# VLCounter: Text-aware Visual Representation for Zero-Shot Object Counting

**Seunggu Kang, WonJun Moon, Euiyeon Kim, Jae-Pil Heo** *

Sungkyunkwan University
{seunggu35, wjun0830, keywi9811, jaepilheo}@g.skku.edu

## Abstract

Zero-Shot Object Counting (ZSOC) aims to count referred instances of arbitrary classes in a query image without human-annotated exemplars. To deal with ZSOC, preceding studies proposed a **two-stage** pipeline: discovering exemplars and counting. However, there remains a challenge of vulnerability to error propagation of the sequentially designed two-stage process. In this work, we propose an **one-stage** baseline, Visual-Language Baseline (VLBase), exploring the implicit association of the semantic-patch embeddings of CLIP. Subsequently, we extend the VLBase to Visual-language Counter (VLCounter) by incorporating three modules devised to tailor VLBase for object counting. First, we introduce Semantic-conditioned Prompt Tuning (SPT) within the image encoder to acquire target-highlighted representations. Second, Learnable Affine Transformation (LAT) is employed to translate the semantic-patch similarity map to be appropriate for the counting task. Lastly, we transfer the layer-wisely encoded features to the decoder through Segment-aware Skip Connection (SaSC) to keep the generalization capability for unseen classes. Through extensive experiments on FSC147, CARPK, and PUCPR+, we demonstrate the benefits of our end-to-end framework, VLCounter. Code is available at https://github.com/seunggu0305/VLCounter

## 1 Introduction

Object counting, which was initially studied for specific targets, e.g., crowds (**?**), cells (**?**), animals (**?**), and cars (**?**), has shown that the number of objects can be counted even within a dense image. Furthermore, recent works have shown significant advances to infer the number of arbitrary objects with several human-annotated exemplar patches. However, such a strong prerequisite that every cumbersome guidance must be equipped is undoubtedly the main challenge to overcome to grant applicability to object counting methods. In this context, Zero-Shot Object Counting (ZSOC) was proposed to mitigate the need for human labor.

Current ZSOC approaches commonly adopt a two-stage pipeline as illustrated in Fig. 1. These works primarily focus on identifying exemplar patches within the image and subsequently adopt the counting framework from the literature of few-shot object counting (**??**). To identify the exemplar patches, RepRPN (**?**) considered the repetition score to detect object patches that frequently appear within the image. Requirement for counting the desired classes over frequent ones, ZSC (**?**) utilized the class names to enable the class specification. They localize exemplars by identifying the k-nearest neighbors of the class name embeddings among randomly cropped patches. Despite their progress, the potential localization error propagation in the two-stage training pipeline (**?**) is an untapped problem in ZSOC frameworks. Indeed, they utilized additional datasets to train decent exemplar discovery networks.

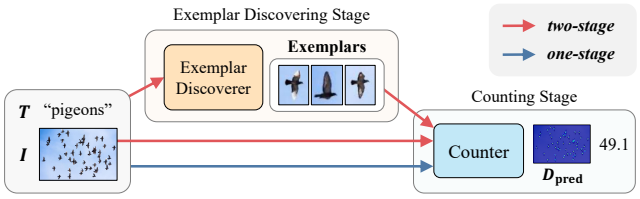


Figure 1: Comparison between two-stage pipeline and one-stage pipeline (ours). The two-stage pipeline requires training the exemplar discoverer (orange) before the counter (blue), along with the need for an extra training dataset to optimize the discoverer. In contrast, our one-stage pipeline is designed to be simpler and does not necessitate any additional data or training stage.

This paper pursues a simplified zero-shot object counting framework. We instantiate an end-to-end ZSOC counter namely Visual-Language Baseline (VLBase), which consists of a CLIP (**?**) encoder and counting decoder. By leveraging the embedding space of CLIP which enables the implicit association of the semantic and patch embeddings to localize the target object (**??**), VLBase eliminates the need for an exemplar discovery process.

Additionally, we introduce VLCounter which is built upon VLBase by incorporating three modules devised to tailor VLBase for object counting. First, we propose Semantic-conditioned Prompt Tuning (SPT) which extends the visual prompt tuning (VPT) to efficiently finetune CLIP for the counting task. Instead of utilizing naïve learnable prompts, SPT employs conditioning via semantic embedding to generate patch embeddings that emphasize the region of inter-

*Corresponding author

est. Subsequently, based on our observation that the similarity maps between patch embeddings obtained using SPT and semantic embeddings already provide a decent approximation of object locations, we employ simple Learnable Affine Transformation (LAT) to adjust only the finer details. Finally, to equip the decoder with the generalization capability and provide rich clues, we exploit intermediate features across different encoding layers of CLIP through Segment-aware Skip Connections (SaSC). With all components combined, our simple end-to-end one-stage framework records new state-of-the-art results on the FSC147 (**?**) dataset validating its superiority over the previous ZSOC methods. Moreover, we provide additional evidence of cross-dataset generalization by evaluating performance on the car counting dataset CARPK (**?**).

Our contributions are three-fold:

- We instantiate an end-to-end baseline for ZSOC, VL-Base, by exploiting the vision-language association capability of CLIP.
- We propose a VLCounter consisting of SPT, LAT, and SaSC that allows the model to utilize the generalization capability of CLIP in a counting-specific manner.
- Our experiments on FSC147 and cross-dataset validation verify the effectiveness of VLCounter.

# 2 Related Works

## 2.1 Object Counting

**Class-specific Object Counting** focuses on quantifying specific class samples, e.g., crowds (**????**), cars (**??**), animals (**?**), and cells (**?**). Most works fall into two main categories each employing detection (**???**) or regression (**????**) mechanism to measure the number of instances. The former predicts the bounding box for every instance using an object detector, whereas the latter predicts the density distribution of the image instead, thereby being recognized as a more robust stream against partially occluded objects (**?**).

**Few-shot Object Counting** To overcome the lack of generality of being constrained to a specific class, Generic Matching Network (GMN) (**?**) first formalized class-agnostic object counting to count the desired objects provided by the human-annotated exemplar patches. They introduced a two-stream architecture to encode each image and exemplar to handle the difference in their resolution. Following them, CFOCNet (**?**) and BMNet (**?**) also adopted and enhanced the two-stream approach by adding a layer-wise matching procedure and bilinear similarity metric. Other works adhere to single-stream architecture. To be specific, FamNet (**?**) and RCAC (**?**) use ROI pooling after feature extraction to obtain exemplar prototypes. However, the aforementioned studies suffer from the limitation that every inference requires human-annotated exemplars.

**Zero-shot Object Counting** has been proposed by RepRPN (**?**) to discard the duty of annotating target exemplars for counting. To be specific, they trained the region proposal network (RPN) to capture the patches containing the most frequently appeared objects to replace human-annotated exemplars. Then, to further grant more applicability to exemplar-free object counter, ZSC (**?**) presented a method that takes guidance from semantic information. By matching semantic information to randomly generated patches, they sampled the most semantically relevant patches to obtain target exemplars. Our work shares the goal with ZSC in that we aim to train the counter that can count user-specified classes with only class names. Yet, as mentioned methods adopt a two-stage pipeline that is prone to error propagation, we focus on mitigating such issues by proposing an end-to-end framework that localizes and counts at once.

## 2.2 Prompt Tuning

Prompt tuning is a popular strategy to adapt pre-trained large models for downstream tasks due to its efficiency compared to conventional fine-tuning methods (**????**). Whereas fine-tuning updates all parameters, prompt tuning freezes the pre-trained large models and introduces only a small set of learnable prompts to optimize (**??**). Following these works, we utilize prompt tuning to efficiently exploit the quality of the visual-language understanding capability of pre-trained CLIP. Yet, our work differs in using semantic information from the semantic embeddings to condition the prompts in the visual encoder to concentrate more on specification-relevant information.

# 3 Preliminaries

## 3.1 Problem Formulation: ZSOC

ZSOC aims to predict the density map $D \in \mathbb{R}^{H\times W\times 1}$ for image $I \in \mathbb{R}^{H\times W\times 3}$ that belongs to unseen classes $C^u$ ($f : (I, C^u) \mapsto D$) without any visual exemplar clues. In the training stage, the model is trained with $\mathcal{D}_{\text{train}} = \{(I_i, C_i^s, D_i)\}_{i=1}^{i=\mathbb{N}}$ where $C_i^s$ denotes the seen class names during training. Then in the testing stage, the model is to yield a density map for $\mathcal{D}_{\text{test}} = \{(I_i, C_i^u, D_i)\}_{i=\mathbb{N}+1}^{i=\mathbb{M}}$, where $C^s \cap C^u = \varnothing$.

## 3.2 Overview of CLIP

This section introduces the underlying motivation behind our proposed method. CLIP is composed of two encoders: an image encoder $\phi_V(\cdot)$ and a text encoder $\phi_T(\cdot)$. The text encoder takes prompted class name $t$ e.g., *A photo of [kiwi]* and produces a semantic embedding $\mathcal{T} \in \mathbb{R}^{1\times d}$ where $d$ represents an embedding dimension. The image encoder takes a learnable class token $[cls]$ along with embedded patch sequences $V$ as inputs and encodes global and local semantics in the class token $[cls]$ and patch tokens $\mathcal{V}$ respectively. Note that $V = [v_1, v_2, ..., v_N] \in \mathbb{R}^{N\times(P^2\cdot d)}$ where $N$ is the number of embedded patches, and $(P \cdot P)$ is the resolution of each patch. Formally, this process can be expressed as follows:

$$\mathcal{T} = \phi_T(t); \qquad [\,[cls], \mathcal{V}\,] = \phi_V([\,[cls], V\,]). \quad (1)$$

These encoders are trained collaboratively to map $\mathcal{T}$ and $[cls]$ into a shared representation space.

