ZSI: Zero-shot Instance Segmentation

CVPR2021

211105 한지수

Zero-shot Instance segmentation

- Task: Zero-shot Instance segmentation
    - Train & Test
        - Training: seen data
        - Testing: segment seen and unseen instances
- Problem Formulation
    - Given
        - Training set $D_{train} = (x_s, w_s)$ (x_s: image, w_s: seen classes word-vectors)
        - Test set $D_{test} = (x, w)$ (w: seen & unseen classes word-vectors)
    - Train: $\theta = argmax_\theta \sum_{i=1}^{D_{train}} log(p(y_{si} \in C_s | x_{si}, w_s, \theta)$
    - Inference, Test: $argmax_\theta \sum_{i=1}^{D_{test}} log(p(y_{si} \in C_s, y_{ui} \in C_u | x_i, w, \theta)$

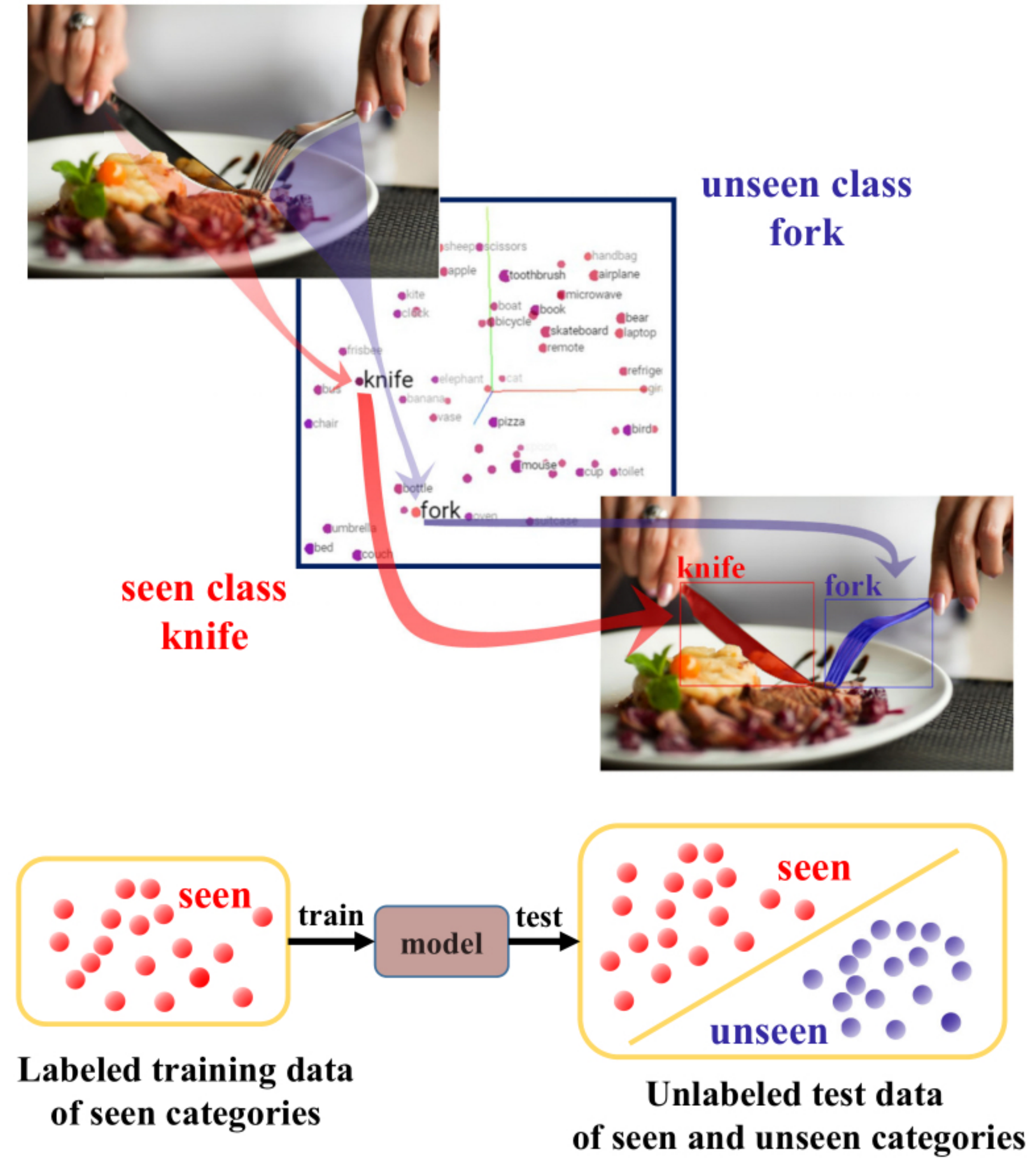


Figure 1. In zero-shot instance segmentation, we can only use the labeled data of seen categories for training but predict the instance segmentation results for both seen and unseen categories. In our method, we use the seen classes data, *e.g.*, "knife" to establish the mapping relationship between visual and semantic concepts during training and then transfer it to segment unseen instances, *e.g.*,"fork" in inference.

