We'd like to express our gratitude towards *all* the reviewers who recognized the novelties of the proposed intermediate representation for 3D object detection and the template-based prediction, as well as the significantly improved performance. We further appreciate R3 for commenting that *"predicting 3D properties by their projections is the right way to go and is the direction that the 3D vision community needs to hear more about. I believe future researchers should be more comfortable with this concept and use this as their default setup."*

R1, R2, R3: More design details about the templates. Are templates in the same class have different poses? The templates design is both class-specific and instance-specific: (1) Class-specific: we decouple the prediction of the perspective point and object class as illustrated in Eq.2 and L158-L159. (2) Instance-specific: the templates $T \in \mathbb{R}^{C\times K\times 2\times 9}$ are inferred for each RoI; hence, they are specific to each object instance as shown in Fig.3 and L152-L153. The templates are automatically learned for each object instance from data with the end-to-end learning framework; thus, both the templates and coefficients for each instance are optimizable and can fit the data better. We will make the description of the class-specific and instance-specific design more clear in revision using the extra page.

R3: What would happen if the intermediate representation is class agnostic? Very insightful observation. We theoretically and empirically explain our class-specific and instance-specific design. **Theoretically**, if we want to stay with Marr's theory [1], the 3D shape should be represented by class-agnostic *3D primitives* and class-specific *3D model descriptions* of a shape. By analogy, they are similar to the templates of perspective points and their coefficients in this paper, respectively. Hence, the intermediate representation should be class-agnostic by Marr's theory. However, the class-agnostic design would encourage the competition across classes during training, and the data we use in SUN RGB-D dataset is imbalanced (rare objects in certain categories). To avoid such competition, we make the intermediate representation **class-specific**. Similar with defining general 3D primitives like cuboid and cylinder, if we could design certain class-agnostic templates by analyzing the manifold in the projected 2D space (without data-driven method), it would be an exciting direction to see if it could learn the intermediate representation with such a class-agnostic way. Meanwhile, in complex indoor scenes with man-made objects, the number of object categories is large (>30 categories) with significant intraclass variations of appearance and geometry (*e.g.*, desk, lamp, and cabinet). Given a limited (5k) and imbalanced training data, it would be challenging to model the objects with rare appearance, 3D dimensions, and shapes if we purely use shared-templates within the same class. Therefore, we adopt an **instance-specific** template design to model the complex data distribution. **Empirically**, the results from the class-agnostic design tend to fit the distribution of most frequent object categories, making the performance of the rare objects much worse, and the pix error of the perspective point estimation much larger. Our experiments also indicate that an instance-specific design improves performance. However, a more principled method to define the template would be a combination of the shared-templates and instance-specific templates (similar with [70]), which would be a promising future direction. We will clarify and discuss the template design in revision using the extra page.

R3: Is 3D bounding box branch necessary? Comparison with optimization-based prediction. Two motivations: (1) The **single view ambiguity** due to projection. Given only the results from 2D box branch and perspective points branch, it is almost impossible to get a unique solution of the 3D bounding boxes; there exist multiple 3D bounding boxes with different sizes and distances that could be projected to the very same perspective points. (2) Assume the estimated 3D size or distance is given, it is possible to compute the 3D bounding box with an optimization-based method like efficient PnP. However, the optimization-based methods are sensitive to the accuracy of the given known variables. It is more feasible for tasks with smaller solution spaces (*e.g.*, 6-DoF pose estimation where the 3D shapes of objects are fixed), But it would be difficult for tasks with larger solution spaces (*e.g.*, 3D object detection where the 3D size, distance, and object pose could vary greatly). Therefore, we argue that directly estimating each variable with constraints imposed among them is an easier and more straightforward solution.

R1: Applying the proposed method to an outdoor environment will be interesting. We concur. It is interesting to see how our method will perform on outdoor 3D object detection dataset like KITTI. The differences between the indoor and outdoor dataset for the task of 3D object detection lie in various aspects including the diversity of object categories, the variety of object dimension, the severeness of the occlusion, the range of the camera angles, and the range of the distance (depth). We hope to adopt the proposed method in the future to the outdoor with ablation studies.

R1: Potential incorporation with depth information will be interesting. The proposed method estimates the distance between the 3D object center and camera center based on visual features (RGB without depth) only. If the depth were also provided, the proposed method would be able to make a much more accurate distance prediction. In theory, the depth information would help to improve the overall performance significantly, but it would make the problem less challenging or interesting, in our opinion. However, we hope to explore a better way to incorporate depths.

R2: Backbone design for the heatmap-based approaches may influence the perspective point estimation. It is possible to devise a two-stage method—detect the object in the first stage and infer the perspective point in the second stage with [34]. However, [34] is not originally designed for regressing the keypoints for multiple categories of objects. We will try our best to include an additional ablative study by comparing with networks similar to [34] in revision.