# Transferable Reinforcement Learning for Smart Homes

## Preprint

Xiangyu Zhang, Xin Jin, Charles Tripp, David J. Biagioni, Peter Graf, and Huaiguang Jiang

*National Renewable Energy Laboratory*

*Presented at First International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities (RLEM)*
*November 17, 2020*

**NREL is a national laboratory of the U.S. Department of Energy**
**Office of Energy Efficiency & Renewable Energy**
**Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

**Conference Paper**
NREL/CP-2C00-77933
November 2020

# Transferable Reinforcement Learning for Smart Homes

## Preprint

Xiangyu Zhang, Xin Jin, Charles Tripp, David J. Biagioni, Peter Graf, and Huaiguang Jiang

*National Renewable Energy Laboratory*

**Suggested Citation**
Zhang, Xiangyu, Xin Jin, Charles Tripp, David J. Biagioni, Peter Graf, and Huaiguang Jiang. 2020. *Transferable Reinforcement Learning for Smart Homes: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-2C00-77933. https://www.nrel.gov/docs/fy21osti/77933.pdf.

**NREL is a national laboratory of the U.S. Department of Energy**
**Office of Energy Efficiency & Renewable Energy**
**Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

**Conference Paper**
NREL/CP-2C00-77933
November 2020

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

## NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.



*Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.*

NREL prints on paper that contains recycled content.

# Transferable Reinforcement Learning for Smart Homes

Xiangyu Zhang
Xiangyu.Zhang@nrel.gov
National Renewable Energy Laboratory
Golden, Colorado

Xin Jin
Xin.Jin@nrel.gov
National Renewable Energy Laboratory
Golden, Colorado

Charles Tripp
Charles.Tripp@nrel.gov
National Renewable Energy Laboratory
Golden, Colorado

David J. Biagioni
Dave.Biagioni@nrel.gov
National Renewable Energy Laboratory
Golden, CO

Peter Graf
Peter.Graf@nrel.gov
National Renewable Energy Laboratory
Golden, Colorado

Huaiguang Jiang
Huaiguang.Jiang@nrel.gov
National Renewable Energy Laboratory
Golden, Colorado

## ABSTRACT

To harness the great amount of untapped resources on the demand side, smart home technology plays a vital role in solving the "last mile" problem in smart grid. Reinforcement learning (RL), which has demonstrated an outstanding performance in solving many sequential decision-making problems, can be a great candidate to be used in smart home control. For instance, many studies have started investigating the appliance scheduling problem under dynamic pricing scheme. Based on those, this study aims at providing an affordable solution to encourage a higher smart home adoption rate. Specifically, we investigate combining transfer learning (TL) with RL to reduce the training cost of an optimal RL control policy. Given an optimal policy for a benchmark home, TL can jump-start the RL training of a policy for a new home, which has different appliances and user preferences. Simulation results show that by leveraging TL, RL training converges faster and requires much less computing time for new homes that are similar to the benchmark home. In all, this study proposes a cost-effective approach for training RL control policies for homes at scale, which ultimately reduces the controller's implementation costs, increases the adoption rate of RL controllers, and makes more homes grid-interactive.

## CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; • **Hardware** → **Smart grid**.

## 1 INTRODUCTION

The intelligent home energy management system (HEMS) is an important component of the modern energy system because it can not only provide benefits to homeowners but also help to form clusters of dispatchable load, which can be leveraged during gird critical times to improve system reliability [14]. To develop automated HEMS, control algorithms based on optimization have gained popularity: Du *et al.* propose a controller for scheduling thermostatically controlled home appliance based on price to minimize homeowner's payment [3]; Rastegar *et al.* propose a similar load commitment controller and consider responsive appliances, storage system and electric vehicle [6] and there are many other related studies including [10, 11]. However, in real-life applications, these approaches might suffer from high real-time computation complexity (e.g., require solving mixed integer programming (MIP) problems on-the-fly) and expensive implementation (e.g., require powerful computers for real-time optimization and costly commercial solvers) [7]. To overcome these drawbacks, reinforcement learning (RL) has been recently proposed as an alternative for optimal scheduling of home appliances [4, 12]. Due to RL's off-line training capability, an RL agent can learn an optimal control policy based on home appliance models and occupant preferences beforehand. During on-line control, the only computation needed is the forward evaluation of the policy network, which can be implemented by low-cost embedded systems or at cloud. As a result, homeowners can achieve optimal home energy management with little or none hardware cost and an RL policy training cost.

