GON: GENERALIZABLE OUTCOME NETWORKS FOR LEARNING ACTION EFFECTS IN PHYSICAL SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained control models such as Decision Transformer, Gato, RT-1/RT-2, and OpenVLA learn policies from large collections of observed trajectories to generalize across tasks, environments, and operating conditions. However, when control actions take time to affect a physical system, external conditions can change the system during the same period. We study this problem in smart buildings, where indoor temperature is affected by both heating, ventilation, and air conditioning (HVAC) control and external conditions such as weather, sunlight, and occupancy. A recorded temperature change alone does not show how much was caused by HVAC control. As a result, a controller trained on recorded trajectories may not reliably learn what its actions do, leading to higher energy use or reduced comfort when conditions change.

To address this problem, we design GON (Generalizable Outcome Networks), which learns how sequences of control actions affect a physical system over time. During training, GON simulates different HVAC action sequences from the same building state, keeping external conditions the same across the simulations. Comparing the resulting temperature trajectories shows how the building's response changes when the action sequence changes. An outcome model uses recent building history and a proposed HVAC action sequence to predict temperature and energy use. Given a target temperature trajectory, GON uses conditional flow matching to generate several candidate HVAC action sequences. At deployment, GON scores each sequence based on predicted discomfort and energy use, applies its first action, and repeats the process after observing the building. We evaluate GON on 100 unseen buildings using a single frozen controller without target-building adaptation. Compared with Decision Pretrained Transformer, GON reduces energy use by 34% and comfort penalty by 43% on the comfort-focused task. On the energy-focused task, it reduces energy use by 18% and comfort penalty by 43%. GON performs comparably to model predictive control (MPC) configured separately for each building.

1 INTRODUCTION

Reinforcement learning (RL) has been widely used to control cyber-physical systems, including smart buildings. Through interaction with an environment, an RL controller learns which actions improve its objective. However, physical systems differ in how quickly they respond to control. In many robotics and autonomous vehicle tasks, an action produces a change that can be observed quickly. In buildings, energy systems, and industrial processes, the effects of an action can take much longer to appear, and external conditions continue to change the system during that time. Training a controller across many buildings and operating conditions can require extensive interaction, and a policy trained for one setting may need retraining or adaptation in another.

Pretrained control models offer a way to reuse experience collected across tasks and environments. Decision Transformer models sequences of states, actions, and desired returns (Chen et al., 2021), and Trajectory Transformer models longer sequences for prediction and control (Janner et al., 2021). Gato learns across different tasks (Reed et al., 2022). RT-1, RT-2, Open X-Embodiment, and OpenVLA learn from data collected across robots and platforms (Brohan et al., 2022; Zitkovich et al., 2023; Open X-Embodiment Collaboration (incl. Guist et al., 2024; Kim et al., 2024). Generative

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.

methods also produce action sequences: Diffusion Policy uses diffusion (Chi et al., 2025), and π_0 uses flow matching (Black et al., 2024). Much of this recent work has focused on robotic control, where action effects are often observed quickly.

For slower physical systems, learning from recorded trajectories presents an additional challenge. In a smart building, indoor temperature is affected by heating, ventilation, and air conditioning actions as well as outdoor temperature, sunlight, occupancy, and stored heat in the building. If indoor temperature changes after an HVAC action, the recorded trajectory does not show how much of that change was caused by the action. A model can therefore predict recorded temperatures accurately without learning how the building would respond if the HVAC action changed. Each trajectory shows the outcome of the action that was taken; it does not show the outcome of another action from the same building state under the same external conditions.

We study this problem using the EnergyPlus-based Building2Building benchmark. Its buildings differ in thermal dynamics and operating conditions, allowing us to test a controller on buildings excluded from training. Building control has already begun to use pretraining across systems. HVAC-DPT learns from interaction histories collected across controllers and transfers to unseen buildings (Berkes, 2024), and HOT provides trajectories across many buildings and climates (Berkes et al., 2025). These data provide experience from different settings. Our focus is on providing training examples that directly show how the outcome changes when the control action changes.

To address this problem, we introduce **GON (Generalizable Outcome Networks)**. At selected points during training, we simulate several HVAC action sequences from the same building state under the same future weather and external conditions. One simulation keeps HVAC off, and the others use different action sequences. Let S^0 denote the resulting temperature trajectory with HVAC off and S^A the trajectory under action sequence A. Their difference, $C(A) = S^A - S^0$, shows how A changes the building's response relative to HVAC off under those conditions. We call these simulations matched counterfactual branches. Repeating them across different buildings, states, and conditions provides direct training examples of HVAC action effects.

GON uses recent building measurements and actions to represent the building's current behavior. Its outcome model takes this history and a proposed HVAC action sequence and predicts the resulting temperature trajectory and energy use. During training, the model learns to predict each outcome and the difference between outcomes from matched branches. The matched comparison directly tests whether the model predicts how temperature changes when the action sequence changes.

Given a target temperature trajectory, GON needs to find HVAC action sequences that can produce the required change. Several sequences can produce similar temperature changes because airflow and supply temperature can be combined in different ways. GON therefore uses conditional flow matching to generate multiple candidate action sequences. At deployment, GON uses recent building history to estimate the change needed from HVAC. Its outcome model predicts the temperature and energy use of each candidate, and GON selects a sequence based on the task's comfort and energy objective. It applies the first action, observes the building again, and repeats the process. The learned models remain fixed, and no matched simulations are needed on the test building.

