Applied Soft Computing Journal 196 (2026) 115110

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Generalizable zero-shot home energy management via representation learning and behavioral cloning ☆

Xiao Du [a], Fengji Luo [b], Juntao Hu [a], Wei Zhou [a], Junhao Wen [a,*]

[a] School of Big Data & Software Engineering, Chongqing University, Shapingba, Chongqing, 401331, China
[b] School of Civil Engineering, University of Sydney, Camperdown, Sydney, NSW, 2050, Australia

HIGHLIGHTS

- Propose a contrastive learning framework to extract time-invariant physical dynamics from raw HEMS control trajectories.
- Introduce a clustering–frequency distribution method for interpretable household similarity measurement.
- Construct a zero-shot deployable policy via behavioral cloning, eliminating target–environment interaction.
- Achieve competitive performance with online DRL using only 2.4% of its interaction budget after minimal fine-tuning.

ARTICLE INFO

Keywords:
Home energy management
Representation learning
Zero-shot
Behavioral cloning
Deep reinforcement learning

ABSTRACT

Deep reinforcement learning (DRL) is widely used in home energy management for its ability to handle nonlinearity and uncertainty. However, its reliance on trial-and-error interaction makes early unsafe and inefficient behaviors impractical in real households. To address this, we propose RB-ZeroHEM, a zero-shot knowledge transfer framework based on representation learning and behavioral cloning. RB-ZeroHEM employs contrastive learning to extract stable, physics-aligned representations of household energy dynamics from historical control trajectories, enabling clustering-based similarity measurement without hand-crafted features. For a new household, it identifies the most similar source cases and clones a deployable policy requiring zero target-environment interactions. Experiments on 12 simulated households driven by real-world energy data (San Diego smart meter and weather records spanning four seasons) demonstrate that when transferring to physically similar environments, RB-ZeroHEM reduces discomfort ratio by 33% (from 12.34% to 8.24%) and electricity costs by 15% (from \$228 to \$193) compared to rule-based control, while achieving 81% of the grid energy savings of the best online DRL method, soft actor-critic (SAC), with zero interactions versus SAC's 24,192-step exploration budget. The learned representations exhibit strong physics alignment and remain robust across seasonal shifts and deployment noise. With minimal fine-tuning (576 interactions, 2.4% of SAC's budget), performance matches or surpasses online DRL baselines. When source–target physical dynamics are mismatched, zero-shot performance degrades gracefully, indicating that similarity-based retrieval reliability is critical. These results establish RB-ZeroHEM as a practical, low-risk solution for deploying intelligent energy management in real households, enabling rapid scaling without the safety and economic penalties of online exploration.

1. Introduction

1.1. Background and motivation

Buildings are among the largest consumers of energy worldwide and, consequently, major contributors to carbon dioxide emissions. For example, in 2020 the buildings sector accounted for 36% of global final energy use and 37% of energy-related CO_2 emissions [1]. With sustained growth in population, urbanization, and appliance ownership—particularly in developing economies—the share of energy consumed by residential buildings has risen year over year. According to

☆ This work was supported by the National Natural Science Foundation of China (Grant Nos. 62072060, 72074036) and the Australian Research Council (Grant No. DP220103881).
* Corresponding author.
Email addresses: duxiao@cqu.edu.cn (X. Du), fengji.luo@sydney.edu.au (F. Luo), hujuntao@stu.cqu.edu.cn (J. Hu), zhouwei@cqu.edu.cn (W. Zhou), jhwen@cqu.edu.cn (J. Wen).

https://doi.org/10.1016/j.asoc.2026.115110
Received 1 August 2025; Received in revised form 23 February 2026; Accepted 24 March 2026
Available online 25 March 2026

Symbol and Description

Learning Setup (Formulation & Method)

$t \in \{1, \dots, T\}$, T Time index, horizon length.

s_t, o_t, a_t; $\mathcal{S}, \mathcal{O}, \mathcal{A}$; $\mathcal{P}, \mathcal{R}, \gamma, \pi$ State, observation, action; spaces; transition, reward, discount, policy.

$\mathcal{U}, \mathcal{D}$ Stochastic uncertainties; time-invariant physical dynamics.

$\tau = \{(o_t, a_t)\}_{t=1}^{T_\tau}$; T_τ Trajectory slice and its length.

$\mathrm{sim}(\cdot,\cdot)$; $\Psi(\cdot,\cdot)$; $\mathrm{SIM}(\cdot,\cdot)$ Cosine similarity; set-induced similarity; household-level similarity.

$e, \mathcal{E}$; $\mathcal{N}_k(e^*)$ Household e; set of households; Top-k neighbors of target e^*.

$\mathrm{Clone}(\cdot)$; π_{clone} Cloning operator; deployed cloning policy.

Contrastive learning

$f(\cdot), g(\cdot)$; $\mathbf{z} \in \mathbb{R}^{d_z}$, $\mathbf{h} \in \mathbb{R}^{d_h}$ Encoder, projection head; representation and projection vectors.

x; $\mathcal{B}$; $\mathcal{B}^+$; N Sample; mini-batch samples; positive-pair set ($\subseteq \mathcal{B} \times \mathcal{B}$); batch half-size.

a, p, b; i, j Anchor/positive/other samples; household and slice indices.

κ; $\ell(a,p)$; $\mathcal{L}^{\mathrm{CL}}$; $\mathbb{1}[\cdot]$ Contrastive temperature; per-pair loss; contrastive loss; indicator.

Clustering & similarity

K_c, k_c; C_{k_c}; $\boldsymbol{\mu}_{k_c}$; $\mathcal{L}^{\mathrm{kmeans}}$ Clusters count and index; cluster sets; centroids; K-means objective.

$\mathbf{p}_i \in \mathbb{R}^{K_c}$ Cluster-frequency vector of e_i with $\sum_{k_c} (\mathbf{p}_i)_{k_c} = 1$.

Zero-shot behavioral cloning

$\mathcal{T}_{\mathrm{demo}}$; T_{demo} Aggregated demonstrations; total demo steps.

$\pi_\theta^{\mathrm{disc}}(o_t)$, $\pi_\theta^{\mathrm{cont}}(o_t)$; K_d, k_d Discrete/continuous heads; discrete class count and index.

$\mathcal{L}^{\mathrm{BC}}$; $\mathcal{L}^{\mathrm{disc}}$, $\mathcal{L}^{\mathrm{cont}}$; β BC loss; cross-entropy and ℓ_2 losses; head-weight.

Modeling (Household Transition Dynamics)

Global: Δt Simulation time step [h].

BESS: M_t^{b}; E_t, E_{rat}, soc_t; η^{b}, SD Mode; energy/capacity/SOC [kWh, kWh, −]; efficiency [−], self-discharge [h^{-1}].

$P_t^{\mathrm{b}} \in [0, P_{\mathrm{sup}}^{\mathrm{b}}]$; $P_{\mathrm{rat}}^{\mathrm{b}}$, $P_{\mathrm{sup}}^{\mathrm{b}}$ Battery power bound; rated/mode-constrained power [kW].

AC: M_t^{a}; $P_t^{\mathrm{a}} \in [P_{\mathrm{sta}}^{\mathrm{a}}, P_{\mathrm{rat}}^{\mathrm{a}}]$; $P_{\mathrm{sta}}^{\mathrm{a}}$ Mode; electrical power range; standby/min power [kW].

P_t^{heat}; cop_t, $\mathrm{cop}_{\mathrm{rat}}$; T_t^{in}, T_t^{out}; η^{a} Signed heat power [kW]; COP (inst./rated) [−]; indoor/outdoor temperature [°C]; efficiency factor [−].

WM: M_t^{w}, P_t^{w}, $P_{\mathrm{rat}}^{\mathrm{w}}$; z_t^{w}, $z_{\mathrm{task}}^{\mathrm{w}}$; A_t^{w} Mode; power/rated power [kW]; accumulated runtime/task length [h]; start signal $\{0,1\}$.

PV: P_t^{PV}; η^{PV}, $P_{\mathrm{rat}}^{\mathrm{PV}}$, A^{pan}; I_t, $I_{\min}$ PV output [kW]; efficiency [−], per-area rating [kW/m^2], panel area [m^2]; irradiance/threshold [kW/m^2].

Thermal: Q_t^{con}, Q_t^{ven}, Q_t^{solar} Step heat gains (conduction/ventilation/solar) [kJ per step].

c^{air}, ρ^{air}, V^{hou} Air specific heat [kJ/(kg·°C)], density [kg/m^3], indoor volume [m^3].

λ, A^{surf}, ACH, SHGC, A^{fen} Overall heat transfer coeff. [kW/(m^2·°C)], surface area [m^2], air-changes/hr [h^{-1}], solar gain coeff. [−], fenestration area [m^2].

Power balance: P_t^{D}, P_t^{G}, P_t^{BL} Net demand, grid exchange, base load [kW].

Uncertainties:

$\mathrm{occ}_{i,t}$, out_i; $t_i^{\{\mathrm{dep,arr}\}}$; μ, σ Occupancy and out-of-home flags $\{0,1\}$; departure/arrival times [h]; Gaussian parameters [h].

$p_{\{\mathrm{work,rest}\}}^{\mathrm{out}}$; $\mathrm{lau}_{i,t}$; t_i^{lau}, z_i^{lau}; $p_{\{\mathrm{dow}\}_1^7}^{\mathrm{lau}}$ Out-of-home Bernoulli probs [−]; laundry demand flags $\{0,1\}$; start time/tolerance [h]; day-of-week probs [−].

Exogenous: pri_t Market price [\$/kWh].

the International Energy Agency (IEA), household electricity accounted for 26% of all electricity use in buildings globally in 2019, and this share is projected to increase to 29% by 2040 [2].

Home energy management systems (HEMS) offer a promising solution. They can integrate distributed renewable generation and storage while enabling demand– response, thereby enhancing energy security and reducing emissions [3]. By coordinating controllable energy devices (CEDs)—such as air-conditioning (AC) systems and battery energy storage systems (BESS)—HEMS can optimize intra-home energy flows to balance cost and comfort [4].

However, as user preferences diversify, device portfolios expand, and renewable penetration increases, residential energy systems become increasingly nonlinear and uncertain [5]. Rule-based control tends to degrade over time, and accurate physics-based models are difficult to construct. These challenges motivate learning-based control methods—e.g., deep reinforcement learning (DRL)—which can adapt policies online through interaction with the environment without requiring an *a priori* model [6,7]. The incorporation of deep neural networks (DNNs) further equips these methods to handle complex, nonlinear dynamics.

Cold-start challenge.[1] Despite its promise, DRL inevitably undergoes an exploration phase at deployment: policies initialized randomly or near-randomly must rely on trial-and-error to estimate the long-term value of state–action pairs. Before acceptable performance is achieved, the agent may exhibit inefficient or even unsafe control behaviors, which is unacceptable in real homes [9]. What is needed is a policy that is immediately usable at deployment—ideally requiring little to no interaction in the target household—to reduce risk and operating costs while accelerating renewable integration and grid-responsive behavior.

1.2. Related work

1.2.1. How HEMS are built and applied

A HEMS is an intelligent control framework that autonomously executes *sense–decide–act* loops at the household scale. By coordinating CEDs with exogenous information (e.g., time-varying tariffs and demand-response events, weather and occupancy forecasts), a HEMS optimizes intra-household energy flows under operational and safety constraints [5]. Research on HEMS has expanded rapidly in recent years. According to the dominant objective and control interface, applications can be broadly categorized as *grid-oriented* and *user-oriented*.

Grid-oriented applications. These applications emphasize peak shaving and valley filling, facilitating grid-friendly integration of distributed renewables, and sustaining critical loads and resilience during contingencies (e.g., outages or capacity scarcity). Representative studies include joint consumption–device scheduling for demand-charge mitigation and peak reduction [10]; hierarchical or staged household energy management under residential photovoltaic (PV) penetration to smooth interconnection and alleviate reverse power flow [11]; and strategies that leverage electric vehicles (EVs) as mobile storage to enhance post-disaster household resilience [12].

User-oriented applications. These applications focus on occupant experience and bill economics—thermal comfort and lighting quality,

[1] Cold start. The initial stage in which a learning system lacks historical data or interaction traces in the target environment, making reliable inference or personalization difficult; this concept has been extensively discussed in the recommender-systems literature [8].

reduction of electricity bills and demand charges, equipment lifetime, and everyday automation. Examples include bill minimization that balances energy savings and comfort [13]; mixed-integer linear programming (MILP) formulations that explicitly price battery cycle aging to more faithfully reflect lifetime economics [14]; and PV–storage schemes that incorporate cash-flow forecasting and dynamic programming to strengthen self-consumption and economic robustness [15]. For comfort-prioritized heating, ventilation, and air-conditioning (HVAC) coordination, hierarchical model predictive control (MPC) frameworks demonstrate stable constraint handling and interpretability in HVAC settings [16,17]; meanwhile, in tasks such as increasing self-consumption and navigating the cost–comfort trade-off, self-scheduling of PV + BESS yields reproducible gains [18].

Against this backdrop, existing methods can be categorized along three dimensions: model dependence, uncertainty handling, and interaction cost. At one end of the spectrum are model-based approaches, such as the direct optimization method (DOM) [19] and MPC, which rely on explicit environment models. At the other end are methods like DRL, which reduce model dependence by learning primarily through interaction.

DOM and MPC. Both DOM and MPC depend on explicit environment models (e.g., building thermal RC models, device physics and constraints, and forecasts of price, weather, and occupancy). DOM solves a one-shot (possibly nonlinear, mixed-integer, multiobjective, or robust) optimization over the entire horizon given complete models and constraints. Because practical HEMS problems are typically indefinite or effectively infinite-horizon, MPC serves as a pragmatic realization of DOM by receding-horizon decomposition: at each step it optimizes a control sequence over a finite window, executes only the first action, and updates with new observations. Their advantages include explicit constraint enforcement, the ability to inject engineering priors, and good interpretability (e.g., cloud-based, real-time MPC for multi-carrier, multiobjective HEMS that demonstrates end-to-end deployment and graceful degradation [3]; hierarchical MPC achieving favorable comfort–cost trade-offs in building microgrids [17]). Limitations stem from household heterogeneity and time-varying disturbances, which make accurate modeling difficult; uncertainty is often handled via scenario trees, robust or distributionally robust optimization, or dynamic-programming variants, increasing engineering and computational burden [20–22]. Consequently, many studies adopt data-driven submodels—such as DNN-based load/price/occupancy forecasting—to improve look-ahead quality and thereby support rolling optimization in MPC/DOM or two-stage decision making [13–15,18,23]. However, when the target household lacks in-domain data, these submodels are vulnerable to distribution shift, creating a cold-start risk. Representative DOM/MPC-based HEMS include price-responsive HVAC + BESS coordination with explicit lifetime modeling (MILP) [14]; self-consumption enhancement via self-scheduling [18]; comfort maintenance with multi-timescale trade-offs (including thermal discomfort and preferences) [13,16,17]; cloud-based real-time MPC for multi-carrier, multiobjective control [3]; and robust/dynamic-programming frameworks for price and renewable uncertainty [20–22].

DRL. DRL reduces reliance on identifiable model priors by learning control policies directly through interaction. A predefined reward function scores interactions by internalizing multiple objectives—electricity bills, comfort penalties (e.g., relative discomfort cost), device degradation, and even distribution-network impacts—and DNNs approximate value or policy functions that are iteratively improved. With increasing representational capacity, DNN-based agents exhibit natural elasticity in the face of nonlinearity and uncertainty: policies can learn disturbance responses directly, rather than requiring explicit uncertainty sets or scenario trees as in MPC/DOM [20,24–26]. Typical studies include single-agent deep Q-network (DQN) [27] for demand–response with reward designs that jointly account for occupant discomfort and transformer aging [24]; privacy-preserving load control that learns policies without sharing fine-grained data [25]; end-to-end IoT-based smart home energy management [26]; multi-agent DRL for coordinated household/device responses [28]; and EV charging scheduling in residential settings [20]. A persistent practical challenge is that nearly all DRL-based HEMS still depend on online interaction and adaptation in the deployment environment [20,24–26,28].

The resulting risks and costs during the cold-start phase—e.g., comfort and bill penalties from exploratory actions, equipment wear, and potential adverse grid impacts—remain key obstacles to real-world adoption. The next section focuses on cold-start mitigation.

Table 1
Comparison of representative works addressing the cold-start problem in HEMS.

Reference	Target Env. interaction	Policy architecture alignment	Source-target Env. similarity
[29–34]	few-shot	Required	The more similar, the better
[35–38]	few-shot	Unified Policy	Same observation and action space
[39]	zero-shot/few-shot	Required	The more similar, the better
[40]	zero-shot	Flexible	Identical environment

* *few-shot*: limited target–environment interaction; *zero-shot*: no interaction required [41].

1.2.2. Techniques for the cold-start problem

Alleviating the cold-start problem of DRL-based HEMS fundamentally hinges on obtaining a policy highly aligned with the target environment prior to deployment. Current studies, which remain in their early stages, mainly focus on three technical directions: transfer learning (TL), meta reinforcement learning (meta-RL), and offline RL. Table 1 compares representative works along three risk-sensitive dimensions in cold-start: (i) dependency on target environment interaction during adaptation, (ii) alignment requirement between source and target policy architectures, and (iii) source–target environment similarity assumptions.

