Contrastive Preference Learning from User Overrides for Personalized Home Energy Management

Xiao Du, Fengji Luo, *Senior Member, IEEE*, Jianheng Lan, Wei Zhou, *Member, IEEE,* and Junhao Wen, *Member, IEEE*

***Abstract*—Personalizing home energy management systems (HEMSs) to align with dynamic user preferences remains challenging for intelligent energy control. Existing deep reinforcement learning (DRL) approaches typically rely on static reward functions and fixed constraints, limiting their ability to adapt to diverse user behaviors. We propose a framework that leverages user override feedback—instances where users manually correct system decisions—as implicit preference signals. Our method combines contrastive preference learning from pairwise comparisons, context-aware preference embeddings from interaction history, and preference-conditioned DRL policies. This enables the controller to continuously align with personalized preferences without explicit reward engineering. Validation on real-world residential energy data shows that the framework achieves an average override rate of 0.033 across four diverse preference profiles—a 37% reduction over the strongest baseline—while maintaining the lowest user interaction cost (0.394) among all feedback-driven methods. The framework also demonstrates efficient transfer across prototype preferences and generalization to novel behavioral rules.**

***Index Terms*—User preference learning, home energy management, contrastive learning, reinforcement learning, implicit feedback**

NOMENCLATURE

AC	Air conditioner
AMI	Advanced metering infrastructure
BESS	Battery energy storage system
BT	Bradley–Terry (model)
COP	Coefficient of performance
CPL	Contrastive preference learning
DRL	Deep reinforcement learning
GEB	Grid-interactive efficient building
HEMS	Home energy management system
HVAC	Heating, ventilation, and air conditioning
LSTM	Long short-term memory
MDP	Markov decision process
MILP	Mixed-integer linear program
MLP	Multilayer perceptron
MPC	Model predictive control
ODI	Override dependence index
OR	Override rate
PV	Photovoltaic
RBC	Rule-based control
RLHF	Reinforcement learning from human feedback
SAC	Soft actor–critic
SHGC	Solar heat gain coefficient
SOC	State of charge
TOU	Time-of-use (pricing)
UIC	User interaction cost
WM	Washing machine

This work was supported by the Coal-Major Project (Grant No. 2025ZD1700800), the Natural Science Foundation of Chongqing Municipality (Grant Nos. CSTB2025YITP-QCRCX0071, 2407013565272987, CSTB2024NSCQ-MSX0701, and CSTB2024TIAD-STX0023), the Australian Research Council (Grant No. DP220103881), and the China Scholarship Council (Grant No. 202406050163).

X. Du, W. Zhou, and J. Wen are with the School of Big Data and Software Engineering, Chongqing University, Chongqing,China. E-mail: duxiao, zhouwei, jhwen@cqu.edu.cn

F. Luo, J. Lan are with the School of Civil Engineering, the University of Sydney, Australia. Email: fengji.luo, jianheng.lan@sydney.edu.au

I. INTRODUCTION

BUILDINGS account for approximately one-third of global final energy use and energy-related CO_2 emissions [1], making residential efficiency critical to carbon neutrality goals. National initiatives such as the U.S. Department of Energy's grid-interactive efficient buildings (GEBs) roadmap—which aims to triple building-sector efficiency and flexibility by 2030 [2]—further underscore this urgency. In this context, home energy management systems (HEMSs) have emerged as key technologies that coordinate appliances and distributed energy resources to reduce costs, enhance comfort, and improve grid stability [3].

However, the effectiveness of HEMSs critically depends on how well these systems capture and respond to user preferences [4]. Unlike centralized systems with predetermined objectives, households are inherently diverse: different residents prioritize different trade-offs—some favor comfort while others prioritize cost savings—and even individual preferences can shift gradually over time as lifestyles and seasons change [5]. This diversity and long-term dynamics demand preference-aware control.

Yet most existing paradigms fall short. Rule-based controllers (RBCs) rely on static heuristics, while model predictive control depends on pre-specified objectives and constraints [4], [6]. Deep reinforcement learning (DRL) offers flexibility by learning nonlinear policies directly from data, but DRL-based HEMSs still rely on static reward functions that fail to reflect individualized trade-offs [7]. When learned trade-offs diverge from actual preferences, users often override system decisions, undermining satisfaction and long-term adoption.

These overrides, however, are more than mere symptoms of misalignment—they carry rich implicit preference information. As Fig. 1 illustrates, when users override system-recommended actions, each override reveals a pairwise preference: the user's chosen action is favored over the system's suggestion under the current context [4]. Critically, although

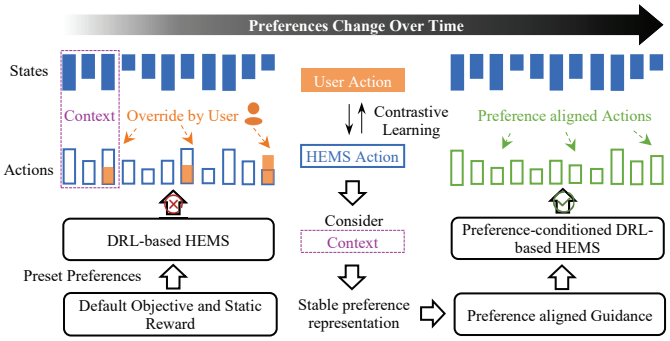

Fig. 1. Motivation: user overrides on HEMS actions carry implicit, context-dependent preference signals that static-reward approaches discard; our framework learns a latent preference representation from these overrides for adaptive control.

contextual factors such as time-of-use prices, weather, and occupancy are observable, preferences are not—users cannot articulate precise trade-off weights, and no contextual signal reveals them. A comfort-oriented and a cost-conscious user facing the same heatwave will override the same recommendation in opposite directions. Yet preferences are relatively stable in the short term: a comfort-oriented user remains comfort-oriented across different hours and weather conditions. What changes across contexts is not the preference itself, but how it manifests: the same "comfort-first" preference leads to different optimal actions under a heatwave versus a mild evening.

Extracting these stable preferences from overrides is nonetheless challenging. Overrides are sparse—users intervene only when system actions noticeably deviate from their intent—and noisy, as users may respond inconsistently across occasions. Moreover, because the optimal action varies with time-of-use prices, weather, and occupancy, the preference signal in each override is entangled with contextual factors, making it difficult to disentangle what reflects the user's preference from what reflects the context.

Recent advances in reinforcement learning from human feedback (RLHF) offer a principled way to address this challenge. By learning from human corrections and preference comparisons, RLHF aligns agents with user intent [8]. Overrides are particularly well-suited: they are effortless, frequent, and directly reflect users' priorities under real operating conditions.

Building on this insight, we propose a contrastive preference learning (CPL) framework that extracts latent preferences from user overrides and encodes them into a context-aware embedding for adaptive, personalized control (Fig. 1); to our knowledge, it is the first such framework for HEMS, integrating and adapting established preference-learning and DRL techniques. Concretely, the framework comprises:

1) A utility function trained via contrastive learning on override pairs, capturing pairwise preferences with richer supervision than scalar rewards.
2) A neural encoder that distills recent interactions into a latent preference vector, aligned with the utility function to form a compact representation of user intent that naturally generalizes across preference targets.

TABLE I
COMPARISON OF PREFERENCE/FEEDBACK LEARNING APPROACHES.

Reference	Domain	Feedback	Preference Repr.	Ctx. aware
[9]	Smart home	Interview	Semantic rules	✗
[7]	Smart home	Semantic	Ontology constraints	✗
[10]	HEMS	Satisfaction rating	Scalar weight	✗
[11]	HVAC	Implicit override	Bayesian posterior	✗
[12]	Lighting	Implicit behavioral	Learned setpoint	✗
[13]	HVAC	Implicit override	Learned setpoint	✗
[4]	HEMS	Implicit override	Negative reward	✗
[14]	HVAC	Active pairwise	Learned reward	✗
[15]	General RL	Active pairwise	Learned reward	✗
[16]	General RL	Offline pairwise	Contrastive policy	✗
[8]	General RL	Offline pairwise	Contrastive embed.	✓
Ours	HEMS	Implicit override	Contrastive utility + latent embedding	✓

Active/Offline pairwise: human actively queried vs. learned from a static dataset.
Ctx.-aware: representation disentangles stable intent from context-dependent expression.

3) A DRL agent conditioned on both states and preference embeddings, enabling adaptive control that evolves with changing preferences and transfers efficiently to novel preference targets with minimal additional interaction.

The remainder of this paper is organized as follows. Section II reviews related work. Section III formulates the problem and preference modeling. Section IV presents the methodology. Section V provides experimental results and analysis. Section VI concludes the paper.

II. RELATED WORK

DRL has emerged as a powerful tool for multi-objective HEMS [17], extending beyond electricity cost to incorporate comfort [18], carbon emissions [19], and equipment longevity [20]. However, most DRL-based approaches balance these objectives with static penalty weights [21], relying on pre-defined objective structures rather than learning from user preferences. Personalization has been explored in broader energy settings, including collaborative filtering for appliance usage recommendations [22], RL-based personalized pricing and preference-aware charging recommendation for electric vehicles [23], and Bayesian neural networks for uncertainty-aware preference modeling in HEMS [24]. Despite these advances, all of them rely on pre-specified preference representations or require explicit user input, leaving the challenge of learning latent, dynamic preferences from implicit signals largely unaddressed.

A growing body of work incorporates user feedback as supervisory signals to infer preferences, though most still depend on explicit forms of engagement. These approaches differ in feedback modality. At the most structured end, (1) elicitation-based methods collect preferences via questionnaires or interviews and translate them into semantic rules [9], while (2) semantic feedback maps natural language descriptions to formal constraints that reshape reward structures [7]. Less demanding but still explicit, (3) satisfaction ratings quantify preferences as scalar scores incorporated into reward functions [10]. At the other end of the spectrum, (4) implicit behavioral signals infer

preferences from users' natural interactions; for example, [12] learns personalized lighting setpoints from occupant switch actions via RL, [11] uses Bayesian inference to learn thermal comfort–cost trade-offs from thermostat overrides, [13] applies multi-armed bandits to learn preferred setpoints from unsolicited override actions, and [4] encodes manual overrides as negative reward signals in a DRL-based HEMS.

RLHF offers a complementary paradigm for preference alignment. Following [15], RLHF learns reward models from pairwise preference judgments: human evaluators compare trajectory segments, providing preference signals that guide policy optimization. Subsequent work improves feedback efficiency through off-policy relabeling and unsupervised pre-training [25], and has achieved success in aligning large language models [26] and robotic systems [27]. More recently, contrastive objectives have emerged as an alternative that bypasses explicit reward modeling: Direct Preference Optimization [28] reparameterizes the RLHF objective into a simple classification loss, and [16] extends this idea to control by deriving policies directly from preferences via a contrastive loss, while [8] separates distinct behaviors in the embedding space to identify informative comparisons with minimal human input. In the energy domain, [14] applies RLHF to HVAC recommendation, but still relies on active user queries rather than passive overrides.

Despite this progress, critical gaps remain. First, methods (1)–(3) require active user engagement—questionnaires, verbal feedback, or daily ratings—that is costly and difficult to sustain [6], [10]. Second, most approaches [4], [9], [10] map feedback to fixed reward signals, ignoring the context dependence of preference expression: the same underlying preference produces different optimal actions under different conditions, a structure that static mappings cannot capture. Third, even implicit override-based methods [4], [11], [13] treat corrections as scalar penalties or learn single-dimensional setpoints rather than extracting structured pairwise preferences, discarding the comparative information each override carries. Finally, RLHF and preference-based RL [15], [25] provide principled frameworks for learning from comparisons, yet they rely on active queries in which a human deliberately selects among presented alternatives. These methods have not been adapted to the passive, sparse, and context-entangled override signals that characterize HEMS interactions. Table I summarizes representative approaches, highlighting these distinctions.

III. Problem Formulation

A. System Modeling

1) Overview: The household is connected to both the utility grid and a rooftop photovoltaic (PV) system, as shown in Fig. 2. The HEMS manages three representative controllable energy devices (CEDs): a battery energy storage system (BESS), an air conditioner (AC), and a washing machine (WM). This selection captures the main categories of residential loads—priority, deferrable, and flexible—commonly analyzed in demand-side management [4]. Other appliances, such as TVs or microwaves, are treated as uncontrollable base load P_t^{BL} (in kW).

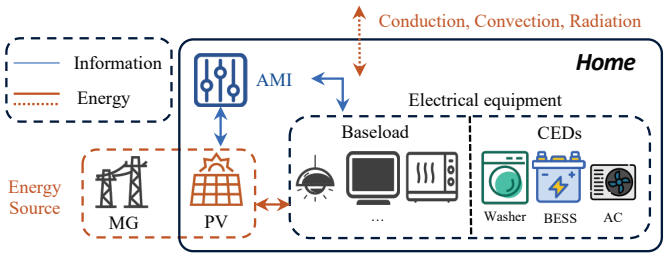

Fig. 2. Architecture of the HEMS with rooftop PV, grid connection, and controllable devices.

