We thank all the reviewers for their valuable comments and positive feedback: R1) Indicating an important aspect of the zero-shot task; R3) Rigorous experiments; R4) A novel method. All the reviewers are satisfied with our paper's contributions, good performance, and clear presentation. Codes for reproducing all the experiments will be released.

Response to Reviewer-1.

R1.1 Comparisons with other papers. We note that previous works [2,27,48] experiment with different settings and datasets, which makes it hard to conduct unified and direct comparisons. We also considered ZS3Net[2] for direct comparison, yet found that its setting may be different from ours. We assume no knowledge of unseen classes in training, while ZS3Net learns implicitly with supervision of unseen classes. We would like to refer reviewers to Issue $\#1, \#4, \#6$ raised (by other researchers) in the official Github repo of ZS3Net for contrasting the settings. To ensure fair comparisons in our work, for the GMMN we adopted the implementation from ZS3Net. And all the methods are trained similarly with independently selected hyper-parameters for each.

R1.2 How our method works. As motivated in Fig.1, in zero-shot learning, optimizing over the noisy/abnormal samples of seen classes may result in models with a biased visual-semantic mapping, thus the inferred classier for unseen classes may be less reliable. In the scenarios of semantic segmentation, the noise appears in two levels: one is at image-level, e.g. some van cars look like 'bus' but are labelled as 'car'; the other is at pixel-level, e.g. pixels near boundaries are hard to distinguish and easily mis-annotated. Our uncertainty-aware model is proposed to address such challenges. We observed that the image-level learning path is critical to the accuracy of segmentation; yet, at the same time, the pixel-level learning path plays as a complementary role in learning refined segmentation details.

R1.3 How the uncertainty-aware learning works. Due to the noisy data collection and annotation process in practice, data samples may have different levels of uncertainty. For example, an abnormal sample is expected to make a less confident prediction than a typical sample, and pixels in the object centers are expected to be more confident than boundary pixels. The level of noise introduced for individual samples is called Heteroscedastic uncertainty and can be formulated as a random variable and estimated nonparametrically along with the learning of deep neural networks [12, 19, 28, 32, 38]. Specifically, in deep networks for classification or regression, the model is trained to learn a feature mapping such that the features are distributed in the form of a certain exponential family, with a unique variance and mean for all samples. Optimizing the feature mapping over noisy/abnormal samples may result in a biased distribution. Yet, explicitly modeling the uncertainty σ for each sample, as in our formulation, allows the model to adaptively adjust the variance for individual samples. As a result, all the data samples can be properly accounted for by the model with less bias toward the abnormal/noisy samples. During the optimization process, the abnormal/noisy samples are typically mapped far from the majority of the samples, thus leading to high uncertainty. We will clarify this in the paper.

R1.4 Parameter selection. We select hyper-parameters based on a validation set split from the seen-class training set.

R1.5 Eq 7: for sake of... We also tried with L-2 loss, but found the accuracy drops in all settings.

R1.6 Table 1: DeViSE results... One main reason may be that DeViSE maps the visual feature into the low-dimensional semantic space for nearest neighbor search. Such a process may shrink useful information and aggravates the Hubness problem in zero-shot learning. Moreover, in this work, we test with a much larger unseen class set than in previous works, thus further increasing the difficulty of classification for DeViSE.

R1.7 Table 2: Adding Random... We removed these two lines from this table to align the height of the figure in the right. On ADE20k data, the mIoU (in $\%$) of the unseen classes for Blank/Random is 11.2/ 5.8 (K=25), 9.3/ 5.1 (K=50), and 9.2/ 4.7 (K=75), which is much lower than our final accuracy. We will add this back.

R1.8 Fig 3b: what is the intuition... We guess that R1 refers to Fig 3c. We found this is a dataset-specific phenomenon and observed consistent improvements of unseen-class performance for all the methods on PC30 under this setting. The reason may be that removing the unseen-class images results in smaller but more compact training set for seen classes, thus leading to better visual-semantic mappings for inferring model for unseen classes. We also observed that the overall mIoU for our final model drops from 36.5$\%$ to 28.6$\%$ when removing all the unseen-class images for PC30.

R1.9 Other comments. We appreciate the detailed and very helpful comments. Due to the page-limit, we can only address part of them in this rebuttal. We will carefully address all the comments in a revised version of our paper.

Response to Reviewer-3.

R3.1 About pixel and observation noise. Yes, noise exists at both the image- and pixel-level. Please see our response to R1.2 and R1.3 for details.

R3.2 Baseline classifying pixels independently. Thanks for the suggestions. With only the pixel-level path, the model achieves mIoU for unseen classes with 13.5$\%$ for PC30, 8.8$\%$ for PC156, and 7.9$\%$ for ADE75. Compared to the results in Tab.3, we found that simply adding the image-level path can improve about 0.4$\%$, 1.9$\%$, and 5.6$\%$ for PC30, PC156, and ADE75 respectively. We will update our paper.

R3.3 Generator-based techniques. We denote [2] as GMMN in our experiments. Please also see our response to R1.1.

Response to Reviewer-4.

R4.1 Results for different uncertainty/noise level. Thanks for the suggestions. To evaluate the effects of different levels of noise, we randomly shuffle the class labels for a training sample. The unseen-class mIoU (in $\%$) for baseline/U-loss under different probability on PC-156 are 11.7/13.3 (p=0.0), 11.1/12.8 (p=0.1), 10.9/12.9 (p=0.25), 9.6/12.1 (p=0.5), which shows that uncertainty-aware learning achieves better robustness. We will update the paper.