Object-Centric Open-Vocabulary Image Retrieval with Aggregated Features - Supplementary Materials

Hila Levi*[1]
hila.levi@gm.com
Guy Heller*[1]
guy.heller@gm.com
Dan Levi[1]
dan.levi@gm.com
Ethan Fetaya[2]
ethan.fetaya@biu.ac.il

[1] General Motors, RND, Israel
[2] Bar-Ilan University, Israel

I Clustering Methods

In our work, we introduce Cluster-CLIP, a method that represents images using a compact representation by adding an aggregation module on top of Dense-CLIP dense emebeddings (as detailed in Section 3.3 of the main article). The aggregation module first clusters the dense embeddings and then transfers a single representative per cluster. We empirically evaluated various clustering methods within the aggregation module, which are presented in this section, with their results reported in Section II.

K-Means (K.M.). In this method, we perform K-Means clustering on top of each image's dense embeddings. Once clustered, the representatives of an image are the clusters' centroids. We hypothesize that in such a way, each group of semantically similar objects will be represented by their common semantics. An example of such behavior can be seen in Figure 4 in the main article, where several wine glasses are represented by a single cluster, which also scores highest among the different clusters when compared to the embeddings of the phrase "Wine Glass".

Agglomerative Clustering (AG). This method applies Agglomerative clustering, which performs hierarchical clustering using a bottom-up approach: each observation starts in its own cluster, and clusters are successively merged together via a linkage criterion. Different linkage criteria were tested, with and without connectivity constraints; Specifically, Ward linkage, which minimizes the sum of squared differences within all clusters, and Average linkage, which minimizes the average of the distances between all observations of pairs of clusters. Average linkage was tested both with Euclidean and Cosine metrics. We found that using Ward linkage works better, and so we present its results in Section II with connectivity constraints, marked AG-T, and without, marked AG-F.

Region Proposals (R.P.). In this method, we use Segment Anything (SAM) [4] to segment each image. Then, Cluster-CLIP is provided with both the image and the masks, with the masks serving as guidance for clustering the dense embeddings. Formally, given an image of dimension $H \times W$, and the matching Dense-CLIP dense embeddings of dimensions $\frac{H}{32} \times \frac{W}{32} \times C_o$, where C_o is the number of channels at CLIP's output. For each binary mask $m \in R^{H \times W}$ predicted by SAM, we first use max pooling to down-sample the mask to the dense embeddings resolution. Then, we aggregate dense embeddings which coincide with the downsampled mask. Once clusters are formed, each cluster is represented with the mean of its embeddings. To adjust the final number of masks per image, we conducted experiments with various quantities of candidate point-prompts to SAM. Specifically, we ran with 64, 256, and 1024 candidates, which resulted in different numbers of masks per image.

Soft Aggregation via Attention (AT). The clustering algorithms in our work, and K-Means specifically, aggregate each cluster's embeddings by taking the mean over them. Therefore, each cluster representative includes information aggregated only in its cluster (Hard Aggregation). In this method, we suggest weighted aggregation of non-local information, i.e., from all of the image embeddings. This idea is implemented by adapting the attention mechanism described in Section 3.1 in two steps: 1) Clusters are computed on the inputs to the attention layer. 2) The means of the clusters' embeddings (centroids for K-Means) are used as queries in the attention mechanism.

Using the notations from eq. 1 in the main article, this can be formulated as:

$$\begin{aligned} y_j &= out\left(concat\left[y_j^1, y_j^2, \ldots, y_j^M\right]\right) \\ y_j^m &= softmax\left(\frac{q^m(c_j) \cdot k^m(X)^T}{\sqrt{C_q}}\right) v^m(X) \end{aligned} \tag{1}$$

Here c_j and y_j are the mean and soft aggregated representation of the j'th cluster, respectively: $\{c_j \in R^{1 \times C_e}\}_{j=1}^N$, $\{y_j \in R^{1 \times C_o}\}_{j=1}^N$, where N is the number of clusters. This reformulation inherits information from the clustering mechanism (here K-Means) and uses CLIP pretrained query, key, and value weights to essentially create aggregated embeddings with the same output space as CLIP, keeping its zero-shot performance.

Adaptive K-Means (A-K.M.). As different images can contain different numbers of categories, applying K-Means with an adaptive number of clusters per image as a function of the image properties might also be beneficial. A-K.M. uses the Bayesian information criterion (BIC) [9], which is a popular criterion for model selection, in an attempt to choose the best number of clusters per image.

