Offline Reinforcement Learning for Efficient and Realizable Fertilization Policies

Mert Erkul, Dr. Luca Corinzia, Scott Sussex, Dr. Matteo Turchetta, Prof. Andreas Krause
{merkul, luca.corinzia, scott.sussex, matteo.turchetta, krausea}@ethz.ch

***Abstract*—Usage of nitrogen in fertilizers has been a key strategy to increase yields in crop management for several decades. Common practice among farmers is to adjust the application of nitrogen-based fertilizers depending on the environmental conditions, the nitrogen amount added so far and the day of the year. In turn, crop management can be modelled as an optimal control problem, in which actions, observations, episodes and rewards are explicitly defined. Reinforcement learning (RL) has gained immense popularity among such problems, prescribing policies which yield better rewards than the status-quo policies. RL is inherently an online learning paradigm, where agents require numerous interactions with the environment. In high-risk tasks such as usage of nitrogen-based fertilizers, interacting with the environment can be expensive and detrimental. For tasks where records of transactions exist, and agent interaction is restricted, the notion of offline RL has emerged. In this study, we apply model-free offline RL algorithms to find near-optimal policies for fertilization using nitrogen. We show that offline RL strategies can easily surpass the existing expert policies, demonstrate the reward increase with larger state spaces, illustrate the policies that are prescribed by the trained agents, and compare them with the expert policies.**

I. Introduction

The exponential growth in world population has forced modern agriculture to its limits around various regions on earth [1]. Acknowledging this challenge, this project focuses on optimizing crop management through modern artificial intelligence (AI) techniques, specifically, offline reinforcement learning. Reinforcement learning (RL) provides mathematical foundations for reward-directed optimization through interactions with the environment to achieve learning-based control [2]. Because of the dependency on environment interaction, inherently, traditional RL is an "online" learning paradigm [3]. However, in realistic settings, such as crop management, interacting with the environment is infeasible, expensive, or dangerous. While some recent papers investigate application of RL in the agricultural domain [4], they do so through interactions with an environment simulator for policy optimization. Instead, we consider studying the offline setting through static data obtained from a simulator. We accomplish this with the help of a modified Open AI Gym setup based on Cycles [5], which simulates biophysical processes and crop management progresses within cropping systems. By avoiding interactions with the environment during optimization, we investigate different offline RL algorithms in the literature for crop management. We aim to find policies that are better than the status-quo expert policies, while also trying to maintain sample efficiency as well as prescribing policies that are realizable.

The experiments we conduct and our goals are five-folds, (i) finding the most successful offline RL algorithm that can surpass the existing expert policies, (ii) finding the best static dataset generation strategy (i.e. behavior policy) for model performance, (iii) comparing the value of added information through agents trained in fully observable and partial observable environments, (iv) finding a sweet-spot for sample efficiency, to ensure success in realistic settings where static datasets can not be generated through a simulator and are highly sparse, through comparisons of model performances, using different dataset sizes, and finally (v) experimenting with offline-to-online fine-tuning in order to realize settings where both static datasets and simulators are simultaneously available.

II. Related Work

a) Machine Learning and Reinforcement Learning in Crop Management & Agriculture: With the increase in interest for data-driven strategies, many new approaches have emerged to achieve not only autonomy, but also efficiency in the agriculture domain [1]. Machine Learning (ML) is used in the literature in various detection and prediction tasks, from forecasting the end-of-year yields [6], to weed detection and crop protection [7]. There also exists several survey papers [8], [9] to convey and summarize the usages of ML in agriculture.

When it comes to iterative strategy selection or optimal control, RL is frequently used in domains such as robotics, autonomous driving or game playing [3]. However, RL strategies require an environment to interact with, and until the recent years, such virtual simulators that have RL-oriented interfaces were not available for agriculture or crop management. With the success obtained through deep RL and with the increase of different crop growth models (CGM), such as DSSAT [10] and Cycles [5], there has been extensive works to wrap CGMs to behave like OpenAI gym [11] setups to develop RL strategies [12]. Utilizing these wrapped-CGM environments, there have been studies for RL for fertilization policies [13] and RL for irrigation strategies [14], however, most of these studies fail to focus on prescription of realistic policies and in the end mostly express that the trained agents act very frequently for the policy prescribed to be realizable [12].

b) Offline RL, Techniques and Challenges: In many real-world scenarios, online interactions are expensive, but large batches of historical data are available. Consequently, the historical data available are assumed to be collected by following a behavioral policy (π_β, sometimes also referred to as the data-generating policy) which may or may not be known, generating several challenges. One of these challenges is the as "distributional shift", such that the model might be used for evaluation of parametric policies (π_θ) under a non-observed distribution. Distributional shift is shown to be detrimental to the performance of policies in Markov Decision Processes (MDP) [15], therefore, it is the most widely studied and understood challenge in modern offline RL [16]. Some techniques such as off-policy evaluation via importance sampling [17], off-policy policy gradient methods with different regularization strategies and/or penalties for reward functions or (action) value functions [18], [19], pessimistic value-based algorithms [16] and policy constraint algorithms [20] are introduced to solve existing challenges.