We train GON on 800 buildings and evaluate a single frozen controller on 100 unseen buildings without target-building adaptation. Compared with Decision Pretrained Transformer, GON reduces energy use by 34% and comfort penalty by 43% on the comfort-focused task. On the energy-focused task, it reduces energy use by 18% and comfort penalty by 43%. GON performs comparably to model predictive control configured separately for each building.

The main contributions of this work are:

- We generate matched counterfactual trajectories from the same building state under the same external conditions, providing direct training examples of how different HVAC actions change temperature.
- We train an outcome model using recent building history and a matched-branch loss so it can predict temperature and energy use and the differences between action sequences.
- We use conditional flow matching to generate action sequences for a target temperature change, then select among them using predicted comfort and energy use. We evaluate this controller and its components on unseen buildings.

2 Related Work

Previous work on learning-based control has studied how to use historical control data to choose actions whose effects unfold over time. Researchers have developed trajectory models to learn from past interactions, predictive models to evaluate candidate actions, and generative policies to produce action sequences for a target response. We discuss how these methods use data and predictions to guide control decisions.

2.1 Learning Control from Trajectories

Decision Transformer uses past states and actions together with a target return to predict the next action (Chen et al., 2021). Trajectory Transformer models sequences of states, actions, and rewards and uses predicted trajectories for planning (Janner et al., 2021). Both methods learn from actions taken in their training trajectories. For example, stronger cooling may be applied on hotter days, so the HVAC action and outdoor temperature are associated in the data. A model may learn when cooling was used without learning how much the cooling changed indoor temperature. Under different weather or building conditions, it may then choose actions that use too much energy or fail to maintain comfort.

In building control, HVAC-DPT learns from histories collected across different controllers and transfers to unseen buildings without updating its model parameters (Berkes, 2024). Its recent history helps it select actions for the new building, but that history does not contain the outcome of an action that was not taken. HOT provides trajectories across 150,000 buildings and different climates (Berkes et al., 2025), and Building2Building provides an EnergyPlus-based benchmark for evaluating transfer across building dynamics and control objectives (Taboga et al., 2026; Crawley et al., 2001). These resources increase the variety of buildings and conditions available for training and evaluation. GON uses the simulator to add matched outcomes for different HVAC action sequences from the same state, then trains its outcome model on the differences between them.

2.2 Predicting Action Effects

Model-based methods help a controller choose an action by predicting the result of each candidate before applying it. PETS learns a probabilistic dynamics model and tests sampled action sequences in that model (Chua et al., 2018). Dreamer learns a latent world model and improves its policy using imagined trajectories (Hafner et al., 2019). In robot manipulation, Ctrl-World generates action-conditioned rollouts that can be used to evaluate policies without executing every rollout on a physical robot (Guo et al., 2026).

These methods use predicted futures to guide control, but choosing between actions requires predicting how the outcome changes when the action changes. In buildings, an HVAC action may change indoor temperature over several hours. Outdoor conditions affect temperature during that period. Weak and strong cooling may produce similar temperatures after a few minutes, but different temperatures later in the afternoon. Suppose only strong cooling keeps the room comfortable. If a model predicts both action sequences will maintain comfort, the controller may choose weak cooling to save energy, and the room becomes too warm. The model has missed the delayed temperature difference caused by changing the HVAC action. This concern is related to causal confusion in sequential decision making (De Haan et al., 2019).

Counterfactual Quotient Models use synchronized rollouts to learn differences between the outcomes of alternative actions and use those differences for action comparison (Chen et al., 2026). ThermoForce separates passive thermal behavior from the HVAC response and uses the resulting model in model predictive control (Wang, 2026). Neither method learns a generator that produces multiple HVAC action sequences from a target temperature response.

2.3 Generating Actions for Target Outcomes

A target temperature response can be produced by different HVAC action sequences. For example, higher airflow with warmer supply air and lower airflow with cooler supply air can produce similar cooling. A generative model can represent these possible sequences and provide several candidates for the controller to consider.

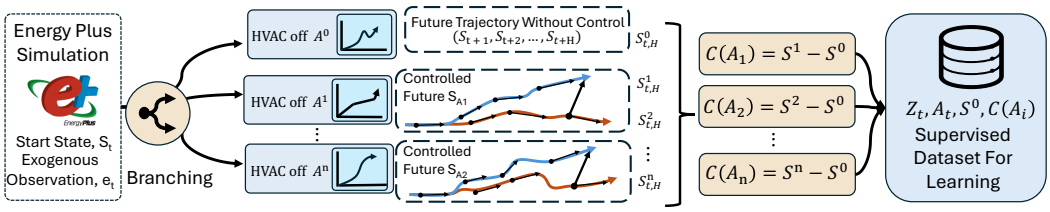


Figure 1: **Matched counterfactual branching.** Different HVAC action sequences are simulated from the same building state under the same external conditions. Comparing each result with the HVAC-off trajectory gives $C(A) = S^A - S^0$.

Diffusion Policy generates robot action sequences using diffusion (Chi et al., 2023; 2025). Nguyen et al. use diffusion over pixel motion to connect language instructions with robot actions (Nguyen et al., 2026). Flow matching was introduced to learn continuous distributions (Lipman et al.) and was later used in π_0 to generate robot actions (Black et al., 2024). We build on this idea to model HVAC action sequences conditioned on the temperature change they produce, using matched simulations to provide the training examples.

