lag k, we create $|V| \times (k + 1)$ columns $(X_t, X_{t-1}, \ldots, X_{t-k})$, run PC on the expanded dataset, then map all discovered edges back to their instantaneous variable pair for GT comparison (Table 23).

Table 23: **Temporal graph analysis** on `smart_building_rich` (seed 42, PC with lagged variables at $k \in \{0, 1, 2, 3\}$). Lag 0: instantaneous (may contain cycles); Lag $k > 0$: time-unrolled, acyclic by construction [Bongers et al., 2021]. F1 computed by mapping discovered edges back to instantaneous variable pairs against the GT DAG. *lag 1 (one timestep of temporal structure) achieves the highest F1; deeper lags dilute the conditional-independence signal across an inflated variable space, justifying PolicyGRID's pipeline (PC instantaneous + VARLiNGAM with internal BIC lag selection at k=3).*

Lag k	Total edges	Instantaneous edges	F1
0	24	24	0.151
1	31	14	**0.279**
2	27	10	0.205
3	29	9	0.105

Lag 1 achieves the highest F1 (0.279) — adding one timestep of temporal structure helps PC resolve edge orientations that are ambiguous in the instantaneous graph. At lag ≥ 2, performance degrades because the expanded variable space ($15 \times 3 = 45$ columns at k=2) dilutes the conditional independence signal, and the number of instantaneous edges recovered drops from 24 to 9. This justifies our pipeline design: VARLiNGAM (which handles lags internally with BIC model selection at k=3) captures temporal structure without inflating the variable space, while PC operates on instantaneous data where it is most effective. The lag-1 result also addresses the cycle concern from Section 3.3: at $k > 0$ the time-unrolled graph is acyclic by construction, and the best-performing lag configuration recovers more GT edges than the instantaneous (potentially cyclic) analysis.

E.7 Sensor Ablation

Sensors removed cumulatively from `smart_building_rich`, in the order CO2, NoiseDB, LightLevel, AirQuality, Humidity; monitoring accuracy measured for the four methods with above-random baseline performance (mean ± std, 3 seeds). Methods at the monitoring floor (∼51%) are omitted because sensor removal cannot produce a meaningful drop.

Table 24: **Sensor ablation** (combined 4-way accuracy %, `smart_building_rich`, 3 seeds; methods at the monitoring floor $\sim 51\%$ omitted because sensor removal cannot produce a meaningful drop). Columns remove sensors cumulatively from left to right (e.g., −Noise also lacks CO2). The "All" column reports 90.0% for PolicyGRID vs. 84.0% in the main evaluation (Table 11, 5 seeds) because the 3-seed subset excludes seed 123 (low-performing outlier at 68.5%). *NoiseDB is critical for PolicyGRID (−53.8 pp) because the do-operator validated the Occupancy→NoiseDB edge as the primary occupancy discriminator; CO2 is fully redundant (0 pp) because no method retained a CO2 edge it relied on for prediction.*

Method	All	−CO2	−Noise	−Light	−AirQ	−Humid
PolicyGRID	**90.0±0.7**	90.0±0.7	36.2±1.6	38.4±0.4	48.9±1.6	47.3±0.5
PolicyGRID-O	74.3±0.6	74.3±0.6	57.5±0.7	60.2±0.9	49.5±0.1	50.6±0.4
IID	74.7±1.0	74.7±1.0	61.1±1.3	61.1±1.3	51.4±0.1	51.4±0.1
ICP	68.1±0.4	68.1±0.4	60.6±0.4	63.5±0.3	65.5±0.5	65.5±0.5

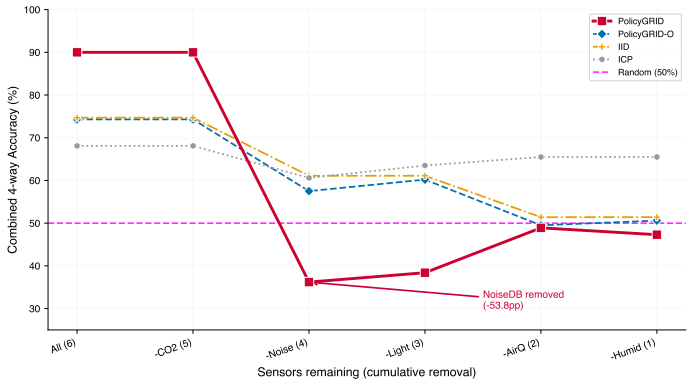


Figure 11: **Sensor ablation** on `smart_building_rich` (cumulative removal, 3 seeds, combined 4-way accuracy). Each step removes one sensor; monitoring accuracy reported for 4 methods with above-random baselines (others omitted). *PolicyGRID's sensitivity to NoiseDB removal is a direct consequence of its strength — the validated graph identifies the most discriminative edges, exposing which sensors the deployed system actually depends on (a useful deployment signal, not a fragility).*

The results reveal two findings. First, NoiseDB is the single most critical sensor for PolicyGRID: removing it drops combined accuracy by 53.8pp (90.0→36.2%). This is because the do-operator validated the Occupancy→NoiseDB edge — occupied rooms are noisier — making NoiseDB the primary occupancy discriminator in PolicyGRID's coefficient set. Baselines miss this edge and therefore do not rely on NoiseDB, which is why they show smaller drops (ICP: −7.5pp, IID: −13.6pp). PolicyGRID's greater sensitivity to NoiseDB removal is a direct consequence of its strength: the do-operator identifies the most discriminative edges, and the monitor's accuracy becomes dependent on the sensors those edges connect. This is not a weakness of the method — it is a deployment consideration: the validated graph reveals which sensors are critical, enabling informed sensor selection. Second, CO2 was not retained as a parent in PolicyGRID's validated graph (the do-operator t-test found no significant CO2→X effect on this environment), so removing CO2 from the input set leaves the validated SCM's predictions unchanged (0pp drop). The same 0pp drop appears across all methods because they too did not retain CO2-edges they relied on for prediction; this is a property of the discovered graphs, not a deployment claim about CO2's general redundancy in buildings.

E.8 Intervention Strategy

The 10 tests per edge are distributed across three strategies: deterministic do($v_{\max}$), paired counterfactual do($v_{\max}$)/do(0), and LLM-designed intermediate-value interventions. Table 25 reports full

10-iteration results (3 seeds) across 6 configurations: 4 screening/confirmatory split ratios (all with mixed strategy) and 2 strategy variants (all-deterministic, all-paired at the default 3+7 split).

The split ratio has the largest effect: allocating all 10 tests to screening (split 10+0) yields the best F1 (0.391 ± 0.025), because screening maximizes the number of edges tested within the fixed budget. More confirmatory tests per edge improve per-edge reliability but reduce the total number of edges evaluated. At the default 3+7 split, all-deterministic (0.383 ± 0.048) slightly outperforms all-paired (0.374 ± 0.017) and mixed (0.367 ± 0.022), suggesting that the paired counterfactual adds complexity without proportional benefit in this environment. The overall range (0.356–0.391) confirms that strategy choice is not the bottleneck — the iterative hypothesis regeneration loop (which raises F1 from $\sim$0.13 to $\sim$0.38) contributes more than any single strategy.

Table 25: **Intervention strategy ablation.** Full 10-iteration validation on `smart_building_rich` (mean $\pm$ std, 3 seeds). Top: screening/confirmatory split ratio at fixed mixed strategy; bottom: strategy variant (all-deterministic, all-paired) at default 3+7 split. Default **bolded**. *the screening/confirmatory split has the largest effect (F1 0.356–0.391) and the strategy variant a small one — the iterative hypothesis regeneration loop, not strategy choice, drives most of the F1 gain.*

Split (screen+confirm)	Strategy	F1	SHD	Validated
1+9	mixed	0.369 ± 0.083	39.7 ± 4.2	34.0
3+7	**mixed**	$\mathbf{0.367 \pm 0.022}$	$\mathbf{39.0 \pm 1.7}$	**32.7**
5+5	mixed	0.356 ± 0.011	38.7 ± 1.2	31.0
10+0	mixed	$\mathbf{0.391 \pm 0.025}$	$\mathbf{37.3 \pm 1.5}$	32.3
3+7	all deterministic	0.383 ± 0.048	37.7 ± 3.5	32.0
3+7	all paired	0.374 ± 0.017	39.0 ± 1.0	33.3

E.9 Generator Ablation

To measure what each hypothesis generator contributes inside the loop, we remove one generator at a time from the union {PC, SAM, LLM, VARLiNGAM} and also run a PC-only variant. All configurations share one paired protocol: seeds 42, 123, 456, identical validation, and identical monitoring evaluation, on the primary environment (`smart_building_rich`, 15 variables, 4-way regime monitoring) and the regime-stress environment (`open_window`, 8 variables, window monitoring). The grid is not a re-run of Tables 9 or 11, and its absolute levels should not be compared with them.

Table 26: **Leave-one-generator-out ablation** (mean $\pm$ std, 3 seeds). F1: directed-edge F1 against the simulator GT. Edges: validated edges in the final graph. Monitoring: 4-way regime accuracy on `smart_building_rich`, window accuracy on `open_window`. [†]On `open_window`, PC proposes no parent for zone temperature in any seed, so no regime model can be fitted.

