We thank all reviewers for their thoughtful feedback that can help enhance the presentation of our results. Below we respond to the major points raised by the reviewers (for each point, we refer to the particular reviewers that raised it).

ConPlanner via unconstrained RL planners under Lagrangian heuristic in experiments (Reviewer 1). The baselines (RCPO and ApproPO) are based on unconstrained RL solvers under Lagrangian heuristic. To enable a more fair comparison to the baselines, we implemented a ConPlanner that is as close to this as possible (implementing it with an unconstrained planner under Lagrangian heuristic). This provides an approximately optimal planner (rather than an optimal constrained planner such as LP) but our theorems are anyway approximation-preserving (this is direct in our proofs but we can add a note about that). As a result, by having an implementation that is close to the one of the baselines, we could better demonstrate the statistical improvement of our methodology.

Lack of concave-convex and knapsack experiments (Reviewer 1). Benchmarks do not currently exist for convex/knapsack constraints. For the sake of transparency, we decided to compare against prior works handling concave-convex or knapsack settings on established benchmarks (for the linear case) rather than come up with our own (although it renders the empirical comparison a bit more limited). We will clarify this decision (as the reviewer recommends).

Comparison to concurrent work (Reviewer 1). The results of the concurrent works are similar to our warm-up basic setting (Section 3). Although we believe that their techniques can extend to more general settings, we do not see a way to do that without using the new technical tools we create in Sections 4 and 5 (e.g., see next point re mean-value).

Intuition behind mean-value theorem (Reviewer 2). The mean-value theorem is applied in two places: a) proving that the optimal policy is feasible for our constrained planner (lines 520-525 of the appendix) and b) to enable the regret decomposition (lines 535-538 of the appendix). The latter application draws a distinction to the multi-armed bandit setting as it is required to address the difference between the model used and the true model — Eq. (20) holds trivially when $H = 1$. As suggested by the reviewer, we will add a discussion about that in the main body as it provides intuition of where mean-value applies and also draws an elegant technical distinction to the bandit setting.

PAC bound (Reviewer 2). A regret bound can be transferred to a PAC bound: a $\sqrt{K}$ regret bound can be turned into a $1/\epsilon^2$ PAC bound by taking the resulting mixture policy. We will add a note in the final version.

Knapsack solver (Reviewer 2). The knapsack solver is provided in Appendix A.3 and is a linear program with occupation measures as variables (thus it is optimally solvable).

Beyond tabular settings (Reviewers 2 and 4). Our results can serve as starting point towards extending to non-tabular settings such as Linear or Lipschitz-continuous MDPs where optimistic algorithms for the unconstrained counterparts exist. We will discuss the additional challenges that arise in these settings and explicitly state them as future directions.

Dependence on d and high probability (Reviewer 3). We assumed function g (line 181) is L-Lipschitz continuous under L_1-norm ($\|1_d\|_1 = d$ here). More generally, our analysis extends when g is L-Lipschitz with respect to L_p norm ($p \geq 0$) and obtains $L \cdot \|1_d\|_p$ dependence. Our results also extend to high-probability by an application of Lemma F.3.

Clarifications on the plots (Reviewer 3). The number 1.65 says that, after running the setting for 500 episodes (or trajectories) each with $H = 30$ steps, the expected reward of the resulting policy π_k at episode k is equal to 1.65. Note that this is not the time-average reward; the plot for cumulative regret looks similar (note that the x-axis is in log-scale, we will add it as the reviewer suggests). The second row corresponds to expected cost of policy π_k, i.e., $\mathbb{E}^{\pi_k}[\sum_h c_h^\star]$. The third row corresponds to the cumulative empirical cost, i.e., $\sum_{t=1}^{k} \sum_{h=1}^{H} c_{t,h}$ (the term *during training* meant to show what happens if we terminate at episode k but we will replace this term by the exact formula to make it clear).

Regarding access to the latent model (Reviewer 3) Algorithm 3 is called with the empirical model transitions $\hat{p}_k$ as a parameter – it does not require access to the latent model; it only uses the model estimates (we will clarify this).

Comparison to TFW-UCRL2 (Reviewer 3). In the experiments, our goal was to find a (constraint-feasible) policy that guarantees a desired reward in as few episodes (trajectories) as possible. In the plots, we display the number of trajectories needed in order for this goal to be achieved (TFW-UCRL2 requires more trajectories to achieve that). We will add a column about the alternative metric of regret – we ran it and it looks qualitatively similar (that said, it is probably useful to show pictorially the vanishing regret property so such an addition would be useful for the paper).

Regarding how we ran TFW-UCRL2, we use the code provided by the author of TFW-UCRL2 (with no algorithmic parameter changed). TFW-UCRL2 uses weights $(L_0, \ldots, L_K)$ in the objective function $g(w)$ (Eq. 1 page 2 of that paper in its NeurIPS version). We only tuned these weights to identify the ones maximizing the reward while guaranteeing constraint satisfaction (for a more fair comparison to the baseline). We will include the additional plots for different values of $(L_0, \ldots, L_K)$ in appendix.