3 Methodology

GON learns how changing a control action sequence changes a physical system's future response. We develop and evaluate GON through building HVAC control, where external conditions affect indoor temperature. The method combines matched counterfactual simulations, an outcome model, and a conditional flow generator, as illustrated in Figure 2. The simulations start from the same building state under the same external conditions and show how different action sequences change temperatures. The outcome model learns to predict temperature and energy use for each sequence, including differences between matched outcomes. The generator produces candidates for a target temperature change. At deployment, GON scores the candidates using predicted comfort and energy use, applies the first action of the selected sequence, and repeats after observing the building.

3.1 Physical Control Problem and Controllable Outcomes

We consider a physical system with state s_t, control action a_t, and external conditions e_t. The system evolves according to $s_{t+1} = F(s_t, a_t, e_t)$. Because an action can affect the system over several time steps, we consider an action sequence $A = (a_t, \ldots, a_{t+K-1})$ over a horizon of K steps. In building control, s_t includes zone temperatures, a_t specifies HVAC inputs, and e_t includes outdoor temperature, solar radiation, and occupancy.

The future response under A reflects both the HVAC actions and conditions outside the controller's control. A recorded trajectory shows this combined response, but does not show how the same building would have responded to a different action sequence. We obtain that comparison by starting two simulations from the same state and giving them the same sequence of external conditions over the K steps.

Let S^A denote the future trajectory under action sequence A, and let S^0 denote the future under a reference sequence with no control input. We define the *controllable outcome* as

$$C(A) = S^A - S^0. \tag{1}$$

In our building setting, these are future zone-temperature trajectories, and S^0 is obtained with HVAC off. Temperature can still change in S^0 because of weather, occupancy, and heat already stored in the building. The difference $C(A)$ measures how applying A changes that future under the same starting state and external conditions. This effect can vary across buildings and operating conditions.

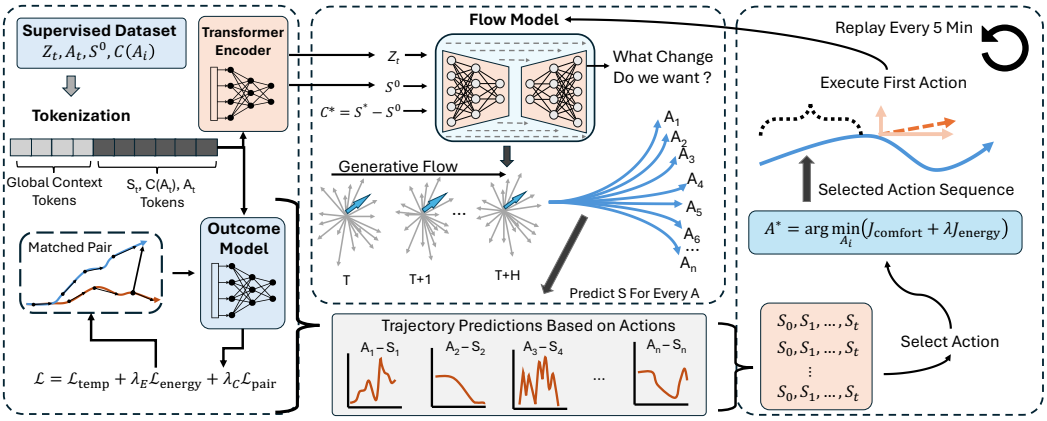


Figure 2: **Overview of GON.** The flow model generates candidate HVAC action sequences for a target temperature change. The outcome model predicts temperature and energy use for each sequence. GON selects the lowest-cost sequence, applies its first action, and repeats after observing the building.

3.2 Matched Counterfactual Branching

A recorded trajectory shows only the future produced by the control actions that were actually applied. It does not show how the same system would have evolved if a different action sequence had been applied from the same state under the same external conditions. We create these alternative futures directly in simulation.

At a selected time step, we replay the preceding episode to reach the same system state and create several future branches. As shown in Figure 1, each branch is rolled forward over the same time horizon and experiences the same sequence of external conditions, but follows a different action sequence. One branch uses a reference sequence with no control input, while the remaining branches apply alternative sequences. Because the initial state and external conditions are identical across branches, the action sequence is the only input we change. This allows us to directly measure how changing the action sequence changes the future trajectory. For a sequence A, let S^A denote its resulting future and S^0 the future under the reference branch. Their difference, $C(A) = S^A - S^0$, captures the change produced by applying A under the same conditions.

In HVAC control, we replay the building to the same thermal state and simulate four hours under the same weather, solar radiation, and occupancy. One branch keeps HVAC off, while other branches apply different combinations of airflow and supply temperature. The difference between their resulting temperature trajectories shows how the HVAC action sequence changes the building response under those conditions. We repeat this procedure across 800 training buildings. The resulting dataset contains about 190,000 four-hour branches from 21,888 branch points, covering different building states, external conditions, and levels of HVAC control. The exact branch construction and action definitions are provided in Appendix A.1.

3.3 Learning Action Outcomes

A building's response to HVAC control depends on more than its current temperature. Recent measurements and actions provide information about how quickly the building heats or cools and how its zones affect one another. GON therefore uses a short history of building states and HVAC actions rather than a single observation.

