Annotation Free Semantic Segmentation with Vision Foundation Models

Xinze Li

Apr. 7

Background

Keywords

- Zero-shot semantic segmentation
- Annotation Free
- Lightweight
- Foundation models

Challenges

- Generalization Across Diverse Domains
- Dependency on Pretrained Models
- Quality of Pseudo Annotations
- How to align any off-the-shelf (现成的) pretrained vision encoder with text semantics, and with no human supervision.

Novelty

- A lightweight **contrastive alignment module**, built on foundation models like **CLIP (for object detection)** and **SAM (for generating object masks)**
- Generate **semantic segmentation masks as pseudo annotation** with zero pixel or image level labels.

Angry!!!!!!!!!!!!!!!!!!!

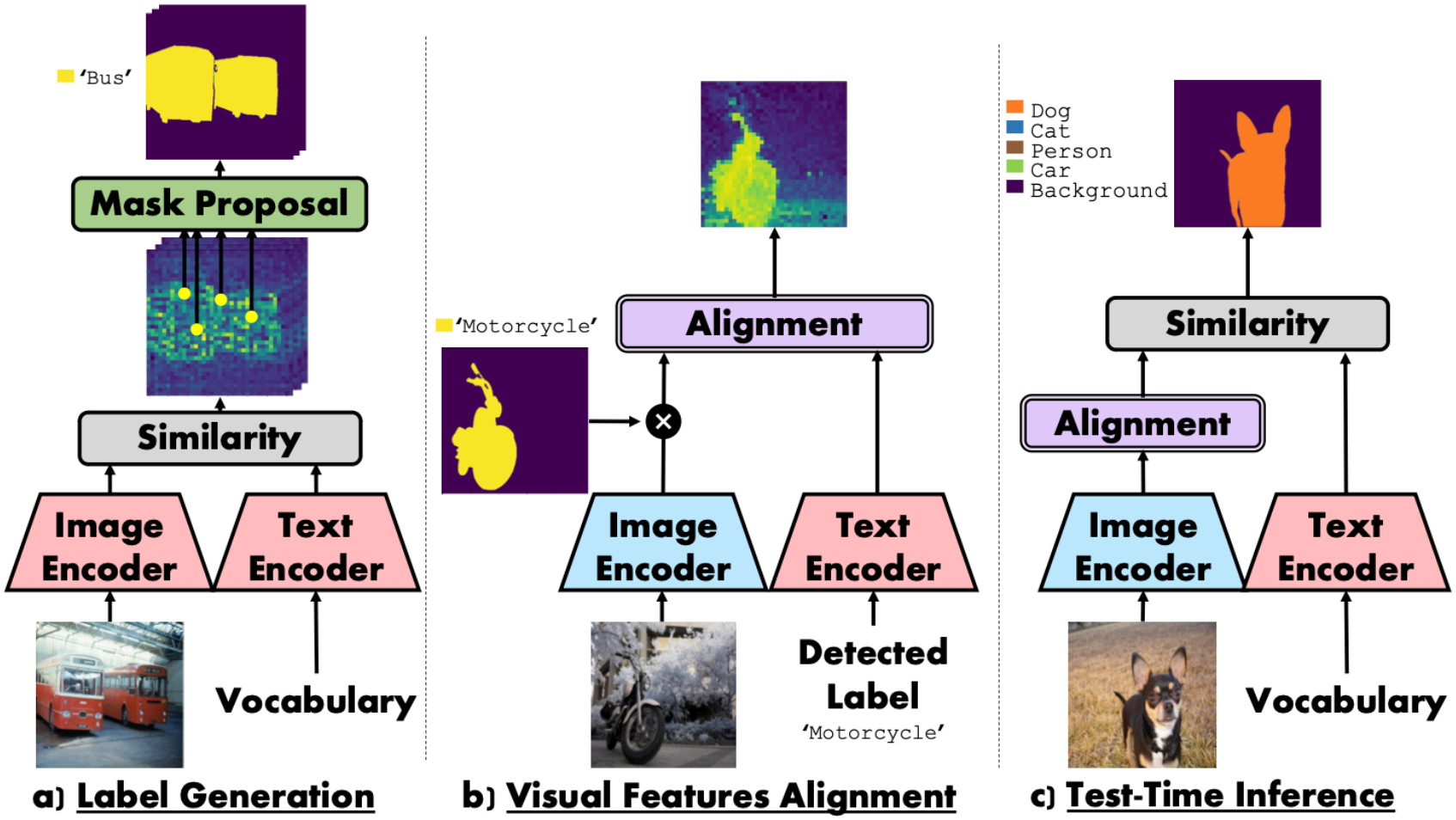
Soroush Seifi[1], Daniel Olmeda Reino[2], Fabien Despinoy[2], and Rahaf Aljundi[2]

