We thank all of the reviewers for their valuable feedback. We appreciate that **Reviewer 4** finds the theoretical connection to decision theory both "sound" and "clear and valuable" and that **Reviewer 3** states that "the formulations are correct." However, we acknowledge that the clarity and organization of the theory could be improved. To address this, we will explain the major theoretical results more clearly in the main body of the text, and include a sketch of the proof of Theorem 2 as suggested by **Reviewer 4**. We will incorporating **Reviewer 2**'s suggestions for improving the clarity of the goals and assumptions of the theoretical results. We will also provide additional experiments in the appendix, which test new methods for computing regret (for **Reviewer 1)** and assess whether a curriculum can be generated without training the protagonist using regret (for **Reviewer 4)**.

Reviewer 2 points out that Therorem 2 shows that PAIRED recovers the minimax regret policy when the adversary and the antagonist successfully coordinate, but it is unclear this condition holds. However, coordination is not required for PAIRED to work. If the policies reach a Nash Equilibrium, the protagonist performs at least as well as the antagonist in every parameterization, since the protagonist could guarantee 0 payoff by using the antagonist's strategy and otherwise the adversary would choose a distribution on which the protagonist performs relatively worse. With a capable antagonist the protagonist would learn the minimax regret policy, even without the adversary and the antagonist coordinating. We will add this explanation to the paper.

Reviewer 2 asks for clarification of the conditions of Theorem 1: there are two outcomes, Success and Failure, and the range of rewards given for Success does not overlap with the range of rewards given for Failure. Specifically, Successful outcomes give rewards in some range $[\mathbf{S}_{min}, \mathbf{S}_{max}]$, and Failure outcomes give rewards some in range $[\mathbf{F}_{min}, \mathbf{F}_{max}]$ such that $\mathbf{F}_{min} \leq \mathbf{F}_{max} < \mathbf{S}_{min} \leq \mathbf{S}_{max}$, and $\mathbf{S}_{max} - \mathbf{S}_{min} < \mathbf{S}_{min} - \mathbf{F}_{max}$ and $\mathbf{F}_{max} - \mathbf{F}_{min} < \mathbf{S}_{min} - \mathbf{F}_{max}$. We thank **Reviewer 2** for bringing this ambiguity to our attention and will update the paper accordingly.

Reviewer 2 also offered the valuable advice to improve clarity by offering a "specific example/ problem that would be easily solved by defining it as a UED". One example is training a robot in simulation to pick up objects from a bin in a real-world warehouse. There are many possible configurations of objects, including objects being stacked on top of each other. We may not know *a priori* the typical arrangement of the objects, but can naturally describe them as parameterizations of a simulated environment. We can provide a curriculum of training configurations with either domain randomization, minimax, or PAIRED. We will introduce this example in the introduction and reference it throughout the formal introduction of UED to improve the clarity of the formalism.

Reviewer 2 points out that "clarity for UED could be improved" with "a clearer description saying we have X and want to achieve Y". In our setting we are given a class of training environments, and our goal is to construct a policy which performs well across a large set of these environments. We evaluate with a set of specific environments used as transfer tasks. To train such a policy, we start with an initially random policy, generate environments which are tuned to help it learn, train the policy on those environments, and repeat until convergence or until we have exceeded available computational resources. We call the problem of choosing how to generate these environments UED. The choice of how to solve the UED problem affects both the curriculum it generates and how the policy prioritizes performance in different environments at convergence. We will add phrasing to this effect in the introduction of the paper.

Reviewer 1 points out that although using the 'relative regret' between the protagonist and an imperfect antagonist leads to an effective curriculum, it may be an inaccurate estimate of the worse-case regret during training. They add, "There are potentially many choices for how to differentially train the protagonist/antagonist policies". We are conducting additional experiments empirically investigating alternative methods for estimating regret: 1) we use a population of agents and compute regret as the difference between the highest performing agent and the population average, 2) we relabel the the current best-performing agent as the antagonist. We will include the results of these additional experiments in the appendix.

Reviewer 4 asked whether it is possible to decouple generating a curriculum of environments using regret, and having the agent itself optimize regret. We have experimented with training the protagonist with environmental reward and not regret and find that it still performs well; we will provide these results in the appendix. In addition, the performance of PAIRED in the random 50 blocks environment indicates that it would perform well as a policy for the domain randomization setting, even though it was designed for a different objective.

Reviewer 3 - we will provide further discussion on scenarios where the method is not applicable, including settings in which you already have an accurate model of the test environments you expect to encounter.

We will improve the Broader Impact statement according to the suggestions of **Reviewer 2**, making it more specific and focusing on PAIRED's ability to train more robust agents that "cover corner cases that may happen in the real world". We will correct all typos identified by the reviewers, including the caption of Figure 3 which mistakenly refers to a Lava environment. We appreciate your detailed feedback.