We thank the reviewers for their insightful and constructive comments.

[R1] Multi-task not novel. We would like to clarify that our contribution is far beyond multi-task learning: a unified and general framework with 3 mechanisms to take advantage of verification tasks to help regression methods. In fact, the multi-task learning only contributes +0.4 mAP gains, while others contribute +1.5 mAP gains (see Tab. 4).

[R1] several works using keypoints, e.g. CentripetalNet and RetinaFace. First, our method is beyond using corner/keypoint verification: it generally exploits various verification tasks to help regression methods, e.g. the foreground verification task, which is not exploited by CentripetalNet and RetinaFace. Second, even considering corner verification alone, our work is significantly different from CentripetalNet and RetinaFace: CentripetalNet is an improvement of CornerNet with a better corner matching mechanism, which is purely verification-based. In contrast, our method focuses on how to combine corner verification into regression methods to improve object localization. RetinaFace requires explicit keypoint annotations and only exploits multi-task learning, while our method does not require additional annotation and is far beyond pure multi-task learning. We will discuss these works in revision.

[R2] Add more description about RepPoints; still predict n sample points? Thanks for the suggestion and we will rewrite it in the revision. Yes, the understanding is correct.

[R2] Within-box verification and FCOS's centerness. We conduct experiments to include the verification task of FCOS's centerness. On a 40.5 AP baseline (+corner in Tab. 3 of this paper), an additional centerness branch achieves 40.6 mAP which has almost no gains over the baseline, while our additional foreground branch has +0.5 mAP gains (41.0 mAP vs. 40.5 mAP). On a 41.0 mAP baseline (+corner+foreground in Tab. 3 of this paper), an additional centerness branch achieves 41.0 mAP, which has no gains over the baseline. Actually, although FCOS's centerness and our within-box verification share some similarity, they are designed for different roles: while FCOS's centerness aims for adjusting object scores and weighting the regression loss, the within-box verification aims to complement the main regression branch. We will add discussions in the revision.

[R2] Explain why corner verification produces more accurate localization results. As described in Line 32-36, there may be two probable intuitive explanations: corner points represent the exact spatial extent of bounding box; each feature in corner point verification is well aligned to the corresponding point.

[R2] Why CornerNet outperforms RepPoints v2 in AP90. It is difficult to rigorously analyze all the factors for the performance gap in AP90 due to too many implementation differences. Nevertheless, we think there are probably two main factors: 1) our backbone is different (Res50 v.s HG-104), and some works have shown that ResNet architectures perform significantly worse than Hourglass ones in detecting corner points, e.g. 30.2 (R101) vs. 38.4 (HG-104) in Tab. 4 of CornerNet; 2) our highest resolution (C3) is lower than that of CornerNet (C2). A possible direction towards better AP90 is to use HG architectures and higher resolution, but it would significantly change the main branch of RepPoints (built on ResNet-FPN-C3-C7 structure), and we will leave it as our future works.

[R2] Why higher resolution does not yield better performance. The corner heatmap is used in both feature-level fusion and result-level joint inference. For feature-level fusion, we compared C3 heatmap with C3-C7 heatmaps, where we find the latter performs better. We hypothesize it is because more positive samples by C3-C7 benefit the training. For result-level joint inference, higher resolution does yield better performance that all levels using C3 heatmap performs better than that using the corresponding heatmap level. We will add the discussion in revision.

[R3] Terminologies. Thanks for the suggestion. We will carefully examine and revise them.

[R3] Ablation about hyperparameter r. $r = 1, 2, 3, 4$ produce mAP of $41.0, 40.8, 40.5, 40.2$, respectively, indicating $r = 1$ performs best. We will add this ablation in revision.

[R4] Details of joint inference. Intuitively, for a regressed corner point p^t, Eq. (2) means searching a point with the highest corner heatmap score in a neighborhood of r as its final location. Regressed points on all pyramidal levels use the C3 corner heatmap for searching, and r=1 corresponds to a 8-pixel neighborhood. We will polish the description.

[R4] Doubts about definitions in Tab.1. We agree with the reviewer's comment that the relatively smaller displacement regression targets in RetinaNet does not have benefit over RepPoints (or FCOS) which regress the box extent from the center point. However, this has no contradiction with our categorization of RetinaNet as an verification+regression framework and RepPoints as a regression framework: there are pre-defined anchors (coarse localization hypothesis) in RetinaNet, while there is no such coarse hypothesis for RepPoints (anchor-free). Moreover, our Paragraph 2 delivers the same information as the reviewer's comment, which motivates us to clarify the difference between two verification methods (CornerNet vs. RetinaNet, see Paragraph 3) and to further propose our method (Paragraph 4).

[R4] Computational costs and inference speeds. FLOPs comparison is described in Line 265-271. For real inference speed, the speed of RepPoints v1 is 12.7 FPS (img/s) using ResNet-50 on a Titan XP GPU, while that of RepPoints v2 is 10.1 FPS. With a ResNeXt-101-DCN backbone, the speeds are 4.3 FPS v.s. 3.8 FPS for RepPoints v1 and v2, respectively. We will add inference speed comparisons in the revision.