# Learning Adaptive Mixed-Mode Ventilation Policies via Adversarial Inverse Reinforcement Learning

**Wei Liang** [1] **Adrian Chong** [1]

## Abstract

Mixed-mode ventilation (MMV) control presents a complex decision-making problem due to highly variable outdoor conditions and the need to balance natural ventilation with mechanical cooling. We propose a novel Adversarial Inverse Reinforcement Learning framework for MMV that tackles this complexity by jointly learning a reward function and an adaptive policy from building operational data. Our approach incorporates a physics-constrained neural network model of the MMV environment and a hierarchical policy structure, enabling effective handling of discrete window operations alongside continuous HVAC control. The results show that the learned policy reliably captures the window operation patterns from the rule-based control demonstration, while reducing unnecessary window switching. In addition, the learned policy reduced the temperature comfort range violation from 1.7% to 0.4% compared to the rule-based control. The results demonstrate that the Adversarial Inverse Reinforcement Learning framework can achieve energy-efficient MMV control with significantly fewer window adjustments, thus improving occupant comfort and system stability compared to conventional or heuristic strategies.

## 1. Introduction

Climate change exacerbates heat stress, driving increased air-conditioning (AC) usage, which in turn further accelerates global warming. Mixed-mode ventilation (MMV), combining natural ventilation (NV) and mechanical cooling, mitigates reliance on AC and balances energy efficiency with thermal comfort. However, MMV controls have a complex nature due to dynamic outdoor conditions, requiring robust handling of energy efficiency, thermal comfort, and occupant behavior.

We introduce an adversarial inverse reinforcement learning (AIRL) (Fu et al., 2017) framework to address the issues as follows: (1) learns a latent reward capturing energy, comfort, and window operations from historical rule-based control data, (2) employs a hierarchical actor-critic that decides change vs. stay before issuing continuous set-points, and (3) embeds a physics-constrained CNN-LSTM surrogate model for the MMV environment. By unifying reward inference and policy optimization while respecting thermodynamics, our approach directly addresses the open issues left by prior RL studies for MMV, enabling stable and realistic mixed-mode control in both simulation and hardware-in-the-loop experiments.

## 2. Related work

Reinforcement learning (RL) faces significant challenges in MMV control, particularly in designing the reward function. The traditional design of reward functions combines multiple weighted objectives (Chen et al., 2018; An & Chen, 2023), such as energy consumption, thermal comfort, and indoor air quality. However, the reward function design for MMV systems often overlooks interruptions to occupants due to frequent window operations, outdoor noise (Peng et al., 2023), and additional intricacies of occupant comfort. Recent works address this using imitation learning (IL) and inverse reinforcement learning (IRL). Techniques such as behavior cloning (Silvestri et al., 2025) and adversarial methods like Generative Adversarial Imitation Learning (GAIL) (Liu et al., 2024; Hu et al., 2025) have demonstrated improved efficiency and stability (Dey et al., 2023) by leveraging expert demonstrations, thereby reducing the complexity of manual reward tuning (Giraldo-Pérez et al., 2025).

## 3. Methodology

The diagram of the proposed adversarial inverse reinforcement learning for mixed-mode ventilation systems is illus-

[1]Department of the Built Environment, College of Design and Engineering, National University of Singapore, Singapore. Correspondence to: Wei Liang <wliang@nus.edu.sg>, Adrian Chong <adrian.chong@nus.edu.sg>.

*Proceedings of the $42^{nd}$ International Conference on Machine Learning*, Vancouver, Canada. PMLR 267, 2025. 

trated in Figure 1. The surrogate data-driven environment and expert trajectories were derived from real-world data collected at one-minute intervals from a mixed-mode ventilated open office in the tropics. The AIRL framework performs an inner–outer learning loop. At each AIRL iteration, we (1) roll out the policy trajectory, (2) update rewards using the expert trajectory, and (3) improve the policy via PPO. The complete algorithm is in Appendix A.