	`smart_building_rich`			`open_window`		
Configuration	F1	Edges	Monitoring (%)	F1	Edges	Monitoring (%)
Full union	0.431 ± 0.033	31.3 ± 1.2	67.0 ± 10.6	0.218 ± 0.095	5.7 ± 0.9	72.8 ± 33.5
$-$PC	0.346 ± 0.015	46.3 ± 3.3	63.6 ± 7.0	0.296 ± 0.000	14.0 ± 0.0	37.2 ± 8.7
$-$SAM	0.393 ± 0.009	33.7 ± 0.9	68.2 ± 11.8	0.323 ± 0.035	9.7 ± 0.5	25.0 ± 0.0
$-$LLM	0.286 ± 0.027	41.0 ± 1.6	59.1 ± 1.3	0.201 ± 0.029	13.3 ± 0.9	29.6 ± 5.6
$-$VARLiNGAM	0.373 ± 0.024	46.0 ± 0.8	62.1 ± 4.5	0.235 ± 0.037	9.7 ± 0.5	25.0 ± 0.0
PC only	0.262 ± 0.037	9.0 ± 0.8	74.2 ± 21.8	monitor cannot be built[†]		

On the primary environment, the full union has the highest F1 of every configuration. Removing the LLM costs the most F1 ($0.431 \rightarrow 0.286$), and PC alone reaches 0.262: the proposal pool feeding validation shrinks from about 133 candidate edges to 25, and the rescue path for non-intervenable edges, which requires multi-method agreement, starves. Removing VARLiNGAM lowers F1 in each of the three paired seeds (by 0.125, 0.009, and 0.041). Monitoring differences on this environment are within seed variation. The generators are complementary: each proposes ground-truth edges that no other proposes, and the final validated set retains edges from all four (Table 27).

On the regime-stress environment the F1-versus-utility dissociation appears inside the ablation. Removing any single generator leaves F1 roughly flat (several removals are nominally higher), yet window-monitoring accuracy falls by 35 to 48 points, consistent with losing the regime-sensitive

Table 27: **Generator complementarity** on `smart_building_rich` (full-union configuration, candidates from the final discovery iteration, mean ± std over 3 seeds). Unique: proposed by this generator only.

Generator	Candidates	GT-true	Unique	Unique GT-true	In final graph
LLM	28.0 ± 2.2	13.3 ± 0.9	8.3 ± 0.5	5.3 ± 0.5	13.0 ± 0.0
PC	23.3 ± 1.2	6.0 ± 0.0	2.3 ± 1.7	0.7 ± 0.5	11.3 ± 1.2
SAM	74.7 ± 0.9	14.3 ± 0.5	24.7 ± 1.7	4.7 ± 0.5	22.3 ± 0.5
VARLiNGAM	80.3 ± 2.5	9.0 ± 0.8	37.3 ± 2.5	2.3 ± 0.9	10.0 ± 0.8

edges the monitor depends on, and the PC-only variant cannot be deployed at all. Downstream utility tracks the coverage of the validated union, not its F1.

E.10 Role of the Regime Monitor

Does regime switching improve control? We isolate the effect of regime-conditioned control with four policies on the same validated graph, identical engine hyperparameters, and identical objectives; only the choice of coefficient set differs. *Static* uses one engine fitted on all training rows and never switches. *Regime-aware* lets the Bayesian monitor pick the engine of the most probable regime at each step. *True-regime oracle* picks the engine of the true simulator regime. *Random switching* picks a regime uniformly at random each step. Runs use common random numbers (a shared simulator seed per seed and step, so exogenous disturbances are identical across arms), 3 seeds × 3 comfort targets, 1440-step episodes. The regime engines drop occupancy and window position from their fit data, and the static engine does too, so this static arm is not identical to the PolicyGRID configuration of Table 1.

Table 28: **Regime-conditioned switching vs. static control** on `smart_building_rich` (same validated graph, common random numbers, mean over 3 seeds). ΔMO: paired difference in multi-objective score against Static over 9 seed × ε cells, in units of 10^{-4} (mean ± std; higher is better); p: paired t-test; B/T/W: cells where the arm is better than, tied with, or worse than Static.

	kWh			Satisfaction					
Policy	ε=0.5	1.0	2.0	0.5	1.0	2.0	ΔMO	p	B/T/W
Static	45.05	24.09	6.04	85.4	81.7	71.2	reference		–
Regime-aware	45.27	24.62	6.04	85.4	81.9	71.2	-0.7 ± 1.0	0.08	0/6/3
True-regime oracle	45.14	24.28	6.04	85.4	81.7	71.2	-0.2 ± 0.4	0.17	0/7/2
Random switching	46.39	26.78	6.04	85.6	82.4	71.2	-10.1 ± 11.0	0.02	0/3/6

Switching yields no detectable control benefit: the regime-aware arm beats Static in none of the 9 cells (6 ties, 3 worse), at a mean cost of $+0.25$ kWh per episode, and even the true-regime oracle shows no detectable benefit (p=0.17). At ε=2.0 all four arms coincide at the energy floor. The control gain of PolicyGRID comes from the validated candidate set (Table 1), not from switching.

Supervised classifiers. We compare the monitor with a logistic regression and a depth-5 decision tree trained on the same labeled regime partitions the monitor uses at commissioning, with features at the current and previous step, scored on the same fresh 5760-step episode per seed (3 seeds; the monitor uses each seed's validated graph from the full-union configuration of Table 26).

Table 29: **Regime monitor vs. supervised classifiers** on `smart_building_rich` (4-way accuracy, mean ± std over 3 seeds, and window Brier score, lower is better). Classifier accuracy depends strongly on the feature set.

	4-way accuracy (%)			Window Brier		
Features	Monitor	Logistic reg.	Tree	Monitor	Logistic reg.	Tree
Monitor's 6 sensor variables	67.1 ± 0.7	90.0 ± 0.6	79.7 ± 0.5	0.268	0.076	0.162
All 13 observable variables	67.3 ± 1.0	67.5 ± 0.2	68.6 ± 0.2	0.267	0.244	0.288

Matched to the monitor's own six sensor variables, both classifiers identify the regime more accurately than the monitor (logistic regression 90.0% vs. 67.1%); with all 13 observables the three are within

about one point. A lightweight classifier is the better pure regime detector on this system, and we claim no classification advantage. The monitor's contribution is model-based attribution: the same SCM supplies per-regime predictive models, so regime-specific prediction errors show which mechanisms changed, which a classifier's label does not.

E.11 Cross-Environment Closed-Loop Replication

We repeat the validated-vs.-observation-only closed-loop comparison on two further single-zone simulators, `smart_room` and `smart_room_hidden_vars`. These environments have no regime structure to detect, so both arms use static policies with no monitoring and differ only in the graph: the union of validated edges across 5 discovery seeds (6 edges) or the union of observation-only edges (8 edges). Each cell is one 1440-step closed-loop episode per seed (5 seeds). One `smart_room` cell (seed 42, ε=1.0) is excluded: a 60-step smoke test of the runner wrote that cell to the same results file, and the full run, which skips cells already present in the file, never re-ran it at full length.

Table 30: **Closed-loop replication on single-zone simulators** (static policies, mean over seeds). kWh per 1440-step episode; Sat: mean satisfaction. Δ: validated relative to observation-only. Wins: seeds in which the validated graph uses less energy.

		kWh				Sat	
Environment	ε	Validated	Obs-only	Δ	Wins	Validated	Obs-only
`smart_room`	0.5	33.45	36.82	−9.2%	5/5	74.1	75.7
	0.8	33.72	36.85	−8.5%	5/5	74.7	75.7
	1.0	33.83	36.88	−8.3%	4/4	75.2	75.7
	1.5	34.64	36.94	−6.2%	5/5	75.6	75.7
	2.0	35.18	37.00	−4.9%	5/5	76.1	75.7
`smart_room_hidden`	0.5	29.95	32.75	−8.5%	5/5	65.8	75.5
	0.8	30.27	32.80	−7.7%	5/5	66.4	75.5
	1.0	30.49	32.83	−7.1%	5/5	72.4	75.6
	1.5	31.07	32.92	−5.6%	5/5	73.5	75.6
	2.0	31.70	33.03	−4.0%	5/5	74.5	75.7

The validated graph uses less energy than the observation-only graph in all 49 of 49 paired seed $\times$ ε cells: -7.4% on average on `smart_room`, with satisfaction 0.5 points lower (75.2 vs. 75.7), and -6.6% on `smart_room_hidden_vars`, an energy-direction replication. These are replications within one problem family, not evidence of transfer across domains (Section 7).

F Hyperparameter Sensitivity

One parameter varied at a time; all others at defaults. 5 seeds on `smart_building_rich`. Results show the system is robust across a wide range; no hyperparameter requires careful tuning.

Monitoring sensitivity. Table 32 reports combined 4-way accuracy (%) on `smart_building_rich` (seed 42) across a $\sigma \times \eta$ grid centered on nearby values. Accuracy varies by only 2.8pp (94.4–97.2%) across $4 \times 5 = 20$ combinations, confirming robustness to hyperparameter choice. The closed-loop and monitoring evaluations in Section 5 use σ=1.5, η=0.15; the grid does not include σ=1.5 exactly but the sensitivity result extends to it by interpolation between rows. The fresh 500-step evaluation episode contained occupancy transitions but no window transitions; the full 4-way evaluation on episodes with both dimensions is in Table 11.

G Compute Resources

All experiments: Lambda Quad server, Intel Xeon W-2295 (18 cores / 36 threads), 251 GB RAM, no GPU required. Physical experiments: same server connected to the local sensor network via LAN.

