

# Evaluating Performance of Reinforcement Learning Agents to Control Buildings Efficiently

Judah Goldfeder  
Columbia University  
jag2396@columbia.edu

Gabriel Guerra Trigo  
Columbia University  
ggt2112@columbia.edu

Philippe Martin Wyder  
University of Washington  
pwyder@uw.edu

Neil Kachappilly  
Columbia University  
nbk2122@columbia.edu

Hod Lipson  
Columbia University  
hod.lipson@columbia.edu

## Abstract

The recently introduced Smart Buildings Control Suite provides an open-source, physics-informed simulator for developing building HVAC control agents, yet its initial presentation lacked comprehensive empirical performance results. This paper addresses this gap by presenting a quantitative benchmark evaluation of standard reinforcement learning (RL) algorithms within this specific simulation environment. Our primary objective is to establish performance characteristics and demonstrate the suite’s capability for evaluating modern control strategies. We train and evaluate Soft Actor-Critic (SAC) and Deep Deterministic Policy Gradient (DDPG) agents using the suite’s simulator configured for Building SB1. The evaluation focuses on learning efficiency, final control performance compared to a baseline schedule, and, critically, generalization across diverse seasonal conditions not seen during training. Additionally, with the goal of encouraging the adoption of control policies by real buildings, we analyze the performance implications of a policy extraction technique which aggregates agents’ policies into predictable, static schedules which can be easily implemented in buildings. Our results provide quantitative evidence that both SAC and DDPG agents can be effectively trained within the suite, achieving significantly better performance than the baseline policy, and successfully generalizing across different seasons. Furthermore, the analysis shows policy extraction incurs minimal performance loss for up to 8-hour aggregation bins, demonstrating that there is opportunity for simpler, more interpretable policies, which would still benefit performance. This work establishes initial performance benchmarks for the Smart Buildings Control Suite, validating its use for RL research, and motivating further research.

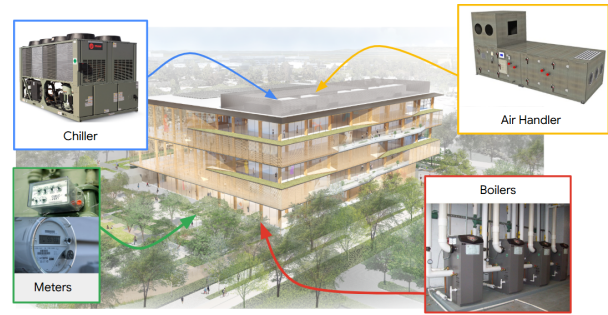


Figure 1. Illustration of an office building and its HVAC devices. Image credit to John Sipple.

## 1. Introduction

Energy optimization in commercial buildings is a crucial and increasingly urgent issue. Buildings contribute approximately 37% of total US carbon emissions, with commercial buildings alone accounting for 17% in 2023 [1]. Within buildings, Heating, Ventilation, and Air Conditioning (HVAC) systems (see Fig. 1) specifically account for about 40-60% of their energy usage [2], which translates to roughly 15% of global energy consumption [3]. Optimizing HVAC systems thus represents a particularly impactful opportunity for reducing overall carbon emissions. Optimizing HVAC control has been an active research area for decades [4–12], and yet, while AI has begun to revolutionize many industries, almost all HVAC systems remain the same as they were many decades ago. Despite extensive literature on the topic, broad real-world adoption remains elusive. One of the biggest factors limiting progress is the lack of a robust public benchmark to facilitate research and measuring progress. The recently introduced Smart Buildings Control Suite [13, 14] aims to address this, and provides an open-source, lightweight physics-informed simulator de-

signed as a Gym-compatible environment for developing reinforcement learning (RL) agents for building HVAC control. While this suite represents a valuable contribution, the original work presented limited empirical results regarding the actual training and performance of RL agents within this specific simulation framework.