[1] Toyota Motor Europe NV/SA associated partner by contracted services
[2] Toyota Motor Europe, Hoge Wei 33, B-1930 Zaventem, Belgium

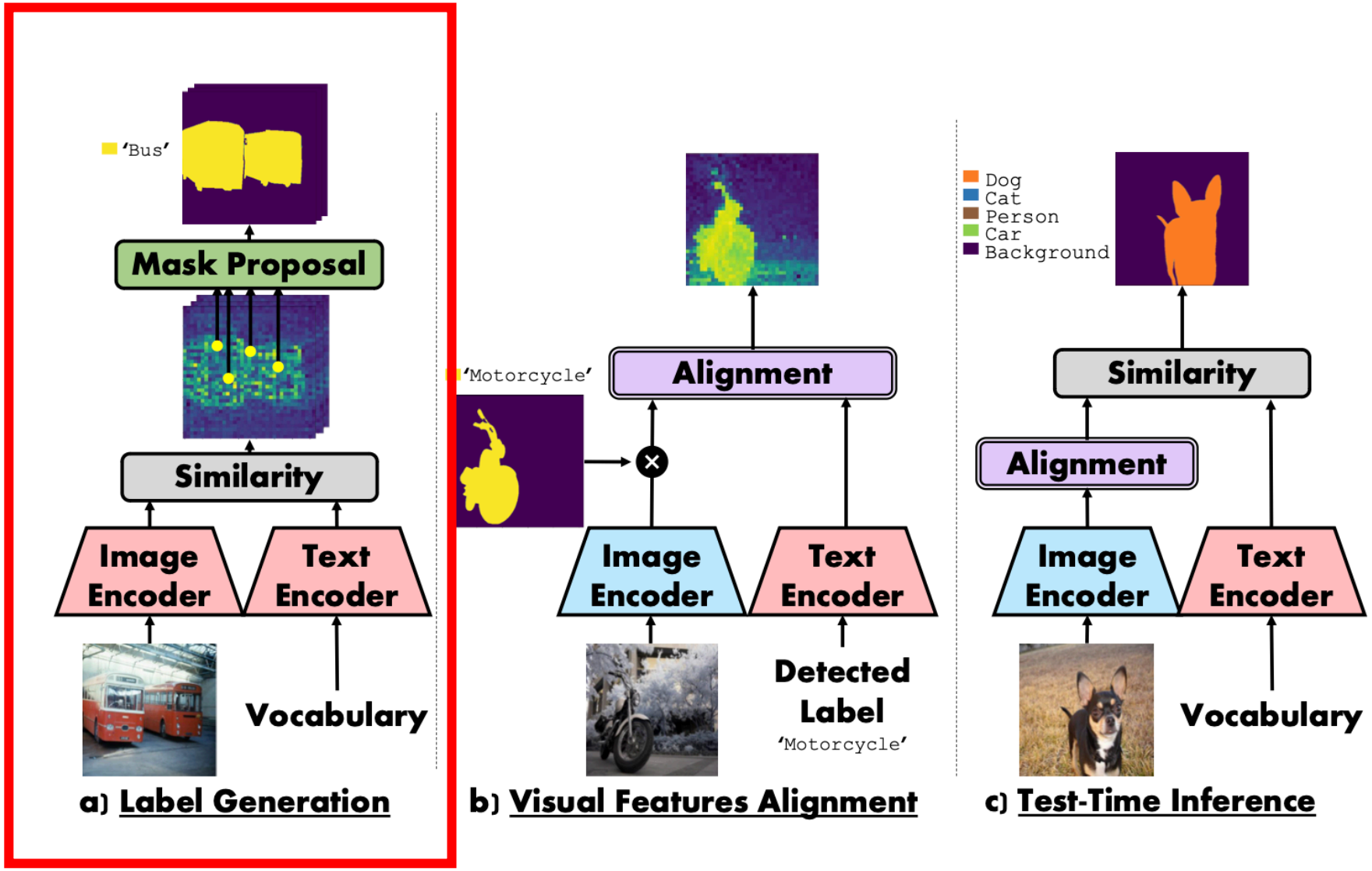
No open source code!!!

