# WHERE SHOULD SCARCE ACTION LABELS GO? OFFLINE RL FROM ACTION-FREE BUILDING LOGS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Offline reinforcement learning assumes logged state–action–reward transitions, yet across the physical world the action channel can be missing entirely. Legacy building automation is a canonical case: years of sensor data exist, but the control commands that produced them need not have been kept. One property makes this regime tractable: in physical control the reward is a function of the observation, so the data are action-free yet reward-complete and policy improvement rather than imitation is possible. Deploying on real actuators still requires some labeled commands, which raises the practical question of where a small acquisition budget should be spent in the pipeline. We answer it with a matched benchmark over twelve building$\times$climate EnergyPlus configurations in which three placement strategies—imputing the missing column with an inverse model, grounding a latent policy learned from the action-free corpus, or supervising the latent during representation learning—are trained from identical labeled transitions and matched seeds. Where the labels enter matters most when they are scarce: at a $5\%$ budget the three recover $36\%$, $59\%$ and $89\%$ of an oracle's improvement, while offline RL on the labels alone falls below the logging policy. On this benchmark's logs the two late placements converge by a fifth of a year; the early one keeps a lead over grounding-only at every budget, though the full-year margin does not survive resampling by building archetype. Decomposed into physical quantities, the recovered improvement is almost entirely a reduction in thermal discomfort, with no consistent energy savings. The early placement is also the one that holds up when the conditions change: it stays ahead under a biased, rarely adjusted logging controller, under labels collected in seasonal campaigns, and across full-pipeline redraws, whereas whether the two late placements converge depends on the logging condition.

## 1 INTRODUCTION

Offline RL promises control improvement from logged data (Levine et al., 2020), but its data contract—$(s, a, r, s')$ tuples—is quietly violated across much of physical infrastructure. Building automation systems in legacy commercial buildings log zone temperatures, weather, occupancy and electricity meters at minute resolution for years, while the setpoint commands issued by the incumbent controller can be absent from the retained record—whether they are depends on how trend logging was configured at commissioning. Logs of that kind are the case we study: a large corpus of *observation-only* operational data that standard offline RL algorithms, which require an action column, cannot consume directly.

Learning from observation (LfO) offers a starting point. LAPO (Schmidt & Jiang, 2024) trains an inverse dynamics model (IDM) and a forward dynamics model (FDM) jointly so that a bottlenecked latent must explain each transition; VPT (Baker et al., 2022) trains a supervised IDM on a small labeled set to pseudo-label large unlabeled corpora. Both target video and game domains where rewards are as unavailable as actions, which confines those methods to imitation. We observe that physical control breaks this symmetry: energy use is metered and thermal comfort is a function of measured zone temperatures, so the reward is computable from the observation alone. The dataset is action-free but reward-complete, and policy improvement—not merely cloning—becomes possible in the latent action space.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A latent action is not something an actuator accepts, so improvement in a latent space is not the whole problem. Grounding the policy needs labeled commands, and those are the expensive part: obtaining them means instrumenting a live building. The question is therefore not whether actions can be dispensed with but where a small number of them should be spent. Three placements are available, and they use the labels for different purposes: fit an inverse model on them and impute the missing column, after which ordinary offline RL applies; learn the latent from the action-free corpus alone and spend the labels on a decoder that grounds it; or let them shape the latent during representation learning, as the latent-action literature does (Nikulin et al., 2025; Alles et al., 2025), and ground it afterward.

We find that this choice matters most when the labels are scarce. At a $5\%$ budget—eighteen labeled days' worth of commands—the three placements recover $36\%$, $59\%$ and $89\%$ of an oracle's improvement. Given a full year of labeled commands the same three recover $84\%$, $84\%$ and $93\%$: the two late placements have converged on each other, and the early one still leads grounding-only, by a margin that §6 qualifies, while its lead over imputation is gone. The supervised-latent placement follows prior latent-action work (Nikulin et al., 2025; Alles et al., 2025), and we do not claim the placement itself as a new method. The contribution is the matched comparison, and what it isolates is that the early placement's lead is the robust part: it survives full-pipeline redraws, a biased and rarely adjusted logging controller, and labels collected in seasonal campaigns, while how the two late placements relate—level by a fifth of a year on these logs—depends on the logging condition. We measured this with one offline RL algorithm and one latent-action architecture, so we state it as a property of this benchmark rather than of the problem in general. Such a comparison needs a domain where rewards are observation-computable, ground-truth actions exist for an oracle comparison, and the data-generating process is controllable. Building simulation satisfies all three, which is why we adopt it as a testbed rather than as an application.

Our contributions are:

- **A matched comparison of where the labels go.** Offline data that lack actions but carry rewards have been studied before (Chang et al., 2022; Neggatu et al., 2026), and sweeping the labeled fraction is not new either (Zheng et al., 2023); we claim novelty for neither. To our knowledge, no prior work directly compares these three strategies at a fixed label-acquisition budget using the same labeled transitions. The strategies differ in where the labels enter the pipeline and whether they are reused downstream. Our protocol on Sinergym/EnergyPlus (Jiménez-Raboso et al., 2021) makes that comparison paired across twelve configurations and four budgets: at every budget at which it is evaluated, each placement strategy draws the same labeled transitions from a single shared selection rule, with matched seeds (§3).

- **Where the labels go matters primarily while they are scarce.** The advantage of the latent route over imputation shrinks with the label budget and is gone by a fifth of a year. Supervised latent learning leads grounding-only at every budget and imputation at the two scarcest (§5).

- **The advantage of supervising early is largest where labels are scarcest and narrows monotonically as they become plentiful**—$+37$, $+31$, $+15$ and $+9$ recovery points at $1\%$, $5\%$, $20\%$ and a full year, with one auxiliary-loss coefficient throughout. The grounding-only arm has $1.9$ times the across-configuration standard deviation of supervised latent learning at the $5\%$ budget. Of four candidate correlates of the advantage, none is separable on twelve configurations (§5).

- **Physical decomposition of the recovered improvement.** Decomposed into annual energy and comfort-violation degree-hours, the recovered improvement is primarily a comfort improvement rather than an energy-saving effect. This pattern reflects the objective rather than any method. The same decomposition separates the placements: at the $5\%$ budget, the two late-placement policies spend no more energy than the oracle yet incur far more discomfort on average (§5).

## 2 Related Work

**Offline RL, with and without action labels.** Offline RL learns from a fixed dataset by controlling how far the learned policy may depart from the behavior distribution: by penalizing out-of-distribution value estimates (Kumar et al., 2020), by constraining the policy toward the data (Fujimoto & Gu, 2021), or by never querying the critic off-distribution at all (Kostrikov et al., 2022). We adopt the last of these, because when the "actions" are themselves inferred quantities rather than logged commands, an algorithm that never evaluates the critic at unseen actions avoids compounding representation error with extrapolation error. This literature assumes throughout that the action column exists. Offline data without actions but with rewards have been studied directly: LAQ (Chang et al., 2022) learns value functions from $(s, s', r)$ tuples by Q-learning over latent actions, and Neggatu et al. (2026) formalize the same $(s, r, s')$ regime for offline-to-online RL. We claim no novelty for the setting. Both differ from us after the latent stage: LAQ pairs its value functions with domain-specific low-level controllers rather than learning a map to commands, and Neggatu et al. (2026) recover the action channel through online interaction, which is unavailable under the deployment constraints considered here. Once a decoder is required, the necessary action labels can be allocated to more than one stage of the pipeline, which motivates our comparison.

Closest to our mixed-label setting is SS-ORL (Zheng et al., 2023), which formalizes datasets mixing action-labeled and action-free trajectories and imputes the missing actions before running standard offline RL. Two differences matter here. First, SS-ORL assumes reward labels are present; we require none, because the reward is recomputable from observations—which is what makes the action the only missing modality. Second, that line of work reports task returns, whereas we ask how much is lost relative to an oracle with ground-truth actions, and under which conditions that loss grows. That difference is one of data preparation rather than of algorithm; we therefore evaluate supervised action imputation with the same offline RL algorithm at matched label budgets (§5), in SS-ORL's own form, where the offline RL dataset is the union of the true-labeled and the pseudo-labeled transitions rather than the pseudo-labeled ones alone. That choice matters at the larger budgets: including the true-labeled transitions in offline RL adds 12 points to the baseline at $20\%$ and 15 at a full year, where it makes the comparison a tie, though it changes little at $5\%$. MAHALO (Li et al., 2023) generalizes further, to datasets where rewards, actions, or both may be missing from part of the data, using pessimism over data-consistent hypotheses; our setting is narrower and the reward is never missing, because it is recomputed.

Closest to our empirical claim is LAWM (Alles et al., 2025), which learns a shared latent action space over a small action-conditioned dataset and a large action-free one and then does model-based offline RL in it, reporting strong control on the DeepMind Control Suite with an order of magnitude fewer action labels. Three things separate our study. LAWM assumes rewards in both datasets, whereas our premise is that they need never have been stored. LAWM uses model-based offline RL, whereas we use model-free IQL in the latent space and then ground the policy with a decoder. LAWM fixes the labeled fraction at $5\%$, whereas we evaluate four budgets spanning two orders of magnitude. LAWM reports the same average ordering in a different domain, using a model-based offline RL method and stored rewards: its inverse-dynamics baseline also trails its latent-action model on average, though not on every individual dataset.

LAWM supplies action supervision during latent learning rather than only afterward. Related latent-action work shows that, with action-correlated distractors, a few percent of action labels used that way substantially improves what the latent captures (Nikulin et al., 2025). We implement that placement in our own architecture—the same auxiliary head as our Stage 3 decoder, the same labeled subset, moved into Stage 1—and at the $5\%$ budget it achieves the highest recovery among the three strategies. Our benchmark compares all three placements across four label budgets, in a setting where the reward need never have been logged.

**Learning from observation and latent actions.** BCO (Torabi et al., 2018) infers actions with a supervised inverse model and clones them; ILPO (Edwards et al., 2019) imitates latent policies; APV (Seo et al., 2022) pre-trains representations from action-free video; VPT (Baker et al., 2022) uses a small labeled set to pseudo-label web-scale video. LAPO (Schmidt & Jiang, 2024) removes the need for an action-labeled inverse model by inferring latent actions from dynamics alone, and the idea has since been scaled to interactive world models (Bruce et al., 2024), to robotic manipulation (Ye et al., 2025), and to adaptable world models (Gao et al., 2025). These studies span video, games

and robot demonstrations, but all lack rewards as well as actions, which limits them to imitation or pre-training. Where a reward does accompany action-free data, that data can support improvement rather than cloning (Schmeckpeper et al., 2021; Wu et al., 2024; Song et al., 2024), but all three of those assume online interaction. Our setting differs structurally: in physical control the reward is a function of the observation, so offline policy improvement is available directly in the latent space, without environment interaction and without reward labels.

**Identifiability of latent actions.** A recent theoretical thread asks when latent actions are recoverable in principle: Lachapelle (2025) formalizes three desiderata for a latent action representation—determinism, disentanglement from the current state, and injectivity with respect to the ground-truth action—and proves that an entropy-regularized LAPO objective recovers representations satisfying them under continuity, injectivity and support conditions. Schur (2026) shows that action-free trajectories tagged with demonstrator identity make the latent actions and the dynamics identifiable when distinct demonstrators share dynamics and are sufficiently diverse. Our contribution is complementary and empirical. Rather than assuming a source of structure, we point to one intrinsic to physical control—how often each actuation axis is exercised—and report how recovery covaries with it across climates. Climate is chosen by us rather than observed, but it moves several things at once, so we treat this as a mechanism the data support rather than one we have isolated.

**RL for building control.** Prior work is predominantly online in simulation (Jiménez-Raboso et al., 2021) or offline from logs that do contain the control commands (Blad et al., 2022; Liu et al., 2022; Goldfeder & Sipple, 2023): B2RL, for instance, distributes zone setpoints as part of the recorded stream. To our knowledge, no offline building-control method targets the action-free regime; we use it as a testbed rather than as an application, for the reasons given in §1.

## 3 Problem Formulation

Thermal mass, equipment mode and delayed response make the instantaneous observation an imperfect state, so the natural formulation is a history-state MDP whose state is the $k$-step window $h_t = (o_{t-k+1:t})$. We adopt that only where it proved necessary: Stage 1's inverse and forward models condition on $h_t$ with $k$=4, while the critic, latent policy and decoder condition on the instantaneous $o_t$ (Table 3). That asymmetry is empirical rather than principled, and it means the control problem we solve is the partially observed one with a policy restricted to memoryless maps (§6). Every module's inputs are available at deployment time. The plant is a heating, ventilation and air-conditioning (HVAC) system, and the observation is $o_t = (x_t, w_t)$, where $x_t$ holds the *endogenous* states the forward model predicts (zone temperatures $T_{i,t}$, HVAC power $P_t$) and $w_t$ the rest of the observation (weather, occupancy, calendar, and any further measured channel we do not model as endogenous, so $w_t$ is not strictly exogenous; Appendix A). Actions $a_t$ (heating/cooling setpoints) are issued by an unknown behavior policy but never logged. We are given a large action-free corpus $\mathcal{D}_{\text{obs}} = \{(o_t, o_{t+1})\}_{t=1}^{N}$ and a separately collected labeled set $\mathcal{D}_{\text{lab}} = \{(o_t, a_t, o_{t+1})\}_{t=1}^{M}$, in which both blocks of $o_{t+1}$ are the ones logged at $t$+1. The ratio $M/N$ is a controlled variable of our study rather than an assumption: $N$ is one year of operation (35,040 transitions at the 900 s control step) and we report results across $M/N \in \{1\%, 5\%, 20\%, 100\%\}$, so that the cost of action-freeness is separated from the cost of a small labeling budget. Every method that consumes labels receives exactly the same labeled transitions at each budget, drawn evenly across the year rather than as a contiguous campaign (Appendix A). The reward is a known function of the observation, $r_t = r(o_{t+1})$; we use Sinergym's own `LinearReward` class with its registered hyperparameters ($\omega$=0.5, $\lambda_P$=$10^{-4}$, $\lambda_T$=1),

$$r_t = -\omega\lambda_P P_{t+1} - (1-\omega)\lambda_T \textstyle\sum_i \left[(T_{i,t+1} - T_t^{\max})_+ + (T_t^{\min} - T_{i,t+1})_+\right], \tag{1}$$

where $(x)_+ \equiv \max(x, 0)$, $P_{t+1}$ is the HVAC power in watts, and the sum runs over the controlled zones $i$. We evaluate the reward on the next observation because the effects of $a_t$ appear there. Comfort bands $[T_t^{\min}, T_t^{\max}]$ follow Sinergym's seasonal schedule. The goal is a deployable policy $\pi(a \mid o)$ that outperforms the behavior policy. Actions are clipped to the environment's action interval, so setpoints are always in range; we impose no ordering constraint between the heating and cooling setpoints; in the version we run (Sinergym v3.12.0, see the reproducibility statement) the building model resolves any inversion, as it does for the rule-based controllers we compare against.