In addition, a key challenge in applying such learned RL agents, often complex black-box models, to real-world buildings involves trust and ease of implementation [15]. Operators may be hesitant to deploy controllers whose decision-making is opaque and requires continuous interaction with the building's system. Addressing this deployment barrier, recent work has explored **policy extraction**[15]. The proposed workflow aimed to bridge the gap between complex RL agents and practical deployment by generating simpler, interpretable schedules. This was done as follows:

1. Obtain a **weather forecast** for a future period (e.g., the upcoming week).
2. Use this forecast to drive a **dynamics model** (specifically, an LSTM model in their study) simulating the building's response.
3. **Train an RL agent** interactively within this simulated, forecast-driven environment to learn an optimal control policy for that specific future period.
4. Simulate the trained RL agent operating over the forecast period and **record its actions**.
5. **Process these recorded actions** (e.g., by averaging over time intervals) to **extract a static, human-readable schedule** (e.g., a table of setpoints vs. time).
6. Finally, **deploy this extracted schedule** onto the actual building, rather than deploying the interactive RL agent itself.

However, several critical analyses were absent in that study [15]. Firstly, while employing an LSTM-based dynamics model (diverging from the physics-informed simulator), it crucially **did not present quantitative comparisons** between the extracted schedule's performance and baseline building operation policies. Secondly, the potential **performance degradation inherent in the policy extraction step itself**—resulting from aggregating continuous agent actions into coarser, discrete schedule steps—was not evaluated. Finally, the study **did not investigate the robustness** of the extracted schedules to inevitable deviations between the weather forecast used for generation and the actual experienced weather.

Therefore, a foundational evaluation of RL performance *within the simulator* remains necessary, providing the basis for addressing these gaps. The present study utilizes the simulator component of the suite, configured for Building SB1, to conduct a comprehensive performance benchmark of standard RL algorithms (SAC and DDPG). We assess their learning efficiency, final control performance, and, importantly, their generalization capabilities across diverse sea-

sonal conditions. **This provides the direct, quantitative comparisons against baselines and the analysis of generalization which were missing, thereby addressing key gaps identified in prior work [15]. Our results establish crucial performance characteristics within the physics-based simulator, offering a foundation upon which the performance impacts of extraction techniques and sensitivity to forecast errors can be subsequently evaluated.**

## 2. Objectives

To demonstrate the capability of the Smart Buildings Control Suite as a benchmark for modern control algorithms, we conduct a series of experiments evaluating the performance of some reinforcement learning algorithms. Our objectives here are:

- Compare the learning efficiency and converged performance of SAC, DDPG, and TD3 agents on the HVAC control task.
- Assess the generalization capabilities of these agents when trained and evaluated under different environmental conditions (specifically, varying seasons defined by episode start dates).
- Quantify the potential energy savings and comfort improvements achieved by trained RL agents compared to a standard baseline controller.

It is important to note that all experiments presented in this section were conducted using the developed simulation environment, not through live trials on physical buildings. Consequently, the reported performance improvements demonstrate the potential gains achievable within the simulated environment, predicated on the simulator's fidelity to real-world dynamics. Although the simulator was calibrated using historical data [15], validating these findings through real-world deployment remains a critical next step for future work.

## 3. Experimental Setup

All reinforcement learning experiments were conducted using the lightweight physics-informed simulator detailed in Section 5, configured to emulate Building SB1 from the dataset described in Section 4.

An experimental "episode" corresponds to a simulation run over a specific historical period (e.g., July 7th to July 20th, 2023), utilizing the recorded outdoor air temperatures for that duration. While the agent's actions influence the building's internal state, the external weather conditions for a given episode definition remain fixed across all runs and agents, ensuring reproducible environmental factors. In this context, we use the terminology "train episode" and "test episode" to denote ranges of dates that were or not used during training. The simulation operates with a 15-minute timestep resolution.