In this study, we envision that there exists a cloud service provider (CSP), who helps homeowners train their smart home control policies by leveraging cost-effective cloud computing resources in exchange for a service fee, as proposed in [13]. To further reduce RL training costs, which account for a major portion of the service fee, we investigate using *transfer learning* (TL) and determine if TL can accelerate control policy training in HEMS applications. The insight behind TL is that generalization can occur across related and different tasks [9]. Considering the similarity among homes (the appliance types, numbers, and occupant preferences are similar), the knowledge to optimally control Home A can be used to jump-start the training of an optimal policy for Home B (which is similar to Home A). Thus, instead of training thousands of individual RL control policies for different homes from scratch, the

CSP can accelerate the training by using TL and existing smart home control policies. Reducing training costs will make RL-based HEMS more affordable and thus encourages a larger portfolio of grid-interactive residential buildings. To the best of our knowledge, no previous work has investigated "TL+RL" in smart home/building control. In this paper we study this combination, and demonstrate preliminary results for the efficacy of the proposed idea.

## 2 HEMS CONTROL PROBLEM FORMULATION

Before delving in the RL controller, in this section, the optimal appliance scheduling problem in the HEMS is formulated, conducting cost minimization over a period $\mathcal{T} = [1, 2, ..., T]$ with consideration of real-time electricity price (RTP) $c_t$. Specifically, four different types of common home appliance models ($\mathcal{D} = \{\alpha, \beta, \omega, \theta\}$) [10] used in this paper are shown below:

1) Schedule-based interruptible load (SA-IL) can be operated and interrupted at any time, the goal is to schedule it over $\mathcal{T}$ to meet the users' requirement ($D^\alpha$). Mathematically, $x_t^\alpha \in \{0, 1\}$ is determined for $t \in \mathcal{T}$ and failure to finish the required amount of work $D^\alpha$ within $\mathcal{T}$ will cause a linear user discomfort cost $C^\alpha = \rho^\alpha(D^\alpha - \sum_{t\in\mathcal{T}} x_t^\alpha)$, in which $\rho^\alpha$ is a user-defined unit penalty. In addition, total SA-IL operation time over $\mathcal{T}$ is limited by:

$$\sum_{t\in\mathcal{T}} x_t^\alpha \le D^\alpha. \tag{1}$$

2) Schedule-based uninterruptible load (SA-UL) needs to be scheduled to run at certain steps $t \in \mathcal{T}$, however, as soon as it starts, it cannot be interrupted until the operation is completed. We use $x_t^\beta \in \{0, 1\}$ to represents SA-UL's status and $s_t^\beta \in \{0, 1\}$ for its starting signal. Similarly, user discomfort occurs when a task cannot be finished on time: $C^\beta = \rho^\beta(D^\beta - \sum_{t\in\mathcal{T}} x_t^\beta)$, in which $\rho^\beta$ is a user-defined unit penalty. Additionally, SA-UL operation is constrained by:

$$\sum_{t\in\mathcal{T}} x_t^\beta \le D^\beta, \quad 0 \le \sum_{t\in\mathcal{T}} s_t^\beta \le 1, \tag{2}$$

$$0 \le \sum_{i\in[t,t+D^\beta-1]} x_i^\beta - s_t^\beta D^\beta \quad (\forall t \in [0, T - D^\beta + 1]), \tag{3}$$

$$0 \le \sum_{i\in[0,t-1]} x_i^\beta - s_t^\beta D^\beta \le D^\beta \quad (\forall t \in [T - D^\beta + 1, T - 1]), \tag{4}$$

$$0 \le \sum_{i\in[t,T-1]} x_i^\beta - s_t^\beta (T - t) \le D^\beta \ (\forall t \in [T - D^\beta + 1, T - 1]). \tag{5}$$

where (3)-(5) enforce the uninterruptability of SA-UL's operation.