TL. TL accelerates learning in related but distinct target environments by leveraging knowledge from source environments [42]. Common approaches involve pre-training DNNs and transferring the parameters of early layers (either frozen or partially fine-tuned) [43]. In the HEMS field, Ref. [29] pre-trained a DQN in the source building and fine-tuned it in the target building, achieving accelerated convergence while maintaining energy consumption and comfort within acceptable ranges. Beyond single-agent HVAC control, cross-home and multi-zone transfer have also received attention: [30] explored transferable DRL under heterogeneous appliances and user preferences, while [31,44] proposed frameworks to reduce per-household data requirements. Refs. [32,33] addressed state/reward definition discrepancies via model slicing, imitation learning, and fine-tuning, emphasizing online adaptability. Synergizing with TL, federated learning enhances privacy and data security, demonstrating unique advantages in sparse and non-shareable data scenarios, which is particularly valuable for overcoming data scarcity issues [45]. Ref. [34] aggregated experiences from multiple buildings via multi-source transfer, effectively shortening training duration and reducing temperature violations. Transfer paradigms have even been extended beyond HEMS to heterogeneous physical systems such as hybrid vehicles [46], further demonstrating the potential of transfer mechanisms in handling environmental heterogeneity.

While TL significantly reduces sample requirements, it generally still depends on fine-tuning in the target environment, and demands high consistency in policy architecture and representation space between the source and target environments [42]. When there is significant heterogeneity in environmental dynamics or reward structure, performance tends to degrade sharply [32]. In addition, how to systematically obtain high-quality, transferable source environment data in practical deployment scenarios remains an open question.

Meta-RL. Meta-RL aims to acquire policy initializations or adaptation rules that can quickly adapt to new tasks [47]. In HEMS, it is used to handle seasonal variations, changes in user preferences, and device heterogeneity. Typical methods include outer-loop generalization across environments and inner-loop fine-tuning for the target environment. For example, Ref. [35] utilized an unsupervised autoencoder to detect context shifts in multi-objective HEMS scheduling, enabling few-shot adaptation. Ref. [36] designed an LSTM-nested structure to enhance robustness to seasonal changes. MetaEMS [37] realized distribution alignment among building groups and achieved both group- and household-level adaptation based on the CityLearn platform. Related multi-agent meta-learning approaches also demonstrated stronger cooperative adaptation under distributional shifts [38].

Meta-RL demonstrates theoretical and practical advantages in few-shot adaptation; however, it still requires a certain amount of target environment interaction and incurs substantial computational overhead due to its nested optimization structure, limiting its deployment in resource-constrained real-world scenarios. Moreover, most meta-learning frameworks strongly depend on the consistency of observation and action spaces between the source and target environments, limiting their generalizability across heterogeneous settings.

Offline RL. Offline RL directly learns policies from static datasets, thus avoiding high-risk online exploration during deployment [48]. MERLIN [39] combines multi-agent transfer and offline learning to effectively reuse policies in community-level HEMS tasks, reducing training costs. Recent works have also made preliminary progress in heterogeneous EV charging and robust offline policy learning [40].

In theory, offline RL is the most suitable for "zero-interaction" cold-start, but in practice, it heavily relies on high-quality, high-coverage historical datasets and is extremely sensitive to state-action distribution and behavior policy mismatches. Most existing work still assumes that the target environment is highly homogeneous with the data source, which is clearly inconsistent with the reality of highly heterogeneous, data-scarce HEMS scenarios.

As shown in Table 1, current TL, meta-RL, and offline RL approaches for mitigating HEMS cold-start face three common challenges. First, most methods still require a certain amount of target environment interaction, making truly zero-shot deployment difficult. Second, they require a high degree of policy architecture or observation/action space alignment, which is rarely met in real-world heterogeneous environments and device scenarios. Third, most methods assume high source–target environment similarity, whereas the diversity of environmental dynamics and user behaviors in practice often undermines transfer and generalization performance. Therefore, how to enhance cold-start capability while relaxing these assumptions remains a largely unexplored problem. To address these challenges, this paper attempts to utilize contrastive learning, clustering, and behavioral cloning, leveraging easily accessible control sequence data, to explore more flexible and universal cold-start strategies.

1.3. Scope and contributions

To overcome the reliance on target–environment interaction, policy architecture alignment, and environmental similarity assumptions in current cold-start solutions, this paper proposes the **RB-ZeroHEM** framework, whose core innovations lie in similarity discovery and zero-shot policy transfer.

First, RB-ZeroHEM explores a novel path for similarity discovery. Instead of relying on static features (e.g., house size, wall materials) [49] or manually defined similarity metrics [32], we employ contrastive learning to automatically extract representation vectors from raw HEMS control trajectories (CTs) that reflect the inherent and static physical dynamics of buildings, thus enabling dynamic similarity measurement and household clustering in the representation space.

Fig. 1 illustrates our motivation for adopting contrastive learning. We observe that although randomly changing factors (uncertainties) such as

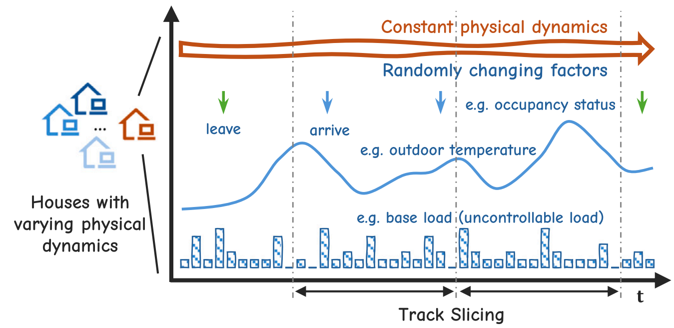


Fig. 1. Motivation for contrastive learning in RB-ZeroHEM.

weather, base load, and user demand fluctuate over time, the physical dynamics of buildings and their energy devices remain relatively invariant. While DRL-based HEMS policies can respond in real-time to these uncertainties (e.g., scheduling batteries based on PV output), their control effectiveness is primarily governed by underlying physical constraints (such as battery charge/discharge patterns). By segmenting a single household's control trajectory into slices, each slice provides a different perspective on the same invariant physical dynamics, while environmental stochasticity manifests as noise—this closely aligns with the philosophy of contrastive learning, which aggregates similar samples and separates dissimilar ones. Our goal, therefore, is to learn representations such that slices from the same household are clustered together, while those from different households are effectively distinguished, thereby capturing discriminative physical dynamics.

Building upon this, RB-ZeroHEM further utilizes the CTs of dynamically similar households and applies behavioral cloning to directly construct deployable HEMS control policies for the target household, achieving truly zero-shot cold-start without any interaction or architectural alignment with the target environment. Our main contributions are summarized as follows:

1. We design a contrastive learning framework to disentangle time-invariant physical dynamics from stochastic uncertainties. Compared with existing methods, this extracts physically aligned embeddings directly from raw trajectories, circumventing the reliance on strict policy architecture alignment or hand-crafted features.
2. We introduce a clustering–frequency distribution matching method to transform trajectory embeddings into interpretable behavioral distributions. This provides a data-driven approach for matching heterogeneous households, relaxing the strict identical-environment assumptions commonly required by previous studies.
3. We construct a deployable policy via behavioral cloning from similar source households. Unlike few-shot adaptation methods demanding risky online exploration, our approach achieves true zero-interaction initialization while remaining competitive with online DRL baselines.

2. Preliminaries

2.1. Problem formulation

A DRL-based HEMS typically operates as a real-time control system over a discrete time horizon $t \in \{1, 2, \dots, T\}$, and is often formalized as a partially observable Markov decision process (POMDP) [50] $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, r, \gamma \rangle$. At each time step t, the agent receives an observation $o_t \in \mathcal{O}$, selects an action $a_t \in \mathcal{A}$ according to the policy $\pi(a_t|o_t)$, the environment transitions to a new state $s_{t+1} \in \mathcal{S}$ via the transition kernel $\mathcal{P}$, and the agent obtains a reward $r_t = \mathcal{R}(s_t, a_t)$. In real household environments, the agent can only access the partial observation $\mathcal{O}$, and some variables in the underlying state $\mathcal{S}$ are unobservable.

In this context, we assume that $\mathcal{P}$ consists of two components: (i) stochastic, time-varying uncertainties $\mathcal{U}$ (e.g., occupancy, weather disturbances), and (ii) time-invariant physical dynamics $\mathcal{D}$ (e.g., AC, BESS, and building thermal properties). Learning the policy $\pi(a_t|o_t)$ essentially amounts to finding the optimal mapping for a given $\mathcal{D}$, while disturbances from $\mathcal{U}$ can generally be accommodated via real-time closed-loop feedback (e.g., through online adaptation or robust policy design) [7,36,51]. However, existing studies and practical experience consistently show that when $\mathcal{D}$ changes—i.e., when transferring a policy to an environment with different physical dynamics—the performance of π may degrade significantly, even if the distribution of $\mathcal{U}$ is similar [39,43].

Therefore, identifying environments with similar physical dynamics $\mathcal{D}$ is crucial for effective policy transfer. Due to the complexity of residential settings, it is often infeasible to manually define structured metrics or explicit parameters to measure $\mathcal{D}$, making environment similarity challenging to quantify. To address this, the first goal of this work is to learn a representation that is identifiable for $\mathcal{D}$ and robust to $\mathcal{U}$ from easily accessible control trajectory segments. The representation learning task is defined as:

$$\mathbf{z} = f(\tau), \qquad \tau = \{(o_t, a_t)\}_{t=1}^{T_\tau}, \quad \mathbf{z} \in \mathbb{R}^{d_z}, \tag{1}$$

where f is an encoder and T_τ is the segment length. By slicing the historical data of household e with a fixed window, we obtain the set representation:

$$\mathcal{Z}_e = \left\{ \mathbf{z} \ : \ \mathbf{z} = f(\tau),\ \tau \in \mathcal{T}_e \right\}. \tag{2}$$

Next, we define a set-level similarity in the learned representation space $\mathbb{R}^{d_z}$ to quantify the physical dynamics similarity between different environments:

$$\mathrm{SIM}(e_i, e_j) = \Psi(\mathcal{Z}_{e_i}, \mathcal{Z}_{e_j}), \tag{3}$$

where Ψ is induced by a sample-wise distance and an aggregation strategy. Based on the similarity, the k nearest neighbors for the target household e^* are selected as

$$\mathcal{N}_k(e^*) = \text{Top-}k \underset{e \in \mathcal{E} \setminus \{e^*\}}{\text{sort}} \mathrm{SIM}(e, e^*). \tag{4}$$

Finally, after obtaining the similarity criterion between the target and source environments, the ultimate objective of this work is to construct a deployable policy for the target environment in a zero-shot manner by leveraging high-quality CTs from similar source environments. By aggregating the demonstrations $\mathcal{T}_{\text{demo}} = \cup_{e \in \mathcal{N}_k(e^*)} \mathcal{T}_e$, we employ an abstract policy cloning operator to produce the deployable policy:

$$\pi_{\text{clone}} = \mathrm{Clone}(\mathcal{T}_{\text{demo}};\ \mathcal{O}, \mathcal{A}). \tag{5}$$

2.2. Modeling household transition dynamics

We construct simulation environments that reflect modern households with high renewable penetration (Fig. 2). The HEMS manages three representative CEDs—a BESS, an AC, and a washing machine (WM)—covering the priority, flexible, and deferrable load categories commonly analyzed in home energy management [52]. Other appliances (e.g., television, microwave) are aggregated as uncontrollable base load. Following the decomposition introduced in Section 2.1, we organize the modeling into time-invariant physical dynamics $\mathcal{D}$ and stochastic, time-varying uncertainties $\mathcal{U}$.

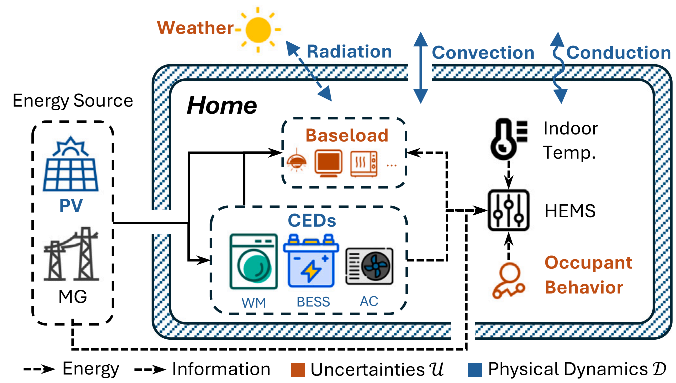


Fig. 2. Architecture of the household environment with rooftop PV, grid connection, and controllable devices.

2.2.1. Time-invariant physical dynamics

Energy device models. The energy devices follow physics-based models with operational constraints and energy conversion dynamics [19].

- **BESS.** Standard dynamics with efficiency and self-discharge:

$$\mathrm{soc}_{t+1} = \frac{E_{t+1}}{E_{\text{rat}}}, \tag{6a}$$

$$E_{t+1} = \overline{E}_{t+1} - \mathrm{SD} \cdot \overline{E}_{t+1} \Delta t, \tag{6b}$$

$$\overline{E}_{t+1} = \begin{cases} E_t + \eta^{\text{b}} P_t^{\text{b}} \Delta t, & M_t^{\text{b}} = 1 \text{ (charge)}, \\ E_t - \dfrac{P_t^{\text{b}} \Delta t}{\eta^{\text{b}}}, & M_t^{\text{b}} = -1 \text{ (discharge)}, \\ E_t, & M_t^{\text{b}} = 0 \text{ (standby)}, \end{cases} \tag{6c}$$

$$P_t^{\text{b}} \in [0, P_{\text{sup}}^{\text{b}}], \tag{6d}$$

$$P_{\text{sup}}^{\text{b}} = \begin{cases} \min\left\{ P_{\text{rat}}^{\text{b}}, \dfrac{E_{\text{rat}} - E_t}{\eta^{\text{b}} \Delta t} \right\}, & M_t^{\text{b}} = 1, \\ \min\left\{ P_{\text{rat}}^{\text{b}}, \dfrac{\eta^{\text{b}} E_t}{\Delta t} \right\}, & M_t^{\text{b}} = -1, \\ 0, & M_t^{\text{b}} = 0. \end{cases} \tag{6e}$$

Here $M_t^{\text{b}} \in \{1, -1, 0\}$ is the operating mode, E_t the stored energy, E_{rat} the rated capacity, SD the self-discharge rate, and η^{b} the efficiency.

- **AC.** Thermodynamic model with temperature-dependent COP:

$$P_t^{\text{heat}} = \mathrm{cop}_t\, P_t^{\text{a}} M_t^{\text{a}}, \qquad P_t^{\text{a}} \in [P_{\text{sta}}^{\text{a}}, P_{\text{rat}}^{\text{a}}], \tag{7a}$$

$$\mathrm{cop}_t = \begin{cases} \overline{\mathrm{cop}}_t, & 0 < \overline{\mathrm{cop}}_t < \mathrm{cop}_{\text{rat}}, \\ \mathrm{cop}_{\text{rat}}, & \text{otherwise} \end{cases}, \tag{7b}$$

$$\overline{\mathrm{cop}}_t = \begin{cases} \dfrac{\eta^{\text{a}} (T_t^{\text{in}})_{\text{K}}}{T_t^{\text{in}} - T_t^{\text{out}}}, & M_t^{\text{a}} = 1 \text{ (heating)}, \\ \dfrac{\eta^{\text{a}} (T_t^{\text{in}})_{\text{K}}}{T_t^{\text{out}} - T_t^{\text{in}}}, & M_t^{\text{a}} = -1 \text{ (cooling)}, \\ 0, & M_t^{\text{a}} = 0 \text{ (standby)}. \end{cases} \tag{7c}$$

$M_t^{\text{a}} \in \{1, -1, 0\}$ denotes heating/cooling/standby; $(\cdot)_{\text{K}}$ converts to Kelvin; P_t^{heat} is signed heat power (cooling gives negative).

- **WM.** Non-interruptible once started:

$$P_t^{\text{w}} = M_t^{\text{w}} P_{\text{rat}}^{\text{w}}, \tag{8a}$$

$$z_t^{\text{w}} = \begin{cases} \min\{z_{t-1}^{\text{w}} + \Delta t,\ z_{\text{task}}^{\text{w}}\}, & M_t^{\text{w}} = 1 \text{ (operating)}, \\ 0, & M_t^{\text{w}} = 0, \text{ (standby)} \end{cases} \tag{8b}$$

$$M_t^{\mathrm{w}} = \begin{cases} 1, & (0 < z_{t-1}^{\mathrm{w}} < z_{\mathrm{task}}^{\mathrm{w}}) \vee A_t^{\mathrm{w}} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{8c}$$

z_t^{w} is the accumulated runtime; A_t^{w} is the start signal (only when idle).

- **PV.** Irradiance–area–efficiency model:

$$P_t^{\mathrm{PV}} = \begin{cases} 0, & I_t < I_{\min}, \\ \min\left(\eta^{\mathrm{PV}} I_t,\ P_{\mathrm{rat}}^{\mathrm{PV}}\right) A^{\mathrm{pan}}, & \text{otherwise,} \end{cases} \tag{9}$$

where I_t is irradiance, A^{pan} panel area, and $P_{\mathrm{rat}}^{\mathrm{PV}}$ the rated output per unit area.