Let H_t denote this history. A transformer encoder maps H_t to a building representation z_t. We first train the encoder on mixed operating trajectories and then refine it using the matched counterfactual branches. Given z_t and a proposed HVAC action sequence A, the outcome model predicts the future

temperature trajectory $\hat{S}^A$ and HVAC energy use. Predicting each future is useful, but choosing between action sequences also requires predicting how their outcomes differ.

For two sequences A_i and A_j from the same branch point, the outcome model uses the same building representation to predict their futures. We define the matched-branch loss as

$$\mathcal{L}_{\text{pair}} = \left\| (\hat{S}^{A_i} - \hat{S}^{A_j}) - (S^{A_i} - S^{A_j}) \right\|_2^2. \tag{2}$$

This loss compares the predicted temperature difference between the sequences with the difference observed in simulation. Both branches start from the same state and experience the same external conditions, so the target difference measures how changing the HVAC action sequence changes the building response.

A standard temperature prediction loss measures each predicted future separately. It can give substantial weight to changes shared by both branches, such as temperature drift caused by weather. The matched-branch loss instead directly penalizes errors in the difference between HVAC actions. We combine the temperature prediction loss, energy prediction loss, and matched-branch loss:

$$\mathcal{L} = \mathcal{L}_{\text{temp}} + \lambda_E \mathcal{L}_{\text{energy}} + \lambda_C \mathcal{L}_{\text{pair}}, \tag{3}$$

where λ_E and λ_C weight the energy and matched-branch terms. After training, we freeze the encoder and outcome model. Architecture and training details are provided in Appendix A.1.

3.4 Outcome-Conditioned Flow Generator

The outcome model predicts the temperature and energy use that will result from a proposed HVAC action sequence. To control the building, GON must also solve the reverse problem: given a target temperature trajectory, find action sequences that can produce the required change. Let S^* be the target future temperature trajectory and S^0 the predicted trajectory with HVAC off. We define the requested controllable outcome as $C^* = S^* - S^0$. This quantity describes the temperature change that HVAC control must produce relative to the HVAC-off trajectory.

Different HVAC action sequences can produce similar temperature responses. For example, airflow and supply-air temperature can be adjusted in different combinations to provide similar cooling. We use conditional flow matching (Lipman et al.) to model these possible sequences. The generator is trained using the matched counterfactual branches described in Section 3.2. Each branch provides an action sequence A and the controllable outcome $C(A)$ it produced. The generator is also conditioned on the building representation z_t and the HVAC-off trajectory S^0, allowing it to use information about the building and its predicted response without HVAC.

Let A^0 be the HVAC-off action sequence and A an action sequence from a matched counterfactual branch. Their difference is

$$\Delta A = A - A^0. \tag{4}$$

During training, the flow model receives intermediate sequences between A^0 and A and learns to predict the change toward A. Let $\widehat{\Delta A}$ denote its predicted change. We train it using

$$\mathcal{L}_{\text{flow}} = \left\| \widehat{\Delta A} - \Delta A \right\|_2^2. \tag{5}$$

The prediction is conditioned on z_t, S^0, and $C(A)$. The matched branches therefore provide examples of which action sequences produced particular temperature changes under different building conditions. At deployment, GON replaces $C(A)$ with the requested outcome C^* and generates candidate HVAC action sequences. The frozen outcome model predicts temperature and energy use for each candidate, and GON selects the sequence with the lowest predicted control cost.

3.5 Deployment

At deployment, the encoder, outcome model, and flow generator remain fixed. GON creates no counterfactual branches on the target building and makes no model parameter updates. Every five

minutes, it encodes the recent building history and forms a target temperature trajectory S^*. Using the predicted HVAC-off trajectory $\hat{S}^0$, it computes the requested controllable outcome $C^* = S^* - \hat{S}^0$. The flow model generates eight candidate HVAC action sequences for this request.

The frozen outcome model predicts the future temperature trajectory and energy use for each candidate sequence A. GON scores each candidate using

$$J(A) = J_{\text{comfort}}(A) + \beta_E J_{\text{energy}}(A), \tag{6}$$

where both terms are computed from the predicted outcome and β_E weights energy use according to the control objective. GON selects the candidate with the lowest predicted cost and applies only its first five-minute action.

After observing the building again, GON generates and evaluates candidate sequences. The HVAC-off prediction is refreshed once every hour, while candidate generation and selection are repeated every five minutes. This allows GON to adjust subsequent actions as building conditions change.

4 Experimental Evaluation

We evaluate whether learning action effects improves control on new buildings using the Building2Building benchmark (Taboga et al., 2026). We first compare GON with other controllers on 100 buildings excluded from training. We then ablate its main components on a separate 16-building validation set. Finally, we study performance across buildings and climates and test how the number of generated action sequences affects control. We report energy use and comfort penalty under both comfort-focused and energy-focused objectives. Further experimental details are provided in Appendix A.2.

Evaluation setting. Building2Building provides buildings with different climates, thermal dynamics, numbers of thermal zones, and HVAC configurations. These differences allow us to test whether a controller trained across buildings can select effective actions in buildings it has not encountered during training. EnergyPlus simulates each building's heat transfer and HVAC response under changing outdoor conditions (Crawley et al., 2001). Every five minutes, the controller observes the building and selects HVAC inputs, such as supply-air temperature, airflow, or zone damper settings, depending on the HVAC system. We evaluate GON and the baseline controllers over the same buildings and operating conditions. For each controller, the objective is to regulate zone temperatures and limit HVAC energy use, with the relative emphasis determined by the comfort-focused or energy-focused task.