3) Thermostatically controlled load (TCL) are in general temperature regulating load (e.g., air-conditioner). Typically, the control involves a thermal dynamics: $J_{t+1} = \mathcal{F}(J_t, x_t^\omega)$, in which $J_t \in \mathbb{R}$, $x_t^\omega \in \{0, 1\}$ and $\mathcal{F}$ represent the temperature, control variable and the thermal dynamics model. Single step thermal discomfort is represented by $C_t^\omega = \rho^\omega[(J_t - \overline{J})^+ + (\underline{J} - J_t)^+]^2$, in which $x^+ = max(0, x)$ and $[\underline{J}, \overline{J}]$ depicts the temperature comfort band. An operation requirement, i.e., the AC unit needs to remain turned OFF for two consecutive control intervals before it can be turned ON again, is considered with (6), where $M$ is a large positive number ('big-M').

$$x_{t-1}^\omega + x_{t+1}^\omega \le 1 + M \cdot x_t^\omega \tag{6}$$

4) Storage stores energy when $c_t$ is low and uses/sells it when $c_t$ increases. Control variables $u_t^\theta \in \{0, 1\}$ and $v_t^\theta \in \{0, 1\}$ represent whether to charge or discharge respectively. For a given time, there is $u_t^\theta + v_t^\theta \in \{0, 1\}$ and $E_{t+1} = \mathcal{G}(E_t, u_t^\theta, v_t^\theta)$, in which $E_t \in [\underline{E}, \overline{E}]$ is the energy stored and $\mathcal{G}$ is the battery dynamics. To prevent depleting energy storage, we assume the cost of $C^\theta = \rho^\theta(0.8E_0 - E_T)^+$, which encourages the controller to bring back the storage to at least 80% of its original value.

For simplicity, in this study, we consider one appliance for each type. As a result, mathematically, the optimal appliance scheduling problem can be formulated by:

$$\begin{aligned} \underset{\mathbf{x}_t}{\text{minimize}} \quad & \sum_{i\in\mathcal{D}} C^i + \sum_{t\in\mathcal{T}} c_t \cdot \mathbf{P}^\top \mathbf{x}_t \cdot \Delta t \\ \text{subject to} \quad & (1) - (6) \\ & \textit{Additional constraints.} \end{aligned} \tag{7}$$

where $\mathbf{P} = [P^\alpha, P^\beta, P^\omega, P_{ch}^\theta, -P_{dis}^\theta]^\top$ shows the power consumption of different appliances, $\mathbf{x}_t = [x_t^\alpha, x_t^\beta, x_t^\omega, u_t^\theta, v_t^\theta]^\top$ is the control signal and $\Delta t$ represents the control interval. The objective function in (7) considers both the cost related to failure of task completion/violation of user's preference and the total electricity cost. Additional constraints are those un-numbered constraints mentioned in the description above (e.g., equality constraints involve thermal dynamics model ($\mathcal{F}$) and battery model($\mathcal{G}$)), which are omitted here for space.

In some studies, RTP $c_t$ are precisely given for the whole horizon $\mathcal{T}$, which makes the problem deterministic. To be more realistic, this paper considers perfect but limited-horizon information about $c_t$. Specifically, we assume only the $c_t$ for the next four hours from the current step are perfectly known and beyond that period, a day-ahead estimation of RTP, namely, the day-ahead price (DAP), will be used to assist the decision-making.

## 3 HEMS RL CONTROLLER

In this section, how to use RL to solve the above-mentioned optimal sequential control problem is presented.

### 3.1 Markov Decision Process Formulation

The optimal control problem (7) can be formulated in a Markov decision process (MDP) in form of a quintuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$. The state transition probability $\mathcal{P}$ is implicitly defined in the RL simulator. Discounted factor $\gamma = 1.0$ is used because the problem is episodic. We will discuss $\mathcal{A}$, $\mathcal{S}$ and $\mathcal{R}$ in detail next.