The HEMS operates in discrete time steps $t \in \{1, 2, \ldots, \tau\}$ with interval Δt (in hours). At each step, the controller observes the system state through advanced metering infrastructure (AMI) and determines CED operations. The objective is to jointly minimize four performance aspects: electricity cost $\mathcal{O}_{\mathrm{c}}$, thermal comfort violation $\mathcal{O}_{\mathrm{t}}$, carbon emissions $\mathcal{O}_{\mathrm{e}}$, and laundry waiting time $\mathcal{O}_{\mathrm{l}}$, whose relative priority is governed by user preferences. The multi-objective optimization is formulated as:

$$\min_{\pi(\cdot|\mathbf{z})} \ \big(\mathcal{O}_{\mathrm{c}}, \mathcal{O}_{\mathrm{t}}, \mathcal{O}_{\mathrm{e}}, \mathcal{O}_{\mathrm{l}}\big), \quad \text{s.t. alignment with } \mathbf{z}, \tag{1a}$$

$$\text{where} \quad \mathcal{O}_{\mathrm{c}} = \frac{1}{\tau}\sum_{t=1}^{\tau} \kappa_t P_t^{\mathrm{G}} \Delta t, \tag{1b}$$

$$\mathcal{O}_{\mathrm{t}} = \frac{1}{\tau}\sum_{t=1}^{\tau} \mathbb{1}\big(|T_t^{\mathrm{in}} - T_t^{\mathrm{set}}| > T^{\mathrm{band}}\big), \tag{1c}$$

$$\mathcal{O}_{\mathrm{e}} = \frac{1}{\tau}\sum_{t=1}^{\tau} \mu_t P_t^{\mathrm{G}} \Delta t, \tag{1d}$$

$$\mathcal{O}_{\mathrm{l}} = \frac{1}{n^{\mathrm{lau}}}\sum_{i=1}^{n^{\mathrm{lau}}} \frac{\min(t_i^{\mathrm{start}} - t_i^{\mathrm{lau}}, z_i^{\mathrm{lau}})}{z_i^{\mathrm{lau}}}. \tag{1e}$$

Here, κ_t (\$/kWh), μ_t (kg CO_2/kWh), and P_t^{G} (kW) denote electricity price, grid carbon intensity, and grid power exchange, respectively; T_t^{in} (°C), T_t^{set} (°C), and T^{band} (°C) are the indoor temperature, setpoint, and comfort tolerance band. Each of the n^{lau} laundry requests is defined by request time t^{lau} (hour of day), maximum tolerable delay z^{lau} (hours), and actual start time t^{start} (hour of day), where the start time depends on the WM operating mode $M_t^{\mathrm{w}} \in \{0, 1\}$ (1: running, 0: idle). The control policy $\pi(\cdot|\mathbf{z})$ is conditioned on a latent preference vector $\mathbf{z} \in \mathcal{Z}$ encoding dynamic user preferences learned from override feedback. The optimization seeks a policy that jointly minimizes the objectives in (1a), with the implicit trade-offs among them governed by the learned preference vector $\mathbf{z}$.

2) Energy Device Models: The energy devices are modeled using standard physics-based equations following [29], which capture operational constraints and energy conversion dynamics.

a) BESS: The BESS follows standard charge/discharge dynamics with stored energy E_t (kWh), rated capacity E_{rat}

(kWh), and state of charge (SOC) $\text{soc}_t = E_t / E_{\text{rat}}$:

$$E_{t+1} = \overline{E}_{t+1}(1 - \text{SD} \cdot \Delta t), \tag{2a}$$

$$\overline{E}_{t+1} = \begin{cases} E_t + \eta^{\text{b}} P_t^{\text{b}} \Delta t, & M_t^{\text{b}} = 1, \\ E_t - \frac{P_t^{\text{b}} \Delta t}{\eta^{\text{b}}}, & M_t^{\text{b}} = -1, \\ E_t, & M_t^{\text{b}} = 0, \end{cases} \tag{2b}$$

$$P_t^{\text{b}} \in [0, P_{\text{sup}}^{\text{b}}], \tag{2c}$$

$$P_{\text{sup}}^{\text{b}} = \begin{cases} \min\{P_{\text{rat}}^{\text{b}}, \frac{E_{\text{rat}} - E_t}{\eta^{\text{b}} \Delta t}\}, & M_t^{\text{b}} = 1, \\ \min\{P_{\text{rat}}^{\text{b}}, \frac{\eta^{\text{b}} E_t}{\Delta t}\}, & M_t^{\text{b}} = -1, \\ 0, & M_t^{\text{b}} = 0, \end{cases} \tag{2d}$$

where $M_t^{\text{b}} \in \{-1, 0, 1\}$ denotes the operating mode (1: charging, -1: discharging, 0: standby), P_t^{b} (kW) and $P_{\text{rat}}^{\text{b}}$ (kW) are the actual and rated power, SD (h^{-1}) is the self-discharge rate, and η^{b} is the charge/discharge efficiency.

b) AC: The AC is modeled thermodynamically with heating/cooling power P_t^{heat} (kW) and coefficient of performance (COP) cop_t dependent on indoor temperature T_t^{in} (°C) and outdoor temperature T_t^{out} (°C):

$$P_t^{\text{heat}} = \text{cop}_t P_t^{\text{a}} M_t^{\text{a}}, \quad P_t^{\text{a}} \in [P_{\text{sta}}^{\text{a}}, P_{\text{rat}}^{\text{a}}], \tag{3a}$$

$$\text{cop}_t = \begin{cases} \overline{\text{cop}}_t, & 0 < \overline{\text{cop}}_t < \text{cop}_{\text{rat}}, \\ \text{cop}_{\text{rat}}, & \text{otherwise}, \end{cases} \tag{3b}$$

$$\overline{\text{cop}}_t = \begin{cases} \frac{\eta^{\text{a}}(T_t^{\text{in}})_{\text{K}}}{T_t^{\text{in}} - T_t^{\text{out}}}, & M_t^{\text{a}} = 1, \\ \frac{\eta^{\text{a}}(T_t^{\text{in}})_{\text{K}}}{T_t^{\text{out}} - T_t^{\text{in}}}, & M_t^{\text{a}} = -1, \\ 0, & M_t^{\text{a}} = 0, \end{cases} \tag{3c}$$

where $M_t^{\text{a}} \in \{-1, 0, 1\}$ is the operating mode (1: heating, -1: cooling, 0: standby), P_t^{a} (kW) is the electrical power consumption bounded by minimum $P_{\text{sta}}^{\text{a}}$ (kW) and rated $P_{\text{rat}}^{\text{a}}$ (kW) values, $(\cdot)_{\text{K}}$ denotes temperature conversion to Kelvin, cop_{rat} is the upper COP limit, and η^{a} is the thermodynamic efficiency factor.

c) WM: The WM operates as a non-interruptible task with power consumption P_t^{w} (kW) and accumulated run time z_t^{w} (hours): once started, it runs continuously until task duration $z_{\text{task}}^{\text{w}}$ (hours) is completed.

$$P_t^{\text{w}} = M_t^{\text{w}} P_{\text{rat}}^{\text{w}}, \tag{4a}$$

$$z_t^{\text{w}} = \begin{cases} \min\{z_{t-1}^{\text{w}} + \Delta t,\ z_{\text{task}}^{\text{w}}\}, & M_t^{\text{w}} = 1, \\ 0, & M_t^{\text{w}} = 0, \end{cases} \tag{4b}$$

$$M_t^{\text{w}} = \begin{cases} 1, & (0 < z_{t-1}^{\text{w}} < z_{\text{task}}^{\text{w}}) \ \vee \ A_t^{\text{w}} = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{4c}$$

where $P_{\text{rat}}^{\text{w}}$ (kW) is the rated power and $A_t^{\text{w}} \in \{0, 1\}$ is the start signal (allowed only when idle).

d) PV: Energy generation P_t^{PV} (kW) depends on solar irradiance I_t (kW/m^2), panel area A^{pan} (m^2), conversion efficiency η^{PV}, and rated output per unit area $P_{\text{rat}}^{\text{PV}}$ (kW/m^2):

$$P_t^{\text{PV}} = \begin{cases} 0, & I_t < I_{\min}, \\ \min(\eta^{\text{PV}} I_t, P_{\text{rat}}^{\text{PV}}) A^{\text{pan}}, & \text{otherwise}, \end{cases} \tag{5}$$

where $I_{\min}$ (kW/m^2) is the minimum irradiance threshold for generation.

3) Building Thermal Dynamics: Indoor temperature T_t^{in} (°C) evolves according to a single-zone heat balance model among conduction Q_t^{con} (kWh), ventilation Q_t^{ven} (kWh), solar gains Q_t^{solar} (kWh), and HVAC heating/cooling P_t^{heat} (kW):

$$T_{t+1}^{\text{in}} = T_t^{\text{in}} + \frac{Q_t^{\text{con}} + Q_t^{\text{ven}} + Q_t^{\text{solar}} + P_t^{\text{heat}} \Delta t}{c^{\text{air}} \rho^{\text{air}} V^{\text{hou}}}, \tag{6a}$$

$$Q_t^{\text{con}} = \lambda A^{\text{suf}} (T_t^{\text{out}} - T_t^{\text{in}}) \Delta t, \tag{6b}$$

$$Q_t^{\text{ven}} = \text{ACH}\, c^{\text{air}} \rho^{\text{air}} V^{\text{hou}} (T_t^{\text{out}} - T_t^{\text{in}}) \Delta t, \tag{6c}$$

$$Q_t^{\text{solar}} = \text{SHGC}\, I_t A^{\text{fen}} \Delta t, \tag{6d}$$

where λ (kW/(m^2°C)) is thermal conductivity, A^{suf} (m^2) is the building envelope surface area, A^{fen} (m^2) is the window area, ACH (h^{-1}) is the air change rate, SHGC is the solar heat gain coefficient, c^{air} (kWh/(kg°C)) is the specific heat capacity of air, ρ^{air} (kg/m^3) is air density, and V^{hou} (m^3) is the indoor volume.

4) Power Balance Constraint: The household's total net demand P_t^{D} (kW) must equal supply from local PV, BESS, and grid exchange P_t^{G} (kW):

$$P_t^{\text{G}} = \max(P_t^{\text{D}}, 0), \tag{7a}$$

$$P_t^{\text{D}} = P_t^{\text{a}} + M_t^{\text{b}} P_t^{\text{b}} + P_t^{\text{w}} + P_t^{\text{BL}} - P_t^{\text{PV}}, \tag{7b}$$

where excess generation is curtailed (no feed-in to grid).

5) Stochastic Occupancy and Demand:

a) Occupancy Modeling: Daily occupancy status $\text{occ}_{d,t} \in \{0, 1\}$ (1: at home, 0: away) is determined by departure time t_d^{dep} and arrival time t_d^{arr} (hours of day), sampled based on a daily work schedule indicator $w_d \in \{0, 1\}$:

$$\text{occ}_{d,t} = 1 - \mathbb{1}(t_d^{\text{dep}} \le t < t_d^{\text{arr}}), \tag{8a}$$

$$t_d^{\{\text{dep,arr}\}} \sim \begin{cases} \mathcal{N}(\mu_{\text{work}}^{\{\text{dep,arr}\}}, (\sigma_{\text{work}}^{\{\text{dep,arr}\}})^2), & w_d = 1, \\ \mathcal{N}(\mu_{\text{rest}}^{\{\text{dep,arr}\}}, (\sigma_{\text{rest}}^{\{\text{dep,arr}\}})^2), & w_d = 0, \end{cases} \tag{8b}$$

$$w_d \sim \begin{cases} \text{Bernoulli}(p^{\text{work}}), & \text{dow}_d \in \mathcal{W}, \\ 0, & \text{otherwise}, \end{cases} \tag{8c}$$

where $\text{dow}_d \in \{1, \ldots, 7\}$ is the day-of-week (Monday=1, Sunday=7), $\mathcal{W}$ is the set of workdays, and p^{work} is the probability of following a work schedule on a workday.

b) Laundry Demand: For each day d, laundry requests are generated probabilistically. If a request occurs ($\text{req}_d = 1$), the request time t_d^{lau} is sampled from valid occupied time slots, and the request remains active for up to z_d^{lau} time steps:

$$\text{lau}_{d,t} = \text{req}_d \cdot \mathbb{1}(t_d^{\text{lau}} \le t \le \min(t_d^{\text{lau}} + z_d^{\text{lau}}, t_d^{\text{end}})), \tag{9a}$$

$$t_d^{\text{lau}} \sim \text{Uniform}\{t \in \mathcal{T}_d \,|\, \text{occ}_{d,t} = 1\}, \quad \text{if } \text{req}_d = 1, \tag{9b}$$

$$\text{req}_d \sim \text{Bernoulli}(p_{\text{dow}_d}^{\text{lau}}), \tag{9c}$$

$$z_d^{\text{lau}} \sim \max(0, \lfloor \mathcal{N}(\mu^{\text{wait}}, (\sigma^{\text{wait}})^2) + 0.5 \rfloor), \tag{9d}$$

where $p_{\text{dow}_d}^{\text{lau}}$ is the day-of-week-dependent request probability, $\mathcal{T}_d = \{t \,|\, t^{\min} \le \text{hour}(t) \le t^{\max}\}$ is the valid request time window (hours of day), t_d^{end} is the last time step of day d, and μ^{wait}, σ^{wait} parameterize the maximum tolerable delay distribution.

6) Exogenous Variables from Data: Exogenous inputs are derived from real-world datasets [30], [31]: I_t and T_t^{out} capture weather, while P_t^{BL} represents uncontrollable appliance consumption. Electricity price κ_t (\$/kWh) follows time-of-use tariffs from local utility records, and grid carbon intensity μ_t (kg CO_2/kWh) is computed from regional generation mix statistics. These time-varying parameters capture the stochastic and seasonal characteristics of real operating conditions.