We report two metrics for each task. Energy is cumulative HVAC energy use over the evaluation period, measured in kWh. Comfort is the Building2Building temperature-deviation penalty, measured in $^\circ\text{C}^2$. Lower comfort penalties mean zone temperatures remain closer to their targets. Lower values are better for both metrics. We report energy and comfort together because reducing energy use may increase temperature deviations, and stronger temperature regulation may require more energy. Unless stated otherwise, results are reported as the mean $\pm$ standard error across buildings using five seeds.

4.1 Control on Unseen Buildings

We compare GON with five baselines on 100 Building2Building test buildings excluded from training. The rule-based controller (RBC) provides a fixed control baseline. Model predictive control (MPC) uses a model configured separately for each building and serves as a building-specific reference (Frahm et al., 2022; Liang et al., 2022). HVAC-DPT learns control from recorded trajectories (Berkes, 2024), a generalist PPO policy provides a reinforcement-learning baseline (Biswas, 2020; Li et al., 2025), and standard flow matching provides a generative-action baseline. GON, HVAC-DPT, PPO, and standard flow matching each use one pretrained model across the test buildings, with no target-building model updates. We evaluate all methods under both comfort-focused and energy-focused objectives. Baseline implementation details are provided in Appendix A.2.

Table 1 shows that GON has the lowest energy use and comfort penalty among the generalist controllers on both tasks. Compared with HVAC-DPT, GON reduces energy use by 34% and comfort penalty by 43% on the comfort-focused task. On the energy-focused task, it reduces energy use

Table 1: Energy use and comfort penalty on 100 unseen Building2Building test buildings under comfort-focused and energy-focused objectives. Values are mean ± standard error across buildings. Lower is better for both metrics.

	Comfort-focused task		Energy-focused task	
Method	Energy (kWh) ↓	Comfort ($^\circ$C^2) ↓	Energy (kWh) ↓	Comfort ($^\circ$C^2) ↓
RBC	$14\,724 \pm 793$	0.753 ± 0.061	$14\,724 \pm 793$	0.753 ± 0.061
MPC	$8\,551 \pm 375$	0.493 ± 0.041	$8\,551 \pm 375$	0.493 ± 0.041
DPT	$13\,592 \pm 653$	0.651 ± 0.068	$10\,592 \pm 512$	0.933 ± 0.098
Flow Matching	$13\,278 \pm 720$	0.612 ± 0.071	$11\,349 \pm 537$	0.913 ± 0.077
PPO Generalist	$14\,048 \pm 783$	0.989 ± 0.055	$10\,048 \pm 582$	2.034 ± 0.082
GON (Proposed)	$8\,957 \pm 406$	0.373 ± 0.052	$8\,657 \pm 402$	0.533 ± 0.061

by 18% and comfort penalty by 43%. Standard flow matching also performs worse than GON on both metrics, suggesting that generating action sequences alone does not account for GON's gains. GON's energy use is within 5% of the MPC reference on both tasks. MPC requires a separate model for each building, making it more time-consuming to deploy across many buildings. GON achieves similar energy use with a single controller that requires no target-building model updates.

4.2 Ablation

We ablate the main components of GON on a separate 16-building validation set. We remove the matched-branch loss to test whether training on differences between matched outcomes improves control. We replace the flow generator with regression to test the value of generating multiple action sequences for a target temperature change. We also compare learned candidates with randomly sampled candidates and reduce the number of flow samples from eight to one. These variants test the roles of matched-outcome training, action generation, and candidate selection.

Table 2: Ablation results on the 16-building validation set under comfort-focused and energy-focused objectives. Values are mean ± standard error across buildings.

	Comfort-focused task		Energy-focused task	
Method	Energy (kWh) ↓	Comfort ($^\circ$C^2) ↓	Energy (kWh) ↓	Comfort ($^\circ$C^2) ↓
GON (full)	$9\,635 \pm 942$	0.414 ± 0.042	$8\,731 \pm 829$	0.561 ± 0.038
w/o $\mathcal{L}_{\text{pair}}$	$12\,505 \pm 1\,505$	0.677 ± 0.083	$11\,662 \pm 1\,355$	0.703 ± 0.087
w/o flow generator	$16\,515 \pm 2\,903$	1.113 ± 0.038	$16\,515 \pm 2\,903$	1.113 ± 0.046
Random candidates	$24\,312 \pm 2\,399$	1.813 ± 0.068	$21\,690 \pm 1\,200$	4.894 ± 0.076
1 flow sample	$14\,765 \pm 1\,422$	1.190 ± 0.034	$14\,765 \pm 1\,422$	1.190 ± 0.034

Table 2 shows that removing the matched-branch loss increases energy use by 30% and comfort penalty by 64% on the comfort-focused task. This result supports training the outcome model to predict how temperature changes when the HVAC action sequence changes. Replacing the flow generator with regression leads to higher energy use and comfort penalty on both tasks. Using one flow sample performs worse than using eight, indicating that generating and evaluating several sequences improves control. Random candidates produce the highest energy use and comfort penalty among the variants, showing that candidate selection depends on the quality of generated sequences.

4.3 Performance Across Buildings and Climates

Figure 3 (top) compares GON and DPT with two building-specific controllers, MPC and DRL, across 100 unseen buildings ordered by climate zone. GON and DPT each use one pretrained controller across all buildings, whereas MPC and DRL use building-specific models or training. The curves show cumulative improvement over the RBC baseline as buildings are added in climate-zone order. This comparison shows how a single controller performs as the evaluation moves across buildings with different climate conditions.

