# Learning to Detect and Segment for Open Vocabulary Object Detection —Supplementary

Tao Wang
Sichuan University
twangnh@gmail.com

## Example Aggregation Weights

To analyze how CondHead learns to consolidate the class-wise knowledge into the expert prediction heads, we plot the aggregation weights for the dynamically aggregated head on some example object categories. As shown in Figure 1, we observe evident clustering of the weight distribution on object categories with close semantic meaning. For example, the aggregation weights on *horned cow*, *shepherd dog* and *black sheep* mainly attend to the first 12 expert heads. This is likely because these are all animals and with similar body architecture and pose. Similarly, we observe the *school bus*, *cabin car* and *tow truck* attend mainly to the 8th to 22th expert heads (Figure 1 (b)). the *thermos bottle*, *wineglass* and *beer can* attend mainly to the 18th to 32th expert heads (Figure 1 (c)). On the other hand, the detailed weight distribution differ for these highly attended expert heads, this may attribute to the different appearance of these categories. It seems CondHead learns to cope with the difference with compositional knowledge from multiple expert heads.

## More Implementation Details

**Architectural Illustration and Training Details** The proposed CondHead does not involve complex training strategies, it is simply trained as a straightforward replacement of standard box regression and mask segmentation heads. Fig. 2 gives an architectural illustration. Concretely, OVR-CNN and RegionCLIP first pre-train the visual-semantic representation and then train open-vocabulary detection by initializing with the learned representation. CondHead is employed in the second-stage training by replacing the original box/mask heads. As for ViLD, CondHead is simply trained together with its text-embedding transfer (ViLD-text) and image embedding distillation (ViLD-image), by replacing the original box/mask heads. The text embedding from the CLIP language encoder is used as the semantic embedding for RegionCLIP and ViLD, and the BERT language embedding is used as the semantic embedding for OVR-CNN. Other hyper-parameters such as training schedules follow these baseline methods.

**Temperature Annealing** As discussed in Section 3.2 of main paper, we optimize the expert heads with a temperature annealing strategy, *i.e.*, applying large temperature during the early training epochs and gradually annealing the temperature to a small value to ensure good dynamics of Softmax output. Concretely, we set the temperature to 20.0 and linearly decay it to 1.0 during the first 5k iterations.

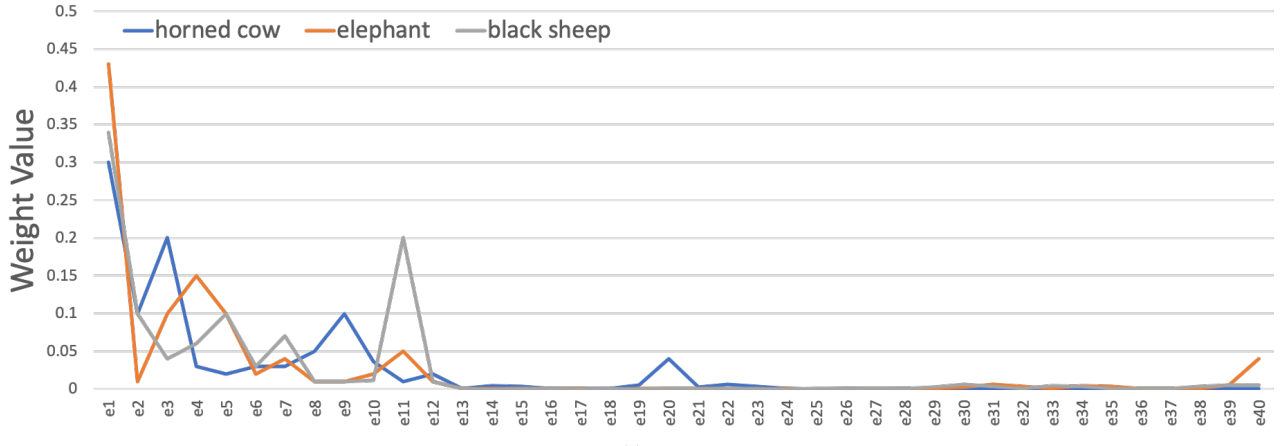
**Integrating Shapemask into CondHead** Shapemask [2] introduces shape prior and multi-stage refinement to achieve strong class-agnostic instance segmentation, which is validated on the partially supervised instance segmentation task [1]. This prior-based design can be integrated into CondHead to further improve its segmentation quality on open vocabulary objects.