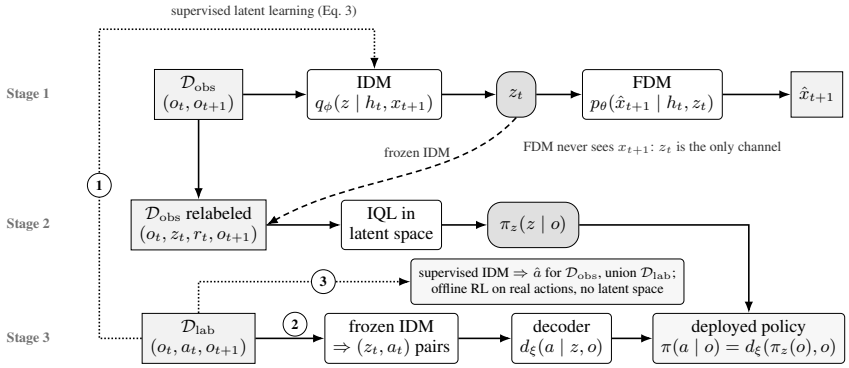



Figure 1: Pipeline and the three ways of spending the labeled set. In the grounding-only arm, **Stage 1** discovers latent actions from observation transitions alone; the forward model is denied $x_{t+1}$, so whatever the transition reveals about the intervention passes through the bottleneck $z_t$. **Stage 2** relabels the action-free corpus with those latents, attaches observation-computed rewards and runs offline RL. **Stage 3** grounds the result with a labeled set whose size is a controlled variable. Ground-truth actions never enter via $\mathcal{D}_{\text{obs}}$. The numbered paths mark the three strategies, which acquire the same labels and differ in how they use them: ① shaping Stage 1 and then grounding (Eq. 3, dotted); ② grounding only (solid, through Stage 3); ③ fitting an inverse model that fills $\mathcal{D}_{\text{obs}}$'s action column, after which offline RL runs on the union of that corpus and the true-labeled transitions and no latent space is used (dotted). Strategy ① therefore spends its labels twice, and ③ uses them both to fit the inverse model and as real transitions in offline RL; §4 states the estimand this implies.

**Benchmark protocol.** We instantiate five Sinergym buildings—5-zone variable-air-volume (VAV), medium office (15 controlled zones), a retail shop (5 zones), a warehouse (single controlled zone), and a 2-zone data center—under three climates (hot, cool, mixed). The main matched-budget analysis pools the four occupied-building archetypes, giving twelve building–climate configurations; the data center is retained for diagnostic analyses and is not included in the main aggregates. $\mathcal{D}_{\text{obs}}$ is generated by a seasonal rule-based controller (RBC) with Gaussian setpoint perturbations ($\sigma$=1.0 °C) as a stand-in for operator variability; the action column is withheld from all learning and retained only for evaluation. $\mathcal{D}_{\text{lab}}$ uses a wider perturbation ($\sigma$=2.5) modeling a commissioning period. Setpoint observations are removed from $o$ to prevent action leakage. We evaluate comfort on all controlled zones, which is stricter than Sinergym's per-building defaults. Those defaults evaluate a zone subset that does not match the control authority: a single zone in the 5-zone building and unconditioned plenums in the office.

**Choice of testbed.** Building control can be simulated at very different fidelities, and the choice is not neutral here. Lightweight, telemetry-calibrated simulators (Goldfeder & Sipple, 2023; Goldfeder et al., 2024) enable randomization over many buildings and would give far broader coverage than our 12 configurations. We nonetheless use first-principles whole-building simulation for three reasons. Our claims concern the identifiability of actions from dynamics, and we expect a learned reduced-order simulator to have smoother dynamics that an inverse model finds easier to invert—this is the reasoning behind our choice rather than a property we measured. The mechanisms we identify—deadband non-identifiability, fifteen zones served by one setpoint pair—are properties of realistic HVAC topology that device-schedule abstractions do not expose. And calibrating a data-driven simulator of the kind we would otherwise use (Goldfeder & Sipple, 2023) against a target building requires telemetry that includes the control commands, the very modality missing in the regime we study. This choice limits the breadth of configuration coverage (§6).

## 4 METHOD

Our pipeline (Figure 1) separates three questions that the action-free setting entangles: what the unobserved intervention was, what a better intervention would have been, and how to express it in the actuator's units.

For the grounding-only arm, the first is answered without supervision. An inverse model proposes a latent $z_t$ for each observed transition and a forward model must reproduce the transition from the past and $z_t$ alone. Denying the forward model access to $x_{t+1}$ is what makes the arrangement informative: any predictive content it needs about the future must pass through a low-dimensional bottleneck, and the intervention is the part of the transition that carries most of the short-horizon information about the controlled variables. The resulting latent space supports action-like control despite being learned without action labels. It is not a pure intervention representation: the bottleneck also absorbs exogenous innovation from weather and occupancy. We quantify this content in Appendix B; across configurations it is not significantly associated with the benefit of supervision (§5).

The second question is where our setting departs from prior learning-from-observation work. Because the reward is a function of the observation, the relabeled corpus $(o_t, z_t, r_t, o_{t+1})$ is a complete offline RL dataset in the latent space—the missing modality was the action, not the learning signal—so a conservative offline RL algorithm can return a policy better than the one that generated the logs rather than merely cloning it.

The third question requires supervision, and we treat the size of the labeled set as an experimental variable. Passing a labeled set through the frozen inverse model yields paired $(z_t, a_t)$ examples—a parallel corpus between the discovered latent space and the real actuator—from which a decoder is fit. In that arm, ground-truth actions enter only here; the other two arms spend them at Stage 1 and before Stage 2 (Figure 1).

**Stage 1: latent action discovery.** Following the LAPO template, an IDM $q_\phi(z_t \mid h_t, x_{t+1})$ infers a latent action from a history window $h_t = (x_{t-k+1:t}, w_{t-k+1:t})$ and the next endogenous state, while an FDM $p_\theta(\hat{x}_{t+1} \mid h_t, z_t)$ reconstructs $x_{t+1}$ without access to it: the bottlenecked latent ($\dim z{=}4$, Gaussian, KL-regularized) is the only channel for transition information. History ($k{=}4$: one hour at 900 s, forty minutes in the 600 s data center) is essential in thermally inert systems, where a setpoint change manifests over several steps.

The Stage 1 objective is the sum, over transitions, of a weighted squared reconstruction error on the next endogenous state and a KL term that limits what the bottleneck may carry,

$$\mathcal{L}_1 = \mathbb{E}\Big[\tfrac{1}{d_x}\textstyle\sum_j w_j\big(\hat{x}_{t+1,j} - x_{t+1,j}\big)^2 + \beta_{\mathrm{KL}}\, D_{\mathrm{KL}}\big(q_\phi(z_t \mid h_t, x_{t+1}) \,\|\, \mathcal{N}(0, I)\big)\Big], \tag{2}$$

with $\hat{x}_{t+1} = p_\theta(h_t, z_t)$, $z_t$ reparameterized from $q_\phi$, $d_x$ the number of endogenous channels, targets standardized per channel, and weights $w_j$ of mean one that favor zone temperatures threefold over the power channel. The Huber loss enters only through the physics residuals of Appendix G. We use $\beta_{\mathrm{KL}}{=}10^{-3}$; the main method sets the optional physics terms of Appendix G to zero, so Equation 2 is the whole objective, with the expectation over $(h_t, x_{t+1}) \sim \mathcal{D}_{\mathrm{obs}}$.

**Where the labels enter.** Stage 3 spends $\mathcal{D}_{\mathrm{lab}}$ after the latent is fixed; the alternative spends the same transitions inside Stage 1, adding an auxiliary head $g_\psi(z_t, o_t) \to a_t$—architecturally identical to the Stage 3 decoder—and an action-reconstruction term over the labeled subset:

$$\mathcal{L}_1^{\mathrm{sup}} = \mathcal{L}_1 + \lambda_{\mathrm{act}}\, \mathbb{E}_{(h_t, x_{t+1}, a_t) \sim \mathcal{D}_{\mathrm{lab}}^{(f)}}\big[\tfrac{1}{d_a}\|g_\psi(\mu_\phi(h_t, x_{t+1}), o_t) - \tilde{a}_t\|_2^2\big], \tag{3}$$

where $\mu_\phi$ is the inverse model's posterior mean—also what Stages 2 and 3 consume—$d_a$ the action dimensionality, $\mathcal{D}_{\mathrm{lab}}^{(f)}$ the budget-$f$ subset the decoder receives, and $\tilde{a}_t$ the action affine-normalized per dimension to $[-1, 1]$ over that subset's range. The Stage 3 decoder uses the same construction: a $\tanh$ output mapped dimension-wise onto the range observed in its labeled subset, expanded by $0.5°$C at each endpoint, after which the environment clips to the action bounds. This arm spends the labels twice, to shape the latent and then to ground it, at an unchanged acquisition budget; the estimand it identifies is therefore the marginal value of adding early supervision to a grounding pipeline, not an exclusive choice between two stages. Stages 2 and 3 are untouched and $\lambda_{\mathrm{act}}{=}0$ recovers the grounding-only pipeline exactly (coefficient selection: Appendix G).

**Stage 2: offline RL in latent space.** The frozen IDM relabels $\mathcal{D}_{\mathrm{obs}}$ into $\{(o_t, z_t, r_t, o_{t+1})\}$ with observation-derived rewards (Eq. 1). We run IQL (Kostrikov et al., 2022) with expectile $\tau{=}0.7$, advantage-weighted regression (AWR, $\eta{=}3$), twin critics, rewards standardized to zero mean and unit variance over the dataset and then scaled by $(1{-}\gamma)$, and batch advantage normalization; with no environment interaction available for model selection, we take the critic checkpoint with minimal TD-error EMA.

**Stage 3: grounding.** Passing $\mathcal{D}_{\text{lab}}$ through the frozen IDM yields parallel pairs $(z_t, a_t)$; a decoder $d_\xi(a \mid z, o)$ is fit by regression and the deployed policy is the composition $\pi(a \mid o) = d_\xi(\pi_z(o), o)$.

**Exposing time-varying reward parameters.** The comfort bounds $(T_t^{\min}, T_t^{\max})$ of Eq. 1 switch discontinuously with season, and a policy that must infer that schedule from raw calendar features smooths it, interpolating setpoints precisely at the switch. Supplying the current bounds as explicit policy and decoder inputs—an operator-known schedule, hence legitimate at deployment—largely removes the failure (Appendix A). We report this as a failure mode and implementation fix observed in our experiments.

## 5 EXPERIMENTS

**Setup.** All arms share the data of §3 and are scored by a full-year rollout under Eq. 1. Three acquire the same labeled transitions and differ in how they use them: **imputation** fits a supervised inverse model on $\mathcal{D}_{\text{lab}}$, fills in $\mathcal{D}_{\text{obs}}$'s action column and runs IQL on the union of the filled-in corpus and the true-labeled transitions, as Zheng et al. (2023) do; the **grounding-only** arm learns the latent from $\mathcal{D}_{\text{obs}}$ alone and uses the labels to ground it; **supervised latent learning** spends them inside Stage 1 (Eq. 3) and grounds afterward. We also evaluate **BCO** (Torabi et al., 2018), which clones the imputed actions rather than optimizing them, two **labeled-only** controls—a clone and the same offline RL stage on the labeled budget alone (Appendix J)—the **behavior policy** and an **oracle** IQL on ground-truth actions. Appendix B reports two further controls.

**Evaluating the pipeline across the grid.** We first evaluate the grounding-only pipeline across the building–climate configurations (Tables 5 and 7). At a full year of labels the grounding-only arm recovers $84.0\%$ of the oracle's improvement on average (median $92.8$, gain-pooled $79.9$; range $17.0$–$107.9$) and beats the logging policy everywhere; the hardest configuration is the office, which drives fifteen zones from one setpoint pair. Both labeled-only controls fall below the logging policy (the critic at $5\%$), so the learned policies are not regressions to an unperturbed controller.

**At small label budgets the latent route outperforms imputation, and the two are level by a fifth of a year (Table 1).** We first compare the two late placements. Both use the labels after the representation is fixed, but for different mappings: an inverse model that fills in $\mathcal{D}_{\text{obs}}$'s missing action column, or a decoder that grounds a latent learned from $\mathcal{D}_{\text{obs}}$ alone. Both draw the same labeled transitions at every budget (one shared selection rule in the code) and share observations, comfort-band inputs, reward parameters and optimization budget, so the comparison is paired run by run. The imputation baseline has weaker and stronger implementations, depending on whether the true-labeled transitions also enter offline RL, whether the pseudo-labels are constrained to the decoder's output range, and whether the inverse model is trained on the decoder's recipe (same epoch budget, same validation split above $1\%$, the range as a parameterization rather than a clip). That last rung makes the comparison one of placement; at $5\%$ it recovers $164$ points more than the third rung and $194$ more than the first (Appendix D).

The margin decreases with the label budget and reaches zero before the sweep ends: $+63.9$ points at $1\%$, $+22.2$ at $5\%$, $-5.5$ at $20\%$ and $+0.1$ at a full year, with bootstrap intervals excluding zero at the first two budgets and containing it at the last two—"level" and "converge" below mean no detected difference, not equivalence. The inverse model is trained only from the labeled transitions, whereas the latent route also learns transition structure from the full action-free corpus; that difference is worth tens of points while the labels are scarce and nothing once they are not. How the labels are spread matters as much: four seasonal campaigns of the same count cost every placement tens of points, and only the early one stays above the logging policy (Appendix G).

At the full-year budget, the imputation arm also receives more transitions: because $\mathcal{D}_{\text{lab}}$ is a separate rollout, merging it adds $1\%$ to the dataset at the smallest budget and $100\%$ at a full year, where that arm alone trains on two years of data. We read the full-year column as a tie for that reason, not as a placement result (§6).

**Physical decomposition of the recovered improvement.** Decomposing the shaped score into annual HVAC energy and comfort-violation degree-hours (Appendix B) shows that the improvement recovered over the logging policy is almost entirely a reduction in discomfort, with no consistent

energy savings. For the grounding-only arm at a full-year budget, energy use changes by $-7\%$ to $+15\%$ across the twelve configurations (median $+1\%$) while violation degree-hours fall by $3\%$ to $83\%$ (median $57\%$). This is a property of the objective rather than of any method: the oracle moves the same way, because the reward makes a zone-degree of discomfort far more expensive than a kilowatt-hour. Recovery would otherwise read as energy saved, which it mostly is not, and the ranking is a statement about this objective (Appendix E).