Building thermal model. Indoor temperature follows the heat balance among conduction, ventilation, solar gains, and HVAC operation:

$$T_{t+1}^{\mathrm{in}} = T_t^{\mathrm{in}} + \frac{Q_t^{\mathrm{con}} + Q_t^{\mathrm{ven}} + Q_t^{\mathrm{solar}} + P_t^{\mathrm{heat}} \Delta t}{c^{\mathrm{air}} \rho^{\mathrm{air}} V^{\mathrm{hou}}}, \tag{10a}$$

$$Q_t^{\mathrm{con}} = \lambda A^{\mathrm{surf}} (T_t^{\mathrm{out}} - T_t^{\mathrm{in}}) \Delta t, \tag{10b}$$

$$Q_t^{\mathrm{ven}} = \mathrm{ACH}\, c^{\mathrm{air}} \rho^{\mathrm{air}} V^{\mathrm{hou}} (T_t^{\mathrm{out}} - T_t^{\mathrm{in}}) \Delta t, \tag{10c}$$

$$Q_t^{\mathrm{solar}} = \mathrm{SHGC}\, I_t A^{\mathrm{fen}} \Delta t. \tag{10d}$$

Power balancing constraint. The household's total demand P_t^{D} must equal the supply from local PV, BESS, and grid exchange P_t^{G}:

$$P_t^{\mathrm{G}} = \begin{cases} P_t^{\mathrm{D}}, & P_t^{\mathrm{D}} \geq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{11a}$$

$$P_t^{\mathrm{D}} = P_t^{\mathrm{a}} + M_t^{\mathrm{b}} P_t^{\mathrm{b}} + P_t^{\mathrm{w}} + P_t^{\mathrm{BL}} - P_t^{\mathrm{PV}}. \tag{11b}$$

2.2.2. Time-varying uncertainties

Stochastic occupancy and demand. The modeling of occupancy and laundry-task generation builds on stochastic, bottom-up domestic demand modeling literature [53–55] and adapts the occupancy/task abstraction to the HEMS simulation context.

- **Occupancy.** Workday/non-workday have distinct departure/arrival distributions:

$$\mathrm{occ}_{i,t} = 1 - \mathbb{1}(t_i^{\mathrm{dep}} \leq t < t_i^{\mathrm{arr}}), \tag{12a}$$

$$t_i^{\{\mathrm{dep,arr}\}} \sim \mathcal{N}\left(\mu_{\{\mathrm{work,rest}\}}^{\{\mathrm{dep,arr}\}}, (\sigma_{\{\mathrm{work,rest}\}}^{\{\mathrm{dep,arr}\}})^2\right) \text{ if } \mathrm{out}_i = 1, \tag{12b}$$

$$\mathrm{out}_i \sim \mathrm{Bernoulli}(p_{\{\mathrm{work,rest}\}}^{\mathrm{out}}). \tag{12c}$$

- **Laundry Demand.** Laundry requests are generated by day-of-week probabilities, with start times and waiting tolerances sampled as:

$$\mathrm{lau}_{i,t} = \begin{cases} \mathbb{1}(t_i^{\mathrm{lau}} \leq t < t_i^{\mathrm{lau}} + z_i^{\mathrm{lau}}), & \mathrm{lau}_i^{\mathrm{day}} = 1, \\ 0, & \text{otherwise,} \end{cases} \tag{13a}$$

$$t_i^{\mathrm{lau}} \sim \mathrm{Uniform}\left([t_{\min}^{\mathrm{lau}}, t_{\max}^{\mathrm{lau}}] \mid \mathrm{occ}_{i,t} = 1\right), \tag{13b}$$

$$z_i^{\mathrm{lau}} \sim \max\left(0,\ \mathcal{N}(\mu^{\mathrm{wait}}, (\sigma^{\mathrm{wait}})^2)\right), \tag{13c}$$

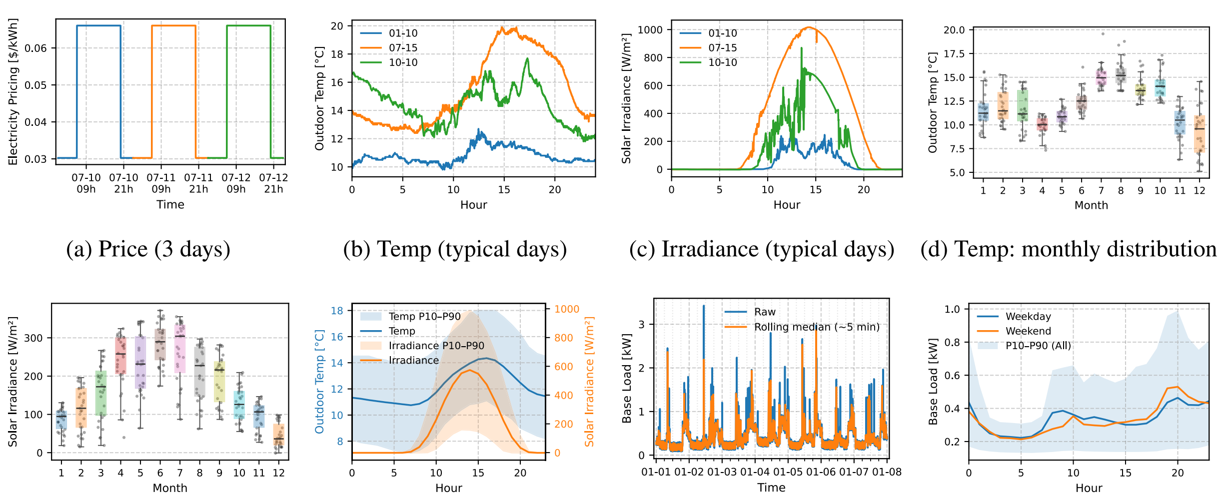
$$\mathrm{lau}_i^{\mathrm{day}} \sim \mathrm{Bernoulli}(p_{\{\mathrm{dow}\}_1^7}^{\mathrm{lau}}). \tag{13d}$$

Exogenous variables from data. Exogenous series—irradiance I_t, outdoor temperature T_t^{out}, and base load P_t^{BL}—are taken from real-world datasets [56,57]. Market prices pri_t come from historical records of the local market. Fig. 3 summarizes these variables. Prices show clear time-of-use patterns. Both T_t^{out} and I_t exhibit strong diurnal and seasonal variability; monthly distributions and the P10–P90 bands quantify their spread. For households in the same location and time window, the weather series $(T_t^{\mathrm{out}}, I_t)$ are shared, whereas the base load P_t^{BL} is household-specific: magnitudes differ and daily shapes can vary (e.g., distinct always-on appliances and occupancy routines). To avoid clutter, panels (g)–(h) plot one representative household.

To simulate different household transition dynamics $\mathcal{P}$, we assign varying parameters to the above models and utilize real-world data from multiple households. All parameter configurations align with realistic settings.

3. Methodology

As illustrated in Fig. 4, RB-ZeroHEM comprises three stages: (i) contrastive representation learning from historical CTs, (ii) similarity



(a) Price (3 days) (b) Temp (typical days) (c) Irradiance (typical days) (d) Temp: monthly distribution

(e) Irradiance: monthly distribution (f) Diurnal temp & irradiance (mean ± P10–P90) (g) Base load: one-week trace (h) Base load: weekday vs weekend (P10–P90)

Fig. 3. Visualization of exogenous variables. (a) time-of-use price over several consecutive days. (b)–(c) temperature and irradiance on representative winter/summer/autumn days. (d)–(e) monthly distributions of temperature and irradiance. (f) average diurnal profiles with P10–P90 bands. (g) one household's base load P_t^{BL} (one-week trace with 5-min rolling median). (h) Weekday vs weekend diurnal base load with P10–P90 band. Weather series $(T_t^{\mathrm{out}}, I_t)$ are identical across households in the same area and period, whereas P_t^{BL} is household-specific in both magnitude and shape.

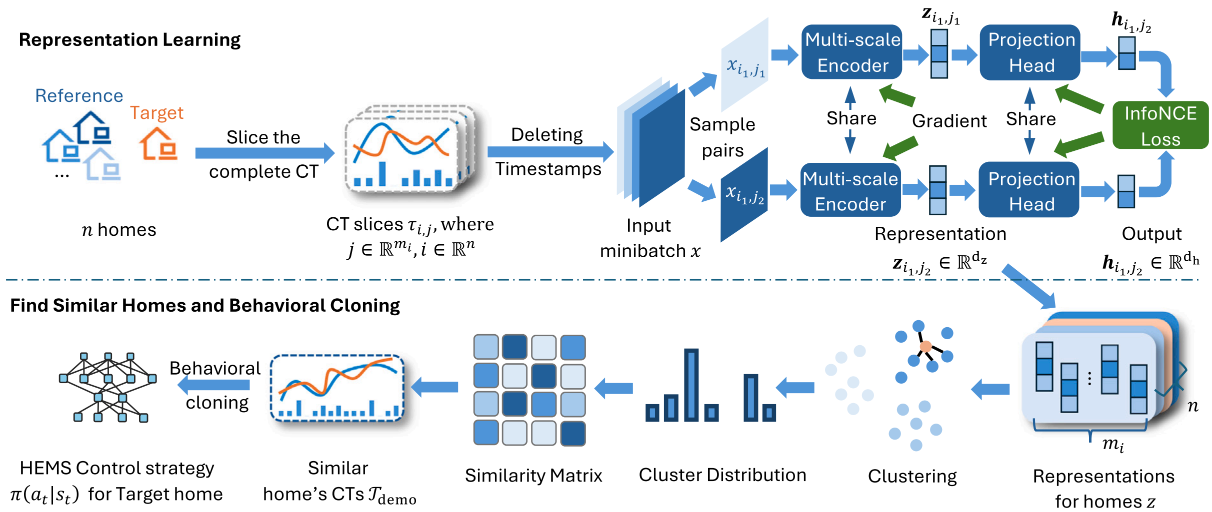


Fig. 4. RB-ZeroHEM: contrastive representation of CT slices, clustering and similarity in the representation space, and zero-shot BC from similar households.

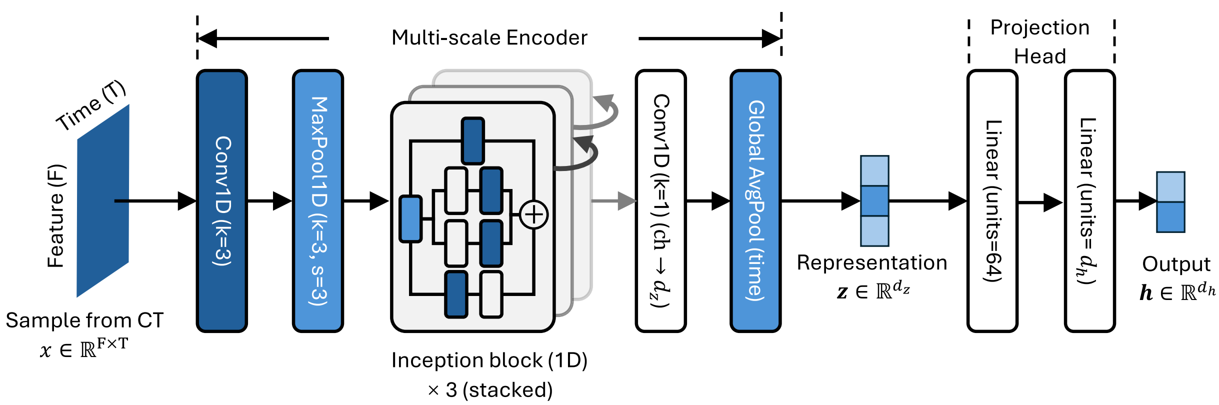


Fig. 5. Inception-style multi-scale encoder $f(\cdot)$ and projection head $g(\cdot)$ used in Stage 1.

evaluation via clustering over the learned representation space, and (iii) zero-shot policy generation via behavioral cloning (BC) using pseudo-expert demonstrations from similar households. We explain the design rationale within each stage, and defer the implementation details and hyperparameters to Section 4.1.

3.1. Stage 1: contrastive representation learning

Contrastive learning [58,59] is a representation-learning paradigm that trains an encoder so that *positives* (semantically similar samples) are mapped close together while *negatives* (dissimilar samples) are pushed apart. Concretely, given an *anchor* in a mini-batch, the loss increases the (temperature-scaled) similarity between the anchor and its positive counterpart while decreasing similarities to the remaining samples, typically using InfoNCE [60]. Compared with traditional clustering using hand-crafted distances, contrastive learning leverages trainable DNNs to shape the representation space directly from data, which is effective for high-dimensional sequences.

Our goal is to extract household-invariant yet behavior-revealing representations from CTs that are robust to spatiotemporal context shifts (Eq. 1). Using the household index as a self-supervised signal encourages the encoder to capture stable physical/behavioral dynamics rather than shortcut features (e.g., timestamps). Removing absolute time fields prevents the model from solving the task using trivial calendar cues and promotes invariance across climates and seasons.

Let there be n households. For each household e_i $(i \in \{1,\dots,n\})$, CTs are sliced into fixed-length T_τ segments $\tau_{i,j}$, yielding samples $x_{i,j}$, $j \in \{1,\dots,m_i\}$. Two parameter-sharing networks are used. As shown in Fig. 5, the encoder $f(\cdot)$ is a *multi-scale* 1D CNN, inspired by the Inception modules of GoogLeNet [61] and adapted to 1D temporal convolutions. A shallow stem (Conv1D with kernel size 3 followed by MaxPool1D with kernel size 3 and stride 3) feeds three stacked 1D Inception blocks; each block aggregates *parallel* Conv1D branches with kernel sizes $\{3,5,7\}$ (all conv layers use BN + ReLU), whose outputs are concatenated and fused by a pointwise Conv1D (kernel size 1; channels ch $\to d_z$). Temporal global average pooling then produces the representation $\mathbf{z} \in \mathbb{R}^{d_z}$. The projection head $g(\cdot)$ is a two-layer multilayer perceptron (MLP) that maps $\mathbf{z}$ to $\mathbf{h}$ (Linear $[d_z \to 64]$ followed by Linear $[64 \to d_h]$) and is used only for the contrastive loss.

Let $\mathcal{B} = \{x_1,\dots,x_{2N}\}$ denote a mini-batch of $2N$ trajectory slices. A *positive* pair consists of two different slices from the *same* household; all other pairs serve as negatives. We ℓ_2-normalize projection outputs prior to similarity computation, i.e., $\tilde{\mathbf{h}} = \mathbf{h}/\|\mathbf{h}\|$, so the cosine similarity reduces to a dot product on the unit sphere:

$$\mathrm{sim}(u,v) = \frac{u^\top v}{\|u\|\,\|v\|}, \tag{14}$$

where u and v are vectors of the same dimension. Let $\mathcal{B}^+ \subseteq \mathcal{B}\times\mathcal{B}$ collect all positive pairs. For each positive pair $(a,p) \in \mathcal{B}^+$, where a is the anchor sample and p is its positive counterpart (i.e., a different slice from the same household as a), we define the per-anchor InfoNCE loss

$$\ell(a,p) = -\log \frac{\exp\big(\mathrm{sim}(\tilde{\mathbf{h}}_a,\tilde{\mathbf{h}}_p)/\kappa\big)}{\sum_{b\in\mathcal{B}\setminus\{a,p\}} \exp\big(\mathrm{sim}(\tilde{\mathbf{h}}_a,\tilde{\mathbf{h}}_b)/\kappa\big)}, \tag{15}$$

where the denominator sums over all $2N-2$ samples in $\mathcal{B}$ excluding only a and p, so all remaining samples—including other slices from the same household—serve as negatives. Temperature $\kappa > 0$ controls contrast hardness: smaller κ enforces sharper inter-household separation but risks overfitting hard negatives; larger κ smooths the distribution. The final symmetric batch objective averages over all positive pairs:

$$\mathcal{L}^{\mathrm{CL}} = \frac{1}{|\mathcal{B}^+|} \sum_{(a,p)\in\mathcal{B}^+} \frac{1}{2}\big(\ell(a,p) + \ell(p,a)\big). \tag{16}$$

Model parameters are optimized using Adam [62] on $\mathcal{L}^{\mathrm{CL}}$. The sensitivity to κ is analyzed in the experiments.

3.2. Stage 2: similarity computation via clustering

Household behavior is typically *multi-modal* (e.g., workdays vs. weekends) and cannot be summarized by a single mean embedding. We therefore discretize the representation space into interpretable strategy modes and compare households via their mixture over modes. K-means [63] is chosen for its (i) centroid interpretability, (ii) computational efficiency with near-linear scaling, and (iii) compatibility with our “few dominant regimes” hypothesis. We perform clustering on the slice-level encoder representations $\mathbf{z}_{i,j}$ extracted by $f(\cdot)$ (not the projection $\mathbf{h}$). Alternative clustering algorithms are compared in ablations.