Method

Overview


'Bus'
Mask Proposal
Similarity
Image Encoder
Text Encoder
Vocabulary
a) Label Generation
'Motorcycle'
Alignment
Image Encoder
Text Encoder
Detected Label
'Motorcycle'
b) Visual Features Alignment
Dog
Cat
Person
Car
Background
Similarity
Alignment
Image Encoder
Text Encoder
Vocabulary
c) Test-Time Inference

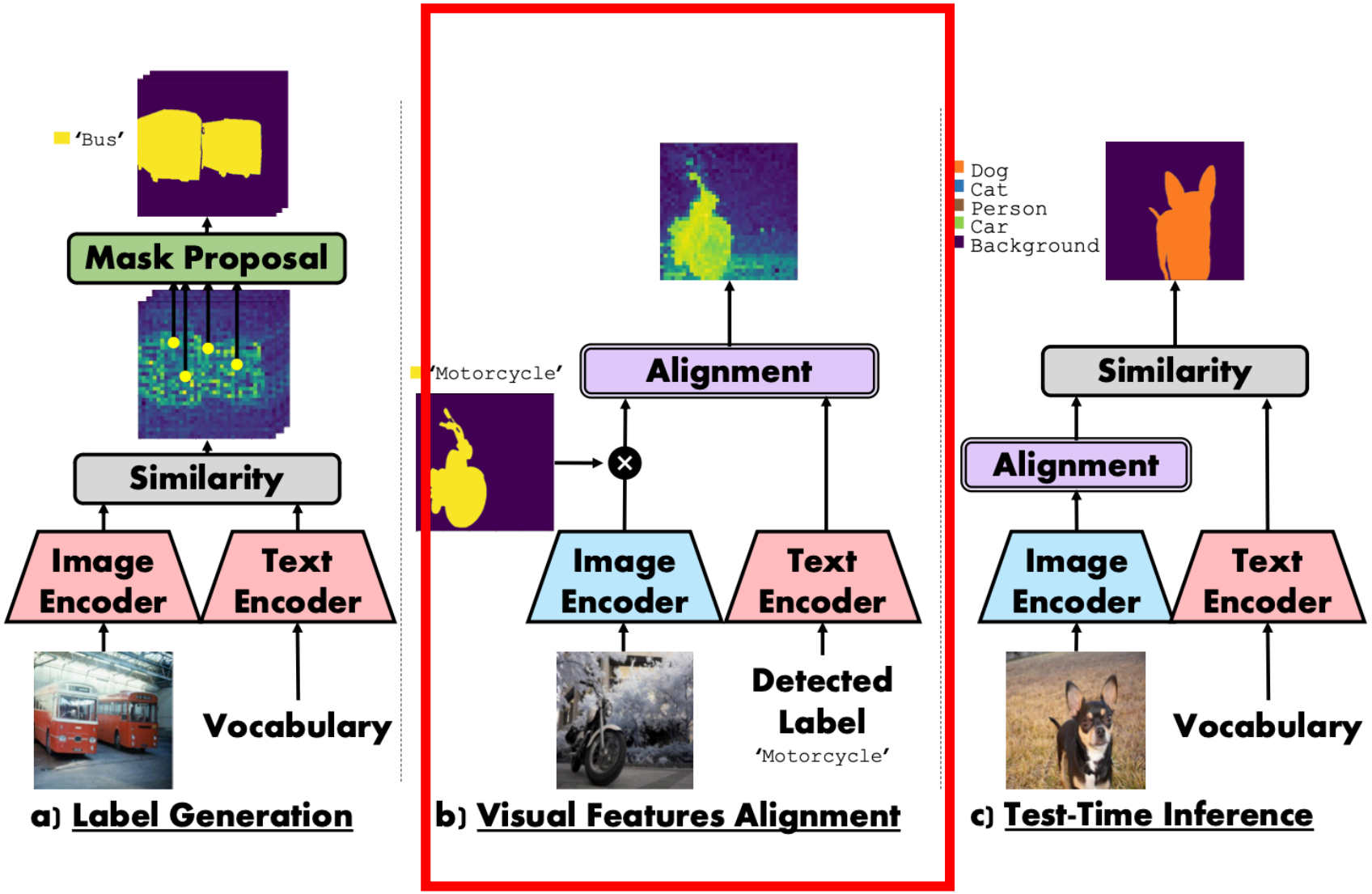
Label Generation



Label Generation

- An image and a text encoder: CLIP to recognize categories present in the image.
- CLIP is trained to maximize the cosine similarity between the image class token and the text representation.
- Sliding window
- Vocabulary: all possible labels in a given dataset are considered as the vocabulary