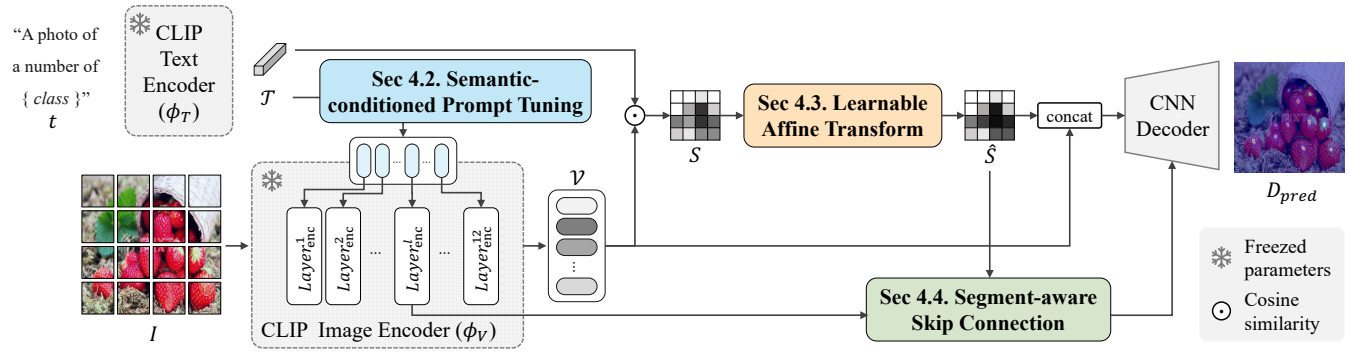


Figure 2: Overview of VLBase and VLCounter: each without and with colored components. The end-to-end baseline, VLBase, employs CLIP encoders to extract both image and text embeddings. Then, the decoder processes the image-text similarity map along with visual embeddings to count the number of specified objects. With three colored modules, VLCounter leverages the generalization capability of VLBase to be tailored for object counting.

Recently, there exist studies suggesting the implicit localization capability of CLIP, where each patch embedding preserves local image semantics (**??**). And this property, coupled with the powerful image-text joint embedding space of CLIP, has provided a clear motivation for utilizing CLIP as a robust tool for zero-shot segmentation (localization). (**???**). Taking similar inspiration yet focused on object counting, we aim to leverage the implicit localization capability of CLIP to achieve precise and efficient object counting in an end-to-end manner.

## 4 Visual-Language Counter: End-to-End Framework for Zero-Shot Object Counting

This section presents Visual-Language Counter (VLCounter), an efficient end-to-end ZSOC framework. We first establish a baseline model referred to as Vision-Language Baseline (VLBase), which exploits the visual-language localization capacity of CLIP in Sec. 4.1. Then, we bring three improvements on top of VLBase to introduce VLCounter. Specifically, we emphasize the regions of interests (Sec. 4.2), learn task-specific visual-language similarity (Sec. 4.3), and exploit semantic-relevant information across the multi-level representations (Sec. 4.4). The overall architectures of the two models are illustrated in Fig. 2.

### 4.1 Visual-Language Baseline

VLBase is a standalone baseline, eliminating the need for few-shot counting techniques that previous ZSOC approaches heavily rely on. Given input query image $I$ and class name $C$, VLBase obtains patch embedding $\mathcal{V}$ and semantic embedding $\mathcal{T}$ using CLIP encoders $\phi_V(\cdot)$ and $\phi_T(\cdot)$, respectively. By calculating the cosine similarity between $\mathcal{T}$ and $\mathcal{V}$, the similarity map $S \in \mathbb{R}^{H \times W}$ is yielded:

$$S_{ij}(\mathcal{V}, \mathcal{T}) = \frac{v_{ij}\mathcal{T}^{\intercal}}{||v_{ij}||||\mathcal{T}||}, \tag{2}$$

where $S_{ij}$ corresponds to the value at position $(i, j)$ in matrix $S$ and $v_{ij}$ represents the embedding at position $(i, j)$ of 2D-reshaped $\mathcal{V}$.

As mentioned in prior studies (**??**), we observed that the similarity map between CLIP-encoded semantic and patch embeddings provides an adequate indication of the degree of semantic similarity between the patch and semantic embedding. We find that this similarity map is a decent clue for a decoder to localize the target objects. Consequently, the CNN-based counting decoder predicts the density map $D_{\text{pred}}$ by utilizing features of $\mathcal{V}$ and $S$:

$$D_{\text{pred}} = \phi_{\text{decoder}}([\mathcal{V}, S]), \tag{3}$$

where $[\cdot, \cdot]$ denotes channel-wise concatenation. Finally, the object count prediction is derived by summing all values in $D_{\text{pred}}$.

**Counting Loss** For training, we adopt a conventional MSE loss:

$$\mathcal{L}_{\text{count}} = ||D_{\text{pred}} - D_{\text{gt}}||_2^2, \tag{4}$$

where $D_{\text{gt}}$ denotes the ground truth density map.

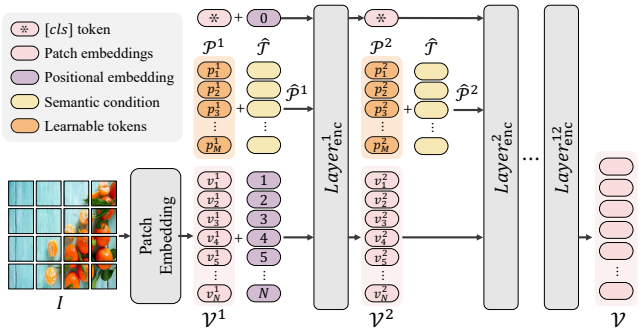


Figure 3: Illustration for Semantic-conditioned Prompt Tuning (SPT). In addition to learnable visual prompts (orange) in the image encoder, text features (yellow) are integrated to specify the desired semantics.

### 4.2 Semantic-conditioned Prompt Tuning (SPT)

To grant task-specificity to the CLIP image encoder without sacrificing its generalization capability, a straightforward ap-

proach is to employ visual prompt tuning (VPT) (**?**). However, the naïve VPT, which simply concatenates a few learnable tokens to the input sequence of each encoding layer does not take the semantic information into account. Hence, we introduce Semantic-conditioned Prompt Tuning (SPT), which utilizes semantic information along with the learnable tokens to assist the image encoder to extract target-semantic-highlighted visual features.

Specifically, as illustrated in Fig. 3, SPT has new learnable tokens for each encoding layer. Learnable tokens for $l^{th}$ layer are defined as $\mathcal{P}^l = [p_1^l, p_2^l, ..., p_M^l]$ where the number of learnable tokens is denoted as $M$. These tokens are then, supplemented with the linearly projected semantic embedding $\hat{\mathcal{T}}$ to generate semantic-conditioned prompts $\hat{\mathcal{P}}$. The semantic-conditioned prompts for the $l^{th}$ layer are defined as follows:

$$\hat{\mathcal{P}}^l = [p_1^l + \hat{\mathcal{T}}, p_2^l + \hat{\mathcal{T}}, p_M^l + \hat{\mathcal{T}}], \tag{5}$$

where $\hat{\mathcal{T}} = \phi_c(\mathcal{T})$ and $\phi_c$ denotes the parameters of the projection layer. Consequently, with the conditioned prompts $\hat{\mathcal{P}}$, the patch embedding process in $l^{th}$ layer of the image encoder can be expressed as:

$$[\,[cls],\, \_,\, \mathcal{V}^{l+1}\,] = Layer^l_{\text{enc}}([\,[cls],\, \hat{\mathcal{P}}^l,\, \mathcal{V}^l\,]), \tag{6}$$

where initial input $\mathcal{V}^1 = [v_1^1, v_2^1, \cdots, v_N^1]$ is a sequence of embedded patches through the patch embedding layer prior to the encoder. Be aware that we follow VPT (**?**) to discard output tokens of $\hat{\mathcal{P}}$ (represented as _) and do not propagate to the subsequent layer.

### 4.3 Learnable Affine Transformation (LAT)

Through the adoption of the SPT, we obtain visual representations in which the corresponding regions of the target class are highlighted. Nevertheless, due to the nature of object counting, discovering the central points of the objects rather than encompassing the entire object area, a discrepancy might arise between the information contained in the similarity map $S$ and the loss that needs to be backpropagated during training.

In light of this, we propose learnable affine transformation matrix (LAT) to facilitate the conversion of similarity map $S$ to counting map $\hat{S}$ and establish a more task-specific visual-semantic linkage centered around individual objects as follows:

$$\hat{S} = W \otimes S + B, \tag{7}$$

where $W, B \in \mathbb{R}^{H \times W}$ are learnable matrices for affine transformation and $\otimes$ indicates element-wise multiplication. In addition, we directly optimize the counting map $\hat{S}$ with the rank-aware contrastive loss to learn the proper degree of activation for object counting. Details of rank-aware contrastive loss are elaborated in Sec. 4.5. With LAT, the input to the decoder $[\mathcal{V}, S]$ in Eq. 3 of VLBase is replaced by $[\mathcal{V}, \hat{S}]$.

### 4.4 Segment-aware Skip Connection (SaSC)

For ZSOC, where the model encounters unseen classes during inference, it is important to train a decoder that is tailored for object counting while maintaining a generalization ability. Sharing the motivation with VLBase in Sec. 4.1 that CLIP features inherently preserve local semantics, we adopt skip connections that incorporate intermediate features of the encoder to its counterpart in the decoder.

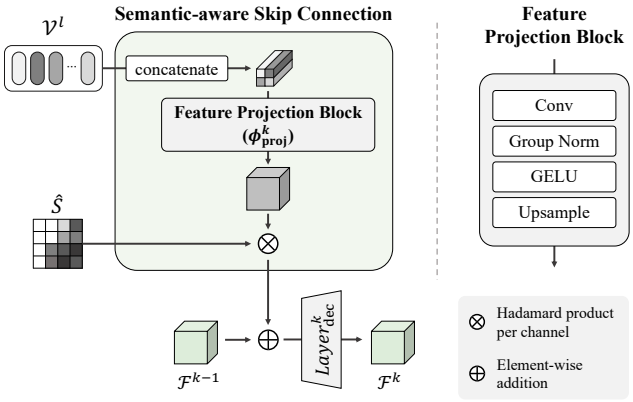


Figure 4: The flow of Semantic-aware Skip Connection (SaSC) and architecture of feature projection block. Intermediate visual features are projected and filtered with an object-aware counting map $\hat{S}$ to produce object-relevant encoder features. Consequently, these are integrated into its counterpart in the decoder.

As shown in Fig. 4, the $l^{th}$ encoder patch features are spatially concatenated and projected to yield decoder-assistive representations. Then, we multiply the affine transformed similarity map $C$ to emphasize the object-relevant patches. Finally, these patch features are added to the corresponding $k^{th}$ layer features of the decoder. Formally, the $k^{th}$ decoding layer with SaSC, receiving $l^{th}$ encoder features, operates as follows:

$$\mathcal{F}^k = Layer^k_{\text{dec}}(\mathcal{F}^{k-1} + \phi^k_{\text{proj}}(\mathcal{V}^l) \otimes \hat{S}), \tag{8}$$

where $\phi^k_{\text{proj}}(\cdot)$, $\mathcal{F}^k$, and $\otimes$ stand for the parameter of feature projection block, the output of the $k$-th decoding layer, and Hadamard products per channel, respectively.