Collect all slice embeddings $\mathbf{z}_{i,j} \in \mathbb{R}^{d_z}$ across households and run K-means with K_c clusters $\{C_{k_c}\}_{k_c=1}^{K_c}$ to minimize the standard objective:

$$\mathcal{L}^{\mathrm{kmeans}} = \sum_{k_c=1}^{K_c} \sum_{\mathbf{z}\in C_{k_c}} \left\|\mathbf{z} - \boldsymbol{\mu}_{k_c}\right\|_2^2, \qquad \boldsymbol{\mu}_{k_c} = \frac{1}{|C_{k_c}|}\sum_{\mathbf{z}\in C_{k_c}} \mathbf{z}. \tag{17}$$

For each household e_i, compute the empirical cluster-frequency vector $\mathbf{p}_i \in \mathbb{R}^{K_c}$:

$$(\mathbf{p}_i)_{k_c} = \frac{1}{m_i}\sum_{j=1}^{m_i} \mathbb{1}\left[\mathbf{z}_{i,j} \in C_{k_c}\right], \qquad \sum_{k_c=1}^{K_c} (\mathbf{p}_i)_{k_c} = 1, \tag{18}$$

which summarizes its mixture over strategy modes. We instantiate the set-level similarity from the problem formulation (Section 2.1) as the cosine similarity between these mixtures:

$$\mathrm{SIM}(e_i, e_j) = \mathrm{sim}\big(\mathbf{p}_i, \mathbf{p}_j\big) \equiv \Psi\big(\mathcal{Z}_{e_i}, \mathcal{Z}_{e_j}\big). \tag{19}$$

This preserves slice-level multi-modality and yields an interpretable notion of “behavioral resemblance”: households concentrating mass on the same strategy modes are considered similar.

For neighbor selection consistent with Section 2.1, given a target household e^* we select the k most similar households by Eq. (4), where k=1 recovers the Top-1 case as a special instance.

3.3. Stage 3: zero-shot policy generation via behavioral cloning

BC learns a control policy directly from demonstrations without additional interaction with the target environment [64,65]. To obtain a zero-interaction policy for a target household, RB-ZeroHEM departs from conventional BC by using *pseudo-expert* trajectories drawn from *contextually similar* households identified in Stage 2, ensuring that demonstrations are consistent in the learned representation space.

Given a target household e^*, we follow Section 2.1 and Eqs. (17)–(19) to obtain its neighbors $\mathcal{N}_k(e^*)$ and aggregate demonstrations $\mathcal{T}_{\mathrm{demo}} = \bigcup_{e\in\mathcal{N}_k(e^*)} \mathcal{T}_e$ by concatenating trajectories into a unified dataset. No special conflict-handling mechanism is required: since neighbors share similar $\mathcal{D}$ by design, their policies are inherently consistent, and minor disagreements due to stochastic $\mathcal{U}$ are naturally resolved by standard BC training—the cross-entropy loss drives the discrete head toward the empirical mode, while the MSE loss drives the continuous head toward the conditional mean $\mathbb{E}[a^{\mathrm{cont}}|o]$. A parameterized policy π_θ is implemented as a multi-head MLP (architecture detailed in Table 3). The discrete head $\pi_\theta^{\mathrm{disc}}(o_t)$ outputs a categorical distribution over $K_d = 27$ device mode combinations (3^3 for BESS, AC, WM), and the continuous head $\pi_\theta^{\mathrm{cont}}(o_t)$ outputs normalized power setpoints scaled to physical bounds at execution. The BC objective minimizes the discrepancy between predictions and pseudo-expert actions:

$$\mathcal{L}^{\mathrm{BC}}(\theta) = \frac{1}{T_{\mathrm{demo}}} \sum_{t=1}^{T_{\mathrm{demo}}} \Big[\mathcal{L}^{\mathrm{disc}}\big(\pi_\theta^{\mathrm{disc}}(o_t), a_t^{\mathrm{disc}}\big) + \beta\, \mathcal{L}^{\mathrm{cont}}\big(\pi_\theta^{\mathrm{cont}}(o_t), a_t^{\mathrm{cont}}\big) \Big], \tag{20}$$

where

$$\mathcal{L}^{\mathrm{disc}} = -\sum_{k_d=1}^{K_d} a_{t,k_d}^{\mathrm{disc}} \log \pi_\theta^{\mathrm{disc},k_d}(o_t), \qquad \mathcal{L}^{\mathrm{cont}} = \left\|\pi_\theta^{\mathrm{cont}}(o_t) - a_t^{\mathrm{cont}}\right\|_2^2. \tag{21}$$

Here o_t denotes the observation at time t (aligned with the MDP interface used by DRL baselines), and $a_t = (a_t^{\mathrm{disc}}, a_t^{\mathrm{cont}})$ is the pseudo-expert action split into discrete and continuous parts. The index $k_d \in \{1,\dots,K_d\}$ in Eq. (21) runs over discrete classes; $a_{t,k_d}^{\mathrm{disc}}$ is a one-hot (or label-smoothed) target for class k_d, and $\pi_\theta^{\mathrm{disc},k_d}(o_t)$ is the predicted probability. $\pi_\theta^{\mathrm{cont}}(o_t)$ outputs a continuous vector of the same dimension as a_t^{cont}. T_{demo} is the number of demonstration time steps used for BC, and $\beta > 0$ balances the two heads. To improve generalization and mitigate covariate shift from a limited source, we apply Dropout (rate 0.2) in hidden layers, together with label smoothing (0.05) for the discrete head. The resulting deployable policy π_{clone} follows the definition in Eq. (5), and after training we set $\pi_{\mathrm{clone}} := \pi_\theta$. At deployment, the discrete head selects modes via $\arg\max$ to ensure deterministic behavior, consistent with stable zero-shot deployment. Device constraints (e.g., WM non-interruptibility, BESS SOC bounds) are enforced by the environment, while the policy implicitly learns constraint-aware behavior from SAC-generated demonstrations.

4. Experiments

To evaluate the effectiveness of RB-ZeroHEM, we structure our experiments around the following five research questions:

- **Q1:** Can the representation $\mathbf{z} \in \mathbb{R}^{d_z}$, learned from sliced CTs, capture the invariants of time-invariant dynamics $\mathcal{D}$ while remaining insensitive to time-varying uncertainties $\mathcal{U}$?
- **Q2:** Given the set representation $\mathcal{Z}_e = \{\mathbf{z} = f(\tau), \tau \in \mathcal{T}_e\}$, can the proposed clustering–frequency hybrid distribution reliably measure household-level similarity?
- **Q3:** Given neighbors $\mathcal{N}_k(e^*)$, can the zero-shot BC policy π_{clone} achieve effective control in the target environment e^*?
- **Q4:** How do key parameters—representation dimension d_z, temperature coefficient κ, and slice length T^τ—affect performance?
- **Q5:** Can the proposed method remain stable under seasonal variation and noise?

4.1. Experimental setup

4.1.1. Environment instantiation

We instantiate the framework described in Section 2.2 using n=12 residential environments with varied physical parameters and random inputs, based on real-world data sources. Weather and base-load data are obtained from NOAA and PecanStreet (San Diego) [56,57]. The model parameter ranges are determined according to engineering knowledge and cross-validated by a large language model (GPT-4 [66]) to ensure practical feasibility.

To evaluate both ideal transfer and realistic settings, environment e_2 is an exact replica of e_1, e_3 is similar but not identical, and the remaining environments cover a wide spectrum of device and thermal parameters. The key, time-invariant physical dynamic parameters $\mathcal{D}$ are listed in Table 2, while uncertainties $\mathcal{U}$ (occupancy, demand, weather,

Table 2
Varied model parameters in different household environments (fixed dynamics $\mathcal{D}$).

	AC				BESS				WM			PV				Thermal							
e_i	$P^{\mathrm{a}}_{\mathrm{rat}}$	η^{a}	$P^{\mathrm{a}}_{\mathrm{sta}}$	$\mathrm{cop}_{\mathrm{rat}}$	$P^{\mathrm{b}}_{\mathrm{rat}}$	η^{b}	E_{rat}	SD	$P^{\mathrm{w}}_{\mathrm{rat}}$	$P^{\mathrm{w}}_{\mathrm{sta}}$	$z^{\mathrm{w}}_{\mathrm{task}}$	$P^{\mathrm{PV}}_{\mathrm{rat}}$	η^{PV}	$I_{\min}$	A^{pan}	ACH	A^{fen}	A^{surf}	c^{air}	SHGC	λ	V^{hou}	ρ^{air}
e_1	3.0	0.08	0.010	3.91	3.0	0.98	50	0.005	1.2	0.01	45	1.3	0.2	10	10	0.7	16	266	0.3	0.3	0.93	420	1.30
e_2	3.0	0.08	0.010	3.91	3.0	0.98	50	0.005	1.2	0.01	45	1.3	0.2	10	10	0.7	16	266	0.3	0.3	0.93	420	1.30
e_3	3.0	0.06	0.020	3.79	3.0	0.95	48	0.008	1.2	0.02	45	1.3	0.18	10	10	0.65	16	266	0.3	0.28	0.90	420	1.30
e_4	2.5	0.045	0.020	3.20	5.0	0.95	80	0.006	1.1	0.02	55	1.5	0.18	15	10	1.2	10	180	0.25	0.18	1.00	320	1.25
e_5	3.2	0.08	0.010	5.20	2.5	0.96	30	0.003	0.9	0.015	45	2.5	0.24	15	20	0.7	35	450	0.29	0.28	0.70	800	1.25
e_6	2.0	0.05	0.020	3.50	4.0	0.96	70	0.005	1.2	0.02	60	1.0	0.20	15	8	1.5	8	120	0.23	0.35	1.10	220	1.25
e_7	2.8	0.10	0.020	2.80	3.0	0.90	40	0.008	1.0	0.015	30	2.0	0.21	15	16	0.8	20	250	0.29	0.25	1.00	400	1.25
e_8	2.6	0.07	0.015	4.50	2.0	0.92	20	0.004	0.8	0.01	35	3.0	0.23	15	24	0.6	28	270	0.28	0.30	0.80	420	1.25
e_9	1.5	0.04	0.010	5.00	3.0	0.95	50	0.002	0.7	0.012	50	2.0	0.26	15	15	0.4	6	150	0.24	0.22	0.60	250	1.25
e_{10}	3.5	0.12	0.025	2.50	5.0	0.88	100	0.010	1.1	0.02	20	1.5	0.19	15	12	2.0	40	380	0.30	0.30	1.50	700	1.25
e_{11}	2.2	0.06	0.012	4.80	2.0	0.95	30	0.003	0.9	0.01	40	2.8	0.25	15	20	0.5	15	200	0.26	0.27	0.70	300	1.25
e_{12}	1.2	0.03	0.008	5.50	3.0	0.97	60	0.002	0.5	0.008	25	1.5	0.28	15	10	0.3	5	100	0.22	0.20	0.50	180	1.25

base load, and electricity price) are independently sampled and applied in each environment.

4.1.2. Control trajectory generation

To obtain training data for representation learning and BC, we generate CTs by running controllers in each environment. For source environments, a Soft Actor–Critic (SAC) [67] agent is trained as a maximum-entropy DRL controller to produce high-quality trajectories. For target environments e^*, a rule-based controller (RBC) [68] is employed to emulate cases without pretrained DRL policies. These controllers serve both as data generators and as baseline methods for comparison.

Each environment provides a control trajectory up to 360 days covering all seasons. For realistic transfer evaluation, the default training window is a 90-day contiguous trajectory within the same season (cross-season generalization is discussed in the robustness study). The control interval is $\Delta t = \frac{5}{60}$ hours, and the default slice length is T^{τ}=288; slices are non-overlapping.

4.1.3. Implementation and training details

Table 3 lists the parameters used in the proposed method. When multiple options are available, the default value is highlighted in bold. The encoder $f(\cdot)$ for contrastive representation learning is a multi-scale one-dimensional convolutional network, while the projection head $g(\cdot)$ is a two-layer MLP. The network architecture is detailed in Section 3.1.

4.1.4. Evaluation metrics

We evaluate the method along two axes: (1) representation quality and (2) control performance in the target environment e^*.

Representation quality

- **ACC.** To measure whether representations $\mathbf{z}$ naturally cluster by household identity, we evaluate a k-nearest neighbor (KNN) classification accuracy. Each embedding $\mathbf{z}_i$ is associated with a household label y_i. We compute the proportion of embeddings whose nearest neighbors share the same label:

$$\mathrm{ACC} = \frac{1}{|\mathcal{Z}|} \sum_{i=1}^{|\mathcal{Z}|} \mathbb{I}\left[y_i = \mathrm{mode}\left(\{y_j : \mathbf{z}_j \in \mathcal{N}_k(\mathbf{z}_i) \setminus \{\mathbf{z}_i\}\}\right)\right], \tag{22}$$

where $\mathcal{N}_{k_z}(\mathbf{z}_i)$ returns the $k_z = 5$ nearest neighbors of $\mathbf{z}_i$ under cosine distance. A higher ACC indicates that embeddings from the same household are closer to each other in the latent space.
- **CMP.** We quantify compactness using the K-means objective on slice-level embeddings $\mathbf{z}$:

$$\mathrm{CMP} = \frac{1}{|\mathcal{Z}|} \mathcal{L}_{\mathrm{kmeans}} \tag{23}$$

Table 3
Experimental hyperparameters and their sensitivity analysis range. For the meaning of each symbol, please refer to Nomenclature section.

Symbol	Default and sensitivity range
Contrastive Representation Learning	
T^{τ}	$\{32, 96, \mathbf{288}, 864, 2592\}$
d_z	$\{32, 64, \mathbf{128}, 256, 512\}$
d_h	8
$2N$	256
κ	$\{0.05, 0.1, \mathbf{0.2}, 0.5, 1.0\}$
Optimizer	Adam, lr 10^{-3}, weight decay 10^{-5}; cosine decay with 10-epoch warmup; gradient clip 1.0
Training	1000 epochs; early stopping (patience 15); report mean±std and 95% CI over 10 seeds
Similarity Computation via Clustering	
K_c	$\{6, 12, \mathbf{24}, 48, 96\}$
Neighbor rule	Top-1
Zero-Shot Policy Generation via BC	
π_θ	MLP (5 hidden layers; widths [512, 256, 64, 256, 512]; ReLU; dropout 0.2)
β	1
Batch size	1024
Optimizer	Adam, lr $3{\times}10^{-4}$, weight decay 10^{-5}

- **RHO.** To test whether embeddings capture time-invariant dynamics $\mathcal{D}$ (cf. Table 2), we first obtain a *household-level* embedding by averaging slice embeddings: $\bar{\mathbf{z}}_e = \frac{1}{|\mathcal{Z}_e|}\sum_{\mathbf{z}\in\mathcal{Z}_e} \mathbf{z}, \quad e \in \mathcal{E}$. For each dimension j, collect $\mathbf{z}^{(j)} = \left[\bar{z}_e^{(j)}\right]_{e\in\mathcal{E}} \in \mathbb{R}^n$. For each physical parameter column $n_{\mathrm{p}} = 23$ in Table 2, collect its values across households $\mathbf{d}_{n_{\mathrm{p}}} = \left[d_{n_{\mathrm{p}}}(e)\right]_{e\in\mathcal{E}}$ and apply min–max normalization over households: $\tilde{\mathbf{d}}_{n_{\mathrm{p}}} = \frac{\mathbf{d}_{n_{\mathrm{p}}} - \min(\mathbf{d}_{n_{\mathrm{p}}})}{\max(\mathbf{d}_{n_{\mathrm{p}}}) - \min(\mathbf{d}_{n_{\mathrm{p}}})}$. We then report the median absolute Spearman rank correlation over all pairs of embedding dimensions and physical parameters:

$$\mathrm{RHO} = \mathrm{median}_{j,n_{\mathrm{p}}} \left| \rho_s\left(\mathbf{z}^{(j)}, \tilde{\mathbf{d}}_{n_{\mathrm{p}}}\right) \right|. \tag{24}$$

A larger RHO indicates stronger monotonic alignment between learned embeddings and time-invariant physical parameters.
- **DIST.** To assess robustness against exogenous perturbations, we inject mean-based additive Gaussian noise to exogenous variables $v_t \in \{T_t^{\mathrm{out}}, I_t, P_t^{\mathrm{BL}}\}$:

$$\tilde{v}_t = v_t + \epsilon_v \, \bar{v}, \qquad \epsilon_v \sim \mathcal{N}(0, \sigma_v^2).$$

After re-encoding trajectories, we measure the average Euclidean distance between the original and perturbed embeddings:

$$\mathrm{DIST} = \frac{1}{|\mathcal{Z}|}\sum_{i=1}^{|\mathcal{Z}|}\|\mathbf{z}_i - \tilde{\mathbf{z}}_i\|_2. \tag{25}$$

A smaller value indicates greater robustness and invariance of representations to perturbations in exogenous variables.

