The CityLearn Challenge 2022: Overview, Results, and Lessons Learned

Kingsley Nweye NWEYE@UTEXAS.EDU
Zoltan Nagy NAGY@UTEXAS.EDU
The University of Texas at Austin, Austin, TX, USA

Sharada Mohanty MOHANTY@AICROWD.COM
Dipam Chakraborty DIPAM@AICROWD.COM
AI Crowd

Siva Sankaranarayanan SSANKARANARAYANAN@EPRI.COM
Electric Power Research Institute, Palo Alto, CA, USA

Tianzhen Hong THONG@LBL.GOV
Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Sourav Dey SOURAV.DEY@COLORADO.EDU
Gregor Henze GREGOR.HENZE@COLORADO.EDU
University of Colorado Boulder, Boulder, CO, USA

Jan Drgona JAN.DRGONA@PNNL.GOV
Pacific Northwest National Laboratory, Richland, WA, USA

Fangquan Lin FANGQUAN.LINFQ@ALIBABA-INC.COM
Wei Jiang ALICE.JW@ALIBABA-INC.COM
Hanwei Zhang HANWEI.ZHANGHW@ALIBABA-INC.COM
Li Wang WANGLILION12@GMAIL.COM
Zhongkai Yi YZK_ARTICLE@163.COM
Jihai Zhang JIHAI.ZJH@ALIBABA-INC.COM
Cheng Yang CHARIS.YANGC@ALIBABA-INC.COM
Alibaba Group, Hangzhou, China

Matthew Motoki MMOTOKI@UW.EDU
University of Washington, Seattle, WA, USA
Sorapong Khongnawang KHONGNAW@HAWAII.EDU
University of Hawaii at Manoa, Honolulu, HI, USA

Michael Ibrahim MICHAEL.NAWAR@ENG.CU.EDU.EG
Computer Engineering Department, Faculty of Engineering, Cairo University, Giza, Egypt

Abilmansur Zhumabekov ZHUMABEK@UALBERTA.CA
Daniel May DCMAY@UALBERTA.CA
University of Alberta, Edmonton, Canada
Zhihu Yang YZHCODE@GMAIL.COM
Zuoyebang Education Technology, Beijing, China

Xiaozhuang Song SHAWNSXZ97@GMAIL.COM
Southern University of Science and Technology, Shenzhen, China
Han Zhang HAN-ZHAN17@MAILS.TSINGHUA.EDU.CN
Xiaoning Dong DONGXN20@MAILS.TSINGHUA.EDU.CN
Tsinghua University, Beijing, China
Shun Zheng SHUN.ZHENG@MICROSOFT.COM
Jiang Bian JIANG.BIAN@MICROSOFT.COM
Microsoft Research Asia, Beijing, China

Editors: Marco Ciccone, Gustavo Stolovitzky, Jacob Albrecht

Abstract

The shift to renewable power sources and building electrification to decarbonize existing and emerging building stock present unique challenges for the power grid. Building loads and flexible resources e.g. batteries must be adequately managed simultaneously to unlock the full flexibility potential and reduce costs for all stakeholders. Simple control algorithms based on expert knowledge e.g. rule-based control (RBC), as well as, advanced control algorithms e.g. model predictive control (MPC) and reinforcement learning control (RLC) can be utilized to intelligently manage flexible resources. The CityLearn Challenge is an opportunity to compete in investigating the potential of artificial intelligence (AI) and distributed control systems to tackle multiple problems within the built-environment. The CityLearn Challenge 2022 is the third of its kind with the overall objective of crowd-sourcing generalizable control policies that improve energy, cost and environmental objectives by taking advantage of batteries for load shifting in a CityLearn digital twin of a real-world grid-interactive neighborhood. Highlighted here are the uniqueness of this third edition, baseline and top solutions, and lessons learned for future editions.

Keywords: reinforcement learning, model predictive control, sustainability, building energy management, demand response

1. Introduction

Residential building stock in the United States accounts for ≈ 21% of energy consumption (Energy Information Administration) and 20% of greenhouse gas (GHG) emissions (Goldstein et al., 2020) thus, has a significant potential for climate change action. Active storage systems such as batteries reduce grid peaks by shifting building energy use to different times. When coupled with high thermal performance envelopes, efficient energy systems and appliances, and solar photovoltaic (PV) generation, batteries can reduce the overall demand on the grid while also reducing carbon emissions. However, all these resources must be carefully managed simultaneously in all buildings to unlock their full energy potential and reduce costs for stakeholders e.g. homeowners.

Reinforcement learning (RL) has gained popularity in the research community as a model-free and adaptive controller for the built-environment. RL has the potential to become an inexpensive controller that can be easily implemented in any building regardless of its model, unlike MPC, and coordinate multiple buildings for demand response and load shaping. Despite its potential, there are still open questions regarding its plug-and-play capabilities, performance, safety of operation, and learning speed.

The CityLearn Challenge was launched in the year, 2020 to address these open questions in CityLearn, a standard OpenAI Gym Environment for benchmarking of advanced control algorithms for demand response studies (Vázquez-Canteli et al., 2019). The CityLearn Challenge is an opportunity to compete in investigating the potential of AI and distributed control systems to tackle multiple problems within the built-environment domain. It attracts a multidisciplinary participation audience including researchers, industry experts, sustainability enthusiasts and AI hobbyists as a means of crowd-sourcing solutions to these multiple problems. We refer the reader to the official CityLearn website[1] for more details about the environment, its usage, previous challenges and related publications.

This third edition of The CityLearn Challenge, The CityLearn Challenge 2022, utilized a novel dataset from a real-world grid-interactive community to crowd-source and reward top solutions that minimized grid-level purchased electricity cost, carbon emissions and ramping while increasing the load factor. An overview of the competition, results and key takeaways are presented in this work and the structure of the remainder of the paper is as follows: Section 2 provides an overview of the competition objectives, resources, phases and evaluation criteria. Then, a summary of participant submissions is provided in Section 3 and the top five solutions are spotlighted. In Section 4, we discuss the implications of the solutions in solving the challenge as well as lessons learned from this third edition. Other competitions in the building domain and previous editions of The CityLearn Challenge are discussed in Section 5 whilst we conclude and provide a future outlook in Section 6.