Specifically, policy constraint methods ensure the policies that are found by the algorithm are "close enough" to the behavior policy to avoid introducing errors by generating policies that suggest out-of-distribution actions. For our domain, it is important to note that recommending out-of-distribution policies such as including extensive amounts of nitrogen to the soil might not only be detrimental to the yield, but also to the environment, eventually causing implicit and explicit negative rewards. Therefore, we focus on studying model-free offline RL techniques that are based on policy constraint techniques.

III. Environment, Static Dataset Generation & Algorithms

To utilize offline RL strategies, one requires to create static datasets using a behavior policy, saving the actions, observations, rewards and terminal states in a tabular format. In this section, we elaborate on the different data collection strategies, environment, and the offline RL strategies used to train the agents.

A. Environment

Our environment is a modified version of Cycles [5]. Environment initialization requires a crop file, a weather file and a soil file, as well as determining a year for selecting the appropriate conditions. For this study, we used the soil file obtained from "Hagerstown", the default crop file for corn management and two different weather conditions obtained from "Rock Springs" (RS) and "New Holland" (NH).

The episodes are fixed to be one year long, with weekly transitions, and the episode terminations are set at the last week of the year. Hence, every episode consists of 53 steps. Rewards are determined through 2020 US[1] corn prices per tonne, while penalizing for the amount of fertilizers, using the average anhydrous ammonia prices obtained in 2020[2]. For the amount of fertilizers used per time step (i.e, the action), we set a threshold to 150 kilograms per hectare, and discretized the action space with bins of 15 kilograms per hectare, causing the discrete action space to contain values between [0, 10]. For the fully observable environment, the state space includes 27 observations, related different to crop conditions, including nitrogen fixation, cumulative biomass, water stress, plant height as well as environment conditions such as wind, and screening height. In the partial observable setting, we use only the last two dimensions, which are the day-of-year and the nitrogen added so far.

B. Static Dataset Generation

As behavior policies, we selected random perturbations of the expert policies specified by Cycles and Agroscope. Exact expert policies for Cycles and Agroscope prescribe fertilization using 150 kilograms of nitrogen per hectare on the 110th day, and fertilization using 35 kilograms of nitrogen per hectare on the 110th day and fertilization using 120 kilograms of nitrogen per hectare on the 155th day, respectively. For these expert policies across weeks in a single year, see blue and orange curves in Figure 5. After discretizing these action-sequences to be 53-dimensional vectors, to generate the behavior policies, we modified the amount of N used by $\pm$ 20% for all peaks, as well as, changed the fertilization dates by $\pm$ 2 weeks for all peaks. The behavior policy was determined through selecting a random perturbation via a Bernoulli trial, where all possible random perturbations of the selected expert policy have uniform probability to be selected. The selected behavior policy was applied to a random configuration of the environment between years 1980 and 2000, using a random selection of RS or NH as the weather file. In total, 100000 different episodes were generated for both policy perturbations and the action, observation, reward and terminals were recorded. The 3-D plots, which indicate the perturbed policies (with respect to the action taken versus it's week number) and their frequencies in the z-axis, can be observed in Figure 1. Note that, with all possible perturbations, weeks between 11 and 27 are the only non-zero action containing weeks. Hence, the 3-D plots contain only the informative weeks.

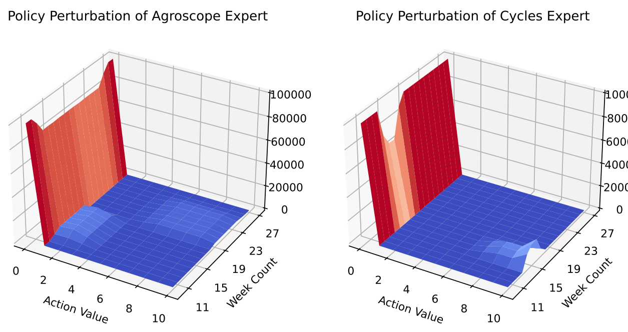


Fig. 1. Policy perturbations to generate offline datasets

C. Algorithms

For training the agents using the offline datasets, we chose four different model-free offline reinforcement learning algorithms, and used their versions for discrete control. These are

[1]https://quickstats.nass.usda.gov/results/BA8CCB81-A2BB-3C5C-BD23-DBAC365C7832

[2]https://farmdocdaily.illinois.edu/2021/08/2021-fertilizer-price-increases-in-perspective-with-implications-for-2022-costs.html

Behavior Cloning (BC) [21], Conservative Q-Learning (CQL) [19], Batch-Constrained Q-Learning (BCQ) [22] and Implicit Q-Leaning (IQL) [23]. We also experimented with Soft Actor-Critic (SAC) [24], Advantage Weighted Actor Critic (AWAC) [25] and Temporal Difference to Behavioral Cloning (TD3-BC) [26], specifically, their versions for discrete offline RL. However, these yielded suboptimal rewards in all configurations, thus, we exclude them in the following subsections. We believe that one possible reason is the sparsity of the behavior (data-generating) policy. At best, the action space consists of two peaks (an action is conducted only twice out of possible 53), causing some models to easily collapse in a highly-suboptimal action space, failing to conduct meaningful gradient updates.