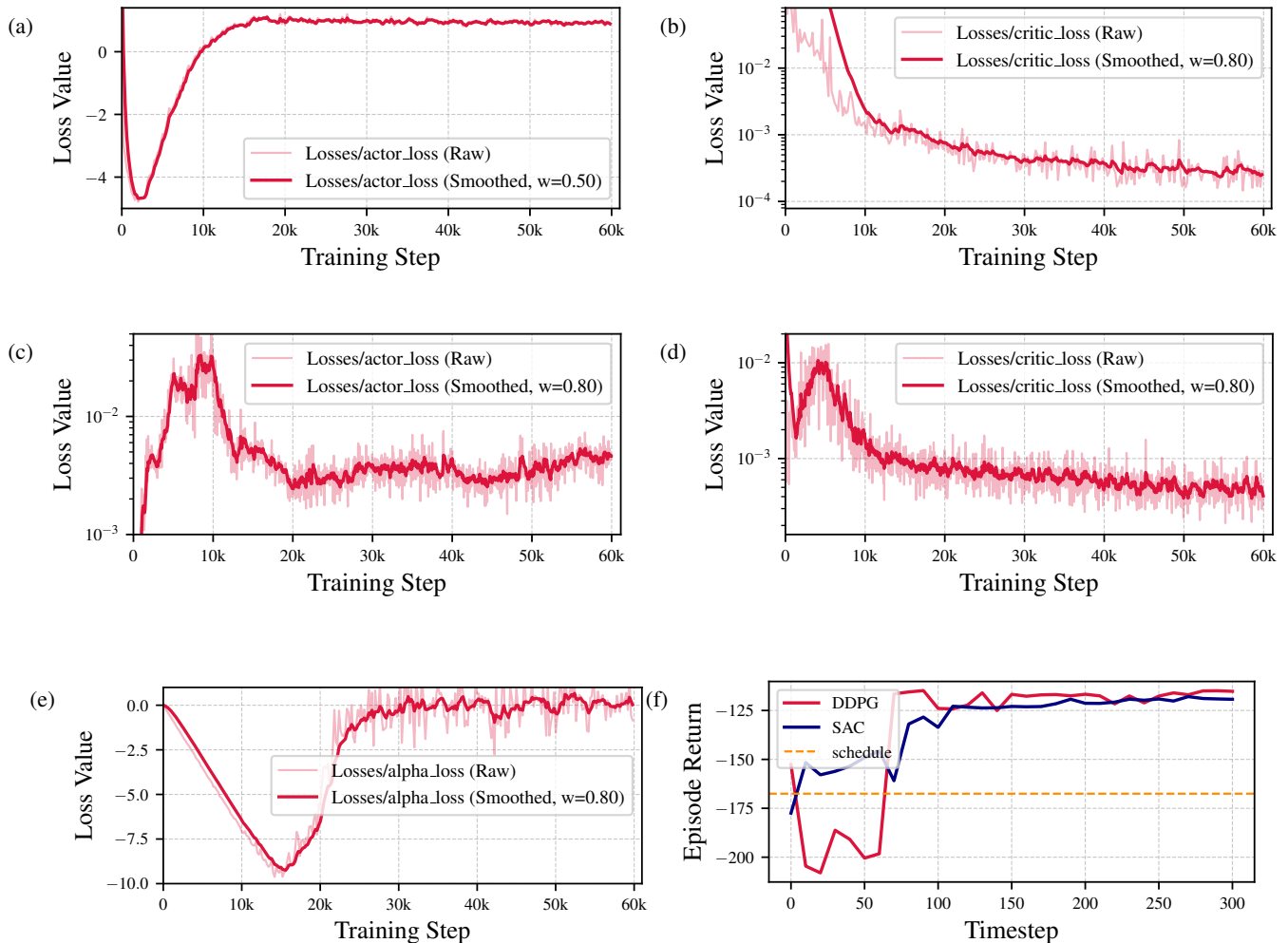


Figure 2. Plots describing the training process of SAC and DDPG agents. Plots (a) through (e) show training losses, with loss values plotted against the training step, while plot (f) shows the evolution of the agents’ validation episode return throughout training. Plot (a) through (c) show the SAC agent’s actor, critic, and alpha losses respectively. Plots (d) and (e) show the DDPG agent’s actor and critic losses respectively.

