Smart Balancing of E-scooter Sharing Systems via Deep Reinforcement Learning: A Preliminary Study

Gianvito Losapio[a], Federico Minutoli[a], Viviana Mascardi[a], and Angelo Ferrando[a]
[a] *DIBRIS, University of Genova, Italy*
E-mail: gvlosapio@gmail.com, fede97.minutoli@gmail.com, viviana.mascardi@unige.it, angelo.ferrando@unige.it

Abstract. Nowadays, micro-mobility sharing systems have become extremely popular. Such systems consist in fleets of dockless electric vehicles which are deployed in cities, and used by citizens to move in a more ecological and flexible way. Unfortunately, one of the issues related to such technologies is its intrinsic load imbalance, since users can pick up and drop off the electric vehicles where they prefer.

In this paper we present ESB-DQN, a multi-agent system for E-Scooter Balancing (ESB) based on Deep Reinforcement Learning where agents are implemented as Deep Q-Networks (DQN). ESB-DQN offers suggestions to pick or return e-scooters in order to make the fleet usage and sharing as balanced as possible, still ensuring that the original plans of the user undergo only minor changes.

The main contributions of this paper include a careful analysis of the state of the art, an innovative customer-oriented rebalancing strategy, the integration of state-of-the-art libraries for deep Reinforcement Learning into the existing ODySSEUS simulator of mobility sharing systems, and preliminary but promising experiments that suggest that our approach is worth further exploration.

Keywords: Micro-mobility, Dockless E-scooter Sharing Systems, Smart Balancing, Multi-agent Systems, Deep Reinforcement Learning

1. Introduction

The adoption of Electric Vehicles (EV) has been constantly growing in the last few years and this trend is expected to accelerate exponentially. From the analysis of the electric vehicle market growth across U.S. cities published in September 2021, it turns out that

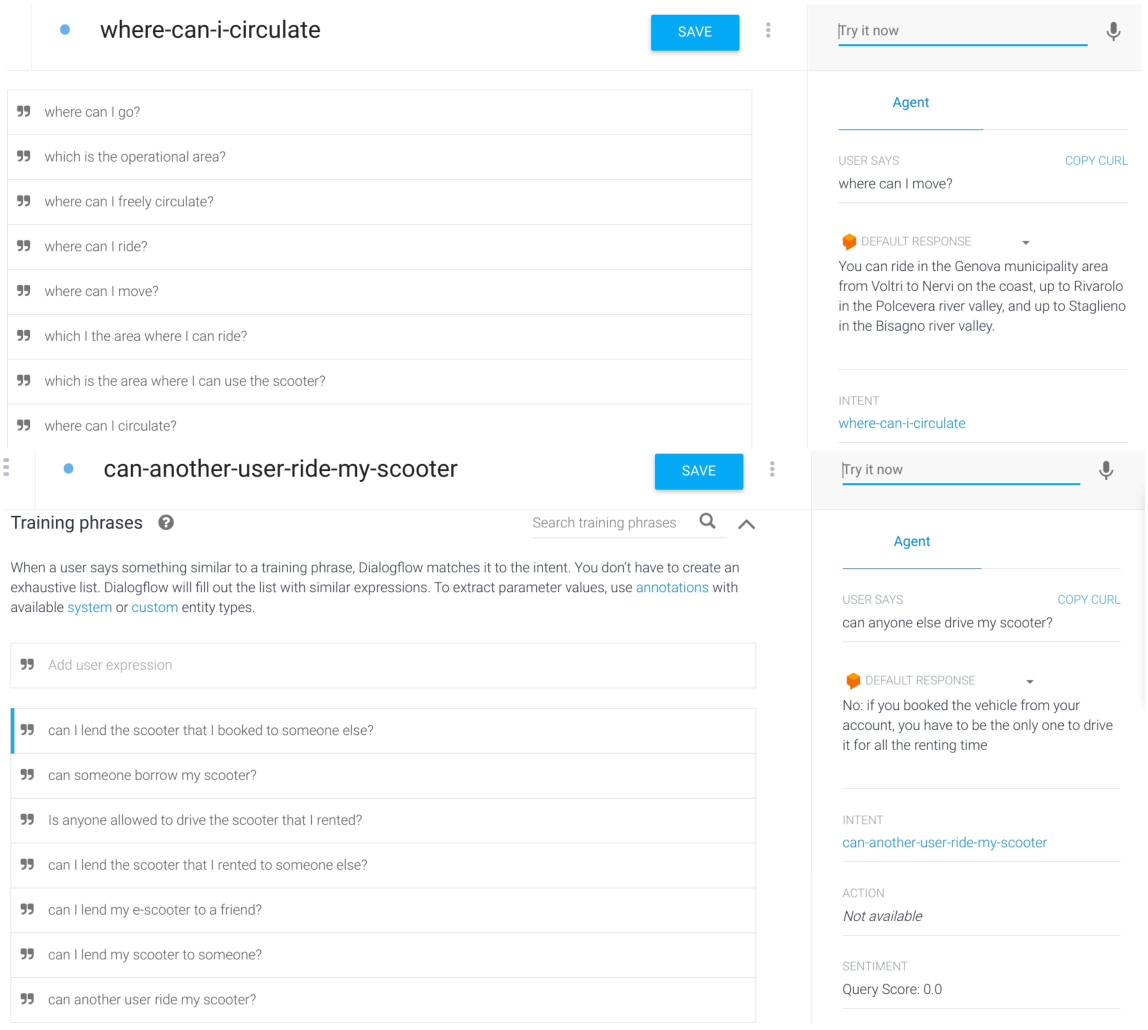
> *The electric vehicle market in the United States has grown from a few thousand vehicles in 2010 to more than 315 thousand vehicles sold annually from 2018 to 2020. In 2020, the electric share of new vehicle sales was approximately 2.4%, an increase from about 2% in 2019* [4].