Action space $\mathcal{A}$ includes all legal actions the RL agent can take. Here, there is $\mathbf{a}_t = [x_t^\alpha, x_t^\beta, x_t^\omega, x_t^\theta] \in \mathcal{A}$. In (7), $u_t^\theta$ and $v_t^\theta$ are used to handle different charging/discharging power while keep the problem linear; in contrast, a single variable $x_t^\theta \in \{0, 1, -1\}$ is used in RL formulation for the three status of a battery since linearity is not necessary. As a result, considering all possible values of $x_t^i (i \in \mathcal{D})$, there are in total 24 discrete actions in $\mathcal{A}$ (Cartesian product

of the four control variables. E.g., $\mathbf{a}_t = [1, 0, 1, 2]$). Apparently, $\mathcal{A}$ grows exponentially with the number of appliances, also known as the *curse of dimensionality*, which is another reason to investigate techniques that can accelerate learning.

State $\mathbf{s}_t \in \mathcal{S}$ represents the information that the RL agent needs for decision making. In this study, the state structure is defined in (8), with the first five elements represent the current task completion percentage of SA-IL, whether SA-UL has started, normalized TCL temperature, battery current storage and energy shortage towards 80% of initial energy stored. $\mathbf{P}_t \in \mathbb{R}^L$ and $L$ are the RTP and number of control steps in the next 4-hours, respectively. Finally, $\mathbf{M}_t \in \mathbb{R}^{24}$ is an action mask indicating which actions are unavailable at step $t$ (action availability is influenced by constraints such as (1) and (6)).

$$\mathbf{s}_t = \left[ \frac{\sum_{t' \in [0,t]} x^{\alpha}_{t'}}{D^{\alpha}}, \sum_{t' \in [0,t]} s^{\beta}_{t'}, \hat{J}_t, \frac{E_t}{E_{max}}, \frac{0.8E_0 - E_t}{E_{max}}, \mathbf{P}_t, \mathbf{M}_t \right] \quad (8)$$

Reward structure $\mathcal{R}$ defines how RL agent's action is rewarded or penalized. Here we align $\mathcal{R}$ with the control objective illustrated in (7). Single step reward is defined by:

$$r_t = \begin{cases} -C^{\omega}_t - \mathcal{E}_t & (t \neq T) \\ -C^{\omega}_t - \mathcal{E}_t - \sum_{i \in \mathcal{D}, i \neq \omega} C^i & (t = T) \end{cases} \quad (9)$$

where $\mathcal{E}_t = c_t \cdot \mathbf{P}^{\top} \mathbf{x}_t \cdot \Delta t$ is the step-wise energy cost.

## 3.2 RL Controller Training

Based on the MDP formulation, an OpenAI Gym environment [1] is developed for the RL controller training. In this study, the proximial policy optimization (PPO) algorithm is used for the control policy training. PPO is a policy-based deep RL algorithm, it employs a neural network to implement a parameterized policy $\pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s})$ that maximizes a performance metrics $J(\boldsymbol{\theta})$. Typically, the policy parameter $\boldsymbol{\theta}_0$ is randomly initialized and then at each training iteration, PPO agent will estimate the gradient $\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ using collected trial-and-error experiences, and $\boldsymbol{\theta}$ is updated by:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}), \quad (10)$$

until the optimal parameter $\boldsymbol{\theta}_*$ is converged to. Interested readers should refer to [8] for more details and the PPO implementation used in this study is based on [5].

It is worth noting that there are some constraints (e.g., (1) and (6)) that need to be addressed in our RL problem. Namely, in some states, some actions are not available. There are two ways to enforce these constraints: *strong enforcement* and *weak enforcement*. Strong enforcement is handled by the RL agent, which uses a more complicated neural network structure and integrate the action availability mask $\mathbf{M}_t$ into the final layer of the network to force the sample probability of unavailable actions to be zero. Weak enforcement is handled by the simulator: the RL agent can output any action (e.g., $\mathbf{a}_t = [1, 1, 0, 2]$), however, the simulator will apply a sanity check, transforming invalid actions into a valid ones. If, for instance, constraint (1) is bounded (means $x^{\alpha}_t$ can only be 0), then the simulator will modify the above action to be $\mathbf{a}_t = [0, 1, 0, 2]$. By implementing and comparing these two approaches, we found that weak enforcement is more suitable for the smart home control applications for two reasons. First, the majority of control variables in our problem setting are binary, which makes weak enforcement easy to implement and logically sound (e.g., since $x \in \{0, 1\}$, the simulator can directly choose $x = 1$ if $x = 0$ violates operation constraints, or the other way around). Second, the masking process in neural networks introduces strong non-linear complexity during training, significantly increasing the training time needed to converge, making strong enforcement overkill for this application. In all, soft enforcement is used in later experiments.