2. Competition Overview

The CityLearn Challenge 2022 competition was run in three phases where each phase presented a new challenge of control policy generalization and introduced a new function for evaluation. The competition was hosted on AIcrowd, a platform for crowdsourcing AI to solve real-world problems, which vastly increased the competition's visibility compared to previous editions.

There was a total of 15,000 USD in cash prizes awarded to the top three solutions while community prizes were awarded to three other participants based on their contributions during the course of the competition.

We refer the reader to the official competition page[2] for other competition details that are not included in this paper.

2.1. Task

The task in the CityLearn Challenge 2022 was for participants to train control policies that reduced grid electricity cost, carbon emissions and ramping, and increased load factor by managing the charging and discharging of a battery in each building in a grid-interactive neighborhood's CityLearn digital twin.

There were 17 single-family buildings that made up the neighborhood and were based on data from a real-world zero net energy neighborhood in Fontana, California, USA that were studied for grid integration of zero net energy communities as part of the California Solar Initiative program specifically exploring the impact of high PV penetration and on-site electricity storage (Narayanamurthy et al., 2016).

The choice of control policy algorithm was left to participants' discretion and could feature expert control algorithms or advanced control algorithms e.g. RBC, single/multi agent RLC or MPC. Where applicable, as in the case of RLC, participants could choose to design their own custom reward function to improve their policies' learning outcomes.

[1] https://www.citylearn.net

[2] https://www.aicrowd.com/challenges/neurips-2022-citylearn-challenge

2.2. Resources

2.2.1. Starter-Kit

Participants were provided with a starter-kit repository[3] that served as a submission template. It included competition dependencies, and provided source-code for baseline solutions (see Section 2.5), the 5-building train dataset (see Section 2.2.2) and instructions on how to make submission for online evaluation.

2.2.2. Datasets

The CityLearn environment makes use of datasets to define the simulation environment as well as provide observation values. The data files include a schema that is used to initialize the environment and flat files containing time series data that provide the control policy (control agent) with observations that are independent of control actions (i.e. observations that are not a function of the control actions).

The CityLearn Challenge 2022 datasets (Nweye et al., 2023b) were competition phase-specific where the Phase I dataset was a train dataset that represented five of the 17 buildings in the neighborhood. This train dataset was the only publicly available dataset that was included in the starter-kit for participants to use in training their policies. The validation dataset represented five out of 17 other buildings and was introduced privately to the online evaluator in Phase II. Likewise, the test dataset was released privately to the Phase III online evaluator and represented the remaining seven buildings in the neighborhood. These datasets contained one year of hourly typical-meteorological year weather time series from a nearby weather station, real-world carbon intensity time series, electricity rate time series and building-level static observations time series. The set of building-level time-series data files were the only files that differed amongst the train, validation and test datasets.

2.2.3. Other Resources

Other resources utilized in the competition were communication tools including a dedicated discussion board and Discord channel. These tools provided means to disseminate competition updates to participants and an avenue for participants to ask the organizers questions or share insights amongst themselves. There were also community-driven resources such as Jupyter notebooks that provided example use-cases and custom baseline solutions.

2.3. Phases

The execution of the CityLearn Challenge 2022 spanned over a period of about five months that was split into three phases where participants made submissions, a post-competition review and winner selection period and finally, an online workshop where top solutions and winners were announced (see Fig. 1 for competition timeline). Each phase had its own leaderboard. We provide more detail on the phases and post-competition activities in Sections 2.3.1 to 2.3.4.

[3] https://gitlab.aicrowd.com/aicrowd/challenges/citylearn-challenge-2022/citylearn-2022-starter-kit

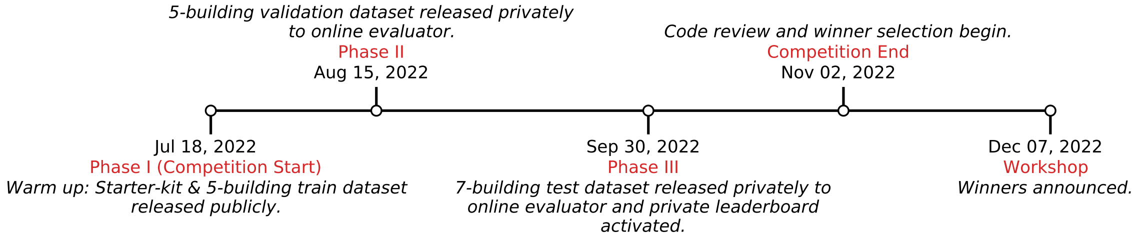
5-building validation dataset released privately to online evaluator.
Phase II
Aug 15, 2022
Code review and winner selection begin.
Competition End
Nov 02, 2022
Jul 18, 2022
Phase I (Competition Start)
Warm up: Starter-kit & 5-building train dataset released publicly.
Sep 30, 2022
Phase III
7-building test dataset released privately to online evaluator and private leaderboard activated.
Dec 07, 2022
Workshop
Winners announced.

Figure 1: Competition timeline.

2.3.1. Phase I

Phase I marked the beginning of the competition on July, 18, 2022 and lasted for 29 days. In Phase I, the five-building train dataset was publicly released with the starter-kit via a GitLab repository. Phase I is also regarded as a warm-up phase where participants familiarized themselves with the CityLearn environment, submission and evaluation processes.

On the organizers side, Phase I provided an opportunity to improve the source code documentation and fix existing bugs based on participant feedback and interaction with the CityLearn environment.

2.3.2. Phase II

Phase II of the competition began on August 15, 2022 and lasted for 47 days. In this second phase, the private five-building validation dataset was released to the online evaluator. Participants needed to ensure that their solutions generalized well to these unseen buildings when evaluated online.

2.3.3. Phase III

Phase III of the competition began on November 02, 2022 and ended 34 days later when submission acceptance was discontinued. Unlike Phases I and II that constituted only public leaderboards, Phase III included a private leaderboard that was only visible to the organizers. The private leaderboard displayed scores that factored in the remaining private 7-building test dataset during evaluation. Phase III also introduced a new grid key performance indicator (KPI), D (see Section 2.4).

Towards the end of Phase III, participants selected two submissions that were submitted during Phase III to be entered into the review and winner selection process. These were the submissions displayed on the private leaderboard.

2.3.4. Review and Workshop

The organizers began a code review process on the submissions that were entered for winner selection. The process checked for data fair-use and ensured that no leaked datasets were used in the policy training process. Finally, on December 07, 2022 a virtual workshop was held where the top five submission participants were invited to present their solutions followed by the announcement of the competition winners.