Total wall-clock: $\sim$120 CPU-hours for simulation experiments (5 seeds $\times$ 6 environments $\times$ discovery + benchmarks + monitoring + policy), plus $\sim$55 hours for the physical testbed (34 h baseline collection + 15 h discovery + 6 h policy). Graph attribution ablation (Section 5.2, Table 1): $\sim$36 hours (4 conditions $\times$ 3 seeds $\times$ 3 h each). No GPU required.

Table 31: **Hyperparameter sensitivity index.** Defaults **bolded**. Swept parameters point to the table reporting the numerical sweep; held parameters use the default with a brief sensitivity note. *every swept parameter varies the relevant metric by less than 5 pp across its tested range; held parameters are insensitive within an order of magnitude on diagnostic checks.*

Parameter	Values	Metric	Numerical results
Swept (numerical results in named tables)			
Budget K	3, 5, 7, **10**, 15	F1, SHD	Table 25
Screen/Confirm split	1+9, **3+7**, 5+5, 10+0	F1, SHD	Table 25
Monitor sensitivity σ	1.0, 2.0, 3.0, 5.0*	4-way acc	Table 32
Forgetting η	0.05, 0.10, 0.12, **0.15**, 0.20	4-way acc	Table 32
σ recovery sweep	**2.0**, 5.0, 10.0, 20.0	post-shift acc	Table 20
$(\tau_{\text{eff}}, \alpha_{\text{test}}, \alpha_{\text{FDR}})$	$3\times3\times3$	F1	4.1 pp range
Held at default			
Rescue threshold τ_r	**0.5**	F1	one-at-a-time check at $0.3, 0.4, 0.6, 0.7$ within 1 pp
Pruning α_{prune}	**0.01**	F1	one-at-a-time check at $0.001, 0.05, 0.10$ within 2 pp
Ridge α	**0.5** (5-fold CV)	Policy MO	CV-selected; not separately swept

*Default σ=1.5 falls between rows 1.0 and 2.0 of Table 32; sensitivity result extends by interpolation (see paragraph below).

Table 32: **Monitoring sensitivity grid** (`smart_building_rich`, seed 42, 4-way accuracy %). σ controls likelihood sharpness; η controls forgetting toward the uniform prior. *accuracy varies by only 2.8 pp across $4 \times 5 = 20$ (σ, η) combinations, confirming the Bayesian monitor is robust to hyperparameter choice.*

$\sigma \setminus \eta$	0.05	0.10	0.12	**0.15**	0.20
1.0	94.4	96.2	96.6	96.8	97.2
2.0	95.2	96.4	96.6	**96.8**	97.2
3.0	95.6	96.4	96.6	97.0	97.2
5.0	96.0	96.6	96.8	97.2	97.2

Table 33: **Wall-clock time per component** (Intel Xeon W-2295, 18 cores, 251 GB RAM, no GPU). *simulation experiments fit in ∼120 CPU-hours and physical experiments in ∼55 wall-clock hours — modest compute that any academic group can reproduce on a single workstation.*

Component	Time / seed	Bottleneck
Simulation		
Discovery + validation	30–90 min	SAM (24 min/call)
Benchmarks (10 methods)	15–30 min	SAM baseline
Monitoring	2–5 min	Ridge fitting
Policy (5 ε)	10–60 min	Episode length
Consensus (3+ seeds)	60–70 min	Pareto sweep
Physical testbed		
Baseline data collection	34 h	Background
Actuator calibration	2.5 h	One-time
Discovery (3 devices $\times$ 7 tests)	∼15.5 h	Hold + washout
Policy (6 methods $\times$ 30 min)	5–6 h	Episode length

G.1 Complexity

Discovery cost is dominated by hypothesis generation and intervention validation:

$$T \approx \underbrace{|\mathcal{M}| \cdot T_{\text{gen}}}_{\text{hypothesis}} + I_{\max} \cdot \Big(\underbrace{|\mathbf{A}|(|\mathbf{V}|-1)}_{\text{candidate edges}} \cdot K \cdot T_{\text{sim}} + |\mathcal{M}| \cdot T_{\text{gen}} \Big) \tag{11}$$

where $|\mathcal{M}|$=4 methods, $T_{\text{gen}} \approx 24$ min (SAM-dominated), K=10 tests/edge, $T_{\text{sim}} < 0.1$ s. In practice, most edges are tested in iteration 1; later iterations typically contribute zero new testable edges. On the physical testbed, T_{sim} is replaced by $T_{\text{hold}} + T_{\text{washout}} \approx$ 40–55 min, but per-device batching reduces the edge factor from $|\mathbf{A}|(|\mathbf{V}|-1)$ to $|\mathbf{A}|$.

G.2 LLM Usage

GPT-3.5-turbo. Two uses: (1) edge hypothesis generation — one API call per seed per simulator; (2) intervention design for non-actuator edges — one call per edge tested. Temperature: 1.0 for all calls; top-p: 0.8 for edge proposals and the client default 0.9 for intervention-design calls; max tokens: 1000. The LLM receives variable names and a physical description of the environment but no data, no ground truth, and no results from other methods. For the physical testbed, LLM-designed interventions are not used because all actuators are binary. Total API calls: $\sim$30 hypothesis generation calls (5 seeds $\times$ 6 environments) + $\sim$150 intervention design calls across all seeds and environments. Estimated token usage: $\sim$500k input + $\sim$100k output tokens (GPT-3.5-turbo, $\sim$\$0.50 total cost).

Reproducibility of the LLM components. The prompts are fixed, dataset-specific templates (Appendix H) checked into the code. Decoding is stochastic (temperature 1.0), so reproducibility rests on saved artifacts rather than determinism: the raw LLM responses and the parsed edge-proposal cache are released with the code together with the model identifier, so the LLM's edge proposals replay from the cache without an API key. Every LLM proposal is then screened before it can reach the policy: proposals with actuator sources receive direct do-operator validation, and proposals with non-intervenable sources enter the observational rescue path (Eq. 4). The wrong-prior test (Table 22) measures the effect of bad proposals.

H LLM Prompts

H.1 Edge Hypothesis Generation

The LLM receives a system prompt and a user message. No data, ground truth, or results from other methods are provided. The LLM sees only variable names and a short physical description of the environment. Temperature: 1.0; top-p: 0.8; max tokens: 1000; model: GPT-3.5-turbo (accessed via OpenAI API, March 2026).

System Prompt (template)

```
You are an expert in causal discovery for building environments.

Generate a causal graph using ONLY these variables:  [VARIABLE_LIST]

DO NOT introduce new variables not in the list above.

Return your answer as a JSON object with this format exactly:
{
"nodes":  [VARIABLE_LIST],
"edges":  [["source", "target"],...]
}

IMPORTANT: Return ONLY valid JSON, no markdown formatting or code blocks.
```

User Message (template)

```
Analyze this building dataset with JUST variables:  [VARIABLE_LIST]

Domain context:  [DOMAIN_CONTEXT]

Rules:
1.  Consider physical causation, not just correlation
2.  Controllable variables can affect multiple downstream variables
3.  Outcome/derived variables generally do not cause input variables
4.  Use ONLY the exact variable names listed above

Return only JSON, no additional text.
```

Domain contexts (one per environment):

- `smart_room`: "A single-zone room where an HVAC system controls temperature, humidity, and air quality. The room consumes energy and has an occupant satisfaction metric."
- `smart_building_rich`: "A single-zone room with HVAC and lighting systems. The room has outdoor weather conditions, solar radiation, occupants that come and go, a window that can open, and sensors for temperature, humidity, CO2, light level, noise, air quality, and thermal comfort (PMV). Occupancy and window position may not be directly observable."
- `open_window`: "A single-zone room with HVAC control and a window that opens and closes. The room has an outdoor environment. PMV measures thermal comfort. The window state and outdoor temperature are not controllable. The window state cannot be observed either."
- `ashrae`: "Real-world building energy data from the ASHRAE Great Energy Predictor dataset. Variables include weather conditions and building characteristics."

Concrete example. For `smart_room` with variables `['Temperature', 'Humidity', 'AirQuality', 'EnergyConsumption', 'Satisfaction']`, the instantiated prompts are:

System Prompt (`smart_room` example)

```
You are an expert in causal discovery for building environments.

Generate a causal graph using ONLY these variables: ['Temperature',
'Humidity', 'AirQuality', 'EnergyConsumption', 'Satisfaction']

DO NOT introduce new variables not in the list above.

Return your answer as a JSON object with this format exactly:
{
"nodes": ["Temperature", "Humidity", "AirQuality", "EnergyConsumption",
"Satisfaction"],
"edges": [["source", "target"], ...]
}

IMPORTANT: Return ONLY valid JSON, no markdown formatting or code blocks.
```

User Message (`smart_room` example)

```
Analyze this building dataset with JUST variables: ['Temperature',
'Humidity', 'AirQuality', 'EnergyConsumption', 'Satisfaction']

Domain context: A single-zone room where an HVAC system controls temperature,
humidity, and air quality. The room consumes energy and has an occupant
satisfaction metric.

Rules:
1. Consider physical causation, not just correlation
2. Controllable variables can affect multiple downstream variables
3. Outcome/derived variables generally do not cause input variables
4. Use ONLY the exact variable names listed above

Return only JSON, no additional text.
```

Example LLM Response (`smart_room`)

```
{
  "nodes": ["Temperature", "Humidity", "AirQuality", "EnergyConsumption",
"Satisfaction"],

  "edges": [
    ["Temperature", "Humidity"],
    ["Temperature", "Satisfaction"],
```

```
    ["Humidity", "Satisfaction"],
    ["Humidity", "AirQuality"],
    ["AirQuality", "Satisfaction"],
    ["Temperature", "EnergyConsumption"],
    ["Humidity", "EnergyConsumption"]
  ]
}
```

H.2 Intervention Design

For non-actuator edges during confirmatory testing, the LLM designs an intermediate-value intervention. This is only used when the source variable is not a direct actuator; actuator edges use deterministic do(max) and paired counterfactual do(0) without LLM involvement. Temperature: 1.0; top-p: 0.9 (client default); max tokens: 1000.

