We thank the reviewers for their time and thoughtful feedback. Taking their helpful comments into account, we have sought to clarify the presentation of our work. We have also extended the bias-variance tradeoff from the two task case to multiple tasks (see description below), which addresses the main criticism by R1 and R2.

Why do we focus on predicting a particular task? [R2] As R2 pointed out, we focus on a situation where we have a target task for which we only have limited labeled data and several source tasks. We study when training the tasks together can benefit the target task. While this setting is different from traditional MTL that studies the average performance of all tasks, it is also a common setting in practice. For example, the task of interest might be about predicting a rare event or classifying an Xray-scan. For such settings, collecting large amounts of labeled data for the task is either not possible or very expensive, but auxiliary labeled data are often easier to obtain. Traditional MTL theory that studies the average performance of all tasks does not help predict whether training the tasks together can benefit the target task. Our theoretical framework applies to this setting and takes a step towards filling the gap.

What can we say for multiple tasks? [R1, R2] (1) We have focused on the two task setting in the submission to provide insight, since this is simplest setting which we don't understand how tasks transfer in MTL. **(2)** We understand that the setting of multiple tasks is more general, therefore, we have *extended our result on bias-variance tradeoff of two tasks to multiple tasks*. That is, we can now show that *as long as the output dimension of the shared layer B is smaller than the total number of tasks, the variance of the MTL estimator for the target task is always smaller than the variance of the STL estimator but the bias is always larger*. We have included this result in the updated draft. **(3)** For multi-label settings where all tasks have the same features, i.e. $X_i = X$ for any i, using Theorem 3.6 *all of our insight for two tasks except covariate shift applies to multi-label settings* (covariate shift does not apply since tasks have the same features).

Writing: [R2, R3] We have corrected the typos that R2 pointed out and clarified the issues that R3 raised. **(1)** L112-118: we use t to denote the number of tasks hence for two tasks $t = 2$. **(2)** Validation set size: we only need it to be larger than the size of the hidden layer times the number of tasks, which can be much smaller compared to the size of the training set (cf. L108). **(3)** Def. of the prediction loss L113: the expectation is over a test sample x whose label is $x^\top \beta_t$. Taking expectation over ε gives the bias-variance decomposition, following standard linear regression literature [17,18].

R1: We thank R1 for suggesting that we look at qualitative predictions of Thm 3.6 as in the two-task case, which we have added in the updated draft. For task similarity, the more similar tasks are, the closer $\|v_t\|$ is to 1 and the more variance reduces, which leads to positive transfer as in Prop 3.3. For sample ratio, the more dissimilar tasks are, the more bias increases by source task samples, which leads to negative transfer as in Prop 3.4. R1 asks how does our method compares to standard techniques such as loss reweighing. Note that our method is equivalent to increasing the weight of a task until performance drops. Our method is preferable to loss reweighting since we require less compute over the training data as shown in Section 4.2. We thank R1 for pointing out the vague use of "similar performance" in experiments, which we have replaced w/ the accuracy numbers (that are comparable).

Regarding R1's comment about computing similarity via distance between classifier parameters, we have tried it for predicting whether MTL outperforms STL (smaller distance implies better transfer) but the result is worse than Table 1. We suspect the reason is that the distance mainly captures difference between the trained model but does not capture other properties of task data. The closest work to ours is [15] and that work uses standard concentration bounds to show that when two tasks are sufficiently similar, MTL guarantees positive transfer. Our result in Thm 3.2 does not make such an assumption by using advanced tools from random matrix theory. This also allows us to study the impact of varying sample sizes and covariate shift, both of which cannot be studied using standard concentration bounds.

R2: We thank R2 for bringing up the confusion of which sample size regime does our theory/algorithm apply, which we have clarified in the updated draft. **(1)** R2 is correct that "our theory applies when the sample sizes are tens or hundreds of feature dimension". We think this is a reasonable regime to consider; for example, in our sentiment analysis experiment, the feature dimension of a sentence is 300 and the training set size ranges from 3 to 10 thousand. **(2)** R2 mentions "having imbalanced sample size btw source/target task": Our incremental training scheme does not assume that the tasks have imbalanced sample size; for example, in our sentiment analysis experiment, we have observed that our method can be effective even when the source task is smaller than the target. As shown in our theory, the transition threshold between positive/negative transfer provably depends on task similarity and can be less than one (Prop. 3.4).

We thank R2 for pointing out the connection between our incremental training procedure and curriculum learning. We are not aware of any previous work that proposes such an idea in MTL while having a strong theoretical basis. Adding more context, there is an ongoing discussion of how much data from each task the model should be trained on (cf. Google T5 and refs therein). While we have focused on evaluating training efficiency, it's conceivable that by combining our procedure w/ other ideas one might get better final performance of the target task. It is an interesting research question to further investigate the idea in future work.

R3: Here's our detailed response. **L108:** We disagree that "the validation set is much larger than the training set" is suggested anywhere in the paper. **L113:** We disagree that "there are duplicate notations" in this line - t is the number of

tasks not samples. **L117:** The sample covariance of task 1 is *not* Σ_1 but $X_1^\top X_1$ and it shows up in both eqs. **L187:** γ is a free parameter and by varying it one can recover the entire precision-recall curve. **L220:** Our theory provides a theoretical basis for the algorithm. For two tasks, the algorithm can provably find the optimal sample ratio. As shown in Fig 1b, the performance curve, which is a quadratic function, has a single peak and our algorithm stops at the peak. The fact that the curve is quadratic is shown in our proof and we have added the connection to the updated draft.