We answered all questions posed by the reviewers and added comparisons with other five algorithms they asked about. Our methods still provide lower recovery error than any competitor. We sincerely thank the reviewers for their comments and time spent on our paper.

Response to reviewer $\#1$ We propose to estimate d as $d = |\Omega|/(m+n)$. Given a rank-r matrix $X \in \mathbb{R}^{m\times n}$, the number of degrees of freedom is $(m+n)r - r^2$. Suppose the number of observed entries is $|\Omega|$. Then $|\Omega| \geq (m+n)r - r^2$ ((1)) should hold; otherwise, X can not be determined uniquely. Considering incoherence property and random sampling, Candès and Recht (2009) proved that the minimum number of observed entries required to recovery X (whatever methods used) with high probability is $C\mu nr\log n$ (suppose $m \leq n$), where $\mu \geq 1$. It means $r \leq |\Omega|/(Cn\log n)$ ((2)). Our method FGSR requires $d \geq r$. Thus, according to inequalities (1) or (2), we set $d = |\Omega|/(m+n)$.

We added truncated nuclear norm [ex1], weighted nuclear norm [ex2], and Riemannian pursuit [ex7] to the experiments. Figure 1(a) shows that the recovery errors of the three supplemented methods are higher than those of our FGSR methods when the missing rate is high. Note that in truncated nuclear norm, we have used the true rank (though difficult to know beforehand in practice); otherwise, the recovery error will be much higher. Figure 1(d) shows that our FGSR methods are much faster than all methods except Riemannian pursuit. In Figure 2(a)(b), FGSR methods also outperformed Riemannian pursuit. In the noisy cases (Figure 2), FGSR was solved by PALM (faster than ADMM used in the noiseless case) and its time costs are within $[1s, 3s]$, while Riemannian pursuit's time costs are within $[1.5s, 2.5s]$. Note that the code of Riemannian pursuit was written by mixed programming C&MATLAB, which is much faster than pure MATLAB (utilized in all other methods). In Figure 3, FGSR methods outperformed Riemannian pursuit on real data. In sum, our FGSR methods are more accurate than all other methods. In terms of computational cost, FGSR methods are comparable to Riemannian pursuit and are much faster than other methods.

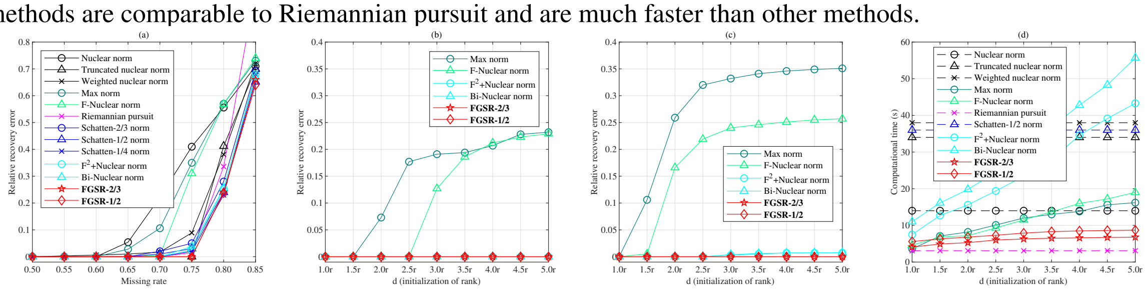

Figure 1: Matrix completion on noiseless synthetic data: (a) different missing rate; (b)(c) different rank initialization (missing rate $=$ 0.6 or 0.7); (d) computational cost (missing rate $=$ 0.7).

Response to reviewer $\#2$ We added the results of FGSR-1/2 in the experiments (shown in Figures 1, 2, and 3). FGSR-1/2 is more accurate than FGSR-2/3. We also added the comparison of the improved case of Bi-nuclear norm (S-2/3, Shang et al. TPAMI2017), which is denoted by F^2+Nuclear norm. As shown in Figure 1, our FGSR-1/2 and FGSR-2/3 are slightly more accurate and much faster than Bi-nuclear norm (S-1/2) and F^2+Nuclear norm (S-2/3). Similar comparative results can be found in the noisy cases and the results of Bi-nuclear norm, F^2+Nuclear norm, Schatten-2/3, and Schatten-1/4 were omitted in Figure 2 for simplicity.

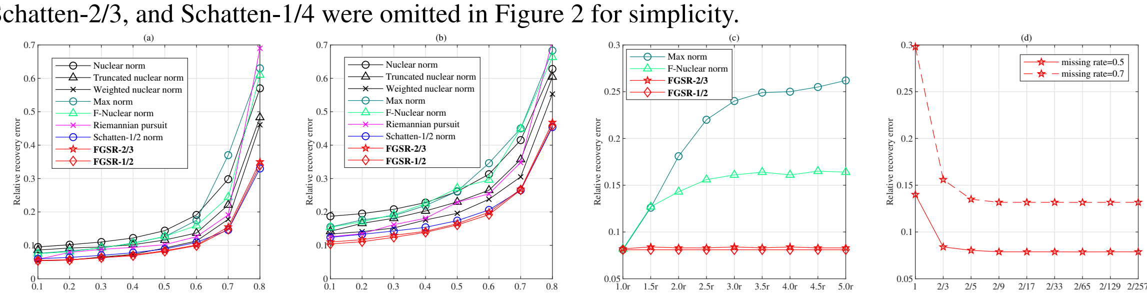

Figure 2: Matrix completion on noisy synthetic data: (a)(b) recovery error when SNR $=10$ or 5; (c) the effect of rank initialization (SNR $=10$, missing rate $=0.5$); (d) the effect of p's value in Schatten-p norm (solved by FGSR if $p<1$).

Response to reviewer $\#3$ Our motivation is to provide a class of SVD-free and accurate nonconvex regularizations for matrix rank with theoretical guarantees. We improved the illustration of our motivation according to your suggestion. The numerical results (e.g. Figure 2(d)) showed that smaller p leads to lower recovery error but the improvement is not significant when p is too small (e.g. <2/5). The phenomenon is consistent with our generalization error bound. Therefore, in practice, we suggest using $p = 2/3$ or $1/2$ because they are faster than $p \leq 2/5$. The results of FGSR-1/2 have been added to Figures 1, 2, and 3. FGSR-1/2 is more accurate than FGSR-2/3 but is slightly slower. In sum, we suggest FGSR-1/2 if time cost is relatively less demanding.

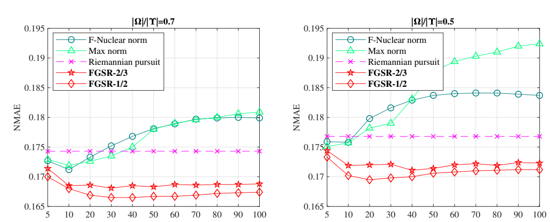

Figure 3: NMAE on Movielens-1M data