Challenges on Zero-shot Instance segmentation

- How to do instance segmentation for unseen classes

  ➡ Extra semantic knowledge contained in pre-trained word-vectors to correlate the seen and unseen classes

  ➡ **From the semantic word vector and image data of seen classes to establish the visual semantic mapping relationship in a detection-segmentation manner and transfer it from seen to unseen classes**

  ➡ Zero-shot detector and Semantic Mask Head

- How to reduce the confusion btw background and unseen classes
  - Background class drawbacks

    1. The existing semantic representation of background is unreasonable
    2. The existing semantic representation of the background class is fixed

  ➡ Background Aware RPN (BA-RPN) and Synchronized Background Strategy (Sync-bg)

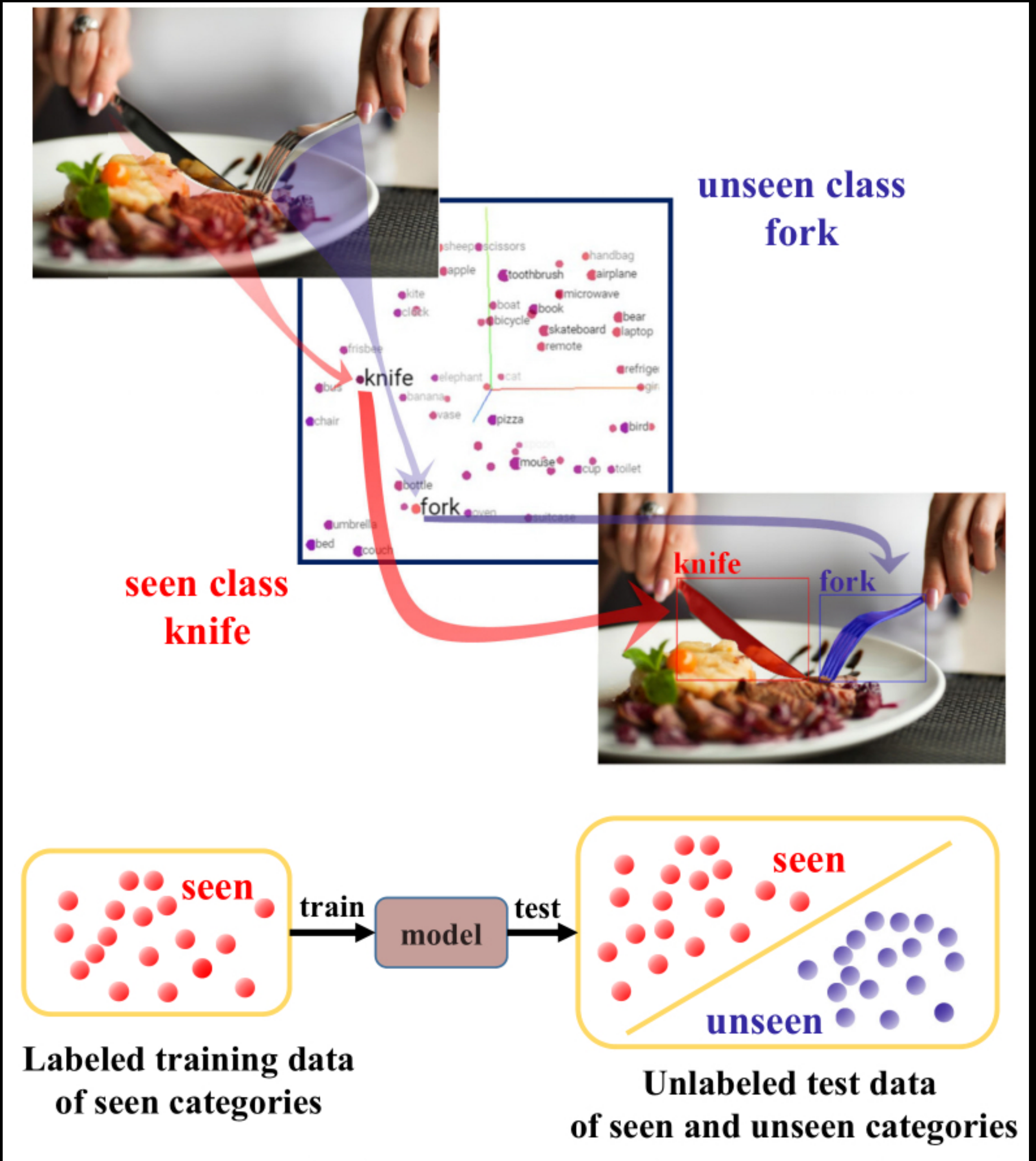


Figure 1. In zero-shot instance segmentation, we can only use the labeled data of seen categories for training but predict the instance segmentation results for both seen and unseen categories. In our method, we use the seen classes data, *e.g.*, "knife" to establish the mapping relationship between visual and semantic concepts during training and then transfer it to segment unseen instances, *e.g.*,"fork" in inference.