2.4. Evaluation

Refer to Appendix A for a detailed formulation of the KPIs and functions used for evaluation of participants' submissions.

2.5. Baselines

CityLearn provides a number of baseline policies in its *agents* sub-module. Here, three of such baseline control policies are highlighted to include a poorly tuned RBC a fine-tuned RBC and an RL policy that starts off by using the fine-tuned RBC in the beginning 7,000 time steps of training for safe initialization and exploration instead of risking random actions before switching to an Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018). The SAC policy was trained for eight episodes and used a reward function that was the sum of the hourly electricity rate, T_h, and carbon intensity, O_h, multiplied by the negative of the electricity consumption from the grid, $\min(0, -E_h)$ where E_h is the net electricity consumption in the neighborhood (Eq. (1)).

$$r_h = \min(0, -E_h) \times (T_h + O_h) \tag{1}$$

3. Solutions

Over the course of the competition, there were 1,005 valid submissions (196 in Phase I, 401 in Phase II and 408 in Phase III) from 655 participants. Of the total participation, 201 participants formed 105 teams with as many as seven participants per team. By Phase III, 41 participants and teams indicated two submissions made to the Phase III public leaderboard to be entered into the private leaderboard and used for winner selection.

We refer the reader to Appendix B.1 for a summary of the Phase III private leaderboard. Also, all phase-level leaderboards and submitted policies are accessible through the official competition leaderboard[4]. Nevertheless, the top five submissions in the Phase III private leaderboard from which winners were selected are elaborated in Sections 3.1 to 3.5.

3.1. First Place: Team Together

3.1.1. Solution Summary

Team Together proposed an ensemble approach of forecasting, optimization and RL[5]. As shown in Fig. B.2, the methodology consisted of three stages, including data-preprocessing, forecasting and decision-making. In terms of data-preprocessing, feature engineering was used to prepare the input for the following two stages. The input included historical time-series of load demand and solar generation, parameters of number of PV arrays, battery capacity and efficiency, as well as the future information of weather forecasting, electricity price and carbon intensity.

In the forecasting stage, Team Together used gradient-boosted decision trees (GBDT) and the linear least squares model to predict future load and solar generation, based on the given values of loads, solar generation and weather information in multi-timescales. An

[4] https://www.aicrowd.com/challenges/neurips-2022-citylearn-challenge/leaderboards

[5] https://gitlab.aicrowd.com/Kafka/citylearn-2022-starter-kit/-/tree/master

ensemble of two prediction models was employed to improve the prediction accuracy. Due to the difference in the distribution of training data and testing datasets, they use self-adaptive prediction adjustment to perform feedback correction by minimizing the difference between the historical loads and the predicted loads.

In the decision-making stage, it was required to decide the battery charging and discharging actions based on all given information. The decision model was an ensemble of a stochastic optimization model RL model. The stochastic optimization model performed a linear approximation of the complex optimization objective and used the MindOpt solver (MindOpt, 2022), which is a powerful, efficient and user-friendly solver. During training, the stochastic optimization model used stochastic data augmentation to improve the generalization ability thus, preventing inaccurate predictions for future loads. Rolling-horizon control was used to re-predict and re-schedule at regular time intervals. Moreover, the problem was modeled as a cooperative multi-agent problem and solved with MAPPO (Yu et al., 2021), a state-of-the-art multi-agent RL algorithm. Particularly during training, the RL model modified the rewards in the trajectory after sampling to learn the real reward. All RL agents shared parameters and buffers.

We refer the reader to Appendix B.2 for lessons learned and acknowledgments by Team Together.

3.2. Second Place: Team ambitiousengineers

3.2.1. Solution Summary

Team ambitiousengineers' solution consisted of forecasting and policy optimization via evolutionary strategies[6]. Evolution strategies offer an attractive alternative to RL when the problem is not differentiable and/or easily parallelized Salimans et al. (2017). The policy optimization consisted of generating single-agent policies and then combining them to obtain a final multi-agent policy.

During Phase I, the approach first solved the single-agent optimization problem using dynamic programming (DP). Because the grid costs are non-causal, they are replaced with a proxy that aimed to penalize large net energy usage. Next, the single-agent actions were improved using a multi-agent neural network policy trained with CMA-ES. The input to the policy network consisted of the single-agent DP actions, future costs, and future energy usage. Furthermore, the energy use was calculated for a given action using the battery implementation in the CityLearn environment and then used as additional input to the policy network.

The approach during Phases II and III, was similar to that of Phase I with a few notable exceptions. The single-agent DP was replaced with a single-agent neural network policy and the single-agent and multi-agent policy are trained end-to-end using historical data from Phase I. The remainder of the procedure remained the same as in Phase I with the exception that future values were replaced with forecasts of energy consumption and solar generation at each building. The forecasts consisted of a multilayer perceptron that used seasonal exponential smoothing and lags of the historical data as input. The model for energy consumption additionally learned embeddings for each time of day and day of

[6] https://gitlab.aicrowd.com/pluto/citylearn-2022-ambitiousengineers-solution

the week. The models are trained on data from Phase I. In the final solution, the models were trained with different seeds and blended to obtain a final prediction.

We refer the reader to Appendix B.3 for lessons learned and acknowledgments by Team ambitiousengineers.

3.3. Third Place: Team CUFE

3.3.1. Solution Summary

Team CUFE's solution consisted of linear forecasting models to forecast the power consumption and generation at each building, and policy optimization at each building using a linear programming optimizer. The implementation and the detailed description of the developed system are available in the team's repository [7].

For the power generation and consumption forecasting tasks, a linear regression model was trained that forecasts the generated and consumed energy for each building j based on the observations from the five-building train dataset $\hat{C}^{j}_{i+1\dots i+24} = \sum_{b=1}^{5} \alpha_b \hat{C}^{b}_{i+1\dots i+24}$, where the parameters $\alpha_1 \dots \alpha_5$ were updated every 168 simulation steps based on the observed power consumption and generation. A second autoregressive forecasting model was trained with a lag of 168 for each building. The coefficients of this second model were calculated offline using the observations from the five-building train dataset. Finally, the forecast for generated and consumed energy for each building during the simulation was a linear combination of the two forecasts with coefficients calculated every 168 simulation steps based on the observed power consumption/generation.