## 3.3 Transfer Learning

In contrast to the typical start of RL, where a control policy is learned from scratch (i.e., randomly initialized policy parameter $\boldsymbol{\theta}$), TL focuses on starting learning from existing knowledge. In this study, specifically, given an RL controller $\pi^{\mathcal{H}_0}_*(\mathbf{a}|\mathbf{s})$, which is able to optimally solve (7) for home $\mathcal{H}_0$ (i.e., source task), how to leverage this known policy to optimally control a different home $\mathcal{H}$ (i.e., target task) is what TL tackles.

For two homes, $\mathcal{H}_0$ and $\mathcal{H}$, if they have the same types and number of appliances to be controlled but possibly with different parameters or user preferences, the problem spaces (i.e., $\mathcal{S}$ and $\mathcal{A}$) of the source and target tasks are of the same dimension. As a result, the neural networks representing the control policies ($\pi^{\mathcal{H}_0}$ and $\pi^{\mathcal{H}}$) share the same structure (i.e., input and output dimension). In this case, by directly copying the policy parameters (i.e., $\boldsymbol{\theta}^{\mathcal{H}}_0 \leftarrow \boldsymbol{\theta}^{\mathcal{H}_0}_*$), the control behavior/knowledge is transferred to the new controller. Considering the similarity between $\mathcal{H}_0$ and $\mathcal{H}$, the optimal policies for them should also not be too different. Therefore, this weight-copying warm-start can effectively bring $\boldsymbol{\theta}^{\mathcal{H}}_0$ to the vicinity of $\boldsymbol{\theta}^{\mathcal{H}}_*$ and thus potentially accelerate the learning process. Admittedly, due to the difference between $\mathcal{H}_0$ and $\mathcal{H}$, the new controller needs to be further refined using RL.

Note that for actor-critic algorithms like PPO, both policy and value networks will be copied to warm-start the new controller's training.

# 4 EXPERIMENTS AND RESULTS

## 4.1 Experiment Setup

The electricity price signal $c_t$ used in this study are from ComEd, a utility company in the U.S. [2]. A simulated home $\mathcal{H}_0$ is developed based on models introduced in Section 2 and appliance parameters shown in Table 1 ($\mathbf{P}^{\beta}_0$ is a load profile representing SA-UL's power consumption pattern). In addition, the control horizon considered is from 08:00 to 20:00, and we assume occupants are at work/school during this period so their behavior can be ignored in the model for simplicity. The control interval $\Delta t$ is 5 minutes. In the rest of this section, we briefly discuss the efficacy of such an RL controller by comparing it with a baseline controller; and then we focus on experimental examination of transfer learning to see whether it can accelerate training for various target tasks.

## 4.2 RL Controller's Efficacy

To show the efficacy of the RL controller, its control performance is compared with that of a model predictive controller (MPC), which solves (7) repeatedly at each control interval. To train the RL controller, RTP from July (2019) is used and the controller is tested

**Table 1: Appliances Parameters used for Home $\mathcal{H}_0$**

| Appliances | Parameters |
|---|---|
| SA-IL | $D^\alpha = 72$, $\rho^\alpha = 10.0$, $P^\alpha = 10.0$ |
| SA-UL | $\rho^\beta = 5.0$, $\mathbf{P}^\beta = \mathbf{P}_0^\beta$ |
| TCL | $P^\omega = 4.5$, $\rho^\omega = 1.0$, $\underline{J} = 73.0$, $\overline{J} = 76.0$ |
| STORAGE | $P_{ch} = 4.5$, $P_{dis} = 5.5$, $\eta_{ch} = 0.95$, $\eta_{dis} = 0.9$, $\rho^\theta = 20.0$, $\underline{E} = 1.5$, $\overline{E} = 15.0$ |