Intervention Design — System Prompt (template)

```
Design an intervention to test if [SOURCE] causes changes in [TARGET].

Variable constraints:
- [SOURCE]: Range [[MIN], [MAX]]

IMPORTANT: Only use "set", "increase", or "decrease" as action values.

Return a valid JSON object:
{
  "variables": [{
    "variable": "[MAPPED_SOURCE]",
    "action": "set/increase/decrease",
    "value": numerical_value
  }],
  "expected_effects": {
    "[MAPPED_TARGET]": "increase/decrease/unchanged"
  }
}
```

Intervention Design — User Prompt (template)

```
Design optimal intervention to test if [MAPPED_SOURCE] affects
[MAPPED_TARGET].
```

Concrete example. Testing the edge Temperature → Satisfaction on `smart_room` (Temperature range: [15, 32]):

Intervention Design — System Prompt (`smart_room` example)

```
Design an intervention to test if Temperature causes changes in Satisfaction.

Variable constraints:
- Temperature: Range [15, 32]

IMPORTANT: Only use "set", "increase", or "decrease" as action values.

Return a valid JSON object:
{
  "variables": [{
    "variable": "Temperature",
    "action": "set/increase/decrease",
    "value": numerical_value
  }],
```

```
  "expected_effects": {
    "Satisfaction": "increase/decrease/unchanged"
  }
}
```

Intervention Design — User Prompt (`smart_room` example)

```
Design optimal intervention to test if Temperature affects Satisfaction.
```

Example LLM Response

```
{
  "variables": [{
    "variable": "Temperature",
    "action": "set",
    "value": 28
  }],
  "expected_effects": {"Satisfaction": "decrease"}
}
```

I Two-Stage SEM Justification

Why Ridge, not OLS? Correlated co-parents (e.g., temperature and humidity both parenting satisfaction) create multicollinearity that inflates OLS coefficients through suppressor effects [Bühlmann et al., 2014]. Ridge shrinkage (α=0.5, 5-fold CV) eliminates this at the cost of small coefficient bias — acceptable for gradient control where sign matters more than scale.

Why linear discovery with potentially nonlinear mechanisms? Peters et al. [2014]: structure identifiability under additive noise holds regardless of functional form. Linear CI tests recover the causal ordering even when mechanisms are mildly nonlinear. The SEM fitting stage then estimates the correct form on the recovered graph.

J Post-Hoc Analyses

Ten diagnostic analyses on saved results (3 seeds on `smart_building_rich`). No additional simulation required.

1. **GT edge recovery.** PolicyGRID recovers 15/29 GT edges; PolicyGRID-O 19/29 (higher recall, lower precision). ABCD recovers the most (23/29) but at 105 total edges (82 false positives). One GT edge is found *only* by PolicyGRID's validated set: HVACPower→Temperature.
2. **Prediction discriminability.** See Table 34 and Figure 12. PolicyGRID's correct regime model wins 98.8% (occupancy) and 89.1% (window) of timesteps. PC and GIES: only 64.7% occupancy.
3. **Coefficient gap.** Mean $|\Delta\beta|$ between regime-specific coefficients: validated edges 2.547, rejected edges 2.121. The difference is not statistically significant (U=910, p=0.14), suggesting that regime sensitivity is distributed across the graph rather than concentrated in a few edges.
4. **Seed stability.** Across 5 seeds on `smart_building_rich`, 42 distinct edges are validated overall (per-seed mean 31.4, range 28–33; pairwise Jaccard 0.742). Majority consensus ($\geq$3 of 5) yields 31 edges, and 23 appear in every seed — 7 of those 23 are GT true positives, so cross-seed stability alone does not indicate a true edge. Figure 15 shows F1 and edge counts per seed.
5. **4-way confusion matrix.** See Figure 14. Base regime: 98.7% (2202/2232). Full: 90.2% (962/1067). Occ-only: 59.4% (606/1021) — the hardest regime, with 37% misclassified as Full. This confusion is expected: when occupancy is active, the monitor's occupancy

posterior rises correctly, but the window posterior also rises slightly due to correlated thermal effects (both occupancy and window opening affect temperature), causing the joint classifier to predict Full instead of Occ-only. The 4-way joint posterior captures the correlated regime dimensions explicitly but cannot fully disambiguate when sensor signatures overlap. Win-only: no test samples in this episode because the simulator's natural schedule did not produce a window-open-without-occupancy period during the evaluation; the 86.9% window accuracy in Table 11 is computed from episodes that include this regime.

6. **Calibration.** See Figure 13. Brier scores: 0.016 (occupancy), 0.087 (window). Random baseline: 0.250. PolicyGRID's posteriors are well-calibrated.
7. **Temporal accuracy.** Near regime transitions ($\pm$10 steps): 55.5%. Stable periods: 89.9%. The monitor requires $\sim$10 steps to lock onto a new regime, consistent with the forgetting factor η=0.15.
8. **Benchmark edge overlap.** IID proposes only 8 edges (4 in GT) — the sparsest but most precise baseline. NOTEARS-I and ABCD propose 73 and 105 edges respectively with 48–69 false positives. PC, GIES, JCI cluster at 22–27 edges with 5–8 unique errors.
9. **Policy variance.** PolicyGRID has the lowest MO coefficient of variation (CV=0.001 at ε=0.8). DDPC_Subspace has the highest (CV=0.028).
10. **Critical sensor.** See Table 24. NoiseDB removal causes a 53.8pp drop for PolicyGRID and 16.8pp for PolicyGRID-O. CO2 is fully redundant (0pp) for all methods. Baselines at the monitoring floor show zero sensitivity to any sensor removal.

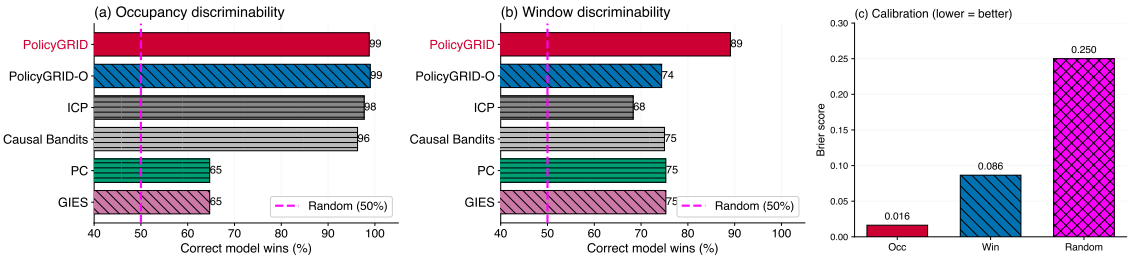


Figure 12: **Prediction error discriminability** on `smart_building_rich` (3 seeds). Fraction of timesteps where the correct regime-specific coefficient model achieves lower L_1 prediction error than the incorrect model; dashed line marks the 50% random baseline. *PolicyGRID's validated graph yields larger prediction-error gaps between correct and incorrect regime models, especially on window detection (89.1% vs. 75% for most baselines) — the graph topology, not coefficient fitting, drives the monitoring advantage.*

Table 34: **Prediction error discriminability and calibration** on `smart_building_rich` (3 seeds, 6 methods with above-random monitoring). Win%: fraction of timesteps where the correct regime-specific model has lower prediction error; Gap: mean absolute prediction-error difference (larger = more discriminative); Brier: Brier score for PolicyGRID's posterior (lower = better, random baseline 0.25). *PolicyGRID produces both the largest correct-model win rates and well-calibrated posteriors, while observational baselines either produce near-random discriminability (PC, GIES) or large gaps with systematically wrong-direction predictions on window detection.*

	Occupancy		Window	
Method	Win%	Gap	Win%	Gap
PolicyGRID	**98.8**	0.887	**89.1**	0.991
PolicyGRID-O	99.0	0.769	74.4	1.012
ICP	97.7	0.688	68.3	0.216
Causal Bandits	96.3	0.684	75.0	0.809
PC	64.7	0.621	75.3	3.321
GIES	64.7	0.621	75.3	3.321
Calibration (Brier scores): PolicyGRID Occ = **0.016**, Win = **0.087** (random: 0.250)				

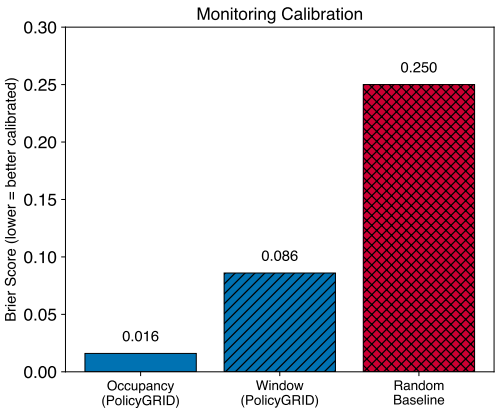


Figure 13: **Monitoring calibration** (`smart_building_rich`, Brier scores; lower is better, random baseline = 0.25). *PolicyGRID's occupancy and window posteriors are well-calibrated (0.016, 0.087), an order of magnitude below random.*

