Method	Segmentation	Depth	Surface Normal	
	mIoU	Abs Err	Angle Distance	Within 11.25°
Cross-Stitch	15.69	0.6277	32.69	21.63
Cross-Stitch + PCGrad	**18.14**	**0.5805**	**31.38**	**21.75**
Dense	16.48	0.6282	31.68	21.73
Dense + PCGrad	**18.08**	**0.5850**	**30.17**	**23.29**

Table 1: As requested by reviewer 2, we show new experiments with PCGrad combined with other methods on three-task learning on the NYUv2 dataset.

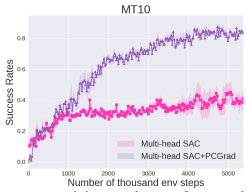

Figure 1: As requested by reviewer 2, we show the comparison between Multi-head SAC and Multi-head SAC + PCGrad on MT10.

We thank all the reviewers for the constructive feedback. We will incorporate the valuable suggestions in the revised version. We have addressed all of the comments below:

R4, Q1: I suggest the authors to draw some plots of the cosine similarity between the gradients from different losses? We already have these plots in the paper, see Figure 4 in Appendix B. We will move these plots into the main paper. The middle plot in Figure 4 visualizes the cosine similarity between the gradients of two Meta-World tasks. Based on the plot, we can see that the cosine similarity between task gradients are negative more than 50% of the iterations and condition (a) in Thm 2 holds for most of the time that both tasks haven't been solved. This suggests that the conflicting gradients problem is indeed a challenge in practice.

R4, Q2: comparison with more methods, e.g. GradNorm [1] and Sener et al [2]? We already compare to [1] in the rightmost plot in Figure 3. We already compare to [2] in Table 3 and Table 5 in Appendix F. PCGrad outperforms both [1] and [2]. We will make sure these comparisons appear prominently in the main paper.

R4, Q3: The output of Alg. 1 is not deterministic due to random sampling ... the generalization of Theorem 1 and 2 to the case $n > 2$ (the number of task loss functions) is non-trivial? We use random sampling to make our gradient projection procedure symmetric to the order of projection in expectation. We already provided analysis of the convergence of Theorem 1 where we have more than 2 tasks in Corollary 1 in Appendix A.1. We will also include the generalization of Theorem 2 with $n > 2$ in the final paper.

R1, Q1: Limited significant ML contributions and experiments? Our paper identifies three conditions that lead to poor MTL performance and proposes a new MTL algorithm that tackles them. In the paper, we provide the theory that motivates the practical algorithm and observe strong empirical results on 8 challenging benchmarks (5 supervised MTL benchmarks, 2 MTRL benchmarks and 1 goal-conditioned RL benchmark). Hence, we think the paper meets the standard of NeurIPS papers in terms of ML contributions and experiments.

R2, Q1: The results except for Table 1 are not totally better than the baselines, and even worse in the depth task in Table 4? PPCGrad is better than the baselines in four out of five supervised learning and all three RL experiments. In Table 1, PCGrad outperforms all prior methods in segmentation and depth, and in 4/5 metrics of surface normal prediction. In Table 4 as R2 said, PCGrad does not outperform the baseline in the depth task. This is potentially because CityScapes dataset only contains 2 tasks and the two tasks may not have much interference.
We also emphasize that the results are not cherry-picked. We report results on all benchmarks that we tried, and seven of the eight benchmarks were proposed in prior works.

R2, Q2: PCGrad alone...Conduct more experiments in which PCGrad is combined with more methods? PCGrad is considerably simpler than routing networks [44], as it doesn't require an RL optimization and requires only a few lines of code on top of a vanilla multi-task learning model.
We added more experiments to test whether PCGrad can improve performances when combined with more methods. In Table 1 of the response, we find that PCGrad does improve the performance in all four metrics of the three tasks on the NYUv2 dataset when combined with Cross-Stitch and Dense. In Figure 1 of the response, we also show that PCGrad + Multi-head SAC outperforms Multi-head SAC on its own. We will include the additional experiments in the final paper.

R2, Q3: the assumptions of convexity of the losses are probably too strong in Theorem 1? In Proposition 1 in Appendix A.1, we show that PCGrad converges to a stationary point when the losses are nonconvex, which is more connected to practice.

R3, Q1: There was no special reference to run-time and computation complexity? For all the multi-task supervised learning experiments, PCGrad converges within 12 hours on a NVIDIA TITAN RTX GPU while the vanilla models without PCGrad converge within 8 hours. PCGrad consumes at most 10 GB memory on GPU while the vanilla method consumes 6GB on GPU among all experiments. For the multi-task RL experiments, PCGrad + SAC converges in 1 day (5M simulation steps) and 5 days (20M simulation steps) on the MT10 and MT50 benchmarks respectively on a NVIDIA TITAN RTX GPU while vanilla SAC converges in 12 hours and 3 days on the two benchmarks respectively. PCGrad + SAC consumes 1 GB and 6 GB memory on GPU on the MT10 and MT50 benchmarks respectively while the vanilla SAC consumes 0.5 GB and 3 GB respectively. We will include the discussion in the final paper.