**The crossover is not a single budget.** Imputation leads the latent route in two of the twelve configurations at $5\%$, in eight at $20\%$ and in four at a full year, so the budget at which the two meet differs by configuration, and the benchmark does not identify a predictor of where it falls.

**At the $5\%$ budget, supervised latent learning achieves the highest recovery among the three strategies.** Both arms above spend the labels late, so the representation is settled before the labels are consulted; the latent-action literature spends them earlier (Nikulin et al., 2025; Alles et al., 2025). Implemented as Eq. 3, that placement achieves $89.4\%$ recovery, compared with $58.7\%$ for grounding-only and $36.5\%$ for imputation. It exceeds grounding-only by $30.7$ points, with interval $[21.7, 40.3]$, and imputation by $52.9$ points, $[35.0, 79.2]$, leading in all $12$ configurations against both. At a full year it keeps an edge over grounding-only ($+9.3$, $[1.7, 19.8]$, ahead in 9 of 12) and is level with imputation ($+9.4$, $[-4.4, 28.3]$, ahead in 6 of 12). One auxiliary-loss coefficient is used at every budget (Appendix G); the grid mean moves by $0.8$ points under the value the pilot ranks last. Recovery also varies less across configurations at $5\%$: 66–$104\%$, s.d. $12.4$, against 4–$89\%$, s.d. $23.0$.

Table 1: Matched-budget comparison, as a percentage of the oracle's improvement over the logging policy (higher is better; negative means worse than the logging policy), grouped by where the labels are spent. Bold marks the highest recovery among the three placements in each column; "CI" is the $95\%$ bootstrap interval; "raw / ahead" is the raw-return difference and the number of configurations in which the first arm leads. All budgets cover the same twelve occupied-building configurations, each of the three placement strategies is run three times per cell, and at each configuration, budget and seed the arms are trained from the same labeled transitions. All three placements are swept over all four budgets. The supervised-latent arm uses a single auxiliary-loss coefficient at every budget, the value with the highest pilot recovery at both of the budgets the pilot covered (Appendix G). The imputation row is the fourth rung of Appendix D; two imitation controls, BCO and a labeled-only clone, are worse than every row here at every budget (Appendix B). The lower blocks give paired differences, the raw-return difference and the count of configurations ahead out of twelve. Intervals resample configurations after averaging tier-A reruns; they do not capture variation from tier-B full-pipeline redraws (Table 2).

| Labels (fraction of one year) | 1% | 5% | 20% | 100% |
|---|---|---|---|---|
| Supervised latent (Stage 1) | **74.0** | **89.4** | **86.8** | **93.3** |
| Latent + decoder (Stage 3) | 36.9 | 58.7 | 71.6 | 84.0 |
| Imputation + IQL (inverse model) | −27.0 | 36.5 | 77.1 | 83.9 |
| *Paired difference, latent+dec. − imputation* | | | | |
| recovery [CI] | +63.9 [46.3, 84.5] | +22.2 [8.1, 42.2] | −5.5 [−13.8, 2.8] | +0.1 [−10.1, 10.8] |
| raw / ahead | +0.236 / 12 | +0.074 / 10 | −0.017 / 4 | +0.000 / 8 |
| *Paired difference, sup. latent − latent+dec.* | | | | |
| recovery [CI] | +37.2 [23.2, 52.5] | +30.7 [21.7, 40.3] | +15.2 [8.8, 23.9] | +9.3 [1.7, 19.8] |
| raw / ahead | +0.138 / 12 | +0.139 / 12 | +0.068 / 11 | +0.048 / 9 |

The physical outcomes differ across placements. At the $5\%$ budget the mean energy ratios relative to the oracle are $98\%$, $100\%$ and $106\%$ for imputation, grounding-only and supervised latent learning, and the mean discomfort ratios $188\%$, $160\%$ and $114\%$ (medians 189, 156, 103; Appendix B): the two late placements spend no more energy than the oracle and leave far more discomfort in place.

This pattern describes the observed policy behavior, not its mechanism. Of four candidate correlates of the benefit of supervision (Appendix C), the non-actionable content of the unsupervised latent is unrelated to it ($r = -0.16$); how far the auxiliary loss moved the latent ($r = +0.60$, permutation $p =$

$0.04$), the late decoder's held-out fit ($+0.51$) and its output spread ($+0.49$) all correlate with it, but none survives a correction for three looks and the three are not separable on twelve configurations.

**Uncertainty under tier-A reruns and tier-B full-pipeline redraws (Table 2).** Tier B redraws every component listed in Table 2, including the data-generation process and Stage 1, whereas tier A resamples the network initialization and minibatch order, the labeled subset being fixed by a deterministic rule. Stage 1 is retrained per seed only where the labels enter it, in the supervised-latent arm; doing so for the grounding-only arm as well moves it by at most $2.2$ points (Appendix F). In the two configurations evaluated under both protocols, the tier-B recoveries of the grounding-only arm fall outside the tier-A range in three of the four draws (Appendix F). We therefore treat the main recovery estimates as conditional on the particular data-generation and training realization rather than as properties of the method alone. The ordering is preserved across all four tier-B full-pipeline redraws, against the same imputation rung this table reports (Appendix F), and, below a full year, under an alternative definition of the Stage 2 transition that moves almost every absolute recovery in this paper, some by tens of points (Appendix H); on a biased, rarely adjusted log the early placement's lead survives but the late placements no longer converge (Appendix K).

## 6 Limitations

Our evaluation is in simulation; deployment on instrumented buildings is future work. The testbed has mechanistic fidelity but limited breadth (§3); lightweight calibrated simulators (Goldfeder et al., 2024) would test the same claims more broadly. The imputation arm is matched on the labels and on the recipe that consumes them, not on transitions consumed (§5); relabeling $\mathcal{D}_{\text{lab}}$ through the frozen inverse model and adding it to Stage 2 is an experiment we have not run, and we tested one inverse-model architecture, where matching the recipe was worth $164$ points at $5\%$ (Appendix D). Every log in the main grid is Gaussian-perturbed rule-based control; on one biased, rarely adjusted log the early placement's lead survives but the late placements no longer converge (Appendix K). The ordering is also specific to an objective that prices discomfort heavily: it is unchanged under a fourfold increase in the energy weight, and the gap between the two latent placements closes to zero at ninefold (Appendix E). The oracle is itself a learner: a noise-free expert controller still outscores every learned policy where we ran it (Table 8), so recovery measures what this pipeline extracts from the labels, not the best control the building admits. Statistical coverage is uneven: tier B is run on two configurations (Table 2), enough to show that a full-pipeline redraw moves grounding-only recovery outside the tier-A range in three of four draws but not to characterize the distribution. Those redraws were made at $5\%$ only, so the full-year margin of supervised latent learning over grounding-only rests on tier A alone. The twelve configurations are four archetypes crossed with three climates, not twelve independent draws: resampling archetypes takes that margin to $[-0.3, 25.4]$, and dropping the office archetype takes it from $+9.3$ to $+1.5$ points. The scarce-budget margins survive both (Table 11). The supervised-latent coefficient was piloted at only two of the four budgets, and on one configuration. The main grid trains and evaluates on one weather year. Re-evaluating on Sinergym's stochastic weather variant leaves recovery essentially unchanged ($82.1\%$ against $83.4\%$ over the four hot configurations), but the logging policy barely moves either, so this is robustness to weather noise, not transfer to a new year; transfer to an unseen building or climate is untested. The policy, critic and decoder are memoryless (Table 3); if residual partial observability matters for control and not only for inferring the latent, a history-conditioned policy may perform better, and the sensitivity of the critic to what its bootstrap target contains (Appendix H) suggests that it does.

We did not identify the mechanism behind early supervision's benefit: of four candidates, one is unrelated and three correlate with it at similar strength, which twelve configurations cannot separate (§5).

## 7 Conclusion

The "sensors logged, actions lost" regime is not a dead end for offline RL: rewards there are computable from observations, so latent actions discovered from passive data support policy improvement rather than imitation. A deployable policy still needs real commands, and while they are scarce it matters where they go—the same $5\%$ budget recovers $36\%$, $59\%$ or $89\%$ by placement—though on these logs the two late placements converge by a fifth of a year.

## AI USE STATEMENT

We disclose our use of large language models in accordance with the ICLR policy on LLM usage and the Code of Ethics. We used a general-purpose large language model assistant for research coding, for locating related work, and for manuscript preparation. We did not use generative AI to produce experimental results or to decide which findings to report. All simulator-based results were produced by executing the released code with the EnergyPlus simulator; derived analyses, including statistical summaries and fixed-policy re-scoring, were computed from the resulting logs. The authors inspected the underlying logs for each table and figure, and verified every reference against the original source, including works identified with AI assistance. The research questions, the experimental design, and the interpretation of the results—including the decision to report the negative findings in Appendix G—are the authors'. AI-assisted code was reviewed by the authors and its behavior cross-checked against independently computed quantities where feasible. We take responsibility for the final content of this work, including all text, claims and artifacts.

## ETHICS STATEMENT

This work involves no human subjects, no personally identifying data, and no data collected from real occupants: all experiments run in building-energy simulation. The application domain concerns occupant thermal comfort, and we note that a controller optimizing an aggregate objective can distribute discomfort unevenly across zones; we therefore report comfort violation depth in addition to violation frequency, so that deep excursions affecting a subset of occupants are visible rather than averaged away. Any deployment on a real building would require safety envelopes on the commanded setpoints that we do not study here.

## REPRODUCIBILITY STATEMENT

The benchmark construction—how the action-free corpus is generated, which observations are withheld, and how rewards are computed—is specified in §3. Appendix A lists the architecture, optimizer and data hyperparameters shared across configurations, together with four implementation details that we found necessary; Appendix I covers the one input the supervised-latent arm and the Stage 3 decoder standardize differently. Appendix G reports the budget-specific and ablation coefficients. Appendix B reports the raw full-year scores for the grounding-only analysis and the physical outcomes behind the $5\%$ and full-year placement comparisons. Learned-policy entries average three tier-A seeds where the table captions indicate it; deterministic baselines are single runs. The released artifacts carry the per-configuration evidence behind the remaining aggregate tables: the per-run evaluation outputs from which each table is aggregated, the latent probes behind the alignment table and the mechanism analysis, and the held-out-weather evaluations. Experiments run inside the Sinergym v3.12.0 image (Jiménez-Raboso et al., 2021; Campoy-Nieves et al., 2025) built from the project's own Dockerfile (commit `f0b4106`, extras `drl`) with EnergyPlus `25.1.0-1c11a3d85f`; no environment, building model or reward function is patched. The image is built locally rather than pulled from a registry, so the commit above specifies the rebuild recipe; the released artifacts record the image digest and installed package versions for the exact image used to produce the reported results. Release-specific behavior on which we rely—particularly the treatment of inverted setpoints—was checked against this image rather than inferred from the framework description.

The reward is Sinergym's own `LinearReward` with its registered hyperparameters, evaluated over every controlled zone rather than the registered default subset, so the objective is the published functional form on the full zone set. Two code paths evaluate it: the offline RL targets are produced by calling that class on the stored next observations, while the deployment scores reported throughout accumulate the same published linear expression directly, which is what makes the closed-form re-scoring of Appendix E possible. A script released with the code recomputes every number these tables report—scores, recoveries, paired differences and bootstrap intervals—from the released per-run outputs, and fails on a mismatch. That check covers the reported quantities rather than the datasets the runs consumed; the properties of those datasets—action-free masking, the shared label rule, and the time index of each block of the next observation—are asserted separately by the released smoke test. The labeling budgets for the main comparison are defined in §3 and Appendix A; each auxiliary analysis states its own budget in the relevant ap-

pendix section or caption. Code, configuration files and the data-generation scripts are available at `https://anonymous.4open.science/r/action-free-rl-8F51`.

## REFERENCES


Marvin Alles, Xingyuan Zhang, Patrick van der Smagt, and Philip Becker-Ehmck. Latent action world models for control with unlabeled trajectories. *arXiv preprint arXiv:2512.10016*, 2025.

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (VPT): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Christian Blad, Simon Bøgh, and Carsten Skovmose Kallesøe. Data-driven offline reinforcement learning for HVAC-systems. *Energy*, 261:125290, 2022.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 4603–4623, 2024.

Alejandro Campoy-Nieves, Antonio Manjavacas, Javier Jiménez-Raboso, Miguel Molina-Solana, and Juan Gómez-Romero. Sinergym – a virtual testbed for building energy optimization with reinforcement learning. *Energy and Buildings*, 327:115075, 2025. doi: 10.1016/j.enbuild.2024.115075.

Matthew Chang, Arjun Gupta, and Saurabh Gupta. Learning value functions from undirected state-only experience. In *International Conference on Learning Representations (ICLR)*, 2022.

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *International Conference on Machine Learning (ICML)*, 2019.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning adaptable world models with latent actions. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, 2025.

Judah Goldfeder and John Sipple. A lightweight calibrated simulation enabling efficient offline learning for optimal control of real buildings. In *Proceedings of BuildSys*, 2023.

Judah Goldfeder, Victoria Dean, Zixin Jiang, Xuezheng Wang, Bing Dong, Hod Lipson, and John Sipple. The smart buildings control suite: A diverse open source benchmark to evaluate and scale HVAC control policies for sustainability. *arXiv preprint arXiv:2410.03756*, 2024.

Javier Jiménez-Raboso, Alejandro Campoy-Nieves, Antonio Manjavacas-Lucas, Juan Gómez-Romero, and Miguel Molina-Solana. Sinergym: A building simulation and control framework for training reinforcement learning agents. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)*, pp. 319–323. ACM, 2021. doi: 10.1145/3486611.3488729.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Sébastien Lachapelle. On the identifiability of latent action policies. *arXiv preprint arXiv:2510.01337*, 2025. Presented at the NeurIPS 2025 Workshop on Unifying Representations in Neural Models (UniReps), Extended Abstract Track (non-archival).

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Anqi Li, Byron Boots, and Ching-An Cheng. MAHALO: Unifying offline reinforcement learning and imitation learning from observations. In *International Conference on Machine Learning (ICML)*, volume 202, pp. 19360–19384, 2023.

Hsin-Yu Liu, Xiaohan Fu, Bharathan Balaji, Rajesh Gupta, and Dezhi Hong. B2RL: An open-source dataset for building batch reinforcement learning. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)*, pp. 462–465. ACM, 2022. doi: 10.1145/3563357.3566164.

Natinael Solomon Neggatu, Jeremie Houssineau, and Giovanni Montana. Action-free offline-to-online RL via discretised state policies. In *International Conference on Learning Representations (ICLR)*, 2026.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *International Conference on Machine Learning (ICML)*, 2025.

