Semantic-Promoted Debiasing and Background Disambiguation for Zero-Shot Instance Segmentation (Supplementary Material)

Shuting He[1†] Henghui Ding[2†✉] Wei Jiang[1]
[1]Zhejiang University [2]Nanyang Technological University
https://henghuiding.github.io/D2Zero

A. More Ablation Study

In Fig. I, we report the performance with different numbers of transformer layers utilized in our proposed Input-Conditional Classifier. Increasing the layer number from one to three brings a little performance gain. While overly large layers consume more computing resources and degrade the performance. Therefore, we just utilize one transformer layer to achieve a trade-off between accuracy and efficiency.

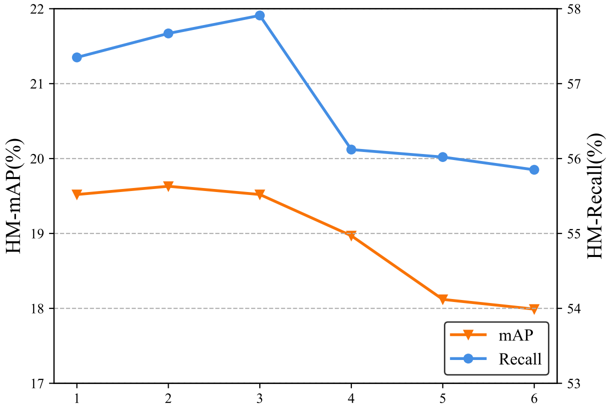

Figure I. Influence of different numbers of transformer layers in our proposed Input-Conditional Classifier.

B. Pseudo Unseen Label Visualization

To better understand the effectiveness of unseen CE loss, we visualize the pseudo unseen labels generated by our model via Gumbel-Softmax [1] according to semantic similarity $e_{i,j}$, as shown in Fig. II. For each of the seen classes, we sample 10,000 times on pseudo unseen label generation to make the results more reliable. From the figure, we can see that the unseen classes sharing more similar semantic characteristics with the seen class have a higher probability of being chosen as the pseudo unseen label. Taking `horse` in seen class as an example, the top four chosen unseen categories are `cow`, `dog`, `elephant`, and `cat`, which conforms to their semantic relationships with `horse`. Conversely, classes with lower semantic similarity to `horse` are rarely selected as the pseudo unseen label of `horse`, *e.g.*, `umbrella` and `snowboard`. Our pseudo unseen label method is superior to only taking the top 1 category as the unseen label because we can involve all the unseen labels in training with different frequencies, which greatly improves the generalization ability.

C. More Analysis on Unseen-Constrained Visual Feature Learning

We clarify the different effects on seen/unseen classes with the unseen constraint. Unseen constraint enables feature extractor to distinguish both seen and unseen classes, rather than exclusively seen classes. Feature extractor herein enhances generalization capabilities towards unseen classes at the cost of less overfitting to seen classes. Thus, the significant improvement of unseen classes may slightly sacrifice the performance of the seen classes.

D. More Background Mask Visualization

More qualitative background mask results by our approach are shown in Fig. III. Our generated background masks do a good job of excluding unseen classes from the background. The proposed adaptive background prototypes are extracted from the background regions. The predicted background regions and extracted background prototypes change according to the input image. They can better capture image-specific and discriminative background visual clues, which greatly helps to avoid mistaking novel objects for background.

†Equal contribution.
✉ Corresponding author (henghui.ding@gmail.com).

