We thank all the reviewers for their time spent on our submissions and for their valuable comments. We would like to make the following clarifications here. Minor points will be addressed in the revised manuscript if accepted.

We thank Reviewer 1 for their positive feedback. As is standard in prior work on LUCB [1,2], determining β_t requires S, L, λ, and R. Our algorithm additionally requires λ_- defined in (4) for correct tuning of T'. It is possible to explicitly determine λ_- as follows. Find the largest $0 < \epsilon \leq C/S$ such that $\{x \in \mathbb{R}^d \mid \|Bx\|_2 = \epsilon\} \subset \mathcal{D}^w$. Then, by generating iid samples $x_t = \epsilon B^{-1} z_t$, where z_t is uniform on the unit sphere, it can be shown that $\lambda_- = \frac{\epsilon^2}{d\|B\|^2}$. We have chosen to defer the discussion on computational issues to the appendix due to space constraints and because similar ideas have been mostly developed in previous work [1]. However, as per the reviewer's recommendation, we will explicitly state after Eqn. (8) that the involved optimization is non-convex in general and a computationally tractable modification is presented in Appendix D. We also thank the reviewer for their suggestion to provide error bars for our experimental results in Fig. 1. We have now done this and will include in the paper (not shown here due to space constraints).

As per Reviewers' 2 and 3 suggestion, we acknowledge that a more elaborate comparison with prior works [14,22,25,27] will benefit the reader; we will do so in Sec. 1.2 of the manuscript. As a general comment: despite certain similarities to these works, we are confident that our submission differs substantially in its core contributions as explained next. In [14], the authors study a variant of LUCB in which the actions $x_1, \ldots, x_t$ are constrained such that the *cumulative* reward remains *strictly* greater than $(1-\alpha)$ times a given baseline reward for all t. In contrast, the safety requirements in our paper requires $\mu^T Bx_t \leq C$ which is same for every action x_t, *independently of actions chosen at other time instants*. The two constraints are different, thus the algorithm and analysis of [14] are *not* applicable in our setting. Interestingly though, the assumption $\alpha r_\ell > 0$ in [14] is somewhat reminiscent of the case $\Delta > 0$ studied in our paper. Similarities to the recent work [25] include the defined safety constraint and using confidence region to ensure that actions are safe (also similar to [22,27]). However, the two works differ drastically as we aim to provide *regret guarantees* for a linear but otherwise unknown objective, whereas [25] allows for more general convex objective and aims at *convergence guarantees* rather than regret bounds. We thank Reviewers 2 and 3 for bringing [27] to our attention. To the best of our knowledge, [22,27] are important "safe" counterparts of [28], which introduces a UCB-type algorithm and proves *regret guarantees* extending standard *Linear*-UCB works [1,2,3] to *nonlinear* bandits modeled by Gaussian processes (GPs). Regret guarantees imply convergence guarantees from an optimization perspective (see [28]), *but not the other way around*. The algorithms in [22,27] come with convergence guarantees, but *no* regret bounds as done in our paper. This is the first important difference to our work that proves regret bounds providing a "safe" counterpart of [1,2,3]. Even beyond theoretical guarantees, the experiments in [22] show a notion of regret ($r_t = f_0^* - \max_{i \in [t]} f(x_i)$) that deviates from the more popular notion used in our work ($r_t = f^* - f(x_t)$). Of course, our analysis relies on the fact that the cost function comes from a *finite* dimensional linear space. Extensions to infinite-dimensional linear spaces (hence to GPs) is beyond the paper's scope, but it is very interesting to attempt combining our ideas with those in [27] to prove *regret bounds* for the nonlinear bandit with GPs. In this direction, it is worth emphasizing (we will do so in the revised manuscript) that the algorithm in [27] also consists of two phases: one that expands the safe region and a second that aims at utility optimization. We hope that our contribution motivates further investigations in this critical direction. Some other differences of our work to [22,27] are as follows. The finite-dimensional setting allows us to compare performance against the optimal cost *within the actual true safe set*, rather than an *estimated* subset of it (Eqn. (1) in [22]) as done in [22,27]. Also, Algorithm 1 and Thm. 2 & 3, do apply beyond the K-arm setting to compact convex decision sets that include *infinite* number of actions. For supporting experimental results please see Figs 1.b and 2.

Now, we respond to other questions posed by Reviewer 2. Regarding solving Eqns. (7) & (8), please see App. D and lines 6-10 here. For GSLUCB, we remark that *by design* the duration of its first phase never exceeds the worst case T', i.e. T_0. Thus, even if the safety gap is overestimated, the second phase begins after *at most* T_0 rounds and Thm. 3 naturally applies. Also, please refer to App. E for details on how we calculate the lower confidence bound Δ_t. Regarding reducing the duration of the pure exploration phase, it is actually possible to achieve a *constant* T' (rather than logarithmic as in Lem. 4) by simply taking intersection of the previous sets with the confidence set at round t such that $\ldots \subseteq \mathcal{D}^s_{t-1} \subseteq \mathcal{D}^s_t \subseteq \mathcal{D}^s_{t+1} \subseteq \ldots$. Thus T' is the smallest value satisfying $2\sqrt{2}\|B\|L\beta_{T'} \leq \Delta\sqrt{2\lambda + \lambda_- T'}$. Note however that this does *not* change the order of regret in Thm. 2. Finally, the reviewer makes an interesting point about having the constraint depend on another unknown vector other than μ. We have also thought of this modification and we agree that is worth discussing in the appendix. Having the constraint depend on another unknown parameter does not affect the analysis. We have chosen to focus on the current setting in the main paper since: (a) we believe it makes the presentation clearer without loosing anything substantial; (b) our initial motivation comes from specific power applications where the safety constraints and the cost functions both depend on the same parameter. Besides, none of the two settings is a special case of the other: choosing $\lambda = B^\dagger \mu$ is close but not identical to our current setting since we do *not* observe (noisy versions of) $\lambda^\dagger x_t = \mu^\dagger Bx_t$.

[27] Sui, Zhuang, Burdick, Yue: "Stagewise-safe Bayesian Optimization with Gaussian Processes"; **[28]** Srinivas, Krause, Kakade, Seeger: "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design".