Karl Schmeckpeper, Oleh Rybkin, Kostas Daniilidis, Sergey Levine, and Chelsea Finn. Reinforcement learning with videos: Combining offline observations with interaction. In *Proceedings of the 2020 Conference on Robot Learning (CoRL)*, volume 155 of *PMLR*, pp. 339–354, 2021.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations (ICLR)*, 2024.

Felix Schur. Identifying latent actions and dynamics from offline data via demonstrator diversity. *arXiv preprint arXiv:2603.17577*, 2026.

Younggyo Seo, Kimin Lee, Stephen James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *International Conference on Machine Learning (ICML)*, 2022.

Yuda Song, Drew Bagnell, and Aarti Singh. Hybrid reinforcement learning from offline observation alone. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 46019–46049, 2024.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Chengjie Wu, Hao Hu, Yiqin Yang, Ning Zhang, and Chongjie Zhang. Planning, fast and slow: Online reinforcement learning with action-free offline data via multiscale planners. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, 2024.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025.

Qinqing Zheng, Mikael Henaff, Brandon Amos, and Aditya Grover. Semi-supervised offline reinforcement learning with action-free trajectories. In *International Conference on Machine Learning (ICML)*, 2023.

## A Implementation details

Reproducing the comparison requires knowing what the reported error bars cover (Table 2) and which quantities each stage may read (Table 3), since both bear on whether a placement is being evaluated fairly. This appendix also specifies the shared hyperparameters, how the labeling budget is constructed, which reward bounds the modules see, how observation dimensionality varies by building, and the practical requirements we found necessary.

All three stages use MLPs with SiLU activations (Table 4). Hyperparameters are shared across buildings and climates; the observation dimension, the number of controlled zones and the action dimension vary as the environment requires.

Table 2: What each tier of repetition resamples. Tier A isolates optimization randomness; tier B retrains from data collection onward and is run on two configurations for cost. The labeled subset is chosen from $\mathcal{D}_{\text{lab}}$ by a deterministic rule, so it is fixed across tier-A repetitions and shared by every method at a budget; tier B resamples it by collecting a new $\mathcal{D}_{\text{lab}}$. Neither tier is the resampling unit for the intervals in Table 1: those average the three tier-A runs within each configuration and then bootstrap the twelve configurations, so they express cross-configuration variation, not optimization noise or label selection. Re-running the 5-zone-hot pipeline with the same collection seed as the reported runs reproduces the seed-42 grounding-only score to within $10^{-6}$ reward units and the oracle exactly; the three-seed mean lies $0.014$ away and its recovery $3.7$ points. For this configuration and arm, training noise alone therefore does not explain the wider tier-B spread. The three evaluation outputs used for that check are released. †Per seed for the supervised-latent arm, once per configuration for the others (§5, Appendix F).

| Resampled | Tier A | Tier B |
|---|---|---|
| Network initialization, minibatch order | ✓ | ✓ |
| Stage 1† | ✓ | ✓ |
| Labeled subset of $\mathcal{D}_{\text{lab}}$ | | ✓ |
| Behavior-policy perturbation | | ✓ |
| Simulator initial state | | ✓ |
| $\mathcal{D}_{\text{obs}}$ and $\mathcal{D}_{\text{lab}}$ | | ✓ |
| Configurations × repetitions | $12 \times 3$ | $2 \times 2$ |

Table 3: What each module conditions on. Only Stage 1 uses a history window; everything downstream is memoryless, which scopes the policy class we search over (§6).

| Module | Stage | Input | Output |
|---|---|---|---|
| Inverse model $q_\phi$ | 1 | $h_t$, $x_{t+1}$ | $z_t$ |
| Forward model $p_\theta$ | 1 | $h_t$, $z_t$ | $\hat{x}_{t+1}$ |
| Critic $Q, V$ | 2 | $o_t$, $z_t$ | value |
| Latent policy $\pi_z$ | 2 | $o_t$ | $z_t$ |
| Decoder $d_\xi$ | 3 | $o_t$, $z_t$ | $a_t$ |

**Labeling budget.** $\mathcal{D}_{\text{lab}}$ is collected as a full year so that any budget can be drawn from it as a controlled variable, but no method ever trains on more than the stated fraction, and every method that consumes labels draws the same transitions at a given budget (a single shared selection rule in the released code). That rule selects evenly spaced transitions, so a budget fixes the number of labeled commands rather than the number of days over which they were collected: the $5\%$ budget is $1{,}751$ commands at roughly five-hour spacing, the count of eighteen days of operation but not eighteen consecutive days of instrumentation. Appendix G redistributes the same count into a single contiguous window. Where we report a full-year budget it is as the upper end of the sweep, not as the operating point we advocate.

**Exposing the comfort bounds.** The failure the main text refers to is concrete. With the seasonal comfort schedule left to be inferred from calendar features, the office building concentrated most of its annual violation in the two shoulder months, where the MLP interpolated between the summer and winter bands instead of switching; supplying the current bounds as explicit policy and decoder inputs removed most of that depth. This was one building and one seed, run while the observation set was still being settled rather than as part of the released sweep, so no figure from it is quoted here. The mechanism—an MLP smoothing a step schedule it has to infer—is not specific to buildings.

**Observation dimensionality.** Each environment contributes its registered variable set plus the air temperature of every controlled zone, since the comfort term is evaluated over all of them, so the non-endogenous block $w_t$ differs in width across buildings: the retail shop, for example, reports on-site storage variables that the other buildings do not. Because $w_t$ is the complement of the endogenous channels rather than a list of exogenous ones, it also carries channels that respond to the control—zone humidity, and in some buildings a further metered quantity. No module in the reported pipeline conditions on any of $w_{t+1}$ before the transition is observed; the critic sees

Table 4: Nominal hyperparameters, shared across the twelve occupied-building configurations; DC denotes the diagnostic data-center environments, which use a 600 s control step.

| **Stage 1: latent action discovery** | | **Stage 2: latent-space IQL** | |
|---|---|---|---|
| latent dimension $\dim z$ | 4 | hidden / layers | 256 / 3 |
| hidden / layers | 256 / 2 | learning rate | $3 \times 10^{-4}$ |
| history window $k$ | 4 steps | discount $\gamma$ | 0.99 |
| physical span | 60 min (40 in the DC) | AWR temp. $\eta$ | 3.0 |
| KL weight $\beta_{\mathrm{KL}}$ | $10^{-3}$ | expectile $\tau$ | 0.7 |
| learning rate | $3 \times 10^{-4}$ | advantage clip | 100 |
| batch size | 512 | target update | 0.005 |
| epochs | 50 | batch / steps | 256 / $10^5$ |
| validation split | 0.1 | | |
| **Stage 3: action decoder** | | **Data** | |
| hidden / layers | 256 / 2 | $\mathcal{D}_{\mathrm{obs}}$ | 1 year, perturbed RBC $\sigma$=1.0 |
| learning rate | $10^{-3}$ | $\mathcal{D}_{\mathrm{lab}}$ pool | 1 year, perturbed RBC $\sigma$=2.5 |
| | | budget used | 1–100%, evenly spaced |
| epochs / batch | 100 / 512 | steps per year | 35,040 (900 s) |
| output | $\tanh$-bounded | data center | 52,560 (600 s) |

the whole of $o_{t+1}$, which is available at deployment. Every configuration is trained and evaluated separately, and all comparisons in the paper are made within a configuration.

**Practical requirements.** Four implementation details proved necessary rather than optional. (i) The control timestep is detected from the environment rather than assumed: the data center runs at 600 s while other buildings run at 900 s, and a mismatched value corrupts both the reward integration and any time-derivative term. (ii) The reward is evaluated on $o_{t+1}$; scoring $o_t$ misattributes the consequence of $a_{t-1}$ to $a_t$, which is harmless where thermal inertia dominates but damaging where power responds within one step. (iii) Setpoint observations must be removed from $o$; otherwise the action is directly visible and the action-free setting is void. (iv) The next observation must carry the non-endogenous block at $t+1$. Substituting the block at $t$ looks harmless, on the reasoning that weather and calendar are exogenous, but that reasoning does not cover the whole block: $w$ is the complement of the endogenous channels rather than a list of exogenous ones, so it also holds zone humidity and, in some buildings, a further metered quantity, all of which respond to the control within a step. The critic would then bootstrap from a state that does not occur. Appendix H runs the benchmark both ways: the substitution raises almost every absolute recovery, our own arms included, and narrows the margins between placements, while leaving the ordering in that comparison unchanged below a full-year budget.

## B PER-CONFIGURATION RESULTS

Section 5 aggregates results over twelve building–climate configurations, and an average can hide both the spread across configurations and the reason a placement wins. This appendix reports per-configuration recovery, compares the aggregation rules, presents the controls that separate policy improvement from imitation, and decomposes the shaped score into energy and discomfort.

**Recovery configuration by configuration.** At a full-year budget, the grounding-only arm beats the logging policy in every occupied-building configuration, but the margin ranges from 17.0 to 107.9% of the oracle's improvement (Table 5). The office, which drives fifteen zones from a single setpoint pair, is the hardest by a wide margin: its three climates recover 61, 17 and 71%, and no other configuration falls below 79%. Two configurations, the cool warehouse and the mixed retail shop, exceed 100%, which means the grounding-only policy scores above the single oracle run there. Which aggregate one reports also matters, because a mean of ratios is dominated by configurations with a small oracle headroom, so Table 6 gives the mean, the median and the gain-pooled ratio side by side. Table 7 gives the raw scores underlying the full-year grounding-only results in Tables 1, 5 and 6; the released artifacts contain the per-configuration outputs for the remaining method–budget combinations.

Table 5: Cross-configuration evaluation of the grounding-only arm at a full-year budget: percentage of the oracle's improvement recovered, averaged over three tier-A seeds. Each configuration is trained separately with identical nominal hyperparameters; this is not transfer of a policy or latent space, which we do not claim. Zones and actuated setpoints per building in parentheses.

| Building (zones / setpoints) | Hot | Cool | Mixed | Mean |
|---|---|---|---|---|
| 5-zone VAV (5 / 2) | 93.3% | 92.3% | 99.8% | 95.1% |
| Warehouse (1 / 2) | 79.8% | 107.9% | 84.6% | 90.8% |
| Retail shop (5 / 2) | 99.2% | 99.5% | 101.6% | 100.1% |
| Medium office (15 / 2) | 61.3% | 17.0% | 71.2% | 49.9% |

Table 6: Recovery by method and budget over the twelve occupied-building configurations, seeds averaged within a configuration before aggregating. The mean and the gain-pooled ratio differ by at most 4.1 points for either arm at any budget; the clipped rung of Appendix D separates them by 92 and 42 points at the two scarcest and by about 5 at the other two, since a mean of ratios is dominated by the low-headroom configurations, in which that rung fails worst. Pooling weights a configuration by the oracle's headroom, which is uneven: the hot office alone carries 23% of it and the three largest carry 47% between them. Imputation is the fourth rung of Appendix D; the worst/best rows are for latent + decoder.

| Method | 1% | 5% | 20% | 100% |
|---|---|---|---|---|
| *mean of per-configuration recoveries* | | | | |
| Supervised latent | 74.0 | 89.4 | 86.8 | 93.3 |
| Latent + decoder | 36.9 | 58.7 | 71.6 | 84.0 |
| Imputation + IQL | −27.0 | 36.5 | 77.1 | 83.9 |
| *median* | | | | |
| Supervised latent | 72.3 | 93.0 | 87.3 | 94.8 |
| Latent + decoder | 37.7 | 59.1 | 79.7 | 92.8 |
| Imputation + IQL | −15.9 | 41.2 | 83.0 | 92.1 |
| *gain-pooled ratio* | | | | |
| Supervised latent | 70.9 | 91.1 | 86.3 | 92.1 |
| Latent + decoder | 36.0 | 56.0 | 69.2 | 79.9 |
| Imputation + IQL | −23.8 | 37.4 | 73.4 | 79.8 |
| Worst configuration | −47.2 | 4.4 | 3.4 | 17.0 |
| Best configuration | 82.9 | 89.3 | 92.3 | 107.9 |

**Two controls.** Masking the latent collapses performance below the logging policy even though the decoder still sees calendar and weather, so the latent is what the policy acts on. And a noise-free expert rule-based controller still outscores every learned policy here: the gap we close is to the action-labeled oracle, not to the best controller a domain expert could write (Table 8).

**Imitation controls.** Both fail at every budget, which is why they are not in Table 1. BCO, which clones the imputed actions rather than improving on them, recovers −463.5, −228.6, −145.8 and −91.1% at the four budgets; a clone of $\mathcal{D}_{\text{lab}}$ alone recovers −376.3, −279.3, −52.9 and −65.0%.

**Behavior-policy scope.** All logs come from one family of perturbed rule-based controllers, so dependence on the logging-policy family remains a limitation (§6).

**How the small-budget runs fail.** At the 5% budget, the grounding-only runs are worse on average and more variable: per-configuration standard deviations across seeds run from 2 to 39 points, and two runs land below the logging policy they were trained to improve on, at −26% and −1%, both in the office. The mixed 5-zone configuration shows the shape of the failure most clearly: its three seeds consumed 23.2, 22.7 and 21.4 MWh while accumulating 3,993, 4,871 and 9,092 violation degree-hours—the worse the run, the less energy it used. There the loosely grounded decoder under-actuates, saving energy at the cost of discomfort. The office runs are less tidy, their energy varying by 2% across seeds while their violation varies by 17 to 23%, but the direction is the same: the small-budget failure appears as increased discomfort rather than increased energy use. Table 9 provides the grid-wide physical outcomes underlying the aggregate operating points reported in §5.