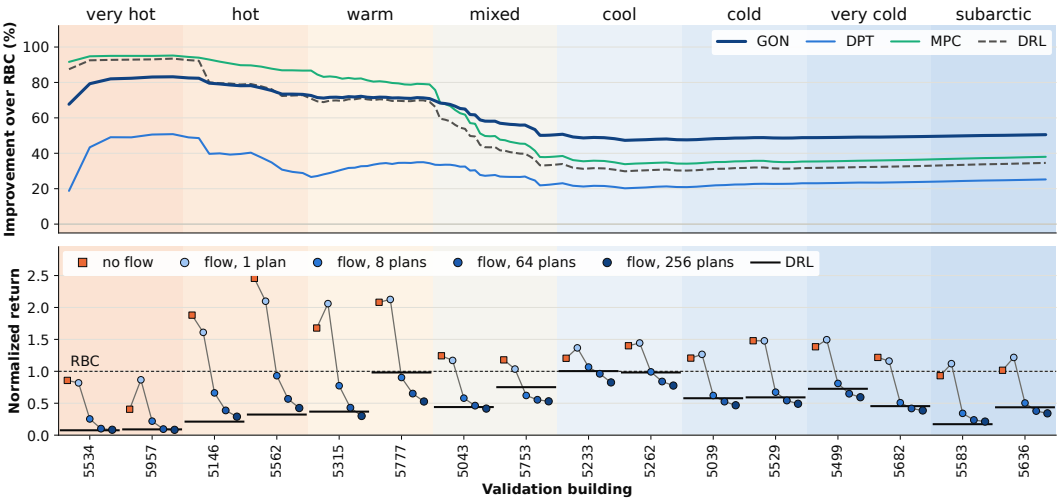


Figure 3: **Control across buildings and candidate counts.** Top: cumulative improvement over RBC across 100 unseen buildings ordered by climate zone. Bottom: normalized control return on 16 validation buildings for different numbers of flow-generated action sequences.

GON maintains a higher cumulative improvement than DPT across the climate zones. Its performance also approaches or exceeds that of the building-specific controllers in several zones, despite using one frozen model across all test buildings. These results show that GON's gains extend across buildings with different climates and thermal dynamics.

4.4 Evaluating Flow Matching Through Candidate Selection

We test whether flow matching improves control by providing multiple action sequences for GON to evaluate. We keep the trained models fixed and vary the number of sequences sampled from the flow generator. For each sample count, the outcome model predicts the temperature and energy use of the candidates, and GON selects the sequence with the lowest predicted control cost. Figure 3 (bottom) reports the results on 16 validation buildings.

Without flow generation, the mean normalized return is 1.35, and one flow sample gives 1.40. The reported value decreases to 0.65 with eight samples, 0.49 with 64, and 0.42 with 256. With 256 samples, GON outperforms the version without flow generation on all 16 validation buildings and the building-specific DRL controller on 12 of them. These results suggest that GON benefits from generating several candidate sequences for the same target temperature change and evaluating their predicted outcomes before selecting an action.

5 Conclusion

We designed GON to learn how control actions change the future response of a physical system when external conditions also affect the observed outcome. We studied this problem in building control. GON compares simulations that start from the same building state and experience the same external conditions but use different HVAC action sequences. It learns the resulting temperature differences, generates candidate sequences for a target response, and selects an action using predicted temperature and energy use. Across 100 unseen buildings, a single frozen GON controller reduces both energy use and comfort penalty compared with Decision Pretrained Transformer. Its energy use is within 5% of MPC configured separately for each building.

Future work will train GON on a larger and more varied collection of buildings and study whether fine-tuning with limited data improves control in new buildings. We will also test the approach in other physical systems as a step toward a foundation model for physical control.

AI USE STATEMENT

In this work, we used generative AI tools primarily to suggest revisions to the wording of the manuscript and improve its clarity and readability. We reviewed and edited all AI-assisted suggestions. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have released the source code and data used in this study at `https://huggingface.co/GeneralOutcomeNetworks/GON`. In this paper, Section 3 describes the GON methodology, and Section 4 presents the experimental settings, baselines, metrics, and results.

REFERENCES

Anaïs Berkes. Hvac-dpt: A decision pretrained transformer for hvac control. *arXiv preprint arXiv:2411.19746*, 2024.

Anaïs Berkes, Yoshua Bengio, David Rolnick, and Donna Vakalis. A hot dataset: 150,000 buildings for hvac operations transfer research. In *Proceedings of the 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 171–180, 2025.

Debmalya Biswas. Reinforcement learning based hvac optimization in factories. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, pp. 428–433, 2020.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Junlin Chen, Ruijie Wang, and Jianxin Li. Counterfactual quotient models: Learning what actions change, not what the world does. *arXiv preprint arXiv:2608.22092*, 2026.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

C Chi, Z Xu, S Feng, E Cousineau, Y Du, B Burchfiel, R Tedrake, and S Song. Diffusion policy: Visuomotor policy learning via action diffusion. the intl. *Journal of Robotics Research*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

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

Pim De Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *Advances in neural information processing systems*, 32, 2019.

Moritz Frahm, Philipp Zwickel, Jan Wachter, Felix Langner, Pascal Strauch, Jörg Matthes, and Veit Hagenmeyer. Occupant-oriented economic model predictive control for demand response in buildings. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, pp. 354–360, 2022.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *8th Annual Conference on Robot Learning*, 2024.