Zero-shot Instance segmentation

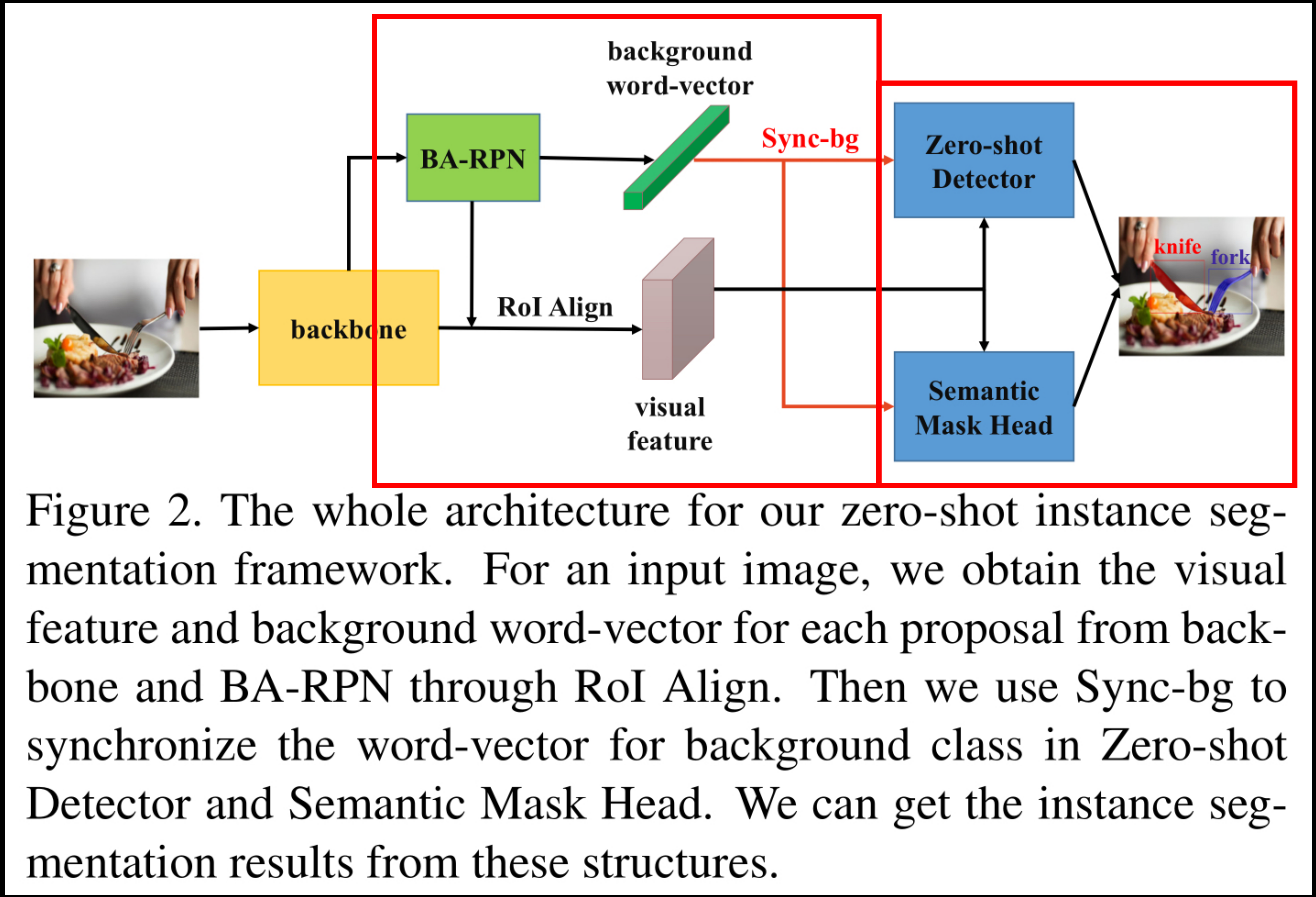


Figure 2. The whole architecture for our zero-shot instance segmentation framework. For an input image, we obtain the visual feature and background word-vector for each proposal from backbone and BA-RPN through RoI Align. Then we use Sync-bg to synchronize the word-vector for background class in Zero-shot Detector and Semantic Mask Head. We can get the instance segmentation results from these structures.

- End-to-end network that adopts the semantic word-vector to detect and segment unseen instances
- Structure
    1. Zero-Shot Detector
    2. Semantic Mask Head (SMH) > enable the instance segmentation for unseen classes by learning visual-semantic relationship with an encoder-decoder structure
    3. Background Aware RPN(BA-RPN) and Synchronized Background > learn a dynamically adaptive word-vector for background class

1. Zero-shot Detector & 2. Semantic Mask Head

A. Learning the relationship btw visual and semantic concepts from seen classes data

B. transferring it to detect unseen objects

1. Zero-shot Detector
   - Faster R-CNN with the visual-semantic alignment (for classification)

2. Semantic Mask Head: Encoder-Decoder Module
   - for a more discriminative visual-semantic alignment
   - Encoder T_e: encode the visual feature for the input RoI → semantic feature (segmentation result from semantic word-vectors)
   - Decoder T_d: decode the semantic feature → visual feature
     - during training, Removed on inference