Table 7: Raw full-year scores for the logging policy, the grounding-only arm and the oracle, used for the full-year grounding-only entries in Tables 1, 5 and 6. Behavior, latent + decoder (full-year labels) and oracle are the budget-sweep runs. Recovery is $(J_{\text{ours}} - J_{\text{beh}})/(J_{\text{oracle}} - J_{\text{beh}})$ and is left blank where the oracle does not improve on the behavior policy, so the ratio is undefined; it is computed from unrounded returns, so recomputing it from the three-decimal columns can differ in the last digit. The latent + decoder column averages three tier-A seeds; the behavior and oracle columns are single runs. †The data center is not part of the budget sweep; its rows come from the standalone chain of Table 8 and are not pooled with the twelve occupied-building configurations.

| Building | Climate | Behavior | Latent + decoder | Oracle | Recovery |
|---|---|---|---|---|---|
| 5-zone VAV | hot | $-0.683$ | $-0.321$ | $-0.295$ | 93.3% |
| | cool | $-0.773$ | $-0.354$ | $-0.319$ | 92.3% |
| | mixed | $-0.664$ | $-0.276$ | $-0.276$ | 99.8% |
| Warehouse | hot | $-0.335$ | $-0.274$ | $-0.258$ | 79.8% |
| | cool | $-0.186$ | $-0.115$ | $-0.120$ | 107.9% |
| | mixed | $-0.244$ | $-0.164$ | $-0.150$ | 84.6% |
| Medium office | hot | $-3.632$ | $-2.972$ | $-2.556$ | 61.3% |
| | cool | $-1.503$ | $-1.449$ | $-1.188$ | 17.0% |
| | mixed | $-2.695$ | $-2.217$ | $-2.023$ | 71.2% |
| Retail shop | hot | $-0.831$ | $-0.362$ | $-0.358$ | 99.2% |
| | cool | $-4.116$ | $-3.791$ | $-3.790$ | 99.5% |
| | mixed | $-2.458$ | $-2.033$ | $-2.039$ | 101.6% |
| Data center† | hot | $-1.219$ | $-1.239$ | $-1.258$ | — |
| | cool | $-0.301$ | $-0.292$ | $-0.292$ | 104.2% |
| | mixed | $-0.806$ | $-0.831$ | $-0.838$ | — |

Table 8: One-year deployment, 5-zone building, hot climate. Score is the Sinergym `LinearReward` mean (higher is better). These are the earlier per-configuration runs, retained because they carry two controls the sweep does not: a noise-free expert controller and a latent-masked ablation. They are not the sweep runs. For this configuration the sweep gives $-0.295$ and $-0.321$ for the oracle and the latent route (Table 7); those scores are the ones used in the full-year grounding-only entries and aggregates. The oracle and latent + decoder rows are mean±std over three tier-A seeds; the expert controller and behavior policy are deterministic single runs, and BCO and the observation-only ablation are single control runs.

| Method | Score | Gain over behavior |
|---|---|---|
| Expert RBC (reference) | $-0.250$ | $+63.4\%$ |
| Oracle IQL (true actions) | $-0.301 \pm 0.006$ | $+56.0\%$ |
| Latent + decoder (full-year labels) | $-0.323 \pm 0.021$ | $+52.7\%$ |
| Behavior policy (logging) | $-0.683$ | — |
| BCO (Torabi et al., 2018) | $-0.787$ | $-15.3\%$ |
| Observation-only (no latent) | $-0.992$ | $-45.2\%$ |

Table 9: Physical outcomes behind the shaped score, for every placement of the labels at both budgets. Placement-strategy rows average three tier-A seeds within each configuration; the logging-policy and oracle rows are single runs. The operating points these imply are read off in §5; the imputation rows are the fourth rung of Appendix D.

| Configuration | Labels spent on | Energy (kWh) | Viol. h | Viol. deg-h | Score |
|---|---|---|---|---|---|
| 5-zone, hot | Logging policy | 19,398 | 4,170 | 10,031 | $-0.683$ |
| | Imputation, 5% | 20,017 | 4,569 | 7,257 | $-0.528$ |
| | Latent+decoder, 5% | 20,080 | 4,627 | 6,025 | $-0.459$ |
| | Supervised latent, 5% | 22,720 | 2,853 | 3,199 | $-0.312$ |
| | Imputation, full year | 21,135 | 3,626 | 4,156 | $-0.358$ |
| | Latent+decoder, full year | 21,632 | 3,176 | 3,467 | $-0.321$ |
| | Supervised latent, full year | 21,695 | 2,828 | 3,085 | $-0.300$ |
| | Oracle (true actions) | 20,710 | 2,981 | 3,102 | $-0.295$ |

*continued on the next page*

*Table 9, continued*

| Configuration | Labels spent on | Energy (kWh) | Viol. h | Viol. deg-h | Score |
|---|---|---|---|---|---|
| 5-zone, cool | Logging policy | 18,557 | 4,671 | 11,684 | −0.773 |
| | Imputation, 5% | 18,771 | 5,085 | 9,070 | −0.625 |
| | Latent+decoder, 5% | 20,005 | 4,668 | 6,531 | −0.487 |
| | Supervised latent, 5% | 21,263 | 2,946 | 4,622 | −0.385 |
| | Imputation, full year | 20,572 | 3,627 | 5,004 | −0.403 |
| | Latent+decoder, full year | 21,359 | 2,838 | 4,062 | −0.354 |
| | Supervised latent, full year | 21,066 | 2,741 | 3,953 | −0.346 |
| | Oracle (true actions) | 20,414 | 2,616 | 3,542 | −0.319 |
| 5-zone, mixed | Logging policy | 21,238 | 4,232 | 9,504 | −0.664 |
| | Imputation, 5% | 22,322 | 4,478 | 6,978 | −0.526 |
| | Latent+decoder, 5% | 22,433 | 4,259 | 5,985 | −0.470 |
| | Supervised latent, 5% | 25,077 | 2,452 | 2,477 | −0.284 |
| | Imputation, full year | 23,812 | 3,463 | 3,739 | −0.349 |
| | Latent+decoder, full year | 24,438 | 2,516 | 2,397 | −0.276 |
| | Supervised latent, full year | 23,913 | 2,558 | 2,360 | −0.271 |
| | Oracle (true actions) | 23,423 | 3,010 | 2,488 | −0.276 |
| Warehouse, hot | Logging policy | 38,780 | 3,190 | 1,984 | −0.335 |
| | Imputation, 5% | 38,596 | 3,726 | 1,711 | −0.318 |
| | Latent+decoder, 5% | 38,672 | 3,122 | 1,213 | −0.290 |
| | Supervised latent, 5% | 38,746 | 2,369 | 1,033 | −0.280 |
| | Imputation, full year | 38,804 | 1,765 | 645 | −0.258 |
| | Latent+decoder, full year | 38,772 | 2,097 | 916 | −0.274 |
| | Supervised latent, full year | 38,801 | 1,949 | 833 | −0.269 |
| | Oracle (true actions) | 38,818 | 1,724 | 641 | −0.258 |
| Warehouse, cool | Logging policy | 12,151 | 2,994 | 2,042 | −0.186 |
| | Imputation, 5% | 12,072 | 2,031 | 1,326 | −0.145 |
| | Latent+decoder, 5% | 12,108 | 1,961 | 1,020 | −0.127 |
| | Supervised latent, 5% | 12,202 | 1,510 | 842 | −0.118 |
| | Imputation, full year | 12,135 | 1,270 | 636 | −0.106 |
| | Latent+decoder, full year | 12,128 | 1,663 | 804 | −0.115 |
| | Supervised latent, full year | 12,114 | 1,610 | 808 | −0.115 |
| | Oracle (true actions) | 12,096 | 1,729 | 898 | −0.120 |
| Warehouse, mixed | Logging policy | 21,983 | 3,102 | 2,069 | −0.244 |
| | Imputation, 5% | 22,054 | 1,812 | 898 | −0.177 |
| | Latent+decoder, 5% | 21,833 | 2,138 | 916 | −0.177 |
| | Supervised latent, 5% | 22,002 | 1,361 | 562 | −0.158 |
| | Imputation, full year | 22,082 | 794 | 198 | −0.137 |
| | Latent+decoder, full year | 21,918 | 1,560 | 685 | −0.164 |
| | Supervised latent, full year | 21,937 | 1,350 | 634 | −0.161 |
| | Oracle (true actions) | 22,129 | 957 | 411 | −0.150 |
| Office, hot | Logging policy | 175,497 | 6,933 | 46,087 | −3.632 |
| | Imputation, 5% | 170,312 | 7,574 | 38,597 | −3.175 |
| | Latent+decoder, 5% | 172,415 | 7,268 | 35,120 | −2.989 |
| | Supervised latent, 5% | 179,875 | 6,274 | 26,080 | −2.515 |
| | Imputation, full year | 173,082 | 6,913 | 29,664 | −2.681 |
| | Latent+decoder, full year | 172,534 | 7,279 | 34,820 | −2.972 |
| | Supervised latent, full year | 173,599 | 6,792 | 30,368 | −2.724 |
| | Oracle (true actions) | 168,374 | 6,430 | 27,944 | −2.556 |
| Office, cool | Logging policy | 51,230 | 6,057 | 21,202 | −1.503 |
| | Imputation, 5% | 42,738 | 8,104 | 28,247 | −1.856 |
| | Latent+decoder, 5% | 48,085 | 7,480 | 21,276 | −1.489 |
| | Supervised latent, 5% | 56,625 | 6,190 | 17,046 | −1.296 |
| | Imputation, full year | 43,438 | 7,622 | 23,113 | −1.567 |
| | Latent+decoder, full year | 47,501 | 7,070 | 20,636 | −1.449 |
| | Supervised latent, full year | 49,222 | 6,502 | 17,335 | −1.270 |
| | Oracle (true actions) | 42,662 | 6,694 | 16,551 | −1.188 |

*continued on the next page*

*Table 9, continued*

| Configuration | Labels spent on | Energy (kWh) | Viol. h | Viol. deg-h | Score |
|---|---|---|---|---|---|
| Office, mixed | Logging policy | 118,796 | 6,411 | 35,340 | −2.695 |
| | Imputation, 5% | 111,165 | 7,404 | 31,541 | −2.435 |
| | Latent+decoder, 5% | 111,284 | 7,518 | 32,211 | −2.474 |
| | Supervised latent, 5% | 122,968 | 6,280 | 25,980 | −2.185 |
| | Imputation, full year | 111,358 | 7,038 | 28,239 | −2.247 |
| | Latent+decoder, full year | 115,056 | 6,856 | 27,333 | −2.217 |
| | Supervised latent, full year | 117,531 | 6,317 | 24,858 | −2.090 |
| | Oracle (true actions) | 111,408 | 6,287 | 24,309 | −2.023 |
| Retail, hot | Logging policy | 39,056 | 4,133 | 10,660 | −0.831 |
| | Imputation, 5% | 40,857 | 3,375 | 3,994 | −0.461 |
| | Latent+decoder, 5% | 40,949 | 3,233 | 3,436 | −0.430 |
| | Supervised latent, 5% | 43,862 | 2,351 | 2,632 | −0.401 |
| | Imputation, full year | 43,624 | 2,377 | 2,054 | −0.366 |
| | Latent+decoder, full year | 45,005 | 1,921 | 1,837 | −0.362 |
| | Supervised latent, full year | 43,853 | 2,165 | 2,040 | −0.367 |
| | Oracle (true actions) | 43,427 | 2,220 | 1,925 | −0.358 |
| Retail, cool | Logging policy | 64,858 | 7,426 | 65,626 | −4.116 |
| | Imputation, 5% | 64,462 | 7,662 | 62,191 | −3.918 |
| | Latent+decoder, 5% | 64,594 | 7,580 | 62,488 | −3.935 |
| | Supervised latent, 5% | 66,510 | 7,324 | 60,071 | −3.808 |
| | Imputation, full year | 65,890 | 7,194 | 60,052 | −3.804 |
| | Latent+decoder, full year | 66,426 | 7,318 | 59,784 | −3.791 |
| | Supervised latent, full year | 66,251 | 7,439 | 60,028 | −3.804 |
| | Oracle (true actions) | 65,414 | 7,230 | 59,859 | −3.790 |
| Retail, mixed | Logging policy | 62,537 | 5,477 | 36,806 | −2.458 |
| | Imputation, 5% | 65,934 | 4,810 | 31,600 | −2.180 |
| | Latent+decoder, 5% | 64,910 | 4,951 | 30,919 | −2.135 |
| | Supervised latent, 5% | 67,439 | 4,303 | 29,228 | −2.053 |
| | Imputation, full year | 68,025 | 4,103 | 29,191 | −2.054 |
| | Latent+decoder, full year | 66,842 | 4,410 | 28,931 | −2.033 |
| | Supervised latent, full year | 66,271 | 4,437 | 28,925 | −2.029 |
| | Oracle (true actions) | 65,098 | 4,468 | 29,222 | −2.039 |

**Is the latent an intervention or a transition residual?** The inverse model sees $x_{t+1}$, so nothing in the architecture forces the bottleneck to carry the action rather than an unpredictable exogenous component of the transition—a latent of the second kind would not be reproducible at deployment. We probe this directly: the incremental $R^2$ with which $z_t$ predicts the exogenous innovation (outdoor temperature, solar, occupancy step change) over and above the history window to which the Stage-1 models already have access is comparable to what it carries about the action itself. In the 5-zone hot configuration, the incremental $R^2$ for the outdoor-temperature and occupancy innovations is $0.21$ and $0.19$, against $0.22$ and $0.27$ for the two action dimensions. Across the five buildings, the ratio of the mean exogenous-innovation $R^2$ to the mean action $R^2$ ranges from about $0$ in the data center to $0.56$ in the 5-zone building. The latent is therefore not a pure intervention variable; it also carries information about exogenous changes in the transition. Table 10 gives the probe scores for every configuration.

**The twelve configurations are not twelve draws.** They are four archetypes crossed with three climates, so a bootstrap that resamples configurations treats correlated units as exchangeable. Table 11 splits the supervised-latent margin over grounding-only both ways and resamples archetypes instead, keeping each archetype's three climates together. The margin is positive in every stratum but one and at every budget, and it is not evenly spread: the office archetype's full-year margin is $+32.9$ against $+2.9$, $+2.9$ and $-1.4$ for the other three, and dropping it takes that margin from $+9.3$ to $+1.5$ points. Resampling archetypes rather than configurations widens the intervals accordingly, and the full-year one then includes zero, $[-0.3, 25.4]$ against $[1.7, 19.8]$. The scarce-budget margins survive either resampling; the full-year one is the claim this structure will not carry.

Table 10: Latent–action alignment ($R^2$ of a random-Fourier probe from $z$ to ground-truth actions, HVAC-active steps only). Ground truth is used for this diagnostic only, never for training or model selection. The estimator is the one reported by Stage 1 at the end of training; that diagnostic draws its own subsample of the HVAC-active steps. The standalone probe released with the code computes the same quantity on a different subsample and agrees to within $0.008$ in every cell.

| | hot | | cool | | mixed | |
|---|---|---|---|---|---|---|
| Building | heat | cool | heat | cool | heat | cool |
| 5-zone VAV | 0.252 | 0.368 | 0.351 | 0.246 | 0.393 | 0.354 |
| Warehouse | 0.126 | 0.361 | 0.270 | 0.128 | 0.232 | 0.191 |
| Retail shop | 0.166 | 0.271 | 0.220 | 0.089 | 0.229 | 0.152 |
| Medium office | 0.092 | 0.410 | 0.098 | 0.273 | 0.186 | 0.381 |
| Data center[1] | 0.779 | | 0.855 | | 0.781 | |

[1]The data center has a single cooling setpoint, so one value is reported per climate rather than a heating and a cooling entry.