For the battery charging and discharging decision, for each building, the linear program defined in Eq. (B.1) was solved at each simulation step where the variable X_0 was then used as a charging/discharging decision, $\hat{C}_i$ was the forecast net consumption after i steps, P_i was the price of the electricity after i steps and I_i was the carbon intensity after i steps. $\hat{P}$ was the total price for the next 24 steps when there is no battery. $\hat{I}$ was the total carbon cost for the next 24 steps when there was no battery. $\hat{R}_{no}$ was the total ramp for the next 24 steps when there was no battery and $\hat{M}$ was the maximum net electricity consumption of the previous 730 steps of the simulation. $\hat{L}$ was the load change weight, and it was equal to $\frac{1}{\hat{M}(1-L_{no})}$ and was calculated from the previous 730 steps of the simulation.

As for the variables, X_i represented the decision at step i. S_i was the battery storage at step i, N_i^+ and N_i^- were the positive and negative parts of the net consumption at step i. R_i was the ramp at step i, and L was slack between the maximum at the current net consumption. Note S_0 was known since it was the battery status at the current simulation step. N_0^+ and N_0^- are also known since they were the current consumption at the current simulation step.

We refer the reader to Appendix B.4 for lessons learned and acknowledgements by Team CUFE.

[7] https://gitlab.aicrowd.com/MichaelIbrahim/citylearn-2022-submission-205030/-/tree/main

3.4. Fourth Place: Team DivMARL

3.4.1. Solution Summary

Ensembles of diverse models are widely employed in machine learning to enhance generalization in tasks such as regression, classification, clustering, etc. (Zhou, 2012; Rokach, 2019). Team DivMARL finds that using diverse models in an ensemble can improve generalization in continuous control tasks as well. For the battery control problem, Team DivMARL designed an ensemble of a hand-crafted (HC) policy and a deep reinforcement learning (DRL) policy, which combined the individual outputs by taking the unweighted average. It was found that the ensemble performed better during validation and testing compared to either policy alone.

The DRL policy was trained end-to-end using the SAC algorithm (Haarnoja et al., 2018). It controlled each battery in a decentralized manner, but parameters were shared across buildings.

The HC policy consisted of two modules: a predictor of net demand in the next hour, and a decision tree that relied on the predictions to choose an action. Here, net demand was defined as non-shiftable load minus solar generation. To predict the non-shiftable load of a building, the XGBoost algorithm (Chen and Guestrin, 2016) was employed. Its input features were: 1) periodically normalized hour of the day 2) past 16 days average of non-shiftable load for the next hour 3) rolling 24-hour non-shiftable load profile. For predicting solar generation, a linear regressor was trained with the following features: 1) solar generation in the past two hours 2) average solar generation in the district in the past two hours.

Fig. B.3 shows the two-level decision structure of the HC controller. δ was the difference between non-shiftable load and solar generation values predicted for the next hour. If $\delta > 0$, demand was predicted to be higher than generation, then theHC controller attempted to charge by the amount δ. Otherwise, the HC controller tried to discharge by $|\delta|$. However, in both cases, there was a limit to charging/discharging. Since the electricity pricing was much higher during hours 16 through 20 (15:00-19:00), called the 'crucial' hours. Up to 40% of the battery capacity was allowed to be discharged during crucial hours, but only 10% during other hours. Charging was not limited during non-crucial hours, other than by $|\delta|$ and the battery's remaining capacity. On the other hand, when the next hour was crucial, even if surplus generation was predicted — charging was not allowed because predictions were imperfect, and charging by mistake was very costly during those hours. The limiting parameters were tuned based on the algorithm's performance on the five-building train dataset.

We refer the reader to Appendix B.5 for lessons learned and acknowledgments by Team DivMARL.

3.5. Fifth Place: Team Greener

3.5.1. Solution Summary

Team Greener adopted an imitation-learning approach to train a single policy network that fulfilled decision making of all buildings. To obtain effective imitation labels, The overall objective of all buildings was first decomposed into a specific sub-objective for each building.

For those costs not supporting an exact decomposition, such as price and ramping costs, an upper bound was derived that can be decomposed naturally for each as a surrogate subobjective. Specifically, certain surrogate sub-objectives were generated that can be solved by dynamic programming. In this way, optimal action trajectories were obtainable for these surrogate sub-objectives, which functioned as very good approximations for original costs. Then given these near-optimal action trajectories, imitation learning was applied to train a single policy network, which was essentially a standard multi-layer perceptron. It was found that a single shared-parameter policy network can generalize to various distributions of different buildings. Moreover, to facilitate more robust generalization for unseen buildings, different surrogate objectives were developed by slightly adjusting some hyper-parameters and produced multiple imitation networks as ensemble candidates.

The ensemble mechanism generated the final action by averaging that of each well-trained policy network. Data augmentation on the original environment was also performed to create new environments with different distributions of non-shiftable loads and solar generations. For each augmented environment, the aforementioned imitation learning procedures can be re-run to produce multiple policy networks with various hyper-parameters. Accordingly, the final solution was an average ensemble of diversified policy networks, which corresponded to different hyper-parameters and data distributions.

We refer the reader to Appendix B.6 for lessons learned by Team Greener.

4. Discussion

Nweye et al. used the CityLearn Challenge 2022 dataset in their MERLIN framework to address the data requirement, control security and generalizability challenges hindering real-world adoption of RL in building control applications (Nweye et al., 2023a). They showed that while independent RL controllers for batteries improved electricity price, carbon emissions and grid KPIs compared to the baseline, transferring the RL policy of any one of the buildings to other buildings provided comparable performance to a policy trained on building-specific data, while reducing the cost of training despite unique occupant behaviours. This highlights the importance of an adaptive control approach especially as new developments occur in communities, or more homes are fitted with energy management systems. In such cases, the lack of historical data can be alleviated by using existing building control model to jump-start the use of the available distributed energy resources (DERs) efficiently.

The CityLearn Challenge 2022 was another learning experience for the organizers and highlighted some areas for improvement. In future competitions, separate tracks for purely deterministic control solutions e.g. expert RBC and adaptive control solutions e.g. MPC, RLC could be provided as early stages of the challenge showed that participants were likely to opt for a simpler control algorithm if there was no incentive to provide expensive but adaptive algorithms. Also, given that CityLearn is under continuous development, it is susceptible to bugs hence, Phase I should be strictly a warm-up round to test software, data and receive feedback from participants but, not be used in evaluations. Additionally, future competitions could provide integration with standard RL and MPC Python libraries e.g. Stable-Baselines3 (Raffin et al., 2021) and do-mpc (Lucia et al., 2017) to provide ample baselines for participants. Lastly, the control action space could be increased to include other DERs e.g. electric vehicle (EV), heat pump and other typical flexible assets in buildings.