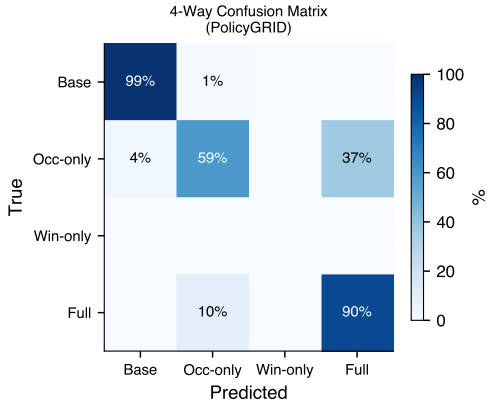


Figure 14: **4-way regime confusion matrix** (PolicyGRID, `smart_building_rich`). Cells show fraction of true-regime steps assigned to each predicted regime. *Base (98.7%) and Full (90.2%) are detected reliably; Occ-only is the hardest (59.4%), most often confused with Full because occupancy and window-open share thermal effects that the joint posterior cannot fully disambiguate.*

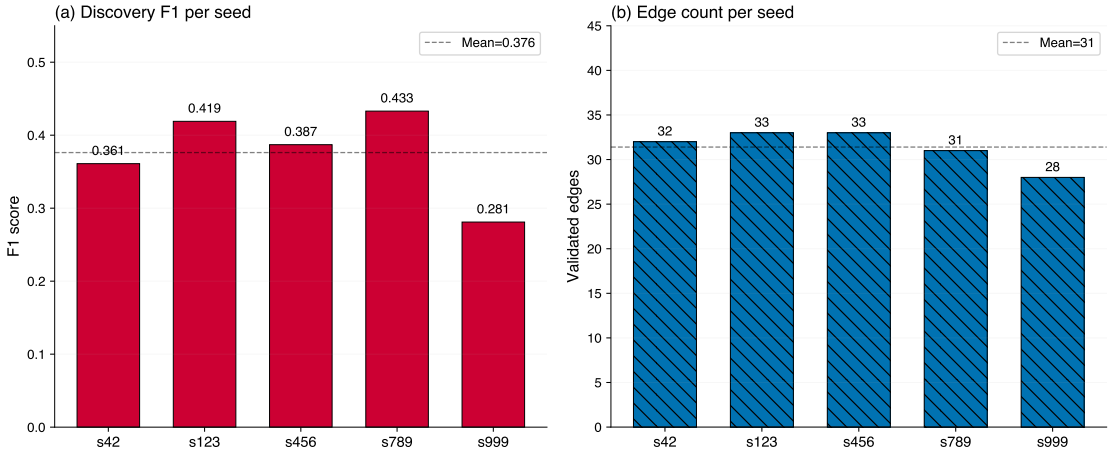


Figure 15: **Per-seed discovery on** `smart_building_rich` (5 seeds: 42, 123, 456, 789, 999). Left: F1 per seed (mean = 0.376); right: validated edge count per seed (mean = 31). *discovery is stable across seeds with one mild outlier (seed 999, F1=0.281, 28 edges); the 21% closed-loop result is not driven by a single lucky run.*

K Extended Related Work

The compressed Section 2 covers the load-bearing positioning of PolicyGRID against the closest prior threads (active causal discovery, causal control, building energy management). This appendix expands each thread with the citations and detail required for a comprehensive comparison.

K.1 Causal Discovery: From Passive to Interventional

Causal discovery has matured from constraint-based methods that recover structure from observational data alone [Spirtes et al., 2000, Colombo et al., 2012] through score-based and continuous-optimization approaches [Kalainathan et al., 2022, Zheng et al., 2020] to LLM-augmented frameworks that inject language priors into the discovery process [Kıcıman et al., 2023, Khatibi et al., 2024].

Each generation addresses a different identifiability bottleneck: latent confounders, acyclicity enforcement, orientation ambiguity. They share a fundamental limitation: they treat the data distribution as fixed — the graph is learned once from a static dataset and deployed without ever being tested against the physics of the environment. This is problematic in any setting where the controller's own actions reshape the data distribution, because edges learned from passive observation may not survive intervention [Glymour et al., 2019].

Active causal discovery addresses this by selecting interventions to resolve structural ambiguity that observational data cannot. Hauser and Bühlmann [2012] prove that interventional data strictly refines the Markov equivalence class, and introduce the GIES algorithm for joint observational-interventional learning. Squires et al. [2020] extend this to unknown intervention targets. Toth et al. [2022] propose ABCI, a fully Bayesian framework that designs maximally informative experiments without requiring a fully specified graph. Brouillard et al. [2020] introduce DCDI, a continuous-optimization approach that handles both interventional data and cyclic graphs — particularly relevant to physical systems where feedback loops are common. Zhang et al. [2023] provide theoretical guarantees for active intervention selection in linear causal models. CAASL [Annadani et al., 2024] amortizes the intervention design process via a transformer-based RL policy, achieving zero-shot generalization to unseen graphs at test time. DODO [Gregorini et al., 2025] is the closest structural parallel to our Phase 2 pipeline: it screens edges with a t-test on observation-vs-intervention samples and prunes indirect effects via Fisher z-test partial correlation. We retain that two-stage core but extend it in five ways that are load-bearing for physical deployment: (1) Multi-source hypothesis fusion: we union proposals from PC, SAM, VARLiNGAM, and an LLM, ranked by inter-method disagreement, rather than DODO's single-source candidate set. (2) Iterative scheduling: ten rounds of screening-then-confirmatory tests with discovery re-fit on intervention-augmented data each round, versus DODO's uniform single-pass budget allocation. (3) Non-intervenable rescue: a back-door-adjusted partial-correlation step retains edges from exogenous and latent variables, where DODO assumes causal sufficiency. (4) Cyclic-graph support under equilibrium SCM semantics, required for physical systems with feedback, where DODO restricts to DAGs. (5) Downstream evaluation: the validated graph drives policy and shift detection on a real CPS testbed, where DODO terminates at structure recovery on synthetic Erdős–Rényi graphs. These methods advance the theory and practice of efficient intervention selection, but they optimize for *structural information gain*, not for a downstream control objective. The interventions they select are designed to learn the graph, not to operate the system. This separation means the intervention budget is spent on pure exploration, with no simultaneous exploitation of the discovered structure for control.

Huang et al. [2020] extend causal discovery to heterogeneous and nonstationary data, showing that distributional changes across contexts can be exploited for identifiability — a theoretical foundation for our multi-regime setting where coefficient shifts across regimes provide the detection signal.

A more recent thread considers how causal structure changes across regimes. Zhao et al. [2025] introduce the meta-causal graph, a representation encoding how causal relationships shift across latent environmental states, and train a curiosity-driven agent to discover regime-specific subgraphs. DyCAST [Cheng et al., 2025] models temporal evolution of causal structure via neural ODEs on the graph manifold, recovering time-varying graphs from non-stationary time series. FANTOM [Rahmani and Frossard, 2025] combines conditional normalizing flows for regime detection with Bayesian structure learning, providing identifiability guarantees for multi-regime settings. These represent genuine progress on the non-stationarity problem, but each leaves part of the loop open: FANTOM and DyCAST operate passively (no interventions), while Zhao et al. evaluate in simulation only with no downstream policy. None of these methods validates discovered edges through physical intervention, and none uses the validated structure for multi-objective control.

K.2 Causal Structure for Control

A parallel line of work incorporates causal structure into control and decision-making [Zeng et al., 2025], but starts from a different assumption: the causal graph is *given*. Invariant causal prediction [Peters et al., 2016] exploits the invariance of causal mechanisms across experimental settings to identify stable predictive relationships, enabling policies that transfer across environments. Causal bandits [Lattimore et al., 2016] formalize intervention selection under a known graph, proving that causal structure yields provably lower regret than purely observational approaches. Buesing et al. [2019] use structural causal models to generate counterfactual trajectories from logged off-policy

data, improving policy search in POMDPs. Ding et al. [2022] augment goal-conditioned RL with learned causal graphs as latent variables, alternating between graph inference and policy learning, but the graph is inferred once and held fixed at deployment. Causal Bayesian Optimization [Aglietti et al., 2020] uses causal structure to optimize a black-box objective by reasoning about interventional effects — sharing our goal of causal-gradient-based optimization, but in a static bandit setting rather than a dynamic physical system with regime shifts.

Closest to our work are two recent methods that attempt to close the gap between causal learning and control. Cai et al. [2026] interleave causal structure learning and policy optimization in an online framework: interventions during exploration simultaneously discover structure and accumulate reward. This is an important step toward unifying the two objectives. However, their framework assumes a *single static causal structure* throughout deployment — it has no mechanism for detecting when the active operating context changes and no mechanism for resolving regime ambiguity through targeted online intervention. C-MBPO [Caron et al., 2025] integrates causal learning into model-based RL by fitting a local structural causal model (SCM) of state-reward dynamics and generating counterfactual on-policy transitions, demonstrating robustness to distributional shifts that affect spurious correlations. C-MBPO advances causal robustness in RL but operates within a single environment regime and does not perform physical edge validation; it learns the SCM from observational rollouts rather than testing edges through targeted do-operator interventions.