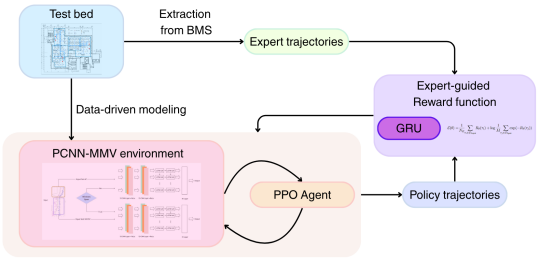


*Figure 1.* Diagram of the proposed adversarial inverse reinforcement learning framework. The PCNN-MMV environment is introduced in Section 3.1, the reward function learning is in Section 3.3, the PPO agent is in Section 3.4.

### 3.1. Physics-Constrained Neural Network Model

MMV systems operate under various conditions, and each condition exhibits distinct behaviors. Therefore, it is difficult to construct a sole data-driven model that can work for both modes. To address this issue we introduced a data-driven physics-constrained neural network (PCNN) to serve as the environment of the AIRL framework. The PCNN treats a mixed-mode system as two thermodynamically distinct regimes, AC and NV modes, and utilize a gate to pick the corresponding surrogate dynamics at each time step. The window status $a_w \in 0, 1$ is a discrete switching variable representing the active regime at time $t$. The assumption is that there is a distinct sub-model associated with each regime. Each model branch features a Convolutional Neural Networks-Long Short Term Memory (CNN-LSTM) architecture, where the convolutional layer captures the spatial patterns between different thermal zones and the LSTM layer extracts the temporal dependencies.

### 3.2. Mixed-Mode Surrogate Environment Design

Our MMV environment comprises controllable (zone temperatures, window status history), exogenous (outdoor temperature, wind), and actionable (window signals, HVAC setpoints) states. Actions involve window operation signals and setpoint adjustments for mechanical cooling via fan coil units (FCUs) and dedicated outdoor air (DOAS) units. The DOAS system can serve as a supplement to personalized cooling when windows are open, as well as provide necessary ventilation in mechanical cooling mode. Both types of units are constant volume.

*Table 1.* State variables used in the MMV MDP. All continuous quantities are normalized to $[0, 1]$ before entering the network.

| **Controllable State** | **Unit** | **Range** |
|---|---|---|
| $T_{t,i}^{\text{zone}}$, zone temperature | °C | $[23, 31]$ |
| $t^{\text{win}}$, time since last window flip | min | $[0, 720]$ |
| $\text{dwell}_t = \frac{1}{\max(1, t - t^{\text{win}})}$ | $\text{min}^{-1}$ | $(0, 1]$ |
| $\bar{w}_{t-1}$, previous window status | – | $\{0, 1\}$ |
| **Exogenous State** | **Unit** | **Range** |
| $T_t^{\text{out}}$, outdoor temperature | °C | $[23, 34]$ |
| $v_t^{\text{wind}}$, wind speed | $\text{m s}^{-1}$ | $[0, 10]$ |
| $\phi_t^{\text{wind}}$, wind direction | deg | $[0, 360]$ |
| **Action** | **Unit** | **Range** |
| $a_t^{\text{win}}$, window operate signal | - | $\{0, 1\}$ |
| $T_{t,i}^{\text{fcu-sa}}$, FCU supply temperature | °C | $[18, 31]$ |
| $T_{t,j}^{\text{doas-sa}}$, DOAS supply temperature | °C | $[18, 31]$ |

Custom window states include normalized timing since the last state change, dwell time, and previous window status, preventing frequent window flips (Dai et al., 2023) without imposing window lockout or delay time as a penalty (An & Chen, 2023). They encode information explicitly related to window operation history and dynamics. Thereby, the proposed framework remains an IRL approach without imposing prior domain knowledge. The window variable states are as follows: time since last flip $t_{\text{win}}$; $\text{dwell}_t = \frac{1}{\max(1, t - t^{\text{win}})}$, quantifies the recency of window state changes. It's already normalized to $(0, 1]$ by its form. A smaller value of $\text{dwell}_t$ indicates a longer recency of window states, which is contrary to $t^{\text{win}}$; previous window state ($\bar{w}_{t-1}$). By explicitly capturing the temporal and historical dynamics of window operations, expert demonstrations can guide RL policies to learn towards stability, achieving energy and thermal comfort goals while recognizing the window operation issue.

