Reviewer #1

We thank the reviewer for the constructive feedback. We will make the suggested clarifications and fix the typos.

The framework of the paper uses the model to improve the reparameterization directly. When the model is not specified or perhaps does not exist, using covariates as an alternative objective to optimize could be an extension of the current framework. Reparameterizing in such an extension is an interesting future direction to explore.

Reviewer #2

We thank the reviewer for the constructive feedback. We will clarify Figure 7. For 7(b), the intuition is that the learned reparameterizations put more weight on movies with higher average ratings (x-axis) and higher variation (y-axis) — the reparameterization is focused on distinguishing between different top-rated movies with some variance in opinions to specialize the recommendation set to the particular users.

Linearity of reparameterization. Whereas this paper shows that linear reparameterization provides a significant benefit, we agree that this opens the door to further research in reparameterization. In particular, exploring non-linear reparameterizations is an interesting future direction. It will require substantial additional development due to the resulting non-convex constraints, and the theoretical guarantees may not hold.

Theorem 2. We thank the reviewer for pointing out the confusion. We will remove the term "for simplicity" as recommended. Extending to convex objectives is an interesting topic for future work. It would allow us to adopt more flexible (e.g., convex) reparameterization while maintaining the theoretical guarantees.

Reviewer #3

We thank the reviewer for the constructive comments. We want to highlight that our approach should be understood in terms of how we reframe the predict-then-optimize problem in a conceptually different way. By doing optimization in a learned representation space instead of the problem's original space, we enable substantial benefits against the state-of-the-art approaches.

Theorem 1. Yes, Theorem 1 still holds without the condition $P \geq 0$. Throughout the paper, we assume the feasible region (if bounded) to be in the first quadrant, so that a non-negative reparameterization suffices. The reviewer is correct that the theorem and the proof still holds without this condition. We thank the reviewer for pointing this out and will clarify this in the write-up.

Theorem 2. The constant C is defined as $C \coloneqq sup_\theta(max_x f(x, \theta) - min_{x'} f(x', \theta))$. We will clarify this by adding a formal definition.

Convergence of the predictive model. When the objective function is linear and the hypothesis class has a finite Natarajan dimension, the generalization error will converge to 0 as the number of training examples approaches infinity. This indicates that the performance of the predictive model will converge to its expected performance too. (Technically, the parameters of the predictive model may not converge as they could alternate between two optimal solutions, but the performance converges asymptotically.) We will clarify this in the write-up.

Reviewer #4

We thank the reviewer for the constructive suggestions. We will make the suggested clarifications and fix the typos.

Theorem 2. We agree that extending to non-linear objective functions is an open and interesting question. In particular, our empirical results have shown that our reparameterization approach also works for non-linear objective functions. We think the theoretical result for a linear objective serves as an important step toward the convex case. The sample complexity of the linear case depends on the slope of the linear objective function (constant C in Equation 4). An analogous term (e.g., Lipchitz constant) will likely appear in the convex case, so the result would likely be in terms of convex functions that are Lipschitz over the feasible region.

Time complexity. Using a smaller dimensional reparameterization reduces both the theoretical and empirical time complexity, despite having to learn and back-propagate through an additional parameter P. The reduced computational cost includes 1) inverting a smaller dimensional KKT matrix which takes roughly cubic less time 2) solving a lower dimensional optimization problem. The increased computational cost includes 1) matrix-vector multiplication $x = Py$ and 2) additional back-propagation to the parameter P, which is also matrix-vector multiplication and thus takes square time. Thus, overall the time complexity is reduced.

Hyperparameters. We hand-tuned the learning rate and reparameterization size for all competing methods. We will add more details about how the parameters are chosen to the appendix. We agree that adding an additional experiment varying reparameterization size would be informative for hyperparameter selection. We will also add this to the appendix.