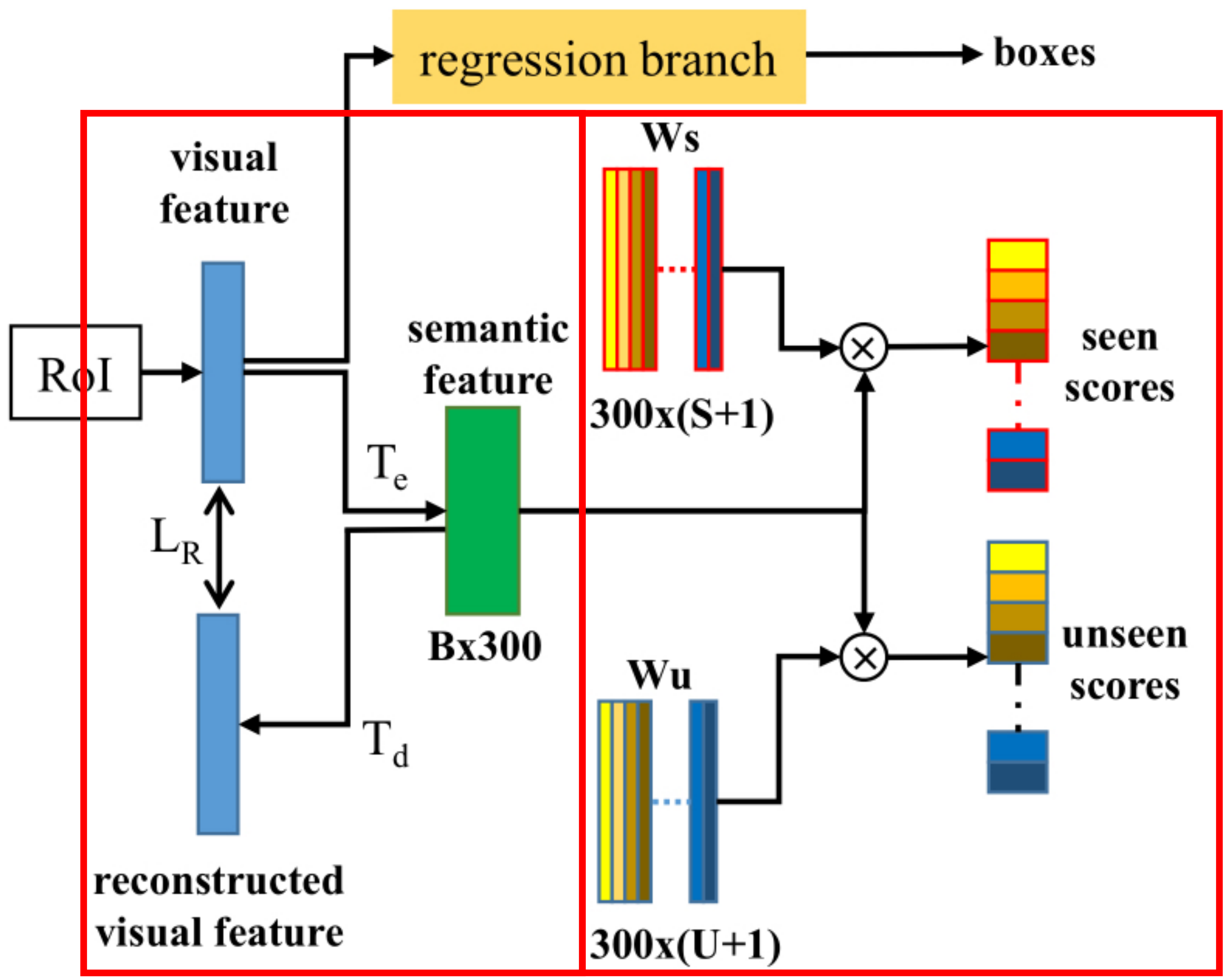


Figure 3. The details for zero-shot detector. It is trained in an encoder-decoder manner and we only use the encoder T_e in testing process. W_s is the word-vectors of all seen classes and background class. W_u is the word-vectors of all unseen classes and background class. S is the number of seen classes and U is the number of unseen classes. Each class has a 300-dimensional word-vector. B is batch size.

1. Zero-shot Detector & 2. Semantic Mask Head

1. Zero-shot Detector
   - Classification Module: Calculate the similarity btw the word-vector of each element and the word-vectors of all seen and unseen classes

2. Semantic Mask Head: Encoder-Decoder Module
   - Encoder T_e, Decoder T_d
   - Reconstruction Loss function L_R
     - Minimize the difference btw reconstructed visual feature and original visual feature
     - $L_R = \sum_{i=1}^{E} (O_i - R_i)^2$ (O_i: original visual feature, R_i: reconstructed visual feature)