### 3.3. Adversarial Reward Learning with AIRL

Our AIRL framework learns the underlying reward function as a neural network from the expert behavior. We leverage historical Building Management System (BMS) data following an RBC policy as expert demonstrations. In the NV mode, the RBC policy regulates the window opening when the outdoor temperature $T_{out} < 29°\text{C}$. If the outdoor temperature becomes warmer ($29,°\text{C} < T_{\text{outdoor}} \leq 31,°\text{C}$), DOAS provides personalized cooling to the occupants. In full AC mode ($T_{out} > 31°\text{C}$) or during rain or haze, the windows are fully closed, and the system switches to mechan-

ical cooling with FCUs and DOAS units running. These logged trajectories inherently encode the building operator's strategy for MMV. In addition to the indoor temperature comfort range, the BMS trajectories also contain information, such as time-delays after mode switches and window operations, to avoid frequent and unnecessary opening/closing of the window and short cycling of the mechanical systems.

Let $\mathcal{D}_{\mathrm{E}} = \{\tau_{\mathrm{E}}\}$ denote expert trajectories from rule-based control (RBC) over days of operation, and $\mathcal{D}_{\pi} = \{\tau_{\pi}\}$ be the trained policy trajectories. We recover a reward function $r_{\psi}(s, a)$ such that the expert policy appears optimal under $r_{\psi}$. We parametrize $r_{\psi}$ with a discriminator, specifically a Gated Recurrent Unit (GRU) that processes sequences of states and actions, outputting a likelihood that a given trajectory is from the expert. The GRU architecture allows $r_{\psi}$ to capture temporal dependencies. We adopt the maximum-causal-entropy (MaxEnt) principle (Ziebart et al., 2008) to make sure among all trained policy trajectory distributions that match the expert's feature expectations by choosing the one with the highest causal entropy. The loss function is as follows to maximize causal entropy (Finn et al., 2016):

$$\mathcal{L}(\psi) = \frac{1}{N_E} \sum_{\tau_i \in \mathcal{D}_{\mathrm{E}}} r_{\psi}(\tau_i) + \log \frac{1}{N_A} \sum_{\tau_j \in \mathcal{D}_{\pi}} \exp(-r_{\psi}(\tau_j)), \tag{1}$$

where $N_E$ is the data points of daily trajectories of RBC demonstrations, and $N_A$ is that of the policy trajectories, $\psi$ is the parameter to be optimized for the GRU-based reward network.

### 3.4. Hierarchical Policy and PPO Optimization

Given the learned reward network $r_{\psi}(s, a)$, we employ a Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) to find a hierarchical policy (Pateria et al., 2022) $\pi_{\theta}$ that maximizes the expected return in stable iterations. PPO is a policy-gradient method that uses a clipped surrogate objective to prevent overly large updates. In our setting, we use PPO to iteratively improve the hierarchical policy $\pi_{\theta}(a_t^{\text{win}}, u_t | s_t)$, which factorizes into high-level mode selection based on window operation and low-level mechanical cooling: set-point control:

$$\pi_{\theta}(a_t^{\text{win}}, u_t \mid s_t) = \pi_{\theta}^{\text{hi}}\left(a_t^{\text{win}} \mid s_t\right) \pi_{\theta}^{\text{lo}}\left(u_t \mid s_t, a_t^{\text{win}}\right), \tag{2}$$

where $u_t = [T_t^{\text{fcu-sa}}, T_t^{\text{doas-sa}}]$ represents the actions of changing mechanical cooling setpoints. At each AIRL iteration, we collect trajectories with the current $\pi_{\theta}$, compute the rewards learned at the current iteration from Section 3.3 , and update the parameter set $\theta$ via PPO. The PPO algorithm optimizes the policy network parameters $\theta$ by maximizing the clipped surrogate objective:

$$L(\theta) = \mathbb{E}_t \left[ \min \left( \rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \tag{3}$$

where:

$$\rho_t(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\text{prev}}(a_t | s_t)}, \tag{4}$$