B. Preference-Augmented MDP Formulation

We formulate preference-aware HEMS control as a preference-augmented Markov Decision Process (MDP):

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{Z}, \gamma \rangle, \tag{10}$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition kernel, $\mathcal{R}$ is the reward function, $\mathcal{Z}$ is the latent preference space, and $\gamma \in [0, 1)$ is the discount factor. The inclusion of $\mathcal{Z}$ enables the policy to condition on latent preference embeddings $\mathbf{z}_t \in \mathcal{Z}$.

1) State Space: The state $s_t \in \mathcal{S}$ aggregates observable system conditions and context: indoor/outdoor temperatures (T_t^{in}, T_t^{out}, T_t^{set}), device operating modes (M_t^{b}, M_t^{a}, M_t^{w}) and power consumptions (P_t^{b}, P_t^{a}, P_t^{w}), BESS SOC (soc_t), PV generation (P_t^{PV}), baseline load (P_t^{BL}), occupancy and laundry indicators (occ_t, lau_t), electricity price (κ_t), grid carbon intensity (μ_t), and temporal features (e.g., dow_d).

2) Action Space: The action $a_t \in \mathcal{A}$ specifies control commands for the three CEDs:

$$a_t = \left[P_t^{\text{b}},\, M_t^{\text{b}},\, P_t^{\text{a}},\, M_t^{\text{a}},\, A_t^{\text{w}}\right], \tag{11}$$

subject to the operational constraints defined in Section III-A. We denote by a_t^{rec} the HEMS recommendation and by a_t^{ovr} the executed action after a possible user override; $a_t^{\text{ovr}} = a_t^{\text{rec}}$ when no override occurs. Unqualified a_t in transition dynamics and reward definitions refers to the executed action throughout.

3) Transition Dynamics: The transition probability $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ is determined by the physics-based device models (Eqs. (2)–(5)), building thermal dynamics (Eq. (6)), power balance (Eq. (7)), stochastic processes for occupancy and laundry demand (Eqs. (8)–(9)), and exogenous inputs (Section III-A6).

4) Reward Function: The immediate reward $r_t = \mathcal{R}(s_t, a_t)$ transforms system-level objectives (Eq. (1)) into bounded component rewards suitable for RL training:

$$\mathcal{R}_{\text{c}}(s_t, a_t) = \exp\big(-|P_t^D|\,\kappa_t\big), \tag{12a}$$

$$\mathcal{R}_{\text{t}}(s_t, a_t) = \text{occ}_t \cdot \exp\big(\min\{0,\; T^{\text{band}} - |T_t^{\text{in}} - T_t^{\text{set}}|\}\big), \tag{12b}$$

$$\mathcal{R}_{\text{e}}(s_t, a_t) = \exp\big(-|P_t^D|\,\mu_t\big), \tag{12c}$$

$$\mathcal{R}_{\text{l}}(s_t, a_t) = \begin{cases} \exp(-|t - t^{\text{lau}}|), & \text{lau}_t = 1 \text{ and } \sum_{t^{\text{lau}}}^{t} M_t^{\text{w}} = 0, \\ 1, & \text{otherwise.} \end{cases} \tag{12d}$$

$\mathcal{R}_{\text{c}}$ rewards low electricity cost (price $\times$ power), $\mathcal{R}_{\text{t}}$ rewards keeping the indoor temperature within the comfort band while occupied, $\mathcal{R}_{\text{e}}$ rewards low carbon emissions (carbon intensity $\times$ power), and $\mathcal{R}_{\text{l}}$ rewards starting laundry promptly after its request. These exponential mappings ensure each component is non-negative, bounded in $(0, 1]$, and monotone in its corresponding objective, facilitating stable gradient-based optimization.

Personalization is achieved by augmenting system rewards with a learned preference utility U_ϕ (detailed in Section III-C):

$$r_t = r_t^{\text{sys}} + \alpha^u\, U_\phi(s_t, a_t^{\text{ovr}}), \quad r_t^{\text{sys}} = \sum_{i \in \{\text{c,t,e,l}\}} \beta_i\, \mathcal{R}_i(s_t, a_t^{\text{ovr}}), \tag{13}$$

where $\beta_i \geq 0$ weight the system objectives and $\alpha^u \geq 0$ sets the preference alignment strength. The four component rewards are standard HEMS objectives; the contribution of this part lies in the additive integration of the learned preference utility, $\alpha^u\, U_\phi(s_t, a_t^{\text{ovr}})$, which injects a context-dependent preference signal directly into the reward and thereby turns the fixed-weight objective into a preference-adaptive one. Unlike the multiplicative gating of [14], this additive integration preserves the original objective scaling and interpretable trade-offs while improving optimization stability.

5) Preference-Conditioned Policy: The policy is conditioned on both current state and latent preference embedding:

$$\pi_\theta(a_t \mid s_t, \mathbf{z}_t) : \mathcal{S} \times \mathcal{Z} \to \Delta(\mathcal{A}), \tag{14}$$

where θ denotes the policy parameters, $\Delta(\mathcal{A})$ is the probability simplex over actions, and $\mathbf{z}_t$ encodes dynamic user preferences inferred from interaction history. The optimal policy maximizes expected discounted return:

$$\pi_\theta^* = \arg\max_{\pi_\theta} \mathbb{E}_{\pi_\theta}\Big[\sum_{t=0}^{\infty} \gamma^t r_t\Big]. \tag{15}$$

C. Preference Modeling

Based on observations from practical HEMS deployments, we model preferences under the following assumptions:

- **(A1) Latent and revealed preferences.** User preferences are latent—they cannot be directly observed—but are revealed through user-system interactions. When a user overrides the recommended action a_t^{rec} with a_t^{ovr}, this reveals a pairwise preference $a_t^{\text{ovr}} \succ a_t^{\text{rec}}$ given state s_t.
- **(A2) Short-term stability and context dependence.** Preferences represent relatively stable orientations that persist across contexts, yet their behavioral expression varies with contextual factors. Over longer horizons, preferences may drift gradually.

1) Utility Function: To leverage revealed preferences from overrides (A1), we define a utility function $U_\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that assigns a desirability score to each state–action pair:

$$u_t = U_\phi(s_t, a_t), \tag{16}$$

where higher values indicate stronger alignment with user preferences. This function provides an observable proxy for latent preferences via pairwise comparisons. Specifically, each override enforces the constraint:

$$U_\phi(s_t, a_t^{\text{ovr}}) > U_\phi(s_t, a_t^{\text{rec}}), \tag{17}$$

capturing instantaneous preference signals at individual timesteps.

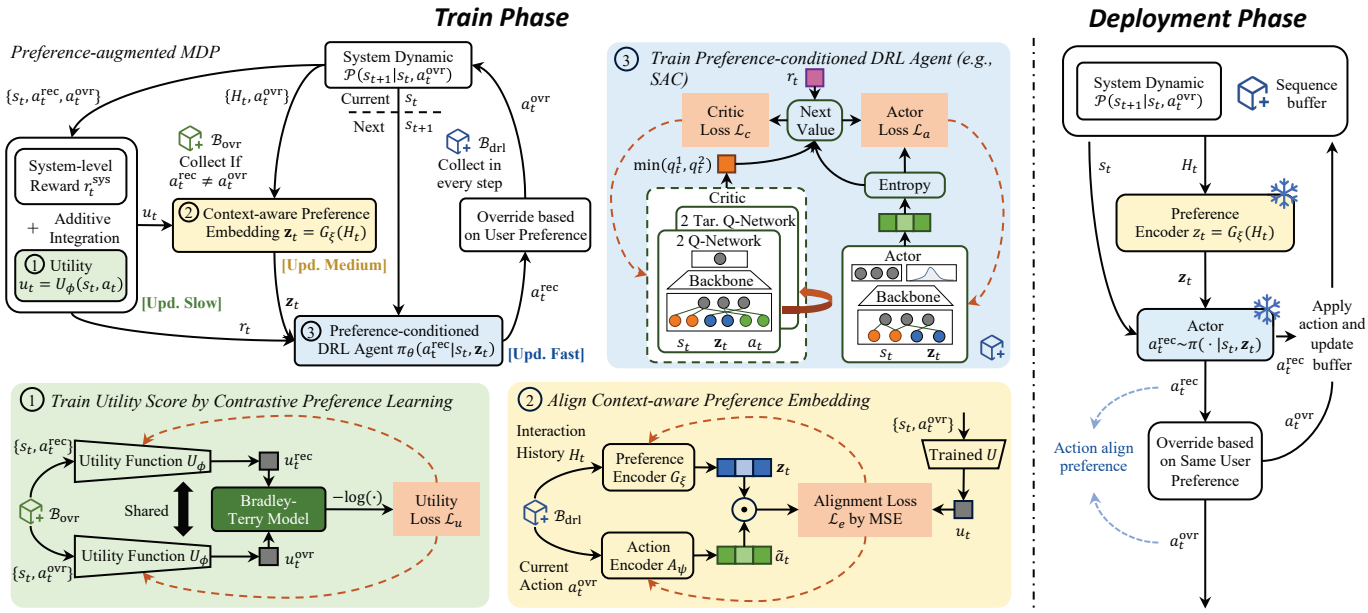

Fig. 3. Overview of the proposed framework. Left: training phase with three modules forming a closed feedback loop. Right: deployment phase using only G_ξ and π_θ.

2) Latent Preference Embedding: To capture the stable yet context-dependent nature of preferences (A2), we learn a compact latent embedding $\mathbf{z}_t \in \mathcal{Z}$ that encodes recent interaction history (Eq. (22)). Unlike U_ϕ, which quantifies instantaneous preference signals from individual overrides, $\mathbf{z}_t$ aggregates temporal patterns to extract the underlying long-term orientations—the relatively stable priorities that guide user behavior across contexts. The embedding is trained to be consistent with U_ϕ (Section IV-B), and the policy $\pi_\theta(a_t \mid s_t, \mathbf{z}_t)$ (Eq. (14)) conditions on $\mathbf{z}_t$ to adapt to the user's evolving yet stable preference state.

IV. Methodology

As illustrated in Fig. 3, our approach integrates user preferences into HEMS control through three interacting modules forming a closed feedback loop during training (left), which yields a deployable system that adapts to user intent without further gradient updates (right). During training, overrides guide utility learning (U_ϕ), utilities supervise preference embeddings ($\mathbf{z}_t$), and embeddings condition the policy (π_θ). The following subsections detail each module; the training procedure and deployment protocol are described in Section IV-D.

A. Contrastive Preference Learning

1) Synthetic Override Generation: Real-time preference collection is expensive and human-in-the-loop studies are impractical for large-scale experiments. We generate synthetic overrides using teacher policies $\{\pi^{(k)}\}_{k=1}^{K}$, each trained with distinct system objective weights $\beta^{(k)}$ in Eq. (13). Each training run uses a single teacher $\pi^{(k)}$, fixed throughout all episodes, to represent one resident's preference profile. This ensures that all override signals in $\mathcal{B}_{\text{ovr}}$ are consistent with the same underlying preference, providing clean supervision for the utility function U_ϕ. The teacher produces preferred action $a_t^{\text{pref}} = \pi^{(k)}(s_t)$.

Let $a^{\text{norm}}(\cdot)$ denote normalized action vectors with continuous setpoints scaled to $[0, 1]$ and one-hot encoded device modes. The discrepancy between agent recommendation and teacher preference is:

$$\delta_t = \left\| a^{\text{norm}}(a_t^{\text{rec}}) - a^{\text{norm}}(a_t^{\text{pref}}) \right\|_2. \tag{18}$$

Overrides are generated using a piecewise rule combining deterministic corrections with stochastic, human-like reactions:

$$a_t^{\text{ovr}} = \begin{cases} a_t^{\text{pref}}, & \delta_t \geq \tau_h, \\ a_t^{\text{pref}} \text{ w.p. } p(\delta_t), & \tau_s \leq \delta_t < \tau_h, \\ a_t^{\text{rec}} \text{ w.p. } 1 - p_{\min}, & \delta_t < \tau_s, \end{cases} \tag{19a}$$

$$p(\delta_t) = p_{\min} + \frac{(p_{\max} - p_{\min})(\delta_t - \tau_s)}{\tau_h - \tau_s}, \tag{19b}$$

where $0 < p_{\min} \ll p_{\max} < 1$ are minimum and maximum override probabilities, and $\tau_s < \tau_h$ are discrepancy thresholds. Large discrepancies trigger deterministic corrections, moderate ones cause stochastic overrides with linearly increasing probability, while small deviations still permit occasional overrides to emulate realistic user variability.

In our simulation, override decisions are evaluated at each control timestep Δt, synchronized with the HEMS decision cycle. In practice, user overrides are event-driven and asynchronous—a resident may adjust the thermostat or reschedule the washing machine at any time. To bridge this gap, the framework discretizes any override arriving within the interval $(t, t+\Delta t]$ as an override event at step $t+1$, while intervals without user action default to acceptance ($a_t^{\text{ovr}} = a_t^{\text{rec}}$). This event-triggered design requires no continuous monitoring by the user.

2) Bradley–Terry Model: User override behavior is inherently stochastic and inconsistent—residents may respond differently to the same recommendation depending on mood, context, or attention. To capture this uncertainty in preference strength, we adopt the Bradley–Terry (BT) model [32], which provides a principled probabilistic framework for pairwise comparisons. Unlike margin-based ranking losses that impose hard decision boundaries, the BT model naturally accommodates noisy feedback through its logistic form, aligning with modern RLHF methods [8], [15] and offering robust gradient signals for stable training.