IV. Experiments & Results

Using the datasets described in Section III-B, we train the models using different combinations of sample sizes, data generating policies, different state-spaces and iteration counts. We start with a more coarse-grained aggregation of results in Section IV-A.

A. Algorithms Compared

In this section, we aggregate the trained models across different sample sizes, different state observabilities, different iteration counts and behavior policies used for the training datasets. The training set contains episodes from both locations (NS and RS), selected randomly for years between 1980 and 2000. Initially, we do the most coarse-grained aggregation to observe which behavior policy generates the most reward yielding models on average, through all possible variations. The aggregation of model performances is done per strategy and per weather file, across different years. One can observe the average model performances, with $\pm$ 1 standard deviation, indicating the change with respect to the experiment configuration and their comparisons with exact expert policies in Table I.

Model Name	RockSprings Averages	NewHolland Averages
CQL	**1807.51 $\pm$ 195.28**	**1806.29 $\pm$ 171.86**
IQL	1295.29 $\pm$ 488.55	1165.79 $\pm$ 390.07
BCQ	1634.27 $\pm$ 240.25	1686.79 $\pm$ 270.29
BC	1652.78 $\pm$ 226.39	1725.35 $\pm$ 263.47
Agroscope	1873.14	1729.52
Cycles	**1895.15**	**1754.18**

TABLE I
A comparison of model performance and expert policies over two weather conditions, averaged over different sample size trained models

Here, we can observe the superiority of Cycles expert to Agroscope expert on average, for both locations. Although the model performances that are aggregated to generate Table I also include configurations where sample sizes and iteration counts are too small to perform decently, as well as training sets generated via partially observable states, CQL shows competitive performance with the experts. IQL, being originally designed as a continuous control algorithm, shows the worst performance, which also demonstrates high standard deviations across years for both locations, and is thus the least robust model. We believe that this is due to the discretization of the action space, and its sparsity. BCQ and BC perform similarly, and worse than the experts and CQL. For a detailed year-wise comparison of models and experts, one can observe Figure 2. The dashed orange line indicates the cut-off point for the dataset generation, i.e. it indicates that the static datasets for training were generated by configuring the environment using years between 1980 and 2000, for both RS and NH, years after 2000 were used only during evaluation.

B. Different Training Behavior Policies Compared

The static datasets were generated using strategies conveyed in Section III-B, and in this section, we analyze and aggregate the model performances based on the training dataset used, with different sample sizes and iteration counts. In particular, we have four different setups, namely, (i) perturbations of Agroscope expert policy, (ii) perturbations of Cycles expert policy, (iii) equal-sized concatenations of (i) and (ii), finally (iv) unique year-perturbation combination using Agroscope expert policy perturbations. For (iv), it is useful to note that in total, there are 882 unique perturbations of the Agroscope expert, given that there are 21 different years and 2 different weather conditions, the total dataset size is 37044 instead of 100000, containing only unique episodes. The aggregated model performances with respect to different data generating policies can be seen in Table II.