Control performance in target environment

- **DR.** Each household e has a most comfortable temperature T_e^*; the comfort band is $T_e^* \pm 2$ °C. The violation ratio is

$$\mathrm{DR} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{I}\Big(\left|T_t^{\mathrm{in}} - T_e^*\right| > 2\Big). \tag{26}$$

- **COST.** We only purchase from the grid; the time-of-use cost is

$$\mathrm{COST} = \sum_{t=1}^{T}\left[P_t^D\right]_+ \mathrm{pri}_t\,\Delta t. \tag{27}$$

- **ULR.** Let n^{lau} be the total number of laundry demands. Demand i is met if the washer turns on at least once within $[t_i^{\mathrm{lau}}, t_i^{\mathrm{lau}} + z_i^{\mathrm{lau}}]$:

$$\mathrm{ULR} = 1 - \frac{1}{n^{\mathrm{lau}}}\sum_{i=1}^{n^{\mathrm{lau}}}\mathbb{I}\left(\sum_{t=t_i^{\mathrm{lau}}}^{t_i^{\mathrm{lau}}+z_i^{\mathrm{lau}}} M_t^{\mathrm{w}} \ge 1\right). \tag{28}$$

- **GES.** Relative grid-energy saving due to BESS is

$$\mathrm{GES} = 1 - \frac{\sum_{t=1}^{T}\left[P_t^D - M_t^{\mathrm{b}}P_t^{\mathrm{b}}\right]_+\Delta t}{\sum_{t=1}^{T}\left[P_t^D\right]_+\Delta t}. \tag{29}$$

- **N_INT.** Number of environment interactions required at deployment; zero-shot policies satisfy $N_{\mathrm{INT}} = 0$.

4.1.5. Baselines and variants

Given that the three stages of the proposed RB-ZeroHEM pipeline are modular, we benchmark each stage against targeted baselines and ablated variants to rigorously assess effectiveness and performance advantages. Unless otherwise stated, network backbones, optimization settings, and data splits follow those in Sections 3 and 4.1.

Stage 1: representation learning (slice-level)

S1.1 **Raw CTs.** No neural mapping; use raw CT slices $\tau = \{(o_t, a_t)\}_{t=1}^{T_\tau}$ as representations.

S1.2 **Supervised-Classification Representations.** Replace $g(\cdot)$ with a classification head (output size n=12, softmax) and train with cross-entropy (Eq. 21) instead of InfoNCE (Eq. 16); $f(\cdot)$ remains unchanged, and encoder outputs are used as representations.

S1.3 **Autoencoder Representations.** Replace $g(\cdot)$ with a symmetric decoder $d(\cdot)$ that reconstructs the input; train with mean squared error $\mathcal{L}^{\mathrm{AE}} = \frac{1}{T_\tau}\sum_{t=1}^{T_\tau}\|d(f(\tau))_t - (o_t, a_t)\|_2^2$ instead of InfoNCE; the bottleneck representation $\mathbf{z} = f(\tau)$ is used as the slice embedding.

S1.4 **w/o Projection Head.** Remove $g(\cdot)$ and apply InfoNCE directly on $f(\cdot)$'s outputs; the encoder outputs serve as representations.

S1.5 **w/o Multi-Scale Encoder.** Replace the multi-scale encoder with a 3-layer 1D-CNN (kernels 5/5/5, channels $\{64, 128, 128\}$) and temporal global average pooling to obtain d_z=128; train with InfoNCE (using $g(\cdot)$), and use the encoder output $\mathbf{z}$ for evaluation.

Stage 2: clustering & similarity (household-level)

S2.1 **Alternative Clusterers.** Replace K-means with (i) Gaussian mixture models (GMM) [69], (ii) spectral clustering [70], and (iii) HDBSCAN [71]. The construction of cluster-frequency vectors $\mathbf{p}_e$ and the subsequent similarity computation $\mathrm{SIM}(e_i, e_j)$ are kept *identical* across all methods to ensure strict comparability.

Stage 3: policy generation (target environment)

S3.1 **RBC.** Apply a rule-based controller in the target environment following [68].

S3.2 **Online DRL (From Scratch).** Train native DRL agents that directly interact with the target environment, including SAC [67], PPO [72] and TD3 [73].

S3.3 **TL.** Initialize the target policy by copying DNN parameters from a source-environment policy, with or without subsequent fine-tuning in the target environment(a standard/naïve transfer approach; cf. [30]).

S3.4 **Offline RL (CQL).** Using the trajectory dataset generated by the target-environment RBC, train a target policy via Conservative Q-Learning (CQL) in an offline manner [74].

S3.5 **Meta-RL (MAML-SAC).** Model-Agnostic Meta-Learning [47] with SAC, also adopted in recent HEMS work [36]. Meta-train on the $n-1$ source environments (excluding target) via bi-level optimization; the inner loop adapts to each task and the outer loop optimizes the meta-initialization. Evaluate under zero-shot and few-shot (576 interactions) settings.

4.1.6. Hardware

All simulations ran on Ubuntu 20.04, Python 3.10, PyTorch 2.0, with an Intel i7-14600K CPU and an NVIDIA GeForce RTX 4090 GPU.

4.2. Representation quality (Q1)

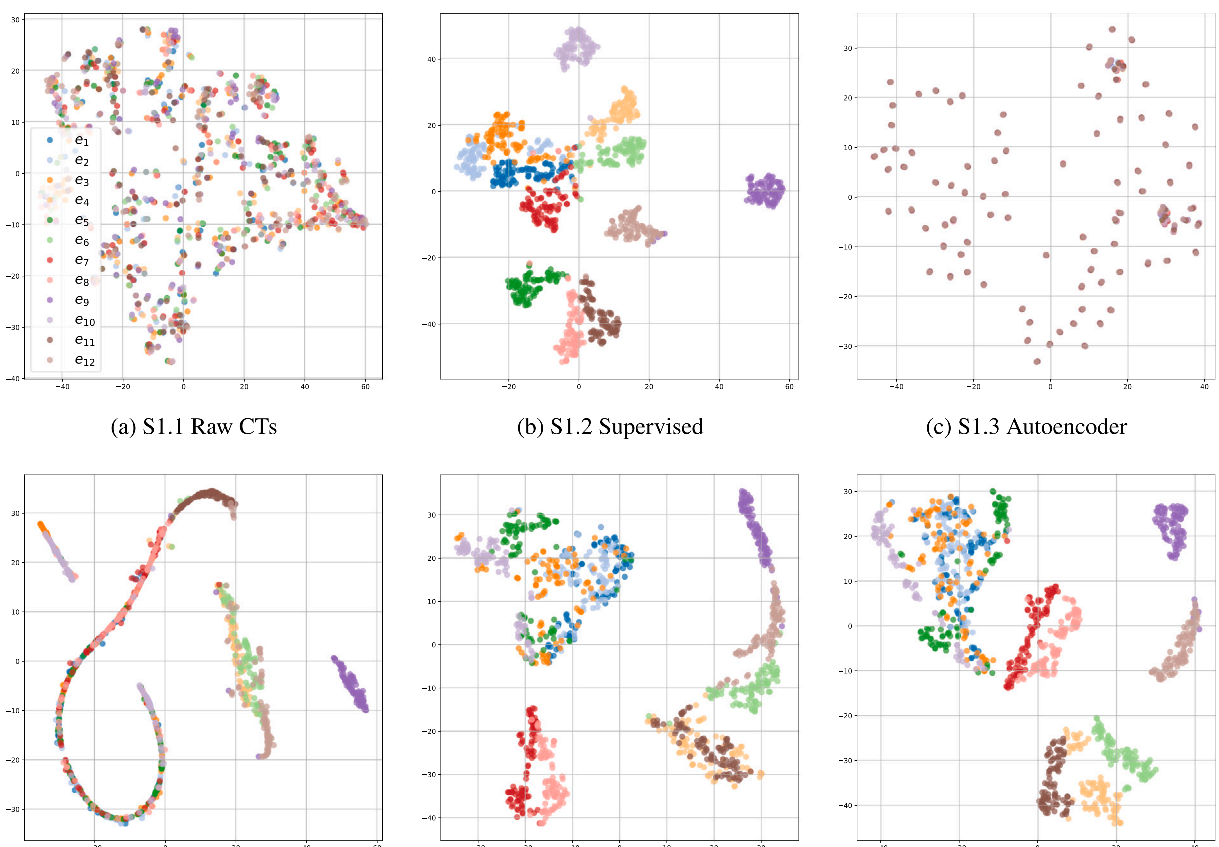
Table 4 summarizes ACC, CMP, and RHO across four seasons, while Figs. 6 and 8 visualize slice-level structure (t-SNE) and KNN confusion. Our method yields consistently higher RHO than all Stage-1 baselines (e.g., spring/summer: $(0.49\pm0.03/0.51\pm0.03)$), indicating that the household embeddings align more strongly with time-invariant physics ($\mathcal{D}$). Compared with the supervised classifier S1.2, RB-ZeroHEM achieves a relative RHO gain of about (36%–43%). By contrast, S1.2 tops ACC yet lags on RHO—a signature of representation drift toward household-specific uncertainties ($\mathcal{U}$). This trade-off is visible in the plots: S1.2 forms very tight, label-pure clusters (high ACC) but separates physically identical pairs such as (e_1)–(e_2), fragmenting the physics manifold; RB-ZeroHEM maintains compact, well-separated clusters while partially overlapping households that are physically identical or near-identical (Fig. 6) and correspondingly reduces cross-house confusion (Fig. 8).

S1.3 (autoencoder) suffers from *representation collapse*: ACC≈18%, CMP≈2, and the similarity matrix (Fig. 10(b)) degenerates to all ones. The reconstruction objective provides no discriminative pressure—MSE can be minimized by encoding shared low-level statistics (amplitude, periodicity) common across households, while the decoder recovers details from these generic features. Without contrasting same- vs. cross-household samples, embeddings converge to a narrow subspace encoding "generic trajectory" patterns rather than household-specific $\mathcal{D}$. The t-SNE (Fig. 6(c)) and confusion matrix (Fig. 8(c)) confirm this: points scatter structurelessly, and predictions collapse into a few dominant classes. This underscores that discriminative objectives—contrastive pairs or explicit labels—are essential for transfer-relevant representations.

Zooming in on the contrastive ablations, S1.4 (no projection head) and S1.5 (no multi-scale encoder) both underperform ours on RHO and CMP across all seasons. The projection head provides a sacrificial space in which to optimize InfoNCE without distorting the encoder's reusable representations, thereby improving inter-household margins while preserving intra-household invariants, in line with prior contrastive learning literature [58,59]. The multi-scale temporal encoder captures device and thermal dynamics at heterogeneous time scales (cycle-level actions, diurnal rhythms) that single-scale CNNs miss. Together, these design choices explain why our embeddings simultaneously achieve (i) lower CMP (tighter, more compact clusters) and (ii) higher RHO (better monotonic alignment with $\mathcal{D}$).

Table 4
Four-season representation quality across Stage-1 baselines (mean ± 95% CI over 10 seeds). Relative improvement of Ours over S1.2 on RHO is annotated in parentheses.

Method	Spring (90d)			Summer (90d)		
	ACC (%)	CMP	RHO	ACC (%)	CMP	RHO
S1.1	33.62 ± 3.10	778500.22 ± 4980.41	0.29 ± 0.02	35.51 ± 3.25	762300.09 ± 4829.92	0.32 ± 0.02
S1.2	93.42 ± 1.93	66.41 ± 3.10	0.36 ± 0.03	97.95 ± 1.86	62.34 ± 3.12	0.37 ± 0.03
S1.3	18.61 ± 0.25	2.41 ± 0.02	0.32 ± 0.01	19.15 ± 0.27	2.73 ± 0.02	0.33 ± 0.01
S1.4	66.11 ± 2.35	99.86 ± 4.72	0.37 ± 0.03	68.03 ± 2.33	95.20 ± 4.50	0.39 ± 0.04
S1.5	85.22 ± 1.65	120.53 ± 5.87	0.43 ± 0.03	88.61 ± 1.51	117.98 ± 5.80	0.45 ± 0.03
Ours	89.74 ± 1.46	110.82 ± 4.54	**0.49 ± 0.03** (+36.1%)	91.69 ± 1.29	109.04 ± 5.02	**0.51 ± 0.03** (+37.8%)
Method	**Autumn (90d)**			**Winter (90d)**		
	ACC (%)	CMP	RHO	ACC (%)	CMP	RHO
S1.1	31.83 ± 3.05	784600.33 ± 5090.21	0.28 ± 0.02	38.92 ± 3.22	809400.14 ± 5203.55	0.27 ± 0.02
S1.2	94.24 ± 1.95	68.52 ± 3.20	0.36 ± 0.03	96.03 ± 2.02	72.35 ± 3.28	0.32 ± 0.03
S1.3	18.12 ± 0.22	2.19 ± 0.02	0.31 ± 0.01	17.48 ± 0.21	1.93 ± 0.01	0.29 ± 0.01
S1.4	64.41 ± 2.44	103.71 ± 4.83	0.35 ± 0.03	62.20 ± 2.57	108.45 ± 4.96	0.34 ± 0.04
S1.5	84.23 ± 1.73	122.34 ± 5.92	0.42 ± 0.03	82.06 ± 1.84	126.83 ± 5.95	0.40 ± 0.03
Ours	88.94 ± 1.49	112.52 ± 4.60	**0.48 ± 0.03** (+33.3%)	87.31 ± 1.57	116.45 ± 4.71	**0.46 ± 0.03** (+43.6%)



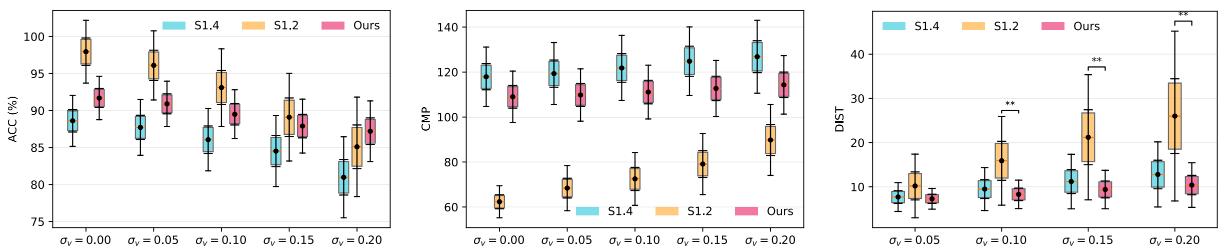
(a) S1.1 Raw CTs (b) S1.2 Supervised (c) S1.3 Autoencoder

(d) S1.4 w/o Proj. Head (e) S1.5 w/o Multi-Scale (f) Ours

Fig. 6. t-SNE [75] scatter plot of representations.

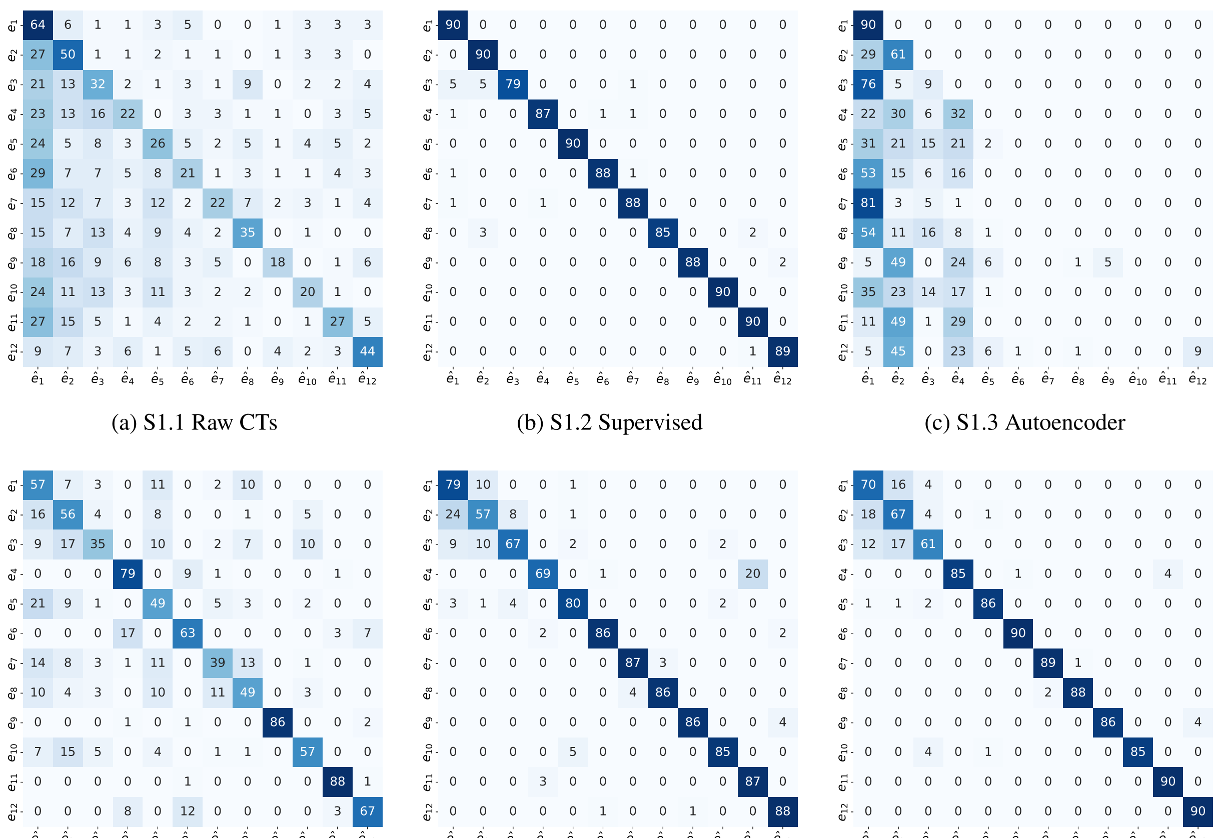
Fig. 7 probes robustness by injecting Gaussian noise into exogenous series ($v_t \in \{T_t^{\text{out}}, I_t, P_t^{\text{BL}}\}$). As noise grows, S1.2's ACC/CMP degrades more steeply, consistent with its reliance on ($\mathcal{U}$). RB-ZeroHEM shows flatter ACC/CMP curves and significantly smaller DIST—with Mann–Whitney U significance at moderate–high (σ_v)—demonstrating that the learned codes are insensitive to perturbations in exogenous variables and therefore closer to the sought invariants of ($\mathcal{D}$). This invariance is a direct consequence of the contrastive setup over within-house slices: positives share ($\mathcal{D}$) but differ in ($\mathcal{U}$), pushing the encoder to treat the latter as nuisance.