Formally, the probability that action a_i is preferred over a_j in state s follows:

$$\begin{aligned} P_\phi(a_i \succ a_j \mid s) &= \frac{\exp(U_\phi(s,a_i))}{\exp(U_\phi(s,a_i)) + \exp(U_\phi(s,a_j))} \\ &= \sigma\big(U_\phi(s,a_i) - U_\phi(s,a_j)\big), \end{aligned} \tag{20}$$

where $\sigma(\cdot)$ is the logistic function, identifying U_ϕ up to an additive constant.

Given override buffer $\mathcal{B}_{\text{ovr}} = \{(s_t, a_t^{\text{ovr}}, a_t^{\text{rec}})\}$, we minimize the negative log-likelihood:

$$\begin{aligned} \mathcal{L}_u(\phi) &= -\sum_{(s_t,a_t^{\text{ovr}},a_t^{\text{rec}})\in\mathcal{B}_{\text{ovr}}} \log P_\phi(a_t^{\text{ovr}} \succ a_t^{\text{rec}} \mid s_t) \\ &= \sum_{(s_t,a_t^{\text{ovr}},a_t^{\text{rec}})\in\mathcal{B}_{\text{ovr}}} \log\big(1+\exp(-\Delta U_t)\big), \end{aligned} \tag{21}$$

where $\Delta U_t = U_\phi(s_t, a_t^{\text{ovr}}) - U_\phi(s_t, a_t^{\text{rec}})$, with ℓ_2 regularization on ϕ. We adopt Bradley–Terry over margin-based ranking losses $\mathcal{L}_u = \sum_t \max(0,\, m - \Delta U_t)$ for superior probabilistic calibration and noise robustness (Section V-D).

B. Context-aware Preference Embedding

While $U_\phi(s,a)$ provides pairwise supervision, it evaluates individual state–action pairs in isolation without capturing temporal preference patterns. To enable policy generalization across contexts, we learn a latent preference embedding $\mathbf{z}_t \in \mathbb{R}^d$ from recent interaction history:

$$H_t = \{(s_j, a_j^{\text{rec}}, a_j^{\text{ovr}})\}_{j=t-\tau_w}^{t-1} \cup \{s_t\}, \tag{22}$$

where τ_w is the look-back window. The current state s_t is included to provide situational context that helps the encoder infer which aspect of the user's preference is most relevant at present. A preference encoder $G_\xi : H_t \to \mathbb{R}^d$ extracts $\mathbf{z}_t = G_\xi(H_t)$.

Direct regression $\mathbf{z}_t \approx U_\phi(s_t, \cdot)$ would entangle slowly-evolving preferences with rapidly-changing actions, preventing transferable representations. Instead, we factorize the preference signal via dual encoders: a context encoder $\mathbf{z}_t = G_\xi(H_t)$ for long-term orientations and an action encoder $\tilde{a}_t = A_\psi(a_t)$ for instantaneous action characteristics, enforcing consistency with U_ϕ via inner-product alignment:

$$\mathcal{L}_e(\xi,\psi) = \mathbb{E}_{(s_t,a_t^{\text{ovr}})\sim\mathcal{B}_{\text{drl}}}\Big[\big\|\langle \mathbf{z}_t, \tilde{a}_t\rangle - u_t\big\|_2^2\Big], \tag{23}$$

where $\tilde{a}_t = A_\psi(a_t^{\text{ovr}})$ encodes the executed (possibly overridden) action, and $u_t = U_\phi(s_t, a_t^{\text{ovr}})$ provides supervision. The multiplicative interaction $\langle \mathbf{z}_t, \tilde{a}_t\rangle$ naturally decomposes utility into orthogonal temporal scales, analogous to collaborative filtering. By decoupling stable preferences from transient actions, $\mathbf{z}_t$ learns context-aware representations that capture how a user's priorities shift across different operating conditions—encoding abstract preference orientations rather than concrete control sequences.

We implement G_ξ as a long short-term memory (LSTM) network for temporal modeling and A_ψ as a lightweight multilayer perceptron (MLP), providing sufficient expressiveness while maintaining computational efficiency.

C. Preference-conditioned DRL Agent

We integrate preference modeling into the SAC [33], a widely-used off-policy DRL method chosen for its entropy regularization and hybrid action space compatibility.

The agent maintains a stochastic policy $\pi_\theta(a_t|s_t,\mathbf{z}_t)$ and two Q-functions $Q_{\varphi_i}(s_t,\mathbf{z}_t,a_t)$, $i\in\{1,2\}$, with target networks $Q_{\varphi_i^{\text{tgt}}}$. Conditioning on $\mathbf{z}_t$ enables adaptive control across contexts and preference shifts.

Because the environment transitions according to the actually executed action $a_t = a_t^{\text{ovr}}$, the critic is trained on override-corrected transitions sampled from $\mathcal{B}_{\text{drl}}$:

$$\mathcal{L}_c(\varphi_i) = \mathbb{E}_{\mathcal{B}_{\text{drl}}}\Big[\big(Q_{\varphi_i}(s_t,\mathbf{z}_t,a_t) - y_t\big)^2\Big], \tag{24}$$

where a_t, r_t, and s_{t+1} in each sampled transition are all consistent with the executed (possibly overridden) action. The target value is:

$$\begin{aligned} y_t = r_t + \gamma\, \mathbb{E}_{a_{t+1}\sim\pi_\theta}\Big[&\min_j Q_{\varphi_j^{\text{tgt}}}(s_{t+1},\mathbf{z}_{t+1},a_{t+1}) \\ &- \alpha^{\mathcal{H}} \log \pi_\theta(a_{t+1}|s_{t+1},\mathbf{z}_{t+1})\Big]. \end{aligned} \tag{25}$$

The policy outputs hybrid actions (continuous Gaussian components and discrete Gumbel–Softmax) by minimizing:

$$\mathcal{L}_a(\theta) = \mathbb{E}_{s_t,\mathbf{z}_t,a_t}\Big[\alpha^{\mathcal{H}} \log \pi_\theta(a_t|s_t,\mathbf{z}_t) - \min_i Q_{\varphi_i}(s_t,\mathbf{z}_t,a_t)\Big], \tag{26}$$

where $(s_t,\mathbf{z}_t)\sim\mathcal{B}_{\text{drl}}$ and $a_t\sim\pi_\theta(\cdot|s_t,\mathbf{z}_t)$; the expectation of $-\log\pi_\theta$ is the policy entropy, giving both objectives the standard maximum-entropy form.

D. Training Procedure

The three modules update at different frequencies to balance training stability and computational efficiency.

Two replay structures support training. The DRL buffer $\mathcal{B}_{\text{drl}}$ stores complete episode trajectories, where each step records the state, recommended and executed actions, cached preference embedding, and reward. Storing trajectories (rather than isolated transitions) allows reconstruction of the history window H_t (Eq. (22)) by looking back τ_w steps within the same episode, which is essential for updating G_ξ with proper gradient flow. The override buffer $\mathcal{B}_{\text{ovr}}$ stores only the sparse override events $(s_t, a_t^{\text{ovr}}, a_t^{\text{rec}})$ for the BT loss on U_ϕ.

Specifically, the SAC agent performs a gradient step at every timestep (once $|\mathcal{B}_{\text{drl}}| \geq B_{\min}$), sampling a mini-batch of size K from $\mathcal{B}_{\text{drl}}$ and using the cached $\mathbf{z}_t$ as a fixed input (no gradient through G_ξ). The preference encoders

TABLE II
WORKED NUMERICAL EXAMPLE: ONE CONTROL STEP UNDER THE COMFORT PREFERENCE. REWARD COMPONENTS FOLLOW (12); NO LAUNDRY IS PENDING, SO $\mathcal{R}_{\mathrm{L}}$=1 FOR BOTH ACTIONS AND IS OMITTED. BOLD MARKS THE ACTION SELECTED AT EACH STAGE.

Quantity	a_t^{rec} (low-heat)	a_t^{ovr} (high-heat)
P_t^{a} / P_t^{D} (kW)	0.3 / 0.8	2.0 / 2.5
$\mathcal{R}_{\mathrm{c}} = e^{-\lvert P_t^D \rvert \kappa_t}$	0.73	0.37
$\mathcal{R}_{\mathrm{t}}$	0.37	1.00
$\mathcal{R}_{\mathrm{e}} = e^{-\lvert P_t^D \rvert \mu_t}$	0.67	0.29
$r^{\mathrm{sys}} = \sum_i \beta_i \mathcal{R}_i$ (equal β)	**0.69**	0.66
U_ϕ (converged)	-0.6	$+0.6$
$r_t = r^{\mathrm{sys}} + \alpha^u U_\phi$	0.09	**1.26**

G_ξ, A_ψ update every N steps by sampling transitions from $\mathcal{B}_{\mathrm{drl}}$, reconstructing H_t, and recomputing $\mathbf{z}_t = G_\xi(H_t)$ to enable backpropagation. The utility function U_ϕ updates every M steps ($M > N$) by sampling override pairs from $\mathcal{B}_{\mathrm{ovr}}$, providing stable supervision from sparse feedback. After G_ξ or U_ϕ updates, cached $\mathbf{z}_t$ and the utility component of r_t in $\mathcal{B}_{\mathrm{drl}}$ become slightly stale; however, because both modules update infrequently (N and M steps, respectively) with small per-step parameter changes, this staleness is bounded and consistent with the off-policy nature of replay-based learning, where all buffered transitions are inherently collected under earlier policy parameters. Algorithm 1 outlines both phases; at deployment (Phase 2), only G_ξ and π_θ are needed.

E. Illustrative Example

To make Algorithm 1 concrete, we trace one control step with explicit numbers, isolating the AC decision (idealized for interpretability). Reward values follow (12)–(13); U_ϕ and the embedding are illustrative and shown at their converged values.

Step 1. On a peak-price winter evening (κ_t=0.40 \$/kWh, μ_t=0.50 kg/kWh, T_t^{in}=19 °C, setpoint 21±1 °C, base load 0.5 kW, occupied, no laundry), a comfort-oriented resident has preference embedding $\mathbf{z}_t$=$[0.2, 1.0]$ (a 2-D projection for readability).

Step 2 (Phase 1). Before preference learning, the controller recommends the low-heat action a_t^{rec} (P_t^{a}=0.3 kW, leaving T_t^{in} 2 °C below setpoint), the choice a fixed-weight controller with equal weights would also make (higher system reward; Table II). The comfort-oriented teacher overrides with the high-heat action a_t^{ovr} (P_t^{a}=2.0 kW), stored in $\mathcal{B}_{\mathrm{ovr}}$.

Step 3. Repeated slow-timescale BT updates (21) drive $U_\phi(s_t, a_t^{\mathrm{ovr}})$=+0.6 and $U_\phi(s_t, a_t^{\mathrm{rec}})$=−0.6 ($\Delta U$=1.2, $\sigma(\Delta U)$=0.77), and the medium-timescale embedding updates (23) make $\langle \mathbf{z}_t, \tilde{a}_t \rangle$ reproduce them ($\tilde{a}^{\mathrm{ovr}}$=$[-0.5, 0.7]$, $\tilde{a}^{\mathrm{rec}}$=$[0.6, -0.72]$). The same actions yield −0.36/+0.46 for a cost-oriented user $\mathbf{z}_t$=$[1.0, 0.2]$—opposite rankings no fixed weight vector can express.

Step 4 (Phase 2). Adding the converged utility to the reward (13) flips the ranking (r_t=1.26 for a_t^{ovr} vs. 0.09; Table II): the deployed policy recommends a_t^{ovr}, matching the resident, so the override vanishes—whereas a fixed-weight controller, unable to absorb overrides, keeps being overridden on similar evenings.