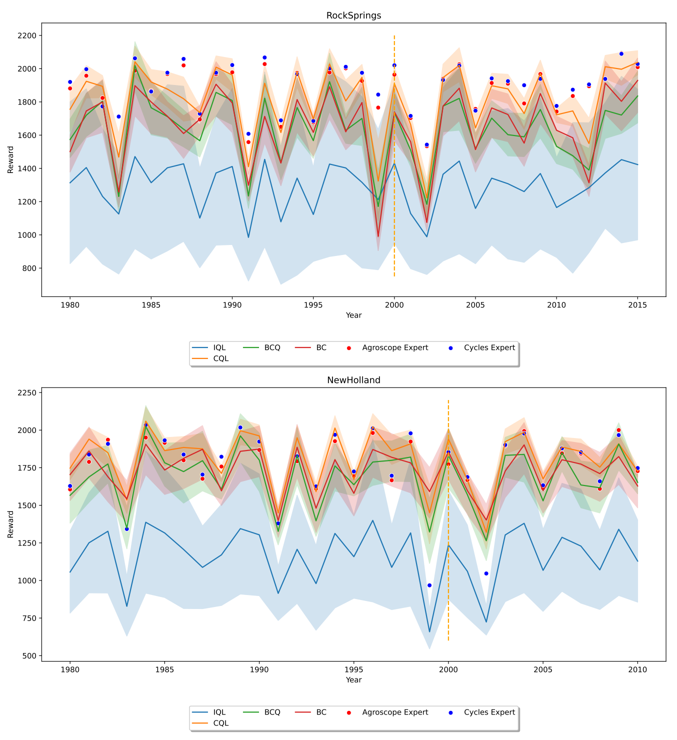


Fig. 2. Algorithm performances across years

Model-Location Name	Agroscope	Cycles	Mixed	Agroscope-Smart
CQL-RS	**1913.12±239.5**	1724.25±254.46	1759.75±262.8	1842.13±329.72
BCQ-RS	1680.46±340.89	**1723.67±256.58**	1698.25±341.73	1512.12±346.96
BC-RS	**1763.5±297.85**	1763.04±261.15	1361.0±390.11	1668.19±318.5
CQL-NH	1781.15±294.61	**1858.55±152.88**	1809.93±211.6	1847.41±311.15
BCQ-NH	1579.73±346.55	**1835.88±164.89**	1818.34±249.86	1569.49±349.16
BC-NH	1783.58±242.46	**1884.73±150.31**	1419.17±419.24	1814.71±260.04

TABLE II
MODEL PERFORMANCES WHEN TRAINED WITH DIFFERENT BEHAVIOR POLICIES TO GENERATE DATASETS

The bolded results show the best training dataset for every model location combination. On average, the static dataset generated via the perturbations of Cycles expert policy causes the models to perform better. This result is expected, and is consistent with the results obtained in Table I, as Cycles expert policy surpasses the Agroscope expert policy on average. However, while conducting the more fine-grained experiments, where the models are averaged over specific sample sizes, dimensionalities, iteration counts and data-generating policies, we observed that with good configurations and hyperparameter tuning, models trained with Agroscope expert perturbations perform the best. We believe that this is because there are more possible perturbations of the Agroscope expert, and well-configured models can generalize better when exposed with a variety of data generating policies. Thus, in the following sections, we will use the Agroscope perturbations for the training sets, but when comparing with experts and scoring the models in Section IV-F, we will also use rewards obtained by the Cycles expert policy perturbations for comparison and scoring.

C. Partial Observability in State Space

The expert policies we experimented with in this study are determined solely by the day-of-year and the N-so-far. No matter the crop conditions, the weather nor the year, expert policies of Agroscope and Cycles are assumed to fertilize the crops in pre-determined dates. To verify the value of the additional information through the remaining state dimensions, we conducted a comparative study. For several sample size and iteration count configurations, we trained the models with the same configurations twice, once using only nitrogen-to-date and day-of-year states (partially observable, 2-D) and once using all states (fully observable, 27-D). We subtracted the partially observable rewards from the fully observable rewards and aggregated them based on the year. The complete results can be seen in Table III.

Model Name	RockSprings	NewHolland
ΔCQL	447.69±400.31	474.5±433.08
ΔBCQ	405.97±421.97	414.39±417.39
ΔBC	237.32±364.36	253.2±420.47

TABLE III
MODEL PERFORMANCE CHANGES ACROSS LOCATIONS, PARTIAL OBSERVABLE SETTING SUBTRACTED FROM FULLY OBSERVABLE SETTING

In all aggregated configurations, we can see that on average, with the fully observable setting, the rewards obtained through prescribed policies increase. This is expected as in the partial observable setting with 2-D states, the agents learn to fertilize with respect to the day-of-year, converging to the average reward in the training set. CQL and BCQ show reward increases more than 400 in both locations. Considering the maximum reward achieved through the experiments in the study is approximately 2400, this increase is quite significant. One can also see that the average reward increase for NH is larger than RS. We believe that this is due to bad weather conditions observed for some years in NH (can be seen more detailed in Figure 4), making the additional information even more valuable for the trained agents.

D. Sample Efficiency

Sample efficiency is a major challenge for all types of reinforcement learning paradigms, whether they are model-based or model-free, or whether they are online or offline. Finding an optimal or a near-optimal policy requires numerous interactions with the environment for online RL, or large amounts of static episodes obtained via a near-optimal data generating policy for offline RL. In this extent, using once again the dataset generated by perturbing the Agroscope expert policy (i.e. dataset specified in Section IV-B (i)), we conducted a comparative analysis using CQL, BCQ and IQL. Controlling for the number of episodes in the dataset, we altered the number of iterations (epoch counts) to see model performances for a given number of samples. For the sample size in the training set, we selected the following values [1000, 5000, 10000, 25000, 50000, 100000]. The aggregations were done with respect to different years and iteration counts. The comprehensive collection of results can be seen in Table IV, as well as the line plots demonstrating the average rewards versus sample sizes compared against the expert rewards as horizontal lines in Figure 3.