The BIC score of a probabilistic model Q is defined as

$$BIC(Q) = \kappa \ln(n) - 2\ln(\hat{L}) \tag{2}$$

Here, κ is the number of estimated parameters in Q, n is the number of samples observed, and $\hat{L}$ is the model's maximized likelihood function for the observed samples. A lower BIC value is commonly considered better, as it balances the model's complexity (in terms of the number of parameters) and the model fit. To that end, the term $\kappa \ln(n)$ functions as a penalty against utilizing models with a larger number of parameters in order to inflate the likelihood of the model.

To apply a BIC score for K-Means, the method models K-Means with k clusters as a Gaussian Mixture Model (GMM) with k components and spherical covariance. Each GMM component represents a cluster by setting the component's mean to the cluster's centroid and estimating the covariance by the cluster's embeddings.

Using the above definition of BIC score for K-Means, the following algorithm is used to select the best number clusters. Let $k_1, k_2, \ldots k_n$ be a collection of choices for the number of clusters selected apriori, such that $k_i < k_{i+1}$, and mark by BIC_{k_i} the BIC score computed over clusters produced by K-Means with k_i clusters. If $\exists k_i : BIC_{k_i} < BIC_{k_{i+1}}$, then k_i is selected as the number of clusters for the image; otherwise k_n is selected.

Anchors (AN). In this method, the dense representations are clustered according to a spatial division. The resized image is divided into equal-sized squares in multiple resolutions, and the matching embeddings at each resolution are clustered together. The embeddings in each cluster are averaged to create a single representative per cluster.

II Cluster-CLIP Results

This section extends Cluster-CLIP results from Section 4 of the main article by evaluating the additional clustering methods described in Section I, and extending the results for other

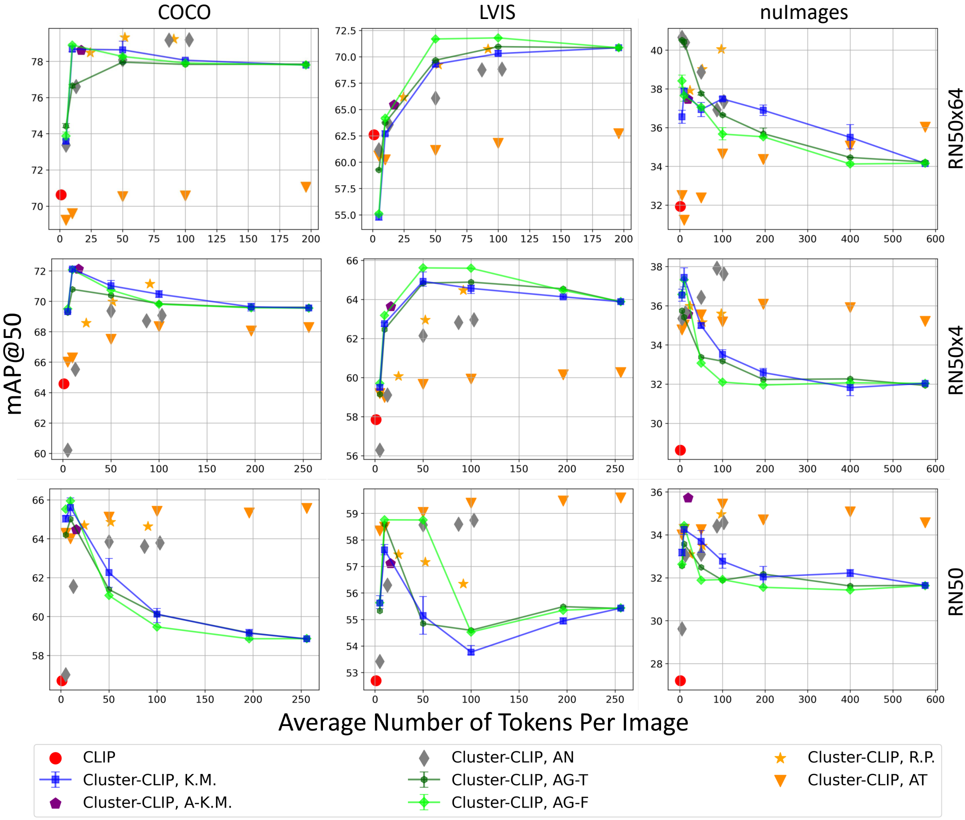

Figure 1: Cluster-CLIP accuracy-efficiency scatter plots for different clustering methods: retrieval accuracy (mAP@50) vs. average numbers of embeddings per image for COCO, LVIS, and nuImages datasets. Top left is better.

ResNet backbones. Figure 1 presents the results in terms of retrieval accuracy (mAP@50) vs. average number of embeddings per image using RN50x64, RN50x4, and RN50 backbones (first, second, and third rows) on COCO [5], LVIS [2], and nuImages [1] datasets (left, middle and right columns). K-Means and Agglomerative Clustering are presented by blue and green solid lines, while other clustering methods are depicted by scatter plots. CLIP is denoted by red circles.