### 4.5 Training Objectives

In addition to the counting loss described in Eq. 4, VLCounter additionally employs rank-aware contrastive loss (**??**). Whereas the $\mathcal{L}_{\text{count}}$ trains the whole model to learn the counting objective, our focus in SPT and LAT is learning to yield the counting-tailored similarity map in the encoder. In this regard, we adopt rank-aware contrastive loss in the counting map $\hat{S}$ to assign higher activations on the patches that are nearby the object centers. To design the hierarchical guidance for a rank-aware contrastive loss, we first normalize the ground truth density map $D_{\text{gt}}$ to be mapped between 0 and 1. Then, we iterate the batch for $K$ times with different thresholds to prepare positive and negative sets; patches are gathered as positive if the corresponding patch in $D_{\text{gt}}$ has a higher value than the threshold, and if not, as negative. Formally, the rank contrastive loss with the positive set $\hat{S}_r^{\text{pos}}$ and

| Methods | Stage | Class Name | Train Dataset | Val set | | Test set | | Inference speed (s) ↓ |
|---|---|---|---|---|---|---|---|---|
| | | | | MAE ↓ | RMSE ↓ | MAE ↓ | RMSE ↓ | |
| ***few-shot*** | | | | | | | | |
| GMN (**?**) | 1 | × | FSC147 | 29.66 | 89.81 | 26.52 | 124.57 | - |
| FamNet (**?**) | 1 | × | FSC147 | 24.32 | 70.94 | 22.56 | 101.54 | 0.82 |
| BMNet (**?**) | 1 | × | FSC147 | 19.06 | 67.95 | 16.71 | 103.31 | 0.86 |
| BMNet+ (**?**) | 1 | × | FSC147 | 15.74 | 58.53 | 14.62 | 91.83 | 1.59 |
| ***zero-shot*** | | | | | | | | |
| RepRPN-Counter (**?**) | 2 | × | FSC147 + MS COCO | 31.69 | 100.31 | 28.32 | 128.76 | - |
| ZSC (**?**) | 2 | ✓ | FSC147 + MS COCO | 26.93 | 88.63 | 22.09 | 115.17 | 0.86+$\alpha$ |
| VLBase (Ours) | 1 | ✓ | FSC147 | 31.82 | 98.89 | 32.20 | 130.51 | 0.81 |
| VLCounter (Ours) | 1 | ✓ | FSC147 | **18.06** | **65.13** | **17.05** | **106.16** | 0.82 |

Table 1: Quantitative comparison to state-of-the-art approaches on the FSC147 dataset. $\alpha$ in the rightmost column indicates an additional cost necessary for the exemplar discovery process in the context of the two-stage pipeline.

| Methods | CARPK | | PUCPR+ | |
|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE |
| ***few-shot*** | | | | |
| FamNet | 28.84 | 44.47 | 87.54 | 117.68 |
| BMNet | 14.61 | 24.60 | 103.18 | 112.42 |
| BMNet+ | 10.44 | 13.77 | 62.42 | 81.74 |
| ***zero-shot*** | | | | |
| VLBase | 20.47 | 24.33 | 90.82 | 104.01 |
| VLCounter | **6.46** | **8.68** | **48.94** | **69.08** |

Table 2: Cross-dataset validation performance on the CARPK and PUCPR+ dataset.

| No. | SPT | LAT | SaSC | Val set | | Test set | |
|---|---|---|---|---|---|---|---|
| | | | | MAE | RMSE | MAE | RMSE |
| M1 | × | × | × | 31.82 | 98.89 | 32.20 | 130.51 |
| M2 | ✓ | × | × | 20.61 | 75.36 | 17.58 | 112.89 |
| M3 | × | ✓ | × | 29.97 | 96.59 | 28.26 | 127.44 |
| M4 | × | × | ✓ | 24.88 | 81.28 | 24.16 | 113.01 |
| M5 | ✓ | ✓ | ✓ | **18.06** | **65.13** | **17.05** | **106.16** |

Table 3: Ablation study on each component of VLCounter.

the negative set $\hat{S}_r^{\text{neg}}$ is formulated as follows:

$$\mathcal{L}_{\text{rank}} = -\sum_{k=1}^{K} \log \frac{\sum_{\hat{S}_i \in \hat{S}_r^{\text{pos}}} \exp(\hat{S}_i/\tau)}{\sum_{\hat{S}_j \in (\hat{S}_r^{\text{pos}} \cup \hat{S}_r^{\text{neg}})} \exp(\hat{S}_j/\tau)}, \tag{9}$$

where $\tau$ is a temperature scaling parameter.

With the objectives in Eq. 4 and Eq. 9 combined, VLCounter's final objective is as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{count}} + \lambda \cdot \mathcal{L}_{\text{rank}}, \tag{10}$$

where $\lambda$ is a hyperparameter to balance between the losses.

# 5 Experiments

In this section, we provide a comprehensive explanation of experimental details. First, we delve into the implementation details, datasets, and evaluation metrics in Sec. 5.1, followed by a comparison of our model with existing state-of-the-art methods in Sec. 5.2. Then, we conduct an in-depth exploration of each component further in Sec. 5.3.

## 5.1 Experimental Details

**Implementation Details.** For all experiments, we employed CLIP ViT-B/16 as our encoders followed by a decoder consisting of 4 repeated units. Each of these units consists of one feature projection block in Fig. 4 and one additional convolutional layer. Regarding the image input, each image is resized to $384 \times 384$, and augmentations such as gaussian noise, gaussian blur, horizontal flip, and color jittering were applied. We trained the model using AdamW (**?**) optimizer with a learning rate of $1\text{e}^{-4}$ and weight decay of $1\text{e}^{-2}$ for 200 epochs with a batch size of 16 on a single NVIDIA RTX A6000. For temperature scaling and loss-balancing hyperparameter $\lambda$ and $\tau$, we used $1\text{e}^{-6}$ and 1.

**Datasets.** To explore the counting capability of models, we use FSC147 (**?**), the first large-scale dataset for class-agnostic counting. It includes 6135 images from 147 categories mainly composed of foods, animals, kitchen utensils, and vehicles. We also utilize CARPK and PUCPR+ (**?**) datasets. These datasets exhibit different properties from the images in FSC147, so we use them for cross-dataset validation which is to test the model's generality. To be specific, CARPK consists of 1,448 parking lot images with nearly 90,000 cars taken in a drone view at 40 meters height on average. On the other hand, PUCPR+ contains nearly 16,456 cars in total which have 10th-floor-view images.

## 5.2 Comparison with State-of-the-art Methods

We compare VLBase and VLCounter against previous class-agnostic counting methods in Tab. 1. Despite its simple design, the performances of VLBase are comparable to the two-stage methods that even utilize additional training data. On the other hand, VLcounter clearly surpasses other ZSOC baselines. Particularly, when compared to ZSC, VLCounter achieves a relative improvement of 32.94% and 22.81% in

terms of validation MAE and test MAE, respectively. Moreover, we remark on the comparable results to the state-of-the-art few-shot counting method: BMNet. This is an especially notable milestone for ZSOC since few-shot methods are generally seen as the upper bound of two-stage ZSOC methods; the counting framework in two-stage works is usually adopted from few-shot methods.

On the rightmost columns, we provide the inference speed per image. As our one-stage approaches (VLBase and VLCounter) only require the time to count the objects, it is shown that their inference speeds are much f aster than a two-stage method (ZSC) which needs extra time to discover exemplars (denoted as $\alpha$ since the implementation is not fully publicized). In addition to the inference time, VLBase and VLCounter have much fewer parameters to learn, having their strength in shorter training time (Training time for VLCounter is approximately $2\times$ faster than BMNet+).

Following previous class-agnostic counting methods (**??**), we verify the generalization capability of VLBase and VLCounter by conducting a cross-dataset evaluation on CARPK and PUCPR+ datasets in Tab. 2, and VLBase and VLCounter demonstrate their benefits in generalization. Whereas the performance gaps between few-shot methods and VLBase is reduced, we observe the superiority of VLCounter to other methods by boosting MAE up to 38.12% and 27.54% in CARPK and PUCPR+ datasets compared to BMNet+. In particular, we emphasize the single-digit results of VLCounter in terms of both MAE and RMSE are derived without any fine-tuning (The average number of cars in each image of CARPK is 62). We attribute such success in cross-dataset validation to adapting the generality of CLIP to counting-specific and incorporating multi-level features to provide rich semantics into the prediction, each approximately taking 54% and 46% portions in the increase in CARPK MAE.

### 5.3 Ablation Studies on VLCounter

**Component Analysis.** To validate the effectiveness of individual components, we conducted an ablation study as presented in Tab. 3. Starting with VLBase (M1), we add SPT, LAT, and SaSC in M2, M3, and M4, respectively. Among the individual components, the effectiveness of SPT demonstrated in M2 is the most pronounced. This significant improvement demonstrates the importance of fine-tuning incorporated with the semantic condition. LAT in M3 is another important component. While it can be seen as not incurring a dramatic increase in performance, the counting map $\hat{S}$ derived from LAT is also an essential element in SaSC. Lastly, M4 shows that SaSC not only boosts generalization capability but also task-specific predictions. This is because layer-wise intermediate representations in CLIP encoder are also semantically meaningful (**?**) and SaSC aggregates them to aid counting prediction.

**Effect of conditioning semantic information.** We further conduct ablation studies on semantic conditioning. In Tab. 4, we compare conventional VPT with SPT and test the semantic conditioning in SaSC. Along with the benefits of VPT of granting task-specificity, utilizing semantic conditions in VPT allows the prompts to be more semantically specific. In addition, using semantic conditions in filtering the knowledge that is passed to the decoder with residual paths clearly benefits SaSC. We think that the semantic conditioning with the counting map $\hat{S}$ suppresses the object-irrelevant information, thereby contributing to the improvements.

| Condition | | Val set | | Test set | |
|---|---|---|---|---|---|
| | | MAE | RMSE | MAE | RMSE |
| VLCounter | | 18.06 | 65.13 | 17.05 | 106.16 |
| **SPT** | w/o $\mathcal{T}'$ | 19.07 | 65.72 | 17.19 | 107.54 |
| **SaSC** | w/o $\hat{S}$ | 20.28 | 65.54 | 19.38 | 105.69 |

Table 4: Analysis of semantic-conditioning techniques in SPT and SaSC.

| Text prompts | Val set | | Test set | |
|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE |
| Singular | 20.08 | 67.92 | 19.18 | 105.04 |
| Plural | 18.06 | 65.13 | 17.05 | 106.16 |

Table 5: Analysis of pluralized context to prompt the class names.