Algorithm 1 Training and Deployment of Preference-Aware HEMS

Input: Utility U_ϕ, encoders G_ξ, A_ψ, SAC agent $(\pi_\theta, Q_{\varphi_1}, Q_{\varphi_2})$, buffers $\mathcal{B}_{\mathrm{drl}}, \mathcal{B}_{\mathrm{ovr}}$.
Parameters: Discount γ, entropy temperature $\alpha^{\mathcal{H}}$, mini-batch size K, minimum buffer size $B_{\min}$, encoder update interval N, utility update interval M.
Output: Deployed policy $\pi_\theta(a|s, \mathbf{z})$.
Phase 1: Training
Select teacher $\pi^{(k)}$ (fixed for all episodes).
for each training episode **do**
  Initialize s_0; compute $\mathbf{z}_0 = G_\xi(H_0)$.
  for each timestep t **do**
    Agent recommends $a_t^{\mathrm{rec}} \sim \pi_\theta(\cdot|s_t, \mathbf{z}_t)$.
    Teacher provides $a_t^{\mathrm{pref}} = \pi^{(k)}(s_t)$.
    Generate override a_t^{ovr} via Eq. (19).
    Execute $a_t \leftarrow a_t^{\mathrm{ovr}}$; observe (s_{t+1}, r_t).
    Compute $\mathbf{z}_{t+1} = G_\xi(H_{t+1})$.
    Append $(s_t, a_t^{\mathrm{rec}}, a_t^{\mathrm{ovr}}, \mathbf{z}_t, r_t, s_{t+1}, \mathbf{z}_{t+1})$ to current episode in $\mathcal{B}_{\mathrm{drl}}$.
    if $a_t^{\mathrm{ovr}} \neq a_t^{\mathrm{rec}}$ **then**
      Store $(s_t, a_t^{\mathrm{ovr}}, a_t^{\mathrm{rec}})$ in $\mathcal{B}_{\mathrm{ovr}}$.
    end if
    if $|\mathcal{B}_{\mathrm{drl}}| \geq B_{\min}$ **then**
      [Fast] Sample K transitions from $\mathcal{B}_{\mathrm{drl}}$; update SAC via Eqs. (24), (26).
    end if
    if $t \bmod N = 0$ **then**
      [Medium] Sample transitions from $\mathcal{B}_{\mathrm{drl}}$; reconstruct H_t and recompute $\mathbf{z}_t = G_\xi(H_t)$; update G_ξ, A_ψ via Eq. (23).
    end if
    if $t \bmod M = 0$ **then**
      [Slow] Sample from $\mathcal{B}_{\mathrm{ovr}}$; update U_ϕ via Eq. (21).
    end if
  end for
end for
Phase 2: Deployment (using G_ξ and π_θ only)
Freeze all trained parameters.
for each deployment episode **do**
  Initialize s_0; compute $\mathbf{z}_0 = G_\xi(H_0)$.
  for each timestep t **do**
    Agent recommends $a_t^{\mathrm{rec}} \sim \pi_\theta(\cdot|s_t, \mathbf{z}_t)$.
    User accepts or overrides: $a_t \leftarrow a_t^{\mathrm{ovr}}$.
    Observe s_{t+1}; update H_{t+1}; compute $\mathbf{z}_{t+1} = G_\xi(H_{t+1})$.
  end for
end for

V. EXPERIMENT

We evaluate the proposed framework along five dimensions: (1) preference alignment—whether each model internalizes its target preference and outperforms baselines (Sections V-B–V-C); (2) component and device contribution—which design choices drive alignment (Section V-D); (3) robustness—sensitivity to override parameters, feedback noise, and

TABLE III
HYPERPARAMETER SUMMARY.

Category	Parameter	Value	Rationale
MDP	Timestep Δt	15 min	AMI standard interval
	Episode length τ	96 steps	One day
Override	Soft threshold τ_s	0.10	Low discrepancy tolerance
	Hard threshold τ_h	0.60	Deterministic correction
	Min probability $p_{\min}$	0.02	Rare spontaneous override
	Max probability $p_{\max}$	0.80	Stochastic zone ceiling
Update schedule	SAC policy	every step	Off-policy; mini-batch K=32
	Pref. encoder N	256 steps	Medium: track pref. drift
	Utility function M	1024 steps	Slow: stable supervision
Training	Episodes	5000	Fixed budget per preference
	Warm-up $B_{\min}$	1024	Buffer diversity
	Discount γ	0.99	Long-horizon planning
	Polyak τ_{polyak}	0.005	Soft target update
	Learning rate	3×10^{-4}	Adam optimizer
	Replay buffer	10^6	Capacity
Embedding	Embedding dim d_z	32	Compact representation
	History window τ_w	32 steps	8 hours of context
Reward	Pref. weight α^u	1.0	Balance with r^{sys}

time-step resolution (Section V-E); (4) adaptation—transfer to new preferences and online tracking of preference drift (Section V-F); and (5) scalability—computational cost in multi-household deployments (Section V-G).

A. Experimental Setup

We instantiate the HEMS using real-world traces: base load and metadata from Pecan Street (San Diego apartment) [30], and outdoor temperature and irradiance from NOAA for the same region [31]. For each month, one day is reserved for validation and one for testing, with the remaining days used for training. All results are averaged over 10 random seeds.

The control MDP uses 15-min timesteps and one-day episodes (Table III). Reward weights β sum to one (Table IV); five prototypical preferences define teacher policies for synthetic override generation (Eq. (19)). Each preference is trained independently from random initialization: one teacher is assigned per run and held constant across all episodes. The main comparison (Table V) uses the four skewed preferences; the Balanced preference serves as the weight vector for the static baselines (SAC-s-β, MPC-s-β).

All methods share the same SAC [33] backbone and training budget (Table III). Actor and critic networks use MLPs with hidden dimensions [256, 512, 64, 512, 256]. The utility network U_ϕ shares the same backbone; input features are normalized via LayerNorm before concatenation, and output scale is regularized (coefficient $\lambda_s = 0.01$) to prevent logit explosion. The preference encoder G_ξ is a two-layer LSTM (256 units) with a [256, 32] projection head; the action encoder A_ψ is a two-layer MLP [32, 32]; their inner product operates in a shared 32-dimensional embedding space. Experiments run on an Intel i7-14600K CPU and an NVIDIA RTX 4090 GPU.

The linear weights $\beta^{(k)}$ in Table IV define the reward-function weights used to train teacher policies for synthetic override generation (Section IV-A1); they do not enter or constrain the preference learning modules. Because $\mathbf{z}_t = G_\xi(H_t)$ and U_ϕ are realized by neural networks (LSTM and MLP), the framework can capture nonlinear, context-dependent preference structures beyond what any fixed weight vector can express.

TABLE IV
TEACHER-POLICY REWARD WEIGHTS $\beta^{(k)}$ FOR SYNTHETIC OVERRIDE GENERATION.

ID	Preference	β_c	β_t	β_e	β_l
1	Cost-saving	0.55	0.15	0.15	0.15
2	Comfort-prioritized	0.15	0.55	0.15	0.15
3	Low-carbon	0.15	0.15	0.55	0.15
4	Fast-laundry	0.15	0.15	0.15	0.55
5	Balanced	0.25	0.25	0.25	0.25

1) Baselines: We compare against baselines spanning three categories.

No human feedback:

1) **RBC** [34]. Rule-based control using fixed heuristics (e.g., thermostat bands, time-of-use switching) with no learning.
2) **SAC-s-β**. Standard SAC trained with the Balanced weight vector from Table IV, ignoring user feedback entirely.
3) **MPC-s-β** [35]. Rolling-horizon MPC that maximizes $\sum_{k=0}^{H-1} r_{t+k}^{\text{sys}}$ subject to Eqs. (2)–(7) over H=24 steps with perfect forecasts, formulated as a mixed-integer linear program (MILP) with binary variables for AC mode and WM scheduling, and solved with Gurobi [36]. The Balanced weight vector is used regardless of the evaluation preference. Together with SAC-s-β, this pair shares the same fixed weights but differs in optimization method (MILP planning vs. RL), enabling direct comparison.

Explicit human feedback:

4) **Discrete-Feedback (DF-7L)** [37]. Reward shaping with 7-level comfort labels:

$$r_t \;=\; r_t^{\text{sys}} + \alpha^d d_t, \tag{27}$$

where $d_t = 3 - \lfloor 6\,\hat{\delta}_t + U_t \rfloor \in \{3,2,1,0,-1,-2,-3\}$ with $\hat{\delta}_t = \text{clip}((\delta_t - \tau_s)/(\tau_h - \tau_s), 0, 1)$, $U_t \sim \text{Uniform}(0,1)$, and discrepancy δ_t from Eq. (18). This maps small discrepancies to satisfaction and large ones to dissatisfaction.
5) **Continuous-Feedback (CF-Reg)** [6]. A regressor $\hat{d}_t = \mathcal{G}^d(d_t)$ fits the discrete labels above and is added to r_t^{sys} with a fixed coefficient.

Implicit human feedback:

6) **RLHF-Multiplicative Shaping (RLHF-MS)** [14]. Learns a utility model $u_t = \hat{U}(s_t, a_t)$ from pairwise comparisons for multiplicative reward shaping:

$$r_t \;=\; -\,r_t^{\text{sys}} \cdot \frac{u^{\text{upp}}}{u^{\text{upp}} + u_t}, \tag{28}$$

where u^{upp} is an upper bound. Unlike our method, the learned utility is not provided as input to the policy.
7) **Two-Stage**. A policy-independent baseline with the same architecture and training budget as our method, decoupling preference learning from policy optimization. In Phase 1, the policy π_θ is held at random initialization while U_ϕ, G_ξ, and A_ψ are trained from the resulting abundant overrides via the same losses (Eqs. (21), (23)).

In Phase 2, all preference modules are frozen and only π_θ and Q_φ are updated via SAC. This isolates the contribution of joint co-evolution.

2) Evaluation Protocol: After training converges, all learnable parameters (π_θ, Q_φ, U_ϕ, G_ξ, A_ψ) are frozen. Evaluation proceeds on held-out test days that were never used during training. Each test episode is run in two modes under the same random seed:

1) **Policy-only mode** ($\mathcal{O}_i^{\text{rec}}$): the frozen policy $\pi_\theta(\cdot|s_t, \mathbf{z}_t)$ generates actions without any teacher override ($a_t = a_t^{\text{rec}}$). The preference embedding $\mathbf{z}_t$ is still computed by the frozen G_ξ from the interaction history, but no new overrides are injected.
2) **Deployment mode** ($\mathcal{O}_i^{\text{ovr}}$): the same frozen policy generates recommendations, but the teacher policy provides overrides via Eq. (19) as in training. The executed action is $a_t = a_t^{\text{ovr}}$.

Comparing these two modes under identical exogenous conditions (fixed historical records for base load, weather, pricing, and carbon intensity) isolates the contribution of user corrections from the policy's autonomous capability. Dynamic system states (indoor temperature, battery SOC, etc.) differ across runs as they depend on agent actions via Eqs. (2)–(7). Override behavior is generated in real time by teacher policies observing the current state, following the synthetic-preference paradigm in preference-based RL [15], [25].

3) Evaluation Metrics: Traditional HEMS evaluations report system-level objectives (cost, comfort, carbon) but cannot distinguish whether good performance arises from the policy itself or from frequent user corrections. We introduce three complementary metrics: Override Rate (OR) and Override Dependence Index (ODI) evaluate alignment quality at test time, while User Interaction Cost (UIC) evaluates alignment cost during training.

a) Override Rate (OR): Under deployment mode, let $O_t \in \{0, 1\}$ indicate an override at time t generated by Eq. (19). The per-episode rate is

$$\text{OR} \;=\; \frac{1}{\tau}\sum_{t=1}^{\tau} O_t, \tag{29}$$

i.e., the fraction of timesteps where the agent's recommendation is rejected (lower is better).

b) Override Dependence Index (ODI): For each objective $i \in \{\text{c,t,e,l}\}$, the per-objective relative change due to overrides is

$$\text{ODI}_i \;=\; \frac{\left|\mathcal{O}_i^{\text{rec}} - \mathcal{O}_i^{\text{ovr}}\right|}{\mathcal{O}_i^{\text{ovr}}}, \tag{30}$$

where $\mathcal{O}_i^{\text{rec}}$ and $\mathcal{O}_i^{\text{ovr}}$ are the objective values under policy-only and deployment modes, respectively. A low ODI confirms that the policy has internalized user preferences rather than relying on ongoing corrections. We report $\left[\text{ODI}_\text{c}, \text{ODI}_\text{t}, \text{ODI}_\text{e}, \text{ODI}_\text{l}\right]$ per preference; lower is better.

c) User Interaction Cost (UIC): Let $A_t \in \{0, 1\}$ denote an active feedback event (e.g., a survey rating) and $O_t \in \{0, 1\}$ a passive override (e.g., a button press). The per-step interaction cost is

$$\text{UIC} \;=\; \frac{1}{T}\sum_{t=1}^{T}\left(w_{\text{act}}A_t + w_{\text{pas}}O_t\right), \quad T = \sum_{\text{train episodes}} \tau, \tag{31}$$

with weights w_{act}=4 and w_{pas}=1 based on time-cost evidence (survey ∼10.3 s [38] vs. point-and-click ∼2.5–3.0 s [39]). Note that OR and ODI are computed on frozen models over test episodes only, while UIC is accumulated over the training horizon.

B. Performance Analysis

Table V shows that all learning-based controllers significantly outperform RBC across preference profiles. RBC relies on fixed heuristics and static comfort bands, preventing adaptation to temporal variations in household behavior. Consequently, it exhibits high override rates (OR>0.2) and large objective deviations, reflecting frequent misalignment with evolving user intent.

SAC-s-β and MPC-s-β both use the Balanced weight vector without preference feedback. Despite MPC-s-β having access to rolling-horizon optimization with perfect foresight, it yields OR comparable to SAC-s-β, confirming that without preference knowledge, even optimal planning cannot align with heterogeneous user intents. This highlights a central limitation of fixed-weight methods: they perform well only when the assumed weights happen to match the user's actual preference, whereas our framework discovers preferences implicitly from overrides.

Among feedback-driven methods, DF-7L shows the weakest performance. The discrete 7-level labels introduce quantization noise and reduce gradient precision, while each label requires explicit user input, yielding the highest interaction cost (UIC=2.327). CF-Reg alleviates these issues by regressing continuous satisfaction scores from limited labeled data, improving convergence and reducing user effort (UIC=1.296). However, CF-Reg remains purely reward-shaped without mechanisms to capture user context or long-term preference evolution, limiting adaptability across preference modes.

The remaining three methods—RLHF-MS, Two-Stage, and ours—rely exclusively on implicit feedback from passive overrides, keeping user burden low (UIC<1.1). RLHF-MS applies the learned utility multiplicatively, rescaling rewards and distorting objective trade-offs; without a learned preference vector, it also cannot anticipate context-dependent preference shifts. Two-Stage shares our architecture and training budget but decouples preference learning from policy optimization. Its OR increases relative to joint training, which we attribute to distribution shift: the preference modules, trained on data from a random policy that generates frequent, large-magnitude overrides, cannot generalize to the refined override distribution encountered once the policy improves in Phase 2.