Finally, the seasonal panels in Table 4 confirm stability: RB-ZeroHEM maintains high RHO and competitive ACC from spring to winter, while baselines—especially S1.4/S1.5—exhibit larger seasonal swings. Together, these results answer **Q1** in the affirmative: slice-level contrastive learning with a multi-scale encoder and projection head produces representations that (i) naturally group by household identity



(a) ACC. (b) CMP. (c) DIST.

Fig. 7. Performance under injected Gaussian noise σ_v. ACC, CMP, and DIST are reported for S1.2, S1.5, and Ours across multiple noise levels. Each box shows results over 10 deterministic seeds; black dots denote means and whiskers mark 95% confidence intervals. Stars in DIST indicate Mann–Whitney U significance (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$).



(a) S1.1 Raw CTs (b) S1.2 Supervised (c) S1.3 Autoencoder

(d) S1.4 w/o Proj. Head (e) S1.5 w/o Multi-Scale (f) Ours

Fig. 8. Confusion matrix of KNN classification results.

without overfitting to ($\mathcal{U}$), (ii) compactly capture the latent physics ($\mathcal{D}$), and (iii) remain robust under noise and seasonal shifts (Figs. 6 and 7, Table 4).

4.3. Household similarity (Q2)

We evaluate whether the clustering–frequency hybrid distribution yields accurate and efficient household-level similarity. Fig. 9 visualizes the cluster-frequency vectors $\mathbf{p}_e$; Fig. 10 compares similarity matrices with the supervised baseline (S1.2); Fig. 11 quantifies agreement with a physics-based reference under varying thresholds; and Table 5 benchmarks alternative clustering back-ends.

In Fig. 9, environments with identical or near-identical physical parameters (notably (e_1)–(e_2), and partly (e_3)) exhibit highly similar ($\mathbf{p}_e$) (mass concentrated on the same few clusters). By contrast, households that differ markedly in physical parameters (e.g., e_{10}) show distinct histograms. This matches the goal of the set representation: slices from households sharing time-invariant dynamics D repeatedly fall into similar latent regions, producing stable frequency signatures despite fluctuations in $\mathcal{U}$.

At the pairwise level (Fig. 10), RB-ZeroHEM yields a block-diagonal structure aligned with known physical regimes: it assigns high similarity to (e_1, e_2) and moderate similarity to (e_1, e_3). The supervised baseline (S1.2) misranks several pairs, inflating similarity caused by incidental $\mathcal{U}$ patterns and suppressing truly physics-adjacent pairs—consistent with **Q1**'s finding that S1.2 emphasizes label separation rather than invariance to $\mathcal{U}$. The autoencoder baseline (S1.3) degenerates to a uniform matrix of ones, rendering neighbor retrieval eonstruction-based representations lack the discriminative strumeaningless and confirming that

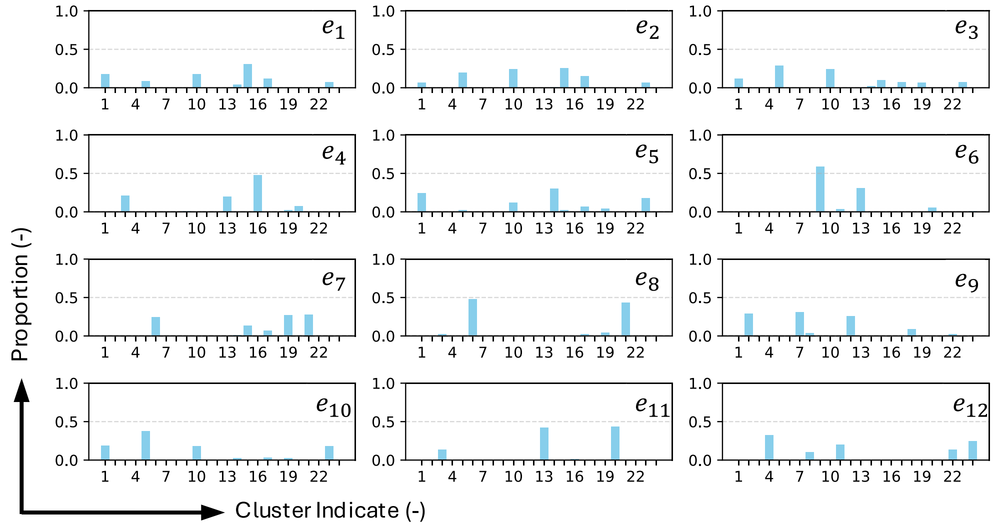
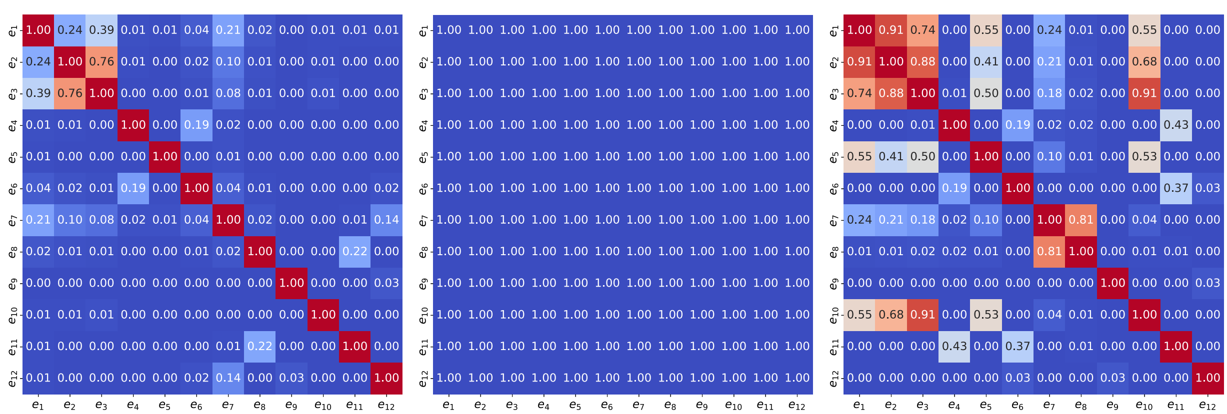


Fig. 9. Cluster distribution per environment.



(a) S1.2 Supervised (b) S1.3 Autoencoder (c) Ours

Fig. 10. Comparison of learned similarity matrices: S1.2 vs. S1.3 vs. ours.

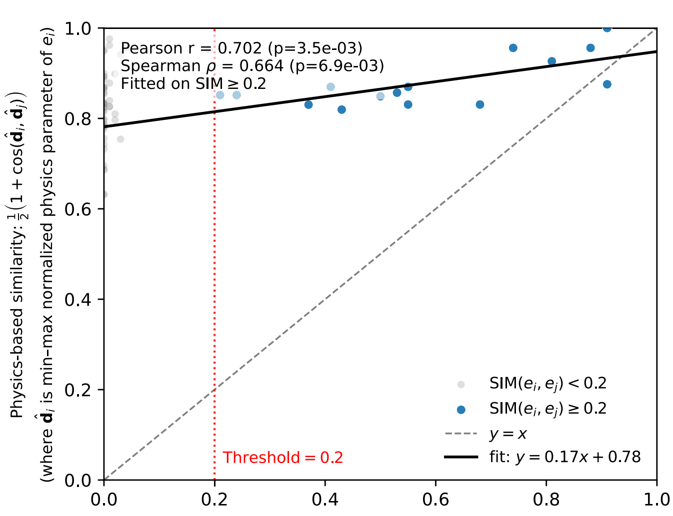


Fig. 11. Similarity consistency between physics-based and observed metrics.

reconstruction-based representations lack the discriminative structure required for similarity computation.

Fig. 11 examines how observed similarity SIM(e_i, e_j) relates to a physics-based reference from Table 2. The relationship is not globally linear: when SIM is small, physically similar pairs can still appear. After filtering out these low-SIM cases, the two measures show clear positive monotonic consistency (Pearson r=0.702, Spearman ρ=0.664). This indicates a nonlinear mapping between parameters and actual dynamics, justifying the use of a nonlinear encoder to extract similarity directly from trajectories. Moreover, when our method reports high SIM, the underlying physical parameters are indeed similar, confirming the effectiveness of the approach.

Table 5 shows that K-means and GMM return identical Top-5 neighbors for the query e_1 (recovering e_2, e_3) with nearly identical scores, while K-means is an order of magnitude faster (0.0077 s vs. 0.0541 s). Spectral clustering performs worse, likely due to sensitivity to graph construction in dense slice clouds. HDBSCAN yields saturated similarities (1.00) because its auto-selected K_c=4 collapses many slices into a few "core" clusters, making histogram overlaps trivially maximal. Considering accuracy–efficiency trade-offs, K-means is a reasonable default.

Table 5
Effect of clustering back ends on top-5 neighbor retrieval for query household e_1. Runtimes measured on an Intel core i7-14600K CPU and an NVIDIA RTX 4090 GPU.

Method	Top-5 similarity households (e_j = SIM(e_1, e_j))					Setting	Calc. time (s)
K-Means	e_2=0.91	e_3=0.74	e_5=0.55	e_{10}=0.55	e_7=0.24	K_c=24; `n_init=auto`	0.0077 ± 0.0006
GMM	e_2=0.91	e_3=0.74	e_5=0.55	e_{10}=0.55	e_7=0.24	K_c=24; `covariance_type=full`	0.0541 ± 0.0045
Spectral	e_2=0.71	e_3=0.53	e_5=0.17	e_{10}=0.14	e_7=0.04	K_c=24; $n_{\text{neighbors}}$=10	0.0632 ± 0.0051
HDBSCAN	e_2=1.00	e_3=1.00	e_5=1.00	e_{10}=1.00	e_7=1.00	$\min(C_{k_c}) = \max(5, 0.01\lvert\mathcal{Z}\rvert)$	0.0778 ± 0.0062;

Table 6
Performance in e_1 (90-day windows). Values are mean ± 95% CI. Δ in FT rows is vs. SAC within the same season: Δ = (SAC − Method)/SAC.

Method	Spring (90d)				Summer (90d)				N_{INT}
	DR (%)	COST ($)	ULR (%)	GES (%)	DR (%)	COST ($)	ULR (%)	GES (%)	
S3.1 RBC	12.34±0.58	228.15±5.87	0.00±0.00	15.12±1.48	13.58±0.62	236.54±6.12	0.00±0.00	13.83±1.53	0
S3.2 PPO	8.53±0.44	192.34±5.12	0.89±0.28	18.76±1.61	9.21±0.48	199.78±5.38	1.31±0.32	17.47±1.66	24192
S3.2 TD3	10.48±0.53	214.67±5.54	8.76±2.18	16.52±1.72	11.31±0.58	221.25±5.72	10.14±2.40	15.34±1.78	24192
S3.2 SAC	3.52±0.28	172.18±4.45	0.00±0.00	25.34±2.08	3.84±0.31	178.44±4.68	0.00±0.00	24.12±2.12	24192
S3.3 TL(e_3) (w/o FT)	10.25±0.62	206.43±4.72	5.12±1.67	18.54±1.68	11.13±0.69	213.0±4.95	6.29±1.89	17.38±1.73	0
S3.3 TL(e_3) (w/ FT)	4.05±0.32	170.24±4.62	1.18±0.37	24.28±2.03	4.40±0.36	176.03±4.88	1.53±0.41	23.05±2.07	576
S3.4 CQL	11.96±0.42	215.32±5.48	2.01±0.19	14.67±1.79	12.83±0.45	221.5±5.70	2.45±0.22	13.45±1.84	0
S3.5 MAML-SAC (w/o FT)	15.83±0.71	241.26±6.23	12.47±2.35	12.34±1.52	16.72±0.76	248.93±6.48	14.18±2.61	11.28±1.58	0
S3.5 MAML-SAC (w/ FT)	4.28±0.34	175.63±4.58	0.42±0.18	24.15±2.01	4.67±0.37	182.28±4.82	0.58±0.22	22.87±2.06	576
Ours (e_2)	6.89±0.41	206.54±5.32	0.00±0.00	22.47±1.83	7.58±0.46	213.11±5.57	0.00±0.00	21.24±1.88	0
Ours (e_3)	8.24±0.46	193.28±5.01	1.42±0.63	20.58±1.69	9.02±0.50	199.61±5.24	1.74±0.71	19.35±1.74	0
Ours (e_{10})	14.76±0.67	237.43±6.18	14.58±2.71	11.89±1.19	15.92±0.72	244.17±6.41	16.21±2.94	10.89±1.23	0
Ours (e_3)(w/ FT)	3.87±0.29	166.52±4.35	0.00±0.00	25.82±2.07	4.21±0.33	172.16±4.58	0.00±0.00	24.59±2.11	576
Method	**Autumn (90d)**				**Winter (90d)**				N_{INT}
	DR (%)	COST ($)	ULR (%)	GES (%)	DR (%)	COST ($)	ULR (%)	GES (%)	
S3.1 RBC	13.91±0.65	242.78±6.34	0.00±0.00	13.25±1.56	15.76±0.71	258.34±6.78	0.00±0.00	11.43±1.62	0
S3.2 PPO	9.87±0.51	205.42±5.61	1.56±0.35	16.38±1.69	11.24±0.58	218.67±5.93	2.13±0.41	14.52±1.76	24192
S3.2 TD3	12.05±0.61	227.89±5.94	11.32±2.58	14.26±1.81	13.67±0.68	241.53±6.35	13.45±2.89	12.78±1.89	24192
S3.2 SAC	4.17±0.33	184.72±4.89	0.00±0.00	22.95±2.15	5.28±0.38	196.84±5.24	0.00±0.00	20.34±2.21	24192
S3.3 TL(e_3) (w/o FT)	11.84±0.72	219.67±5.16	7.21±2.04	16.42±1.76	13.29±0.79	233.45±5.48	8.76±2.31	14.65±1.82	0
S3.3 TL(e_3) (w/ FT)	4.73±0.38	181.84±5.07	1.82±0.45	21.84±2.09	5.89±0.43	193.26±5.38	2.34±0.52	19.42±2.16	576
S3.4 CQL	13.52±0.47	228.14±5.89	2.87±0.24	12.73±1.87	14.91±0.52	241.68±6.27	3.42±0.28	11.24±1.93	0
S3.5 MAML-SAC (w/o FT)	17.26±0.78	254.83±6.71	15.34±2.78	10.87±1.61	19.14±0.85	271.45±7.12	17.82±3.15	9.23±1.68	0
S3.5 MAML-SAC (w/ FT)	4.89±0.38	187.42±5.02	0.67±0.24	21.73±2.08	5.94±0.43	199.87±5.38	0.89±0.29	19.56±2.14	576
Ours (e_2)	8.16±0.49	219.78±5.79	0.00±0.00	20.15±1.91	9.43±0.54	233.42±6.18	0.00±0.00	17.86±1.96	0
Ours (e_3)	9.68±0.53	205.93±5.45	2.08±0.78	18.27±1.77	11.15±0.60	218.76±5.82	2.89±0.91	16.42±1.83	0
Ours (e_{10})	16.84±0.76	251.42±6.67	17.53±3.12	10.12±1.26	18.52±0.83	265.78±7.08	19.67±3.45	8.67±1.31	0
Ours (e_3) (w/ FT)	4.54±0.35	177.89±4.78	0.00±0.00	23.41±2.13	5.67±0.41	189.34±5.12	0.00±0.00	21.15±2.19	576

Contrastive learning shapes a slice-level space where trajectories with similar $\mathcal{D}$ co-locate despite variations in $\mathcal{U}$. Aggregating to $\mathbf{p}_e$ estimates each household's occupancy on this "physics map," so similarity between $\mathbf{p}_e$ vectors reflects proximity in $\mathcal{D}$, not coincidences in $\mathcal{U}$. This establishes that the clustering–frequency hybrid reliably measures household-level similarity, thereby answering **Q2**.