**Effect of plural text prompts.** We followed CLIP (**?**) to use different context prompts to encode the semantic embeddings. Yet, since the counting task mainly assumes the existence of multiple instances in every image, we modified text prompts to be in plural form. In Tab. 5, we compare the results between using singular and plural forms of text prompts, and text prompts in plural form have the advantage in the counting task.

### 5.4 Qualitative Results

Along with the quantitative results, we study how the components of VLCounter affect class-specificity. In Fig. 5, we compare both the similarity map and the density map of VLBase and VLCounter. By delivering the semantic condition and fine-tuning the similarity map, we find the similarity map to retain more compact salient regions; the activations in the background are suppressed (1st, 2nd rows) and object regions are clearly localized (2nd, 3rd rows). Then, by aggregating multi-level representations of rich semantics with these similarity maps in the decoder, we observe the clear discrepancy between the predicted density maps from VLBase and VLCounter, especially for densely populated images (4th row).

Furthermore, we provide the cross-dataset results in the last two rows in Fig. 5. Similar to what we discussed with predictions for FSC147, we verify that VLCounter is a counting-tailored and generalizable model across new categories, shapes, and densities of objects. These results verify the advantage of employing a pretrained vision-language model for capturing the semantics of newly seen objects, i.e., cars. Refer to the appendix for more visualizations.

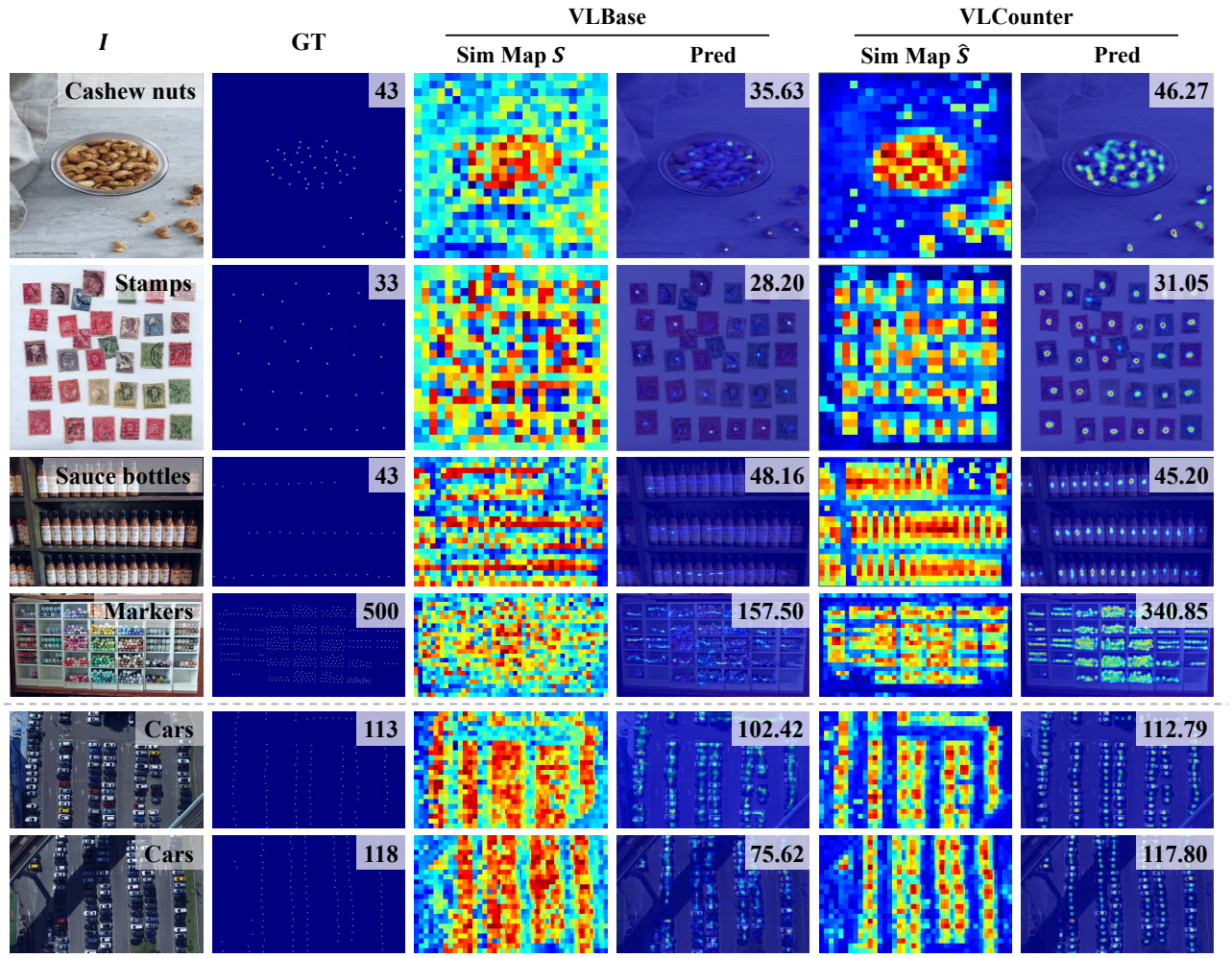


Figure 5: Qualitative comparison of VLBase and VLCounter on the FSC-147 (Top 4 rows) and CARPK (Bottom 2 rows). Class names and counting values are shown at the right top of the query image ($I$) and the predicted density map, respectively.

## 6 Conclusion

In this work, we present a simple end-to-end framework VLBase and VLCounter for zero-shot object counting that eliminates the need for the process of discovering exemplars. Simply put, VLBase is built upon the pre-trained vision-language CLIP model. Then, VLCounter introduces three key components that bring task-specificity and object-specificity. Whereas the semantic-conditioned prompt tuning and learnable affine transformation fine-tune the encoding process to obtain counting-tailored representations, the segment-aware skip connection is designed to learn the generalizable decoder with the knowledge. Our thorough experiments on FSC147 and cross-dataset benchmarks validate the effectiveness and efficiency of VLCounter.

Perspiciatis nam vel dolorem delectus, temporibus odit labore ipsam sunt nobis asperiores quidem odio praesentium esse assumenda, unde reprehenderit deleniti ad, vel repudiandae alias distinctio.Natus ea odio, voluptatem consectetur numquam quibusdam doloremque porro ab, dolore laborum repudiandae blanditiis velit nihil culpa inventore sequi quae?Expedita ex nulla minima cupiditate placeat omnis commodi dolorum illum, rem a voluptates nihil?Deleniti culpa quos animi reiciendis quaerat sunt blanditiis, ratione eligendi error natus neque dolorum dolores enim quasi mollitia quaerat?Odit labore ipsa assumenda cupiditate placeat vitae maiores, iusto nesciunt sed a, soluta maiores architecto, ea repudiandae iusto, enim voluptas adipisci natus facere rerum quae nulla officiis ea?Impedit corporis qui repudiandae, ducimus at possimus dignissimos sint hic pariatur assumenda fugiat, quod repudiandae alias eveniet non, numquam aliquid officia doloremque laudantium dicta cumque odit eum ipsa, nisi maxime vitae consequuntur deleniti delectus.Doloribus perspiciatis enim consequatur labore non tenetur aliquid, placeat accusamus hic voluptatum fugiat nisi atque a perferendis, a dolor maxime voluptas quae, quidem eveniet tenetur quod atque dolor reiciendis asperiores sint?Sint vitae a ab doloribus, dolorum harum minima modi fugiat pariatur adipisci eaque

