We thank the reviewers for the time and effort put into the reviews. Below we address the major comments raised in the reviews; we apologize for brevity due to the space constraints.

Reviewer 3:

- **Computational complexity considerations, etc:** Our algorithm is indeed computationally efficient. While this is mentioned in passing, we agree that this merits an explicit discussion and will add it in the final version. Thanks!
- **Some definitions are dense:** We have given this matter considerable thought while writing. We found that the definitions given as preliminaries are the minimum requirements for the exposition to be self contained. While it is possible to state the surrogate costs as an existence result and thus forego the cumbersome details, we thought this would be too vague for the reader and thus decided to include them in full. In light of your comment, we will reconsider whether some of the definitions could be deferred or simplified.

Reviewer 4:

- **Whether the use of preconditioned GD necessary:** Our methods can indeed apply to an interior point method such as self-concordant barriers, which would obviate the distinction between the comparator and decision sets. However, we chose preconditioned GD as it yielded the simplest and cleanest algorithm, while also explicitly showing the structural differences inside the decision set.
- **Minor comments on writing:** Thank you for pointing these out! We will make the corrections and check for others.

Reviewer 5:

Thank you for thoroughly reading the paper and appreciating its contributions! It appears that your main concerns are around presentation issues; we address these here.

- **Connection between BLC and BCO:** ξ_t in BCO is in fact w_t in BLC. This is mentioned in line 273, which is perhaps a bit late. We will add a discussion between Sections 4.1 and 4.2 that will explain the relation between BLC and BCO.
- **Overloaded notation:** We used this overloading to name equivalent constructs in the BCO and BLC settings. We agree that this may cause some confusion and will revise the notations in the final version of the paper. In particular, to your comments:
  1. We will rename the BLC comparator set in line 94, such that it does not clash with the BCO comparator set;
  2. f_t: We will rename the surrogate costs in line 140, such that they are distinct from the BCO adversarial functions in line 111.
- **Other responses:**
  3. **"In line 144, ... what is the need to assume that they are bounded in line 104?":** In line 104 we assume that a bound on x, u implies a bound on c_t and its gradient. In line 144 we claim that under the proposed policy x, u are indeed bounded and thus so is c_t.
  4. **"Subsection 2.3 can be shorted":** Please see second point in our response to Reviewer 3.
  5. **"In Algorithm 1, what is $\mathcal{S}$?":** This denotes sampling from the unit sphere. It is explained in point (2) of the algorithm description (lines 168-170).
  6. **"Will the feedback be independent of learner's decision?":** Our intention was that the feedback is statistically independent of the player's decision, which in turn means that it provides an unbiased estimate of the desired underlying adversarial feedback.

Reviewer 6:

- **How generic is the Disturbance Action Policy:** This is not a contribution of our work and is discussed in further detail in previous work (that we deferred details to). In short, the DAP policy can approximate (in terms of cost) any strongly stable linear policy arbitrarily well (see part (iii) of Lemma 6). We will consider adding an appendix with further details to make the paper more self contained.
- **Noise in line 6 of algorithm 1 not used in subsequent updates:** The noise w_t is used to calculate the action u_t (in line 5 of the algorithm). Interestingly, it is otherwise unnecessary for the algorithm, namely, one can separate the controller from the learning algorithm.
- **How iterating the updates in lines 9-12 rolls into the regret:** This is best seen in Lemma 11 and its proof. Essentially, lines 9-12 get called about T/H times, incurring the regret of the base algorithm $R(T/H)$. The randomized schedule prevents the adversary from malicious actions during no update rounds and thus overall the regret is given by $HR(T/H)$ plus an additional term that accounts for the functions' memory. We appreciate that this is a bit hard to discern from the current exposition and will attempt to clarify it in the final version.
- **State novelty right at the beginning:** We used the convention that the preliminaries summarize previous work and the following sections contain only novel contributions unless stated otherwise. Our contributions are summarized in the introduction but we will make an attempt to distill them to make the distinction even more clear.