$\epsilon$ is the clipping parameter preventing drastically large policy updates, $\pi_{\text{prev}}$ is the policy from the previous PPO update, and $\hat{A}_t$ is the advantage estimate computed using Generalized Advantage Estimation (GAE). Table 2 presents the parameter sets and ranges utilized for grid search of the PPO training.

| Hyperparameter | Range |
|---|---|
| Learning Rate | {1e-2, **1e-3**, 1e-4} |
| Batch Size | {5, 10, **20**} |
| GRU Hidden Units | {32, **64**, 128} |
| PPO Epochs | {**6** 8 10} |
| Clipping Parameter ($\epsilon$) | {0.1, **0.2**, 0.3 } |
| Discount Factor ($\gamma$) | {0.95, **0.99**} |
| GAE Factor ($\lambda$) | {**0.95**, 0.99} |

*Table 2.* Hyperparameter set in PPO with final value in bold.

## 4. Results

All the experiments were performed on a 2019 MacBook Pro with an Intel Core i9 CPU (2.3 GHz, 8-core) and 16 GB DDR4 RAM. All experiments used a random seed set to 42 across all libraries. The AIRL loop used five outer iterations; in each, the reward network received ten gradient updates at a learning rate of 0.0001.

### 4.1. Data-driven Modeling Result

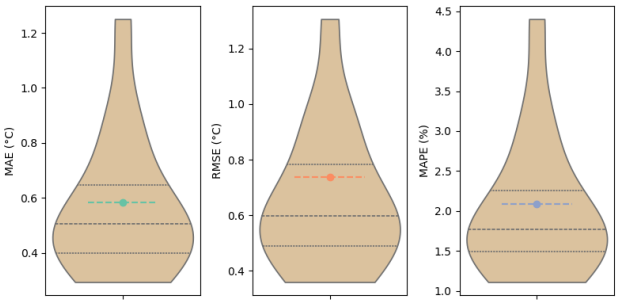


*Figure 2.* Validation results of PCNN for MMV environment.

Figure 2 shows the validation results of the proposed physics-constrained CNN-LSTM model with three metrics, Mean Absolute Error (MAE), Root Mean Square Error

(RMSE), and Mean Absolute Percentage Error (MAPE). The results show that the validation MAE is less than 0.6 ∘ C, RMSE is less than 0.8 ∘ C, and MAPE is slightly above 2%. The distributions of the three metrics also indicate that the majority of errors, falling within the interquartile range (IQR) of these metrics, are below the mean values. The results demonstrate the reliability of the proposed data-driven model as a foundation for the subsequent AIRL framework.

### 4.2. AIRL Training Results

During the AIRL training, we monitor the evolution of the learned reward function and the policy performance to ensure stable convergence. Figure 3 shows that both the adversarial reward network in the outer loop and the updates of the PPO policy for each reward function converge rapidly.

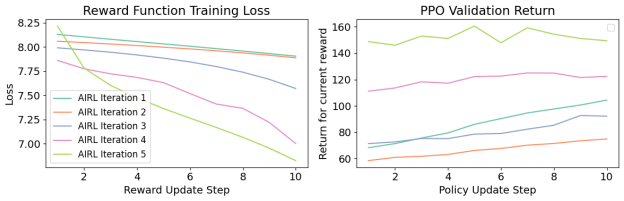


*Figure 3.* Left: Reward function update during each iteration of the AIRL training loop. Right: The reward return of the PPO agent during each iteration after each reward-net update.

Figure 4 further illustrates how AIRL policy gradually emulates the expert's control decisions, regarding window operation timing. Throughout the AIRL iteration, the agent transitions from rapid oscillations initially to accommodate the improved reward function that learns from the window usage patterns in the rule-based control. In other words, the AIRL agent has internalized when it is appropriate to open or close windows versus when to keep the system in a steady state, thereby avoiding a negative impact on the productivity of occupants.