Visual Features **Alignment**



Stage 1.1: Querying SAM with CLIP

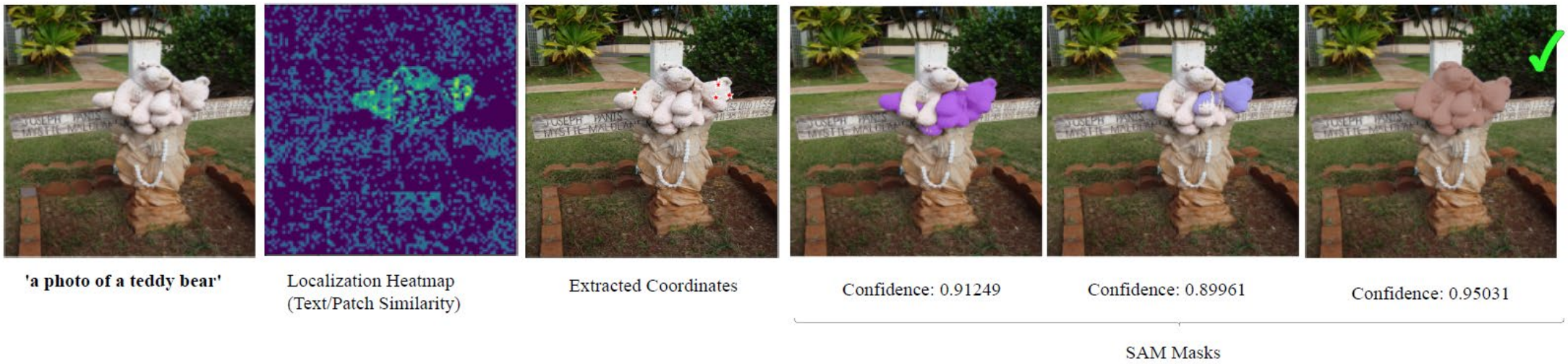


Fig. 2: Query point selection and mask generation with SAM: Our method selects 5 patches with the highest similarity to the detected object's text embedding. The coordinates for the center of these patches are forwarded to SAM which generates 3 different segmentation masks for the object. Our method selects the segmentation mask with the highest confidence.

Stage 1.2: SAM Masks classification

- Disadvantage of Stage 1.1: May ignore small objects or generate partial masks for an image