Similar figures, at a global scale, are reported in Global EV Outlook issued in April 2021 by the International Energy Agency, IEA[1].

While these reports deal with any kind of electric vehicles including cars and public transportation means, lightweight two-wheels vehicles play a very important role in boosting the green trend by changing the way we conceive mobility in our cities.

As reported in the SLOCAT Transport and Climate Change Global Status Report 2nd Edition published in June 2021[2],

[1] `https://www.iea.org/reports/global-ev-outlook-2021`, accessed on January 10th, 2022.
[2] `https://tcc-gsr.com/`, accessed on January 10th, 2022.


where-can-i-circulate

where can I go?
which is the operational area?
where can I freely circulate?
where can I ride?
where can I move?
which I the area where I can ride?
which is the area where I can use the scooter?
where can I circulate?

USER SAYS: where can I move?

DEFAULT RESPONSE: You can ride in the Genova municipality area from Voltri to Nervi on the coast, up to Rivarolo in the Polcevera river valley, and up to Staglieno in the Bisagno river valley.

INTENT: where-can-i-circulate

can-another-user-ride-my-scooter

Training phrases

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types.

can I lend the scooter that I booked to someone else?
can someone borrow my scooter?
Is anyone allowed to drive the scooter that I rented?
can I lend the scooter that I rented to someone else?
can I lend my e-scooter to a friend?
can I lend my scooter to someone?
can another user ride my scooter?

USER SAYS: can anyone else drive my scooter?

DEFAULT RESPONSE: No: if you booked the vehicle from your account, you have to be the only one to drive it for all the renting time

INTENT: can-another-user-ride-my-scooter

ACTION: *Not available*

SENTIMENT: Query Score: 0.0

Fig. 5. Example of sentences that can be used inside DialogFlow to train the chatbot to answer questions related with the boundaries of the e-scooter operational zone, and with possibility to lend the rented e-scooter.

[10] Timo Eccarius and Chung-Cheng Lu. Adoption intentions for micro-mobility – insights from electric scooter sharing in Taiwan. *Transportation research part D: transport and environment*, 84:102327, 2020.

[11] Christine Fricker and Nicolas Gast. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *EURO Journal on Transportation and Logistics*, 5(3):261–291, 2016.

[12] M. Ramos García, Daniel A. Mántaras, Juan C. Álvarez, and David Blanco F. Stabilizing an urban semi-autonomous bicycle. *IEEE Access*, 6:5236–5246, 2018.

[13] Suining He and Kang G. Shin. Distribution prediction for reconfiguring urban dockless e-scooter sharing systems. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021.

[14] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Daniel Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *CoRR*, abs/1710.02298, 2017.

[15] IMARC. Chatbot Market: Global industry trends, share, size, growth, opportunity and forecast 2021-2026. `https://www.imarcgroup.com/chatbot-market`, accessed on January 10th, 2022., 2020.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[17] Dániel Kondor, Xiaohu Zhang, Malika Meghjani, Paolo Santi, Jinhua Zhao, and Carlo Ratti. Estimating the potential for shared autonomous scooters. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2021.

[18] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. Rebalancing bike-sharing systems: A multi-source data smart optimization. In *Proceedings of the 22nd ACM SIGKDD Inter-*

national Conference on Knowledge Discovery and Data Mining, KDD '16, page 1005–1014, New York, NY, USA, 2016. Association for Computing Machinery.

[19] Gianvito Losapio, Federico Minutoli, Viviana Mascardi, and Angelo Ferrando. Smart balancing of e-scooter sharing systems via deep reinforcement learning. In Roberta Calegari, Giovanni Ciatto, Enrico Denti, Andrea Omicini, and Giovanni Sartor, editors, *Proceedings of the 22nd Workshop "From Objects to Agents", Bologna, Italy, September 1-3, 2021*, volume 2963 of *CEUR Workshop Proceedings*, pages 83–97. CEUR-WS.org, 2021.

[20] Volodymyr Mnih, Koray Kavukcuoglu, and other authors. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[21] Ling Pan, Qingpeng Cai, Zhixuan Fang, Pingzhong Tang, and Longbo Huang. A deep reinforcement learning framework for rebalancing dockless bike-sharing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1393–1400, 2019.

