We are grateful to the reviewers for their time and comments. In the rest of this rebuttal, we address their questions in order, tagging the reviewers concerned in each as **#RX**.

#R1: Examples of learning with inexact models. The reviewer is stating as a foremost concern the contention that we are not discussing settings and examples captured by our framework (other than the bandit case). This is factually incorrect: **we are providing a detailed description of such a setting in Lines 143-153,** at the end of Section 2. In this example (online clique prediction), the optimizer gets an imperfect observation $\hat{A}_t$ of the adjacency matrix A_t of the underlying, time-varying graph $\mathcal{G}_t$ (a social network, a feature map, etc.), and reconstructs the actual payoff model via the Motzkin-Straus theorem. As we explain in the text, the quality of the constructed model depends linearly on the accuracy of $\hat{A}_t$ as an estimator. In applications to social networks (see for example the highly cited paper [18] by Fortunato), $\hat{A}_t$ is typically obtained by mapping out a random portion of the graph starting with its most central nodes; as a result, $\hat{A}_t$ – and hence $\hat{u}_t$ – has *exponentially* small bias. In this way, the online clique prediction problem satisfies all the requirements of the "inexact model" framework and requires its full capacity.

We will be happy to use the extra page to expand our description of the online clique example and/or add others (e.g., online path planning with variable traffic demands; see also the point on parametric models below). Since this was the reviewer's foremost concern, we hope that the above clarifies the merits of our work.

#R1: Unbiased feedback in Krichene et al. The reviewer is incorrect in claiming that Krichene et al. already deal with unbiased stochastic feedback: please see Algorithms 1 and 2 of Krichene et al. where it is clearly specified that the optimizer receives **perfect, deterministic information** on the loss functions, *not* stochastic observations thereof.

#R1#R3: Open question of Krichene et al. The last phrase of the published ICML paper of Krichene et al. reads:

"*One question is whether one can generalize the Hedge algorithm to such a bandit setting, so that sublinear regret can be achieved without the need to explicitly maintain a cover*".

The Hedge variant of Algorithm 2 provides exactly such a generalization, i.e., **it achieves sublinear regret with bandit feedback and without needing to explicitly maintain a covering grid.** We hope that this resolves any doubts.

#R1#R3: Infinite-dimensional input. In many practical applications, constructing an inexact model *does not* require infinite-dimensional feedback. A prime example is parametric models of the form $u_t \equiv u(\cdot;\theta_t)$ where θ_t is a partially observable, time-varying vector of parameters. This makes our framework applicable to a very wide range of relevant problems **that do not require infinite-dimensional input.** This is a key feature of the "inexact models" setting.

#R1: Benefits over Besbes et al. We will rephrase accordingly; at the same time, please note that the dynamic regret framework focuses by necessity on *slowly varying* loss functions, so forgetting the past could be a severe drawback.

#R1: Missing references. The cited references mainly concern *stochastic* bandits, not *adversarial* problems. This is a drastically different setting with incompatible results and guarantees; we will cite these papers to explain this in detail.

#R1: On covering nets. An extended discussion on covering nets had already appeared in Krichene et al., so we did not deem it relevant to repeat one here. However, we will gladly use the extra page to discuss this.

#R2: On the problem of Eq. (7). Indeed, a badly chosen ψ could make (7) intractable. However, **ψ is chosen by the optimizer,** a fact which mitigates this problem. In particular, we provide a series of examples with closed form expressions – like Hedge (our centerpiece algorithm) and the decomposable regularizers that we discuss in Appendix A.

#R2: On the structure of $\mathcal{K}$. We clarify in the beginning of the paper that $\mathcal{K}$ is assumed convex (see Lines 13-14).

#R2: On knowing ν in Theorem 2. Please note that **this is a very difficult unsolved problem**, even in the context of online *linear* optimization with unbiased stochastic input. [Certain papers [15,24] have proposed online convex optimization methods that adapt to ν with *perfect, deterministic input*, but the imperfect information case remains open.]

#R3: Approximating p_t. Yes, representing p_t as a finite-dim. object is exactly why we focus on simple strategies.

#R3: On Agarwal et al. Please note that Agarwal et al. also requires full knowledge of the loss functions (see (1) in Alg. 1), so its information requirements are the same as Krichene et al. – and hence heavier than an inexact model.

#R3: On σ_t and M_t. Good point, thanks! Indeed, since $\mathcal{K}$ is compact, there is little point in the distinction, will do.

#R3: On the role of $\mathcal{C}$. Good point, we understand how this can be confusing. As the reviewer suggests, we will explain in the main text that $\mathcal{C}$ is only needed for the analysis, not the derived bounds.

#R3: On the tightness of $\mathcal{O}(\sqrt{T})$. This was shown rigorously by Abernethy et al. [1], we will give an exact pointer.

#R1#R4: On experiments. On R4's comment, we would like to point out that we already provide experiments in App. E; on R1's comment, we can provide a github link with all code and notebooks for running the various experiments.

#R4: On convex vs. concave. We understand R4's presentation concern, we will do a sweep to minimize switches.

#R4: Tabulation of previous bounds. Excellent suggestion, will do.