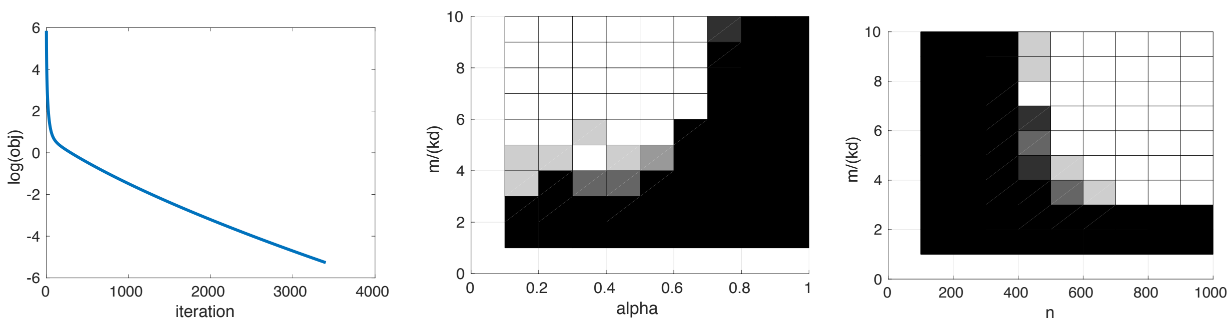
We thank all the reviewers for their constructive feedback!

To Reviewer #1. 1. We agree (and will acknowledge more explicitly) that the overall proof program is similar to existing results in the area. However, NIMC problem presents two key challenges: a) The Hessian has entangled terms for items' features and queries' features, which are challenging to handle. 2) In addition to non-convexity arising due to non-linearity of the activation function which standard 1-2 hidden layer NNs also face, we have to handle additional noise/uncertainty due to missing ratings, and provide strong sample complexity bounds for the results to be meaningful.

2. We'll reduce section 3.4 to a short sentence.

3. Here we provide more experimental results in Fig. 1. We use sigmoid as the activation function, and set $k = 10, d = 100$, which are larger than those in the paper ($k = 5, d = 10$). We set the initialization as $W^{(0)} = (1-\alpha)W^* + \alpha W^{(r)}$, where W^* is the ground truth, $W^{(r)}$ is a Gaussian random matrix, and $\alpha \in [0, 1]$. In (a), $\alpha = 0.1, n = 1000$, and $m = 10000$. In (b), $n = 500$. In (c), $\alpha = 0.1$. The other settings are same as those in the paper. As we can see, (a) shows how the objective value converges, which is almost linear. (b) shows that when the initialization is purely random ($\alpha = 1$), gradient descent doesn't converge to the ground truth. In the paper, when $k = 5, d = 10$, pure random initialization still converges to the ground truth. We believe that it is because when k, d are larger, random initialization can be further away from the ground truth. Hence, gradient descent can get stuck in local optima more easily. Finally, comparing (c) with Fig. 1(a) of the paper, we can obtain a similar conclusion, i.e., when n is sufficiently large, the number of observed ratings required for successful recovery remains the same.



(a) log(obj) v.s. iteration. (b) Recovery rate v.s. initializations. (c) Recovery rate v.s. (m, n).

4. To remove the redundancy in the ReLU case, we assume that $u^*_{1,i}$ is nonzero for all $i \in [k]$ and know the number of positives in $\{u^*_{1,i}\}_{i=1,\cdots,k}$. Note that if the columns of U and the columns of V do the same permutation, the output doesn't change. Without loss of generality, we can assume $\{u^*_{1,i}\}_{i=1,\cdots,k_+}$ $(0 \le k_+ \le k)$ are positive and and the remaining $\{u^*_{1,i}\}_{i=k_++1,\cdots,k}$ are negative. So if we fix $u_{1,i} = 1$ for all $i \le k_+$ and $u_{1,i} = -1$ for all $i > k_+$, we can remove the redundancy and the target solutions for U and V are $u_{:,i} = u^*_{:,i}/|u^*_{1,i}|$ and $v_{:,i} = v^*_{:,i}|u^*_{1,i}|$ respectively.

To Reviewer #2. We will like to stress that Non-linear Inductive Matrix Completion is a significantly different architecture than the 1-layer NNs and hence theoretical analyses for the two are quite different. As mentioned in response to R1, while the high level approach is same, we have to deal with non-linearity of NNs along with the noise due to missing ratings and entangled Hessian due to non-linearity in both item's features and query's features. These challenges require a significantly different analysis than existing results.

To Reviewer #3. Movielens dataset: our main goal in these experiments is to study the problem in the *inductive* setting, i.e., to predict ratings for *new* users. R3's observations for collaborative filtering (CF) are valid but they apply only to *transductive* setting which does not allow for new users.

a) SVD based solution: SVD based CF does not predict ratings for new users and hence does not apply in the inductive setting. Furthermore, as we are predicting ratings for completely new users, for which only weak features are available, naturally the resulting RMSE is worse than the results for the standard collaborative filtering settings (where several ratings of a user are available a priori).

b) Generalization error: as mentioned above, the only information about new users is their relatively weak features, hence non-linear methods can extract more information from them compared to the linear ones, and might be the reason for the superior performance of NIMC over IMC.