As summarized in Table 3, the learned MMV control policy significantly improves upon its initial iteration and closely matches, and in some cases even exceeds, the performance of the rule-based control on key metrics. The initial AIRL agent, before undergoing complete adversarial training, overachieved energy savings of 52.8 kWh of AC energy consumption, but at the cost of 4.9% of temperature range violation time and excessive window operations. After the AIRL iterations, the final AIRL policy's behavior aligns with the expert demonstration, with a total energy consumption of about 133.7 kWh, slightly lower than the energy usage of the rule-based control (133.9 kWh), while improving the indoor temperature out of the comfort range for only 0.4% of the time compared to 1.7% discomfort violations of the rule-based control. With the window operation behaviors aligned, the results show that the AIRL framework can successfully capture the performance and intricacies of rule-based control.

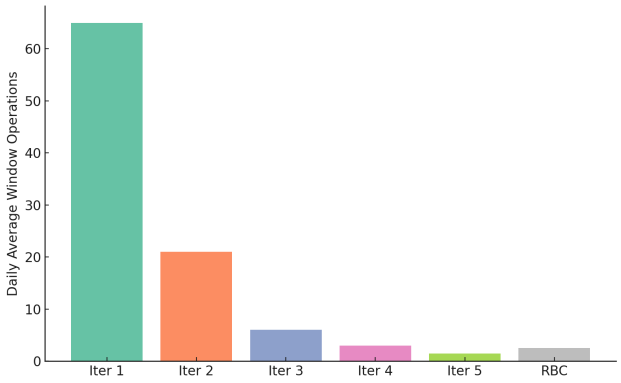


*Figure 4.* The PPO agent can learn the window operation patterns of the rule-based controller after a few iterations of the reward function.

*Table 3.* MMV Metric Comparison. **E**: Energy; **C**: Percentage of indoor temperature out of comfort range ($\geq$ 30 °C); **W**: Average times of daily window operations

| Agent | E (kWh) | C | W |
|---|---|---|---|
| Initial AIRL | 52.8 | 4.9% | 59 |
| Final AIRL | 133.7 | 0.4% | 1 |
| Expert | 133.9 | 1.7% | 2 |

## 5. Conclusion

In this work, we presented an adversarial inverse reinforcement learning framework for adaptive mixed-mode ventilation control, which learns effective HVAC and window operation policies directly from rule-based control demonstrations. We used a physics-informed surrogate model, designed a hierarchical action space, and unified reward function inference with proximal policy optimization through the adversarial inverse reinforcement learning framework. Our approach addresses the complexity of MMV systems without requiring hand-crafted reward functions. The results demonstrate that the learned policy closely replicates expert control strategies and even improves upon them: it maintains thermal comfort more consistently while using comparable energy and drastically reducing unnecessary window flips. These findings highlight that an AI agent can autonomously discover a balanced control strategy for hybrid building systems, such as mixed-mode ventilation systems. The framework minimizes mode switching and maximizes comfort, effectively handling the challenge of when to rely on natural ventilation versus air conditioning.

For future work, we need to further test the robustness of this framework on different mixed-mode ventilated environments. Also, we need to compare the effectiveness of the AIRL framework with direct RL with fine-tuned rewards, as well as other IRL techniques, such as T-REX (Brown et al., 2019). Lastly, future work involves deploying the proposed framework in a long-term real-world testbed, which would allow us to assess its performance under evolving conditions. This would further validate the robustness and versatility of our approach in practical and dynamic environments.

## Acknowledgements

This research project is supported by a joint grant from the Japan Science and Technology Agency (JST) and the Agency for Science, Technology and Research of Singapore (A*STAR) on Artificial Intelligence (AI). The views, opinions, findings, conclusions, or recommendations expressed in this material are solely those of the author(s) and do not reflect the views of these funding agencies.

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning applications in the built environment. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

## References

An, Y. and Chen, C. Energy-efficient control of indoor PM2.5 and thermal comfort in a real room using deep reinforcement learning. *Energy Build.*, 295(113340):113340, 15 July 2023. doi: 10.1016/j.enbuild.2023.113340.