Our framework additively integrates the learned utility and conditions both actor and critic on a dynamic embedding $\mathbf{z}_t$, preserving objective scaling and enabling continuous tracking of user intent. Joint co-evolution avoids the distribution shift

TABLE V
ALIGNMENT AND OVERRIDE RELIANCE PER PREFERENCE (MEANS OVER SEEDS; LOWER IS BETTER). "–" INDICATES METHODS WITHOUT HUMAN-IN-THE-LOOP TRAINING (EFFECTIVELY UIC≈0).

Method	Cost-saving					Comfort-prioritized					Low-carbon					Fast-laundry					UIC (train)
	OR	ODI_c	ODI_t	ODI_e	ODI_l	OR	ODI_c	ODI_t	ODI_e	ODI_l	OR	ODI_c	ODI_t	ODI_e	ODI_l	OR	ODI_c	ODI_t	ODI_e	ODI_l	
RBC	0.372	0.777	0.295	0.818	0.783	0.230	0.311	1.810	0.191	0.754	0.312	0.688	0.222	1.205	0.798	0.229	0.477	0.610	0.474	0.590	–
SAC-s-β	0.132	0.147	0.238	0.149	0.125	0.119	0.123	0.524	0.209	**0.037**	0.148	0.098	0.276	0.360	0.176	0.093	0.024	0.068	0.045	0.400	–
MPC-s-β	0.125	0.140	0.250	0.142	0.118	0.112	0.115	0.505	0.200	0.040	0.138	0.090	0.262	0.340	0.165	0.085	0.022	0.062	0.040	0.385	–
DF-7L	0.146	0.245	0.364	0.141	0.335	0.061	0.069	0.282	0.098	0.053	0.103	0.293	0.358	0.184	0.417	0.083	0.056	0.178	0.094	0.127	2.327
CF-Reg	0.082	0.069	0.229	**0.057**	0.119	0.043	**0.014**	0.148	0.046	0.092	0.082	0.116	0.270	**0.037**	0.111	0.053	**0.019**	0.082	**0.027**	0.079	1.296
RLHF-MS	0.108	0.071	0.073	0.131	0.095	0.079	0.058	0.212	0.087	0.212	0.073	**0.025**	0.097	0.102	0.185	0.064	0.027	0.052	0.037	0.118	1.032
Two-Stage	0.068	0.038	0.058	0.098	0.032	0.045	0.090	0.018	0.035	0.088	0.052	0.032	0.055	0.115	0.022	0.042	0.055	0.015	0.082	0.048	0.512
Ours	**0.043**	**0.020**	**0.042**	0.081	**0.018**	**0.028**	0.087	**0.005**	**0.019**	0.085	**0.031**	0.029	**0.036**	0.108	**0.007**	**0.028**	0.053	**0.003**	0.078	**0.035**	**0.394**

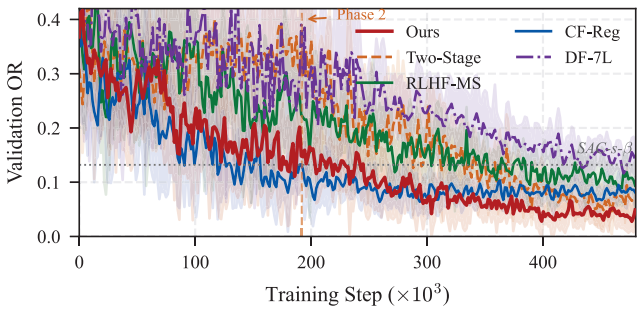

Fig. 4. Training convergence of validation override rate (OR) under the Cost-saving preference. Two-Stage trains preference modules in Phase 1 (steps 0–192k) with a random policy, then freezes them and trains the policy in Phase 2. The dotted reference line shows SAC-s-β (no feedback, fixed Balanced weights).

TABLE VI
CROSS-PREFERENCE OR MATRIX (SAN DIEGO) AND CROSS-CLIMATE MATCHED OR. DIAGONAL ENTRIES ARE MATCHED PREFERENCE (BOLD); THE RIGHTMOST COLUMN REPORTS MATCHED OR WHEN RETRAINED ON AUSTIN, TX TRACES.

	San Diego (Eval Teacher)				Austin
Train	Cost	Comf.	Carbon	Laund.	Matched
Cost	**0.043**	0.218	0.095	0.176	0.051
Comfort	0.224	**0.028**	0.198	0.152	0.034
Carbon	0.103	0.207	**0.031**	0.165	0.039
Laundry	0.189	0.155	0.174	**0.028**	0.033

that degrades Two-Stage. Across all four preference profiles, our method achieves the lowest OR and the best or near-best ODI on the preference-relevant dimensions, while maintaining the lowest UIC among feedback-based approaches (Table V). The same ranking holds under the Balanced preference (OR=0.021), where no-feedback baselines trivially match the teacher since their fixed weights coincide with the target.

Fig. 4 shows the training dynamics under the Cost-saving preference. Our method converges fastest in OR, benefiting from joint co-evolution of preference learning and policy optimization. Two-Stage remains flat during Phase 1 (data collection with random policy) and drops sharply only after Phase 2 begins, but plateaus at a higher level due to distribution shift. CF-Reg drops quickly in early steps thanks to direct regression but stagnates once its reward-shaping capacity saturates. RLHF-MS converges steadily but to a higher ceiling, limited by its multiplicative utility integration.

As with all DRL-based HEMS, training runs offline (480k steps ≈ 2.4 h on commodity hardware; Table VIII), so the adaptation timeline is bounded by computation, not by calendar days of occupant interaction. UIC remains meaningful: lower UIC indicates fewer override events needed in training data, easing data collection requirements.

C. Preference Alignment

Since each preference is trained independently with its own teacher (Section IV-A1), we assess alignment through two complementary analyses: cross-preference evaluation and utility function inspection.

a) Cross-preference evaluation: Table VI presents the cross-preference OR matrix. Diagonal entries (matched preference) are 2–8× lower than off-diagonal entries, confirming that each model has internalized its target preference rather than learning a generic policy. The off-diagonal structure is also interpretable: the Cost–Carbon pair yields the lowest cross-OR (0.095/0.103), consistent with their shared tendency to reduce grid import, whereas Cost–Comfort is the most divergent pair (0.218/0.224), reflecting the inherent tension between energy saving and thermal comfort.

Beyond cross-preference consistency, we further assess climate robustness by retraining each model on Austin, TX weather traces (humid subtropical, vs. San Diego's semi-arid climate). All Austin matched-preference OR values remain at or below 0.051, comparable to the San Diego diagonal, confirming that preference learning generalizes across climate conditions.

b) Utility function inspection: To verify that U_ϕ captures the intended preference direction, we record a one-day trajectory from the Balanced teacher and evaluate all four trained utility models on the same state–action sequence (Fig. 5). Despite receiving identical state–action inputs, the four models produce visibly different u_t temporal patterns. The Cost-saving model responds most strongly during peak-price intervals (hatched regions), assigning negative scores to grid-intensive actions that the Balanced teacher takes during expensive hours. The Comfort model reacts to thermal deviations: u_t drops when T^{in} moves away from the comfort band and recovers when temperature is well-regulated. The Low-carbon model assigns positive scores during periods of high PV generation

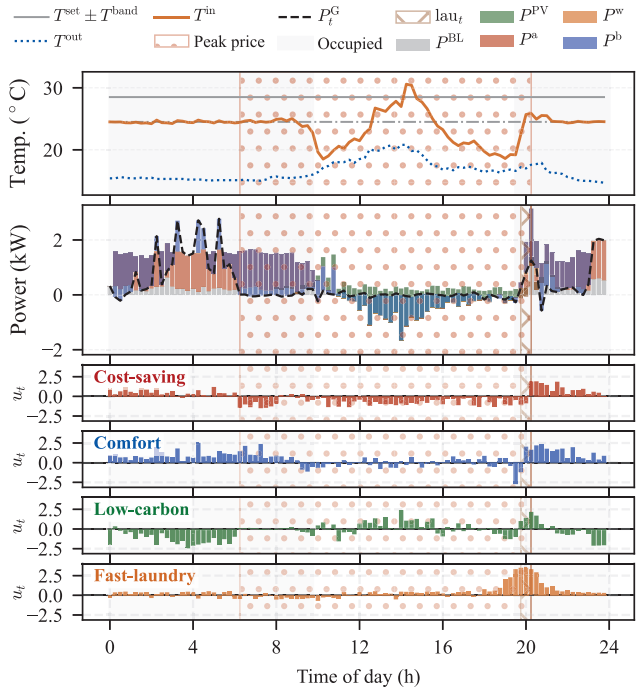

Fig. 5. Learned utility scores on a one-day Balanced-teacher trajectory. Upper two panels: temperature profile and stacked power components. Lower four panels: $U_\phi(s_t, a_t)$ from four independently trained preference models on the same state–action sequence.

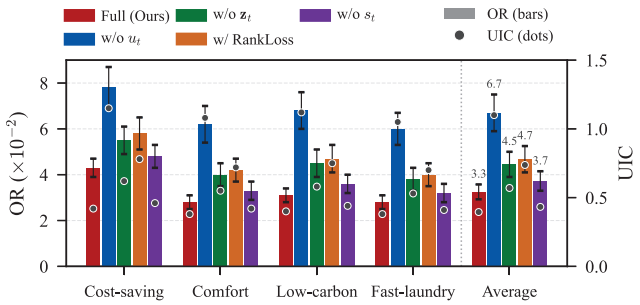

Fig. 6. Ablation of key components. Solid bars show OR (left axis) and dots show UIC (right axis, accumulated over training), per preference and averaged over preferences (rightmost group); numeric labels on the average group denote OR ($\times 10^{-2}$).

or BESS discharge that reduce net grid draw. The Fast-laundry model shows a distinct pattern concentrated around laundry request events. These divergent responses confirm that each U_ϕ has learned a preference-specific evaluation, providing meaningful supervision for the preference encoder G_ξ.

D. Ablation Study

We examine four variants (Fig. 6): (i) w/o u_t — removing the learned utility term from the reward (α^u=0 in (13)); (ii) w/o $\mathbf{z}_t$ — removing the preference embedding, so (14) becomes $\pi_\theta(a_t \mid s_t)$ while retaining utility shaping; (iii) w/ RankLoss — replacing the Bradley–Terry loss (21) with margin-based ranking loss $\mathcal{L}_u = \sum_t \max(0, m - \Delta U_t)$ (m=1); (iv) w/o s_t — excluding the current state s_t from the preference encoder input in (22).

TABLE VII
SENSITIVITY ANALYSIS. (A) OVERRIDE PARAMETERS: EACH IS VARIED INDEPENDENTLY WHILE OTHERS STAY AT DEFAULT (*). (B) FEEDBACK NOISE: A FRACTION η OF OVERRIDES ARE RANDOMLY FLIPPED. (C) TIME STEP: EPISODE LENGTH SCALES TO MAINTAIN A ONE-DAY HORIZON.

Group	Setting	OR (↓)	UIC (↓)	ODI_{avg} (↓)
τ_s	0.05	0.012	0.53	0.020
	0.10*	0.021	0.38	0.034
	0.15	0.036	0.23	0.052
	0.20	0.063	0.12	0.088
τ_h	0.20	0.847	0.94	0.318
	0.40	0.374	0.78	0.441
	0.60*	0.021	0.38	0.034
	0.80	0.095	0.17	0.124
	1.00	0.167	0.12	0.222
$p_{\min}$	0.01	0.031	0.24	0.046
	0.02*	0.021	0.38	0.034
	0.05	0.015	0.54	0.024
	0.10	0.011	0.68	0.018
Flip η	0 (clean)	0.021	0.38	0.034
	0.05	0.025	0.38	0.040
	0.10	0.033	0.40	0.050
	0.15	0.051	0.42	0.074
Δt	5 min (288 steps)	0.015	0.33	0.026
	15 min* (96 steps)	0.021	0.38	0.034
	30 min (48 steps)	0.031	0.45	0.047

Averaged across the four preferences (rightmost group in Fig. 6), removing the learned utility (w/o u_t) raises the override rate the most—roughly doubling it from the full model's ≈ 0.033—while removing the context-aware embedding (w/o $\mathbf{z}_t$) raises it only about a third as much. This isolates their roles: the utility is the primary alignment driver, supplying the direct feedback signal without which the agent reverts to system-level optimization and loses personalization, whereas the embedding is a complementary contributor that provides contextual cues about user intent (UIC follows the same ordering). The other two variants are secondary design checks: substituting a margin ranking loss for the BT loss also raises OR markedly, confirming the robustness advantage of the probabilistic BT formulation under noisy overrides, while excluding the current state (w/o s_t) has the smallest effect, indicating that historical action–override pairs already capture most of the information needed to infer user intent.

E. Sensitivity Analysis

We assess robustness to override generation parameters, feedback noise, and control time step. All experiments use the Balanced preference and report OR and UIC averaged over 10 seeds.

a) Override parameters: We independently vary τ_s, τ_h, and $p_{\min}$ in Eq. (19) (Table VII, top). The soft threshold τ_s and $p_{\min}$ show moderate but interpretable effects. As τ_s increases, more actions fall below the soft threshold into the rare-override zone ($\delta_t < \tau_s$): UIC falls (from 0.53 to 0.12) because fewer discrepancies trigger corrections, while OR rises (0.012 to 0.063) as the sparser feedback weakens preference learning. Raising $p_{\min}$ increases the baseline override probability, expanding UIC (0.24 to 0.68) while modestly improving alignment (OR from 0.031 to 0.011), consistent with more

frequent but lower-quality corrections. Both parameters show smaller OR sensitivity than τ_h; $p_{\min}$ in particular leaves OR nearly flat once τ_h is fixed.