As shown in Figure 3, we introduce the semantic conditioning on each stage of Shapemask [2], *i.e.*, shape estimation, coarse mask prediction and shape refinement. Concretely, the shape distribution weights are dynamically generated based on the semantic embedding, the weights are used to average the shape priors to obtain the segmentation prior. Then the two consecutive convolution kernels within the coarse mask prediction module are conditionally parameterized. This is conducted with a set of dynamically combined expert convolution kernels, same as introduced in Section 3.2 of main paper (dynamically aggregated head). Based on the coarse mask prediction, the three consecutive convolution kernels within the shape refinement module are similarly parametrized and utilized to obtain the refined mask segmentation result.

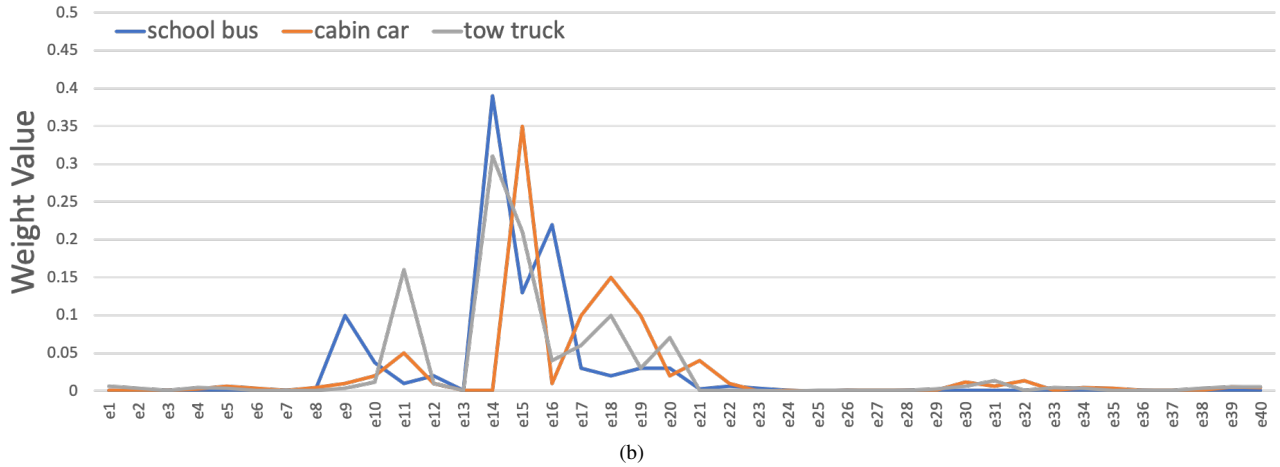
The above are used to instantiate the dynamically aggregated head stream of CondHead (*i.e.*, $\hat{\mathcal{B}}$), the original dynamically generated head stream ($\dot{\mathcal{B}}$) is maintained as introduced in Sec. 3.2.

## The Effect of Architectural Instantiation

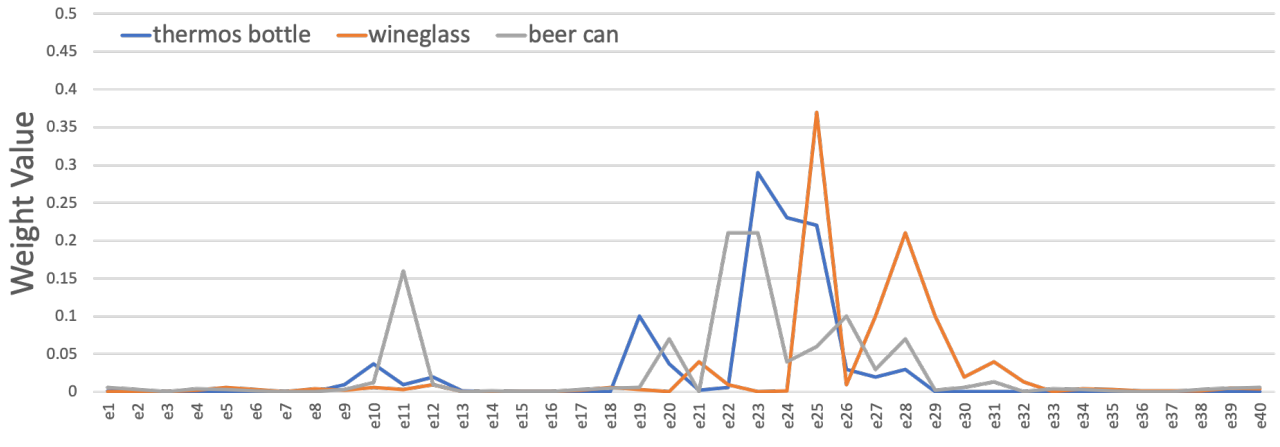
We exam the effect of architectural instantiation on CondHead, specifically with the depth and hidden dimension of the networks.




(a)




(b)




(c)

Figure 1. Example aggregation weight distribution. Dynamic aggregation weight on some example object categories of LVIS. The horizontal axis corresponds to the index of expert heads. The vertical axis corresponds to the normalized weight value. The weights are from a CondHead model based on RegionCLIP. The weight indexes are permuted to better show trends of distribution.