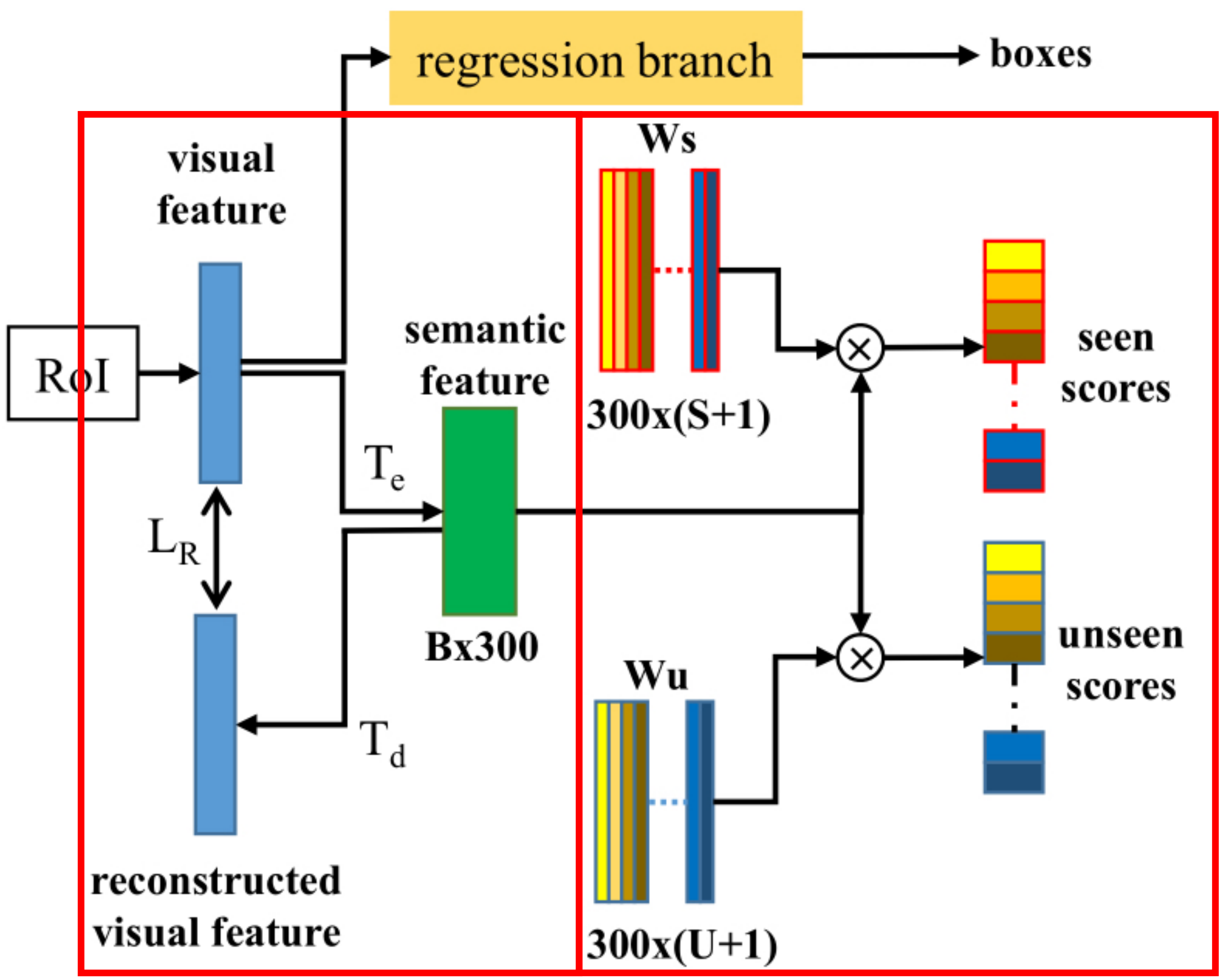


Figure 3. The details for zero-shot detector. It is trained in an encoder-decoder manner and we only use the encoder T_e in testing process. W_s is the word-vectors of all seen classes and background class. W_u is the word-vectors of all unseen classes and background class. S is the number of seen classes and U is the number of unseen classes. Each class has a 300-dimensional word-vector. B is batch size.

1. Zero-shot Detector & 2. Semantic Mask Head

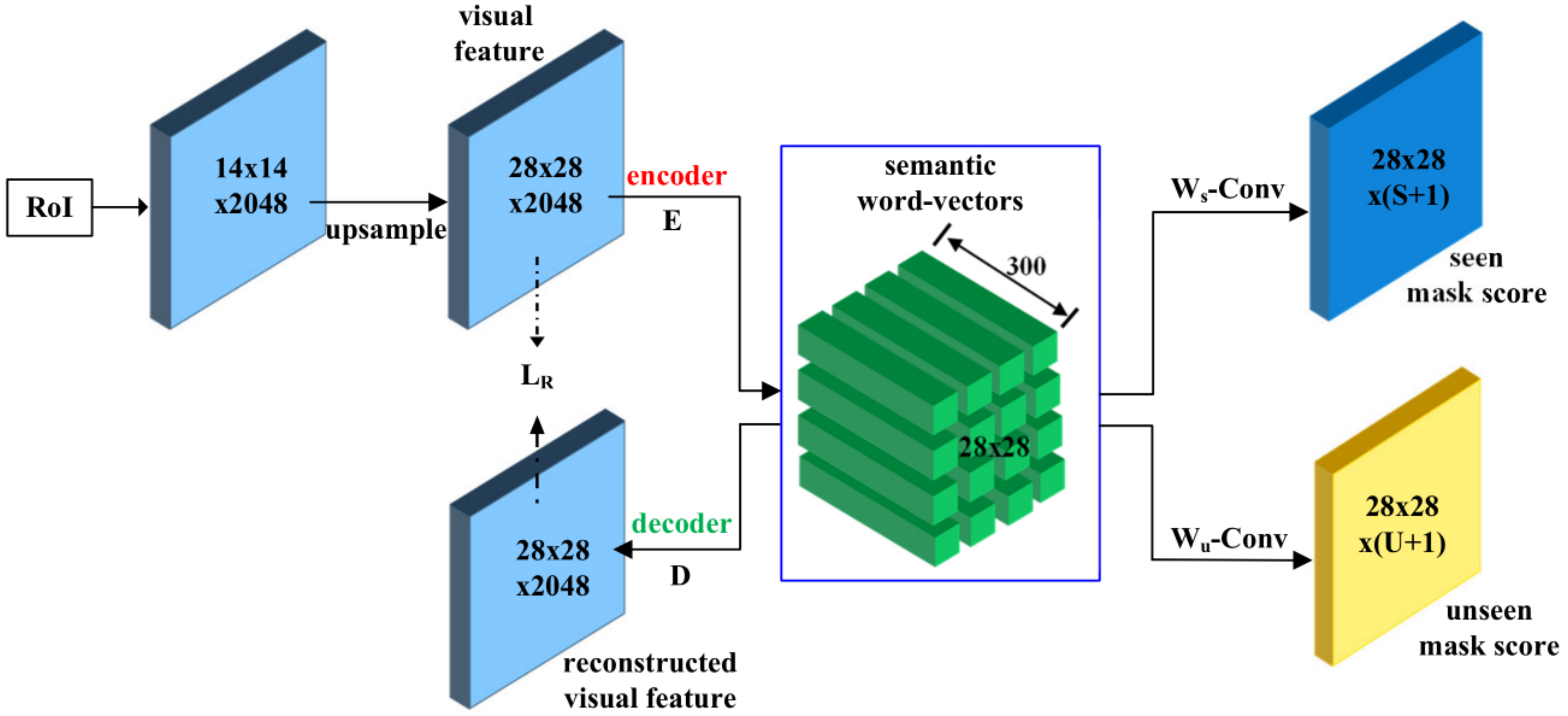


Figure 4. Our Semantic Mask Head is an encoder-decoder structure. In training, we use the encoder E to encode the visual feature into the semantic word-vectors. Then we adopt the decoder D to decode the semantic word-vectors back to reconstructed visual feature and use the loss function $\mathcal{L}_R$ to minimize the difference between the two visual features. D is be removed in inference. W_s-Conv and W_u-Conv are both fixed convolutional layers and we use them to perform pixel-by-pixel convolution on the semantic word-vectors to get the seen and unseen classes instance segmentation results.

