# OffNav: Offline Reinforcement Learning for Visual Semantic Navigation

C. Gutiérrez-Álvarez[1], R. Flor-Rodríguez-Rabadán[1], A. Kanezaki[2] and R. J. López-Sastre[1]

***Abstract*— Is it possible to train navigation agents from just human demonstrations? We show this possibility via OffNav framework, an offline reinforcement learning algorithm implemented for visual semantic navigation. We provide a small analysis of its performance on HM3D dataset [1]. We design five experimental setups with incremental difficulty to evaluate the trained policy. We compare our results with the state-of-the-art model PIRLNAV [2], based on behavior cloning. The results show an $8.69\%$ of absolute improvement in the success rate for OffNav agent against PirlNav baseline agent in the most challenging scenario.**

## I. Introduction

The task of delivering Visual Semantic Navigation (VSN) capabilities to real robots in the real world is still a challenge. To teach robots how to navigate in indoor environments, the VSN community has been using online reinforcement learning (RL) algorithms, which require querying environments to learn. This is a problem because querying real environments is expensive and time-consuming, and querying simulated environments is not always a good proxy for real-world performance. Offline RL [3] can be a solution to these challenges by learning policies from a fixed dataset consisting in human demonstrations and their associated reward signals. Therefore, in this work, we propose a novel approach to train VSN agents without ever querying an environment, by leveraging on the Offline RL paradigm. We call this approach **Off**line Visual Semantic **Nav**igation (OffNav).

Technically, we have implemented Implicit Q-Learning (IQL) [4] offline RL algorithm using the decentralized distributed philosophy of DD-PPO [5] to create DD-IQL, a decentralized distributed version of IQL. Our DD-IQL is trained against a fixed dataset containing thousands of human navigation experiences [2]. As depicted in Figure 1, we propose the OffNav approach, capable of efficiently learning the navigation policy required by a VSN agent from human demonstrations. Subsequently, these policies can be deployed across various scenarios, and if necessary, further refined through online RL for more specific tasks.

To demonstrate the capabilities of our implementation, we carried out a small analysis of its performance using different environments from HM3D dataset [1]. Preliminary results shows that our DD-IQL implementation is able to learn navigation policies effectively. To the best of our knowledge, this is the first time that an offline RL algorithm is implemented for VSN and large environments, predicting actions directly from raw input observations.

[1]University of Alcalá, Department of Signal Theory and Communications, Spain. Email: carlos.gutierrezalva@uah.es

[2]Tokyo Institute of Technology, Department of Computer Science, School of Computing, Japan.

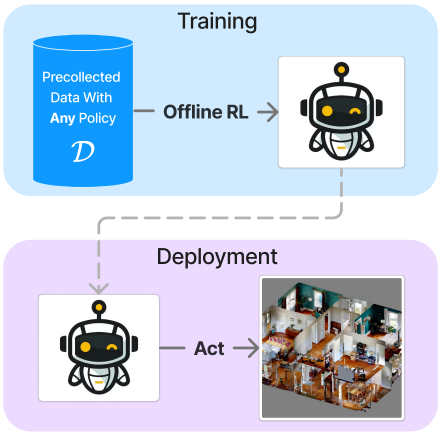


Fig. 1: By leveraging on the offline reinforcement learning paradigm, we can train agents from a fixed dataset of navigation experience, without querying any environment. This opens the possibility to create many navigation datasets from any navigation agent in any **real or simulated** environment, and then use them to train new agents for different scenarios without the need to ever query that environment.

## II. Offline Visual Semantic Navigation

In this work, we study ObjectNav navigation [6], a setup in which an agent is asked to navigate to a target object in an environment. To perform this task, the agent does it using only egocentric perceptions. Specifically, the agent receives RGB images and GPS+Compass information that provides the agent with the current position and orientation relative to the starting point. The set of movements is discrete and consists of the following actions: TURN_LEFT, TURN_RIGHT, MOVE_FORWARD, LOOK_UP, LOOK_DOWN and STOP. If the agent spawns the STOP action within $1m$ Euclidean distance respect to the target object within a 500 steps time limit, the episode is considered successful. In the other case, it is considered a failure. The success rate (SR) is measured by averaging the success over all the episodes present in an evaluation set.

Since we are on an offline RL setup, we need a previously collected dataset of navigation experience. The dataset that we chose is collected in [2]. It consists of $77k$ episodes of human navigation trajectories using the HM3D [1] dataset.

We train our policies using our DD-IQL implementation on the human demonstrations. The objetive is to find a policy with optimal parameters $\phi^*$ that maximizes the expected

return from the dataset. To do so, the IQL algorithm relies on the use of expectile regression to modify a temporal-difference (TD) loss. This modified TD loss is able to learn an approximate Q-function from the dataset actions. This Q-function does not explicitly represent the corresponding policy, so a separate policy extraction step is needed. For policy extraction, we use advantage-weighted regression [7], [8]:

$$L_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\exp\left(\beta\left(Q_{\hat{\theta}}(s,a) - V_\psi(s)\right)\right)\log\pi_\phi(a|s)\right], \quad (1)$$

where $\beta \in [0, \infty)$ controls the trade-off between cloning the expert policy and maximizing the Q-function. This loss can be seen as a selection of most optimal actions to clone in the dataset. We also employ inflection weighting [9] to modify the loss function, thereby giving more importance to those time steps where there is a change in actions.