Brown, D. S., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. *arXiv [cs.LG]*, 12 April 2019.

Chen, Y., Norford, L. K., Samuelson, H. W., and Malkawi, A. Optimal control of HVAC and window systems for natural ventilation through reinforcement learning. *Energy Build.*, 169:195–205, 15 June 2018. doi: 10.1016/j.enbuild.2018.03.051.

Dai, X., Cheng, S., and Chong, A. Deciphering optimal mixed-mode ventilation in the tropics using reinforcement learning with explainable artificial intelligence. *Energy Build.*, 278(112629):112629, 1 January 2023. doi: 10.1016/j.enbuild.2022.112629.

Dey, S., Marzullo, T., and Henze, G. Inverse reinforcement learning control for building energy management. *Energy Build.*, 286(112941):112941, 1 May 2023. doi: 10.1016/j.enbuild.2023.112941.

Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. *arXiv [cs.LG]*, 1 March 2016.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv [cs.LG]*, 30 October 2017.

Giraldo-Pérez, J. P., Bruse, J. L., Odriozola, J., and Mejía-Gutiérrez, R. Balancing indoor air quality and ventilation efforts via deep reinforcement learning: An agent-based approach applied to an office building. *Energy Build.*, 335(115501):115501, May 2025. doi: 10.1016/j.enbuild.2025.115501.

Hu, Z., Gao, Y., Sun, L., Mae, M., and Imaizumi, T. A novel reinforcement learning method based on generative adversarial network for air conditioning and energy system control in residential buildings. *Energy Build.*, 336 (115564):115564, 1 June 2025. doi: 10.1016/j.enbuild.2025.115564.

Liu, M., Guo, M., Fu, Y., O'Neill, Z., and Gao, Y. Expert-guided imitation learning for energy management: Evaluating GAIL's performance in building control applications. *Appl. Energy*, 372(123753):123753, 15 October 2024. doi: 10.1016/j.apenergy.2024.123753.

Pateria, S., Subagdja, B., Tan, A.-H., and Quek, C. Hierarchical reinforcement learning: A comprehensive survey. *ACM Comput. Surv.*, 54(5):1–35, 30 June 2022. doi: 10.1145/3453160.

Peng, Y., Antanuri, N., Lau, S.-K., Jebelli, B., Jusuf, S. K., Miller, C., Teo, Y. T., Chua, Y. X., and Chong, A. Experimental assessment of thermal and acoustics interactions on occupant comfort in mixed-mode buildings. *Build. Environ.*, 238(110342):110342, 15 June 2023. doi: 10.1016/j.buildenv.2023.110342.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv [cs.LG]*, 19 July 2017.

Silvestri, A., Coraci, D., Brandi, S., Capozzoli, A., and Schlueter, A. Practical deployment of reinforcement learning for building controls using an imitation learning approach. *Energy Build.*, 335(115511):115511, 15 May 2025. doi: 10.1016/j.enbuild.2025.115511.

Ziebart, B. D., Maas, A. L., Bagnell, J., and Dey, A. Maximum entropy inverse reinforcement learning. *National Conference on Artificial Intelligence*, pp. 1433–1438, 13 July 2008.

