We thank all reviewers for their time and appreciate the thoughtful feedback. Below, we address the main comments.

Reviewer 1: *"In the example given by the author, the agent is allowed to run until it reaches a terminal state during training time, but during test time is cut off after h timesteps."*

We understand why this would be a concern, but it is actually not what we do. First, note that Figure 1 simply plots the optimal policies, which are theoretical entities that are independent of the training procedure (we computed them analytically). That being said, there is no reason why these optimal policies could not be learned with finite-length training episodes, all starting from the same initial state. As an example, consider that we want to learn π_l^* using training episodes of length 12 (the same length as the performance metric). For simplicity, consider that we use Q-learning with a behavior policy that selects actions uniformly at random. Under this behavior policy there is a non-zero probability for each state-action pair that it will be visited within a single training episode. Hence, sufficient exploration occurs to enable convergence in the limit. A key detail to achieve convergence is that the moment the training episode reaches its final time step, this is not treated as a terminal state, but normal bootstrapping is used.

On the topic of terminal states, note that we have not explicitly defined any terminal states for the tasks from Figure 1. It might seem logical to define the state for which all positive objects are collected to be terminal, but this is not strictly necessary (it does not affect the optimal policy, nor the metric gap, nor the ability to learn these). We consider terminal states to be part of the MDP definition (see line 49). In other words, they are independent of the performance metric F or learning metric F_l. A finite performance metric does not introduce terminal states, just like stopping training after a finite number of steps does not introduce a terminal state. We will clarify this point further in the paper.

R1: *"Their approach was marginally better than DQN on most Atari games [...] it would be nice to see some demonstrations where the advantage of the author's method was clear and convincing [...]"*

We hope that our clarification of the Figure 1 plots has increased your appreciation of low discount factors. There is likely room for further improvement of the non-linear results, but we also believe that our current results show a decent improvement. Especially, because our technique is very general and can likely be combined with other techniques for improving performance. Furthermore, we argue—as you mention in your review as well—that the main contribution of this paper is more fundamental than the (obligatory) Atari evaluations.

R1: *"It seems to me that value methods have fallen out of favor relative to policy gradient methods, but it is perfectly clear how to generalize the author's approach. Perhaps it would be worth a mention by the authors?"*

Our approach can indeed be easily combined with policy-gradient methods. We will make a mention of this in the next version of the paper.

Reviewer 2: Thanks you very much for the kind words. We are delighted that you enjoyed reading it!

Reviewer 3: *"[...] the plotted curves somehow look like 'smoothed'. Please let me know if I missed something."*

This is due to an artifact of our plotting routine; in reality there is indeed a sudden change. We will update our plotting routine to remove this apparent smoothing.

R3: *"[...] is logDQN more robust compared to the DQN baseline for different gamma values (in the investigated interval [0.84, 0.99])? It is of great interest to see the performance as a function of different gamma values for logDQN and DQN."*

Generally spoken, applying our logarithmic mapping increases the robustness with respect to γ. This can be observed for the linear case by comparing Figures 3 and 8. However, properly evaluating the γ-dependent behavior for the non-linear case is non-trivial. The main reason for this is that DQN contains a lot of hidden hyper-parameters that work well for $\gamma = 0.99$, but it's unclear if these are also a good choice for different γ-values. As we mention in the Discussion section, we suspect that the reason that the optimal γ we found for logDQN ($\gamma = 0.96$) is very close to the one for DQN ($\gamma = 0.99$) is related to this. We do plan to further investigate the non-linear setting in the future, including an in-dept evaluation of the γ-dependence between logDQN and DQN, but it's outside the scope of the current paper.

R3: *"Since the logDQN also changed the network architecture by adding more output units, the empirical result would be more significant if comparison against Dueling DQN is provided."*

We argue that a comparison of logDQN with Dueling DQN would not be as meaningful as our current comparison against DQN, because Dueling DQN does not just increase the output units, it is a different method altogether that enables generalization across actions. Furthermore, our logarithmic mapping approach is a general approach that can be combined with many existing methods, including Dueling DQN.