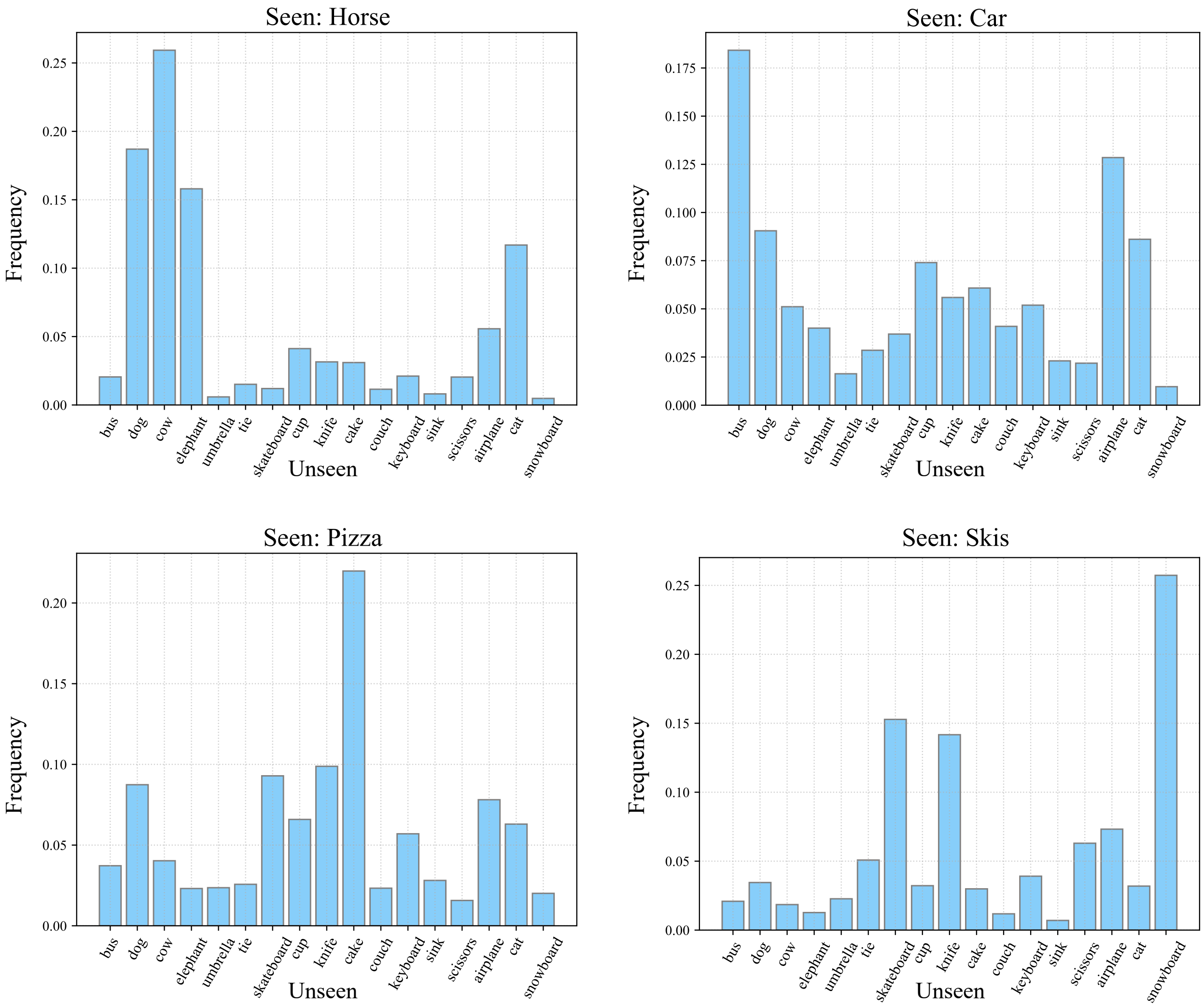

Figure II. Examples of the pseudo unseen label. For each of the four seen categories, we sample 10,000 times on pseudo unseen label generation and count the frequency of unseen categories being selected. Unseen classes that share more similar semantic features with seen classes have a higher probability of being selected as pseudo unseen labels.

E. More GZSI Visualization

We provide more qualitative examples of the proposed **D^{2}Zero** in Fig. IV and Fig. V under 65/15 split on MS-COCO. We can see that ZSI [2] suffers from bias issue and background ambiguation issue, for example in Fig. IV, the unseen class `train` is incorrectly labeled as a seen class `bus` (bias issue) and the unseen class `parking meter` is incorrectly treated as background (background ambiguation issue). In contrast, our method can predict satisfactory results for both seen and unseen instances and outperform ZSI [2] owing to our semantic-promoted debiasing and background disambiguation.

F. More Implementation Details

We implement data augmentation during training by randomly horizontal flipping the training images, resizing the images with 0.1-2.0 scales and then cropping/padding to size of 960 × 960.

As presented in the main paper, we generate our background mask from the foreground masks. Specifically, according to the foreground masks of each Transformer decoder layer, we generate the corresponding background mask for this layer, where the foreground masks are the binarized outputs (thresholded at 0.5) of the resized mask predictions of the current Transformer decoder layer. It is worth noting that our background mask visualizations are from the last layer of Transformer.