Semantic representation : 300x28x28 Semantic feature tensor, each channel represent a dimension of the word-vector and each 300x1 is a word-vector

Calculate the similarity btw the word-vector of each element & (seen and unseen classes) => W_s, W_u mask score

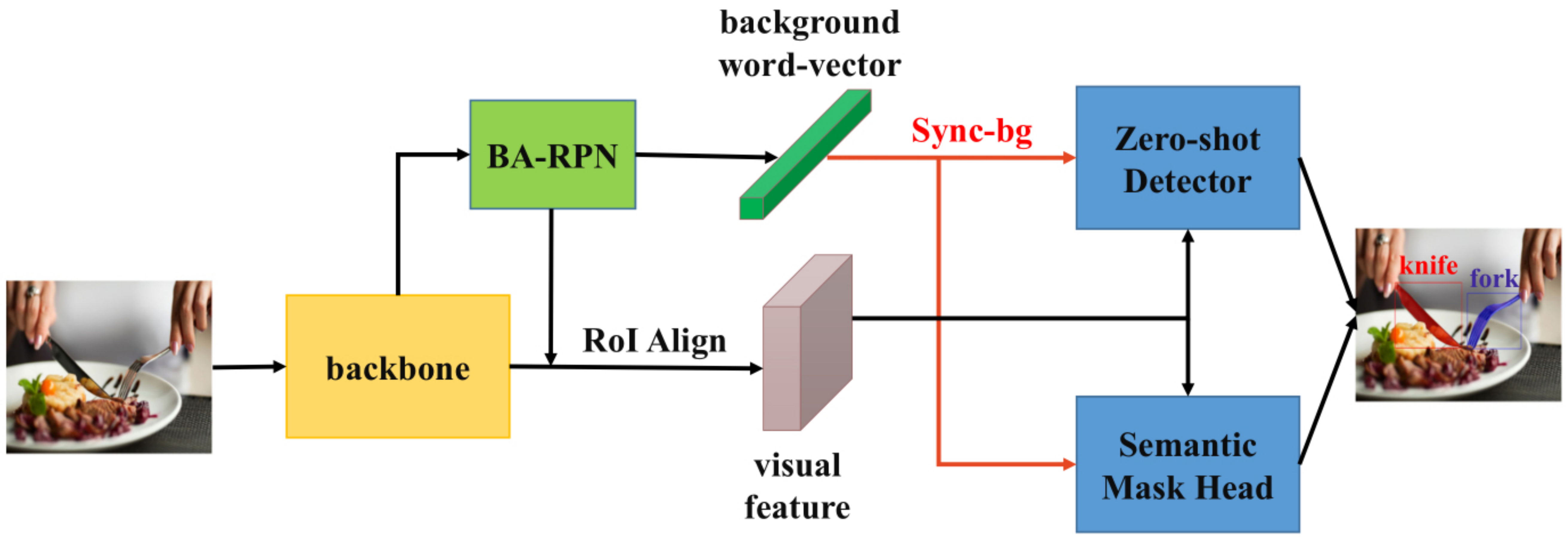
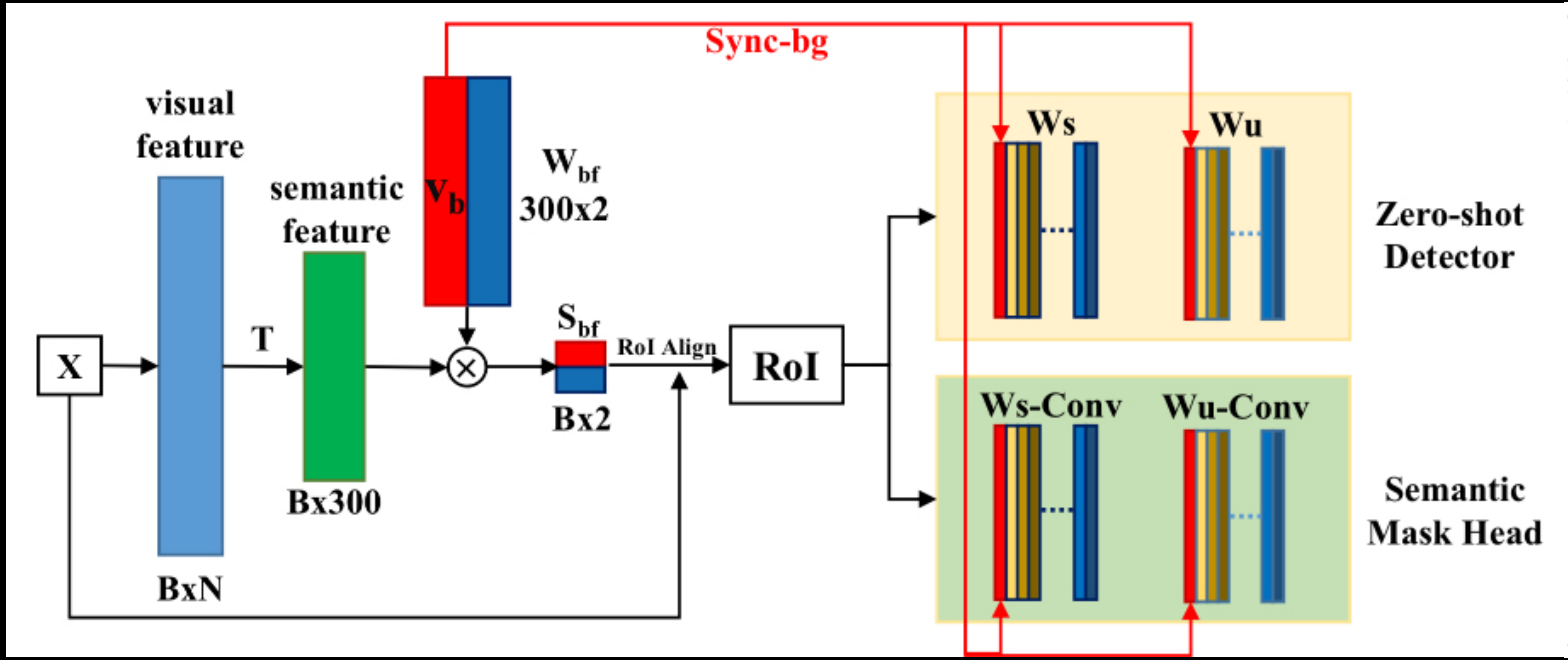
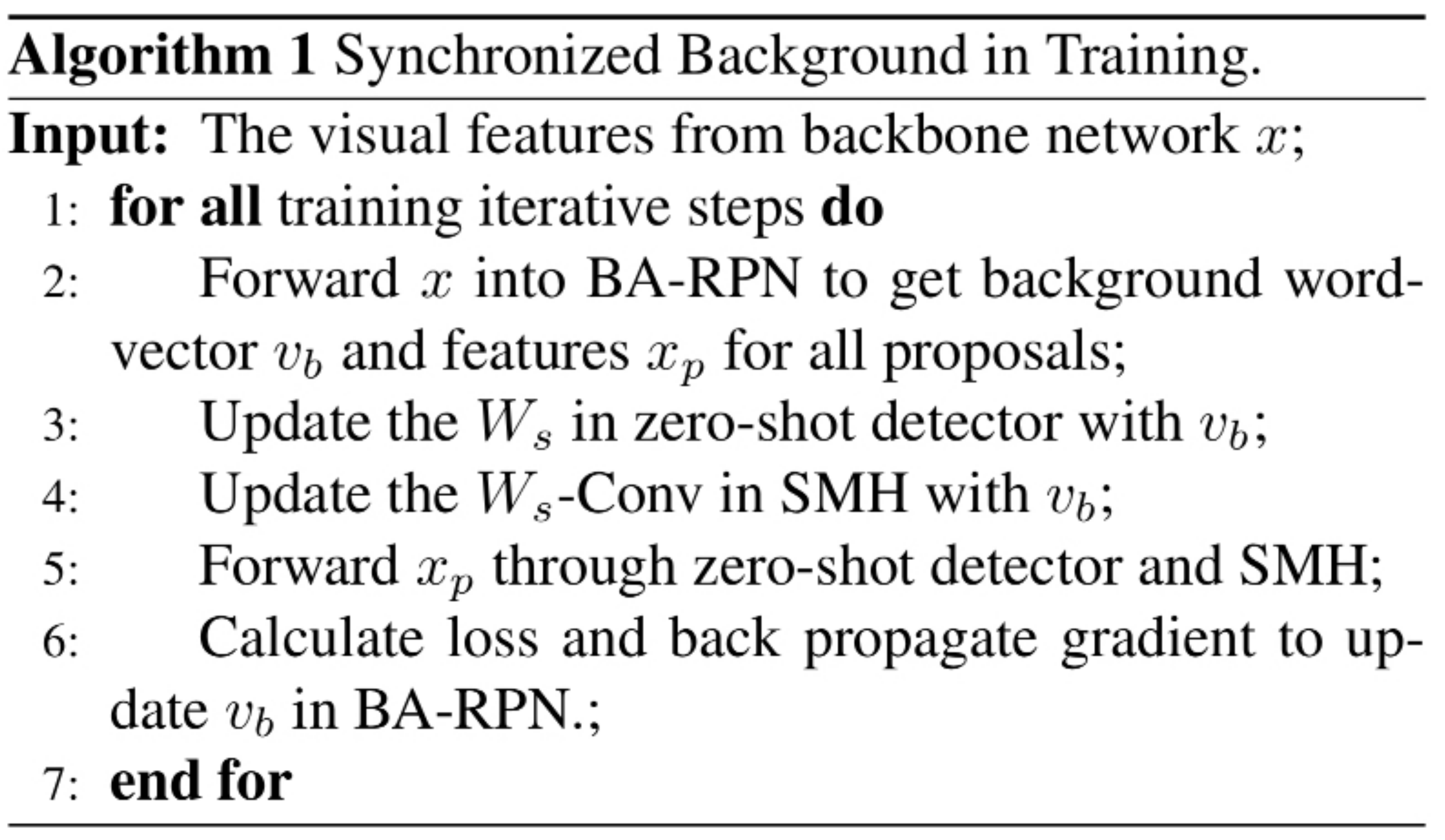


