We *deeply* thank the reviewers for their very thoughtful reviews that will help improve our paper. All of them commended the novelty of our model and the results. We address them separately below. The citation numbers are from the paper.

Lines 102 – 103 (R1, R4): We are indebted to **R1** and **R4** for highlighting this typo. The utility function that matches the (correct) text is equal to: $\delta' \cdot \mathbb{1}\{\text{sgn}(\langle \alpha_t, \mathbf{r}_t(\alpha_t, \sigma_t)\rangle) > 0\} - \|\mathbf{x}_t - \mathbf{r}_t(\alpha_t, \sigma_t)\|_2$. We will fix it in the revision.

R1: (Upper Bound – Thm 4.1.) If $\mathcal{A}$ is a finite set, then GRINDER becomes similar to standard[1] feedback graph (FG) algorithms ([1]), but the FG is built according to δ-BMR agents, and its regret scales as $O(\sqrt{\alpha(G)T\log T})$ ($\alpha(G)$ being the independence number of the FG). We will add a clarifying paragraph after line 271. The case where $\mathcal{A}$ includes both continuous and discrete sets of actions is a great direction for future work; in the standard online learning setting, results only for metric spaces are known, so it is a big undertaking for cases where $\ell(\alpha, \mathbf{r}_t(\alpha))$ is not Lipschitz (lines 72 – 73).

R1: **(Lower Bound)** The σ_t-induced polytope sets are defined in lines 228 – 230, but following **R1**'s comments we will move the definition outside of the proof. At a high level, these polytopes can be constructed in the dual space by a super powerful learner who knows $\{\sigma_t\}_{t\in[T]}$ (hence the name), but *not exactly the agents' utility function* (footnote 10). These polytopes (with $\tilde{p}$ being the smallest among them) are not affected *or observed* by GRINDER. We use them only as tools for the proof of Thm 4.1. and the lower bound. GRINDER's polytopes (lines 200 – 214) differ from σ_t-induced ones in $\texttt{poly}(d)$[2] factors, which translates to a $\log(\texttt{poly}(d))$ gap between the two types due to the logarithmic dependence of the regret on the smallest polytope (Thm 4.1.). So our bounds are matching up to logarithmic factors (hence the "nearly"). **R1** was rightly confused, and we will add a clarifying sentence.

If the learner *knew* that the *agents are truthful*, then we would have a standard bandit problem. However, she does *not* know neither $\mathbf{x}_t$, nor the agents' precise utility function (hence the dependence on the σ_t-induced polytopes and $\tilde{p}$).

R1: **(Related Work)** The strategic nature of our agents (making $\ell(\alpha, \mathbf{r}_t(\alpha))$ not Lipschitz) is what differentiates us from any related standard online learning setting (lines 70 – 73). We will include an extra sentence highlighting this.

R1: **(Additional Feedback)** $\mathcal{P}_0$ should be $\mathcal{P}_0 = \{p(\mathcal{A})\}$ ($p(\mathcal{A})$ being the polytope representation of $\mathcal{A}$). **R1** is also right that GRINDER illustrates an advantage of adaptive discretization over non-adaptive ones[3]. In line 96, this is true for the settings of interest (e.g., spam identification). We will clarify accordingly lines 196 and 96 respectively.

R2: We thank **R2** for the positive comments, the typos, the suggestion for Fig. 1, and $\ell(p, \mathbf{r}_t(p))$ (lines 210 – 213) all of which we will incorporate. We have extra simulations in Fig. 5, 6 of the paper, and we will also include the benchmark that was suggested by **R4** (Fig. 1). For the incompatibility, we remark that it is a worst-case result and that EXP3 (Fig. 4, 5, 6) could also be viewed an algorithm that minimizes *external* regret.

R4: (Model) The schools' admission process can be thought of as another example of our protocol: the school commits to a classifier, students get to "observe" it via proxies (e.g., past admitted students), and then, according to their original point (e.g., features of their resume) and their manipulation power (e.g., money they spend on SAT) they can best-respond. The distribution over classifiers is an excellent direction for future work, as it can "enforce" exploration over the dataset. The modeling of nature choosing $\mathbf{x}_t$'s adversarially is because we wanted to take a worst-case approach, but we conjecture that having the stochasticity that **R4** mentions would lead to better regret guarantees, and it is an exciting avenue for future work. We agree with **R4** regarding $h^{\star}$[4]. We will add these clarifications to our model.

R4: **(Dependence on δ)** δ affects the size of $\underline{p}$ via the definition of the σ_t-induced polytopes (lines 228 – 230), and the analysis of Q_1 (Eq. (2), lines 235 – 243). We will change $\underline{p}$ to $\underline{p}(\delta)$ to make this clearer.

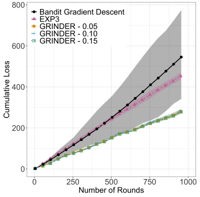

Figure 1

R4: ("Convex" Surrogates) *No standard convex surrogate can be used* since the learner does not know precisely *function* $\mathbf{r}_t(\alpha)$. Hence, the learner cannot guarantee that $\ell(\alpha, \mathbf{r}_t(\alpha))$ is *convex* in α, even if $\ell(\alpha, \mathbf{z})$ ($\mathbf{z}$ being independent of α) is convex in α! Concretely, we will include this counterexample: let $h = (1, 1, -1), h' = (0.5, -1, 0.25)$ two hyperplanes, a point $\mathbf{x} = (0.55, 0.4), y = +1$, $\delta = 0.1$, and let $\ell(h, \mathbf{r}(h)) = \max\{0, 1 - y \cdot \langle h, \mathbf{r}(h)\rangle\}$ (i.e., hinge). We will show that when $(\mathbf{x}, y)$ is a δ-BMR agent, $\ell(\alpha, \mathbf{r}(\alpha))$ is no longer convex in α. Take $b = 0.5$ and construct $h_b = 0.5h + 0.5h' = (0.75, 0, -0.375) = (1, 0, -0.5)$. $(\mathbf{x}, y)$ only misreports to (say) $(0.61, 0.4)$ when presented with h (as h_b and h' classify $\mathbf{x}$ as $+1$). Computing the loss: $\ell(h_b, \mathbf{r}(h_b)) = 0.95, \ell(h, \mathbf{r}(h)) = 0.99$ and $\ell(h', \mathbf{r}(h')) = 0.875$, so, $\ell(h_b, \mathbf{r}(h_b)) > b\ell(h, \mathbf{r}(h)) + (1 - b)\ell(h', \mathbf{r}(h'))$. Since in general $\ell(\alpha, \mathbf{r}(\alpha))$ is not convex, it is unfair to compare Bandit Gradient Descent (BGD) with GRINDER (so we did not include these experiments originally). Since **R4** asked for them, we will include Fig. 1, where GRINDER greatly outperforms BGD. Identifying surrogate losses that are convex against δ-BMR agents is an exciting future direction!

[1] This is *precisely* how we implemented GRINDER for our simulations with the discrete set of actions (left graphs in Fig. 4, 5, 6)!

[2] σ_t polytopes are defined by hyperplanes with a margin of 2δ, but GRINDER's polytopes are defined with a margin of $4\sqrt{d}\delta$.

[3] But we cannot use the "standard" adaptive discretization techniques ([21,9]) since $\ell(\alpha, \mathbf{r}_t(\alpha))$ is not Lipschitz (lines 72 – 73).

[4] We introduced $h^{\star}$ in order to explain that the mapping $\mathcal{X} \to \{-1, +1\}$ does not have to be linear, but it can be arbitrary.