References

[1] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *ICLR*, 2017. 1

[2] Ye Zheng, Jiahong Wu, Yongqiang Qin, Faen Zhang, and Li Cui. Zero-shot instance segmentation. In *CVPR*, 2021. 2, 4, 5

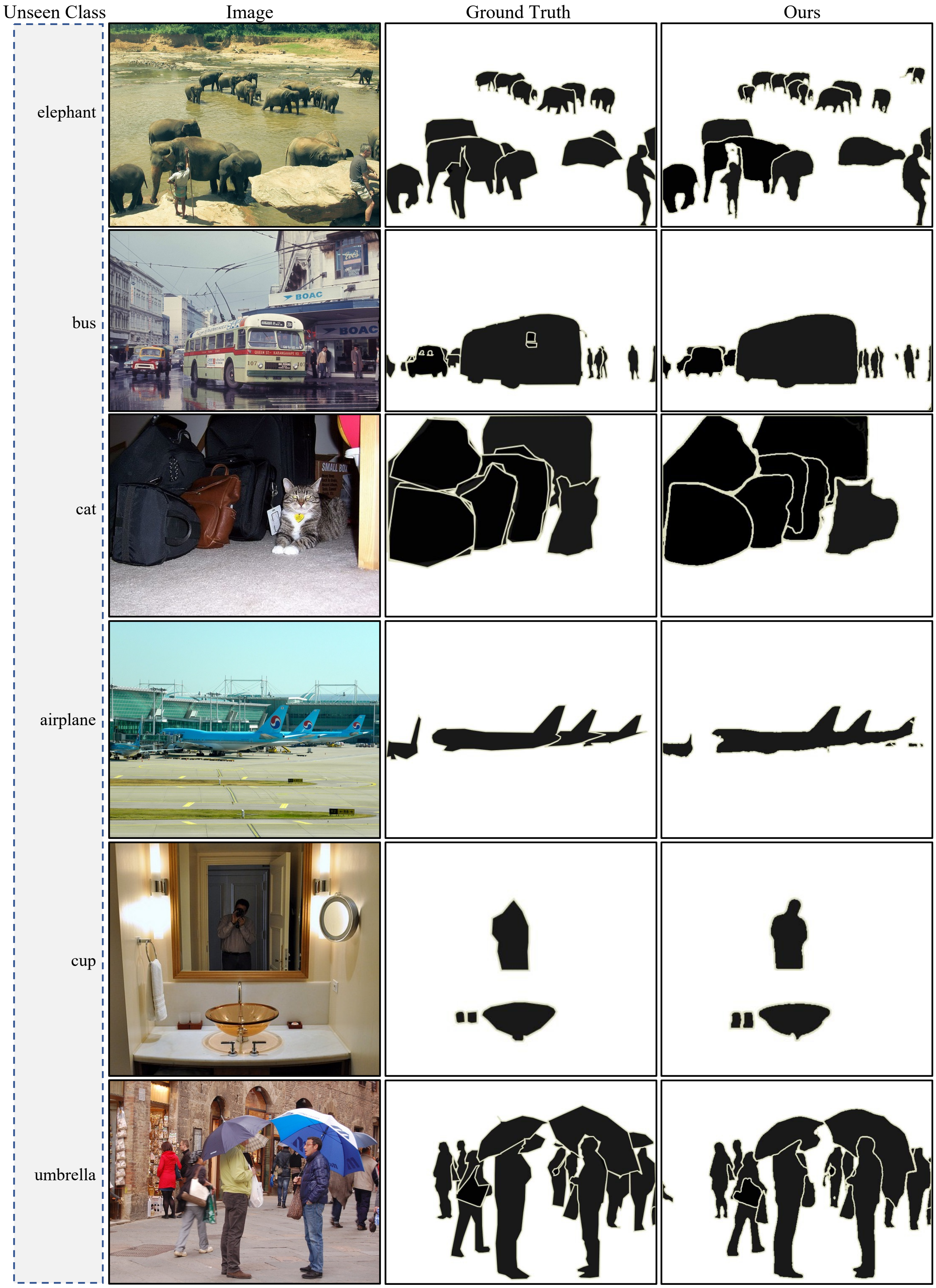

Figure III. (Best viewed in color) Qualitative background results on COCO val set under 48/17 split. Our generated background masks do a good job of excluding unseen classes (*e.g.*, `elephant` and `bus`) from background.