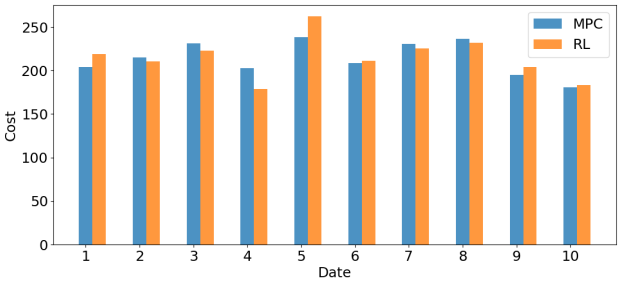


**Figure 1: Total cost comparison for ten test cases.**

using first ten day's RTP in August. Figure 1 shows the comparison of total cost (i.e., objective function value in (7)) for 10 testing days between the RL controller and baseline MPC. For these 10 days, the average daily total cost when using RL controller is 214.93 cents while the value is 214.30 cents for MPC. Therefore, in this experiment, we observed a equivalent control performance between RL controller and MPC. Specifically, considering that MPC uses DAP instead when RTP is unavailable (beyond four hours ahead, see Section 2), its performance depends on the difference between DAP and the actual RTP: RL performs better when DAP is an inaccurate estimation of RTP, which leads MPC to sub-optimal control.

## 4.3 Transfer Learning Experiments

To examine the TL effectiveness, in this section, several homes similar to $\mathcal{H}_0$ are created by perturbing the parameters used in $\mathcal{H}_0$. Here, $\mathcal{H}_0$ can be considered as a benchmark home that CSP has and other created homes are new ones that need trained RL control policies. Specifically, these new homes have the same appliance types and numbers, but are different in user preference or/and appliances parameters. Table 2 shows four new homes: Compared with $\mathcal{H}_0$, Cases 11 and 12 show homes ($\mathcal{H}_{11}$ and $\mathcal{H}_{12}$) with different user preferences and Cases 21 and 22 show homes ($\mathcal{H}_{21}$ and $\mathcal{H}_{22}$) with different appliances parameters. In the following experiments, we compare using RL to directly train a controller for these new homes with using TL to jump-start the RL training. To implement TL, weights of policy network $\pi_*^{\mathcal{H}_0}(\mathbf{a}|\mathbf{s})$ are used as initial values for the new policies to be trained, and a comparison between "TL+RL" and pure RL (with randomly initialized weights) is presented in Figure 2 for the above-mentioned cases. Using PPO, the RL policy is synchronously updated by 35 parallel RL learners. We used a two-layer fully connected neural network as the policy network with 256 hidden neurons in each layers and a learning rate of 5e-6.

**Table 2: New Homes Similar to $\mathcal{H}_0$**

| Home | Differences $< \mathcal{H}', \mathcal{H}_0 >$ |
|---|---|
| $\mathcal{H}_{11}$ | $\rho^\alpha = 10.0 \rightarrow \rho^\alpha = 0.5$ |
| $\mathcal{H}_{12}$ | $\rho^\omega = 1.0 \rightarrow \rho^\omega = 0.5$ and $J \in [73, 76] \rightarrow J \in [70, 73]$ |
| $\mathcal{H}_{21}$ | $P^\alpha = 10.0 \rightarrow P^\alpha = 6.0$ and $P^\omega = 4.5 \rightarrow P^\omega = 7.5$. |
| $\mathcal{H}_{22}$ | $\mathbf{P}^\beta = \mathbf{P}_0^\beta \rightarrow \mathbf{P}^\beta = \mathbf{P}_1^\beta$ (a different load profile), $P_{ch} = 4.5 \rightarrow P_{ch} = 6.0$ and $P_{dis} = 5.5 \rightarrow P_{dis} = 8.5$ |

**Table 3: TL Performance Metrics**

| Case | 11 | 12 | 21 | 22 | 31 | 32 |
|---|---|---|---|---|---|---|
| $\Delta_{init}$ | 181.05 | -48.79 | 211.8 | 253.45 | 0.84 | -13.77 |
| $\Delta C$ | 22.85 | 29.43 | 29.78 | 30.12 | 33.08 | 6.44 |
| $T_{th}$ | 30.6% | 2.2% | 0.0% | 0.0% | 2.8% | 86.2% |