Stage 1.2: SAM Masks classification

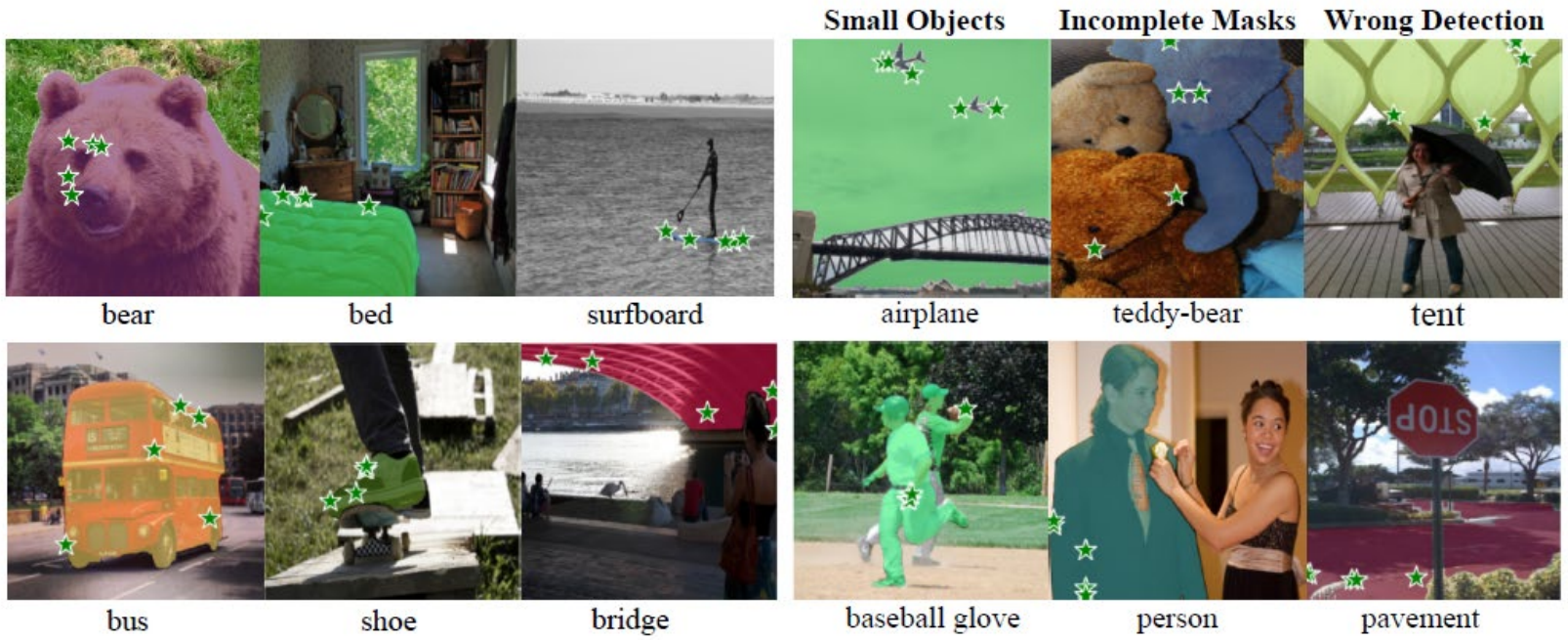


Fig. 5: Qualitative evaluation of Stage.1.1. SAM query points are shown in green stars. Top sub-figure shows instances of correct detected labels and segmentations by Stage.1.1. Below demonstrates its limitations. Small objects, wrongly detected classes (due to ambiguities) and not enough query points to cover all instances. Stage 1.2 alleviates the issue with small objects and incomplete masks since it labels all image masks generated by SAM.

Stage 1.2: SAM Masks classification

- **Automatic mask generation**
  - Masks extracted from the full image using SAM's automatic mask generation pipeline
  - Constrain the masks by size and predicted IOU to filter out the low quality and duplicate masks
- **Mask labelling:** In order to classify each generated mask by SAM
  - CLIP's mean feature embedding corresponding to the area covered by the mask
  - Its similarity to the text features of the detected categories in the image
  - The class with the highest similarity is selected as the pseudo label for the corresponding mask.

Stage.2: Lightweight semantic segmentation

- **Alignment Module:** Map the image patch features to the text embedding
  - Predicted masks along with their corresponding predicted categories as pseudo labels.
  - Apply DINOv2, a recent model trained in a fully self-supervised fashion with no text alignment.
  - To overcome the noisy: pseudo annotations
    - frozen pretrained text features as anchors
    - already discriminative image patches features
    - a loss function that is robust to noise

Stage.2: Lightweight semantic segmentation

- **Pseudo label generation: …**
- **Training the alignment module: New Robust loss function SupCon**

text pairs $< \mathbf{z}_i, \mathbf{t}_k >$ are positive if $y_i = k$. We construct a loss function of two terms operating on the two types of said pairs:

$$\ell_{\text{TSupCon}} = \frac{1}{B * N + K} \left(\sum_{k=1}^{K} \ell_{\mathbf{t}}(\mathbf{t}_k) + \sum_{i}^{B*N} \ell_{\mathbf{im}}(\mathbf{z}_i) \right), \tag{1}$$

where B is the batch size and N is the number of patches in an image; K is the number of text features. $\ell_{\mathbf{t}}$ is designated for optimizing patch-text pairs

$$\ell_{\mathbf{t}}(\mathbf{t}_k) = \frac{1}{N_k} \sum_{i:y_i=k} \ell_{\mathbf{t}}(\mathbf{z}_i, \mathbf{t}_k), \tag{2}$$

where N_k is the number of patches with label $y = k$.