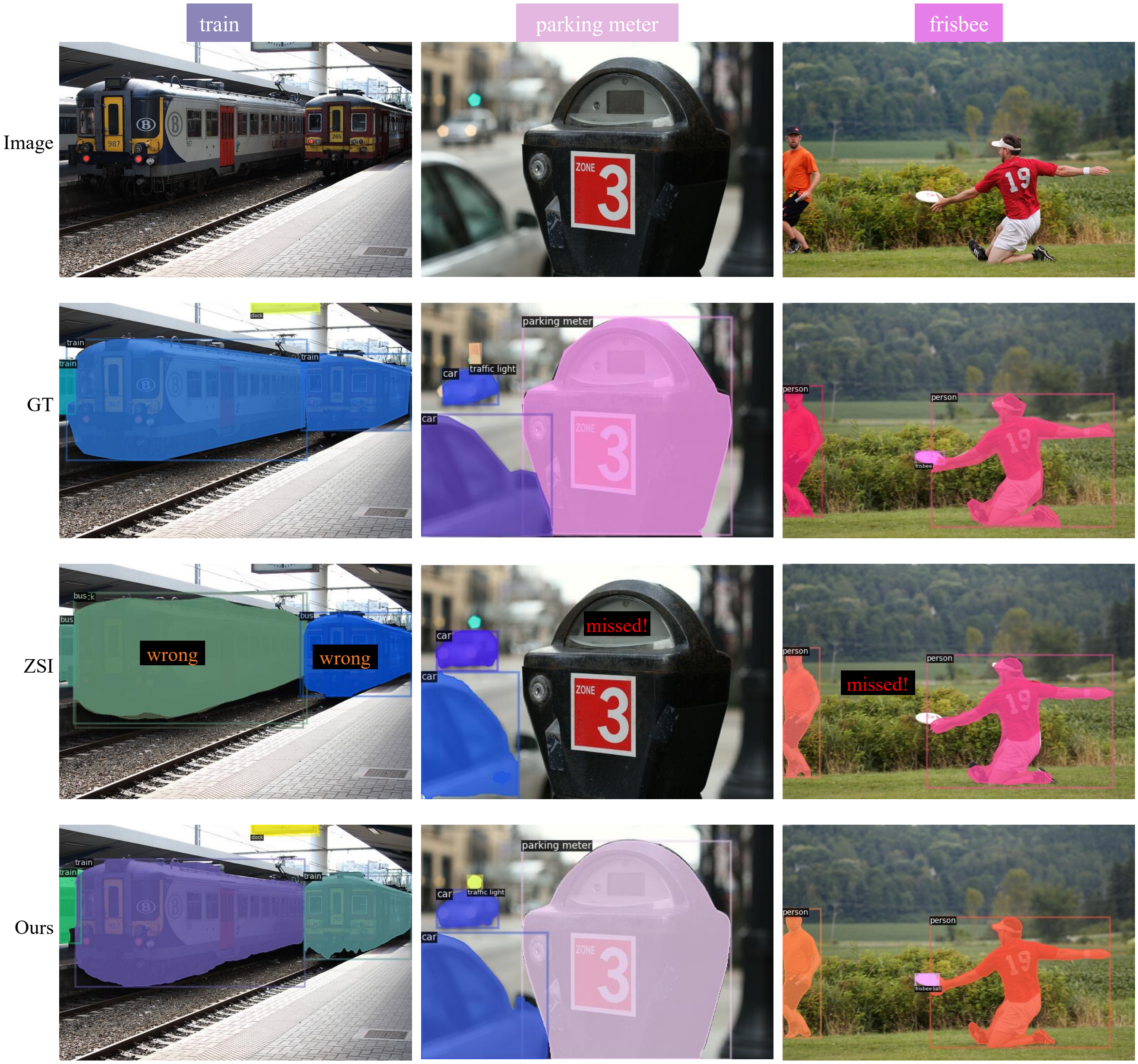

Figure IV. (Best viewed in color with zooming-in) Qualitative zero-shot instance segmentation results under 65/15 split. The four rows are image, ground truth, predictions by ZSI [2], and predictions by our **D^{2}Zero**, respectively. "wrong" denotes that the novel object is detected but incorrectly classified to a wrong label, which is a bias issue. "missed" denotes that the novel object is not detected but incorrectly treated as background, which is a background ambiguation issue. Our **D^{2}Zero** is capable of mitigating these issues.