Table 11: Supervised latent learning minus grounding-only, in recovery points, split by archetype and by climate, and with one archetype held out at a time. Bottom rows: the mean over all twelve with a percentile bootstrap over configurations, and the same statistic resampling the four archetypes as units.

| | 1% | 5% | 20% | 100% |
|---|---|---|---|---|
| *by archetype* | | | | |
| 5-zone VAV | +55.5 | +35.9 | +6.8 | +2.9 |
| Warehouse | +37.1 | +16.0 | +10.4 | +2.9 |
| Medium office | +48.5 | +49.4 | +34.7 | +32.9 |
| Retail shop | +7.6 | +21.6 | +9.1 | −1.4 |
| *by climate* | | | | |
| Hot | +34.8 | +25.2 | +10.5 | +8.4 |
| Cool | +52.7 | +34.3 | +20.3 | +13.6 |
| Mixed | +24.0 | +32.7 | +15.0 | +6.0 |
| *one archetype held out* | | | | |
| without 5-zone | +31.1 | +29.0 | +18.0 | +11.5 |
| without warehouse | +37.2 | +35.7 | +16.9 | +11.5 |
| without office | +33.4 | +24.5 | +8.8 | +1.5 |
| without retail | +47.0 | +33.8 | +17.3 | +12.9 |
| All twelve | +37.2 | +30.7 | +15.2 | +9.3 |
| CI over configurations | [23.2, 52.5] | [21.7, 40.3] | [8.8, 23.9] | [1.7, 19.8] |
| CI over archetypes | [17.8, 52.0] | [18.8, 42.7] | [7.7, 28.3] | [−0.3, 25.4] |

## C WHAT COVARIES WITH RECOVERABILITY

**Recoverability covaries with control-axis activity (Fig. 2).** When a thermostat sits inside its deadband, any setpoint produces the same transition and the action is unrecoverable in principle. Aggregated over a year, this predicts that an axis is recoverable to the extent it is exercised. Climate moves that quantity, among others: moving from hot to cool/mixed raises heating-axis recovery in all four two-axis buildings ($0.25\to0.39$, $0.13\to0.27$, $0.17\to0.23$, $0.09\to0.19$), while cooling-axis recovery is highest in the hot climate in all four. Climate is not a clean intervention on excitation, however: it also changes load, operating mode and the return of the logging policy. Isolating it would require holding the building, the weather year and the reward fixed while varying the perturbation applied to each setpoint axis independently, and tracing the chain from axis excitation to latent–action alignment to recovered oracle gain. We did not complete that controlled sweep. We therefore state the relationship as a mechanism the data are consistent with, not one we have isolated.

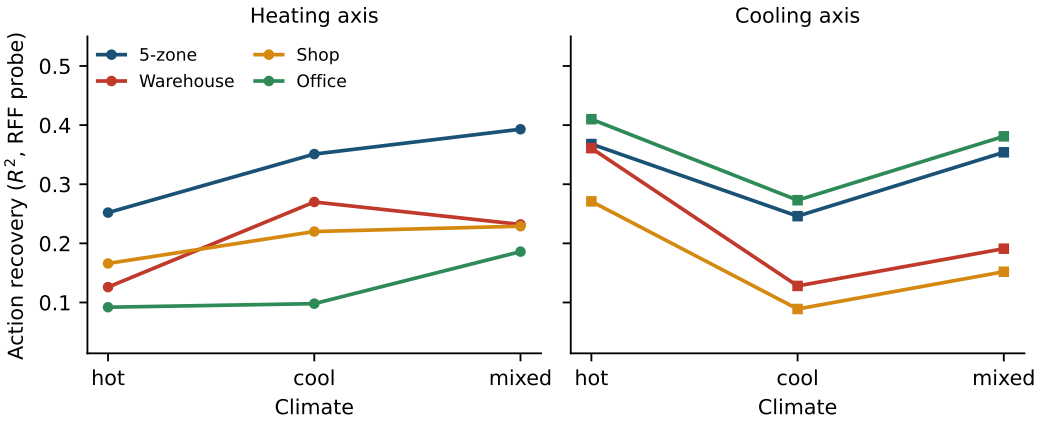



Figure 2: Climate covaries with control-axis activity. The vertical axis is the random-Fourier (RFF) probe of Table 10, using the same Stage-1 estimator described in that caption. Heating-axis recovery is lowest in hot climates in all four two-axis buildings; cooling-axis recovery is highest there. Climate is chosen by us, but it moves thermal load, operating mode and state visitation along with axis activity, so this figure motivates the mechanism rather than isolating it. The controlled sweep that would isolate it—varying each axis's excitation independently at a fixed building, weather year and reward—is one we did not complete.

**Excitation is required for latent-action recovery in this setting.** With a deterministic seasonal controller, actions are a function of the observed calendar; the FDM learns the seasonal effect from $w$ directly and the latent's action content decays to zero over training. Latent action discovery therefore requires excitation in the behavior policy—a latent-space analogue of persistent excitation in system identification. This result is consistent with the theoretical failure mode: Lachapelle (2025) shows that a deterministic ground-truth policy admits a degenerate solution in which the forward model ignores the latent entirely, violating informativeness. What our experiments add is that the condition is not binary in practice—a policy can be stochastic overall yet leave individual control axes unexcited, and recovery then degrades axis-by-axis rather than collapsing. Such excitation can occur in realistic logs; our perturbed-controller protocol models that possibility.

**A degenerate regime.** In the data center, the comfort band (18–27°C) is wide enough that the logging policy violates it for only $4$ hours per year; the objective is therefore effectively energy-dominated, and its optimum lies outside the logged support. In the hot and mixed climates the oracle does not improve on the logging policy at all, and in the cool climate its headroom is $0.009$ reward units, less than a fortieth of the 5-zone building's. The latent route finishes at or above the oracle in all three. In the hot climate the oracle scores $3.3\%$ below the logging policy and the latent route $1.6\%$ below, so the latent route is $1.6\%$ ahead of the oracle; in the cool and mixed climates the two are within $0.001$ and $0.007$ reward units of each other (Table 7). The residual gap here is therefore consistent with the absence of oracle headroom rather than with an additional penalty from action-freeness, and the learned policy tracks the oracle even when the oracle does not improve on the logging policy.

**Why does early supervision help?** Four quantities that vary across configurations were correlated with the $5\%$ benefit of the supervised-latent arm over grounding-only (Table 12). The first is the content of the unsupervised latent that the probe attributes to exogenous innovation rather than to the action—the hypothesis of §4 that supervision should matter most where the latent had absorbed the most non-actionable content. It is unrelated to the benefit. The second is how far the auxiliary loss moved the latent: the probe's $z \to a$ $R^2$ over all steps—Table 10 reports its HVAC-active subsample—on the supervised Stage 1 minus the shared unsupervised one. The loss raises alignment in eight of the twelve configurations by up to $0.21$ and lowers it in the other four by at most $0.08$, once by only $0.002$; the change correlates with the benefit. The third and fourth describe the late decoder that grounding-only relies on instead: its $R^2$ on the labeled transitions the budget did not

select, and the spread of the commands it emits relative to the spread of its labels. Both correlate positively—the benefit of early supervision is larger where the late decoder fits better, not worse.

The three positive correlations are of similar size and none survives a correction for having looked three times. All three rise with the same gain, and each is the kind of quantity that would rise with how informative a configuration's labels are: where the labels carry more, the loss has more to move the latent toward, the decoder has more to fit, and early supervision has more to gain. That reading describes where the benefit appears, not what produces it, and the twelve configurations—four archetypes by three climates—are too few and too structured to separate the candidates. The sweep that could, varying label informativeness at a fixed building, is one we did not run.

Table 12: Four candidate correlates of the $5\%$ benefit of early supervision, across the twelve occupied-building configurations. Pearson and Spearman coefficients with two-sided permutation $p$-values; no correction for multiple comparisons. The configurations are four archetypes crossed with three climates, so the $p$-values describe this set rather than a population.

| Candidate | $r$ | $p$ | $\rho$ | $p$ |
|---|---|---|---|---|
| Exogenous content of the unsupervised latent | $-0.16$ | $0.62$ | $-0.03$ | $0.94$ |
| Change in $z \to a$ alignment from the auxiliary loss | $+0.60$ | $0.04$ | $+0.51$ | $0.09$ |
| Held-out $R^2$ of the late decoder | $+0.51$ | $0.09$ | $+0.46$ | $0.15$ |
| Output spread of the late decoder | $+0.49$ | $0.11$ | $+0.48$ | $0.12$ |

## D How strong an imputation baseline?

Filling in the action column is a family of baselines, not one, and the comparison in §5 is only as informative as the strongest member of it. Three implementation choices affect it, and the first setting of each happened to favor the latent route, so we evaluated progressively stronger imputation baselines until the two placements differed only in what their label-consuming module reads.

The first is whether $\mathcal{D}_{\text{lab}}$ reaches the policy at all. Fitting an inverse model on the labeled transitions and running IQL on the pseudo-labeled corpus alone uses the labels once; Zheng et al. (2023) instead run offline RL on the union of the true- and proxy-labeled data. The second is the output range. Our Stage 3 decoder is bounded to the range of its labeled subset (§4), so it cannot emit a command outside the range its labels attest to. A plain regression inverse model can, and does: at a $1\%$ budget, $65.0\%$ of its pseudo-labels fall outside that range. The excursions are not merely wider than the labeled subset—the largest exceed $60°$C, beyond anything the actuator accepts—so the critic is fitted on state–action pairs the plant cannot produce. We report the fraction outside the decoder-matched labeled range rather than the fraction outside the actuator bounds, because the decoder is explicitly constrained to the former; holding the inverse model to the same rule removes the asymmetry.

The third choice is how the inverse model is fitted at all. The first three rungs train it for a fixed 60 epochs, keep whatever weights the last epoch leaves behind, and emit an unbounded regression that is clipped afterwards. The Stage 3 decoder, over the same labels, runs 100 epochs, keeps the weights with the lowest validation MSE, and can only emit a command inside the labeled range because the range is in its parameterization. Comparing the two therefore compared a tuned module with an untuned one as well as one placement with another. The fourth rung gives the inverse model that recipe, with one exception: each stage floors its validation split at a different size, 32 transitions for the decoder and $64$ for the inverse model. At $1\%$ those floors bind and the splits differ, $315/35$ against $286/64$ of the same $350$ labels; at every other budget a tenth of the subset clears both floors and the splits are identical. The $5\%$ comparison the paper leads with is therefore matched; the $1\%$ column is not, and in the direction that gives the inverse model less to train on.

Each correction helps below a full year, and the last helps by far the most (Table 13): the first two move the $5\%$ figure from $-157.5$ to $-127.4$ and the third to $36.5$, so the gap to the latent route falls from $216.2$ points to $186.0$ and then to $22.2$, where its interval, $[8.1, 42.2]$, still excludes zero with the latent route ahead in $10$ of $12$ configurations. At $20\%$ the fourth rung is level with the latent route ($-5.5$, $[-13.8, 2.8]$, ahead in $4$ of $12$), and so it is at a full year. The crossover in §5 is therefore a property of the matched baseline, not of the untuned one.

Two measurements say what the fourth rung changed. The fixed epoch budget was past the validation minimum in 125 of the 144 main-grid runs: the selected epoch ranges from 13 to 93 with a median of 39, and the last epoch is never the one kept (`results/idm_fit.csv`, which also carries the four tier-B fits, all of them below 60 as well). And the range stops being a post-processing step: the fraction of pseudo-labels outside the labeled range is 0.00% at every budget, against 65.0%, 27.8%, 6.7% and 1.6% for the clipped rung. What remains after all three corrections—22.2 points at 5%, none at 20%—is what the placement is worth.

Table 13: Four imputation baselines, as a percentage of the oracle's improvement. Each rung consumes the same labels as the one above and adds a fairness correction; the fourth gives the inverse model the decoder's own training recipe, so that the two late placements differ only in what their module reads. The paired rows are latent + decoder minus that rung, with bootstrap intervals resampling configurations. The main table reports the fourth rung.

| Labels (fraction of one year) | 1% | 5% | 20% | 100% |
|---|---|---|---|---|
| *recovery* | | | | |
| labels reach the inverse model only | −453.2 | −157.5 | 15.3 | 63.8 |
| + true-labeled data in offline RL | −446.9 | −155.0 | 27.2 | 79.1 |
| + decoder's output range, clipped | −378.7 | −127.4 | 31.5 | 78.6 |
| + decoder's training recipe | −27.0 | 36.5 | 77.1 | 83.9 |
| *paired difference, latent+decoder − that rung* | | | | |
| inverse model only | +490.1 | +216.2 | +56.3 | +20.2 |
| + true-labeled data | +483.8 | +213.7 | +44.4 | +4.9 |
| + output range, clipped | +415.5 | +186.0 | +40.1 | +5.3 |
| + training recipe | +63.9 | +22.2 | −5.5 | +0.1 |
| *pseudo-labels outside the decoder's range* | | | | |
| fraction of $\mathcal{D}_{\text{obs}}$ | 65.0% | 27.8% | 6.7% | 1.6% |

## E Does the ordering depend on the objective?

Because the recovered improvement is mostly a comfort gain (§5), we test whether the placement ordering depends on the weights in Eq. 1. The two late placements under-actuate, and under-actuation is only a failure when discomfort is expensive.

Equation 1 is linear in annual energy and annual violation degree-hours, so the score under any energy weight $\omega$ can be recomputed in closed form from the two physical quantities of Table 9,

$$J(\omega) = -\frac{10^3\,\lambda_P\,\omega\,E + \lambda_T\,(1-\omega)\,V}{H}, \tag{4}$$

with $E$ the annual HVAC energy in kWh, $V$ the annual comfort violation in degree-hours, and $H = 8{,}760\,\text{h}$ the length of the evaluation year. The factor $10^3$ converts kWh to the mean power in watts that Eq. 1 penalizes. Because the reported score accumulates this same linear expression, Eq. 4 agrees with it by construction rather than as an independent test; what the agreement does establish is that the energy and violation columns we re-score from are the ones the reported returns were computed from. We verified this on every released run—at $\omega = 0.5$ the largest discrepancy is $2.4 \times 10^{-4}$ reward units—so any mismatch between a row's physical quantities and its reported score would be detected. We then re-score every policy over a range of weights. This re-scores fixed policies rather than retraining them. It therefore tests whether the recommendation changes under a different energy price, not what retraining under that price would produce.

Reweighting affects the two comparisons differently (Table 14). The supervised-latent–imputation gap narrows from 52.9 to 14.4 points and stays positive at every weight, whereas the supervised-latent–grounding-only gap shrinks steadily and closes at the ninefold energy weight, because grounding-only uses less energy.