## A. Pseudo-code of the AIRL framework for MMV

**Algorithm 1** AIRL for MMV

**Input:** Rule-based MMV control demonstrations $\mathcal{D}_E$, PCNN-MMV surrogate environment $\mathcal{E}$, policy $\pi_\theta$, value net $V_{\theta_V}$, reward net $f_\phi$
**for** AIRL iteration $k = 1, 2, \ldots, K$ **do**
  Roll out $N$ days with $\pi_\theta \rightarrow$ buffer $\mathcal{D}_\pi^{(k)}$ (store states, actions, features $\varphi(s, a)$, log-probs, values)
  // ===== Reward-net update =====
  **Roll-out**: collect $N$ surrogate days with $\pi_\theta$ in $\mathcal{E}$, store trajectories $\mathcal{D}^{(k)}(\pi)$
  **Reward update** (*MaxEnt IRL*):
  **for** Each reward update **do**
    $\mathcal{L}(\psi) = \frac{1}{N_E} \sum_{\tau_i \in \mathcal{D}_\mathrm{E}} r_\psi(\tau_i) + \log \frac{1}{N_A} \sum_{\tau_j \in \mathcal{D}_\pi} \exp(-r_\psi(\tau_j))$
      $\psi; \leftarrow; \psi; -; \alpha\psi, \nabla_\psi L_\psi$
  **end for**
  Assign per-step rewards $r_t \leftarrow r_\psi(s_t, a_t)$ for every $(s_t, a_t) \in \mathcal{D}^{(k)}(\pi)$
  // ===== PPO =====
  Compute advantages $\hat{A}t$ and returns $R_t$ via GAE
  **for** each PPO epoch **do**
    Update actor parameters $\theta$ with clipped surrogate loss
    Update critic parameters $\phi$ with value loss
  **end for**
  Evaluate $\pi\theta$ on validation days; save $\theta$ if performance improves
**end for**