K.3 World Models

We note the distinction from *latent dynamics* world models [Ha and Schmidhuber, 2018, Hafner et al., 2020, 2025], which learn compressed state representations for planning via imagination. Yao et al. [2024] bridge causal representation learning and dynamical systems, but their learned representations remain latent rather than variable-level. A third line synthesizes neurosymbolic finite-state-machine world models that decompose dynamics into per-mode neural primitives stitched by symbolic transition predicates [Hernandez Cano et al., 2025]; these are explicit but non-causal, partitioning state space geometrically (e.g., via mutual-information-regularized clustering) rather than along causal-mechanism lines, and do not address intervention validation or regime-shift detection. Our validated SCM is an explicit, variable-level model — each edge names a physical relationship rather than encoding a latent dimension. This trades representational generality for interpretability and do-calculus reasoning (Section 3.1), critical when the controller must explain its decisions and when edge-level validation against physical measurements is the verification mechanism. Richens et al. [2025] argue general agents require world models for OOD generalization. Ceriscioli and Mohan [2025] prove that agents robust to distribution shifts *must* have learned causal world models, even under mediation — providing a formal justification for the SCM-based approach over purely predictive alternatives. In physical systems, correlational models [Kleissl and Agarwal, 2010, Kathirgamanathan et al., 2021] collapse under intervention because correlation does not imply causation [Glymour et al., 2019].

The gap that emerges across active discovery and causal control is structural: active discovery methods design interventions to *learn* but do not *control*, and causal control methods *optimize* but assume the graph is known or static. No existing method simultaneously uses control actions as causal experiments, physically validates the resulting edges before acting on them, detects which latent operating context is active under distributional shift, and verifies the active regime through targeted actuator probes interpreted via the validated SCM.

K.4 Data-Driven Building Control

Building energy management provides both our motivation and evaluation testbed. Data-driven control is the dominant paradigm [Kathirgamanathan et al., 2021], but only 23% of RL-for-HVAC studies deploy on real buildings [Al Sayed et al., 2024], citing distribution shift as a fundamental barrier. Causal methods have appeared in building fault detection [Zhang et al., 2022] and design [Chen et al., 2022], but exclusively as offline, one-time analysis tools — the graph is never validated against the physical system or adapted to regime changes. PolicyGRID bridges two largely independent research communities: causal discovery under intervention, which develops identifiability theory and experimental design but rarely closes the control loop, and multi-objective building control, which optimizes energy–comfort tradeoffs but treats the system model as a black box.

K.5 Positioning Matrix

Table 35 consolidates the thread-by-thread comparison along five desiderata for a causal control system operating under regime change. No prior method addresses all five.

Table 35: **Positioning of PolicyGRID relative to prior work.** Columns represent desirable properties for a causal control system operating under regime change. No prior method addresses all five. † Row groups GIES [Hauser and Bühlmann, 2012], CAASL [Annadani et al., 2024], and DCDI [Brouillard et al., 2020]; cited individually in Appendix K.1. *To our knowledge, PolicyGRID is the only method to combine interventional discovery, regime detection, edge validation, multi-objective control, and physical evaluation in a single closed loop.*

Method	Interventional data	Regime detection	Edge validation	Multi-objective control	Physical evaluation
PC / FCI [Spirtes et al., 2000]	x	x	x	x	x
GIES / CAASL / DCDI†	✓	x	x	x	x
FANTOM [Rahmani and Frossard, 2025]	x	✓	x	x	x
Zhao et al. [Zhao et al., 2025]	✓	✓	x	x	x
Cai et al. [Cai et al., 2026]	✓	x	✓	x	x
C-MBPO [Caron et al., 2025]	x	x	x	✓	x
DDPC [Coulson et al., 2019]	x	x	x	✓	✓
PolicyGRID	✓	✓	✓	✓	✓

L Supporting Theory

L.1 Proof of coefficient deconfounding (Section 3.3)

Proof. Under the data-generating process in Section 3.3, the population coefficient from regressing Y on A alone (omitting C) is

$$\beta_{\text{obs}} = \frac{\text{Cov}(A, Y)}{\text{Var}(A)} = \frac{\text{Cov}(A,\ \beta A + \gamma C + \epsilon)}{\text{Var}(A)} = \beta + \gamma\, \frac{\text{Cov}(A, C)}{\text{Var}(A)},$$

where the last step uses $\mathbb{E}[\epsilon \mid A, C] = 0$ and hence $\text{Cov}(A, \epsilon) = 0$.

Under an interventional design in which A is drawn from a nondegenerate distribution q independently of C, $\text{Cov}(A, C) = 0$ and $\text{Var}_q(A) > 0$. The same formula then gives $\beta_{\text{obs}} = \beta$. □

L.2 Proof of policy regret bound (Section 3.3)

Proof. Compactness and continuity guarantee that both maxima exist. Then

$$\begin{aligned} F_\varepsilon(\mathbf{a}^*) &\le \hat{F}_\varepsilon(\mathbf{a}^*) + \delta && \text{(uniform approximation)} \\ &\le \hat{F}_\varepsilon(\hat{\mathbf{a}}) + \delta && (\hat{\mathbf{a}} \text{ maximizes } \hat{F}_\varepsilon) \\ &\le F_\varepsilon(\hat{\mathbf{a}}) + 2\delta && \text{(uniform approximation again).} \end{aligned}$$

Rearranging gives $F_\varepsilon(\mathbf{a}^*) - F_\varepsilon(\hat{\mathbf{a}}) \le 2\delta$. For the approximate-optimizer variant, replace the second line with $\hat{F}_\varepsilon(\hat{\mathbf{a}}) \ge \hat{F}_\varepsilon(\mathbf{a}^*) - \eta$, yielding the additional η term. □

L.3 Proof of regime log-odds separation (Section 3.3)

Proof. From Eqs. 8–9 (ignoring the forgetting step in Eq. 10),

$$w_t^{(r)} \propto \ell_t^{(r)} \cdot w_{t-1}^{(r)} = \exp\!\big(-\sigma\, \|\mathbf{y}_t - \hat{\mathbf{y}}_t^{(r)}\|_1\big) \cdot w_{t-1}^{(r)}.$$

Taking the ratio for the true regime r^* vs. an alternative r and applying $\log$:

$$\begin{aligned}\log \frac{w_t^{(r^*)}}{w_t^{(r)}} &= \log \frac{\ell_t^{(r^*)} \cdot w_{t-1}^{(r^*)}}{\ell_t^{(r)} \cdot w_{t-1}^{(r)}} \\ &= \log \frac{w_{t-1}^{(r^*)}}{w_{t-1}^{(r)}} + \log \frac{\ell_t^{(r^*)}}{\ell_t^{(r)}} \\ &= \log \frac{w_{t-1}^{(r^*)}}{w_{t-1}^{(r)}} + \sigma\Big(\|\mathbf{y}_t - \hat{\mathbf{y}}_t^{(r)}\|_1 - \|\mathbf{y}_t - \hat{\mathbf{y}}_t^{(r^*)}\|_1\Big).\end{aligned}$$

The increment $\sigma(\|\mathbf{y}_t - \hat{\mathbf{y}}_t^{(r)}\|_1 - \|\mathbf{y}_t - \hat{\mathbf{y}}_t^{(r^*)}\|_1)$ is positive whenever the correct regime model predicts $\mathbf{y}_t$ more accurately than the alternative. The forgetting step in Eq. 10 subsequently shrinks the log-odds toward zero (uniform prior), so net accumulation requires the prediction-error difference to outpace the forgetting rate on average. □

L.4 Fixed-Pool Version-Space Monotonicity

For a fixed candidate pool E, define the version space after round r as the set of subgraphs consistent with all validation constraints accumulated through that round:

$$\mathcal{V}_r(E) = \big\{G \subseteq E : G \text{ satisfies all validation constraints through round } r\big\}.$$

Since each round only adds constraints, $\mathcal{V}_{r+1}(E) \subseteq \mathcal{V}_r(E)$.

Remark L.1. In PolicyGRID, the candidate pool itself may expand after discovery is re-run on intervention-augmented data, so monotonicity applies to the constraint set for a fixed pool, not necessarily to the raw number of candidate edges. Formally, if the pool at round r is E_r with $E_r \subseteq E_{r+1}$, then $\mathcal{V}_{r+1}(E_r) \subseteq \mathcal{V}_r(E_r)$ still holds, but $|\mathcal{V}_{r+1}(E_{r+1})|$ may exceed $|\mathcal{V}_r(E_r)|$ because the expanded pool admits new graphs. This monotonicity is with respect to the pipeline's accepted constraints; finite-sample testing error can still remove the true graph from $\mathcal{V}_r$, which is why repeated tests and FDR correction are used.

