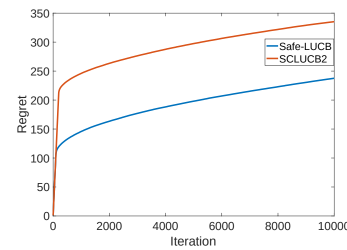
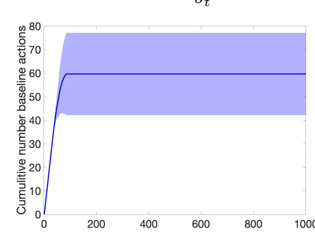


We thank the reviewers for their constructive feedback and their valuable time.

Reviewer 1: First, we respond to the reviewer's **questions/suggestions regarding the experimental results**. Regarding comparisons to the safe-LUCB of [16], we present SCLUCB2 in App. H as a modified version of our main SCLUCB algorithm tailored for the exact safe bandit setting studied in [16]. Importantly, SCLUCB2 comes with a theoretical regret bound, which matches the proposed problem-dependent upper bound in [16]. We now confirm this numerically in the displayed figure, which plots the cumulative regret of the two algorithms averaged over 100 realizations. We will include this new numerical study in the final version. The reason that we only plot regret curves and not the number of times the safety constraint is violated, is because this number is *zero* for almost all realizations. This is expected since all our algorithms guarantee the model's requirement that the safety constraints are not violated for any time step, with high-probability $1-\delta$. Second, **regarding the parameters** (R, S, L) **of Ass. 1-3**, assuming knowledge of them is standard in the literature of linear bandits (see [5,11,10,13-15]). Their specific values are, of course, highly application-dependent, but the underlying hypothesis is that they can be accurately determined based on domain-knowledge/physics, or, estimated from historic data. Even if accurate approximations are not possible, rather loose bounds suffice to run the algorithms. Of course, the quality of these bounds affects the performance, but, the accompanying regret-bounds quantify the effect. **Regarding the parameters** $(r_l, r_h, \kappa_l, \kappa_h)$ **in Ass. 4** that are associated with the baseline policy, it can be reasonably assumed that they can be estimated accurately from data. This is because we think of the baseline policy as "past strategy", implemented before bandit-optimization, thus producing large amount of data (see also [1-3]). If no knowledge is available however, κ_h and r_h can always be set to equal 1 (since for simplicity we assume that the mean rewards are in $[0,1]$). Similarly, κ_l can be set equal to zero. On a related note, we address the **question on tuning the hyper-parameters** $\delta, \lambda, \rho, \alpha$**.** The tuning of δ, λ is standard and is same as in all linear-bandit algorithms [5,11,10,13-15]: $1-\delta \in (0,1)$ is the desired confidence (e.g., 0.95) on the algorithm's realizations to satisfy the regret bounds (here, also the safety constraints); the regularization parameter λ can be set equal to one. The parameter ρ, controlling the exploration level of conservative actions can take any value in the interval specified in Lemma 2.2. The parameter $\alpha \in (0,1)$, controlling the conservatism level of the learning process, is assumed known to the learner similar to [1,2,3]. We will clarify the above in the revision. Finally, **the assumption** $\langle x, \theta_\star \rangle \leq 1$ is not essential and is rather only meant for simplicity. Specifically, the assumptions $\|x\|_2 \leq L$ and $\|\theta_\star\|_2 \leq S$ suffice, as they guarantee the constant bound LS for $\langle x, \theta_\star \rangle$; thus, nothing fundamental changes in our analysis. For example, without this assumption, ρ_3 in Eq. (18) of Theorem 4.1. simply changes to $\rho_3 = \frac{\alpha r_l}{S+LS}$. Contrary to our intention, this assumption appears to be confusing and will be removed in the final version. Minor: The parameter ρ in Algo. 1 appears in the definition of x_t^{cb} in Eq. (11). We will clarify this.

Reviewer 2: First, please refer to lines 18-22 above on how the parameters ($r_l, r_h, \kappa_l, \kappa_h$) are chosen. We further clarify the following. Regarding κ_l: Indeed, there is a typo in line 228 and the related factor in the sample complexity should rather be $\kappa_l + \alpha r_l$ as specified explicitly in Thm. 3.3. What this bound suggests is that while setting $\kappa_l = 0$ is possible, a higher value is preferable (provided that it lower bounds κ_{b_t} in (4)), since it results in smaller regret. Regarding the requirement $r_l > 0$: Indeed, this is necessary for the algorithms to perform well and is critically used in the proofs (e.g. Eq. (23)). That said, this is expected to be met in practice since the baseline policy is the system's current strategy and should have been associated with at least a positive reward. Second, let us clarify the **assumption that the action set contains the unit ball**, eqv. $L \geq 1$. This goes hand-in-hand with our assumption $\|\zeta\|_2 = 1$ (line 199), since together they guarantee that the convex combination x_t^{cb} in (11) is feasible, i.e., satisfies $\|x_t^{\text{cb}}\|_2 \leq L$. However, this requirement remains true as long as $\|\zeta\|_2 = \epsilon$ and $L \geq \epsilon$ for any $\epsilon > 0$. To see this, note that $\|x_t^{\text{cb}}\|_2 \leq (1-\rho)L + \rho\epsilon \leq L - \rho(L-\epsilon) \leq L$. In particular, $\epsilon > 0$ can be chosen small enough for "thin sets". Changing $\|\zeta\|_2 = \epsilon$, we simply adjust $h_1 = 2\rho_1(1-\rho_1)L\epsilon + 2\rho_1^2\epsilon^2$ and $\rho_1 = \frac{\alpha r_l}{S\epsilon + r_h}$ in Thm. 3.3. Minor: Thank you for the comment about LUCB. Also, we agree and will modify line 59 to clarify that the learner Knows x_{b_t}.



Reviewer 3: Thank you for the suggestion on **numerically verifying the number of times that baseline is played.** The figure on the right plots the cumulative number of baseline actions played by SCLTS until time t, for $t = 1, \ldots, 1000$. The solid line depicts average over 100 realizations and the shaded regions show standard deviation. The figure confirms the logarithmic trend predicted by theory. We will upload our code as suggested. We finish with **a brief proof-sketch of Thm. 3.3**, which we will include in the paper. The first idea is based on the intuition that if a baseline action is played at round t, then the algorithm does not yet have a good estimate of the unknown parameter $\theta_\star$ and the safe actions played thus far have not yet expanded properly in all directions. Formally, this translates to small $\lambda_{\min}(V_t)$ and the *upper* bound $O(\log \tau) \geq \lambda_{\min}(V_\tau)$ (Eq. (43)). The second key idea is to exploit the randomized nature of the conservative actions (cf. (11)) to *lower* bound $\lambda_{\min}(V_\tau)$ by the number $|N_\tau^c|$ of times that SCLTS plays the baseline actions up to that round (cf. Lemma D.1). Putting these together leads to the advertised upper bound $O(\log T)$ on the total number $|N_T^c|$ of times the algorithm plays the baseline actions.