## B. Examples of the comparison between the physics-constrained CNN-LSTM model and the historical data

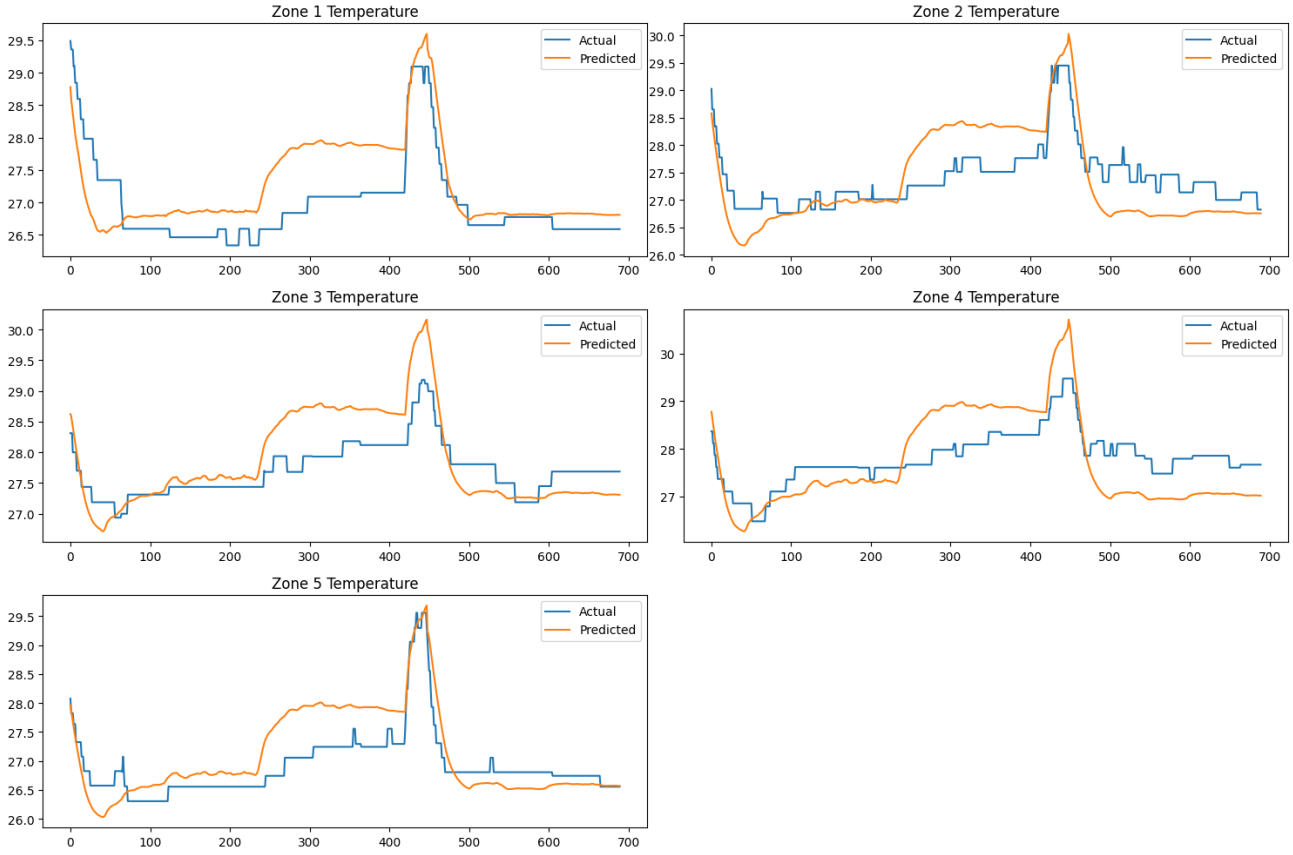


*Figure 5.* A one-day comparison between the physics-constrained CNN-LSTM model and the validation data.

## C. Examples of the comparison between the final policy and expert trajectories

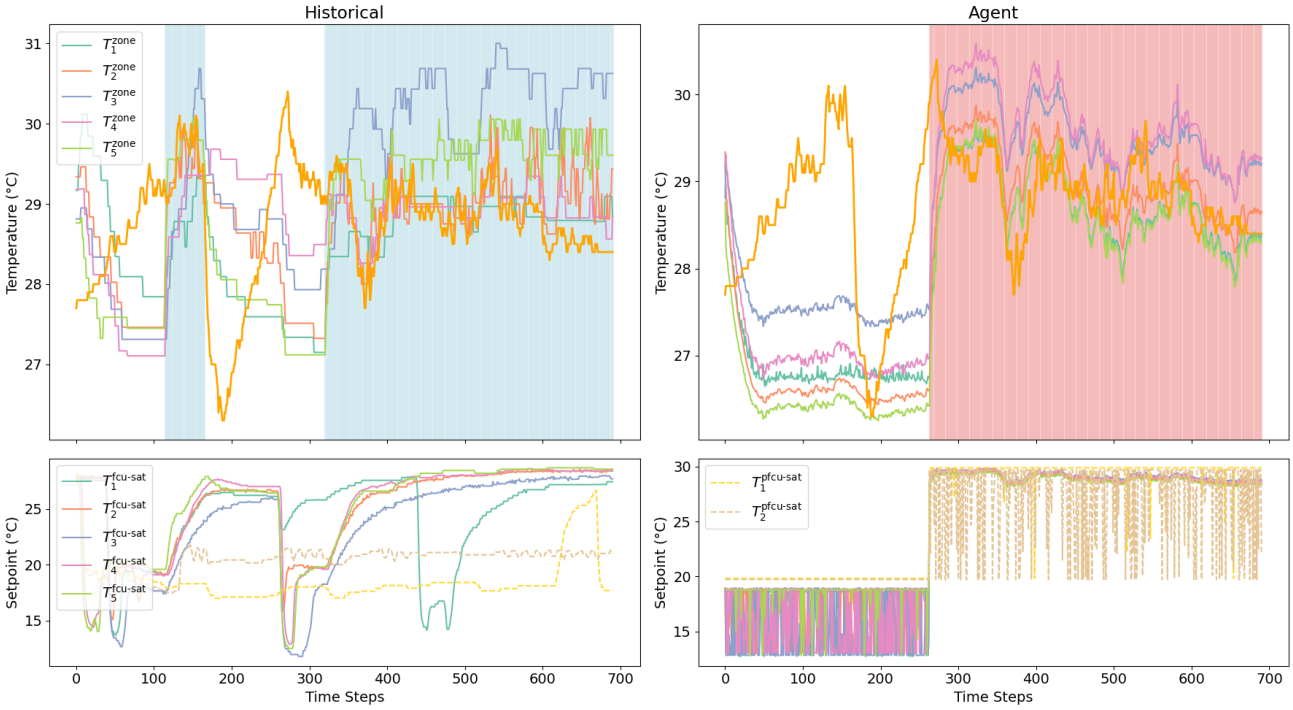


*Figure 6.* A one-day comparison between the expert trajectory and the final policy. The shaded area indicates the window open status.