BCQ and IQL show varying performances across sample sizes in RS, whereas CQL shows a relatively expected and smooth performance, conveying the sample inefficiency challenge. In NH, BCQ also behaves similarly to CQL, however, still underperforming, such that the upper confidence level for the average rewards obtained across all sample sizes are less than CQL's lower bound. IQL proves to be unstable in discrete environments in both NH and RS. It is unrealistic to obtain static datasets with large sample sizes without a simulator, and the results show that even for 1000 years, a number still quite unlikely and large to realize, all models fail to outperform the

Model-Location Name	1000	5000	10000	25000	50000	100000
CQL-RS	1201.7±99.23	1798.4±148.69	1866.05±157.85	1893.55±153.87	1962.35±180.05	**2020.64±205.62**
BCQ-RS	1298.17±135.99	1610.89±149.88	1421.37±141.36	**1795.74±184.22**	1743.91±161.14	1640.8±197.83
IQL-RS	1490.55±186.45	1327.04±152.06	1691.38±221.58	1187.59±124.71	**1745.99±210.77**	1438.97±160.99
CQL-NH	1276.93±164.64	1626.98±229.36	1688.63±254.33	1734.04±291.64	1777.9±288.79	**1794.82±311.47**
BCQ-NH	1132.81±152.6	1395.01±218.28	1446.85±225.23	1518.6±226.52	**1569.31±258.14**	1552.32±267.97
IQL-NH	1291.08±256.17	1175.5±206.29	1484.46±305.04	1076.52±167.96	**1522.79±295.59**	1282.46±221.01

TABLE IV
SAMPLE EFFICIENCY COMPARED ACROSS MODELS AND LOCATIONS

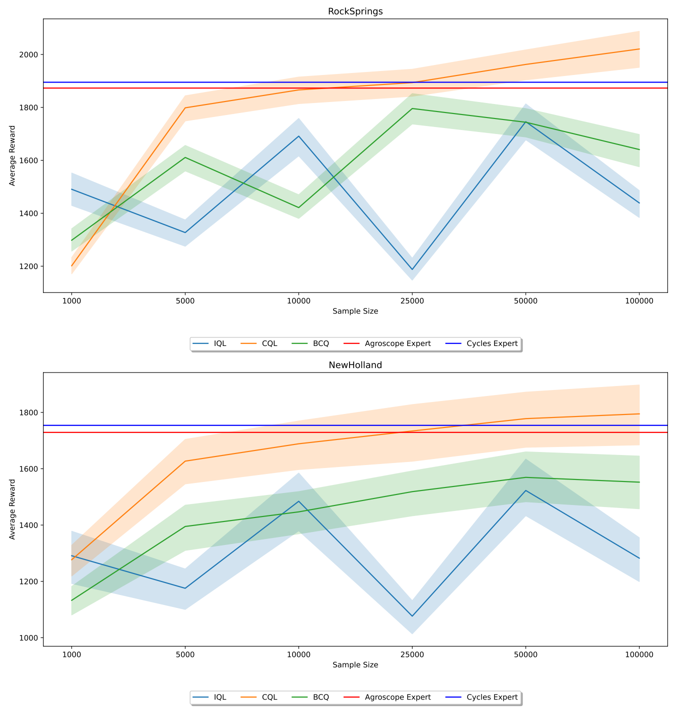


Fig. 3. Sample efficiency performances of models across locations

experts on average, and require around 5000 training episodes with more than 100 iterations to surpass them.

E. Offline-to-Online Fine-Tuning

Several recent studies in the literature try to fine-tune the offline-trained models by enabling interaction with the environment and continuing training in an online fashion, using a pre-determined number of episodes [25], [27], [23]. A theory or a rule of thumb does not exist for the ratio between the number of episodes for offline training and for online training, and the works demonstrate varying results. Thus, we decided to experiment with three candidate models with different epoch counts and sample sizes. We only used CQL for offline-to-online fine-tuning, since other models such as AWAC [25] and IQL [23] performed with extreme variance on discrete action spaces when trained in an offline only fashion. CQL-1 consisted of a training dataset with 100000 episodes and was trained for 50 epochs, CQL-2 consisted of a training dataset with 5000 samples and was trained for 100 epochs, and finally CQL-3 consisted of a training dataset with 1000 samples and was trained for 150 epochs. When trained only in an offline fashion, all models showed comparative average performances, however, when interaction with the environment was allowed, the performances were completely detrimental for CQL-1. Full results, across different online episode counts, can be observed in Table V.

Model Type	Offline	1000	2500	5000	10000
CQL$_{1-RS}$	2061.82	11.66	-783.59	-888.99	-122.26
CQL$_{2-RS}$	2016.71	**2031.30**	**2075.11**	1430.18	1288.11
CQL$_{3-RS}$	1926.57	1536.27	1531.31	1050.16	1921.88
CQL$_{1-NH}$	1778.62	1220.96	-1069.60	-1213.61	-546.40
CQL$_{2-NH}$	1828.34	**1888.26**	**1852.00**	1379.31	1081.54
CQL$_{3-NH}$	1743.81	1478.11	**1847.61**	842.72	1735.91