5. Related Work

Previous competitions in the energy domain focused on building load predictions (Miller et al., 2020), grid power flow optimization (Aravena et al., 2022; Holzer et al., 2021) and pathways to building electrification and decarbonization (NYSERDA RTEM Hackathon). While these competitions provided a plethora of solutions that are pertinent to energy supply and demand-side management, to the best of the authors' knowledge, The CityLearn Challenge is one of its kind with a focus on building energy system controls and algorithm benchmarking for demand response studies. Previous editions of The CityLearn Challenge had investigated transferrability of control policies trained in one climate zone to another (Vázquez-Canteli et al., 2020), and a realistic implementation of a model-free RL in buildings where training evaluation is done on a single four-year long episode (Nagy et al., 2021). In contrast to previous editions, this The CityLearn Challenge 2022 1) made use of a real world data set as opposed to simulation dataset thus presented challenges of data quality and fidelity but also introduced realistic energy use patterns 2) only used batteries for load shifting which on one hand reduced the complexity of the control space but constrained the availability and capacity of flexible resources and 3) was hosted on the AIcrowd platform which increased visibility and participation.

6. Conclusion

The CityLearn Challenge 2022 was the third edition of its kind and saw submissions from 655 participants in the ≈ five-month competition timeline. In this work, we highlighted the uniqueness of this edition's challenge, baseline and top solutions, and lessons learned for future editions. The 2022 edition addressed the problem of optimal battery control algorithms for load shifting using a real-world dataset and constrained the available and capacity of flexible resources compared to previous editions. All top five solutions utilized custom advanced and adaptive control algorithms including reinforcement learning, model predictive control, stochastic optimization and dynamic programming and 4/5 considered future loads and solar generation in their decision making. Key lessons learned bordered on the planning of competition execution and availability of materials and resources to aid participation. In summary, the presented work demonstrated a blue-print for a large scale execution of a buildings control challenge involving multi-disciplinary participants with various levels of expertise.

Acknowledgments

The competition organizers gratefully acknowledge funding from the UT Energy Institute to execute the challenge. We also thank our sponsors, the Electric Power Research Institute (EPRI) for providing the monetary prizes for the top three solutions, and Amazon Science for providing compute credits for the competition. ACM SIGEnergy is appreciated for providing travel grants to the ACM BuildSys 2023 and ACM eEnergy 2023 conferences in the community awards category.

References

Ignacio Aravena, Daniel K. Molzahn, Shixuan Zhang, Cosmin G. Petra, Frank E. Curtis, Shenyinying Tu, Andreas Wächter, Ermin Wei, Elizabeth Wong, Amin Gholami, Kaizhao Sun, Xu Andy Sun, Stephen T. Elbert, Jesse T. Holzer, and Arun Veeramany. Recent developments in security-constrained ac optimal power flow: Overview of challenge 1 in the arpa-e grid optimization competition, 2022. URL https://arxiv.org/abs/2206.07843.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Energy Information Administration. November 2022 Monthly Energy Review, 11 2022. URL https://www.eia.gov/totalenergy/data/monthly/archive/00352211.pdf.

Benjamin Goldstein, Dimitrios Gounaridis, and Joshua P Newell. The carbon footprint of household energy use in the united states. *Proceedings of the National Academy of Sciences*, 117(32):19122–19130, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *ICML 2018*, pages 1329–1338, Online, 2018. PMLR.

Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634, February 2019. URL https://doi.org/10.5281/zenodo.2559634.

Jesse Holzer, Carleton Coffrin, Christopher DeMarco, Ray Duthu, Stephen Elbert, Scott Greene, Olga Kuchar, Bernard Lesieutre, Hanyue Li, Wai Keung Mak, et al. Grid optimization competition challenge 2 problem formulation. Technical report, Tech. rep. ARPA-E, 2021.

Sergio Lucia, Alexandru Tatulea-Codrean, Christian Schoppmeyer, and Sebastian Engell. Rapid development of modular and sustainable nonlinear model predictive control solutions. *Control Engineering Practice*, 60:51–62, 03 2017. doi: 10.1016/j.conengprac.2016.12.009.

Clayton Miller, Pandarasamy Arjunan, Anjukan Kathirgamanathan, Chun Fu, Jonathan Roth, June Young Park, Chris Balbach, Krishnan Gowri, Zoltan Nagy, Anthony D. Fontanini, and Jeff Haberl. The ashrae great energy predictor iii competition: Overview and results. *Science and Technology for the Built Environment*, 26(10):1427–1447, 2020. doi: 10.1080/23744731.2020.1795514. URL https://doi.org/10.1080/23744731.2020.1795514.

MindOpt. Mindopt studio, 2022. URL https://opt.aliyun.com.

Zoltan Nagy, José R. Vázquez-Canteli, Sourav Dey, and Gregor Henze. The citylearn challenge 2021. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '21, page 218–219, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450391146. doi: 10.1145/3486611.3492226. URL https://doi.org/10.1145/3486611.3492226.

Ram Narayanamurthy, Rachna Handa, Nick Tumilowicz, CR Herro, and S Shah. Grid integration of zero net energy communities. *ACEEE Summer Study Energy Effic. Build*, 2016.

Kingsley Nweye, Siva Sankaranarayanan, and Zoltan Nagy. Merlin: Multi-agent offline and transfer learning for occupant-centric energy flexible operation of grid-interactive communities using smart meter data and citylearn, 2023a. URL https://arxiv.org/abs/2301.01148.

Kingsley Nweye, Sankaranarayanan Siva, and Gyorgy Zoltan Nagy. The CityLearn Challenge 2022, 2023b. URL https://doi.org/10.18738/T8/0YLJ6Q.

NYSERDA RTEM Hackathon. NYSERDA RTEM Hackathon Demo Day. URL https://be-exchange.org/nyserda-rtem-hackathon-demo-day/.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Lior Rokach. *Ensemble learning: pattern classification using ensemble methods*. World Scientific, 2019.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017. URL https://arxiv.org/abs/1703.03864.