L.5 Finite-Sample Thresholding for a Nonzero Effect

Proposition L.2 (Finite-sample thresholding)**.** *Suppose the true total effect satisfies $|\Delta_{ij}^*(u)| \geq \tau > 0$ and per-trial intervention measurements are independent and sub-Gaussian with variance proxy σ^2. Thresholding at $\tau/2$ recovers the correct sign of relevance with probability at least $1-\delta$ when the number of intervention samples per edge satisfies*

$$n_{ij} \;\geq\; \frac{8\sigma^2}{\tau^2}\,\log\frac{2}{\delta}. \tag{12}$$

Proof. By sub-Gaussian concentration for the sample mean,

$$\Pr\big(|\hat{\Delta}_{ij} - \Delta_{ij}^*| > t\big) \;\leq\; 2\exp\Big(-\frac{n_{ij}\,t^2}{2\sigma^2}\Big).$$

Taking $t = \tau/2$ gives

$$\Pr\big(|\hat{\Delta}_{ij} - \Delta_{ij}^*| > \tau/2\big) \;\leq\; 2\exp\Big(-\frac{n_{ij}\,\tau^2}{8\sigma^2}\Big).$$

Setting the right side equal to δ and solving for n_{ij} yields the bound. When the deviation is at most $\tau/2$ and $|\Delta_{ij}^*| \geq \tau$, the empirical estimate has the correct sign and magnitude at least $\tau/2$. □

This bound explains the qualitative dependence on noise and effect size. The choice $K{=}10$ in the paper is empirically validated in the experiments; this proposition provides the theoretical scaling but does not pin the constant without domain-specific values of σ and τ.

L.6 LLM as Stochastic Proposal Source

The LLM is modeled as a stochastic map $\mathcal{L}_\theta \colon H_t \mapsto E_t^{(\mathrm{LLM})}$ where $E_t^{(\mathrm{LLM})} \sim P_\theta(\cdot \mid H_t)$ is a random edge set. The candidate pool is $\mathcal{E}_0 = \bigcup_{m=1}^{M} E^{(m)}$ (a set union, not a mixture), and the LLM is one contributor alongside PC, SAM, and VARLiNGAM.

Proposition L.3 (Union proposal coverage)**.** *Let $\mathcal{E}_0 = \bigcup_{m=1}^{M} E^{(m)}$, and define $p_m(e) = P(e \in E^{(m)} \mid H_t)$. Then, without any independence assumption,*

$$P(e \in \mathcal{E}_0 \mid H_t) \;\geq\; \max_m \, p_m(e).$$

In particular, if some method m satisfies $p_m(e) \geq \alpha_m$, then $P(e \in \mathcal{E}_0 \mid H_t) \geq \alpha_m$. If, additionally, each method m satisfies $p_m(e) \geq \alpha_m$ and the events $\{e \in E^{(m)}\}$ are conditionally independent given H_t, then

$$P(e \in \mathcal{E}_0 \mid H_t) \;=\; 1 - \prod_{m=1}^{M} (1 - p_m(e)) \;\geq\; 1 - \prod_{m=1}^{M} (1 - \alpha_m).$$

Proof. The first bound follows from $\mathcal{E}_0 \supseteq E^{(m)}$ for every m. For the independence case, $P(e \notin \mathcal{E}_0 \mid H_t) = P\big(\bigcap_m \{e \notin E^{(m)}\} \mid H_t\big) = \prod_m P(e \notin E^{(m)} \mid H_t) = \prod_m (1 - p_m(e))$. □

The guarantee rests on the union construction and downstream validation. Adding the LLM as a fourth proposal source can only increase edge coverage, since $P(e \in E_{\mathrm{old}} \cup E_{\mathrm{LLM}}) \geq P(e \in E_{\mathrm{old}})$ regardless of dependence. Wrong edges are subjected to the same validation test and are empirically filtered in the wrong-prior ablation (Section E, Table 22), which confirms that all 5 injected false edges are rejected.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state three quantitative claims, all verified in Section 5: (1) 21% closed-loop energy reduction at ε=1.0 on `smart_building_rich` vs. PolicyGRID-O, with satisfaction 1.4 points lower (6.3–6.9 kWh less on each of three weather days, 15/15 seed-by-weather runs; one fixed graph per method, seeds varying only simulator noise; Table 1, Appendix D.3); (2) 4-way latent-regime accuracy $80.8 \pm 1.1\%$ vs. $61.2 \pm 0.5\%$ (paired t=32.5, $p<10^{-5}$, d=14.5, 5/5 favor; per-discovery-seed $84.0 \pm 8.6\%$; Appendix D.2); (3) F1=0.400 on the live physical testbed via 21 do-operator interventions under a hardware-adapted validation gate (Appendix B.4), with target-responsive Fan gradient $0.0 \rightarrow 0.55$ vs. flat $\approx$0.31 for PolicyGRID-O. The introduction also states the scope of these claims: the control gain comes from validation pruning false-positive edges, while the monitor contributes detection and model-based attribution, with no measurable control benefit from regime-conditioned switching and no classification advantage over a feature-matched supervised classifier (Section 4.3, Appendix E.10); evidence is limited to variable-level building-control systems, and transfer to robotics, industrial process control, and multi-zone commercial buildings is untested (Section 7). Row A (Table 1) localizes leverage to candidate-set selection, not coefficient refinement (0/15 favor intervention-augmented coefficients). Operating-point conditioning (ε=1.0, in-distribution split), the OOD failure mode, and an open question on F1/SHD vs. downstream utility are surfaced in Sections 5.3, 7, and 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 is a dedicated Limitations section. Future-work directions are integrated into the same section. It covers: (1) scale ($O(|\mathbf{A}|\cdot|\mathbf{V}|)$ candidate-edge tests per seed, $\sim$50-variable practical ceiling) and evaluation within one problem family (building control), with transfer to robotics, industrial process control, and multi-zone commercial buildings untested; (2) parametric quadratic SCM and coefficient-only shift detection among known regimes, with no none-of-the-above detector or automatic re-discovery trigger for newly appearing pathways; (3) LLM dependency in three advisory roles (full declaration in Item 16) but every candidate passes interventional testing and posterior updates are deterministic from the SCM, with passive-only fallback matching full accuracy at steady state (Appendix D.2); (4) no distribution-free sample-complexity bound for the validation test (its error rates are characterized under a Gaussian working model in Appendix B.4), and validating actions are not screened for safety or cost; (5) physical F1 is computed against a heuristic 14-edge expert-proposed graph rather than system-spec source. The OOD imbalance failure mode (SAM leads on the 81%-Base split, PolicyGRID's lead returns only in-distribution) is surfaced in Section 5.3; Section 6 raises an open empirical question on F1/SHD vs. downstream utility, supported by the Row A ablation (Section 5.2, Table 1).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.

- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 3.3 states two formal results, with full proofs in Appendix L. The coefficient-bias result (OMITTED-CONTEXT BIAS UNDER ACTION–CONTEXT CONFOUNDING) assumes $\mathbb{E}[A^2], \mathbb{E}[C^2], \mathbb{E}[Y^2] < \infty$, $\mathrm{Var}(A) > 0$, and the data-generating process $Y = \beta A + \gamma C + \epsilon$ with $\mathbb{E}[\epsilon \mid A, C] = 0$; under OLS, $\beta_{\mathrm{obs}} = \beta + \gamma\,\mathrm{Cov}(A, C)/\mathrm{Var}(A)$, and interventional assignment of A independent of C vanishes the bias term. The policy-regret result (REGRET FROM SCM ERROR) assumes compact $\mathcal{A}$, continuity of the true and SCM-predicted scalarized objectives, uniform δ-approximation of the latter, and an η-approximate maximizer; it concludes regret $\leq 2\delta + \eta$. Section 3.3 reconciles the coefficient-bias result with Section 5.2's Row A: under the deployed Ridge SEM, shrinkage absorbs the OLS design-effect bias, and the regret bound's overall-SCM-error term is dominated by topology errors regardless of coefficient quality.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: Section 5 and Appendix B specify all hyperparameters, data splits (10k training rows per simulator; fresh 1440-step evaluation episodes for sim closed-loop and paired monitoring; 9-hour physical episode with 7 manually-induced regime transitions), and protocols. Algorithm 1 formalizes the closed loop; Sections 4.1–4.3 detail discovery, SCM-fitting, and monitoring. Appendix C (Tables 7, 8) documents the 11 discovery and 7 policy baselines with implementations and key hyperparameters. The physical testbed protocol — 21 do-operator interventions (3×7 probes, 22–31 minute holds calibrated in Appendix Table 4, 15-minute washout) — is in Section 5.3 and Appendix A.2. LLM prompts are reproduced verbatim in Appendix H; compute and API cost in Appendix G (Table 33). Raw closed-loop logs (`results_{cold,warm,mild}.csv`) and a reproduction script (regenerating the paired t, Cohen's d, 95% CI, and leave-one-seed-out statistics) are released with the code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and data are released at `https://github.com/taqiyaehsan/act-validate-adapt`: the full PolicyGRID framework, the five EnergyPlus-calibrated JavaScript simulators and their ground-truth graphs, all 11 discovery and 7 policy baselines, the physical-testbed sensor and actuator interface with its per-edge validation evidence, LLM prompts with raw responses and the edge-proposal cache (so LLM proposals replay without an API key), per-seed artifacts and exact seeds, and reproduction scripts. The paper also provides sufficient algorithmic detail for independent reproduction: Algorithm 1 specifies