Jiatong Li, Bokai Ji, Guangxia Li, Peilin Zhao, Liu Liu, and Zhigang Ren. Thermal-delay aware deep reinforcement learning for hvac system control. In *Proceedings of the 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 277–281, 2025.

Wei Liang, Sizhe Ma, Erica Cochran, and Katherine A Flanigan. Integrated mpc-ilc control design for thermal control of a large-scale multi-zone building hvac system. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 129–138, 2022.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

E-Ro Nguyen, Yichi Zhang, Kanchana Ranasinghe, Xiang Li, and Michael S Ryoo. Pixel motion diffusion is what we need for robot control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23663–23672, 2026.

S. Open X-Embodiment Collaboration (incl. Guist, J. Schneider, B. Schölkopf, and D.) Büchler. Open x-embodiment: Robotic learning datasets and rt-x models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, May 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://ieeexplore.ieee.org/document/10611477.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Vincent Taboga, Justin Veilleux, Doseok Jang, Anushree Rankawat, and Pierre-Luc Bacon. Building2building: A large scale benchmark for generalizable real-world reinforcement learning. *arXiv preprint arXiv:2607.16534*, 2026.

Yifan Wang. Thermoforce: A physics-structured interventional world model for building hvac control. *arXiv preprint arXiv:2607.03942*, 2026.

B Zitkovich, T Yu, S Xu, P Xu, T Xiao, F Xia, J Wu, P Wohlhart, S Welker, and A Wahid. Rt-2: Vision-language-action models transfer web knowledge to robotic control, conference on robot learning, 2023.

A Appendix

A.1 Implementation Details

This appendix provides the implementation and evaluation settings used for the Building2Building experiments. Training uses only the designated training split, validation buildings are used for model selection and ablations, and the test split is reserved for the final zero-shot evaluation.

Table 3: Building2Building configuration used in the experiments.

Setting	Value
Training / validation / test buildings	800 / 100 / 100
Conditioned zones	5
ASHRAE climate zones	1–8
Episode length	90 days
Control interval	5 min
Target temperature	21°C
History length	48 steps (4 h)
Prediction horizon	48 steps (4 h)
Test-building adaptation	None
Default flow candidates	8

A.1.1 BUILDING2BUILDING SETUP

We use the `OfficeSmall` family from Building2Building (Taboga et al., 2026), simulated with EnergyPlus (Crawley et al., 2001). The training, validation, and test splits are disjoint and span all eight ASHRAE climate zones. A fixed subset of 16 validation buildings is used for the ablation and flow-scaling experiments.

GON receives the standard Building2Building observations and action interface, including zone temperatures, outdoor conditions, time information, HVAC energy use, supply-air temperature, and airflow. At test time, GON does not receive EnergyPlus parameters, building metadata, or future weather, and no model parameters are updated. We evaluate both comfort-focused and energy-focused objectives and report energy and comfort separately for each.

A.1.2 MATCHED COUNTERFACTUAL DATASET

Matched counterfactual branches are generated only on the 800 training buildings. At each branch point, the simulator is restored to the same state and several four-hour futures are generated under identical external conditions while changing only the HVAC plan. One branch uses HVAC off as the reference, while the remaining branches use different airflow and supply-air temperature plans. Across the training set, we generate approximately 190,000 four-hour branches from 21,888 branch points. Each branch is paired with the preceding 48 five-minute observations and actions, giving four hours of recent history and a four-hour prediction horizon.

A.1.3 MODEL ARCHITECTURE AND TRAINING

GON contains a history encoder, an outcome model, and a flow generator. The history encoder summarizes the previous four hours of observations and actions into the building representation z_t. It is first pretrained on ordinary operating trajectories and is then refined with the outcome model using matched counterfactual data.

The outcome model receives z_t and a proposed HVAC plan and predicts future zone temperatures and energy use. It is trained using the temperature prediction, energy prediction, and matched-branch losses described in Section 3.2. After training, the encoder and outcome model are frozen. The flow generator is conditioned on z_t, the predicted HVAC-off future S^0, and the requested controllable outcome C^*. It is trained using the conditional flow-matching objective in Section 3.4 and generates compact HVAC plans that are converted into valid supply-air temperature and airflow commands.

Table 4 gives the architecture and training settings for all three components.

A.1.4 CLOSED-LOOP EVALUATION

The same frozen GON model is used for every test building without online learning or parameter updates. Every five minutes, GON encodes the recent history, generates candidate HVAC plans, and uses the outcome model to predict the temperature and energy consequences of each candidate. It executes the first action of the lowest-cost plan and repeats the procedure after receiving the next observation. The HVAC-off prediction S^0 is refreshed once per hour. Candidate generation and

Table 4: Main implementation settings of GON.