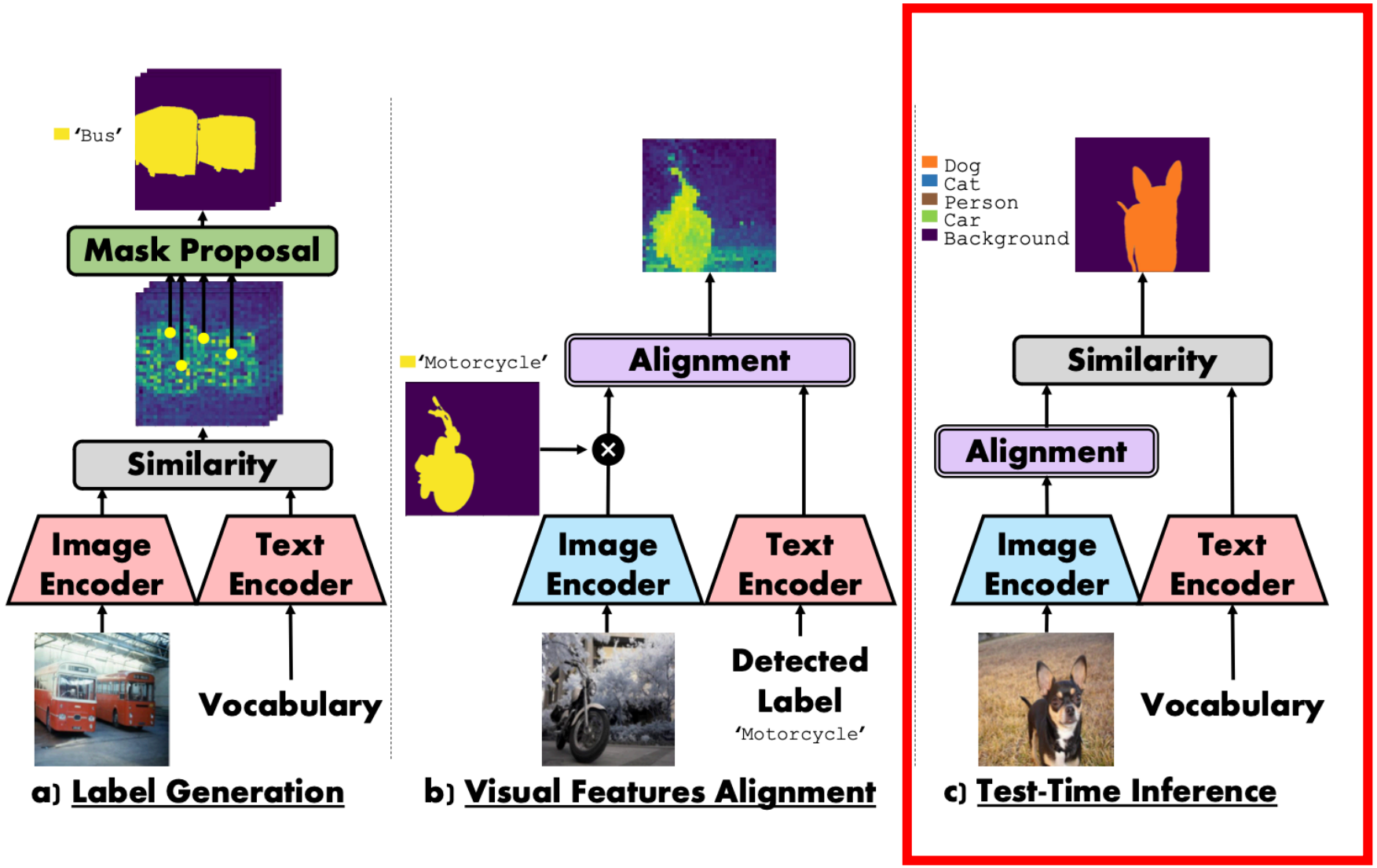
$$\ell_{\mathbf{t}}(\mathbf{z}_i, \mathbf{t}_{y_i}) = -\mathbf{z}_i^\top \mathbf{t}_{y_i} + \log \left(\sum_{k=1}^{K} \exp(\mathbf{z}_i^\top \mathbf{t}_k) + \sum_{j \neq i} \exp(\mathbf{z}_i^\top \mathbf{z}_j) \right). \tag{3}$$

The loss $\ell_{\mathbf{t}}(\mathbf{t}_k)$, defined for each text feature $\mathbf{t}_k$, considers all the patches that belong to the category $y = k$, represented by the text feature $\mathbf{t}_k$. The loss is minimized by maximizing the similarity of the concerned patch-text pairs, normalized over all other constructed pairs (patch-patch and patch-text pairs) for a given patch $\mathbf{z}_i; y_i = k$.