From Figure 1, we can see the effectiveness Cluster-CLIP top-performing clustering methods mentioned in Section 3.3 of the main article. Specifically, K-Means (K.M.), Agglomerative Clustering (AG-T/F), and Region Proposals (R.P) outperform CLIP when using the same backbone architecture across all datasets with merely 5-50 representatives per image, showcasing Cluster-CLIP effectiveness across backbones. When considering the Adaptive K-Means method (purple pentagon), we see that adaptively selecting the number of clusters often scores close to the interpolated score of K-Means with no significant gain. For the Anchors method (gray diamond), it is generally advantageous to partition the embedding space into a greater number of scales or utilize finer divisions, thereby increasing the number of clusters. An exception to this rule arises when using the RN50x64 backbone on nuImages. Compared to other methods, using anchors shows lesser or on-par results, with the only exception being nuImages using RN50x4. Using Attention for soft aggregation of embeddings (orange triangle) is beneficial for RN50 on all datasets and number of clusters; however, it greatly impairs performance for RN50x64.

III Cluster-OwlViT

The Cluster-CLIP architecture is compatible with any dual-encoder VL open-vocabulary model, as elaborated in Section 2 of the main article. This section demonstrates this compatibility by implementing the Cluster-CLIP architecture with OwlViT backbones, referred to as Cluster-OwlViT. To achieve this, we apply the aggregation module outlined in Section 3.3 of the main article on top of OwlViT's dense embeddings.

Figure 2 presents the results of Cluster-OwlViT (represented by gray lines), compared to Cluster-CLIP using different backbones and K-Means clustering. When equipped with

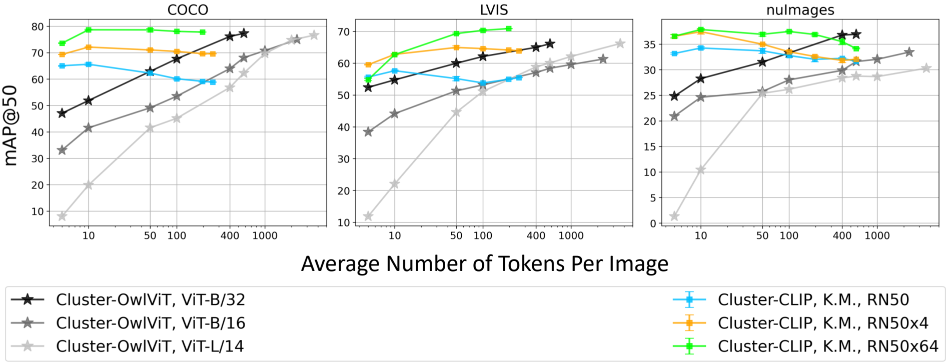

Figure 2: Cluster-OwlViT accuracy-efficiency plots, compared to Cluster-CLIP-K.M.: retrieval accuracy (mAP@50) vs. average numbers of embeddings per image for COCO, LVIS and nuImages datasets. Top left is better.

the ViT-B/32 backbone, Cluster-OwlViT maintains high retrieval rates while reducing the number of clusters by 30% across all three datasets. With the larger ViT-L/14 backbone, Cluster-OwlViT remains competitive while managing to reduce the number of clusters from 3600 to 1000. As Cluster-CLIP outperforms Cluster-OwlViT with fewer representatives, we focus our work on it.

IV Qualitative Examples

For demonstration purposes, and as discussed in Section 4.3 of the main article, we build two image retrieval framework indexes consisting of 120K COCO training set images. The first uses Cluster-CLIP-K.M. with RN50x64 backbone and 10 clusters, while the second uses CLIP with RN50x64 backbone. In both cases, FAISS [3] is utilized to map the embeddings (aggregated embeddings for Cluster-CLIP, global embedding for CLIP) into a large-scale index. Qualitative retrieval examples of interest are presented in Figures 3 and 4.

Figure 3 shows the top retrieval results for 'Helicopter', 'Wall clock', and 'Bulldozer' text queries. Using Cluster-CLIP allows the retrieval of cluttered images with relatively small instances of the requested category. Figure 4 shows top retrieval results for 'Water Tower', 'Globe', 'Passport', 'Earplugs', 'Lemon' and 'Chickpea' text queries, in which Cluster-CLIP produces desired results whereas CLIP prefers larger instances from false categories (sometimes semantically similar). Both figures emphasize the importance of using non-global features for the object-centric image-retrieval task.