Figure 2. The whole architecture for our zero-shot instance segmentation framework. For an input image, we obtain the visual feature and background word-vector for each proposal from backbone and BA-RPN through RoI Align. Then we use Sync-bg to synchronize the word-vector for background class in Zero-shot Detector and Semantic Mask Head. We can get the instance segmentation results from these structures.

3. BA-RPN and Synchronized Background

- BA-RPN
  - Visual-semantic learning process into the original RPN to learn a more reasonable word-vector for background class from images
  - FC Layer T: input visual feature → semantic feature
  - 300x2 FC Layer W_{bf}: get background-foreground binary classification score
- Synchronized Background
  - Problem: the background class has different forms in different images, while the background word-vector learned from BA-RPN is still a fixed one





Algorithm 1 Synchronized Background in Training.

Input: The visual features from backbone network x;

```
for all training iterative steps do
    Forward $x$ into BA-RPN to get background word-vector $v_b$ and features $x_p$ for all proposals;
    Update the $W_s$ in zero-shot detector with $v_b$;
    Update the $W_s$-Conv in SMH with $v_b$;
    Forward $x_p$ through zero-shot detector and SMH;
    Calculate loss and back propagate gradient to update $v_b$ in BA-RPN.;
end for
```

Loss function & Results

- Loss Function
  - $L_{ZSI} = L_{BA} + L_{ZSD} + L_{SMH}$
  - $L_{BA} = l_1(r, \hat{r}) + CE(c, \hat{c})$
  - $L_{ZSD} = l_1(r, \hat{r}) + CE(c, \hat{c}) + \lambda_{ZSD} L_R(O, R)$
  - $L_{SMH} = BCE(c, \hat{c}) + \lambda_{SMH} L_R(O, R)$

- SOTA on Zero-shot instance segmentation

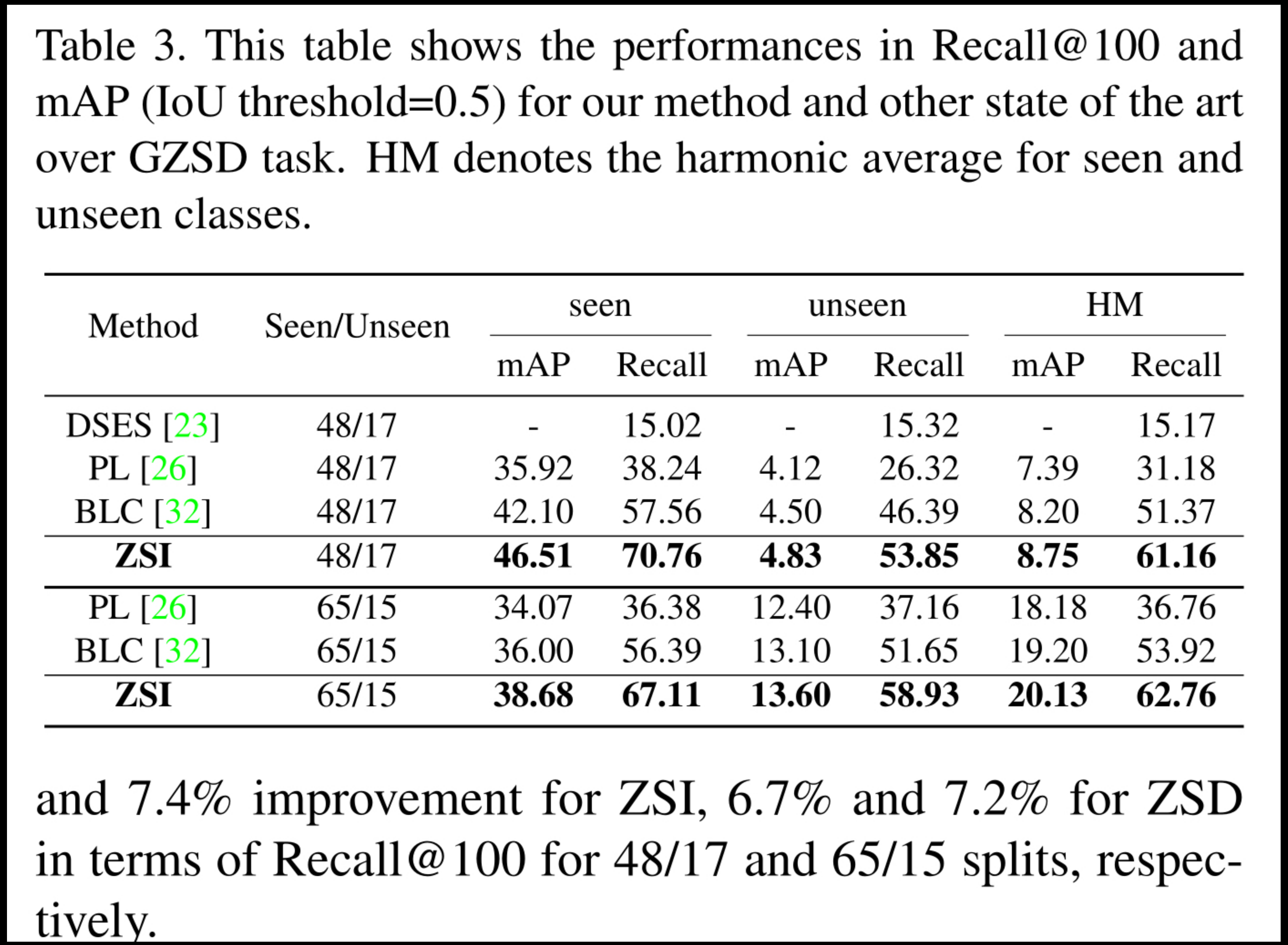


Table 3. This table shows the performances in Recall@100 and mAP (IoU threshold=0.5) for our method and other state of the art over GZSD task. HM denotes the harmonic average for seen and unseen classes.

Method	Seen/Unseen	seen		unseen		HM	
		mAP	Recall	mAP	Recall	mAP	Recall
DSES [23]	48/17	-	15.02	-	15.32	-	15.17
PL [26]	48/17	35.92	38.24	4.12	26.32	7.39	31.18
BLC [32]	48/17	42.10	57.56	4.50	46.39	8.20	51.37
ZSI	48/17	**46.51**	**70.76**	**4.83**	**53.85**	**8.75**	**61.16**
PL [26]	65/15	34.07	36.38	12.40	37.16	18.18	36.76
BLC [32]	65/15	36.00	56.39	13.10	51.65	19.20	53.92
ZSI	65/15	**38.68**	**67.11**	**13.60**	**58.93**	**20.13**	**62.76**

and 7.4% improvement for ZSI, 6.7% and 7.2% for ZSD in terms of Recall@100 for 48/17 and 65/15 splits, respectively.