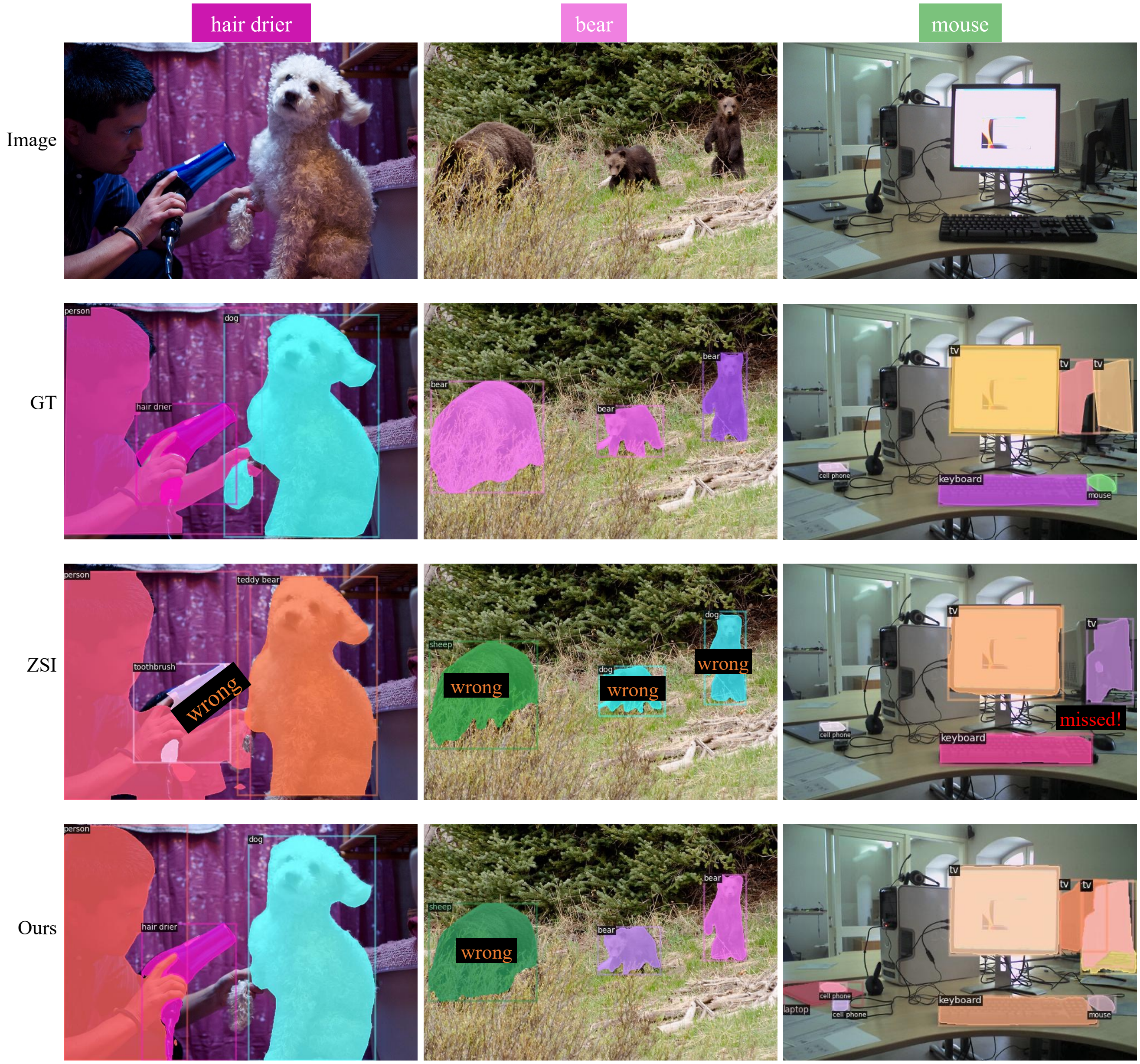

Figure V. (Best viewed in color with zooming-in) Qualitative zero-shot instance segmentation results under 65/15 split. The four rows are image, ground truth, predictions by ZSI [2], and predictions by our **D^{2}Zero**, respectively. “wrong” denotes that the novel object is detected but incorrectly classified to a wrong label, which is a bias issue. “missed” denotes that the novel object is not detected but incorrectly treated as background, which is a background ambiguation issue. Our **D^{2}Zero** is capable of mitigating these issues.