To ensure a fair comparison between TD3, SAC, and DDPG, a common set of training hyperparameters was employed across all agents and random seeds. The training procedure involved 300 main iterations. Within each iteration, 50 new environment steps were collected using the agent’s current policy, and 200 gradient update steps were performed on sampled data batches. Training began with a replay buffer pre-populated with 6,720 transitions (equivalent to 7 days) generated by executing the baseline building schedule policy. relevant hyperparameters included a batch size of 256 and a learning rate of  $4 \times 10^{-4}$  (applied uniformly to relevant network components). An evaluation run was performed every 10 training loop iterations.

#### 4. Learning Framework Verification

Prior to assessing generalization performance, we conducted a preliminary experiment to verify the fundamental learning capability of the agents within our simulation framework, and to validate that the hyperparameters chosen were sensible for the setup. For this verification step, SAC and DDPG agents were both trained and subsequently evaluated exclusively on a single, fixed 14-day episode commencing July 6th. While this train-on-test configuration inherently precludes any claims of generalization, it confirms that the agents can successfully optimize the defined reward signal.

The evaluation results indicate that both the SAC (final episode return: -119.2) and DDPG (final episode return: -115.2) agents developed policies that substantially outper-

formed the baseline schedule (final episode return: -167.5). Moreover, analysis of the evolution of the episode return curves seem to indicate successful optimization: the episodic returns during training increased steadily before reaching a plateau for both agents, around the same value. The convergence of distinct algorithms to similar performance levels suggests consistent optimization behavior and potentially indicates the identification of a near-optimal policy for this fixed environmental condition.

## 5. Generalization Across Varying Conditions

To assess the generalization capabilities of the trained agents, we then evaluated their performance on held-out episodes representing environmental conditions distinct from the training data. A critical aspect of robust HVAC control is the ability to adapt to significantly different scenarios, particularly those driven by seasonal weather changes. Evaluating an agent trained in one period on a temporally adjacent period within the same season (e.g., consecutive weeks in July) might offer limited insight into generalization, as the environmental conditions could be highly correlated. Therefore, a more rigorous assessment involves evaluating across periods with substantially divergent conditions.

With this rationale, we designed the following experiment:

- Agents were trained on a single 14-day episode representing summer conditions (commencing July 6th).
- Evaluation was conducted on four distinct 14-day episodes starting progressively later in the year: August 6th, September 6th, October 6th, and November 6th, thereby capturing the transition from late summer through autumn.
- The performance of the trained RL agents on each evaluation episode was compared against the baseline schedule policy.

This setup allows us to specifically examine how well policies learned under summer conditions generalize to the increasingly different thermal demands of subsequent months.

Figure 3 illustrates the generalization performance of the SAC and DDPG agents compared to the baseline Schedule policy on four distinct evaluation episodes, commencing monthly from August 6th to November 6th, following training on a July episode. The results clearly demonstrate successful generalization: both the SAC and DDPG agents achieved substantially higher episodic returns than the baseline Schedule policy across all four evaluation periods, even the episodes in later months, with considerably different weather scenarios than the training episode. Notably, the absolute performance (total episodic return) decreased progressively for *all* policies from the August to the November episode. This trend aligns with expected seasonal changes; as outdoor temperatures drop, the heating load on the HVAC system increases, inherently leading to higher energy con-

sumption and potentially greater comfort challenges, reflected in lower reward values.

Comparing the two RL agents, SAC and DDPG exhibited broadly comparable performance levels throughout the evaluation periods. DDPG achieved marginally higher returns in the September, October, and November episodes, while SAC performed slightly better in August, although the differences remained relatively small. Overall, the consistent outperformance relative to the baseline across these diverse and temporally distant episodes confirms the ability of the learned RL policies to generalize effectively beyond the specific conditions encountered during training.

As shown in Figures 3(c) and 3(d), SAC and DDPG agents, although trained independently, yielded policies exhibiting marked similarities (e.g., comparable action peak timing) on the November 6th out-of-sample test episode. Such consistency between policies derived from different algorithms suggests the training effectively converged towards robust strategies, thus validating the training procedure.