Table 14: Recovery at a 5% budget when the trained policies are re-scored under a different energy-comfort price. The second column gives the energy weight relative to the published objective. The final column reports the number of configurations in which supervised latent learning achieves the highest recovery among the three strategies. Policies are not retrained.

| $\omega$ | energy price | sup. latent | latent+dec. | imputation | sup.−lat. | sup. best (of 12) |
|---|---|---|---|---|---|---|
| 0.5 (published) | 1.0× | 89.4 | 58.7 | 36.5 | +30.7 | 12/12 |
| 0.6 | 1.5× | 87.5 | 59.2 | 38.1 | +28.3 | 12/12 |
| 0.7 | 2.3× | 84.6 | 60.0 | 40.4 | +24.5 | 12/12 |
| 0.8 | 4.0× | 79.3 | 61.6 | 43.9 | +17.7 | 11/12 |
| 0.9 | 9.0× | 65.4 | 65.9 | 50.9 | −0.5 | 8/12 |

That closing is consistent with the under-actuation pattern described in §5 and is treated as a scope condition on the recommendation. Under this fixed-policy re-scoring, supervised latent learning is preferred near the published weighting; at the ninefold energy weight the two latent placements are level on mean recovery (65.4% against 65.9%), with supervised latent learning still the best of the three in 8 of 12 configurations. These results do not establish the ordering that would result from retraining under the alternative weights.

## F Do the gaps survive a full-pipeline redraw?

The intervals in Table 1 resample configurations after averaging tier-A reruns and therefore do not capture variation from tier-B full-pipeline redraws. To assess it, we ran all three placements on the two tier-B configurations under two independent full-pipeline redraws at the 5% budget. Each tier-B cell resamples all tier-B components listed in Table 2, including the data-generation process, the labeled subset, Stage 1, the network initialization and the minibatch order. This ablation therefore tests the joint effect of a new data draw and retraining rather than dataset-draw variability in isolation.

Table 15: The three placements under four independent tier-B full-pipeline redraws at a 5% budget, as a percentage of the oracle's improvement. Stage 1 is retrained within each draw. Some absolute recoveries vary by tens of points across redraws, while the placement ordering remains unchanged. The imputation arm is the fourth rung of Appendix D, the one Table 1 reports; the second rung, which earlier versions of this table used, recovers −67.9, −37.6, −101.9 and −102.8 on the same four draws. The two difference columns are differences of the rounded cells.

| Tier-B redraw | imputation | latent+dec. | sup. latent | sup.−lat. | lat.−imp. |
|---|---|---|---|---|---|
| 5-zone hot, draw 1 | 10.2 | 55.8 | 95.0 | +39.2 | +45.6 |
| 5-zone hot, draw 2 | 32.7 | 64.6 | 91.5 | +26.9 | +31.9 |
| Warehouse hot, draw 1 | 21.1 | 64.5 | 78.7 | +14.2 | +43.4 |
| Warehouse hot, draw 2 | 8.0 | 55.9 | 66.4 | +10.5 | +47.9 |

**Retraining Stage 1 per seed does not move the grounding-only arm.** Tier A retrains Stage 1 per seed only for the supervised-latent arm, since only there do the labels reach it (Table 2). To check that this asymmetry is not what separates the two latent placements, we reran the grounding-only arm with a fresh Stage 1 at every seed, seed 42 included, over all twelve configurations and four budgets. Its recovery moves by +1.9, −2.2, +0.5 and +1.6 points at the four budgets, with every interval containing zero ($[-4.2, 8.7]$, $[-7.0, 1.8]$, $[-1.9, 3.3]$ and $[-0.4, 4.2]$) and the original run ahead in 9, 6, 5 and 8 of 12 configurations. Table 1 keeps the original run; the seed-symmetric one is released alongside it.

For reference, the tier-A recovery of the grounding-only arm at this budget spans 49.9–62.7% in the 5-zone hot configuration and 56.2–60.3% in warehouse hot; the tier-B draws give 55.8 and 64.6% for the former, one inside that range and one above it, and 64.5 and 55.9% for the latter, one above and one below. Within the 5-zone hot configuration of Table 15, the two redraws differ by 22 points for imputation and 4 points for supervised latent learning; the corresponding differences in warehouse hot are 13 and 12 points. This variation supports the qualification in §5 that the recovery estimates are conditional on the data-generation and training realization. The two paired differences

are positive and large in all four draws: $+10.5$ to $+39.2$ points for supervised latent over grounding-only, and $+31.9$ to $+47.9$ for grounding-only over imputation. Every arm here is run at the main grid's coefficients: $\lambda_{\text{act}}$=1.0 with both physics terms off for the supervised-latent arm, and the fourth ladder rung for imputation. Four full-pipeline redraws on two buildings cannot characterize the distribution of these gaps.

## G Ablations

**Coefficient of the auxiliary action term (Eq. 3).** $\lambda_{\text{act}}$ was selected before the main grid was run, on the 5-zone hot configuration at seed 42, by taking the value with the highest recovery. Recovery is a simulator rollout, so this one coefficient was chosen with environment access, whereas every other model-selection decision in the pipeline—the Stage 1 and Stage 3 checkpoints, the critic—is made from offline criteria alone. The pilot swept four values at both budgets, at a cost of eight runs; no other hyperparameter of that arm was tuned, and Stages 2 and 3 kept the settings of Table 4. Table 16 gives the sweep.

Table 16: Coefficient pilot for the supervised-latent arm: recovery on the 5-zone hot configuration at seed 42, as a percentage of the oracle's improvement. Bold marks the highest recovery at each budget. The same value wins at both, so the main grid uses one coefficient throughout and no budget-specific choice arises.

| $\lambda_{\text{act}}$ | 0.3 | 1.0 | 3.0 | 10.0 |
|---|---|---|---|---|
| recovery at 5% labels | 95.5 | **97.5** | 92.6 | 91.6 |
| recovery at full-year labels | 99.4 | **100.6** | 99.5 | 100.4 |

$\lambda_{\text{act}} = 1.0$ has the highest recovery at both budgets, so a single coefficient serves the whole sweep and the main grid uses it at every budget. Nothing in the main comparison therefore depends on tuning this arm per budget, and the arm is reported as one row of Table 1 rather than two.

The sweep is also flat where it matters: recovery moves by 5.9 points across the four coefficients at 5% and by 1.2 at a full year (Table 16), and the two smallest coefficients are within 2.0 points of the best at both budgets. The absence of further gain at a full year is therefore unlikely to be a coefficient artifact. Nor does the choice matter much on the grid: running the 5% budget at $\lambda_{\text{act}} = 10$, the value the pilot ranks last there, gives 90.2% against 89.4%. Holding the pilot configuration out of the aggregate leaves the 5% advantage over grounding-only at $+30.1$ $[20.4, 40.0]$, ahead in all eleven remaining configurations, so the advantage is not carried by the configuration the coefficient was selected on. The 5% advantage printed in Table 1 is positive in all twelve configurations, with a smallest value of $+6.2$ points; averaged within each building archetype it is $+35.9$ points for the five-zone building, $+49.4$ for the medium office, $+21.6$ for the retail shop and $+16.0$ for the warehouse. The twelve configurations are four archetypes crossed with three climates rather than twelve independent draws (§6); this breakdown says the ordering does not come from one archetype, not that the configurations are independent.

The same physics regularizer that the next paragraph tests on the grounding-only arm was also run on the supervised-latent arm at the 5% budget, with the coefficient of Table 16 and the physics terms at their configuration default of 0.5. It recovers 87.4% against 89.4% without the prior, keeps a $+28.7$ $[20.9, 37.9]$ advantage over grounding-only, and leads in all twelve configurations; with the pilot configuration held out the advantage is $+28.3$ $[20.0, 37.6]$. The prior costs this arm about two points and changes no conclusion about it.

**The tested physics-informed regularizer does not improve average performance.** Physical actuation leaves a metered signature, suggesting that tying the latent to the plant's energy balance should sharpen it. We apply the regularizer to the grounding-only arm at a full-year label budget, adding $\lambda_{\text{phy}}(\mathcal{L}_{\text{bal}}+\mathcal{L}_{\text{pow}})$ to Eq. 2 in Stage 1, where $\lambda_{\text{phy}}$ sets both physics coefficients to a common

value. For zone $i$,

$$r_{i,t}^{\mathrm{bal}} = \frac{C(\hat{T}_{i,t+1} - T_{i,t})}{\Delta t} - \left[\hat{Q}_{i,t} + \frac{q_{\mathrm{occ}} N_t}{Z} + \alpha A_i I_t + UA_i(T_{\mathrm{out},t} - T_{i,t}) + \sum_{j \in \mathcal{N}(i)} U_{ij}(T_{j,t} - T_{i,t})\right], \tag{5}$$

with $C$ the zone thermal capacitance, $\hat{Q}_{i,t}$ the forward model's per-zone heat-injection head, $q_{\mathrm{occ}} N_t / Z$ the occupant gain split over the $Z$ controlled zones, $\alpha A_i I_t$ the solar gain and $UA_i$ the envelope conductance. $\mathcal{N}(i)$ denotes the adjacent zones, $U_{ij}$ the conductance between them, and $\hat{T}_{i,t+1}$ the forward model's predicted next-zone temperature.

The power residual is $r_t^{\mathrm{pow}} = P_t - [\sum_i \max(-\hat{Q}_{i,t}, 0)/\mathrm{COP}_c + \sum_i \max(\hat{Q}_{i,t}, 0)/\mathrm{COP}_h + P_{\mathrm{aux}}]$. Here, $\mathrm{COP}_c$ and $\mathrm{COP}_h$ are the cooling and heating coefficients of performance and $P_{\mathrm{aux}}$ is auxiliary HVAC power. The thermal residual in Eq. 5 and the power residual above are divided by $Q_{\mathrm{nom}}$ and $P_{\mathrm{nom}}$, respectively. Both capacities are read from the building's design capacities in the environment configuration, and the normalized residuals are then passed through a Huber loss at $\delta$=1. Everything downstream of Stage 1 is unchanged, and each entry is a single run at seed 42.

The unregularized model wins 7 of 12 comparisons and is $2.4\%$ better on average (SE $1.1\%$). The five differences favoring the regularized model all lie below $2\%$, whereas two of the seven in the other direction exceed $9\%$. The mean difference favors the unregularized objective, but the paired test does not reject a zero mean difference ($p \approx 0.06$). This is not evidence of equivalence, and a single coefficient does not rule out that another physics prior would help. Per-configuration numbers are in Table 17.

Table 17: Physics-informed latent regularization: paired comparison over the twelve occupied-building configurations of the main analysis. $\Delta = 100\,(J_{\lambda_{\mathrm{phy}}=0} - J_{\lambda_{\mathrm{phy}}=0.5})/|J_{\lambda_{\mathrm{phy}}=0.5}|$ is the relative gain of the unregularized model over the regularized one, so positive $\Delta$ favors $\lambda_{\mathrm{phy}}$=0. The $\lambda_{\mathrm{phy}}$=0.5 column uses the checkpoint selected by the power-consistency criterion throughout; Table 18 separates the effect of that checkpoint rule from the prior itself. Mean $\Delta = +2.4\%$ (SE $1.1\%$ across the twelve paired configuration differences); paired $t$-test $p = 0.06$.

| Building | Climate | $\lambda_{\mathrm{phy}}$=0 | $\lambda_{\mathrm{phy}}$=0.5 | $\Delta$ |
|---|---|---|---|---|
| 5-zone VAV | hot | $-0.307$ | $-0.323$ | $+5.0\%$ |
| | cool | $-0.351$ | $-0.388$ | $+9.4\%$ |
| | mixed | $-0.273$ | $-0.281$ | $+2.8\%$ |
| Warehouse | hot | $-0.274$ | $-0.278$ | $+1.5\%$ |
| | cool | $-0.119$ | $-0.124$ | $+4.0\%$ |
| | mixed | $-0.167$ | $-0.164$ | $-1.6\%$ |
| Medium office | hot | $-3.028$ | $-3.008$ | $-0.7\%$ |
| | cool | $-1.389$ | $-1.423$ | $+2.4\%$ |
| | mixed | $-2.266$ | $-2.492$ | $+9.1\%$ |
| Retail shop | hot | $-0.364$ | $-0.359$ | $-1.2\%$ |
| | cool | $-3.809$ | $-3.762$ | $-1.2\%$ |
| | mixed | $-2.037$ | $-2.014$ | $-1.1\%$ |

**The two sides of that comparison also differ in checkpoint selection.** The physics residuals define the power-consistency criterion that selects the Stage 1 checkpoint, so at $\lambda_{\mathrm{phy}}$=0 that criterion is undefined and the final checkpoint is used instead: the fourth cell of the design does not exist, and Table 17 contrasts an unselected checkpoint with a selected one. We separate the two factors on the 5-zone hot configuration at three seeds by re-evaluating the $\lambda_{\mathrm{phy}}$=0.5 Stage 1 checkpoints with their final checkpoint as well (Table 18). Averaged over the three seeds, the prior alone costs $0.7$ points and checkpoint selection returns $0.9$, so the net $A - C$ contrast is $-0.2$ points: in this configuration the regularized pipeline and the unregularized one are indistinguishable. The seed-level contrasts are not small, however. The prior costs $5.0$ and $3.1$ points at two seeds and gains $6.0$ at the third, so the sign of the effect depends on the seed, while checkpoint selection is positive at every seed. Table 17 makes the same objective-and-checkpoint comparison for this configuration from the seed-42 run alone, in its relative-score metric, where it reads $+5.0\%$; that single seed is the one at which

the prior costs most. Because this decomposition covers only one configuration, we do not treat it as a general finding, and the twelve-configuration test above, with its $p$ of 0.06, remains the better summary: the prior does not help on average, and whether it hurts is not settled at one coefficient.

Table 18: Separating the physics prior from the checkpoint criterion. 5-zone VAV, hot climate, full-year label budget, three seeds. Each cell gives the score with the recovery of the oracle's improvement in parentheses; the lower block gives contrasts in recovery points, which are exactly additive because recovery is affine in the score—unlike the relative score change of Table 17. Groups B and C share the same Stage 1 training runs and differ only in whether the final or the power-consistency checkpoint is used, so $C - B$ isolates the selection criterion. The contrasts are differences of the recoveries printed above them, so a contrast may end in a different digit from the same one taken before rounding.

| Stage 1 objective / checkpoint | s42 | s123 | s2026 | mean |
|---|---|---|---|---|
| A. $\lambda_{\text{phy}}$=0, final | −0.307 (97.0) | −0.347 (86.8) | −0.316 (94.7) | −0.323 (92.8) |
| B. $\lambda_{\text{phy}}$=0.5, final | −0.326 (92.0) | −0.323 (92.8) | −0.328 (91.6) | −0.326 (92.1) |
| C. $\lambda_{\text{phy}}$=0.5, power-consistency | −0.323 (92.8) | −0.317 (94.3) | −0.327 (91.9) | −0.322 (93.0) |
| *Contrast, in recovery points* | | | | |
| $A - B$, the prior alone | +5.0 | −6.0 | +3.1 | +0.7 |
| $C - B$, the criterion alone | +0.8 | +1.5 | +0.3 | +0.9 |
| $A - C$, as compared in Table 17 | +4.2 | −7.5 | +2.8 | −0.2 |

**Seasonal label coverage in the 5-zone hot configuration (Fig. 3).** Sampled evenly across the year, a 5% budget—1,751 transitions, the count of about 18 days of labels—recovers 57.9% of the oracle's improvement and a 20% budget ($\approx$ 73 days) recovers 82.5%; collected as one contiguous block the same budgets recover −45.7% and 34.5%. Below a fifth of a year, the contiguous-window policy underperforms the logging policy, because a winter-only labeled set mistranslates latent actions for the rest of the year. The two coincide only at a full year, where both recover 93.3%—a check that the difference comes from coverage and not from anything else in the pipeline.

