We thank reviewers for useful comments. **All reviewers:** We look at related questions: **(i) How is LCPP different from proximal point and what role is played by g_k, (ii) Other experiments beyond MCP and on inverse problem**. **(i)** LCPP is different from proximal as it uses proximal point in the objective and convexification in constraint. Since, we convexify the constraint in g_k, we can show that subproblem can be solved efficiently and approximate final solution will be feasible. Feasibility allows to characterize a simple MFCQ condition under which $\bar{y}^k$ is bounded. As demonstrated in Fig 1 below, (a)-(c) shows the original nonconvex set and convex subsets dynamically generated in the run of algorithm. It is unclear whether these subsets will be algorithmically well behaved, i.e., the optimal Lagrange multiplier for such constraints will be small. Indeed, as shown in Fig 1(d)-(e), as prox center gets close to the point which violates MFCQ, the convex polyhedron flattens and bound on Lagrange multiplier blows up. Our analysis precisely characterizes such bad conditions and can also provide a priori bounds, giving us full control over complications that may arise in dynamically generated convex subproblems (see Appendix B). This need of bounded $\bar{y}^k$ is crucial for convergence to KKT of constrained optimization and is not a concern for unconstrained or simple set constrained problems in which usual proximal point operates. **(ii)** Due to the space limit in the submitted version, we only report preliminary results on MCP with logistic regression objective. We will compare with other nonconvex penalties and inverse problem objective in a later version.

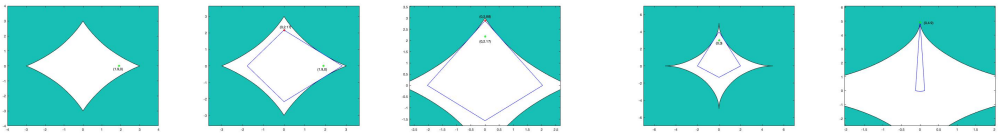

Figure 1: (a)-(e) SCAD constrained optimization. Prox center denoted by green.

Reviewer 1: Weakneess The SCAD paper [12] showed the advantage of nonconvex surrogate penalty, and pointed out the potential problem of biased estimation of large coefficients in Lasso since ℓ_1 penalty does not flatten. Empirical merit of using nonconvex constraint to relax ℓ_0 is also demonstrated by examples in [25]. By suboptimality, we meant to say that using nonconvex formulation is advantageous over ℓ_1 due to the above discussions in the literature and used it to merely motivate the main topic of this paper. We can always improve the language as pointed out by the reviewer. Figure 3 does show that LCPP outperforms Lasso across a wide range of nonzero patterns. See below in other comment. We agree that disclosing form of the constraint beforehand will make presentation better, succinct and allow to have fairer comparison with [5,20] in Section 1. At first glance, it may seem that constraint form is too restrictive, however, it covers variety of nonconvex sparsity inducing functions considered in the literature, e.g., [16,29]. They assume $g(x) = g_1(x) - h(x)$ and we identify more structure that can be exploited to get efficient method for different problems. **Related work** Thanks for pointing out 'CLASH' paper which we will add in the related work. While there exists more advanced models for specific type of problems such as CLASH, they often require strong assumption on data (i.e. RIP) and specific function type. In comparison, LCPP aims to deal with general loss and a variety of nonconvex penalties. The proof technique and convergence results are substantially different from CLASH.
Other Comment: Fig 3: We observe that it is better to impose sparsity constraint. In gisette rcv1.binary and real-sim, using too many features results in overfitting. In mnist, it is possible to obtain nearly the best performance with only using nearly half the features. **Why (5) is hard:** Hardness is not merely in (5) but the question that we want a feasible solution to (5) with faster convergence and with less restrictive constraint qualification that works for nonconvex constraints of interest. Variety of algorithms in classic nonlinear programming [2] showed asymptotic convergence with or without feasibility and [5,20] addressed the complexity part partially but the constraint qualifications requirements were strong to ensure feasibility. We believe addressing these questions is hard for general problems but rates can be further improved for particular class of problems. **Why choose (5) over (4):** While (5) controls the target penalty level directly, (4) controls the level implicitly by tuning weight λ. Unlike the convex case, first-order optimal point of (4) does not guarantee a small value of penalty for large λ. **Matrix rank:** Problem of rank constraint on matrices instead of ℓ_0-norm of vector is a natural extension. See [36] for more information. Very interesting comment.
Reviewer 2: W1. We compare with Logistic regression which uses liblinear (dual coordinate descent) through sklearn interface. We also compare with GIST, an efficient nonconvex proximal gradient method for DC penalized problem. These are quite standard solvers for convex and nonconvex sparse models. W2. In our problem, η is a fixed target constraint level and we don't change its value during the algorithm. By adding a constant to η, we are dealing with a new problem with a more relaxed constraint. If same constant is added on both side of (5) then algorithm does not change at all. W3. To the best of our knowledge, operator splitting and nonconvex ADMM methods are provably convergent only when the constraint function is linear. They are not applicable for (5).
Reviewer 3 Thank you for pointing out Catalyst. As per our knowledge, Frank-Wolfe (FW) method has complexity guarantee if the constraint set is convex which isn't the case in (5). It seems possible to apply FW to solve the convex subproblem in LCPP. We believe that adaptive strategy of Catalyst can be applied in our paper, since the subproblem is convex and admits efficient proximal map. Table 1 describes the complexity to obtain an ϵ-KKT points for different types of objective function $\psi(x)$. The constraint $g(x)$ is nonconvex and nonsmooth according to Assumption 2.1.
Reviewer 4: We will change the language in the abstract. Other concerns addressed in common response.