We thank the reviewers for their excellent feedback, which significantly improved our work. All reviewers agree that our work provides a solid foundation for understanding when MTL performs well compared to STL, a central open question in MTL. We offer better explanations of our theoretical/algorithmic contributions in this response.

Motivation. [R2] We focus on a setting where we have a target task with limited labeled data and study when using MTL w/ multiple source tasks helps the target task. There are many practical applications for this setting from both industry and academia. For example, for predicting a rare video event or classifying an Xray scan, collecting large amounts of labeled data is very expensive. Still, auxiliary labeled data are often easier to obtain. For these settings, while applying MTL is natural, the result can be worse than STL. Classical MTL theory studies the average performance of all tasks and does not address whether MTL helps the target task. Our work builds a foundation for applying MTL to these settings.

What can we say for multiple tasks? [R1, R2] While we have focused on the two-task case to explain our results in the original submission, we can apply all of our conceptual insights to the multi-task case. We have updated our draft to consolidate our results for the multi-task setting and now show the following. **(1)** We show that *as long as the output dim. of the shared layer B is smaller than the number of tasks, the variance of the MTL estimator for the target task reduces compared to STL, but the bias increases*. The idea is similar to the two-task case, and we have included this result in the draft. **(2)** For multi-label settings where all tasks have the same features, i.e., $X_i = X$ for any i, using Thm 3.6, all of our insights from the two-task case still apply except for covariate shift, which doesn't apply since tasks have the same features. For task similarity, the more similar the models are, the more variance reduces ($\|v_t\|$ closer to 1), which leads to positive transfer as in Prop 3.3. For sample ratio, the more different the models are, the more bias increases w/ more source task samples, which leads to negative transfer as in Prop 3.4.

Related work: [R1, R2] We clarify how our work differs from previous work. The closest work to ours is [15], which uses standard concentration bounds to show that when two tasks are similar, MTL ensures positive transfer. Our paper uses new tools from random matrix theory, and Thm 3.2 doesn't require tasks to be similar. Our tools allow us to analyze the empirical phenomenon of negative transfer that is challenging otherwise using standard techniques. In particular, Lemma 3.1 characterizes the variance of MTL w/ covariate shift, which may be of independent interest.

Writing: [R2, R3] We have corrected the typos that R2 pointed out and added more explanation for the questions that R3 asked. **L112-118**: We use t to denote the number of tasks, hence for two tasks $t = 2$. **L108**: We only need the validation set to be larger than the size of the hidden layer times the number of tasks t, which is very small compared to the training set size. **L113**: For the prediction loss, the expectation is over a test sample x whose label is $x^\top \beta_t$. Taking an expectation over ε gives the bias-variance decomposition, following standard linear regression literature [17,18].

R1: (♦) R1 asks how our method compares to loss reweighing. Our approach is equivalent to increasing task weight until performance drops. An advantage of our approach is that we only train over a subset of samples, whereas loss reweighting uses the full set. **(♦)** We thank R1 for pointing out the vague use of "similar performance" in experiments, which we replaced w/ (comparable) accuracy numbers. **(♦)** The setting of 5 tasks w/o TREC: For any set of 5 tasks w/o one task, the result is qualitatively similar. **(♦)** R1 suggests computing similarity via distance between layer parameters, which we tried (along w/ SVCCA) as a proxy for task similarity. It didn't perform as well as our result in Table 1.

R2: We thank R2 for bringing up the confusion about the sample size regime our theory/algorithm applies. We have clarified this in the draft. **(1)** R2 is correct that "our theory applies when the sample sizes are 10-100x of feature dim". We think this is a practical regime to consider. For example, in our sentiment analysis experiment, the feature dim. of a sentence is 300, while the training set size ranges from 3k to 15k. **(2)** R2 refers to having an imbalanced sample size while discussing our approach. We will clarify the writing after Thm 3.2. But our incremental training method does not assume that the tasks have imbalanced sample sizes. For example, in our sentiment analysis experiment, we have observed that our approach can help where the source/target sample ratio is $\leqslant 1$. One practical takeaway of this work is that the level of "imbalance" depends on task similarity, and it can be provably small, e.g., $\leqslant 1$ cf. Prop 3.4.
We thank R2 for pointing out the connection between our incremental training method and curriculum learning. We are not aware of any previous work suggesting adding training data for MTL progressively while having a strong theoretical basis. More broadly, there is an ongoing discussion of how much data from each task the model should be trained on (cf. Google T5 and its references). We have focused on evaluating training efficiency since our primary goal is to build/validate the foundation. It's conceivable that combining our method w/ other ideas could improve the accuracy of predicting the target task. As R1 also suggested, we think this is an excellent direction for future research.

R3: We thank R3 for the comments. **L220:** Value of our theory to our algorithm: For two tasks, we can show that our algorithm provably finds 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 point. We provide a proof that the curve is quadratic, and highlighted the connection in our draft. **L187:** Parameter of the metric: One can recover the entire precision-recall curve by varying it. **L108:** As we stated, a validation set ... larger than $r \cdot t$ suffices. We will clarify a bit more. Replacing $p^{0.99}$ w/ $p^{0.5}$ fixes the issue. **L117:** The sample covariance of task 1 is $X_1^\top X_1$, not Σ_1.