José R. Vázquez-Canteli, Sourav Dey, Gregor Henze, and Zoltan Nagy. The citylearn challenge 2020. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '20, page 320–321, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380614. doi: 10.1145/3408308.3431122. URL https://doi.org/10.1145/3408308.3431122.

J.R. José R. Vázquez-Canteli, Jérôme Kämpf, Gregor Henze, and Zoltan Nagy. CityLearn v1.0: An OpenAI gym environment for demand response with deep reinforcement learning. *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 356–357, 2019. doi: 10.1145/3360322.3360998.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

Appendix A. Evaluation

Participant submissions were evaluated using a combination of KPIs whose values were to be minimized. These KPIs included the normalized electricity cost, C, carbon emissions, G, and a grid KPI, D, that is the average of a normalized ramping KPI, R, and (1 - Load Factor) KPI, $(1 - L)$.

Normalized electricity cost, C, is defined in Eq. (A.1) as the ratio of district electricity cost for a participant's submission, $c_{\text{submission}}$ to the baseline scenario of no battery control, $c_{\text{no battery}}$. c is then defined in Eq. (A.2) as the sum of non-negative district-level net electricity price, $E_h \times T_h$ (\$), where E_h is the district electricity consumption at hour h and T_h is the electricity rate at hour h.

$$C = \frac{c_{\text{submission}}}{c_{\text{no battery}}} \tag{A.1}$$

$$c = \sum_{h=0}^{n-1} \max\left(0, E_h \times T_h\right) \tag{A.2}$$

Normalized carbon emissions, G, is defined in Eq. (A.3) as the ratio of district carbon emissions for a participant's submission, $g_{\text{submission}}$ to the baseline scenario of no battery control, $g_{\text{no battery}}$. g is then defined in Eq. (A.4) as the sum of carbon emissions ($\text{kg}_{\text{CO}_2\text{e}}$), $E_h \times O_h$, where O_h is the carbon intensity ($\text{kg}_{\text{CO}_2\text{e}}$/kWh) at hour h.

$$G = \frac{g_{\text{submission}}}{g_{\text{no battery}}} \tag{A.3}$$

$$g = \sum_{h=0}^{n-1} \max\left(0, E_h \times O_h\right) \tag{A.4}$$

The normalized grid KPI, D, is defined in Eq. (A.5) as the average of the normalized ramping, R, and normalized (1 - load factor), $1 - L$ KPIs. R is a function of r (Eq. (A.6)) while $1 - L$ is a function of l (Eq. (A.7)) where m is the month index.

$$D = \overline{\frac{r_{\text{submission}}}{r_{\text{no battery}}}, \frac{(1-l)_{\text{submission}}}{(1-l)_{\text{no battery}}}} \tag{A.5}$$

$$r = \sum_{h=0}^{n-1} |E_h - E_{h-1}| \tag{A.6}$$

$$1 - l = \left(\sum_{m=0}^{11} 1 - \frac{\left(\sum_{h=0}^{729} E_{730m+h}\right) \div 730}{\max\left(E_{730m}, \ldots, E_{730m+729}\right)} \right) \div 12 \tag{A.7}$$

Using the aforementioned KPIs, we evaluated participant submissions using distinct combination of buildings and KPIs in each phase as summarized in Table A.1. The leaderboard during Phase I was public where participants submissions were evaluated using a district that contained the five train buildings, and submissions that minimized the average of the cost, C, and carbon emissions, G, KPIs were ranked higher.

Table A.1: Public and private leaderboard evaluation score functions for each competition phase. Winners are selected from the Phase III private leaderboard.

Phase	Leaderboard	
	Public	Private
I	🏠 $\overline{C, G}$	-
II	🏠 🏠 $\overline{C, G}$	-
III	🏠 🏠 $0.4 \times \overline{C, G, D} + 0.6 \times \overline{C, G, D}$	🏠 🏠 🏠 $0.2 \times \overline{C, G, D} + 0.3 \times \overline{C, G, D} + 0.5 \times \overline{C, G, D}$

🏠 Public 5-building train dataset.
🏠 Private 5-building validation dataset.
🏠 Private 7-building test dataset.

Similar to Phase I, a public leaderboard was used in Phase II and participants submissions were evaluated using a district that contained the five train and five validation buildings. The same evaluation function used in Phase I was used in Phase II.

The grid KPI, D was introduced in Phase III and submissions on the public leaderboard were evaluated using the weighted score of a district that was made up of the five train buildings and another district that was made up of the five validation buildings. The private leaderboard included the score of a third district that was made up of the seven test buildings. The winners of the competition were selected from the Phase III private leaderboard.

Appendix B. Solutions: Extended

B.1. Phase III Private Leaderboard Summary

Fig. B.1 shows the distribution of normalized electricity cost, C, carbon emissions, G and grid, D KPIs, as well as, final evaluation score for participants' submissions to the Phase III private leaderboard. Participants scores are compared to those of the baseline control policies: poorly-tuned RBC, fine-tuned RBC and SAC.

Recall from Appendix A that the objective of the competition was to minimize the KPIs and consequently, evaluation score. The SAC policy performed better than other baseline policies but under-performed compared to the top eight submissions on average (evaluation score). The poorly-tuned RBC is generally either outperformed or matched by all participants' policies while the fine-tuned RBC is above the median performance of participants' policies. . The median values for C, G, D and the evaluation score amongst participants were 0.792, 0.940, 0.996 and 0.907 respectively as they found it easiest to minimize C but toughest to optimize D.

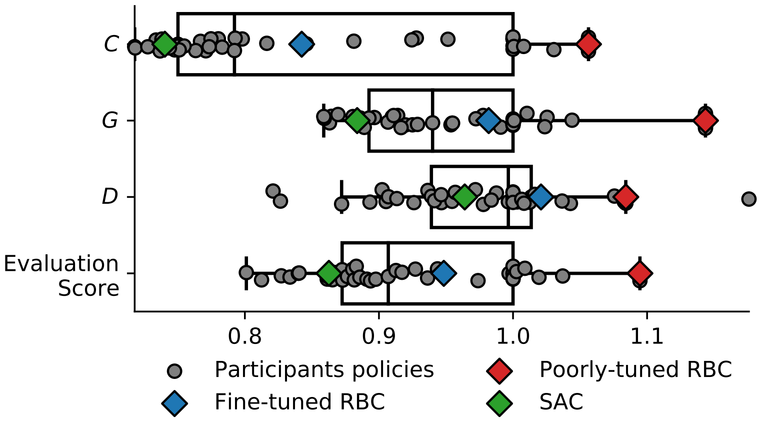

Figure B.1: Distribution of normalized electricity cost, C, carbon emissions, G, ramping, R, (1 - load factor), $1 - L$, and grid, D KPIs, as well as, final evaluation score for baseline and participant-submitted control policies in CityLearn in the Phase III private leaderboard.