Parameter	Encoder	Outcome model	Flow generator
Architecture			
Observation / action dimension	13 / 10	13 / 10	13 / 10
Network	Transformer	MLP	MLP
Layers	4	3	5
Hidden size	128	256	512
Attention heads	4	–	–
Context length (steps)	48	48	48
Prediction horizon (steps)	48	48	48
Output dimension	10 (action)	48×6	6
Optimization			
Optimizer	AdamW	AdamW	AdamW
Learning rate	2×10^{-4}	3×10^{-4}	3×10^{-4}
Weight decay	0.01	0.01	0.01
Warm-up steps	1000	100	200
LR schedule	cosine	cosine	cosine
Gradient clip	1.0	1.0	1.0
Loss			
λ_{BC} / λ_{state}	1.0 / 1.0	–	–
λ_E / λ_C	–	1.0 / 1.0	–
Training			
Batch size	256	128	256
Maximum epochs	12	240	400 (40 min)
Selected epoch (seed 0)	8	236	163
Observation normalization	min–max	min–max	min–max
Action normalization	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$

selection are repeated every five minutes, with eight candidates used by default. For the flow-scaling experiment in Figure 3, the trained models and scoring procedure remain fixed. Only the number of generated candidates is changed. The 8-, 64-, and 256-candidate variants therefore measure the benefit of giving the same frozen controller more candidate plans to evaluate.

A.2 Evaluation Setup

This section gives additional details for the control evaluation in the main paper. All controllers use the same Building2Building test setting described in Appendix A.1.

The main comparison separates *generalist* and *specialist* controllers. Generalist controllers use one model across all test buildings. They are trained only on the training buildings, use the validation split for model selection, and are not updated on a test building. Specialist controllers are configured or trained separately for each building and are included as stronger building-specific references.

A.2.1 Baselines

Rule-based controller (RBC). RBC is the tuned rule-based controller provided by Building2Building. It uses fixed HVAC control rules based on the current building observations and does not learn from the test building. We use RBC as the non-learning reference and as the reference for the normalized-return and improvement measures in Figure 3.

Model predictive control (MPC). MPC is a building-specific model-based specialist. A predictive thermal model is configured separately for each building and is used to evaluate candidate control sequences over a finite horizon. MPC therefore uses building-specific information that the generalist controllers do not receive. We include it as a specialist reference to compare one frozen GON controller, used across all test buildings, with a controller configured separately for each building.

Decision Pretrained Transformer (DPT). DPT is the trajectory-pretraining baseline. A causal transformer is trained on operating trajectories from the training buildings and predicts HVAC actions from the recent history and desired return. The same pretrained DPT model is then used on every test building without parameter updates. DPT therefore tests whether large-scale trajectory pretraining alone is enough to transfer control to unseen buildings.

Standard flow matching. This baseline tests whether generative action modeling alone explains GON's performance. It uses flow matching to generate HVAC actions, but it does not use GON's matched counterfactual action-outcome supervision or the matched-branch loss. The model is trained across the same training-building population and is deployed without parameter updates on the test buildings. The comparison therefore separates the benefit of generative action modeling from the benefit of explicitly learning how outcomes change when the HVAC plan changes.

PPO generalist. The PPO baseline tests whether one model-free reinforcement-learning policy can transfer across buildings. A single PPO policy is trained across multiple training buildings using the same Building2Building observation and action interface.

The resulting policy is used on all test buildings without test-building training or parameter updates.

Building-specific DRL specialist. For the building-wise comparison in Figure 3, we also include a deep reinforcement-learning controller trained separately for each building.

This controller is not a zero-shot generalist. It is included as a specialist reference because each building receives its own trained controller. This allows us to compare one frozen GON controller against a reinforcement-learning controller specialized to the individual building.

GON. GON uses the same pretrained history encoder, outcome model, and conditional flow generator on every test building. All learned components remain frozen during the test episode. At deployment, GON receives only the recent building observations and the standard Building2Building action interface. It does not receive the EnergyPlus model, does not create matched counterfactual branches, and does not update its parameters on the target building.

A.2.2 Evaluation Metrics

Building2Building reports energy use and comfort penalty for each building, where lower values are better. Energy is the total HVAC electricity and gas consumption over the 90-day episode in kWh. Comfort measures the mean squared deviation of zone temperature from the 21°C target,

$$\text{Comfort} = \frac{1}{T|\mathcal{Z}|} \sum_{t=1}^{T} \sum_{z \in \mathcal{Z}} (T_{z,t} - 21)^2, \tag{7}$$

where $\mathcal{Z}$ is the set of conditioned zones. Comfort is reported in $^{\circ}\text{C}^2$. We report energy and comfort separately for the comfort-focused and energy-focused objectives.

For the flow-scaling experiment in Figure 3, controller cost is normalized by the RBC cost on the same building,

$$\text{Normalized Cost} = \frac{J}{J^{\text{RBC}}}, \tag{8}$$

where 1 corresponds to RBC performance and lower values are better.

The top panel of Figure 3 reports cumulative improvement over RBC as buildings are ordered from the hottest to the coldest climate zones,

$$\text{Improvement}(n) = 100 \left(1 - \frac{\sum_{b=1}^{n} J_b}{\sum_{b=1}^{n} J_b^{\text{RBC}}} \right). \tag{9}$$

The final point includes all 100 test buildings. It gives improvements of 50.5% for GON, 38.0% for MPC, and 34.6% for the building-specific DRL specialist.

Percentage reductions in energy or comfort are calculated relative to the comparison method,

$$\text{Reduction} = 100\frac{\text{Baseline} - \text{Method}}{\text{Baseline}}. \tag{10}$$

These reductions are computed using the mean values across the 100 test buildings.

A.2.3 Seeds and Error Bars

Unless stated otherwise, learned controllers are evaluated using five seeds. Results are first averaged across seeds for each building and then averaged across buildings. Tables report mean $\pm$ standard error across buildings, with $N = 100$ for the test set and $N = 16$ for validation experiments.