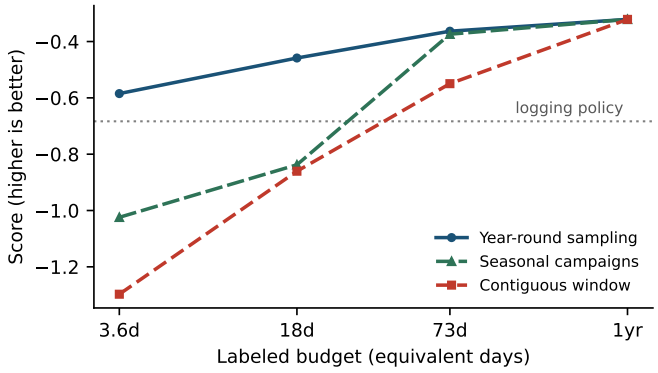



Figure 3: Labeled-data requirements for the grounding-only arm (5-zone, hot climate). At sub-annual budgets, spreading a fixed labeling budget across the year outperforms collecting it as one contiguous block, and a short contiguous window yields a policy worse than the logging policy. Four seasonal campaigns of the same total length do no better than the single window at 5% and come within three points of even sampling at 20%; all three coincide at a full-year budget, where they select the same transitions.

A third condition is the one an operator can actually schedule: the same count split into four contiguous campaigns of equal length, one per season. At the 5% budget, where each campaign lasts under five days, it recovers −39.8%, about as poorly as the single window and below the logging policy; at 20%, where each campaign lasts about eighteen days, it recovers 79.7%, within three points of even sampling. Spreading the labels over every season is therefore not sufficient by itself: four campaigns of under five days each do no better than one window of eighteen, while the same

count sampled evenly does much better. What separates the conditions at this budget is how finely the labels are spread, not whether every season is represented.

That was a grounding-only result, so it could have been a property of the decoder rather than of the sampling. Running the same four campaigns through the other two placements, at $5\%$ on three seeds, says it is the sampling: supervised latent learning falls from $95.6\%$ to $32.0\%$ and imputation from $39.9\%$ to $-119.8\%$, against the grounding-only fall from $57.9\%$ to $-39.8\%$. Every placement loses tens of points, the ordering is unchanged, and only the early placement stays above the logging policy. An operator who can label only in campaigns should expect the placement to matter more, not less. At a full year the seasonal condition reaches $93.3\%$ like the other two, which it must: at that budget all three select every transition. A budget quoted in days is a count of commands, and the seasonal condition is the one to read when the question is how long a building must be instrumented. Because this ablation uses the grounding-only arm in one configuration, the result supports spreading campaigns across seasons in that setting, provided each is long enough. For each label budget, the three conditions differ only in how the labels are distributed over the year.

## H THE BOOTSTRAP TARGET: EXACT TRANSITIONS AGAINST A FROZEN EXOGENOUS BLOCK

Building the Stage 2 transition as $(x_t, w_t) \to (x_{t+1}, w_t)$, holding the non-endogenous block at its value at $t$ instead of advancing it, is an easy thing to do and a tempting one: weather and calendar are exogenous, so freezing them looks harmless. It is not what §3 defines, since the critic then bootstraps $V(x_{t+1}, w_t)$ rather than $V(o_{t+1})$. We ran the whole benchmark both ways. The frozen block moves almost every absolute recovery in the paper, some by tens of points, and leaves the sub-annual orderings intact, which is a fact about the benchmark that neither tier of repetition (Table 2) reaches.

**What the two conventions differ in.** Only the state the critic bootstraps from. The reward is unchanged, since $r_t$ is computed from the stored next observation and depends on $x_{t+1}$ alone (Eq. 1). The latents are unchanged, since the inverse model never reads $w_{t+1}$. The policy's input distribution is unchanged, since $\pi$ maps the observation at $t$. Re-running a configuration under either convention reproduces that convention's score to the last stored digit, so the two runs differ in this and in nothing else.

The pilot of Appendix G is not itself convention-independent, which is worth saying because it is the one setting we chose rather than fixed in advance. Under the frozen block it selects $\lambda_{\text{act}} = 10$ at the $5\%$ budget, a value the exact transition ranks last of the four it swept. Table 19 holds the coefficient at $1.0$ on both sides, the exact transition's choice, so that the columns differ in the transition alone. Which coefficient a pilot selects is therefore itself a property of the convention; the frozen pilot's runs are released alongside the exact one's.

**The exact transition scores lower almost everywhere.** Table 19 gives the two conventions side by side. Against the frozen block, the exact transition lowers the oracle—which trains on ground-truth actions and touches no learned action representation—in ten of the twelve configurations, and each arm moves with the oracle of its own configuration: across the twelve, the change in the grounding-only arm's raw return tracks the change in the oracle's with $r = 0.99$. The effect is therefore a property of the offline RL stage rather than of any placement, which is why the orderings survive: at the three sub-annual budgets the three arms of Table 19 keep their rank, and supervised latent learning leads grounding-only in 12 of 12 configurations at the $5\%$ budget under both conventions. At a full year the point estimates do reorder, but both conventions leave supervised latent learning ahead of grounding-only ($+2.4$ $[0.1, 4.8]$ frozen, $+9.3$ $[1.7, 19.8]$ exact) and the imputation rung level with each of them. The imputation arm in that table is the clipped rung of Appendix D, the strongest that had been run under the frozen block; the matched rung that Table 1 reports was not run under it. The tier-B redraws of Appendix F were also re-run under the exact transition, and all four keep the ordering too, so the rank has now held up under two perturbations of different kinds: a new draw of the data and the labels, and a change to what the critic learns from.

**A mechanism the data are consistent with.** The reward depends on the endogenous block alone, so $w$ enters the value function only through its power to predict future $x$. Advancing it in the

bootstrap target adds the exogenous innovation between $t$ and $t{+}1$—weather and occupancy that no action can influence—to a quantity the critic must fit, without adding anything the reward depends on directly. The frozen block removes that variance at the cost of bias. With a memoryless critic (Table 3) and one year of transitions, the lower-variance target produced the better policy here.

The cross-configuration pattern is consistent with this reading without establishing it. The three office configurations degrade most in raw return, and the office carries 27 non-endogenous channels against 13 in the other archetypes, fifteen zones reporting to one setpoint pair. But degradation is not a function of that width alone: the cool retail configuration degrades substantially while the hot and mixed ones barely move, at equal width. Isolating the mechanism would require varying the exogenous dimension at a fixed building and reward, which we did not do, so we state it as we state the excitation mechanism of Appendix C: a reading the data support rather than one we have separated.

Table 19: Recovery under the two conventions, mean over the twelve occupied-building configurations. "Frozen" holds the non-endogenous block at $t$ in the bootstrap target; "exact" advances it, and is the convention the rest of the paper reports. The imputation row is the clipped rung of Appendix D, not the matched rung of Table 1. Almost every absolute recovery moves, some by tens of points; below a full year the ordering within each column does not change.

| Labels (fraction of one year) | 1% | 5% | 20% | 100% |
|---|---|---|---|---|
| *frozen exogenous block* | | | | |
| Supervised latent | 83.8 | 93.0 | 89.3 | 93.1 |
| Latent + decoder | 53.4 | 68.0 | 80.4 | 90.7 |
| Imputation + IQL | −223.6 | −4.0 | 70.8 | 93.9 |
| *exact transition* | | | | |
| Supervised latent | 74.0 | 89.4 | 86.8 | 93.3 |
| Latent + decoder | 36.9 | 58.7 | 71.6 | 84.0 |
| Imputation + IQL | −378.7 | −127.4 | 31.5 | 78.6 |

**What we take from it.** We report the exact transition throughout, since it is what §3 defines; it is the default in the released code, and the frozen block is reachable behind a flag so that this table can be reproduced. We do not recommend the frozen block as a technique on this evidence—one benchmark, one offline RL algorithm, and a memoryless critic that §6 already flags as a limitation. What it establishes is that the supervised-latent lead over grounding-only survives a change to the offline RL stage large enough to move almost every absolute recovery in the paper.

## I STANDARDIZING THE LABELED WINDOWS

Under joint supervision the encoder sees two kinds of history window: the action-free ones it reconstructs, standardized with $\mathcal{D}_{\text{obs}}$'s per-channel moments, and the labeled ones the auxiliary head reads. The released runs standardize the second with $\mathcal{D}_{\text{lab}}$'s own moments. $\mathcal{D}_{\text{lab}}$ is collected under wider excitation, so its moments are wider and the labeled windows reach the encoder on a slightly narrower scale than the one it was trained on. Stage 3 and the inverse model both standardize the same windows with $\mathcal{D}_{\text{obs}}$'s moments, so this arm was the one that differed.

We reran it with the labeled windows on $\mathcal{D}_{\text{obs}}$'s moments, at the 5% budget and three seeds, on the canonical configuration and on the one where the arm recovers least. The change is $-0.2$ points on 5-zone hot and $+1.0$ on warehouse hot, with per-seed values of $-1.0$, $+0.1$, $+0.2$ and $+3.2$, $+0.5$, $-0.8$. It is smaller than the seed-to-seed spread in both configurations and its sign is not the same in the two, so on the two configurations and the one budget we tested the effect is small and not in a consistent direction. That is weaker than a claim about the whole grid, which we did not rerun. The tables keep the released runs, and `train_stage1.py` takes `--dlab-stats obs` for the other convention.

## J WHAT THE ACTION-FREE CORPUS IS WORTH

The labeled-only control of §5 clones $\mathcal{D}_{\text{lab}}$. That answers whether the labeled budget can be imitated, not whether it can be improved on, and the second question is the one to ask of a pipeline whose premise is the corpus behind the labels. So we ran the offline RL stage on the labeled budget and nothing else: the same $1{,}751$ transitions the other arms receive at $5\%$, with their ground-truth actions, the same observation normalization, the same reward, and the same critic at the same seed. No $\mathcal{D}_{\text{obs}}$, no latent, no imputation.

It fails in every configuration. Mean recovery is $-271.9\%$, median $-116.6$, and the range runs from $-34.3$ to $-1121.7$; it is below the logging policy in twelve of twelve. Every placement beats it by a margin whose interval is far from zero—supervised latent by $+361.3$ $[206.5, 564.5]$, grounding-only by $+330.6$ $[181.4, 524.3]$, imputation by $+308.4$ $[165.1, 480.6]$, each ahead in twelve of twelve.

The three offices fail worst ($-670.8$, $-1121.7$, $-488.7$), where the oracle's headroom is largest and a critic fitted to $1{,}751$ transitions has the most room to be wrong. This is not a claim about offline RL at this sample size in general: it is what happens when the $35{,}040$ action-free transitions are withheld and everything else is held fixed. Where the labels go matters, as the rest of the paper reports; that they arrive alongside a corpus matters more.

## K A BIASED, RARELY ADJUSTED LOG

Every log in the main grid comes from a rule-based controller perturbed with i.i.d. Gaussian noise at each step, and §6 names what that leaves untested: a real operator neither perturbs the setpoint every fifteen minutes nor leaves the controller unbiased, and much of the headroom such a log offers is the noise itself. This appendix runs the three placements on one log with the opposite properties.

The 5-zone hot configuration is re-collected under the same seasonal RBC with a constant bias of $+1°$C on the heating setpoint and $-1°$C on the cooling one, so that the controller over-conditions all year and no amount of averaging recovers the setpoint it should have used; with an adjustment drawn every ten days on average and held until the next; and with the per-step noise on $\mathcal{D}_{\text{obs}}$ reduced from $\sigma$=1.0 to 0.3°C, while $\mathcal{D}_{\text{lab}}$ keeps the main grid's $\sigma$=2.5. The log is therefore still perturbed at every step, only less so, and it is not a noise-free or an operator's log. Everything else is the main grid's: the same three placements at the same coefficients, three seeds, the $5\%$ and full-year budgets, the same oracle construction.

The headroom this log offers is small. The logging policy scores $-0.374$ and the oracle $-0.351$, a gap of $0.023$ against $0.388$ on the main grid's 5-zone hot: a controller that over-conditions is already good on comfort, and an oracle that only ever sees biased actions cannot improve on it by much. Recovery as a percentage of that gap is unstable, so Table 20 reports raw returns.

Table 20: Raw return under the operator-like log on 5-zone hot, mean of three seeds with the per-seed values in parentheses; higher is better. The logging policy scores $-0.374$ and the oracle $-0.351$.

| | 5% | 100% |
|---|---|---|
| Supervised latent | $-0.331$ ($-0.324$, $-0.330$, $-0.340$) | $-0.311$ ($-0.311$, $-0.313$, $-0.309$) |
| Latent + decoder | $-0.457$ ($-0.346$, $-0.432$, $-0.592$) | $-0.371$ ($-0.360$, $-0.401$, $-0.353$) |
| Imputation | $-0.558$ ($-0.510$, $-0.587$, $-0.576$) | $-0.312$ ($-0.306$, $-0.312$, $-0.320$) |

Three things hold and one does not. The early placement leads grounding-only at both budgets in every seed, by $+0.126$ at $5\%$ and $+0.060$ at a full year, and leads imputation at $5\%$ in every seed, by $+0.227$; it is the only placement above both the oracle and the logging policy at both budgets, and its seed-to-seed spread at $5\%$ is $0.016$ against $0.246$ for grounding-only. At a full year the early placement and imputation are level, $0.002$ apart with the sign changing across seeds, as on the main grid. What does not hold is the relation between the two late placements: on the main grid they converge by a fifth of a year, whereas here grounding-only falls below the logging policy at $5\%$ in two seeds of three and imputation is ahead of it at a full year in three of three, by $+0.059$. What the data show is a grounding-only arm that is both weaker and far less stable across seeds on this log;

one reading is that the unsupervised latent it relies on is what the log destabilizes, and supervising that latent is what the early placement does, but the experiment does not isolate that.

This is one configuration and one alternative log, chosen to have the properties the main grid's lacks rather than to stand for operators in general. It says that the early placement's lead does not depend on the log being the main grid's, and that the ordering of the two late placements does. Bias, held adjustments and the noise scale all change together here, so it does not say which of them matters.