To quantify the effectiveness of "TL+RL" (TL for simplicity hereinafter), three metrics introduced in [9] are used: *Jumpstart* ($\Delta_{init}$) shows the initial performance advantage given by using TL; *asymptotic performance* ($\Delta C$) shows the difference between RL and TL reward values by the end of training and *time to threshold* ($T_{th}$) is the time it takes for TL to reach the final reward value of RL. Table 3 shows these metrics for the four cases. According to Figure 2 and Table 3, three observations are made: 1) TL provides faster convergence, with three out of four test cases converge shortly after training was started. Additionally, TL converges within two-hour training sessions for all four cases, while RL doesn't (even in Case 12 where there is $\Delta_{init} < 0$). 2) with $\Delta C > 0$ for all cases, it means that after the training session, TL obtains better control policies that yield lower daily costs. 3) TL is more effective in Case 21 & 22 when compared with Case 11 & 12. This is because changes in appliances parameters can lead to a different converged value (i.e., different daily cost), but the optimal policy can remain the same (as shown by the near-flat learning curves in Case 21 & 22). In contrast, changes in user preferences have direct impact on the reward values and *might* change the coordinating relationship among appliances, and thus need to move to a different optimal policy. We also observed that in Case 11, if $\rho^\alpha$ is changed to 8.0 or 4.0, TL still converge fast (similar to Case 12 in Figure 2) and only when $\rho^\alpha$ is reduce below a threshold will the training time increases (to $T_{th} = 30.6\%$).

Next, we further increase the gap between the new home and $\mathcal{H}_0$ by creating Case 31 and 32: Home $\mathcal{H}_{31}$ contains all the differences in Cases 12, 21 and 22 and $\mathcal{H}_{32}$ contains differences in all four above-mentioned cases. Figure 3 shows the learning curves and TL metrics are also presented in Table 3. According to these results, in Case 32 it appears that the advantage of using TL is largely reduced when compared with other cases (TL only saves 14% of training time while in other cases time saving can be more than 70%, according to $T_{th}$). This is due to a larger difference of $< \mathcal{H}_{32}, \mathcal{H}_0 >$, which makes $\pi_*^{\mathcal{H}_{32}}$ much different from $\pi_*^{\mathcal{H}_0}$ and thus jump-starting using weights from $\pi_*^{\mathcal{H}_0}$ does not benefit as much. In contrast, the difference $< \mathcal{H}_{31}, \mathcal{H}_0 >$ is small enough that TL is still beneficial.

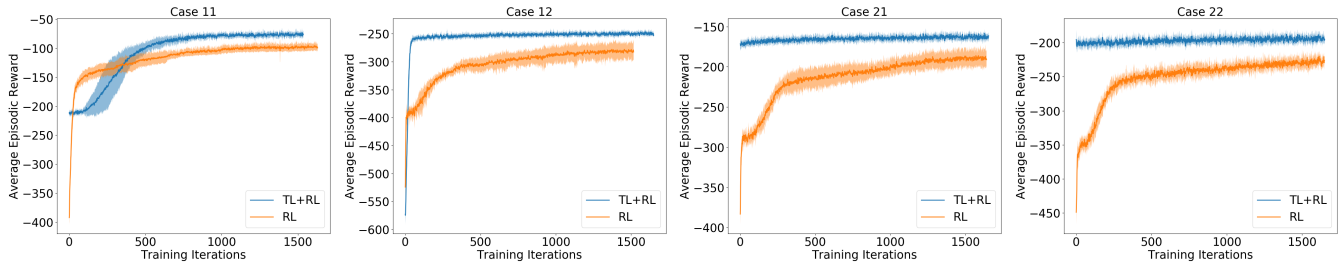


**Figure 2: Learning curves comparison for Case 11, 12, 21, 22. The X-axis shows the training iteration (each iteration consists of 7,000 training steps) and the Y-axis indicates the average episodic reward (negative daily cost in cents). Orange curves represent RL learning progress and blue curves show the "TL+RL" learning progress. Learning curves are averaged from 10 separate trials and the shaded area represents the standard deviation. Each training trial was 2 hours in duration.**