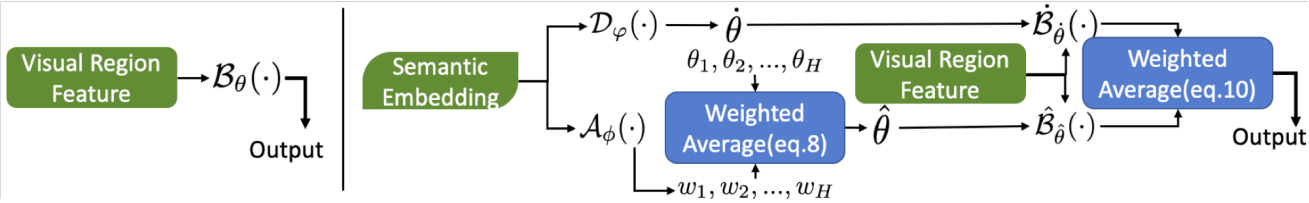



Figure 2. **Left**: illustration of standard box regression head, the learnable parameter is $\theta$. **Right**: illustration of CondHead architecture, the learnable parameters are $\theta_1, \theta_2, ..., \theta_H, \phi$, and $\varphi$. While box regression ($\mathcal{B}$) is illustrated here, the mask segmentation ($\mathcal{M}$) is similar.

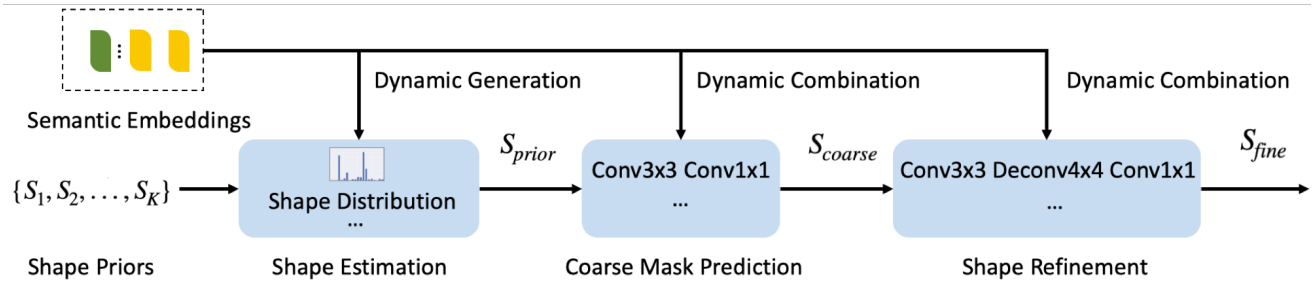



Figure 3. Integrating Shapemask into CondHead. We omit the architecture design and only depicts the parametric components that are affected by the proposed CondHead. Dynamic Generation: the shape distribution weights are directly generated on the semantic embedding. Dynamic Combination: the convolution kernels are generated with the proposed dynamically aggregated expert heads.

| | $\mathcal{A}$ | | | | | | $\mathcal{D}$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | depth | | | hidden dimension | | | depth | | | hidden dimension | | |
| | 1 | 2 | 3 | 128 | 256 | 384 | 1 | 2 | 3 | 128 | 256 | 384 |
| $AP_r^b$ | 19.0 | 19.9 | **20.1** | 19.3 | **19.9** | 19.7 | 19.0 | **19.9** | **19.9** | 19.5 | **19.9** | 19.8 |
| $AP_r^m$ | 18.7 | **20.0** | **20.0** | 18.9 | **20.0** | 19.7 | 19.4 | 20.0 | **20.1** | 18.9 | **20.0** | 19.6 |

Table 1. The effect of architectural setting, with dynamic aggregation weight generator ($\mathcal{A}$) and dynamic parameter generator ($\mathcal{D}$). The hidden dimension is fixed at 256 when evaluating the effect of depth. The depth is fixed at 2 when evaluating the effect of hidden dimension. The experiments are conducted on LVIS, with RegionCLIP model based on ResNet-50 backbone.

| | $\mathcal{B}_h$ | | | | | | $\mathcal{M}_h$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | depth | | | hidden dimension | | | depth | | | hidden dimension | | |
| | 1 | 2 | 3 | 128 | 256 | 384 | 2 | 3 | 4 | 128 | 256 | 384 |
| $AP_r^b$ | 18.4 | **19.6** | 19.5 | 18.3 | 19.6 | **19.8** | - | - | - | - | - | - |
| $AP_r^m$ | - | - | - | - | - | - | 19.4 | **20.0** | **20.0** | 19.6 | **20.0** | 19.9 |

Table 2. The effect of architectural setting, with the expert box regression heads $\{\mathcal{B}_h\}$ and mask segmentation heads $\{\mathcal{M}_h\}$. The hidden dimension is fixed at 256 when evaluating the effect of depth. The depth is fixed at 2 when evaluating the effect of hidden dimension. The box heads are instantiated as fully connected networks, and the mask heads are instantiated as convolution networks. The experiments are conducted on LVIS, with RegionCLIP model based on ResNet-50 backbone.

