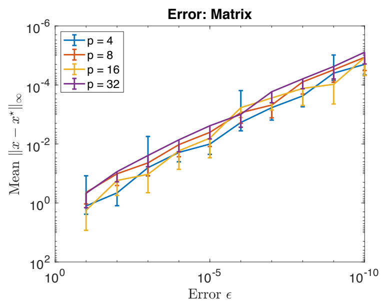
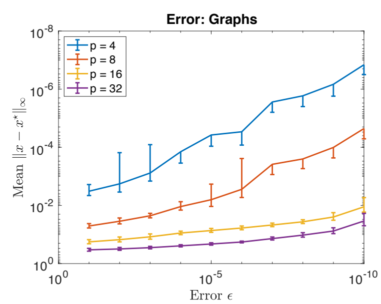
We thank all the reviewers for their helpful comments and suggestions. Below we address the concerns raised.

Importance of Convergence vs Function value (R1). For an algorithm with a $\log \frac{1}{\varepsilon}$ dependence of the running time for computing a $(1+\varepsilon)$-approximate solution, like p-IRLS, the guarantee can be translated into a guarantee for convergence in the solution without any significant loss in the runtime complexity of the method. We demonstrate this theoretically and experimentally below. We thank the reviewer for pointing out that this is inadequately explained in the paper, and we will clarify this in the final version of the paper.

If $\boldsymbol{x}$ is a $(1+\delta)$-approximate solution, using Lemma A.1 from the supplementary material we can show that we can achieve the guarantee $\|\boldsymbol{x}-\boldsymbol{x}^\star\|_\infty \leq \varepsilon \|\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_p$ by picking $\delta = \left(\frac{\varepsilon \sigma_{\min}(\boldsymbol{A})}{4m}\right)^p$, where $\sigma_{\min}(\boldsymbol{A})$ is the smallest singular value of $\boldsymbol{A}$. This gives $\log \frac{m}{\delta} = O(p \log \frac{m}{\sigma_{\min}(\boldsymbol{A})\varepsilon})$, and hence a total iteration count of $O(p^{4.5} m^{\frac{p-2}{2(p-1)}} \log \frac{m}{\sigma_{\min}(\boldsymbol{A})\varepsilon})$. Asymptotically, the running time bound is only off by a factor of p if we wish to measure the convergence in ℓ_∞-norm, as long as $\log \frac{1}{\sigma_{\min}(\boldsymbol{A})} = O(\log \frac{m}{\varepsilon})$.





We also demonstrate this relation experimentally. The plots demonstrate the average resulting ℓ_∞ norm deviation for the solution computed, as we change the ε parameter used in the algorithm. We use the instances described in the paper; matrices of size 1000×800 and graphs with 1000 nodes. For each instance, we: 1) find a very high accuracy solution, by choosing a very small $\varepsilon \sim 10^{-25}$, 2) scale the problem so that the optimum value is 1, and run the algorithm again to find the optimum solution $\boldsymbol{x}^\star$. 3) Now we have a problem such that $\|\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_p = 1$, we run the algorithm again with various values of ε, to obtain solutions $\boldsymbol{x}(\varepsilon)$ and plot $\|\boldsymbol{x}(\varepsilon) - \boldsymbol{x}^\star\|_\infty$ (averaged over 20 samples). These results are very much in agreement with the theoretical $\varepsilon^{\frac{1}{p}}$ dependence proved above. (Note that the error bars indicate $\log(\text{mean} \pm \text{std})$ so they are missing on one side when mean $<$ std.)

Runtime comparison with [AKPS19] and [BCLL18] (R1). As noted by R1, the running time of [AKPS19] (and [BCLL18]) is not stated precisely in the comparison. The running time bounds are not stated precisely in either paper; they hide the p dependencies and $\text{poly}(\log \frac{m}{\varepsilon})$ dependencies. We have focused on the polynomial terms in the comparison because they are the dominant terms. For [AKPS19] the running time is at least $p^{2p+2} m^{\frac{p-2}{3p-2}} \log^2 \frac{m}{\varepsilon}$, for [BCLL18] it seems to be at least $p^{2.5} m^{\frac{p-2}{2p}} \log^2 \frac{m}{\varepsilon}$. The $\log^2 \frac{m}{\varepsilon}$ dependence is worse for both [AKPS19] and [BCLL18], compared to our algorithm, and the p^{2p+2} factor is much worse in [AKPS19]. We will clarify this in the paper.

Experimental comparison to [AKPS19] and [BCLL18] (R1). We agree that a direct comparison to [AKPS19] and [BCLL18] is desirable. Unfortunately, both algorithms are quite complicated to implement, and no implementations are publicly available. The [BCLL18] paper lacks an explicit algorithm description and leaves out several details (e.g. it asks to run accelerated gradient descent (AGD) "until convergence", the specific accuracy target for AGD will have a large impact on the running time). The [AKPS19] algorithm description also leaves out specifying several parameters in the algorithm, hiding p dependencies and $\log \frac{m}{\varepsilon}$ factors. As pointed out above, these large hidden factors make the algorithm, as stated, difficult to implement efficiently. In contrast, our algorithm is far simpler to implement.

Simplicity of p-IRLS compared to [MPT+18] (R3). We thank R3 for this. We will clarify this in the final version.

Combining p-norm with a regularizer e.g. ℓ_1 (Lasso) (R3). This is definitely a great idea for future work. Our current techniques would not suffice for this, but we thank the reviewer for pointing out this potential direction.

Spacing between subfigures in figure 4 (R1) We will address this in the final version.

Proof of claimed bound. We prove the bound on $\|\boldsymbol{x}-\boldsymbol{x}^\star\|_\infty$ claimed above. Given that $\boldsymbol{x}$ is a $(1+\delta)$-approximate solution, using Lemma A.1, we can write the following lower bound on the objective value:

$$(1+\delta)\|\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_p^p \geq \|\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_p^p + p\left(\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{b}\right)^\top \boldsymbol{R}\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^\star) + {}^{p}/_{8}\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^\star)^\top \boldsymbol{A}^\top \boldsymbol{R}\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x})^\star + 2^{-(p+1)}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}^\star\|_p^p,$$

where $\boldsymbol{R} = \text{diag}(|\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{b}|^{p-2})$. Since the gradient at $\boldsymbol{x}^\star$ is 0, simplifying, we get, $2^{p+1}\delta \|\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_p^p \geq \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}^\star\|_p^p$. Now, translating between various norms, we obtain,

$$\|\boldsymbol{x}-\boldsymbol{x}^\star\|_\infty \leq \frac{1}{\sigma_{\min}(\boldsymbol{A})}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}\boldsymbol{x}^\star\|_2 \leq \frac{m^{\frac{1}{2}-\frac{1}{p}}}{\sigma_{\min}(\boldsymbol{A})}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}\boldsymbol{x}^\star\|_p \leq \frac{2m^{\frac{1}{2}}}{\sigma_{\min}(\boldsymbol{A})}\left(\frac{2\delta}{m}\right)^{\frac{1}{p}}\|\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{b}\|_p.$$