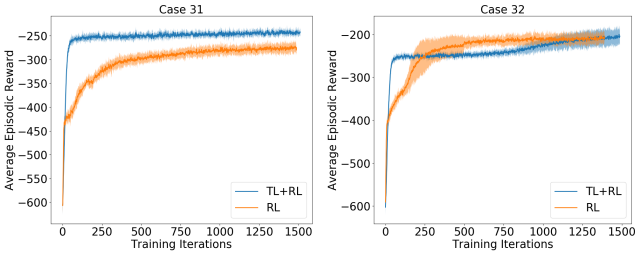


**Figure 3: Learning curve for Case 31 and 32.**

## 5 CONCLUSION AND FUTURE WORKS

In this paper, we conduct some preliminary experiments for transferring the home control knowledge from an existing RL controller to accelerate the training of a new RL controller. The results show that TL can effectively reduce the training time of a new policy if the new home is similar to the benchmark home; otherwise, the advantage of using TL diminishes. In future studies, home similarity should be more formally defined and evaluated according to the home similarity distribution based on a large and realistic data set, and then the quantitative relationship between home similarity and TL acceleration potential should be studied. Finally, the best practice for TL implementation in real-life applications should be explored: either in the "benchmark-new homes" manner as used in this paper or in a daisy-chain TL manner (new homes becomes benchmark homes in other TL sessions).

## ACKNOWLEDGMENTS

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the *Techno-Economic Analysis of a Novel Smart Home Optimal Control Framework based on Cloud Computing and Deep Reinforcement Learning* Project funded by the National Renewable Energy Laboratory's Laboratory Directed Research and Development program. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

This research was performed using computational resources sponsored by the Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory.

## REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
[2] Commonwealth Edison Company. 2020. Five-Minute Real Time Price. Accessed: July. 30th, 2020. [Online]. Available: https://hourlypricing.comed.com/live-prices/five-minute-prices.
[3] Pengwei Du and Ning Lu. 2011. Appliance commitment for household load scheduling. *IEEE transactions on Smart Grid* 2, 2 (2011), 411–419.
[4] Hepeng Li, Zhiqiang Wan, and Haibo He. 2020. Real-Time Residential Demand Response. *IEEE Transactions on Smart Grid* (2020).
[5] Eric Liang, Richard Liaw, Philipp Moritz, Robert Nishihara, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. 2017. RLlib: Abstractions for Distributed Reinforcement Learning. arXiv:1712.09381 [cs.AI]
[6] Mohammad Rastegar, Mahmud Fotuhi-Firuzabad, and Farrokh Aminifar. 2012. Load commitment in a smart home. *Applied Energy* 96 (2012), 45–54.
[7] Frederik Ruelens, Bert J Claessens, Salman Quaiyum, Bart De Schutter, R Babuška, and Ronnie Belmans. 2016. Reinforcement learning applied to an electric water heater: from theory to practice. *IEEE Transactions on Smart Grid* 9, 4 (2016), 3792–3800.
[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
[9] Matthew E Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10, 7 (2009).
[10] Kai Man Tsui and Shing-Chow Chan. 2012. Demand response optimization for smart home scheduling under real-time pricing. *IEEE Transactions on Smart Grid* 3, 4 (2012), 1812–1821.
[11] Chengshan Wang, Yue Zhou, Bingqi Jiao, Yamin Wang, Wenjian Liu, and Dan Wang. 2015. Robust optimization for load scheduling of a smart home with photovoltaic system. *Energy Conversion and Management* 102 (2015), 247–257.
[12] Liang Yu, Weiwei Xie, Di Xie, Yulong Zou, Dengyin Zhang, Zhixin Sun, Linghua Zhang, Yue Zhang, and Tao Jiang. 2019. Deep Reinforcement Learning for Smart Home Energy Management. *IEEE Internet of Things Journal* 7, 4 (2019), 2751–2762.
[13] Xiangyu Zhang, Dave Biagioni, Mengmeng Cai, Peter Graf, and Saifur Rahman. 2020. An Edge-Cloud Integrated Solution for Buildings Demand Response Using Reinforcement Learning. *IEEE Transactions on Smart Grid* (2020).
[14] Bin Zhou, Wentao Li, Ka Wing Chan, Yijia Cao, Yonghong Kuang, Xi Liu, and Xiong Wang. 2016. Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable and Sustainable Energy Reviews* 61 (2016), 30–40.