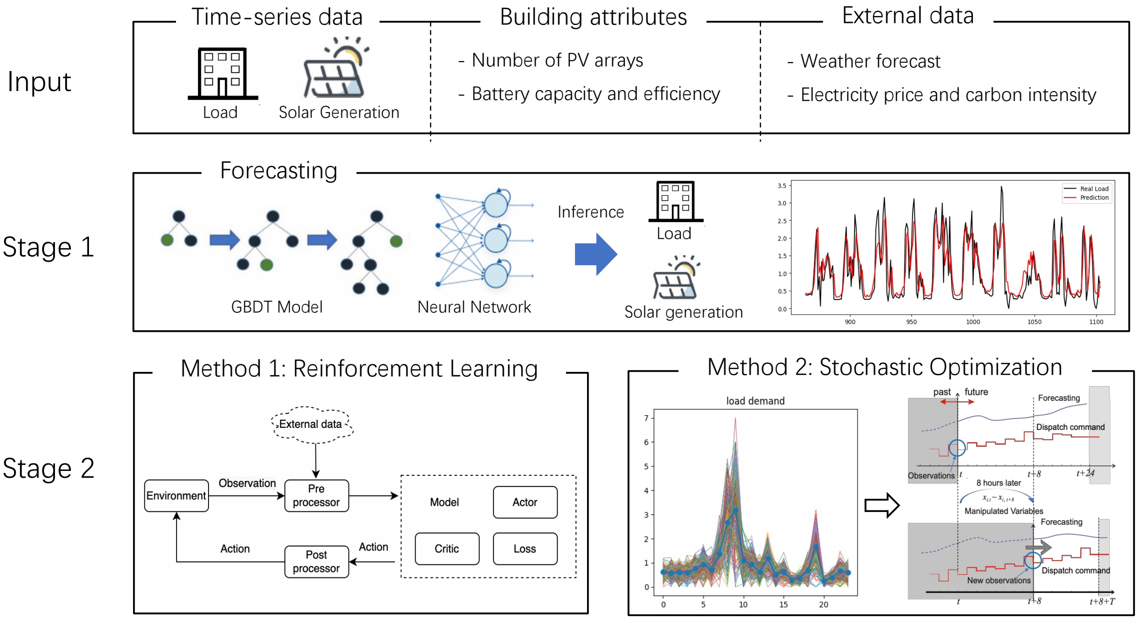

Figure B.2: Team Together: Proposed solution framework consisting of data-preprocessing module, forecasting module and decision-making module (combination of reinforcement learning and stochastic optimization).

B.2. First Place: Team Together

B.2.1. Lesson's learned

The challenge of this competition was to handle a complex long-term decision-making task. One of the most important things Team Together learned was that, it was difficult to achieve end-to-end learning with a single strategy for a complex problem. In this task, the key information for decision-making was the future load and solar energy generation. In addition, the weather and historical information played an auxiliary role in predicting the future key information. It was found that using pre-trained auxiliary task to learn representation and prediction ahead of optimization and RL, outperformed the method of directly feeding all the data into the decision model. This finding also motivates better design of representation models for energy management task in the future to achieve more powerful end-to-end learning.

Additionally, Optimization and multi-agent RL algorithms were used for decision making. The optimization algorithm can achieve better generalization on unknown dataset through target approximation, data augmentation, and rolling-horizon correction. Multi-agent RL can better model the problem and find better solutions on known dataset, but the algorithm effect cannot be guaranteed in buildings with new cooperative relationships. In energy management tasks, data augmentation to improve generalization ability is a problem worthy of research. It was also found that the policies learned by the optimization algorithm and RL performed differently in different months, which also prompted the use of ensemble learning. Similarly, ensemble modeling is also very helpful for improving the prediction effect.

B.2.2. Acknowledgment

Team Together is very grateful for the platform provided by the competition organizers and the support during the competition process. Team Together also thanks Professor Wotao Yin who provided insights and expertise that greatly assisted the research. Team Together thanks Jiayu Han for assistance with developing the codes. Gratitude is also shown to Yupeng Zhang for sharing his pearl of wisdom during the competition.

B.3. Second Place: Team ambitiousengineers

B.3.1. Lesson's learned

The solution for Phase I consisted of single-agent optimization via DP. The state of charge was used as the state, the state space was discretized, and tabular methods were used to solve the problem. Early in the competition, the capacity of the battery was added as a state. However, it was found that this approach did not improve the results. It is believed that the change in capacity as a result of degradation was too small to model accurately and solve the problem in reasonable time. It is believed that using function approximation approaches to DP will produce better results.

The final policy neural network consisted of fewer than 200 parameters and was thus, relatively small compared to modern neural network policies trained with RL. It was observed that increasing the number of parameters in the models did not produce better results. It is believed that this may be due to limitations in the chosen optimization technique, CMA-ES, which is typically applied to problems with up to 100 parameters. Hence, it may be possible to achieve better results using an optimizer that is better suited to training larger models.

B.3.2. Acknowledgment

Team ambitiousengineers is grateful to Hansen et al. (2019) for creating CMA-ES and pycma. The team's solution depends heavily on pycma to optimize policies and the ease of use of the library allowed for quickly iterating on ideas.