- Dynamic aggregation weight generator. As shown in Table 1, the performance improvement is significant when increasing the depth from 1 to 2, while diminishes beyond that, *e.g.*, 0.9 box AP improvement for depth of 1 to 2 and 0.2 box AP improvement for depth of 2 to 3, with dynamic weight generator $\mathcal{A}$. Similar observation holds for the hidden dimension. We thus set the depth and hidden dimension to 2 and 256 for the dynamic weight generator networks.

- Expert box regression heads and mask segmentation heads. As shown in Table 2, for the set of expert heads, the performance improvement is significant when increasing the depth from 1 to 2, while diminishes beyond that, *e.g.*, box AP of 18.4 to 19.6 for depth of 1 to 2 and 19.6 to 19.5 for depth of 2 to 3, Similar observation holds for the hidden dimension. We thus set the depth and hidden dimension to 2 and 256 for the expert box heads and mask heads.

| | $\dot{\mathcal{B}}$ | | | $\dot{\mathcal{M}}$ | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 1 | 2 | 3 |
| $AP_r^b$ | 19.9 | **20.0** | 19.6 | - | - | - |
| $AP_r^m$ | - | - | - | **20.0** | 19.7 | 19.6 |

Table 3. The effect of architectural setting, with the dynamic generated box regression head $\dot{\mathcal{B}}$, and mask segmentation head $\dot{\mathcal{M}}$. The hidden dimension is fixed at 4 due to the limited output dimension for direct parameter generation.The box heads are instantiated as fully connected networks, and the mask heads are instantiated as convolution networks. The experiments are conducted on LVIS, with RegionCLIP model based on ResNet-50 backbone.

| | Object Detection | | | | Instance Segmentation | | | |
|---|---|---|---|---|---|---|---|---|
| - | Novel | Base | All | - | Novel | Base | All | - |
| B | 31.3 | 56.5 | 50.4 | - | 27.5 | 54.1 | 48.1 | - |
| Cd | 33.7 | 58.0 | 52.2 | - | 29.7 | 55.8 | 49.5 | - |
| Cdv | 31.9 | 56.6 | 50.7 | - | 28.4 | 54.7 | 48.5 | - |
| Cde | 31.8 | 56.8 | 50.9 | - | 28.2 | 54.6 | 48.4 | - |
| - | $AP_r$ | $AP_c$ | $AP_f$ | AP | $AP_r$ | $AP_c$ | $AP_f$ | AP |
| B | 22.1 | 31.8 | 37.0 | 32.4 | 21.8 | 30.2 | 35.1 | 30.2 |
| Cd | 25.1 | 33.4 | 37.8 | 33.9 | 24.4 | 31.6 | 35.9 | 31.6 |
| Cdv | 22.5 | 32.1 | 36.9 | 32.5 | 22.1 | 30.6 | 35.3 | 30.5 |
| Cde | 22.9 | 32.5 | 37.2 | 32.8 | 22.5 | 30.6 | 35.4 | 30.8 |

Table 4. B, Cd denote baseline and CondHead, Cdv and Cde denote dynamic condition on visual region feature and ensembling/averaging multiple independently trained heads. Results are obtained with RegionCLIP (ResNet50), on COCO and LVIS.

- Dynamically generated heads. We exam the depth of dynamically generated box and mask heads. Due to the limitation of network output dimension, we set the hidden dimension to 4. As shown in Table 3, increasing the depth actually brings limited benefits, *e.g.*, depth of 1 already achieves 19.9 box AP while depth of 2 and 3 obtain 20.0 and 19.6 box AP. Similar trend is observed with the mask head. We thus employ simple 1-layer network for the dynamic heads.

Based on all experiment results above, we set the depth and hidden dimension as shown in the main paper Table 1.

## Ablation on Dynamic Designs

We also validate the effectiveness of the proposed dynamic conditioning design by examining two other alternative designs: replacing the proposed dynamically combined heads with two simple ensembling baselines (dynamic conditioning on the visual region feature and simply ensembling multiple independently trained heads during inference). As shown in Tab. 4, the performance drops compared to CondHead, especially on novel categories, meaning the proposed method helps better learn class-specific prediction for novel categories.

## References


[1] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4233–4241, 2018. 1

[2] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9207–9216, 2019. 1