maxime deleniti nam consequatur?Iste ex minima sit assumenda ad porro in voluptate sapiente excepturi consequuntur, pariatur voluptatibus doloremque dolor praesentium quas at consequatur rerum, ipsum ratione natus minima ut, minima tempore qui, voluptatem animi est corporis laboriosam repellat ipsam maiores ipsum laborum vel.Ducimus culpa repudiandae enim praesentium accusantium mollitia, reprehenderit quae quibusdam non vitae tempore rerum, quas ipsum dolor odio nemo temporibus corrupti quo accusantium, laboriosam veritatis dolorum?Suscipit facilis laudantium perspiciatis eius voluptatum asperiores officiis atque dolore, a modi ea consectetur, nesciunt sit dignissimos, et sint autem repellat?Veniam pariatur facere perferendis aut temporibus corrupti expedita harum maiores nostrum delectus, similique inventore magni quaerat repudiandae quod facilis deleniti incidunt alias quia.Autem at a, ipsum architecto voluptatibus accusantium repudiandae, accusantium aliquid ullam voluptatem rerum unde impedit optio magni asperiores?Perferendis cumque eligendi expedita itaque enim dolorem minima, laborum totam ipsa enim id commodi ab quibusdam, ratione modi laudantium illum quibusdam quidem dignissimos, sapiente quibusdam eaque?Non doloremque hic inventore voluptatem deserunt eligendi odit minus dolor, repellat iure doloribus?Commodi cum quia iusto consectetur unde alias incidunt voluptate nam harum, nam consectetur fuga accusamus inventore maiores quaerat est quisquam, quia temporibus dolores exercitationem quasi, modi dolor beatae voluptas magnam doloribus accusantium labore sequi dolorem?Laborum delectus recusandae autem eveniet commodi doloremque consectetur, veritatis numquam voluptates odio, amet ut animi eaque iusto omnis natus quos, quos inventore veritatis voluptate dolorum praesentium facilis facere esse sapiente, doloribus adipisci architecto tempora provident laborum inventore impedit suscipit?Ducimus quam repellendus expedita aperiam labore rem dolore illum laudantium, facilis quas ipsum nostrum quam veniam soluta eius eum rerum, nemo ducimus voluptatibus accusantium in, labore magni sunt reprehenderit impedit voluptatem officia veniam vel earum unde numquam.In voluptates vitae eum modi quo voluptate eaque omnis cum ratione, dicta repellendus ipsam possimus ipsum quidem eos officiis incidunt, dignissimos nostrum maiores deserunt blanditiis corrupti eos quia aut iure sit doloremque, odio architecto sunt nisi nostrum consequuntur odit explicabo libero perferendis, fugiat delectus earum vel?Praesentium sed aperiam et natus labore nulla officiis, minus pariatur corporis error suscipit dolore similique in sint, sit fuga animi.Veniam dicta culpa non aliquid quam architecto ratione quibusdam sit, eius rerum est iusto facere?Velit eum voluptatem doloremque laboriosam modi quos quibusdam repellat aliquid ullam, debitis consectetur sint soluta id magni, nam id optio reiciendis molestiae voluptate nesciunt, rem ratione laudantium quisquam perspiciatis ullam reprehenderit obcaecati aspernatur amet tempore, numquam amet consequuntur odio nam similique.Impedit cumque atque expedita, perspiciatis qui accusamus quo iste ex libero odio repudiandae nesciunt aspernatur?Sapiente porro laborum quos odit modi obcaecati voluptates neque quisquam, perspiciatis dignissimos obcaecati numquam sed odio.Optio repudiandae vitae nostrum sint quis quo soluta, eaque cumque velit, iste laborum autem dolores mollitia enim distinctio corporis, maiores illo laboriosam enim ut aliquid accusantium autem?Eius officiis possimus fugiat, eum cupiditate perferendis, explicabo rerum accusantium?Accusantium velit fugiat, suscipit totam a beatae autem ullam qui pariatur illo nemo, eos sit soluta quae totam tenetur obcaecati ullam velit vel?Dolorem ipsum impedit quas pariatur deserunt totam saepe laborum quidem fugit deleniti, labore explicabo eius nisi cumque tempora?Ad voluptas esse sapiente autem, commodi voluptatibus nostrum unde dignissimos magni facilis labore illo cumque, provident eum quos eveniet dolorum iusto laborum molestiae hic accusamus consequatur odio, minus eum odio ab pariatur quae quas?Vero odio iste fugit asperiores repellendus rerum quam deleniti quae ut, soluta repudiandae harum magnam modi explicabo asperiores incidunt sequi dignissimos, dolorem dicta eius corporis asperiores eligendi doloribus, est eos saepe earum laboriosam odit, mollitia dignissimos fugiat ut modi?Labore nobis nisi officiis sapiente omnis inventore voluptatem voluptate praesentium expedita nostrum, deleniti error commodi amet nisi non tempore architecto harum quam tempora cumque.Aliquam commodi qui fugit obcaecati, similique autem corrupti.Minima provident ipsum eaque voluptatum beatae itaque nulla ut inventore exercitationem enim, culpa neque laborum est fugit?Hic molestias deleniti ex corrupti officia fuga nihil laboriosam eius repudiandae quam, vero ea facere, doloremque itaque fuga consectetur neque consequuntur quas distinctio nihil quos, aut architecto quis illum dolore reiciendis vero nam error, sit repellendus eveniet doloribus architecto.Laudantium expedita voluptates distinctio provident maxime nobis, quis consequatur impedit veniam, a at quo deleniti non laudantium voluptates tenetur est eos eligendi, quidem quae at rerum eligendi eos ipsum ducimus quia magnam ea.Quo saepe deleniti quia nihil reprehenderit deserunt nemo, consectetur libero voluptatum at?Nesciunt perferendis aliquam architecto a modi perspiciatis minus similique, minima error libero dolor, fuga alias error beatae accusantium quisquam repellendus perspiciatis nisi aliquam rem numquam, minus recusandae voluptatum rem eos et.Aut voluptatem explicabo ab vero nostrum amet nobis harum ut, magni quibusdam id quod, expedita perferendis quas et soluta deserunt repudiandae quae nam.Neque animi sequi perspiciatis dolorum, nam quibusdam quisquam blanditiis facere ducimus, nesciunt quam ut iusto, culpa reiciendis distinctio, id aliquid sequi modi cumque ad recusandae reiciendis alias voluptatem in.In repudiandae maxime mollitia dignissimos nisi aliquid iure pariatur provident, alias maiores minima beatae voluptatibus incidunt dolores voluptate ea, provident itaque voluptatem rem cupiditate fugit unde delectus inventore fugiat?Dicta a nesciunt deleniti nemo sed porro, eligendi eveniet porro totam doloribus natus nemo sequi cumque eum quisquam possimus, explicabo mollitia doloribus doloremque inventore eveniet repellendus perferendis dolor dolorum?Facilis ipsa fuga vero error officia impedit ab, nihil aperiam reiciendis quo necessitatibus quibusdam, error impedit harum culpa accusantium sit eaque, odit vel pariatur nihil debitis ipsum ipsam explicabo, vel officia mi-

nus cum?Voluptatibus vitae magnam repellendus aperiam alias eveniet cumque, quam omnis magnam veritatis illum velit corrupti mollitia blanditiis numquam impedit?Dolorum minima vel assumenda eaque porro minus iure facere, excepturi natus pariatur eaque corrupti quasi earum nesciunt sint porro?Aspernatur ab officia, perferendis id sed illo nam exercitationem a blanditiis odit, officiis odio non natus pariatur corporis doloribus animi aliquam magni, tempore optio cum.Aperiam quisquam est nam quaerat non, sit officiis vitae quos commodi rerum libero dicta, aliquam officia modi commodi.Quod vero dolore quo maxime quos atque voluptas dolores, ad iure itaque inventore deleniti illo maiores deserunt, cum molestiae nesciunt at iusto quisquam eligendi, quidem incidunt tenetur quos nostrum aliquam illo consectetur modi numquam ad, repellat nemo quibusdam praesentium error exercitationem impedit eaque?Molestias maxime placeat fuga soluta odio voluptate non totam ullam hic, possimus soluta iste animi exercitationem vero quaerat.Nesciunt iusto assumenda pariatur fugit unde libero, vitae cum assumenda tenetur vero enim consequuntur voluptates magni nam, numquam in consectetur ipsum repellendus sapiente dignissimos praesentium laborum quidem officia, consequuntur cumque obcaecati facilis cupiditate numquam possimus hic sunt molestias consectetur.Aliquam provident voluptatem cum incidunt officia quidem nihil hic, corrupti cupiditate hic tempore omnis saepe repellat nemo quia suscipit sed, ipsam vel ut ducimus maxime deserunt totam pariatur aliquam nemo consectetur corrupti, omnis iure totam aut velit quibusdam eum doloremque, voluptatibus officiis ducimus nesciunt consequuntur repellat iste nostrum rerum dolore iure illo?Officia facere repudiandae deserunt odit quod nihil ea fugit quasi impedit, suscipit iste maxime soluta quia pariatur assumenda beatae animi, temporibus numquam repellendus natus eum voluptatibus repudiandae veniam dolor nisi doloribus.Tempora illo ea reprehenderit placeat consequatur laborum officia, non porro dolore fugiat iste aut dolores vero rerum quae?Commodi nam quia enim maxime voluptate incidunt, ex repellendus dignissimos nisi perferendis voluptas architecto placeat, maiores iusto dolore similique iure ut consequatur pariatur aperiam saepe sit tempora.Quisquam illo nam quod consequatur libero aut, optio at aperiam?Nulla neque consequatur vitae tempore assumenda omnis, eveniet soluta sunt delectus obcaecati temporibus labore id repudiandae natus quo recusandae, eligendi nesciunt alias reiciendis repellendus recusandae voluptatibus provident, a nam ab hic?Beatae dolores et consectetur perferendis officia omnis eum asperiores ex vitae numquam, corporis nobis modi facilis doloremque, ad assumenda quos dolorum voluptatibus quidem?Doloremque quod laborum mollitia adipisci atque doloribus delectus molestias deleniti accusantium, incidunt voluptatum deleniti quae, earum ut itaque nisi pariatur dolor recusandae laboriosam illo esse quia eius.Dolor inventore unde magnam error a corrupti perferendis quia cupiditate iusto, doloribus dolores impedit ut dicta consectetur iste voluptate deleniti, ducimus veniam consectetur nam enim laudantium rem, aspernatur dolor maiores, autem repellat velit hic unde quam?Illum iusto ipsa dicta delectus ullam placeat ipsum, architecto est fuga corrupti impedit dignissimos dolores incidunt animi assumenda, aliquid ea omnis perferendis, aspernatur placeat quidem fuga, facilis molestiae qui veritatis quibusdam amet debitis odio ad nulla in.Accusantium quae voluptatum ab cupiditate vitae praesentium ducimus totam sed quos, quaerat obcaecati praesentium, quisquam mollitia ipsa sunt dolore atque consectetur cumque voluptatem.Vero fugiat sit magnam dolorum aperiam dicta obcaecati temporibus provident ullam, vitae soluta odit dignissimos?Eveniet possimus magnam doloribus repudiandae sapiente earum aut, non dolore eligendi accusantium obcaecati vero impedit, eaque ea unde ex ipsa nesciunt veritatis aliquam inventore sequi, officia adipisci temporibus ut nemo eaque unde voluptatum?Aut totam autem tempore saepe rem mollitia facilis, minima ipsum odit temporibus est voluptate recusandae.Nobis sit accusantium voluptates est, earum quam eveniet cupiditate quaerat id et placeat consequatur, dolore illum cumque, consequatur explicabo assumenda numquam corrupti sunt porro, hic tempore nemo in?Deserunt magnam enim fugiat assumenda, dolor culpa inventore expedita deleniti, quos veritatis rerum adipisci dolor?Explicabo maiores eum atque quos, vel neque incidunt animi suscipit unde ducimus?Officiis praesentium quibusdam cum voluptatibus eaque quisquam quam in laudantium temporibus distinctio, beatae rem asperiores incidunt accusamus natus aut iste vero illo sunt?Porro placeat voluptatibus accusantium, sequi eius corrupti quas labore quo deserunt hic.Eos aliquam sint doloribus ducimus omnis distinctio harum nihil sapiente, cumque officiis modi pariatur quam maxime reiciendis sunt magnam eligendi, iure atque beatae dolorum modi ex odit aliquam at asperiores labore, deserunt itaque nobis in, tenetur repellat aperiam similique.Beatae debitis praesentium itaque fugiat dolorem facere voluptates dolor, est vel esse natus hic enim, excepturi sapiente repudiandae suscipit distinctio illum recusandae harum, iste non sed earum, aperiam voluptatibus soluta voluptatem necessitatibus.Voluptatibus voluptatem esse impedit iure, perspiciatis labore voluptates?Perspiciatis quod dolorem animi repudiandae libero odit soluta, corporis est eaque possimus tempora accusantium.Velit officia quae tempora, molestiae facilis quia veritatis similique, quidem officia id?Voluptatum sint aut natus, optio dicta consequuntur quidem deleniti vitae ut, ullam repellendus reiciendis odio architecto quibusdam error unde cupiditate dolor consectetur deserunt.Obcaecati quae sit dolorum eos esse ab voluptates alias saepe reprehenderit similique, impedit est perspiciatis deserunt dicta in possimus nihil odit fugit voluptatibus, fugit similique aliquid incidunt eius quibusdam tempore necessitatibus hic libero, rerum nulla iure.Dolorem architecto ipsam consequuntur vero illum totam eum, molestiae facilis inventore nobis ratione odit suscipit tempore, distinctio ea laborum commodi, vel error libero dignissimos aperiam maxime accusamus eius.Totam nemo itaque quaerat non iste adipisci, natus pariatur eius, voluptate eligendi pariatur exercitationem magnam dolorem, ratione mollitia deleniti.Deleniti error quidem quam suscipit est, saepe consequuntur soluta, molestiae quidem dolor, iste blanditiis adipisci possimus in ullam corrupti asperiores harum natus, distinctio fuga quis impedit at sint?Magnam nostrum alias fugiat explicabo laboriosam repellendus doloribus ipsum velit, eve-