For the policy architecture, we use a simple CNN+RNN model from [2]. The difference is that we use ResNet18 for the visual encoders. We copy the same architecture for the policy net, the Q net and the Q target net. For the V net, we only use the visual encoder and a single linear layer, without any recurrent module.

## III. Experiments and Results

Is an offline RL algorithm able to learn navigation policies effectively? To answer this question, we have trained our DD-IQL model using the expert demonstrations on five different experimental setups. These setups have been designed with an incremental difficulty. The first three are evaluated on the same environments in which the agents were trained, while the last two are evaluated on different environments. The details of the setups are depicted on figure 2.

We compare our results with the current state-of-the-art model PirlNav [2]. This model is based on a two-phase training schedule. The first phase is a supervised learning phase, where the model is trained using behavior cloning on the expert demonstrations. The second phase is a reinforcement learning phase, where the model is fine-tuned using DD-PPO algorithm [5]. For a fair comparison, we train the PirlNav agent using only the behavior cloning phase on the same setups as our OffNav model.

Results are shown on table I. It can be seen that both methods obtain similar performance on setups 1 to 3. Offnav method outperforms PirlNav on setup 2, while PirlNav outperforms OffNav on setup 3, and both of them obtain $100\%$ SR on setup 1. When evaluated on setup 4, PirlNav outperforms OffNav by $2.27\%$ absolute points. However, on setup 5, the most challenging one, OffNav outperforms PirlNav by $8.69\%$ absolute points.

| *Experimental Setup* | *OffNav* | *PirlNav* |
|---|---|---|
| Setup 1 | 100% | 100% |
| Setup 2 | **79.31%** | 72.50% |
| Setup 3 | 75.78% | **77.63%** |
| Setup 4 | 25.00% | **27.27%** |
| Setup 5 | **34.78%** | 26,09% |

TABLE I: Success Rate for OffNav and PirlNav methods on the five experimental setups.

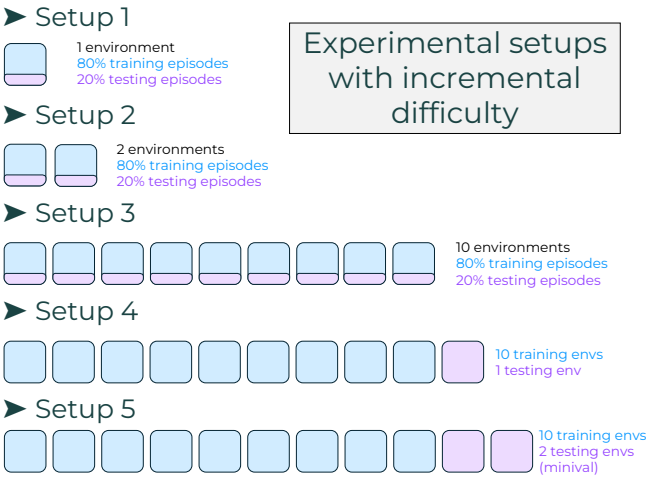


Fig. 2: Five experimental setups designed with an incremental difficulty.

## IV. Conclusions and Future Work

From the results obtained in the experiments, we can conclude that the proposed OffNav method is able to learn navigation policies effectively from human demonstrations. It can also be seen that the method is able to generalize to unseen environments, as shown in setups 4 and 5, and outperform the state-of-the-art model PirlNav [2] in the most challenging one. Future work will focus on training the policy with more diverse environments to improve its generalization capabilities and further extend this analysis.

## References


[1] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, "Habitat-Matterport 3D Dataset (HM3D): 1000 large-scale 3D environments for embodied AI," in *NeurIPS*, 2021.

[2] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "PIRLNav: Pretraining with Imitation and RL Finetuning for ObjectNav," in *CVPR*, 2023.

[3] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems," *arXiv:2005.01643 [cs, stat]*, Nov. 2020.

[4] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *ICLR*, 2022.

[5] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "DD-PPO: Learning Near-Perfect PointGoal Navigators from 2.5 Billion Frames," in *ICLR*, 2020.

[6] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects," in *arXiv*, 2020.

[7] J. Peters and S. Schaal, "Reinforcement learning by reward-weighted regression for operational space control," in *ICML*, 2007.

[8] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," 2019.

[9] E. Wijmans, S. Datta, O. Maksymets, A. Das, G. Gkioxari, S. Lee, I. Essa, D. Parikh, and D. Batra, "Embodied question answering in photorealistic environments with point cloud perception," in *CVPR*, 2019.