The hard threshold τ_h exhibits a non-monotone, U-shaped effect. Too low (τ_h=0.20): nearly all actions are overridden (training OR>90%), so the DRL buffer is dominated by teacher actions. The Critic then learns $Q(s, a^{\text{teacher}})$, and the Actor queries Q at out-of-distribution student actions, triggering extrapolation error analogous to offline RL [40]. The student never learns to act independently (OR=0.847); even at τ_h=0.40 (OR=0.374), partial recovery confirms lasting damage from this covariate shift [41]. Too high ($\tau_h \geq 0.80$): corrections are reserved for severe discrepancies only, making the preference signal sparse; at the extreme τ_h=1.00, deterministic corrections vanish entirely (OR=0.167). The default τ_h=0.60 balances informative deterministic corrections against sufficient student exploration for stable SAC learning.

b) Feedback noise: While Eq. (19) already models stochastic override frequency, real users may also provide directionally wrong corrections. We simulate this by randomly flipping a fraction η of override decisions (Table VII, middle); the BT loss absorbs such flips as low-probability events. Performance degrades gracefully: OR increases from 0.021 to only 0.051 at η=0.15, demonstrating the BT formulation's inherent noise tolerance.

c) Time-step resolution: We retrain with $\Delta t \in \{5, 15, 30\}$ min, scaling the episode length to preserve a one-day horizon (Table VII, bottom). Finer resolution reduces both OR and UIC: at 5 min, smoother control actions reduce the frequency of teacher corrections (UIC=0.33) and improve final alignment (OR=0.015); at 30 min, coarser discretization produces larger per-step errors that trigger more overrides (UIC=0.45, OR=0.031). Sub-minute resolution is not evaluated: residential devices (AC, WM) have physical response times exceeding one minute, and the resulting 1440-step episodes would substantially increase computational cost without commensurate benefit.

F. Adaptation and Generalization

We evaluate adaptation under three settings. In the first two, the system is aware that a preference change has occurred (e.g., a new occupant): buffers are cleared and all modules are fine-tuned from the converged source model. In the third, no notification is given—parameters and buffers carry over, and adaptation must emerge from the shifting override distribution alone.

- **Prototype-to-prototype transfer:** the target is another prototypical preference from Table IV, testing whether pre-trained modules provide a useful warm start.
- **Novel-preference adaptation:** the target follows a behavioral rule outside the prototype space—e.g., *"AC runs only in the evening and after wake-up."* The teacher is a SAC agent whose reward penalizes HVAC actuation outside designated windows, testing generalization beyond the training preference space.
- **Preference drift:** the teacher silently switches from Cost-saving ($\beta^{(1)}$) to Comfort ($\beta^{(2)}$) at step 96k, testing online tracking of preference evolution (Assumption A2).

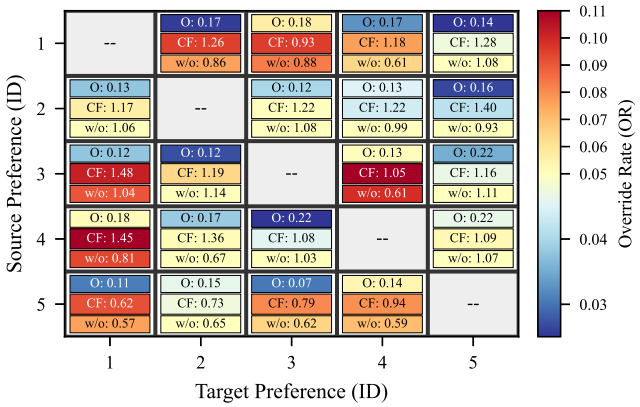

Fig. 7. Prototype-to-prototype preference transfer. Each cell contains three horizontal strips (top: Ours; middle: CF-Reg; bottom: Ours w/o Preference Encoder). Color denotes post-transfer OR (lower is better); text shows fine-tuning UIC. Diagonal (no transfer): "–".

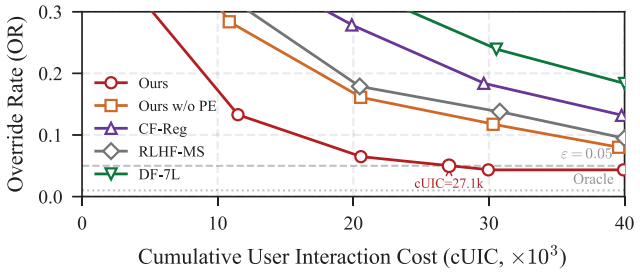

Fig. 8. Novel-preference adaptation (AC runs only in the evening and after wake-up). OR decreases as cumulative user interaction cost (cUIC) grows.

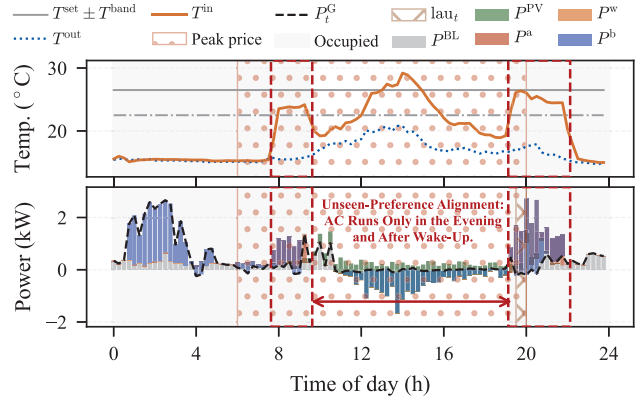

Fig. 9. Novel-preference adaptation case. Top: temperature curve; bottom: power profile.

a) Prototype-to-prototype transfer: Across all source→target transfers among the five prototypes, Fig. 7 shows that our method achieves consistently lower post-transfer OR and smaller fine-tuning UIC than baselines. Because the pre-trained G_ξ and U_ϕ provide a warm initialization that captures general preference structure, the model requires fewer override interactions to re-align with the new target. Transfers involving the Balanced profile (ID 5) exhibit lower UIC, reflecting its role as a neutral intermediary.

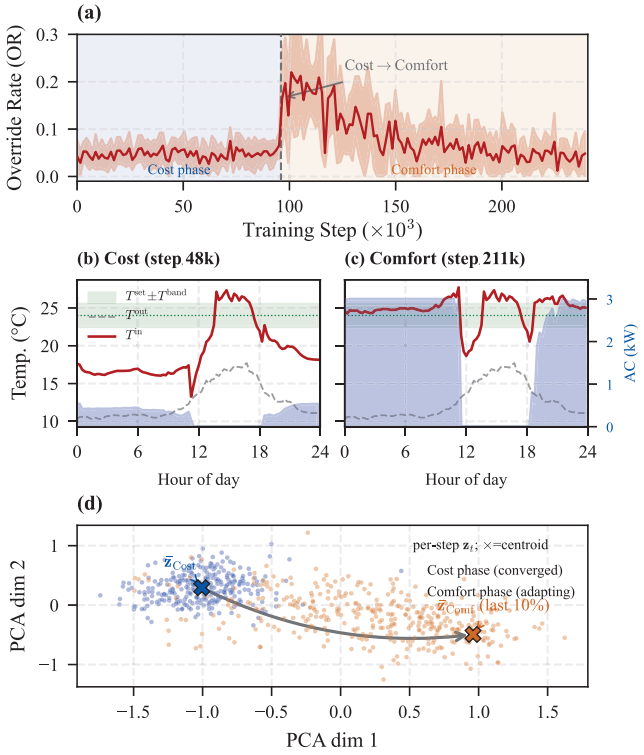

Fig. 10. Preference drift (Cost→Comfort at step 96k, no reset). (a) Validation OR over 240k training steps (±1 std, 10 seeds). (b)–(c) Temperature and AC power on the same day under Cost-phase vs. Comfort-phase control. (d) PCA-projected per-timestep $\mathbf{z}_t$ from validation episodes; $\bar{\mathbf{z}}_{\text{Cost}}$ and $\bar{\mathbf{z}}_{\text{Comf}}$ denote phase centroids.

b) Novel-preference adaptation: Using cumulative interaction cost (cUIC) as the budget axis, Fig. 8 shows that our approach achieves target OR with the smallest cUIC. Variants without the Preference Encoder and RLHF-MS perform slightly worse, while CF-Reg and DF-7L require substantially larger budgets. Fig. 9 illustrates behavior under the novel rule (*AC active only in the evening and after wake-up*), where the policy correctly limits HVAC actuation to the specified periods with power traces consistent with thermal dynamics.

c) Preference drift and step-by-step case study:

Fig. 10(a) shows that OR spikes at step 96k as Cost-oriented actions misalign with the new Comfort teacher, then gradually recovers over approximately 144k additional steps—without any explicit re-initialization. The recovery is non-monotone: temporary setbacks occur as the policy explores new action distributions, before ultimately converging to the Comfort-aligned behavior. This is substantially faster than learning from scratch (∼288k steps; Fig. 4), since the agent retains its environment model and only reshapes its preference representation. Figs. 10(b)–(c) contrast the same winter day evaluated under each phase's control strategy: during the Cost phase, the policy minimizes heating to reduce electricity cost, tolerating indoor temperature excursions below the setpoint; after adaptation to the Comfort preference, it maintains tight temperature regulation near the setpoint at the expense of higher grid consumption. Fig. 10(d) shows the PCA-projected per-timestep embeddings $\mathbf{z}_t$ sampled from validation episodes

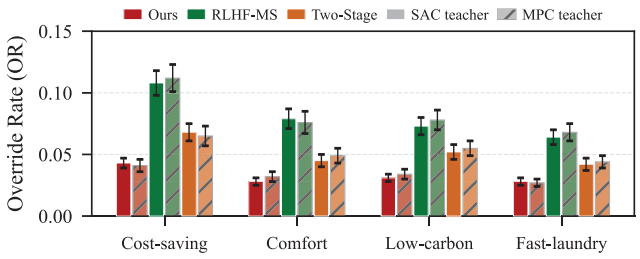

Fig. 11. Teacher-agnostic validation. Matched OR under SAC vs. MPC teacher across four preferences for three methods (Ours, RLHF-MS, Two-Stage). Hatched bars denote MPC teacher. Despite fundamentally different override characteristics, all methods achieve comparable alignment under both teachers, with Ours maintaining the lowest OR in all cases.

during training. The Cost-phase points form a tight cluster reflecting the converged model, while the Comfort-phase points spread progressively along the adaptation trajectory as the encoder reshapes its representation. The Comfort centroid $\bar{\mathbf{z}}_{\text{Comf}}$ is well separated from $\bar{\mathbf{z}}_{\text{Cost}}$, confirming that the encoder tracks the evolving preference without explicit change-point detection.

d) Teacher-agnostic learning: To verify that the framework does not depend on a specific teacher policy class, we repeat the main experiments using an MPC teacher in place of the default SAC teacher. The MPC teacher solves the same weighted-sum objective with the true preference weights β^* and a rolling horizon of $H{=}24$ steps with perfect forecasts, producing deterministic overrides that differ from the SAC teacher's stochastic corrections in both frequency and magnitude.

Fig. 11 compares matched OR when our method is trained with SAC vs. MPC teachers. Despite the MPC teacher producing qualitatively different override patterns—deterministic, finite-horizon corrections rather than stochastic policy-derived ones—our method achieves comparable alignment across all four preferences. This confirms that the preference learning modules (G_ξ, U_ϕ) extract preference information from the structure of overrides (which actions are corrected and in what direction), not from artifacts of a particular teacher optimization method. The result also suggests practical applicability: in deployment, user overrides will not follow any learned policy, and the framework's ability to learn from heterogeneous override sources is essential.

G. Scalability and Computational Analysis

We define four household profiles by varying physical parameters around the default apartment (H1): H2 doubles BESS capacity, H3 increases AC power and building volume, and H4 halves PV capacity and removes the WM. Each profile is trained independently with the Cost preference. Table VIII reports matched OR and computational costs for N parallel households.

All four configurations maintain low OR despite substantial differences in device portfolios and thermal properties. H2 benefits from doubled storage flexibility (OR reduced by 28%), while H3 and H4 face harder landscapes—larger thermal

TABLE VIII
MULTI-HOUSEHOLD SCALABILITY. (A) MATCHED OR (COST PREFERENCE) ACROSS HETEROGENEOUS HOUSEHOLD CONFIGURATIONS. (B) COMPUTATIONAL COST WHEN N HOUSEHOLDS TRAIN IN PARALLEL ON ONE RTX 4090 GPU.