TABLE V
OFFLINE-TO-ONLINE FINE-TUNING ON DIFFERENT CQL CONFIGURATIONS

We observed that the model trained with the massive dataset suffered from offline-to-online training, collapsing completely to negative values on average. Here it is beneficial to note that when one doesn't use any fertilizers (i.e. zero sequence for all 53 time steps), they can obtain a reward between 600-800 in both weather conditions, depending on the year. A negative reward on average shows that the model collapsed on a highly suboptimal region for the action space. This result was also observed in M-CQL and IQL papers [27], [23], authors note that for some environments and models, online fine-tuning causes some models to collapse to highly suboptimal regions. We see that the second CQL configuration actually benefits from online fine-tuning, when a few online iterations (1000 or 2500) are used. What is surprising for CQL-2 are the prescribed policies after online fine-tuning. In Figure 5, one can see that this model actually prescribes a completely different policy from its offline counterpart, such that the initial small fertilization peak (35 kilograms of N-per-hectare) is done either in the first week or the second, and the large fertilization peak is done earlier than what Agroscope expert prescribes. The third configuration performance initially decreases, and starts increasing with increasing number of online iterations. We assume that this is due to the under-training in the offline setting for CQL-3.

1) Challenges Faced in Offline-to-Online Fine-Tuning: Catastrophic Interference (CI) is a common problem in multi-task deep reinforcement learning, especially when one tries to find a common generalization for a near-optimal policy in different environments [28], [29]. CI is often referred to situations in which the models perform decently on the environment configuration it was most recently trained with, and perform poorly in other configurations. It is an emerging research question in the RL domain, observed both in model-free and model-based RL techniques [30], and some solutions have been proposed to prevent CI. Solutions use either simple techniques such as narrowing the Q-Networks [29] or more complex techniques, which incorporate knowledge distillation [4] in the RL algorithm. We also encountered this phenomenon while initializing the online environment randomly and performing the gradient updates after numerous steps, causing the model to perform well on RS and poorly on NH, or vice-a-versa. To bypass this issue, we conducted the online gradient updates immediately after the episode terminated, so that the online tuned models showed similar performances across different environment configurations.

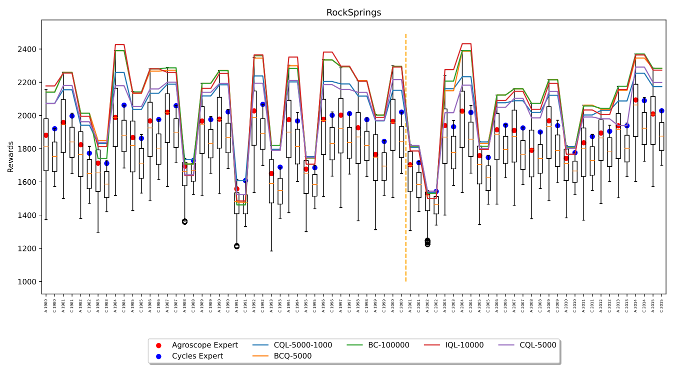
F. Best Configurations and Policies Prescribed

Throughout the study, we trained BCQ, CQL, IQL and BC with various configurations. In this section, we selected the best performing on a score, rather than the average reward the trained agents obtain. This score assigns +1 when a trained agent surpasses the Cycles expert for a selected environment configuration (i.e. for different years and weathers). As we have 36 different configurations for RS and 31 for NH, with two expert policies to compare (Agroscope and Cycles), in total, the maximum achievable score equates to 134. We selected the highest scoring configuration for each offline RL strategy, as well as the best model from offline-to-online fine-tuning. We resolved ties with respect to the sample efficiency, if two different configurations for the same strategy got the same score, we selected the one that was trained with fewer samples and for fewer epochs. The aggregated results can be seen in Table VI, whereas individual comparisons in all possible 67 configurations can be observed in Figure 4. The box plots indicate the reward distributions attainable by using the expert policy perturbations, and colored dots show the rewards obtained by the exact expert policies. Similar to Figure 2, in Figure 4 we also use the dashed orange line to indicate the separation for environment configurations, trained agents have not been trained on an environment configured with the weather and crop conditions after year 2000, for both RS and NH.

Model Name	RockSprings Averages	NewHolland Averages
CQL-5000-1000	2031.30±187.16	**1888.26±306.15**
BCQ-5000	**2091.42±238.32**	1835.33±343.88
BC-100000	2091.31±245.92	1843.56±349.16
IQL-10000	2086.5±258.25	1832.71±349.16
CQL-5000	2016.71±196.99	1828.34±338.11
Agroscope	1873.140	1729.520
Cycles	1895.150	1754.180

TABLE VI
BEST PERFORMING MODELS AND CONFIGURATIONS

We can see the highest scoring configurations surpass the expert policies on average (Table VI) and consistently in