B.4. Third Place: Team CUFE

B.4.1. Linear Program for Battery Control

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{i=1}^{24} \frac{47-i}{46}\left(\left(\frac{P_i}{\hat{P}_{no}}+\frac{I_i}{\hat{I}_{no}}\right)N_i^{+}+0.5\frac{\hat{L}}{12}N_i^{-}+0.5\frac{R_i}{\hat{R}_{no}}\right)+0.5\frac{730}{24}\hat{L}L & \\
\text{subject to} & N_i^{+}-N_i^{-}-6.4X_{i-1} = \hat{C}_i, & i=1,\ldots,24 \\
& S_{i+1}-S_i-X_i = 0, & i=0,\ldots,23 \\
& N_i^{+}-N_i^{-}-N_{i-1}^{+}+N_{i-1}-R_i \leq 0, & i=1,\ldots,24 \\
& N_{i-1}^{+}-N_{i-1}^{-}-N_i^{+}+N_i-R_i \leq 0, & i=1,\ldots,24 \\
& N_i^{+}-L \leq \hat{M}, & i=1,\ldots,24 \\
& -1 \leq X_i \leq 1, & i=0,\ldots,23 \\
& 0 \leq S_i \leq 1, & i=1,\ldots,24 \\
& N_i^{+} \geq 0, & i=1,\ldots,24 \\
& N_i^{-} \geq 0, & i=1,\ldots,24 \\
& R_i \geq 0, & i=1,\ldots,24 \\
& L \geq 0 &
\end{array}
\tag{B.1}
$$

B.4.2. Lesson's learned

During the competition, three valuable lessons were learned. The first lesson was to conduct thorough analysis of the provided data beforehand and develop a model that would make use of the insights gained in the analysis phase. In this challenge, It was observed that the solar energy produced by buildings was highly correlated, so although it was not common in the literature, a linear model was developed that used historical solar energy produced by five buildings to predict the solar energy produced by all buildings. This model was very simple and the error of the model was very small.

The second lesson was to develop different models for different parts of the solution and different parts of the data as needed. It was observed that the energy consumed by the buildings was not correlated, so using the forecasting model developed for solar energy generation was good for the given five buildings in Phase I, but gave very poor forecasts for the other 12 buildings of Phases II and III. Thus, two different forecasting techniques were developed and a meta-learner was trained during the simulation to average the two forecasts based on their performances.

The final lesson was to always start with a simple model, then to implement a complex model if needed. In this competition, A simple linear regression model was used to forecast power generation and consumption. Also, a simple model predictive control was used with a linear program optimizer to control the battery charging and discharging decisions at

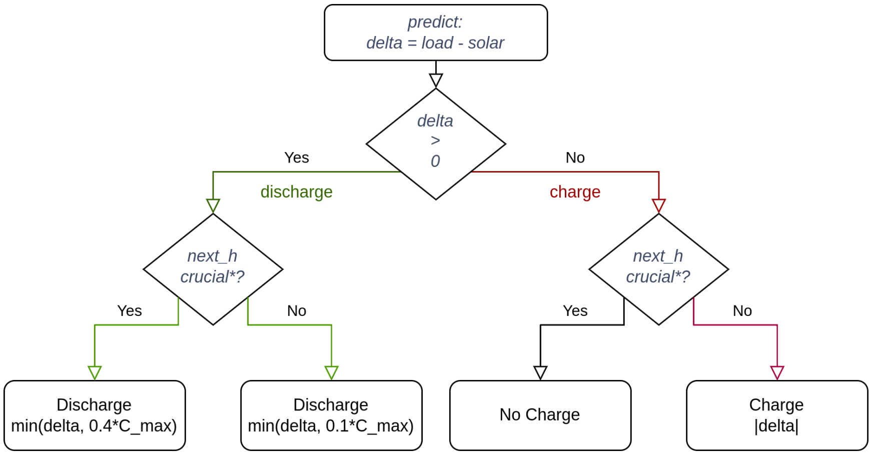

Figure B.3: Team DivMARL: HC policy's decision tree.

each building. Compared to other methods used in the challenge, this solution was easy to understand, implement and analyze, and also had a very fast execution time.

B.4.3. Acknowledgment

Team CUFE would like to thank the organizers of The CityLearn Challenge 2022 for their support and efforts during the competition.

B.5. Fourth Place: Team DivMARL

B.5.1. Lesson's learned

The experiments carried out by Team DivMARL showed that training one shared SAC policy on all Phase I buildings was better for generalization than training one policy per house. It was also discovered that the HC controller combined with the SAC policy in an ensemble can generalize better than either method alone. The possible reason behind this outcome was that the two policies had different biases that partially canceled each other out. The SAC policy was trained end-to-end with minimal human inductive bias limited to observation-space and reward function design. In contrast, the HC policy was heavily based on human intuition and its decisions were more conservative. Crucially, the ensemble's effectiveness highlights the merit of using DRL together with RBC built by human experts, instead of displacing them.

B.5.2. Acknowledgment

Team DivMARL expresses deep gratitude to all organizers of The CityLearn Challenge 2022. The competition greatly stimulated the development of our engineering skills and helped us form ideas for our research. Importantly, it also helped us build essential bonds and friendships with fellow enthusiasts from various parts of the globe.

Part of this work took place in the Intelligent Robot Learning (IRL) Lab at the University of Alberta, which is supported in part by research grants from the Alberta Machine

Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Compute Canada; Huawei; Mitacs; and NSERC.

B.6. Fifth Place: Team Greener

B.6.1. Lessons learned

The primary lesson learned through this challenge was that it was prohibitively expensive to naively explore the exponential parameter space of the original highly non-convex objective, given the requirement of considering both cooperative behaviors across multiple buildings and various dependencies along the temporal dimension to minimize the overall objective. Therefore, it was indispensable to reduce the complexity of this problem or to find an optimization strategy being capable of producing near-optimal solutions. Multiple kinds of approaches were explored to address this problem. For example, one of the most representative and generally plausible paradigm is to formulate a multi-agent RL problem, where multiple buildings (agents) cooperate with each other to optimize an overall objective. However, it was found that simply adapting existing RL algorithms to this problem can only produce approximated solutions with modest performance, which is far from satisfactory. It is conjectured the reason is that in such a highly non-convex scenario involving a very long trajectory (24 per-day steps for 365 days) with significantly delayed rewards, the network may probably converge to some local optima before effectively exploring the whole space of action series. This also explains why the final solution is an imitation-learning approach, in which dynamic programming is relied on to identify optimal solutions for surrogate objectives and then enforce a policy network to learn this behavior and generalize it to unseen buildings. Similarly, other winning teams also transformed the original problem into some appropriate optimization problems, such as constrained linear regression, and then applied extra solvers to obtain solutions. Although none of the existing solutions can make pure RL approaches work very well in this problem, given the soundness and generality of multi-agent RL formulation in this scenario, future research can address the optimization challenge of training a highly competitive policy network from scratch.