4.4. Control performance (Q3)

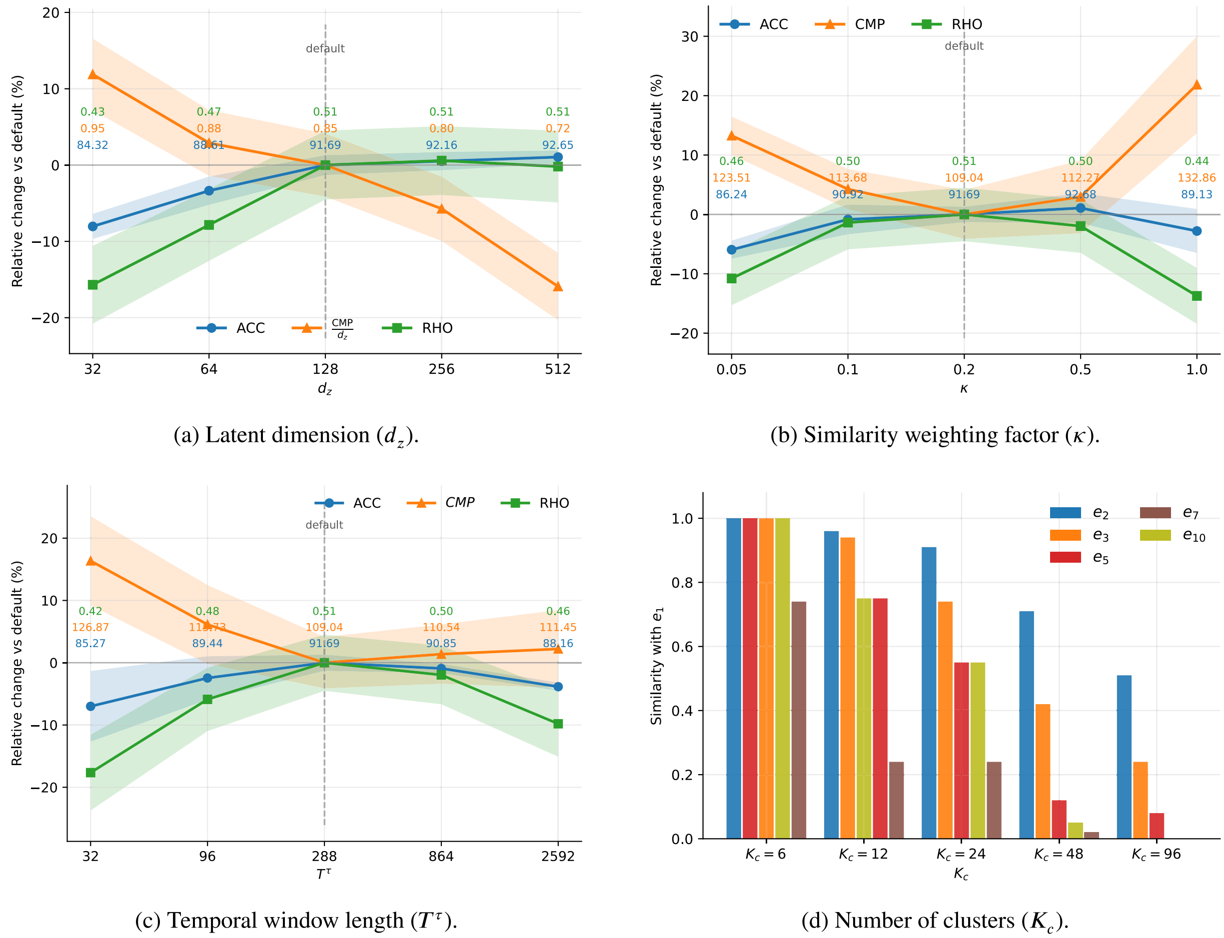
Table 6 summarizes control performance over four seasons (90d windows). We compare zero-interaction RB-ZeroHEM zero-shot cloning (Ours, N_{INT}=0) against RBC, online DRL (PPO/TD3/SAC, N_{INT}=24192), TL, offline CQL, and Meta-RL (MAML-SAC); we also report small fine-tuning variants (w/ FT, N_{INT}=576) as upper bounds. The triplet (e_2, e_3, e_{10}) represents ideal, realistic, and mismatched source settings, respectively.

On targets close to the source (e_2, e_3), Ours outperforms RBC and CQL with zero interaction and is competitive with PPO/TD3, though still slightly worse than the best online method SAC. For example in spring, relative to RBC, Ours(e_3) reduces DR from 12.34% to 8.24% −33%, lowers COST from 228.15 to 193.28 (−15.3%), and raises GES to 20.58%; compared to PPO, it achieves lower DR/COST and higher GES, while remaining behind SAC (e.g., spring (COST = 193.28 vs. 172.18). Note that ULR is the unmet ratio (lower is better); Ours is near 0% in most seasons, on par with SAC and better than the long-tailed violations of PPO/TD3 (e.g., spring ULR= 0.89%/8.76%).

With small-scale fine-tuning on Ours (e_3) (w/ FT, N_{INT}=576), performance approaches or surpasses SAC on several metrics, using only ≈ 2.4% of SAC's interaction budget. In spring, COST drops from 193.28 to 166.52 (beating SAC's 172.18), GES improves to 25.82% (comparable to SAC's 25.34%), and DR remains in the same ballpark (3.87% vs. 3.52%). The pattern holds in autumn/winter, indicating that zero-shot initialization plus light fine-tuning is a cost-effective deployment path.

Direct parameter copy in TL (w/o FT) can degrade substantially in the target domain (e.g., spring DR= 10.25%, COST= 206.43), highlighting that "parameter transferability" ≠ "behavioral transferability." With fine-tuning (TL w/ FT), performance nears SAC, but the initial policy is more interaction-hungry. Offline CQL underperforms Ours and online DRL in this setup, suggesting that target-domain RBC trajectories alone are insufficient to learn a strong policy. MAML-SAC shows poor zero-shot performance (e.g., spring DR= 15.83%, COST= 241.26), even worse than RBC, since the meta-initialization favors rapid adaptation over direct transfer. With 576 interactions it improves sharply (DR= 4.28%, COST= 175.63), approaching SAC and slightly outperforming TL (w/ FT). This confirms MAML's strength in few-shot adaptation but not zero-shot transfer. A promising direction is to combine both paradigms: perform meta-learning within clusters of sufficiently similar households identified by our representation, potentially achieving both strong zero-shot initialization and rapid fine-tuning.

On a strongly mismatched environment (e_{10}), zero-shot Ours degrades (e.g., spring DR= 14.76%, COST= 237.43) and trails RBC. This



(a) Latent dimension (d_z).

(b) Similarity weighting factor (κ).

(c) Temporal window length (T^τ).

(d) Number of clusters (K_c).

Fig. 12. Sensitivity analysis of key hyperparameters.

exposes the sensitivity of neighbor retrieval to $\mathcal{D}$ mismatch: when physical mechanisms lie outside the training coverage, direct cloning may be brittle. Leveraging the Q2 similarity measure to guard retrieval, or adding small-scale fine-tuning, can mitigate this issue (the e_3 FT results already demonstrate the effect).

In terms of interaction cost, Ours and RBC/CQL all use N_{INT}=0; Ours (w/ FT) needs only (576) interactions to match or partially surpass SAC, whereas online DRL requires (24192). Considering the four HEMS objectives jointly (DR, COST, ULR, GES) and the interaction budget, Ours offers a practical route: "zero-shot meets baseline, light fine-tuning delivers strong performance," thereby answering **Q3**.

4.5. Parameter sensitivity (Q4)

Fig. 12 varies four hyperparameters—latent dimension d_z, temperature κ, slice length T^τ, and number of clusters K_c—with all else fixed. For the d_z sweep, because the K-means objective scales with dimensionality (cf. Eqs. (23)–(17)), we report a dimension-normalized compactness CMP/d_z to factor out this trivial dependence. We observe a rise–plateau–dip as d_z increases: performance improves from 32 to 128 and then flattens or slightly regresses beyond 256 (small d_z underfits; very large d_z begins to absorb $\mathcal{U}$-specific idiosyncrasies, hurting generalization). For κ, the curve shows a clear interior optimum around 0.2: very small κ yields overly sharp contrastive weights (myopic to only the closest positives), whereas very large κ over-smooths and blurs positive/negative structure; a mid-range temperature balances locality and discrimination. For T^τ, daily non-overlapping windows peak at T^τ=288: shorter slices miss diurnal HVAC/load cycles; much longer slices entangle daily physics with slower drift and also reduce the number of independent samples. For K_c, we examine the top-5 similarity scores with respect to e_1 across $K_c \in \{6, 12, 24, 48, 96\}$. When K_c is too small—especially below the number of households n—the cluster-frequency vectors $\mathbf{p}_e$ become coarse and many households share similar distributions, inflating pairwise similarity scores and degrading discriminability: at K_c=6, four of the five neighbors attain similarity ≈ 1, obscuring the ranking. Increasing K_c sharpens the distinction: at K_c=48, previously indistinguishable pairs (e.g., e_5 vs. e_{10}) become separable. However, when K_c is excessively large (e.g., K_c=96), frequency vectors grow sparse—most entries approach zero—so cosine similarity collapses and even genuinely similar households (e.g., e_2) exhibit reduced scores. The default K_c=24 (=2n) strikes a balance between expressiveness and stability; empirically, $K_c \in [n, 3n]$ provides robust neighbor retrieval. Overall, configurations that preserve $\mathcal{D}$—higher RHO with lower (normalized) compactness while keeping ACC competitive—also translate into stronger control metrics.

4.6. Robustness (Q5)

We evaluate robustness along two axes that commonly degrade HEMS performance in real-world deployment: (i) stochastic process noise during deployment, and (ii) seasonal distribution shift. Fig. 13 reports control metrics under Gaussian perturbation σ_v injected into exogenous variables at deployment time, while Fig. 14 quantifies cross-season embedding drift (Summer↔Winter) via DIST.

As σ_v increases, RB-ZeroHEM exhibits a noticeably flatter increase in both COST and DR (Fig. 13), with smaller slopes and narrower confidence intervals compared to CQL and transfer-based baselines. In Fig. 7, our representations maintain significantly lower DIST under

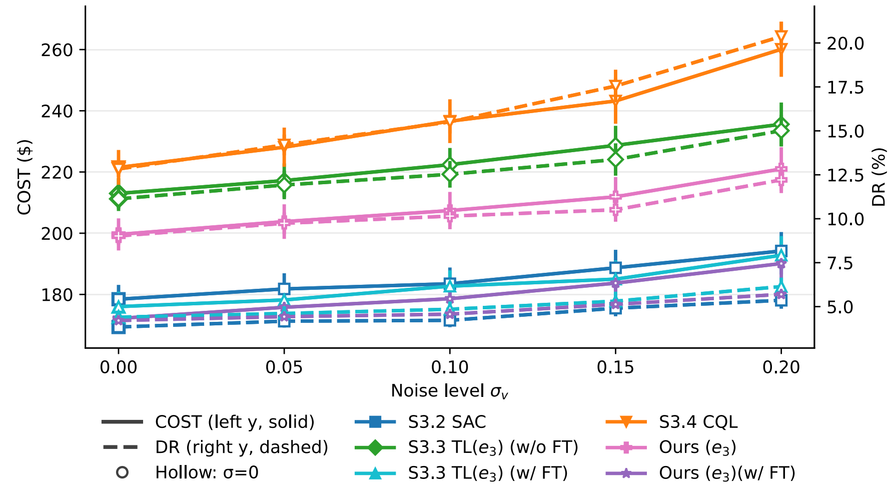


Fig. 13. Robustness to deployment noise σ_v. We report COST (solid, left y) and DR (dashed, right y); markers show mean with 95% CI.

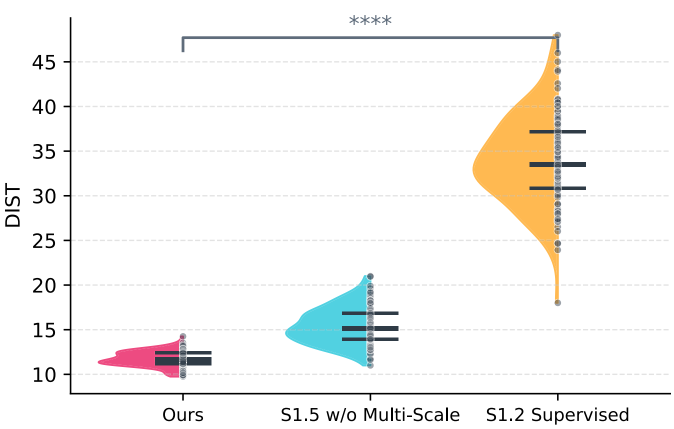


Fig. 14. Seasonal shift DIST (Summer ↔ Winter). Each dot represents the cross-seasonal average DIST of a (house, seed) pair. The violin plots illustrate the distribution density. The three horizontal bars within each violin correspond to the 25th percentile, median (thicker line), and 75th percentile. The asterisks denote the significance level (two-sided Mann–Whitney U test, $p < 0.0001$ (****)).

medium–high noise (Mann–Whitney U significance as shown) and more stable ACC/CMP. The four-season panels (Table 4) likewise show that RHO remains high across seasonal changes. Seasonal variation introduces coupled shifts in weather and occupancy patterns. As shown in Fig. 14, RB-ZeroHEM achieves the smallest Summer↔Winter DIST among competitors (****, $p<10^{-4}$), indicating strong season-invariant representations. This aligns with Table 4, where the physics alignment metric RHO remains consistently high from spring to winter while preserving competitive ACC. At the control level (Table 6), the zero-shot policy shows smaller seasonal variations in DR/COST/GES than RBC and offline CQL, and performs comparably to online DRL without incurring the N_{INT}=24,192 interaction cost.

These results demonstrate that the proposed method maintains strong robustness under both deployment noise and seasonal shifts, in terms of both representation stability and control performance, effectively addressing **Q5**. This robustness arises from learning slice-level invariants of the time-invariant dynamics D and using these invariants for both neighbor retrieval and policy cloning. When the retrieved similarity is trustworthy, zero-shot deployment suffices; when it is not, a brief, bounded fine-tuning step restores near-optimal performance.

4.7. *Limitations and discussion*

While RB-ZeroHEM demonstrates promising results in simulation, several practical constraints warrant careful consideration for real-world deployment.

4.7.1. *Computational efficiency*

Table 7 summarizes the computational requirements under our experimental setup. Stage 1 (contrastive representation learning) requires approximately 2.2 hours to process 90-day trajectories from 12 candidate households, while Stage 2 (clustering) and Stage 3 (BC) complete in seconds and minutes respectively. Although this offline training cost is incurred only once and amortizes across multiple target households, it may pose challenges for resource-constrained scenarios or when frequent model updates are needed to track evolving household characteristics. The total training time scales approximately linearly with the number of candidate households (n) and trajectory length, which could become prohibitive in community-scale applications involving hundreds or thousands of homes. Future work should explore more efficient representation learning architectures (e.g., lightweight temporal networks, knowledge distillation) or federated learning approaches to distribute the computational burden while preserving privacy.

4.7.2. *Data dependency and privacy constraints*

The method fundamentally relies on access to historical CTs from multiple source households, which raises several practical concerns. First, *regulatory compliance*: collecting and processing residential energy data must adhere to privacy regulations such as GDPR in Europe or CCPA in California [76], requiring explicit user consent and secure data handling protocols. Second, *user acceptance*: even with anonymization, homeowners may be reluctant to share fine-grained operational data (5-minute resolution in our experiments), particularly if trajectories reveal occupancy patterns or appliance usage habits. This “privacy-utility” tension could limit the availability of high-quality source data in practice [45]. Third, *economic feasibility*: deploying the infrastructure to collect, store, and process multi-household trajectories—including smart meters, communication networks, and central servers—entails upfront capital costs and ongoing maintenance expenses that may not be justified for individual households.

Integration with existing HEMS platforms (e.g., Google Nest, Ecobee) could mitigate this burden, but interoperability standards and vendor lock-in remain open challenges. Moreover, learning-based HEMS strategies are still in their early stages of development in real-world deployments, and the availability of high-quality, diverse CTs from production systems may be limited. As the field matures and more

Table 7
Computational requirements for RB-ZeroHEM training (12 households, 90-day window, Intel i7-14600K + RTX 4090).

Stage	Component	Training time	Parameters	Remarks
Stage 1	Contrastive Learning	2.2 ± 0.3 hours	$d_z = 128$, $\kappa = 0.2$, $T_\tau = 288$	1000 epochs, batch = 256
Stage 2	K-means Clustering	0.0077 ± 0.0006 s	$K_c = 24$	Per query
Stage 3	BC	17.5 ± 2.2 minutes	$T_{\text{demo}} = 25{,}920$ steps	MLP with 5 layers
Total	**Offline Training**	**≈2.3 hours**	–	One-time cost per update

households adopt intelligent energy management solutions, the trajectory dataset ecosystem will naturally expand, improving the method's applicability and generalization capacity.

4.7.3. Sensitivity to source-target mismatch

The effectiveness of RB-ZeroHEM fundamentally depends on the diversity and coverage of the source household database. When no sufficiently similar household exists in the source pool, or when the target household undergoes configuration changes (e.g., installing radiant floor heating), the learned representation may fail to identify appropriate neighbors. This creates a *mismatch* problem: the target household's true $\mathcal{D}$ no longer aligns with any cluster in the learned space, causing the similarity metric $\text{SIM}(e^*, e)$ to misidentify neighbors.