(a) Preference adaptation across households

ID	Configuration	OR (↓)	UIC (↓)
H1	Default (San Diego apt.)	0.043	0.394
H2	2× BESS capacity	0.031	0.352
H3	1.5× AC power, larger volume	0.059	0.438
H4	0.5× PV capacity, no WM	0.068	0.471

(b) Computational scaling (N parallel households)

N	Train time (h)	Mem. (GB)	Infer. (ms/step)
1	2.4	1.1	0.9
3	7.3	2.2	0.9
5	12.1	3.3	1.0
10	24.2	6.1	1.0

mass and fewer controllable devices, respectively—yielding the highest OR at 0.068. These results confirm that the preference learning modules operate on state–action pairs without assuming a fixed device set. The three devices cover the major residential load categories—flexible storage (BESS), comfort-critical (AC), and deferrable (WM) [4]; adding devices such as EV chargers requires only extending state and action dimensions. Training time and memory scale linearly with N (10 households fit within a single RTX 4090); per-step inference (under 1.0 ms on this hardware) is several orders of magnitude below the 15-min control interval—a margin wide enough to absorb far slower deployment hardware and communication overhead, so computation is not the real-time bottleneck. In deployment, override events are logged continuously; retraining is triggered periodically or when accumulated overrides exceed a threshold, and runs offline on logged data without interrupting the user.

In multi-household deployments, aggregate scheduling must also respect distribution network constraints such as bus voltage limits and line thermal capacities [42]. Our framework is compatible with such extensions: constraints can be enforced via action-space projection (hard) or reward penalty terms (soft) [43]. The preference learning modules remain agnostic to network topology, so incorporating grid constraints modifies only the environment and reward interface. In a coordinated setting, each household transmits only $\mathbf{z}_t \in \mathbb{R}^{32}$ (128 bytes per step), letting a central coordinator dispatch curtailment signals that respect heterogeneous preferences; because the 15-min dispatch cadence is orders of magnitude longer than the second-scale communication round-trips and centralized solve times such coordination would entail, neither the payload nor latency is likely to be a bottleneck.

VI. CONCLUSION AND FUTURE WORK

This paper presents a contrastive preference learning framework for personalized home energy management that leverages user overrides as implicit pairwise feedback. Experiments on real-world residential energy data show that the framework achieves an average override rate of 0.033 across four preference profiles—a 37% reduction over the strongest baseline—with the lowest user interaction cost among all feedback-driven methods (UIC=0.394 vs. 0.51–2.33). Cross-preference evaluation and utility function inspection confirm that each model internalizes its target preference with interpretable preference-specific responses. The framework transfers efficiently to both prototype and novel preference targets, and adapts online to preference drift without explicit change-point detection. Multi-household experiments further confirm that the architecture generalizes across heterogeneous device configurations and scales linearly with the number of households.

Looking ahead, three limitations motivate future work. First, our evaluation relies on synthetic overrides; although the noise analysis (Table VII) shows graceful degradation under random flips, real occupants may act inconsistently or with systematic bias—a regime not fully captured by random noise—so a real-user study is essential, potentially aided by large language models as realistic user simulators and as a channel for natural-language preferences alongside overrides. Second, the framework currently controls each household independently; coordinating many households under distribution-network constraints (Section V-G), e.g., through privacy-preserving federated preference learning, is a promising direction. Third, our results rest on a single dataset and residential archetype under two climates; broader validation across building types, climates, and appliance categories would further confirm generality.

REFERENCES

[1] X. Zhong et al., "Global greenhouse gas emissions from residential and commercial building materials and mitigation strategies to 2060," *Nat. Commun.*, vol. 12, no. 1, p. 6126, Oct. 2021.

[2] A. Satchwell et al., "A national roadmap for grid-interactive efficient buildings," Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States), Technical Report, May 2021. [Online]. Available: https://www.osti.gov/biblio/1784302

[3] L. Yu et al., "Deep reinforcement learning for smart home energy management," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2751–2762, Apr. 2020.

[4] S. S. Shuvo and Y. Yilmaz, "Home energy recommendation system (hers): A deep reinforcement learning method based on residents' feedback and activity," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 2812–2821, Jul. 2022.

[5] Z. Nagy et al., "Ten questions concerning reinforcement learning for building energy management," *Building Environ.*, vol. 241, p. 110435, Aug. 2023.

[6] L. Chen, F. Meng, and Y. Zhang, "Fast human-in-the-loop control for hvac systems via meta-learning and model-based offline reinforcement learning," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 3, pp. 504–521, Jul. 2023.

[7] W. Ran, S. Nantogma, Q. Yang, S. Zhang, Y. Wang, and Y. Xu, "A semantic-based approach for self-configuring smart home systems using feedback-based reinforcement learning," *IEEE Internet Things J.*, vol. 12, no. 11, pp. 17 729–17 741, Jun. 2025.

[8] N. Mu, H. Hu, X. Hu, Y. Yang, B. Xu, and Q.-S. Jia, "CLARIFY: Contrastive preference reinforcement learning for untangling ambiguous queries," in *Proc. 42nd Int. Conf. Machine Learning (ICML)*, May 2025. [Online]. Available: https://openreview.net/forum?id=vOCPctm3nb

[9] S. M. Ali, J. C. Augusto, D. Windridge, and E. Ward, "A user-guided personalization methodology to facilitate new smart home occupancy," *Univ. Access Inf. Soc.*, vol. 22, no. 3, pp. 869–891, Aug. 2023.

[10] M. Ebrahimi, J. M. Fonseca, M. Shafie-Khah, G. J. Osório, and J. P. Catalão, "Machine-learning-based home energy management framework via residents' feedback," in *Proc. 2024 Int. Conf. Smart Energy Systems and Technologies (SEST)*. IEEE, 2024, pp. 1–6.

[11] Y. Li, Z. Yan, S. Chen, X. Xu, and C. Kang, "Operation strategy of smart thermostats that self-learn user preferences," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5770–5780, 2019.

[12] J. Y. Park, T. Dougherty, H. Fritz, and Z. Nagy, "Lightlearn: An adaptive and occupant centered controller for lighting based on reinforcement learning," *Building and Environment*, vol. 147, pp. 397–414, 2019.

[13] H. Elehwany, M. Ouf, B. Gunay, N. Cotrufo, and J.-S. Venne, "A reinforcement learning approach for thermostat setpoint preference learning," *Building Simulation*, vol. 17, no. 1, pp. 131–146, 2024.

[14] S. Chadoulos, I. Koutsopoulos, G. Polyzos, and N. Ipiotis, "Reinforcement learning with partially defined rewards and human feedback for energy efficiency recommendations," in *Proc. 16th ACM Int. Conf. Future and Sustainable Energy Systems (e-Energy)*, 2025, pp. 622–631.

[15] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proc. 31st Int. Conf. Neural Information Processing Systems (NeurIPS)*, I. Guyon et al., Eds., vol. 30. Curran Associates, Inc., 2017, pp. 4299–4307.

[16] J. Hejna et al., "Contrastive preference learning: Learning from human feedback without reinforcement learning," in *Proc. 12th Int. Conf. Learning Representations (ICLR)*, 2024.

[17] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, and X. Guan, "A review of deep reinforcement learning for smart building energy management," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 046–12 063, Aug. 2021.

[18] Z. Tan, M. Zeng, B. Liu, S. Feng, and B. Peng, "Integrating deep reinforcement learning into home energy management system based on soft actor-critic framework," in *Proc. 2022 Int. Conf. Electrical, Control and Information Technology (ECITech)*. VDE, 2022, pp. 1–6.

[19] H. Hou, X. Ge, Y. Chen, J. Tang, T. Hou, and R. Fang, "Model-free dynamic management strategy for low-carbon home energy based on deep reinforcement learning accommodating stochastic environments," *Energy Build.*, vol. 278, p. 112594, Jan. 2023.

[20] Y. Deng, Y. Zhang, F. Luo, and G. Ranzi, "Many-objective hems based on multi-scale occupant satisfaction modelling and second-life bess utilization," *IEEE Trans. Sustain. Energy*, vol. 13, no. 2, pp. 934–947, Apr. 2022.

[21] M. Sumayli and O. Moses Anubi, "Integration of multimode preference into home energy management system using deep reinforcement learning," *ASME J. Eng. Sustain. Build. Cities*, vol. 6, no. 4, p. 041007, Oct. 2025. [Online]. Available: https://doi.org/10.1115/1.4070085

[22] F. Luo, G. Ranzi, W. Kong, G. Liang, and Z. Y. Dong, "Personalized residential energy usage recommendation system based on load monitoring and collaborative filtering," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1253–1262, 2021.

[23] W. Chen, D. Liu, and J. Cao, "Personalized-preference-based electric vehicle charging recommendation considering photovoltaic consumption: A transfer reinforcement learning method," *IEEE Trans. Transp. Electrif.*, vol. 11, no. 1, pp. 4121–4132, 2025.

[24] Y. Liu, J. Ma, X. Xing, X. Liu, and W. Wang, "A home energy management system incorporating data-driven uncertainty-aware user preference," *Applied Energy*, vol. 326, p. 119911, 2022.

[25] K. Lee, L. Smith, and P. Abbeel, "PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pretraining," in *Proc. 38th Int. Conf. Machine Learning (ICML)*. PMLR, 2021, pp. 6152–6163.

[26] Y. Bai et al., "Constitutional ai: Harmlessness from ai feedback," *arXiv preprint arXiv:2212.08073*, Dec. 2022.

[27] Y. Yuan et al., "Uni-RLHF: Universal platform and benchmark suite for reinforcement learning with diverse human feedback," in *Proc. 12th Int. Conf. Learning Representations (ICLR)*, Jan. 2024. [Online]. Available: https://openreview.net/forum?id=WesY0H9ghM

[28] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.

[29] F. Luo, G. Ranzi, and Z. Y. Dong, *Building Energy Management Systems and Techniques: Principles, Methods, and Modelling*. Amsterdam, Netherlands: Elsevier, 2024.

[30] Pecan Street Inc., "Pecan Street Database," http://www.pecanstreet.org/, accessed: Jun. 2024.

[31] National Oceanic and Atmospheric Administration (NOAA), "NOAA Data," Online: https://www.ncei.noaa.gov/, accessed: Jun. 2024.

[32] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: The method of paired comparisons," *Biometrika*, vol. 39, no. 3-4, pp. 324–345, dec. 1952.

[33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Machine Learning (ICML)*, ser. Proc. Machine Learning Research, vol. 80. PMLR, Jul. 2018, pp. 1861–1870.

[34] F. De Angelis, M. Boaro, D. Fuselli, S. Squartini, F. Piazza, and Q. Wei, "Optimal home energy management under dynamic electrical and thermal constraints," *IEEE Trans. Ind. Inform.*, vol. 9, no. 3, pp. 1518–1527, Aug. 2013.

[35] F. Oldewurtel et al., "Use of model predictive control and weather forecasts for energy efficient building climate control," *Energy and Buildings*, vol. 45, pp. 15–27, 2012.

[36] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: https://www.gurobi.com

[37] X. Liang, F. de Nijs, B. Say, and H. Wang, "Human-in-the-loop AI for HVAC management enhancing comfort and energy efficiency," in *Proc. 16th ACM Int. Conf. Future and Sustainable Energy Systems (e-Energy)*. Rotterdam, Netherlands: ACM, Jun. 2025, pp. 359–370.

[38] CloudResearch Blog, "A simple formula for predicting the time to complete a study on Mechanical Turk," Blog post, 2020.

[39] S. K. Card, T. P. Moran, and A. Newell, "The keystroke-level model for user performance time with interactive systems," *Commun. ACM*, vol. 23, no. 7, pp. 396–410, Jul. 1980.

[40] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2052–2062.

[41] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. PMLR, 2011, pp. 627–635.

[42] M. Farivar and S. H. Low, "Branch flow model: Relaxations and convexification—part I," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2554–2564, Aug. 2013.

[43] H. Liu, W. Shi, and H. Zhu, "Distributed voltage control in distribution networks: Online and robust implementations," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6106–6117, Nov. 2018.

Xiao Du received the B.E. degree in renewable energy science and engineering from Chongqing University, Chongqing, China, in 2019, and the M.E. degree in energy and power engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2023. He is currently pursuing the Ph.D. degree in software engineering at Chongqing University, Chongqing, China. He is also currently a visiting Ph.D. student with Nanyang Technological University, Singapore. His research interests include deep reinforcement learning, smart home energy management, and energy informatics.

Fengji Luo (Senior Member, IEEE) received the bachelor's and master's degrees in software engineering from Chongqing University, Chongqing, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from The University of Newcastle, Newcastle, NSW, Australia, in 2014. He is currently a Senior Lecturer with the School of Civil Engineering, The University of Sydney, Sydney, NSW, Australia. He held a UUKI Rutherford International Researcher position with the Future Energy Institute, Brunel University London. His research interests include computational intelligence, energy demand side management, smart grid, and energy informatics. He has over 200 publications in these fields.

Jianheng Lan received the B.Eng. degree from the School of Artificial Intelligence and Data Science, Hebei University of Technology, Tianjin, China, in 2021, and the M.S. degree (Hons.) from the Department of Informatics, King's College London, London, U.K., in 2022. He is currently pursuing the Ph.D. degree in engineering with The University of Sydney, Sydney, NSW, Australia, associated with the Multimedia Laboratory, The Chinese University of Hong Kong, Hong Kong, China. His research interests are smart urban infrastructures and intelligent energy management.

Wei Zhou (Member, IEEE) received the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2015. He was a Researcher with the University of Auckland, Auckland, New Zealand, The Hong Kong Polytechnic University, Hong Kong, China, and the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He is currently a Professor with the School of Big Data and Software Engineering, Chongqing University. His research interests include recommendation systems, social computing, data mining, and energy informatics.

Junhao Wen (Member, IEEE) received the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2005. He is currently a Professor and the Dean of the School of Big Data and Software Engineering, Chongqing University. His research interests include recommendation systems, social computing, service computing, and smart energy systems.