The loss applied to each patch feature is defined as follows:

$$\ell_{\mathbf{im}}(\mathbf{z}_i) = \frac{1}{N_{y_i}} \sum_{l:y_l=y_i} \left(-\mathbf{z}_i^\top \mathbf{z}_l + \log \sum_{j \neq i} \exp(\mathbf{z}_i^\top \mathbf{z}_j) \right), \tag{4}$$

Test-Time Inference



Experiment

Patch-level alignment between image and class

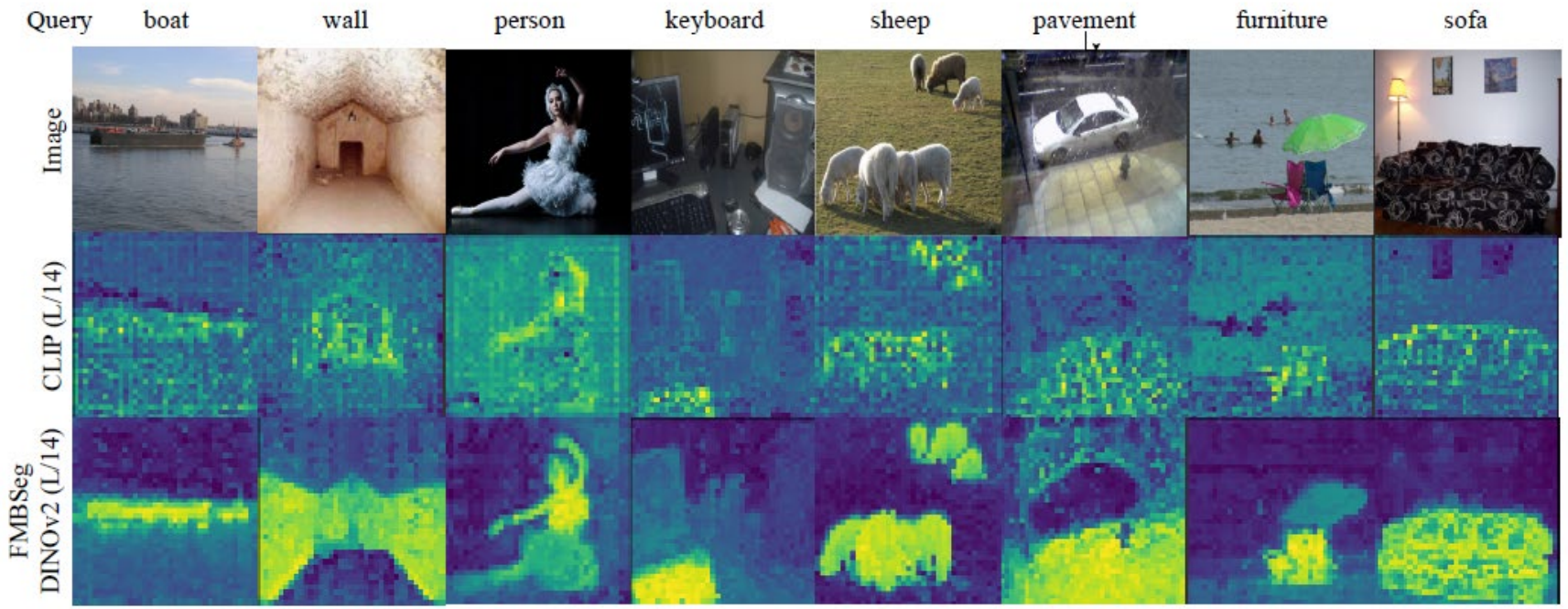


Fig. 3: Patch-level alignment between image and class. First row shows the original image from Pascal VOC. Second row shows the similarity between the high-resolution patch features from CLIP-L/14 and the image-level label. Third row shows the similarity map after aligning a DINOv2-L/14 model with FMbSeg.

