We thank all the reviewers for their constructive comments and useful suggestions. Unfortunately there is significant misunderstanding of our contributions. We will try to clarify here and also expand this in our paper.

Q (R1, 4): Highlight our contributions:
A: We proposed the first primal-dual algorithm for **constrained** problems. We are significantly more efficient compared to the previous state of the art, both theoretically and empirically. Our method applies to a wide class of ℓ_1 norm and trace norm constrained problems including: ElasticNet, regularized SVMs and phase retrieval, among others. This is a wide class of problems and a large body of prior optimization methods have been published. We have the most efficient provable optimization method and this is a significant contribution.

Q (R1, 4): How is the sparsity constraint chosen in practice? What if sparsity is underestimated?
A: It's always safe to choose a relatively large target sparsity s. If one initially chooses a small s, one can increase it if iterates converge but have smaller ℓ_1 norm than the constrained value. If the sparsity is still underestimated, the sparsity constrain dominated the ℓ_1 constrain, and we will end up obtaining a solution with higher sparsity.

Q (R1): Provide test accuracy to highlight effect of regularization
A: We will add test accuracy as well as a comparison with different levels of regularization in the revised version. However, our focus is on strongly convex objectives that guarantee a unique minimizer (same test error for different algorithms), train accuracy has fully interpretation for the performance of the proposed algorithm compared among others. Relevant literature commonly only reports train error in e.g. [1] or the DGPDC or BFW papers.

Q (R1): How to obtain $\tilde{x}$ in equation 8?
A: Equation 8 is a quadratic function about $\mathbf{x}$. Let's say $\tilde{\mathbf{x}} = \text{argmin}_{\|\mathbf{x}\|_0 \leq s, \|x\|_1 \leq \lambda} \|\mathbf{x} - \mathbf{c}\|_2^2$. Then we obtain $\tilde{\mathbf{x}}$ by performing an ℓ_0 projection followed by an ℓ_1 projection for $\mathbf{c}$.

Q (R2): Why Accelerated Projected Gradient Descend (AccPGD) outperforms SVRG? Multi-threading?
A: We implemented our algorithms as well as the baselines in C++ with the Eigen library, **without** multi-threading/multi-processing. As for SVRG, please note that we are solving the constrained problem, and SVRG has to perform a projection in every inner iteration. In the inner iteration of SVRG, the gradient computation is about $\mathcal{O}(sm)$ scaled by $\text{nnz}(A)/nd$ since our data is sparse, where s is the sparsity during the inner step, and m is the mini-batch size. While projection on the ℓ_1 ball requires $\mathcal{O}(d)$ (see Duchi et al. 2008). That is, projection on the ℓ_1 ball can take more time compared to the gradient computation. Empirically, projection takes about 75% of the CPU time of SVRG, and about 40% for AccPGD. As a side proof, the numerical results of [1] show that the performance of SVRG is not competitive.

Q (R2, 4): Compared with Doubly Greedy Primal Dual Coordinate (DGPDC) and Block Frank Wolfe(BFW)?
A: We are motivated from the primal dual reformulation like DGPDC and recent progress on FW like BFW, but our new algorithm is not a trivial combination of previous results because: **1)** This is the first work to analyze constrained problems using a primal-dual formulation. The challenges come from the non-symmetric formulation on primal and dual variables. Prior work bounds the iteration progress and this will not work for our analysis. **2)** Besides, compared to our results, the analysis of DGPDC highly relies on the **sparsity of the whole iterate trajectory**, which actually has no obvious guarantee to be small. While our analysis only depends on **primal optimal's sparsity**, and is guaranteed by the ℓ_1 constraints. **3)** The inexact update in our algorithm 2 boosts the empirical performance, but also introduces error in every iteration that perturbs the primal progress, and hence imposes more difficulties on the analysis under the primal-dual framework.

Empirically we could not compared with DGPDC since it is not capable of solving constrained problems, but theoretically our sparsity requirement is more natural (on the primal optimal) rather than on the entire iterate trajectory. As for BFW, both empirically and theoretically we have clearly demonstrated our improvements, when computing the full gradient is expensive.

Q (R2): The dual variable is assumed to be sparse?
A: This is a very important point. We do not assume that the dual variables are sparse. In fact they will not be. Our benefit is replacing the dimension d to **primal** sparsity s. We will make sure this is more clear in the paper.

Q (R4): Why primal dual formulation?
A: The primal-dual reformulation ensures its gradient computation to be dominated by a bilinear term. Therefore, when we compute the update with some (low-rank/sparse) structure, we are able to maintain the gradient and keep a cheap update that is independent to the ambient dimension. For the primal framework, this only happens when the gradient is linear in the update variable. This is clearly demonstrated in the theoretical vignette section (line 99 - 106).

We have also mentioned in the introduction (line 30 - 39): the primal-dual formulation allows us to exploit the sparsity nature of the solution, and to reduce computational complexity from the ambient dimension to the solution's sparsity.

Q (R4): Time complexity concerns and Elastic Net
A: About quick select of complexity $\mathcal{O}(d)$, we choose the k-th largest value with linear operations and loop over the coordinates to pick up all values greater than this value. Indeed the update operation is also $\mathcal{O}(d)$ or $\mathcal{O}(n)$ but it doesn't affect the overall complexity. Line 104 comes from the warm-up problem where $f = 1/2\mathbf{x}^\top A\mathbf{x}$ and therefore PGD costs $\mathcal{O}(d^2)$. As for elastic net we are refering to the constrained version on the ℓ_1 regularization.

[1] Hazan, Elad, and Haipeng Luo."Variance-reduced and projection-free stochastic optimization." ICML 2016.