[22] Julius Pfrommer, Joseph Warrington, Georg Schildbach, and Manfred Morari. Dynamic vehicle redistribution and online price incentives in shared mobility systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1567–1578, 2014.

[23] Robert Regue and Will Recker. Proactive vehicle routing with inferred demand to solve the bike-sharing rebalancing problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:192–209, 2014.

[24] C. Scott Smith and Joseph P. Schwieterman. E-scooter scenarios: Evaluating the potential mobility benefits of shared dockless scooters in Chicago. Available online., 2018.

[25] Helge Spieker. Constraint-guided reinforcement learning: Augmenting the agent-environment interaction. *CoRR*, abs/2104.11918, 2021.

[26] Peiyu Yi, Feihu Huang, and Jian Peng. A rebalancing strategy for the imbalance problem in bike-sharing systems. *Energies*, 12(13), 2019.

[27] Rui Zhu, Xiaohu Zhang, Dániel Kondor, Paolo Santi, and Carlo Ratti. Understanding spatio-temporal heterogeneity of bike-sharing and scooter-sharing mobility. *Computers, Environment and Urban Systems*, 81:101483, 2020.

Appendix: Configuration of the simulator parameters and short user's guide

ESB-DQN repository on GitHub

`https://github.com/DiTo97/odysseus-escooter-dqn`

Input parameters

Input parameters take into account both simulation parameters of the ODySSEUS environment and parameters for the training/test of the Rainbow agents.

Simulation parameters

Simulation parameters can be found in the folder `esbdqn\configs\escooter_mobility` under the name `sim_conf_<City>.py` with identical structure. Each file comprises two Python objects, named `General` and `Multiple_runs`.

General object

- `city`, name of the city, either Louisville or Austin;
- `relocation_workers_working_hours`, shift hours for relocation workers;
- `bin_side_length`, side length of the square zones each operative area is split into;
- `year`, year of the trip requests to consider;
- `month_start`, `month_end`, start and end of the month of the trip requests to consider;
- `day_start`, `day_end`, start and end of the day of the trip requests to consider;
- `save_history`, whether to save the results CSV after each iteration.

Multiple_runs object

- `n_vehicles`, number of vehicles to spawn in the environment;
- `incentive_willingness`, acceptance probability for each incentive proposal;
- `beta`, battery capacity;
- `alpha`, threshold on the battery level to mark vehicles as out-of-charge in percentage between 0 and beta;
- `battery_swap`, toggle for battery swap events in the environment, either True or False;
- `n_workers`, number of battery swap workers;
- `battery_swap_capacity`, maximum number of vehicles each battery swap worker can process hourly;
- `scooter_relocation`, toggle for relocation events in the environment, either True or False;
- `n_relocation_workers`, number of relocation workers;
- `relocation_capacity`, maximum number of vehicles each relocation worker can move hourly;

All the parameters that have not been modified or are unused with respect to the original ODySSEUS simulator have been omitted here.

Agents parameters

Agents parameters can be found in the file `esbdqn\train.py`. Also, they can be submitted at runtime when launching `esbdqn\train.py` via `CLI`.

– `learning_rate`, learning rate of the Adam optimizer;
– `learn_period`, learning period of the Rainbow agents;
– `batch_size`, batch size of the networks withing the agents;
– `n_steps`, how many steps to look in the past when agents take decisions;
– `max_global_grad_norm`, global gradient norm clipping of the networks weights;
– `importance_sampling_exponent_begin_value` (and similar parameter with `end_value`), range of the importance sampling exponent;
– `replay_capacity`, experience replay buffer capacity. Should amount to about 30 repetitions of any given day;
– `priority_exponent`, priority of the timesteps stored in the experience replay buffer;
– `target_network_update_period`, update period from the online network to the offline network within each agent;
– `num_iterations`, number of training iterations;
– `max_steps_per_episode`, number of trips per episode;
– `num_eval_frames`, total number of validation trips per iteration;
– `num_train_frames`, total number of training trips per iteration (should be at least double the validation trips);
– `n_lives`, total number of lives per iteration; defaults is 50.

Experiment parameters

Each call of `esbdqn\train.py` can be named as a different experiment with its own checkpoints.

– `exp_name`, name of the experiment directory;
– `checkpoint`, toggle on whether to store a checkpoint, either True or False;
– `checkpoint_period`, period of storage of a new training checkpoint.

Output

Run `esbdqn\train.py` to train a new ESB-DQN model from scratch. Otherwise, to train starting from a checkpoint, set the checkpoint toggle to True, and ensure that there is a checkpoint within the experiment directory in the form: `<Experiment_dir> \models\ODySSEUS-<City>`.

Results of each run will be stored as CSV files within the automatically generated directory `<Experiment_dir> \results`.

To reproduce the experiments in the paper:

1. Set `incentive_willingness` to 0 to obtain all the No incentives data.
2. Set `incentive_willingness` to 1 and track the columns `eval_avg_pct_satisfied_demand` and `train_avg_pct_satisfied_demand` from the CSV files for the Validation and Training data, respectively.

All our experiments have been run on Ubuntu 18.04.