Reviewer #1

1. (Hyperparameters) In the Atari experiment section 5.2, we have clearly stated as follows.

"We train EBU and baselines for 10M frames (additional 20M frames for adaptive EBU) on 49 Atari games with the ***same network structure, hyperparameters, and evaluation methods used in Nature DQN"***

Just as stated, to train all the baselines (Nature DQN, PER, Retrace and OT) and our EBU, we used the **same hyperparameters, network structures and evaluation methods that are already defined in Nature DQN paper**. The common specifications in Appendix D are just detailed descriptions of each hyperparameter used in Nature DQN paper that we applied to all the baselines and our method for the experiment. We **did not select any of the parameters manually** to give a good score at 20M frames. We implemented our algorithm based on the code of Nature DQN **by only modifying the target generation process** and episodic sampling process without modifying a single hyperparameter of the original DQN code. Therefore, we are certain that we have provided a fair comparison.

We apologize for the source of confusion about the update period in Appendix D. We meant *"At each* ***update step (4 agent steps or 16 frames)****, we update transitions in minibatch with size 32."* We will correct this to prevent confusion.

2. (Literature Review) Our major focus of this research was to improve the sample efficiency of **model-free, value-based deep** reinforcement learning by making a change as simple but effective as possible – that is to modify the target generation method only. There have been many recent related works including the ones the Reviewer 1 cited. Unfortunately, we could not provide detailed differences/advances of them all in the limited amount of manuscript. We decided to include the most related works such as PER, Retrace, OT that improved the sample efficiency of DQN by **only modifying the target generation process** and do not modify the network or add an additional memory structure.

Many of the recent reinforcement learning methods require changes in the network structures or require additional memory structures (Ephemeral Value Adjustments, RUDDER). Some works investigate algorithms in tabular environments or require tabularization (Is prioritized sweeping the better episodic control?, Efficient Model-Based Deep reinforcement learning with variational state tabulation). These works are orthogonal to EBU, which is a model-free, value-based, deep reinforcement learning algorithm that only modifies the target generation process from the original DQN.

The idea of the backward update is not novel and we have stated in section 3.1 that the tabular backward update (Algorithm 1) is a special case of Lin's method (1992). And we also stated that our main contribution is that we successfully applied the backward update idea in the **deep reinforcement learning domain**, which often fails due to the state correlation. The idea of the backward update may be similar to prioritized sweeping (1993) but prioritized sweeping requires a queue to seek for all predecessor states for value updates. Therefore, it is often inapplicable in the deep learning domain or it requires a state tabularization. However, we agree that some of the recent works share the backward update idea and we will try to include this literature review in the manuscript or at least in the appendix.

3. (Data efficiency of the adaptive scheme) Following the convention, we defined the number of **agent-environment interaction** as the metric for data-efficiency. The training process of the adaptive scheme is described in Appendix A. All the K networks are trained using the same sample episode at the same time. All the networks share the same replay memory. Only one of the K networks is selected at every episode to output a policy and to fill the shared replay. Therefore the adaptive method may be K times computationally inefficient to train but achieves the same data efficiency as the constant diffusion factor method.

Reviewer #2

1. We agree that the comparison of "39 days" and "a couple of hours" is unfair. We will remove the direct comparison.

2. We apologize for the readability issues, we will certainly modify the figures and text to improve readability.

Reviewer #3

1. To sample an episode, we sample one of the terminal states in the replay memory. After sampling a terminal state, we used the episode that includes the terminal state to generate a temporary Q-table and update the values. Therefore, it was not more complicated than sampling a fixed-length trajectory.

2. The goal of the adaptive method is to improve the constant method without harming the data efficiency. We agree that the word 'adaptive' may not be the best description of the method. We will try to find a better description.

3. Even if the n-step Q-learning has a larger n, it does not correspond to the episodic backward update. N-step Q-learning uses the sum of discounted rewards plus the n-step bootstrapped value at the end. However episodic backward update takes the discounted sum of maximum values from backward, so EBU may propagate higher values faster.