the full closed loop; Sections 4.1–4.3 detail discovery, SCM-fitting, policy optimization, and regime monitoring; Appendix B documents protocols, hyperparameters, and seeds; Appendix C (Tables 7, 8) documents every baseline with implementation library and key hyperparameters; Appendix H reproduces every LLM prompt verbatim; Appendix G reports compute and API cost. Ground-truth DAGs are derivable from simulator source (5 JS sims), known physical relationships (ASHRAE), or the expert-proposed 14-edge graph in Appendix Table 3.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All settings are in Section 5 and Appendix B. Training: 10k observational samples per simulator (90% steady-state setpoint + 5% min + 5% max actuator levels; sensitivity in Appendix E.2). Evaluation: fresh 1440-step (24-hour) simulation episodes for closed-loop and paired monitoring; 9-hour physical episode with 7 manually-induced regime transitions. Discovery: K=10 tests/edge in 3+7 phases, α_{test}=0.05, BH FDR α_{FDR}=0.10 ($\geq$ 20 edges), τ_{eff}=0.1, $I_{\max}$=10. Rescue/pruning: τ_r=0.5, α_{prune}=0.01. SCM: quadratic Ridge α_{ridge}=0.5 (5-fold CV), direction floor $\beta_{\min}$=10^{-3}. Monitoring: sim σ=1.5, η=0.15; physical σ=0.5, η=0.2 (Appendix Table 12 caption); active-probe trigger $\max_r w_t^{(r)}$>0.7 for $\geq$ 120 steps applied symmetrically across regimes, probe horizon H=30. Policy: $\varepsilon \in \{0.5, 0.8, 1.0, 1.5, 2.0\}$, α=0.35, $w_{\min}$=0.3. Seeds: {42, 123, 456, 789, 999}; weather: {cold, warm, mild}. Sensitivity in Appendix F (monitoring 4×5 $\sigma-\eta$ grid varies 2.8 pp, Table 32; Table 31 one-at-a-time for discovery, rescue, pruning, Ridge); discovery F1 varies 4.1 pp under a 3^3 ($\tau_{\text{eff}}, \alpha_{\text{test}}, \alpha_{\text{FDR}}$) grid (Table 31). The physical deployment adapts the validation gate to hardware time cost (K=7 per actuator, α_{test}=0.10, BH-FDR 0.10 when $\geq$10 edges are tested, no sign-consistency stage; Appendix B.4).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Mean ± 1 std (sample, *ddof*=1) across 5 seeds (discovery, monitoring) or 5 seeds $\times$ 3 weather $=$ 15 paired conditions (closed-loop); seeds (42, 123, 456, 789, 999) and weather (cold, warm, mild) in Appendix B. **Closed-loop** (Section 5.2, Table 1): the 15 seed-by-weather conditions use one fixed graph per method, and the seed varies only simulator noise, so they are not independent replicates; we therefore report the gap per weather day (6.94/6.26/6.32 kWh cold/warm/mild), its consistency (15/15 favor the validated graph vs. PolicyGRID-O; 0/15 favor intervention-augmented coefficient training, Row A vs. Row B), and the satisfaction cost (1.4 points), with the paired statistics and this caveat in Appendix D.3. **Monitoring** (Section 5.1): paired Δ=+19.5 pp vs. PolicyGRID-O (t=32.5, $p<10^{-5}$, d=14.5, 5/5 favor); vs. IID Δ=+22.9 pp (t=30.0, $p<10^{-5}$, 5/5 favor). Per-discovery-seed std $84.0 \pm 8.6\%$ (Appendix D.2). **Robustness** (final paragraph of Section 5.3, Appendix E): intervention-ordering F1 ± 1 std over seeds (Appendix Table 19); LLM wrong-prior 5/5 injected adversarial edges rejected (Appendix Table 22). A reproduction script released with the code regenerates every closed-loop statistic from the raw logs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix G reports the platform (Intel Xeon W-2295, 18 cores, 251 GB RAM, no GPU), per-component wall-clock (Table 33), and totals: $\sim$120 CPU-hours simulation (5 seeds $\times$ 6 environments $\times$ discovery + benchmarks + monitoring + policy) + $\sim$55 wall-clock hours physical (34 h baseline collection + 12–15 h discovery + 5–6 h policy) + $\sim$36 CPU-hours Row A ablation (Table 1, Appendix E). LLM cost $\sim$\$0.50 ($\sim$30 hypothesis-generation + $\sim$150 intervention-design calls; GPT-3.5-turbo, $\sim$500k input + $\sim$100k output tokens). Pilot/exploratory experiments not in the paper added additional compute; all reported experiments fit within the totals above.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is building energy simulation plus a live physical office testbed (Govee/BME680 sensors and Kasa smart-plug actuators in an unoccupied room or one occupied by the research team) with no human-subjects data, no personal information, no biometric data, and no dual-use risk. Occupancy and window-state labels were logged by the research team (the authors and one acknowledged lab member) via a local Flask web app on the room's network. All experiments comply with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The headline 21% closed-loop energy reduction, at a 1.4-point satisfaction cost (Section 5.2), has direct positive environmental impact via more efficient HVAC and lighting control. Section 7 acknowledges single-domain (building-energy) evaluation and parametric SEM assumptions that constrain generalization. We see no direct path to negative societal applications: the framework controls climate and lighting, not human behavior or sensitive decisions. Indirect risks (misuse on safety-critical actuators outside the validated set) are mitigated by the validation pipeline, which refuses to act on causal hypotheses interventional testing has not confirmed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No pre-trained language models, image generators, or scraped datasets are released. The released artifacts (five EnergyPlus-calibrated building-energy simulators, the PolicyGRID framework, and a physical-testbed actuator interface) pose no high-misuse-risk profile. The validation pipeline is itself a safety mechanism: it refuses to act on causal hypotheses that have not passed interventional testing.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: ASHRAE Great Energy Predictor III dataset cited [ASHRAE, 2019] under its Kaggle research-use license; EnergyPlus calibration follows [Crawley et al., 2001]. Every baseline is cited at point of use in Section 2 and Appendix Tables 7, 8: PC [Spirtes et al., 2000], SAM [Kalainathan et al., 2022], VARLiNGAM [Hyvärinen et al., 2010], GIES [Hauser and Bühlmann, 2012], ICP [Peters et al., 2016], NOTEARS-I [Zheng et al., 2020], JCI [Mooij et al., 2020], Causal Bandits [Lattimore et al., 2016], ABCD [Toth et al., 2022], IID (PC on pooled obs+intv data, [Spirtes et al., 2000]); PID [Åström and Hägglund, 1995], DDPC_Behavioral [Coulson et al., 2019], DDPC_Subspace [Van Overschee and De Moor, 1996], DDPC_Neural (MLP+CEM, [Nagabandi et al., 2018]), DDPC_PETS [Chua et al., 2018], MPC_Linear [Mayne et al., 2000], C-MBPO [Caron et al., 2025]. Implementation libraries credited in Appendix Table 7: `causal-learn` v0.1 (MIT License), `cdt` v0.6 (MIT License), `lingam` v1.8 (MIT License), `causaldag` v0.1 (3-Clause BSD License). GPT-3.5-turbo used under OpenAI's Terms of Use. ISO 7730 PMV [Fanger, 1970, ASHRAE, 2023]. PolicyGRID and all six environments are original; ground-truth DAGs are extracted from simulator source (5 JS sims), known physical relationships (ASHRAE; Appendix Table 9 caption), and an expert-proposed 14-edge diagnostic graph (physical; Appendix Table 3) — all released alongside the framework under the MIT License.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets`

has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces (i) six evaluation environments — five EnergyPlus-calibrated JavaScript simulators (`smart_room`, `smart_room_noise`, `smart_room_hidden`, `open_window`, `smart_building_rich`) plus ASHRAE Great Energy Predictor III with random-forest interventional surrogate (Appendix B), documented in Appendix A with variable lists, ground-truth DAGs, and actuator/regime specs (Table 2); (ii) the PolicyGRID framework (Section 4, Algorithm 1); (iii) a physical testbed with documented hardware (Govee thermohygrometers, BME680 air-quality sensors, Kasa smart plugs), actuator calibration (Appendix Table 4), and a 14-edge expert-proposed diagnostic ground truth (Appendix Table 3). All assets are released at `https://github.com/taqiyaehsan/act-validate-adapt` with a README, reproduction commands, and the MIT License.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human-subjects research. The physical testbed (Section 5.3, Appendix A.2) was operated by the research team (the authors and one acknowledged lab member who helped collect data) with no study participants, no compensation transactions, and no personal or biometric data collection; only environmental state (temperature, humidity, CO_2, occupancy presence, window position) was logged.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human-subjects research. The physical testbed (Section 5.3, Appendix A.2) involves only environmental sensors and actuators operated by the research team, with no study participants, no biometric data, and no personal information collected; IRB review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: GPT-3.5-turbo is a core methodological component in three advisory roles: (1) one of four hypothesis generators (Section 4.1); (2) designer of intermediate-value intervention strategies during Phase 2 confirmatory tests (Section 4.1); (3) probe proposer at deployment when the regime posterior settles (Section 4.3). The LLM receives only variable names and domain context — never data, ground truth, simulator source, or other methods' results — and posterior updates are deterministic from the validated SCM, so correctness does not depend on LLM accuracy; the simulator source is private (not in any public corpus). Prompts are reproduced verbatim in Appendix H; model, temperature, top-p, max tokens, API call count, and total cost ($\sim$\$0.50) in Appendix G. The wrong-prior stress test (Appendix Table 22) confirms the validation pipeline rejects all 5 deliberately injected adversarial edges before they reach the policy. A passive-only fallback matches full-pipeline accuracy at steady state (Appendix D.2), so the LLM's contribution concentrates at regime transitions.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.