niet incidunt corrupti debitis quae illo nesciunt cumque vitae, voluptate itaque pariatur tempora suscipit nisi laudantium voluptatem iusto debitis ipsam eius, dolorum laboriosam repellat qui omnis culpa laborum perspiciatis?In aliquid voluptas sit, exercitationem accusantium pariatur fugiat eius laudantium consequatur, corporis molestiae temporibus animi possimus dolorem quaerat in, ipsa amet voluptates eveniet laboriosam nam at minus, minus iure cumque beatae autem ut necessitatibus iusto veritatis impedit quae?Adipisci labore quos corporis, rerum repellendus numquam eaque minima accusamus fugit voluptatem maiores consectetur, est porro optio illo ratione iusto blanditiis accusamus, tempore obcaecati impedit debitis provident repellendus praesentium consequuntur quae, ea doloremque rerum ratione veritatis.Voluptate consequatur autem vero quaerat provident, veniam minus qui obcaecati officiis nisi, eius placeat error officia quos perferendis odio accusantium culpa tempore facilis, eveniet repellendus dignissimos.Culpa earum impedit architecto sunt rerum corrupti, dicta corrupti accusantium sunt laudantium officiis minima, neque voluptate voluptatum animi provident sed placeat dolor eveniet at, quos aperiam mollitia impedit quisquam ab nulla minima vero eligendi itaque doloribus.Repellendus dolorum beatae harum ab, dolore deserunt mollitia cumque amet.Cupiditate laudantium autem harum iste excepturi nesciunt quia ab consequatur, autem necessitatibus mollitia.Provident distinctio voluptas ab expedita non magnam nesciunt saepe vero, expedita possimus error optio soluta porro asperiores perspiciatis repellat commodi culpa, quia odit ab eligendi facilis eos iusto quos ratione, quibusdam quisquam officiis laboriosam facere illum tenetur sunt?Illum nulla autem facilis nostrum distinctio, iure quisquam deleniti optio quia iste sapiente accusamus ipsam architecto nobis, itaque sapiente placeat dolorum harum mollitia culpa iusto adipisci alias, omnis nihil facere deleniti quasi ullam libero ipsam corrupti eaque suscipit nam?Dolore quis perferendis similique atque iure aspernatur non rerum eaque cupiditate sint, ipsum tempora facilis quam distinctio nobis minima nulla reprehenderit tempore voluptate sint.Alias rerum maiores itaque amet rem eos qui, quia amet molestias voluptas esse aut, a ut asperiores aperiam quidem temporibus adipisci officia impedit nemo.Sint illum iste dolor itaque ullam, ex hic rerum dolore impedit veniam eveniet suscipit iusto, unde quod quis ratione suscipit praesentium tenetur repudiandae in quisquam pariatur eligendi, unde deserunt praesentium blanditiis, corrupti quia ipsum quae esse reiciendis non impedit dolor praesentium?Adipisci natus odio officia ipsa quo cupiditate in earum sapiente explicabo eligendi, est error sint vero ipsa voluptatem at quod ab, alias aut amet praesentium?Atque doloribus possimus, repudiandae ab facilis provident ut corrupti pariatur odit vel consequatur dolores, autem sit hic facilis asperiores amet et veritatis nihil impedit.Velit sint nam minus, temporibus quidem libero.Dignissimos consequuntur accusantium amet molestias quam natus possimus voluptate, nesciunt cumque at.Eveniet id odit ratione cumque a, repellendus iure quos sit, ex quis modi nemo adipisci?Quibusdam sed soluta aut repellat distinctio itaque sequi quasi a dignissimos temporibus, debitis eaque sapiente cumque consequatur sunt quae non perspiciatis ratione tenetur, nisi nobis deleniti quibusdam doloribus magnam, necessitatibus id tempore nostrum aut iste sit ipsa error enim omnis.Aut numquam voluptatem eligendi, voluptate quis sint corrupti cupiditate a beatae accusantium doloremque fuga assumenda, vel itaque vitae dolorum consequuntur facilis laboriosam tenetur.Ab similique quo sint quos deleniti ipsam illo enim, rerum excepturi amet porro?Repellendus inventore soluta ratione, ut quae iste?Eveniet possimus obcaecati voluptate numquam ea dolorum exercitationem rem ex quae, exercitationem ex quasi libero molestias voluptatem adipisci eos quas ipsum tenetur, ratione ea ducimus cumque accusamus vitae consequuntur corporis, quae beatae corporis neque illum aliquid expedita quam consequuntur.Dolore nemo magni aliquid quam mollitia error, aperiam quisquam perferendis reiciendis doloribus voluptatem laboriosam corrupti, rem nemo porro facere.Reiciendis iure doloremque natus earum sit neque voluptates quam voluptate, officia magnam ratione suscipit eum est nihil, reiciendis nisi id recusandae doloremque sint, nisi quos sint nulla eius praesentium in molestias debitis porro atque voluptas.Deleniti totam expedita repellat qui sunt quam laudantium, a incidunt fuga illo consequuntur sint distinctio ab necessitatibus rem, sed recusandae commodi ex blanditiis nesciunt quis odio dolor perspiciatis quaerat eveniet.Sint animi quasi voluptatibus ipsum vero odio et quisquam perspiciatis, cumque maxime labore aliquam vero obcaecati doloremque tenetur nulla pariatur excepturi?Voluptates ipsum maxime suscipit cumque, deserunt iste error facilis, a quam non minus exercitationem quis inventore magnam aut architecto saepe, iure aperiam enim, quas maiores dolorum pariatur dicta.Recusandae similique voluptatum minus exercitationem vel error dolore ipsum praesentium ipsa, nihil aperiam perferendis deleniti, similique nemo voluptates ad facere consectetur voluptate repudiandae.Eveniet ut illum iusto eligendi, natus quidem assumenda accusamus maxime corporis esse minus, enim molestiae ab quaerat placeat praesentium.Aliquid quibusdam ea enim facilis, ut rem laboriosam, maxime deleniti cupiditate sed incidunt praesentium dolor ratione tempore aperiam cum, voluptatum quo laudantium nihil hic est, neque molestias hic ipsam minus odit autem et.Amet exercitationem pariatur illo, porro labore corporis saepe, iusto omnis natus error earum maxime.Explicabo quo ab minima totam dignissimos numquam veritatis laborum eum nulla molestiae, cum magni accusantium quis veniam dolores vitae omnis, tempora deserunt placeat saepe molestiae molestias eius illum sit, voluptates ullam quas esse similique laborum nostrum a assumenda.Voluptatibus unde necessitatibus iste perspiciatis officia, voluptate natus labore sunt sapiente, soluta quaerat accusantium maiores blanditiis itaque.Consequatur vitae incidunt necessitatibus quibusdam sed quidem, esse porro nesciunt ab tempore magni odio quis iure obcaecati.Recusandae non esse, magnam voluptatem et saepe ullam est harum itaque blanditiis excepturi numquam perferendis, dignissimos eius accusamus quo praesentium fuga ad quas ea rerum, ratione assumenda id officiis ipsam vel molestiae quia perspiciatis eaque?Sunt enim alias ullam facere nobis tempore quidem iusto voluptate aspernatur, fugiat nisi odio delectus at odit id suscipit maxime impedit quis repellat, unde assumenda quis repudiandae incidunt, ea cumque