Qualitative results of zero-shot segmentation

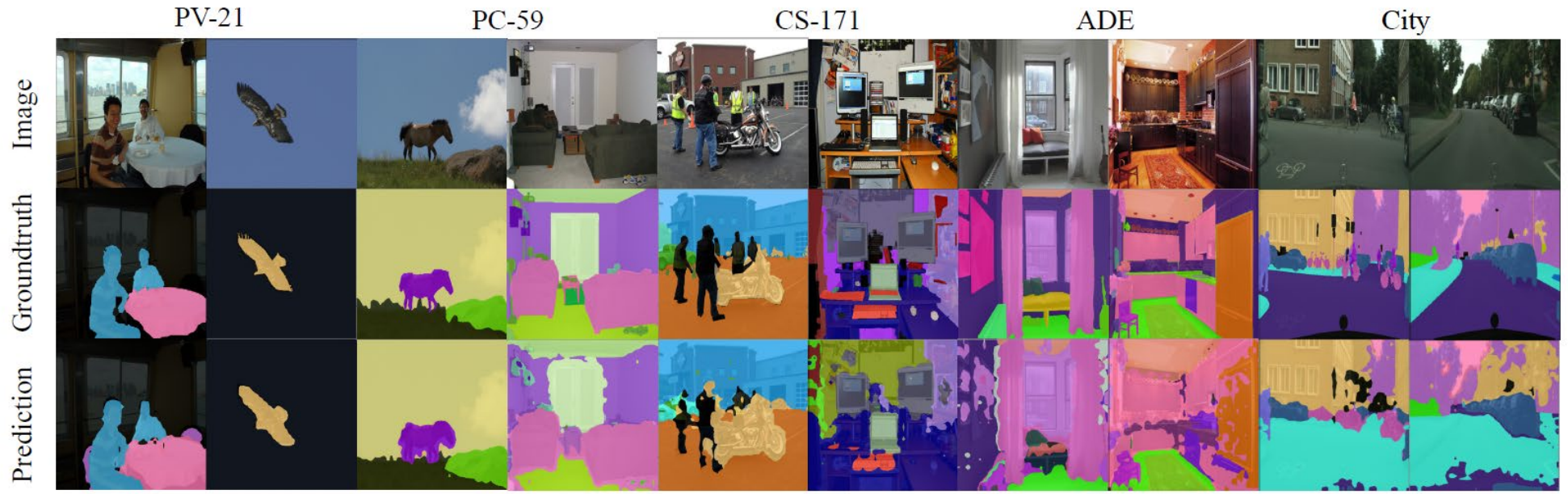


Fig. 4: Qualitative results of zero-shot segmentation. We show 2 samples for each one of the considered datasets. The first row shows the ground truth labels. The second row shows the results of FMbSeg.

Semantic Segmentation performance on various datasets

Method	Training Data	mIoU						
		PV-21	PV-20	PC-59	CO-80	CS-171	City	ADE
GroupVit [34]	41M +Labels	50.4	79.7	18.7	27.5	15.3	11.1	9.2
Mask CLIP [43]	-	38.8	74.9	23.6	20.6	16.4	12.6	9.8
ReCo [29]	-	25.1	57.7	19.9	31.6	14.8	21.1	11.2
TCL [7]	15M + Labels.	55.0	83.2	33.9	31.6	22.4	24.0	**17.1**
CLIP-S4 [16]	0.12M	-	-	33.6	-	22.1	-	-
SCCLIP [31]	-	59.1	-	30.4	-	22.4	-	-
CLIP-DIY [33]	-	59.0	-	30.4	-	-	-	-
CaR [30]	-	67.6	-	30.5	36.6	-	-	
SAM-CLIP [32]	41M	60.6	-	29.2		**31.5**	-	**17.1**
FMbSeg (ours)	0.12M	**67.73**	**85.65**	**42.72**	**57.63**	29.88	**28.36**	16.25

Table 1: Semantic Segmentation performance on various datasets. Best method marked in bold and second best in gray, our method is better or on par with SOTA methods.

Choice of Architecture (Alignment)

Choice of Architecture We design our alignment module as a single transformer block where multi-head self-attention is applied over image patches. We ablate our choice against other designs, namely a single linear layer and a Multi-layer perceptron (MLP) with GELU activation. Table 2 reports reports the mIOU on CS-171. The differences are not substantial, with MLP achieving better better performance than linear. The transformer block further improves over the MLP.

Module Arch.	CS-171 mIoU
Linear	25.32
MLP	26.43
Transformer block	**27.61**

Alignment Loss	CS-171 mIoU
ℓ_{SupCon}	24.61
$\ell_{\mathbf{t}}$ (2)	26.85
ℓ_{TSupCon} (1)	**27.61**

Table 2: Left. **Comparison of different design choices for our alignment module**, a transformer block has a small advantage. Right. **Comparison of different losses**. SupCon alone is inferior to our full loss TSupCon performing the best.

Conclusion

Conclusion

- Foundational models compositionality.
- Alignment module can improve VLM models that are based on CLIP to further strengthen their object localization capabilities.