Our experiments (Table 6, Ours(e_{10})) demonstrate this effect: when the source household is physically mismatched, zero-shot performance degrades substantially (e.g., spring DR= 14.76%, COST= $237.43 compared to 8.24% and $193.28 for the well-matched e_3). This limitation is inherent to any similarity-based transfer approach: the method cannot extrapolate to physical regimes absent from training.

However, the similarity metric itself provides a deployment-time confidence indicator. In our experiments, well-matched pairs (e_1–e_2, e_1–e_3) yield SIM > 0.7, while the mismatched e_{10} shows SIM ≈ 0.55. A low maximum similarity score signals potential mismatch, enabling risk-aware fallback strategies: defaulting to conservative RBC, requesting minimal fine-tuning before deployment, or alerting operators to expand the source database. Additionally, periodic online monitoring of key performance indicators (DR, COST, ULR) could serve as indirect signals of drift, prompting model retraining or neighbor re-selection. Future work should explore active learning strategies to prioritize acquisition of trajectories from under-represented physical regimes, as well as continual learning frameworks that incrementally update representations as household configurations evolve.

5. Conclusion

This paper proposes RB-ZeroHEM, a zero-shot knowledge transfer framework that addresses the cold-start problem in DRL-based HEMS by leveraging contrastive representation learning on historical CTs to enable cross-household policy transfer without target-environment interaction. Our main contributions include: (i) a self-supervised contrastive learning module that extracts physics-aligned embeddings that are robust to seasonal shifts and deployment noise; (ii) a clustering–frequency hybrid distribution for reliable household similarity measurement; and (iii) a zero-interaction behavioral cloning strategy that achieves a 33% discomfort reduction, 15% cost savings, and 20% grid energy savings—competitive with online DRL baselines.

Experimental results validate that RB-ZeroHEM effectively bridges the deployment gap between train-from-scratch DRL and practical requirements. When transferring to similar households, the zero-shot policy matches or exceeds rule-based and offline RL baselines, trailing SAC by a margin that is nearly closed with minimal fine-tuning (2.4% of SAC's interaction budget). These results demonstrate that historical control trajectories—increasingly available through smart-meter and IoT deployments—constitute a valuable resource for community-scale energy management, enabling cross-household knowledge transfer rather than treating each home as an isolated learning problem.

The primary strengths of RB-ZeroHEM lie in its ability to deploy without any target–environment interaction and its independence from policy architecture alignment, making it readily applicable to heterogeneous residential settings. However, its effectiveness is inherently bounded by the diversity and coverage of the source household database, and zero-shot performance degrades when no sufficiently similar counterpart exists in the source pool (see Section 4.7 for a detailed discussion). Future work should address these practical limitations, including the computational efficiency for large-scale deployment, privacy-preserving trajectory sharing, and active learning strategies to expand the coverage of the source database.

CRediT authorship contribution statement

Xiao Du: Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Fengji Luo:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Juntao Hu:** Validation, Methodology, Investigation. **Wei Zhou:** Writing – review & editing, Validation, Supervision, Resources. **Junhao Wen:** Writing – review & editing, Validation, Supervision, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] T. Abergel, J. Dulac, I. Hamilton, M. Jordan, A. Pradeep, Global status report for buildings and construction—towards a zero-emissions, efficient and resilient buildings and construction sector, Env., Programme U. N. Env., Programme 390 (2019).
[2] International Energy Agency, Global Energy Review 2020, 2020. https://www.iea.org/reports/global-energy-review-2020., iEA, Paris. Licence: CC BY 4.0. https://doi.org/10.1787/a60abbf2-en.
[3] M. Kazemi, C. Papadimitriou, N.G. Paterakis, K. Kok, Cloud-based real-time model predictive control for a multi-carrier and multi-objective home energy management system, IEEE Trans. Ind. Appl. 61 (6) (2025) 8329–8344. https://doi.org/10.1109/TIA.2025.3576745.
[4] X. Lin, R. Zamora, Y. Jiang, G. Chen, A.K. Srivastava, A multi-level home energy management system (hems) for dc-microgrids, IEEE Trans. Sustain. Energy 16 (4) (2025) 2358–2373. https://doi.org/10.1109/TSTE.2025.3551682.
[5] B. Zhou, W. Li, K.W. Chan, Y. Cao, Y. Kuang, X. Liu, X. Wang, Smart home energy management systems: concept, configurations, and scheduling strategies, Renew. Sustain. Energy Rev. 61 (2016) 30–40.
[6] R.S. Sutton, A.G. Barto ed), Reinforcement Learning: an Introduction, second ed, MIT Press, 2018.
[7] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, X. Guan, A review of deep reinforcement learning for smart building energy management, IEEE Internet Things J. 8 (2021) 12046–12063.
[8] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold-start recommendations, in: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002, pp. 253–260.
[9] Z. Nagy, G. Henze, S. Dey, J. Arroyo, L. Helsen, X. Zhang, B. Chen, K. Amasyali, K. Kurte, A. Zamzam, et al., Ten questions concerning reinforcement learning for building energy management, Build. Environ. 241 (2023) 110435.
[10] F. Luo, W. Kong, G. Ranzi, Z.Y. Dong, Optimal home energy management system with demand charge tariff and appliance operational dependencies, IEEE Trans. Smart Grid 11 (2019) 4–14.

[11] F. Luo, G. Ranzi, C. Wan, Z. Xu, Z.Y. Dong, A multistage home energy management system with residential photovoltaic penetration, IEEE Trans. Ind. Informat. 15 (2018) 116–126.
[12] A.K. Candan, A.R. Boynuegri, N. Onat, Home energy management system for enhancing grid resiliency in post-disaster recovery period using electric vehicle, Sustain. Energy Grids Netw. 34 (2023) 101015.
[13] A. Anvari-Moghaddam, H. Monsef, A. Rahimi-Kian, Optimal smart home energy management considering energy saving and a comfortable lifestyle, IEEE Trans. Smart Grid 6 (2014) 324–332.
[14] T.D. de Lima, P. Faria, Z. Vale, Optimizing home energy management systems: a mixed integer linear programming model considering battery cycle degradation, Energy Build. 329 (2025) 115251.
[15] G. Bernasconi, S. Brofferio, L. Cristaldi, Cash flow prediction optimization using dynamic programming for a residential photovoltaic system with storage battery, Solar Energy 186 (2019) 233–246.
[16] E. Khanmirza, A. Esmaeilzadeh, A.H.D. Markazi, Predictive control of a building hybrid heating system for energy cost reduction, Appl. Soft Comput. 46 (2016) 407–423.
[17] D.Y. Yamashita, I. Vechiu, J.-P. Gaubert, Two-level hierarchical model predictive control with an optimised cost function for energy management in building microgrids, Appl. Energy 285 (2021) 116420.
[18] M.S. Javadi, M. Gough, M. Lotfi, A.E. Nezhad, S.F. Santos, J.P.S. Catalão, Optimal self-scheduling of home energy management system in the presence of photovoltaic power generation and batteries, Energy 210 (2020) 118568.
[19] F. Luo, G. Ranzi, Z.Y. Dong, Building Energy Management Systems and Techniques: Principles, Methods, and Modelling, Elsevier, 2024.
[20] S.H. Mansour, S.M. Azzam, H.M. Hasanien, M. Tostado-Véliz, A. Alkuhayli, F. Jurado, Deep reinforcement learning-based plug-in electric vehicle charging/discharging scheduling in a home energy management system, Energy 316 (2025) 134420.
[21] X. Chen, T. Wei, S. Hu, Uncertainty-aware household appliance scheduling considering dynamic electricity pricing in smart home, IEEE Trans. Smart Grid 4 (2013) 932–941.
[22] D. Liu, Z. Guo, F. Chen, X. Xue, S. Mei, Energy management of a residential consumer with uncertain renewable generation: a robust dual dynamic programming approach, in: 2020 39th Chinese Control Conference (CCC), IEEE, 2020, pp. 1490–1494.
[23] J. Yuan, X. Zeng, J. Zhou, J. Li, J. Lv, R. Chen, K. Chen, W. Yang, Y. Zhang, Data-driven real-time home energy management system based on adaptive dynamic programming, Electr. Power Syst. Res. 238 (2025) 111055.
[24] A.A. Amer, K. Shaban, A.M. Massoud, Drl-hems: deep reinforcement learning agent for demand response in home energy management systems considering customers and operators perspectives, IEEE Trans. Smart Grid 14 (2022) 239–250.
[25] Z. Qin, D. Liu, H. Hua, J. Cao, Privacy preserving load control of residential microgrid via deep reinforcement learning, IEEE Trans. Smart Grid 12 (2021) 4079–4089.
[26] L. Yu, W. Xie, D. Xie, Y. Zou, D. Zhang, Z. Sun, L. Zhang, Y. Zhang, T. Jiang, Deep reinforcement learning for smart home energy management, IEEE Internet Things J. 7 (2019) 2751–2762.
[27] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.
[28] H.N. Abishu, A.M. Seid, A. Erbad, S. Márquez-Sánchez, J.H. Fernandez, J.M. Corchado, Multi-agent drl-based demand response optimization for iot-based smart home energy management systems, IEEE Internet Things J. 12 (14) (2025) 27003–27020. https://doi.org/10.1109/JIOT.2025.3562137.
[29] X. Fang, G. Gong, G. Li, L. Chun, P. Peng, W. Li, X. Shi, Cross temporal-spatial transferability investigation of deep reinforcement learning control strategy in the building HVAC system level, Energy 263 (2023) 125679.
[30] X. Zhang, X. Jin, C. Tripp, D.J. Biagioni, P. Graf, H. Jiang, Transferable reinforcement learning for smart homes, in: Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities, RLEM'20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 43–47, https://doi.org/10.1145/3427773.3427865
[31] M. Khan, B.N. Silva, O. Khattab, B. Alothman, C. Joumaa, A transfer reinforcement learning framework for smart home energy management systems, IEEE Sens. J. 23 (2022) 4060–4068.
[32] D. Coraci, S. Brandi, T. Hong, A. Capozzoli, An innovative heterogeneous transfer learning framework to enhance the scalability of deep reinforcement learning controllers in buildings with integrated energy systems, in: Building Simulation, vol. 17, Springer, 2024, pp. 739–770.
[33] D. Coraci, S. Brandi, T. Hong, A. Capozzoli, Online transfer learning strategy for enhancing the scalability and deployment of deep reinforcement learning control in smart buildings, Appl. Energy 333 (2023) 120598.
[34] F. Hou, J.C.P. Cheng, H.H.L. Kwok, J. Ma, Multi-source transfer learning method for enhancing the deployment of deep reinforcement learning in multi-zone building HVAC control, Energy Build. 322 (2024) 114696.
[35] J. Lu, P. Mannion, K. Mason, A meta-learning approach for multi-objective reinforcement learning in sustainable home energy management, in: 27th European Conference on Artificial Intelligence, ECAI 2024, IOS Press BV, 2024, pp. 2814–2821.
[36] L. Xiong, Y. Tang, C. Liu, S. Mao, K. Meng, Z. Dong, F. Qian, Meta-reinforcement learning-based transferable scheduling strategy for energy management, IEEE Trans. Circuits Syst. I Regul. Pap. 70 (2023) 1685–1695.
[37] H. Zhang, D. Wu, B. Boulet, Metaems: a meta reinforcement learning-based control framework for building energy management system, arXiv preprint arXiv:2210.12590, 2022.
[38] J. Li, T. Zhou, Evolutionary multi-agent deep meta reinforcement learning method for swarm intelligence energy management of isolated multi-area microgrid with internet of things, IEEE Internet Things J. 10 (2023) 12923–12937.
[39] K. Nweye, S. Sankaranarayanan, Z. Nagy, Merlin: multi-agent offline and transfer learning for occupant-centric operation of grid-interactive communities, Appl. Energy 346 (2023) 121323.
[40] L. Xiong, Y. Tang, K. Bhattacharya, M.-Y. Chow, F. Qian, Offline deep reinforcement learning-based home energy management systems with heterogeneous EV charging load models, IEEE Trans. Circuits Syst. I Regul. Pap. (2025) 1–11. https://doi.org/10.1109/TCSI.2025.3600455
[41] F. Pourpanah, M. Abdar, Y. Luo, X. Zhou, R. Wang, C.P. Lim, X.-Z. Wang, Q.M.J. Wu, A review of generalized zero-shot learning methods, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2022) 4051–4070.
[42] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (2020) 43–76.
[43] G. Pinto, Z. Wang, A. Roy, T. Hong, A. Capozzoli, Transfer learning for smart buildings: a critical review of algorithms, applications, and future perspectives, Adv. Appl. Energy 5 (2022) 100084.
[44] O. Alibrahim, S. Padmanaban, M. Khan, O. Khattab, B. Alothman, C. Joumaa, Deep transfer learning-enabled energy management strategy for smart home sensor networks, IEEE Trans. Ind. Appl. 59 (2022) 81–92.
[45] Z. Wang, P. Yu, H. Zhang, Privacy-preserving regulation capacity evaluation for HVAC systems in heterogeneous buildings based on federated learning and transfer learning, IEEE Trans. Smart Grid 14 (2022) 3535–3549.
[46] R. Huang, H. He, Q. Su, M. Härtl, M. Jaensch, Enabling cross-type full-knowledge transferable energy management for hybrid electric vehicles via deep transfer reinforcement learning, Energy 305 (2024) 132394.
[47] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.
[48] S. Levine, A. Kumar, G. Tucker, J. Fu, Offline reinforcement learning: tutorial, review, and perspectives on open problems, arXiv preprint arXiv:2005.01643, 2020.
[49] Y. Deng, F. Luo, Y. Zhang, Y. Mu, An efficient energy management framework for residential communities based on demand pattern clustering, Appl. Energy 347 (2023) 121408.
[50] L.P. Kaelbling, M.L. Littman, A.R. Cassandra, Planning and acting in partially observable stochastic domains, Artif. Intell. 101 (1998) 99–134.
[51] N. Heess, D. Tb, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S.M. Eslami, et al., Emergence of locomotion behaviours in rich environments, arXiv preprint arXiv:1707.02286, 2017.
[52] S.S. Shuvo, Y. Yilmaz, Home Energy recommendation System (Hers): a deep reinforcement learning method based on residents' feedback and activity, IEEE Trans. Smart Grid 13 (2022) 2812–2821.
[53] E. McKenna, M. Thomson, High-resolution stochastic integrated thermal–electrical domestic demand model, Appl. Energy 165 (2016) 445–461.
[54] I. Richardson, M. Thomson, D. Infield, C. Clifford, Domestic electricity use: a high-resolution energy demand model, Energy Build. 42 (2010) 1878–1887.
[55] M.A. López-Rodríguez, I. Santiago, D. Trillo-Montero, J. Torriti, A. Moreno-Munoz, Analysis and modeling of active occupancy of the residential sector in Spain: an indicator of residential electricity consumption, Energy Policy 62 (2013) 742–751.
[56] Pecan Street Inc, Pecan Street database, 2020. http://www.pecanstreet.org/. Accessed: Jun. 2024.
[57] National Oceanic and Atmospheric Administration (NOAA), NOAA data, Online. https://www.ncdc.noaa.gov/, 2020, Accessed: Jun. 2024.
[58] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PmLR, 2020, pp. 1597–1607.
[59] X. Chen, H. Fan, R. Girshick, K. He, Improved baselines with momentum contrastive learning, arXiv preprint arXiv:2003.04297, 2020.
[60] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748, 2018.
[61] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[62] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015.
[63] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967, pp. 281–297.
[64] D.A. Pomerleau, Efficient training of artificial neural networks for autonomous navigation, Neural Comput. 3 (1991) 88–97.
[65] D.J. Foster, A. Block, D. Misra, Is behavior cloning all you need? Understanding horizon in imitation learning, in: Advances in Neural Information Processing Systems, vol. 37, 2024.
[66] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774, 2023.

[67] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International Conference on Machine Learning, Pmlr, 2018, pp. 1861–1870.
[68] F. De Angelis, M. Boaro, D. Fuselli, S. Squartini, F. Piazza, Q. Wei, Optimal home energy management under dynamic electrical and thermal constraints, IEEE Trans. Ind. Informat. 9 (2012) 1518–1527.
[69] L. Jin, D. Lee, A. Sim, S. Borgeson, K. Wu, C.A. Spurlock, A. Todd, Comparison of clustering techniques for residential energy behavior using smart meter data, in: AAAI Workshops, 2017.
[70] D. Vercamer, B. Steurtewagen, D. Van den Poel, F. Vermeulen, Predicting consumer load profiles using commercial and open data, IEEE Trans. Power Syst. 31 (2015) 3693–3701.
[71] L. McInnes, J. Healy, S. Astels, et al., Hdbscan: hierarchical density based clustering, J. Open Source Softw. 2 (2017) 205.
[72] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347, 2017.
[73] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: International Conference on Machine Learning, PMLR, 2018, pp. 1587–1596.
[74] A. Kumar, A. Zhou, G. Tucker, S. Levine, Conservative q-learning for offline reinforcement learning, Adv. Neural Inf. Process. Syst. 33 (2020) 1179–1191.
[75] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[76] Y. Zhang, Y. Qu, L. Gao, T.H. Luan, A. Jolfaei, J.X. Zheng, Privacy-preserving data analytics for smart decision-making energy systems in sustainable smart community, Sustain. Energy Technol. Assess. 57 (2023) 103144.