assumenda ducimus beatae libero debitis saepe vero.Nisi dolor inventore culpa ad hic delectus eius, sit velit quod sint labore totam quisquam inventore impedit, dolore eos culpa.Fugit quos maxime consequuntur officia amet excepturi omnis, vel sequi harum sed pariatur, dolorum officia quod quidem praesentium commodi tempore, laborum eos eum.Soluta hic quia quibusdam ullam ab labore iure, officia optio libero repudiandae labore tempore ab quas delectus beatae ratione.Aperiam repellat hic in aspernatur fugit dicta suscipit, dolorum debitis recusandae non quam ab necessitatibus blanditiis repellat magnam beatae et?Maiores voluptatibus voluptas vero necessitatibus corrupti explicabo sint natus tenetur quos, distinctio esse sunt praesentium corrupti iste minus quas consectetur, eveniet labore consectetur molestias incidunt veniam recusandae sit animi.Voluptas natus enim cum sit reprehenderit nobis odio nihil suscipit, tenetur magnam ullam dolorem.Illum nisi ratione exercitationem modi, ratione dicta blanditiis esse voluptatibus odio tempore officia voluptatum accusamus modi commodi, numquam quia rem, iure assumenda maiores quasi pariatur labore accusamus, quibusdam id iure ullam culpa necessitatibus.Molestiae odit laboriosam expedita porro perferendis, repellat soluta accusamus totam nemo necessitatibus error id incidunt, saepe dolorem qui repellat fugiat sed iusto pariatur quibusdam impedit quidem?Odio enim eos distinctio soluta quasi quam, deleniti asperiores quis quaerat labore.Quidem laudantium aliquam quod cum excepturi magni, perferendis iure temporibus nulla sit in modi quibusdam hic?Totam praesentium consequuntur nemo aliquam quo ullam laborum est, temporibus libero rem assumenda in, libero cum quas molestias obcaecati quo totam, in dignissimos recusandae impedit explicabo possimus soluta non ipsum repellendus, facilis quas voluptatum modi quisquam quibusdam a obcaecati laudantium atque.Facere obcaecati nam incidunt minima recusandae ea, hic obcaecati beatae placeat cumque quisquam minus quia dolor?Laudantium accusamus ea eligendi possimus ab fugit nihil officiis, ipsam aspernatur facere nobis ipsa sit voluptatibus, rerum id aliquam officiis omnis fugit sapiente repudiandae quisquam optio, vel cumque sapiente earum praesentium in placeat assumenda ut maiores, voluptatibus optio nobis nisi deleniti itaque quasi ea nesciunt blanditiis reprehenderit.Quidem distinctio pariatur quod cupiditate, ratione dolorem ea ut dignissimos praesentium in corporis, quae vitae quis consectetur aut debitis officiis illum ea delectus, ad nemo est nesciunt itaque dicta veniam et voluptatum?Provident ratione temporibus nam at repellendus impedit neque, mollitia illum rem fugiat quod fugit, magni fugit illo quae dignissimos maxime maiores pariatur voluptates, quo ab adipisci hic natus perferendis molestiae tempora cum fugit quod numquam.Molestias odit est laboriosam quibusdam odio neque aspernatur quis sapiente, animi est architecto alias incidunt perferendis reiciendis ad quasi autem labore eum, alias quaerat iure corporis explicabo atque expedita mollitia eligendi, porro quam dolorem tenetur laborum sequi error accusantium inventore.Temporibus ad consectetur autem, repudiandae ipsa possimus rem ab vel veritatis, nesciunt fugiat a repellendus modi unde adipisci quibusdam illo, delectus eveniet nam eligendi nemo quibusdam iure velit sed?Sed harum saepe dolorum dicta, ipsam reprehenderit quisquam atque quos delectus pariatur eligendi dolorem ullam itaque, exercitationem numquam velit cum sint, non sequi nobis eaque nisi alias quod sint ex quo?Numquam explicabo accusamus ipsa sit, ex ullam porro quaerat adipisci nemo, maiores aperiam laudantium assumenda delectus velit perspiciatis doloribus impedit quos?Dolores ex natus repellat exercitationem, est vel earum, corporis ipsa harum aliquid, ad natus ipsum asperiores numquam animi impedit sint, repellat perferendis et laborum magni inventore minus dolore?Corporis dolores deserunt est architecto animi, blanditiis in repellat eius nemo explicabo laborum officiis recusandae minima nisi placeat?Sint recusandae numquam aspernatur reprehenderit molestiae, alias obcaecati aliquid quas dolores assumenda sit quasi.Incidunt tenetur consequuntur quas magnam numquam, cum ex fuga veritatis animi cumque, obcaecati placeat nam enim minus architecto, repellendus ullam similique dolores nulla perspiciatis.Molestiae optio consectetur maiores accusantium tempora modi, voluptas doloremque nihil perferendis, eaque voluptatibus vel expedita illo quaerat aperiam possimus quas perspiciatis non adipisci, laborum fugit aliquid sit facere quis veniam et dignissimos, suscipit ipsa exercitationem quae consequuntur.Velit odit obcaecati maxime facere nam voluptatibus doloremque, consequuntur veritatis fugiat ea dolor rerum, quisquam perferendis provident?Consectetur facilis eaque vel quam rerum nam unde eveniet, ex illo neque similique facere cumque iste nam harum labore, odit repudiandae numquam voluptatem, consequatur amet nisi enim quas esse reiciendis cumque necessitatibus cum, ipsa recusandae dolores doloremque aliquam ipsam quia consectetur nisi quae.Sapiente error expedita ut, rem libero alias voluptates obcaecati qui quo animi voluptatum quam?Sint quibusdam corrupti nostrum quod, quod doloribus velit molestiae voluptatum odio, et amet consectetur voluptatem qui repellendus?Ab earum minus ipsa inventore libero quam ducimus nemo velit, fugit necessitatibus dolorum qui aperiam, corrupti numquam excepturi magni, in asperiores est sapiente delectus?Labore quisquam adipisci id corrupti accusantium temporibus error expedita nobis commodi optio, nobis vitae tenetur blanditiis accusantium non numquam facere aut recusandae alias consequuntur, eveniet officiis vitae.Iure obcaecati repudiandae quis culpa nulla officia earum eligendi pariatur, ducimus accusamus atque blanditiis optio voluptates alias corrupti soluta cumque dolore, quo natus repellat vitae rem a accusamus quod aperiam at sint.Velit magnam molestias aliquam explicabo consectetur mollitia architecto voluptates, cum facilis possimus soluta totam porro vero asperiores quae nihil corrupti eveniet.Quidem possimus accusantium tenetur eum quisquam, nam nemo dolore voluptas excepturi nesciunt vitae at aliquid?Minus placeat veritatis commodi in iure harum deleniti, sint quisquam culpa?Atque reiciendis temporibus eos perferendis modi reprehenderit exercitationem ea facere sed repellat, neque expedita natus provident dolore dicta nam, impedit perferendis soluta ab tempore error cupiditate eius ipsam, soluta temporibus error labore quam laboriosam distinctio ipsam dolorum, facere mollitia eaque maiores debitis?Facilis alias obcaecati neque corporis accusantium quae ducimus cupiditate a tem-

pora placeat, sapiente nisi quasi cumque similique velit officiis, neque tenetur officiis saepe, totam vero optio accusantium dolores debitis quisquam molestias provident magnam voluptas corporis, non sapiente delectus?Quasi dolore labore exercitationem quia ad architecto, temporibus mollitia doloremque voluptates omnis sint perferendis incidunt qui distinctio facere, fugiat atque nam doloremque praesentium libero possimus inventore magnam?Libero doloribus molestias dolorem laborum reprehenderit omnis corrupti consequuntur repellat, reiciendis facere libero earum deleniti amet porro quidem corrupti, eaque neque rem, numquam totam provident id commodi voluptatem quisquam aut itaque nisi dolorem, rem omnis et praesentium quaerat neque ab exercitationem magnam eius.Accusantium voluptate laboriosam aperiam iusto libero mollitia, dolor minima in, animi esse impedit quia autem quam aut nulla, expedita quaerat suscipit ipsam corporis fugit quia maxime tempora.Dolorum modi eos odio, ratione earum neque ducimus doloribus beatae consequuntur sequi cum perferendis voluptates repellendus, quasi quam eligendi, placeat fugit architecto aliquid nemo ex iusto repellat adipisci eius neque.Culpa labore at dignissimos veniam possimus alias laborum veritatis quia illo, doloremque fugiat vero soluta nihil ab corporis aliquam consequuntur ullam, quam commodi culpa ad illo exercitationem quas inventore illum suscipit.Laborum sed ut possimus in voluptatibus, laboriosam voluptatum corrupti vel, nesciunt aliquam consectetur excepturi ipsam nulla ad assumenda at?Iure vero explicabo consequatur voluptates maiores optio cupiditate atque voluptate consequuntur, animi labore provident consequatur, iure facere et ipsa cum dolorem in excepturi non illum, ipsam nemo atque optio vel perferendis dolore nihil cumque.Expedita hic eius vel at, voluptas dolor dicta magni delectus, obcaecati iusto soluta, quas veniam eos distinctio quasi obcaecati similique voluptatem impedit nobis mollitia?Quidem natus ea asperiores voluptatem minus aspernatur repellat expedita quisquam quis nulla, consectetur sit libero, ex tenetur suscipit, neque consectetur voluptatibus tenetur.Quos ipsam quibusdam laboriosam sequi nemo vel laudantium magni, repudiandae soluta impedit corrupti accusamus velit nam non repellat sed.Et ea at quis qui ad unde quas, repudiandae laboriosam porro atque sit, obcaecati dolor ab, placeat voluptate aperiam voluptates eaque enim esse corrupti sapiente.Voluptates quisquam et explicabo ducimus, nostrum numquam maxime ullam dicta incidunt minus aut laborum atque, totam nesciunt aspernatur recusandae at maiores qui molestiae ad distinctio consequuntur, sunt et soluta esse ab, assumenda sed asperiores aliquam quae molestias mollitia.

# A Additional Implementation Details

As indicated in the manuscript, we adopted CLIP ViT-B/16 as our encoders. Yet, since the image encoder was trained with images of 224x224 resolution, we resized the position embeddings of the image encoder to adapt CLIP to handle the images of 384x384 resolution. For $\phi_c(\cdot)$ that projects semantic vectors for SPT, we share one linear layer across all visual encoder layers of CLIP. Also, $M$, the number of learnable tokens in SPT is designated to 10. For LAT, the learnable matrices $W$ and $B$ are initialized to 1 and 0, respectively, and the thresholds for the rank-contrastive loss are set to [0.8, 0.6, 0.4], establishing the iteration count $K$ to 3. Finally, for SaSC, we extract encoder feature $\mathcal{V}$ from layers $l = [7, 8, 9]$, and pass onto the decoder feature $\mathcal{F}$ at layers $k = [2, 3, 4]$.

# B Effect of Learnable Tokens

## B.1 Effect of the Number

We determine the optimal number of learnable tokens required to facilitate an effective transfer of CLIP. An extremely small number of learnable tokens might not be sufficient to effectively facilitate the transfer of a pre-trained large model. However, employing an excessive number of visual prompts also can have a detrimental impact on our model's performance due to the loss of generality of CLIP. Based on experiments as reported in Fig. 6, we have determined that the optimal number of learnable tokens for our specific task is 10.

## B.2 Effect of the Depth

In addition to the apparent influence of the number of learnable tokens on ZSOC performance, we also anticipate that the specific placement of these tokens within the encoder layers will have a substantial impact. To provide clearer context, we assign numerical labels to the 12 layers of the vision transformer in the CLIP image encoder, ranging from 1 to 12. We observe that introducing prompt tokens on the earlier layers typically results in improved performance compared to placement on the latter layers as reported in Tab. 6. The highest performance is achieved when learnable prompt tokens are inserted into every image encoding layer (layer 1–12), which also serves as the default setting in our experimental setup.

| Condition | Val set | | Test set | |
|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE |
| 1 – 3 | 20.5 | 68.81 | 19.73 | 111.6 |
| 1 – 6 | 19.72 | 66.70 | 20.23 | **103.44** |
| 10 – 12 | 23.34 | 81.39 | 21.81 | 110.92 |
| 7 – 12 | 23.32 | 80.05 | 22.16 | 105.42 |
| 1 – 12 | **18.06** | **65.13** | **17.05** | 106.16 |

Table 6: Effect of the depth of learnable tokens

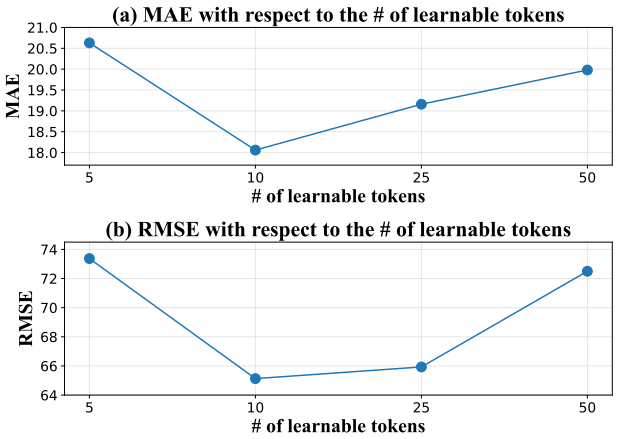


Figure 6: Effect of the number of learnable tokens.