## 6. Extracting and Evaluating Interpretable Policies

As discussed in [15], deploying complex learned controllers can face practical hurdles related to operator trust, interpretability, and potentially unpredictable behavior. The policy extraction technique proposed therein—generating simpler, static schedules by aggregating an RL agent’s actions over time—aims to mitigate these deployment challenges. However, this simplification introduces a potential performance trade-off. A critical question remains: **how significantly does the performance degrade when converting a dynamic RL policy into a coarser, schedule-based one?** Does the extracted policy retain most of the original agent’s effectiveness, and critically, does it still outperform standard baseline schedules?

This section empirically investigates this performance degradation using the DDPG agent previously trained on a single 14-day episode commencing July 6th. We generated static, schedule-based policies by applying temporal action aggregation to this agent’s behavior on specific evaluation episodes. The core procedure for deriving and evaluating each extracted schedule was as follows:

1. The pre-trained DDPG agent was executed on a designated 14-day evaluation episode.
2. The action sequence produced by the agent during this run was recorded.
3. This action sequence was temporally aggregated into discrete bins of a fixed duration (the bin size) to create a static schedule. Actions within each bin were averaged (or aggregated according to a defined rule, e.g., majority vote for discrete actions - specify if needed).
4. The resulting static schedule was then executed on the

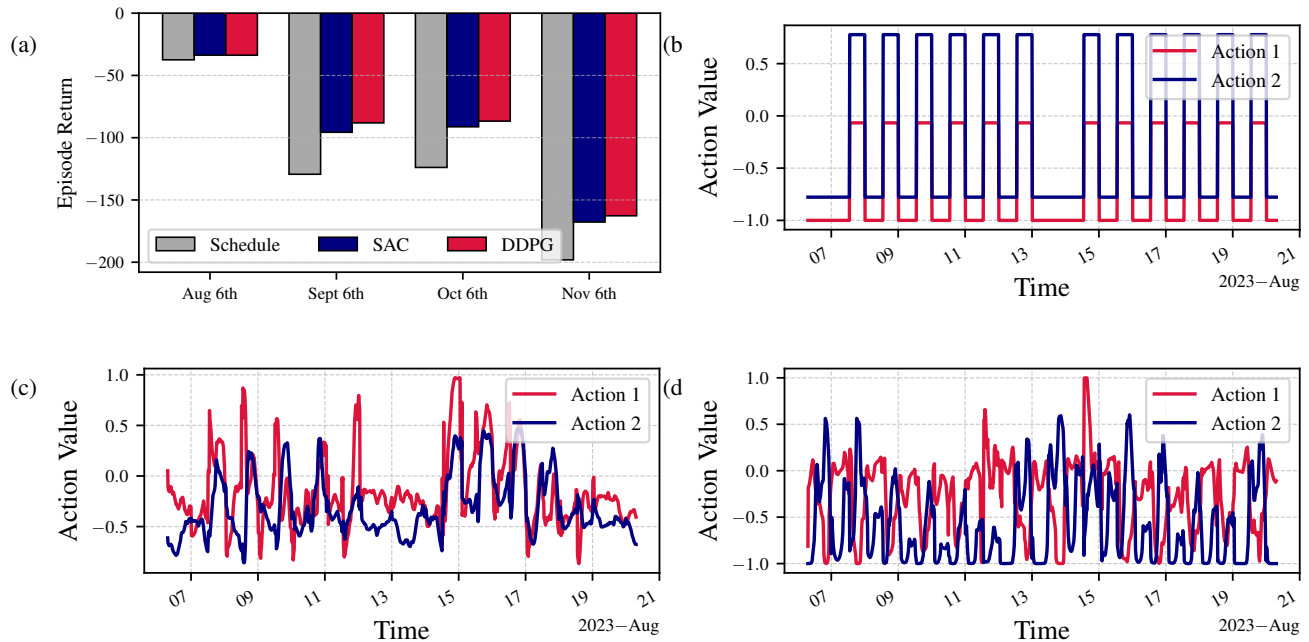


Figure 3. Plot (a) shows the performance comparison between the SAC, DDPG and baseline schedule policies for 4 different 14-day evaluation episodes, starting on August 6th, September 6th, October 6th and November 6th. Plots (b) through (d) show the actions taken by the schedule, SAC and DDPG agents respectively, when those were evaluated in the 14-day episode starting on November 6th.

same evaluation episode from which its generating action sequence was derived, and its performance (episodic return) was recorded.

