We sincerely thank all reviewers and ACs for their time and efforts. We also appreciate the positive comments on our novelty, contributions and state-of-the-art performance, *e.g.*, "impressive,", "meaningful", "detailed ablation study" and "well written and easy to understand". We have released code in GitHub anonymously, which can be searched by querying for "UWSOD". Below we give our responses point-by-point.

R1Q1 & R1Q2: More failure cases and qualitative results. The performance for the smaller objects in COCO.

A: We will analyze some failure cases and qualitative results in the supplementary material as suggested. The mAP of the small object in MS COCO is only 2.2%, as it is more challenge to detect small object than the large one in WSOD.

R1Q3 & R3Q2: The performance of the approach if they use Selective Search (SS), Edge Boxes (EB) or MCG.

A: Replacing SSOPG with traditional SS and EB proposals decreases 1.5% and 1.1% mAP on Pascal VOC 2007. And MCG proposals improve the detection performance of UWSOD by 1.5% mAP. However, MCG takes about 34.3 s. per image to generate proposals, which is time-consuming for practical applications. We will add the above results to paper.

R2Q1: The proposed method is quite complex and its presentation is in some points difficult to follow.

A: Thanks for your suggestions on presentation. We will revise the paper thoroughly under your suggestions.

R2Q2: These approaches seem to add a relevant computational cost. The authors should comment on that.

A: Our implemented WSDDN on VGG16 takes 439 ms. per images in GTX 1080TI GPU and PASCAL VOC. When we sequentially add SWBBFT, SSOPG and MRRP, the training times increase to 618, 848 and $1,581$ ms., respectively.

R2Q3: How the hyper-parameters are chosen without a validation set?

A: Most hyper-parameters are set based on the previous WSOD and FSOD methods. And the hyper-parameters of our components are manually chosen by intuition and experience, which are kept the same for different dataset.

R3Q1: How to set T_i^{obn}, T_i^p, $\bar{B}_i^{obn}$, $\bar{B}_i^p$, and λ^{obn}, λ^p? Are these two hyperparameters sensitive to performance?

A: T_i^{obn}, T_i^p, $\bar{B}_i^{obn}$, $\bar{B}_i^p$ are the classification and regression targets for objecness detection branch and object proposal generator, respectively. They are set based on IoU threshold between proposals and the corresponding pseudo-ground-truths. As we stated in the Implementation details, we set the labelling threshold λ^{obn} and λ^p to 0.5 and 0.7, respectively. And varying λ^{obn} and λ^p within a range of $[-0.1, 0.1]$ only decreases the performance by about $1.3 \sim 2.1\%$ mAP.

R4Q1: All three components considered by the authors have already been explored by existing works[11,17,72].

A: As we emphasized in the Introduction, there are significant differences between our method and the existing works. In particular, WSOD methods in [11,17] relied on external object proposal algorithms or additional instance-level supervision to learn detectors, while work in [72] focused on encoding traditional image descriptions in fully-supervised learning. To our best knowledge, our work is the *first* to propose learnable object proposals (SSOPG) without external modules or additional supervision and aggregate multi-scale in-network contextual information (MRRP) for WSOD task. And SWBBFT improves instance refinement from a new perspective of the trade-off between precision and recall requirements in different branches. Our method has both practical and methodological contributions to facilitate the development of this area. Moreover, detaching detectors from external time-consume modules makes WSOD more capable of handling thousands of real-world categories and taking advantage of large-scale weak annotations.

R4Q2: Why [17-20] cannot obtain such huge performance gain by using similar network modules (SWBBFT)?

A: Actually, careful designs of instance refinement module have shown significant improvement of performance in many works. The instance refinement proposed in OICR [8] improves WSDDN by 6.4% mAP. Recent methods in [17,18,20,35] proposed various strategy to improve the instance refinement module, which achieved gains of 5.4%, 7.0%, 8.9% and 6.0% mAP, respectively, compared to vanilla instance refinement in OICR. We attribute the improvement from SWBBFT to a sequence of effective refinement branches of increasing quality of pseudo-ground-truths and a well balance between positive and negative samples, which are neglected in the previous works.

R4Q3: It is not clear without meaningful proposals, how to train WSDDN, SWBBFT and SSOPG?

A: We further exhibit how our UWSOD works in the learning process. First, WSDDN formulates WSOD as multiple instance learning and captures the target object from a large set of proposals. Therefore, we enforce SSOPG to output redundancy object proposals to ensure high recall, which makes WSDDN capable of selecting positive samples. Although the output proposals of initial SSOPG are messy, WSDDN still has high-probability of finding informative proposals, which either contain discriminative object part or cover the entire object loosely. Second, with the output of WSDDN, a sequence of effective refinements in SWBBFT bootstraps the quality of predicted bounding boxes, which has been demonstrated to improve performance in many existing works [8,17-20,35-36]. Third, a self-supervised learning is leveraged to train SSOPG with supervision distilled from SWBBFT, which in turn improves WSDDN results.