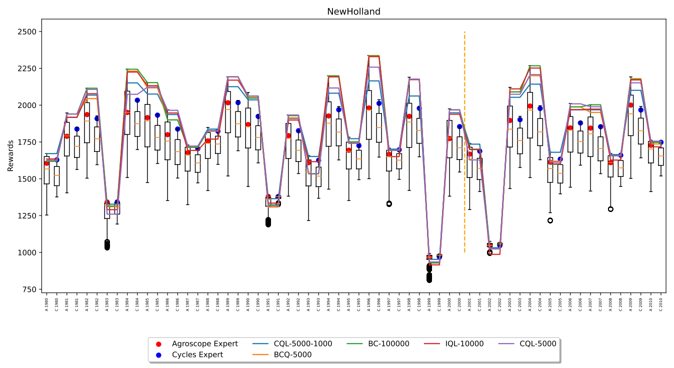


Fig. 4. Best model performances compared with expert distributions and expert policies across years

individual years (Figure 4), not only in RS, but also in NH. As previously mentioned in Section IV-F, the best performing combinations were all trained using the Agroscope perturbations for all algorithms, and although some models trained by only Cycles perturbations got competitive rewards, Agroscope dataset trained models got the highest scores. BCQ and CQL are both algorithms that rely on policy constraint strategies, thus, we expect that the prescribed policies to be similar to the Agroscope expert policies, having two peaks where the former peak to be smaller than the latter. To observe and analyze the prescribed policies by the trained agents, we recorded the 53-D action vectors for all possible environment configurations, which can be seen in Figure 5.

The actions prescribed by the trained agents are stochastic, and depend on the environment condition as expected. The signals resemble the Agroscope expert, also as expected, fertilizing one or two weeks later for both of the peaks on average. One important point that stands out from 5, is that all the policies prescribed by the trained agents are “realizable”. When online RL algorithms were used, it is noted [13] that the algorithms fertilize frequently, which is infeasible in realistic settings. It is possible to claim that the “realizability of the prescribed policies” was mostly satisfied through offline RL.

V. Conclusion

In this project, we aimed to generate realizable policies through various offline RL strategies for crop fertilization. Although obtaining high rewards across episodes is paramount, we wanted to achieve this bypassing several issues encountered

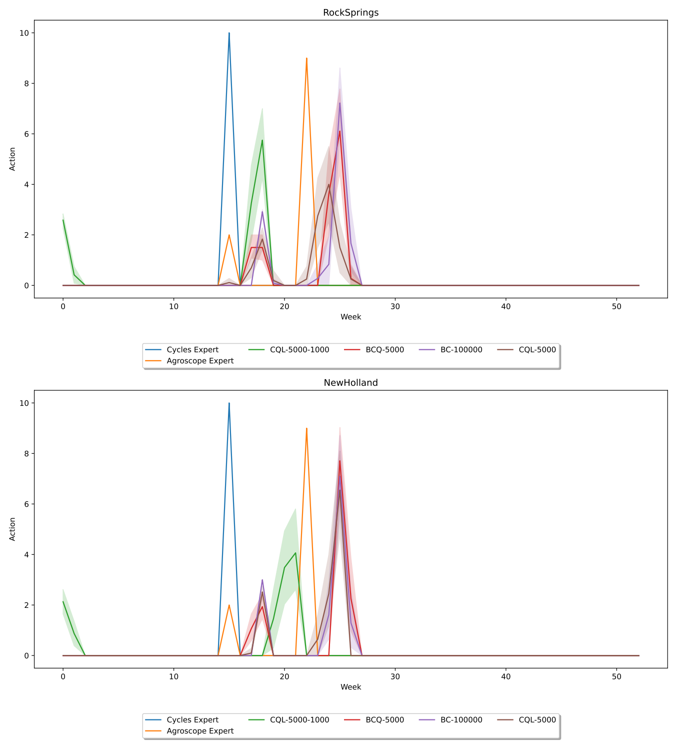


Fig. 5. Actions taken by best models compared with expert actions

in modern RL, such as sample inefficiency, distribution shifts and catastrophic interference. In addition, we also wanted to verify the value of added information through comparisons between fully observable and partially observable environments. Throughout our trajectory, we decided to try some recently emerging paradigms in offline RL, such as offline-to-online fine-tuning. This technique is realizable in our domain, since a wrapped-CGM exists to fine-tune models that have been trained using offline and static datasets.

For future work, we would like to experiment with model-based offline RL to see whether if predicting the MDP (or POMDP) dynamics would increase the rewards, or would result in better sample efficiency. We would also like to see the model performance across different types of crops, rather than focusing solely on corn, and use different soil files to configure the environment, to further increase generalization performance. In addition, another future direction might be the following; by using states that are compatible with other wrapped-CGMs (such as gym-DSSAT [12] or CropGym [13]), expanding the experiment setup and comparing the model performances across different environments.

References

[1] J. Binas, L. Luginbuehl, and Y. Bengio, “Reinforcement learning for sustainable agriculture,” in *ICML 2019 Workshop on Climate Change: How Can AI Help?*, 2019. [Online]. Available: https://www.climatechange.ai/papers/icml2019/32

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[3] S. Levine, A. Kumar, G. Tucker, and J. Fu, “Offline reinforcement learning: Tutorial, review, and perspectives on open problems,” 2020.

[4] W. Zhang, X. Cao, Y. Yao, Z. An, X. Xiao, and D. Luo, “Robust model-based reinforcement learning for autonomous greenhouse control,” 2021.