We performed this extraction and evaluation process for all combinations using four distinct 14-day evaluation episodes (commencing August 6th, September 6th, October 6th, and November 6th) and four different temporal aggregation bin sizes (2 hours, 4 hours, 8 hours, and 168 hours [or one week] as a sanity check). The performance of these resulting 16 extracted schedules was then compared against both the performance of the original interactive DDPG agent and a baseline schedule policy on the corresponding evaluation episodes. Figure 4(a) presents these comparative results.

Evaluation of the temporally aggregated policies reveals two significant insights:

- First, increasing the temporal aggregation bin size up to 8 hours, which simplifies the policy and smooths control actions, does not substantially compromise control performance. The reduction in the achieved return value was minimal relative to the original agent and significantly less than the performance deficit of the baseline policy. This tolerance to aggregation aligns with the characteristically slow dynamics (high thermal inertia) of building thermal systems, rendering them relatively insensitive to the high-frequency control actions removed by the aggrega-

tion process.

- Second, the performance advantage over the baseline was maintained even with extreme aggregation using 168-hour (weekly) bins. That such a simplified, static weekly schedule derived from the agent surpasses the baseline policy indicates a significant inefficiency in the baseline itself. This large performance gap underscores the baseline’s limitations and suggests that developing improved, yet simple, heuristic schedules should be achievable and merits further exploration.

## 7. Conclusion

This work presents an important contribution to the Smart Buildings Control Suite: Benchmarking RL algorithms, and exploring policy extraction into fixed schedules. This represents a step towards buildings that can operate more efficiently and with reduced carbon emissions.

## References

- [1] EIA, “Frequently Asked Questions (FAQs) - U.S. Energy Information Administration (EIA) — eia.gov.” <https://www.eia.gov/tools/faqs/faq.php?id=86&t=1>. [Accessed 04-06-2024]. 1
- [2] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review