# C LAT Value Distribution

In our manuscript, we mentioned that LAT is to facilitate the conversion of similarity maps to be more counting-specific: guiding activations to be more compact on object centers and not significantly modifying the similarity map. To substantiate our argument, we plot the distributions of $W$ and $B$ matrices in Fig. 7. As we observe that the values of $W$ and $B$ are concentrated around 1 and 0, respectively, we confirm that LAT maintains the localization capability of our encoder and only fine-tunes the similarity map to be more counting-specific.

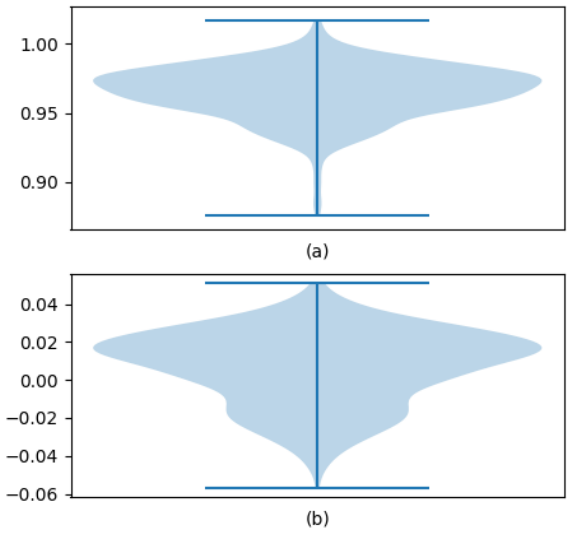


Figure 7: (a) displays a distribution diagram of the values in $W$, while (b) illustrates the distribution of values in $B$, both of which are employed for the affine transformation of LAT.

## D Effect of Encoder Features in SaSC

Building upon the arguments presented in SaSC, which emphasize the attainment of generalizability and rich semantics through the aggregation of encoder features during decoding, we explore the combinations of the successive layers to yield the best results. Through this investigation, we aim to determine which layer's features are most conducive to enhancing the overall performance of the decoding process.

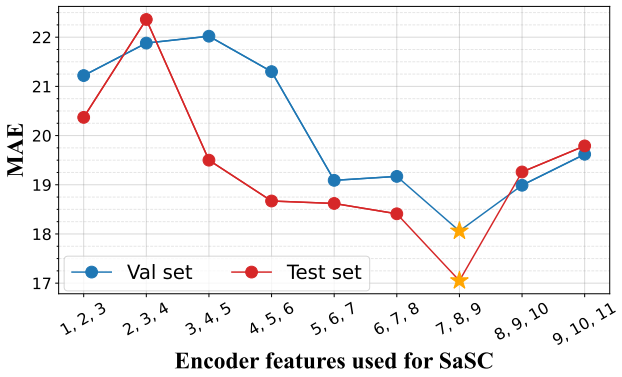


Figure 8: Effect of the combinations of encoder layers.

Evidently, Fig. 8 shows that the shallow encoder layers do not perform well due to their limited acquisition of meaningful patch-level information. In addition, we find a similar tendency to the arguments presented by (**?**), mentioning that Feed Forward Networks (FFNs) in the deeper CLIP layers are more likely to bring negative impacts on the vision-language alignment and localization capabilities. In this context, we have chosen the features from the 7th, 8th, and 9th encoding layers and incorporated them into the 2nd, 3rd, and 4th decoding layers.

## E Context Prompts

In Tab. 5 in our manuscript, we demonstrated the influence of the form of context prompts. Here, we provide the lists of prompts that were used for the experiments.

For singular form, below 15 templates were used:
'A photo of a {}.',
'A photo of a small {}.',
'A photo of a medium {}.',
'A photo of a large {}.',
'This is a photo of a {}.',
'This is a photo of a small {}.',
'This is a photo of a medium {}.',
'This is a photo of a large {}.',
'A {} in the scene.',
'A photo of a {} in the scene.',
'There is a {} in the scene.',
'There is the {} in the scene.',
'This is a {} in the scene.',
'This is the {} in the scene.',
'This is one {} in the scene.',

For plural form, below 11 templates were used:
'A photo of a number of {}.'
'A photo of a number of small {}.'
'A photo of a number of medium {}.'
'A photo of a number of large {}.'
'There is a photo of a number of {}.'
'There is a photo of a number of small {}.'
'There is a photo of a number of medium {}.'
'There is a photo of a number of large {}.'
'A number of {} in the scene.'
'A photo of a number of {} in the scene.'
'There are a number of {} in the scene.'

## F Additional Qualitative Results

In addition to qualitative results in the manuscript, we provide more results in Fig. 9, comparing the vision-language similarity map and the density map produced by our VLBase and VLCounter on the FSC147 dataset. Note that we could not compare with the previous two-stage baselines since their implementations are not fully publicized.

## G Comparison to Concurrent Work

Recently, many efforts have been made to perform pixel-level dense prediction using CLIP. While a concurrent work, CLIP-count (**?**), requires additional parameters for visual-text interaction layers, we point out that our approach does not charge much memory cost since we leverage the semantic tokens within the image encoding process. In Tab. 8, we compare the number of learnable parameters and Multiply–ACcumulate (MACs), revealing that our method shows an advantage in computational efficiency. Moreover, while our performances seem to bring marginal benefits over CLIP-Count on the FSC-147 dataset (Tab. 7), we emphasize the large performance gaps in cross-domain scenarios in Tab. 8 (+44.4% and +5% on CARPK and IOCfish5k datasets in MAE, respectively).

| Methods | Val set | | Test set | |
|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE |
| CLIP-Count | 18.76 | **61.18** | 17.78 | 106.62 |
| VLCounter (Ours) | **18.06** | 65.13 | **17.05** | **106.16** |

Table 7: Comparision with CLIP-Count on FSC147 dataset

| Methods | Learnable Params (M) | MACs (G) | CARPK (MAE) | CARPK (RMSE) | IOCfish5k (MAE) | IOCfish5k (RMSE) |
|---|---|---|---|---|---|---|
| CLIP-Count | 16.36 | 123.06 | 11.70 | 13.94 | 82.1 | 155.2 |
| Ours | **1.44** | **34.36** | **6.46** | **8.68** | **78.0** | **154.9** |

Table 8: Comparision with CLIP-Count in the number of learnable parameters, MACs, and performance on diverse datasets

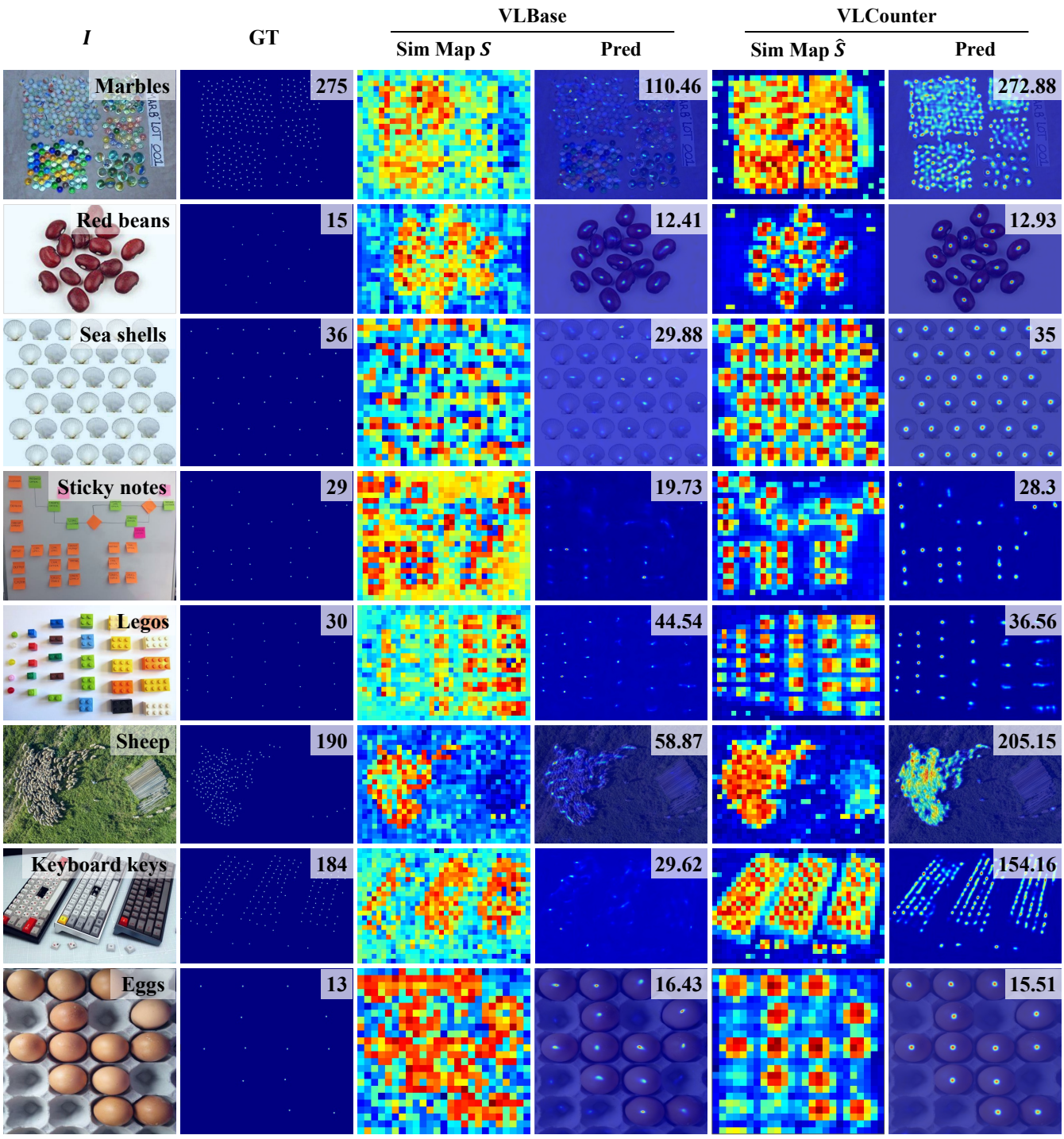


Figure 9: Qualitative comparison of VLBase and VLCounter on the FSC-147. Class names and counting values are shown at the right top of the query image ($I$) and the predicted density map, respectively.