[5] C. White, Y. Shi, and A. Kemanian, “An agroecosystems simulation model,” Nov 2021. [Online]. Available: https://psumodeling.github.io/Cycles/

[6] D. Paudel, H. Boogaard, A. de Wit, S. Janssen, S. Osinga, C. Pylianidis, and I. N. Athanasiadis, “Machine learning for large-scale crop yield forecasting,” *Agricultural Systems*, vol. 187, p. 103016, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308521X20308775

[7] F. K. van Evert, S. Fountas, D. Jakovetic, V. Crnojevic, I. Travlos, and C. Kempenaar, “Big data for weed control and crop protection,” *Weed Research*, vol. 57, no. 4, pp. 218–233, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/wre.12255

[8] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis, “Machine learning in agriculture: A comprehensive updated review,” *Sensors*, vol. 21, no. 11, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3758

[9] N. Hegde, S. Jambarmath, M. R P, and R.P., “Survey paper on agriculture yield prediction tool using machine learning,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 5, pp. 36–39, 11 2017.

[10] J. Jones, G. Hoogenboom, C. Porter, K. Boote, W. Batchelor, L. Hunt, P. Wilkens, U. Singh, A. Gijsman, and J. Ritchie, “The dssat cropping system model,” *European Journal of Agronomy*, vol. 18, no. 3, pp. 235–265, 2003, modelling Cropping Systems: Science, Software and Applications. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1161030102001077

[11] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” 2016.

[12] G. Romain, P. E. J., P. Philippe, B. Julien, M. Odalric-Ambrym, and E. David, “gym-dssat: a crop model turned into a reinforcement learning environment,” 2022. [Online]. Available: https://arxiv.org/abs/2207.03270

[13] H. Overweg, H. N. C. Berghuijs, and I. N. Athanasiadis, “Cropgym: a reinforcement learning environment for crop management,” 2021. [Online]. Available: https://arxiv.org/abs/2104.04326

[14] L. Sun, Y. Yang, J. Hu, D. Porter, T. Marek, and C. Hillyer, “Reinforcement learning control for water-efficient agricultural irrigation,” in *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, 2017, pp. 1334–1341.

[15] S. Ross, G. J. Gordon, and J. A. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” 2011.

[16] Y. Jin, Z. Yang, and Z. Wang, “Is pessimism provably efficient for offline rl?” 2021.

[17] D. Precup, R. S. Sutton, and S. P. Singh, “Eligibility traces for off-policy policy evaluation,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML ’00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, p. 759–766.

[18] S. Levine and V. Koltun, “Guided policy search,” in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1–9. [Online]. Available: https://proceedings.mlr.press/v28/levine13.html

[19] A. Kumar, A. Zhou, G. Tucker, and S. Levine, “Conservative q-learning for offline reinforcement learning,” 2020.

[20] A. Kumar, J. Fu, G. Tucker, and S. Levine, “Stabilizing off-policy q-learning via bootstrapping error reduction,” 2019.

[21] F. Torabi, G. Warnell, and P. Stone, “Behavioral cloning from observation,” 2018. [Online]. Available: https://arxiv.org/abs/1805.01954

[22] S. Fujimoto, D. Meger, and D. Precup, “Off-policy deep reinforcement learning without exploration,” 2018. [Online]. Available: https://arxiv.org/abs/1812.02900

[23] I. Kostrikov, A. Nair, and S. Levine, “Offline reinforcement learning with implicit q-learning,” 2021. [Online]. Available: https://arxiv.org/abs/2110.06169

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” 2018. [Online]. Available: https://arxiv.org/abs/1801.01290

[25] A. Nair, A. Gupta, M. Dalal, and S. Levine, “Awac: Accelerating online reinforcement learning with offline datasets,” 2020. [Online]. Available: https://arxiv.org/abs/2006.09359

[26] S. Fujimoto and S. S. Gu, “A minimalist approach to offline reinforcement learning,” 2021. [Online]. Available: https://arxiv.org/abs/2106.06860

[27] J. Lyu, X. Ma, X. Li, and Z. Lu, “Mildly conservative q-learning for offline reinforcement learning,” 2022. [Online]. Available: https://arxiv.org/abs/2206.04745

[28] T. Zhang, X. Wang, B. Liang, and B. Yuan, “Catastrophic interference in reinforcement learning: A solution based on context division and knowledge distillation,” 2021. [Online]. Available: https://arxiv.org/abs/2109.00525

[29] R. Schiewer and L. Wiskott, “Modular networks prevent catastrophic interference in model-based multi-task reinforcement learning,” 2021. [Online]. Available: https://arxiv.org/abs/2111.08010

[30] W. Fedus, D. Ghosh, J. D. Martin, M. G. Bellemare, Y. Bengio, and H. Larochelle, “On catastrophic interference in atari 2600 games,” 2020. [Online]. Available: https://arxiv.org/abs/2002.12499