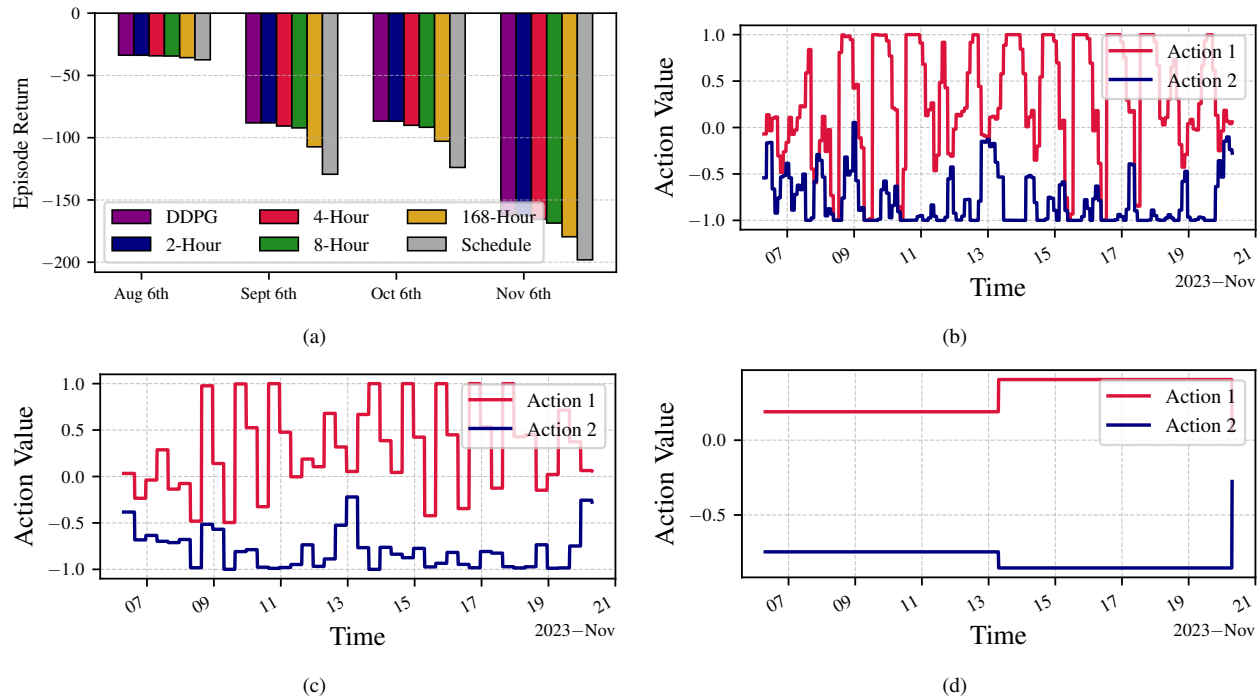


Figure 4. Policy extraction evaluation. Plot (a) compares degradation as the bin size of policy extraction increases. Plots (b)–(d) show extracted schedules with different bin sizes.

on buildings energy consumption information,” *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008. 1

- [3] N. Asim, M. Badiei, M. Mohammad, H. Razali, A. Rajabi, L. Chin Haw, and M. Jameelah Ghazali, “Sustainability of heating, ventilation and air-conditioning (hvac) systems in buildings—an overview,” *International journal of environmental research and public health*, vol. 19, no. 2, p. 1016, 2022. 1
- [4] K. F. Fong, V. I. Hanby, and T.-T. Chow, “Hvac system optimization for energy management by evolutionary programming,” *Energy and buildings*, vol. 38, no. 3, pp. 220–231, 2006. 1
- [5] M. Trčka and J. L. Hensen, “Overview of hvac system simulation,” *Automation in construction*, vol. 19, no. 2, pp. 93–99, 2010.
- [6] P. Riederer, “Matlab/simulink for building and hvac simulation-state of the art,” in *Ninth International IBPSA Conference*, pp. 1019–1026, 2005.
- [7] C. Park, D. R. Clark, and G. E. Kelly, “An overview of hvacsim+, a dynamic building/hvac/control systems simulation program,” in *Proceedings of the 1st Annual Building Energy Simulation Conference, Seattle, WA*, pp. 21–22, 1985.
- [8] M. Trčka, J. L. Hensen, and M. Wetter, “Co-simulation of innovative integrated hvac systems in buildings,” *Journal of Building Performance Simulation*, vol. 2, no. 3, pp. 209–230, 2009.
- [9] A. Husaunndee, R. Lahrech, H. Vaezi-Nejad, and J. Visier, “Simbad: A simulation toolbox for the design and test of hvac control systems,” in *Proceedings of the 5th international IBPSA conference*, vol. 2, pp. 269–276, International Building Performance Simulation Association (IBPSA) Prague . . . , 1997.
- [10] M. Trcka, M. Wetter, and J. Hensen, “Comparison of co-simulation approaches for building and hvac/r system simulation,” in *10th International IBPSA Building Simulation Conference (BS 2007), September 3-6, 2007, Beijing, China*, pp. 1418–1425, 2007.
- [11] M. Blonsky, J. Maguire, K. McKenna, D. Cutler, S. P. Balamurugan, and X. Jin, “Ochre: The object-oriented, controllable, high-resolution residential energy model for dynamic integration studies,” *Applied Energy*, vol. 290, p. 116732, 2021.
- [12] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, *et al.*, “Energyplus: creating a new-generation building energy simulation program,” *Energy and buildings*, vol. 33, no. 4, pp. 319–331, 2001. 1
- [13] J. A. Goldfeder and J. A. Sipple, “A lightweight calibrated simulation enabling efficient offline learning for

optimal control of real buildings,” in *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 352–356, 2023. [1](#)

- [14] J. Goldfeder, V. Dean, Z. Jiang, X. Wang, H. Lipson, and J. 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*, 2025. [1](#)
- [15] J. Goldfeder and J. Sipple, “Reducing carbon emissions at scale: Interpretable and efficient to implement reinforcement learning via policy extraction,” in *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 403–407, 2024. [2](#), [4](#)