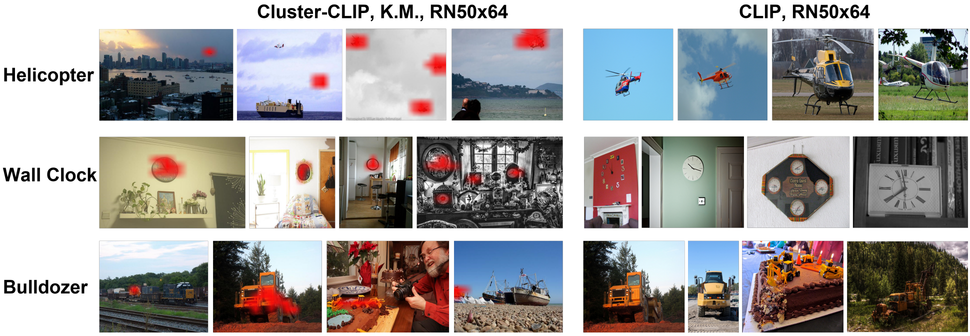

Figure 3: Cluttered images, qualitative examples. Using Cluster-CLIP representation (left part of the figure) allows the retrieval of cluttered images. High-scored patches are emphasized in red. Using global feature (right part) focuses on images with centered objects which occupy a large extent of the images. Best viewed in color while zoomed in.

V Hyperparameters

Dense models. We used the CLIP backbones (RN50, RN50x4, and RN50x64) from the CLIP [8] library and OwlViT framework [6] from the huggingface transformers library [10] with default hyperparameters. Images were resized to a square aspect ratio (for details of the different resolutions, refer to Tables 1 and 2 in the main article), and positional embeddings were interpolated to match the image resolution. For a fair comparison, we ensemble over the embeddings space of the 7 best CLIP prompts [6] in all baselines and experiments that use CLIP or OwlViT text encoders.

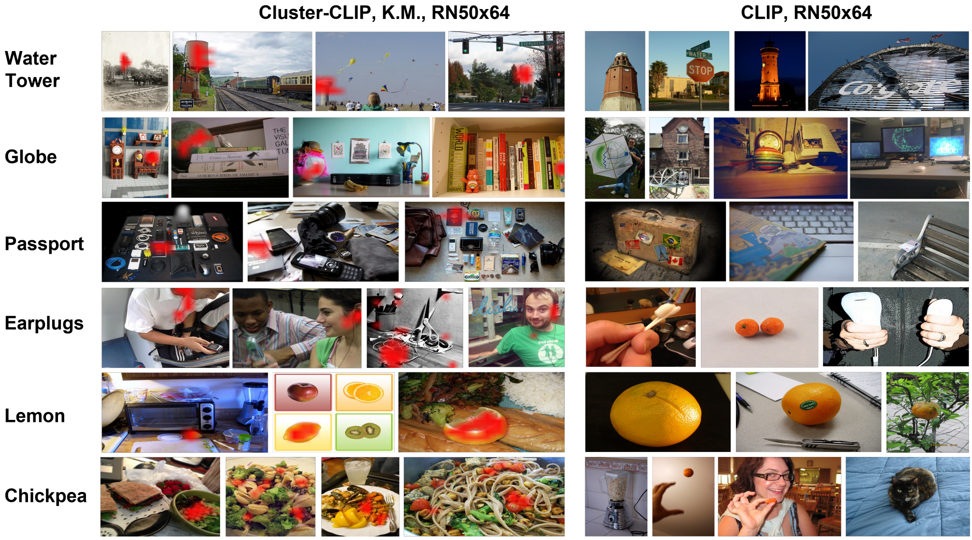

Figure 4: Small objects, qualitative examples. Using Cluster-CLIP representation (left side of the figure) allows the retrieval of images of relevant categories, even if the corresponding instances are relatively small. When global features are used (right side), images with centered and larger objects from semantically similar or even unrelated categories might score higher. Best viewed in color while zoomed in.

Clustering methods. We provide a detailed list of the different hyperparameters used in each of the clustering methods.

- K-Means - We used sklearn library [7] with the following configurations to run the K-Means clustering: *init*=random, *n_init*=10, *max_iter*=300, *tol*=0.0001, *algorithm*=lloyd.

- Region Proposals - We used Segment Anything library [4], using the *vit_h* architecture along with its pre-trained weights, with different number of point-prompts (64, 256, 1024), an IoU threshold of 0.88, stability score threshold of 0.88, stability score offset of 0.1, box NMS threshold of 0.7, and no minimum mask region area nor running separately on crops of the image.

- Agglomerative Clustering - We used sklearn library with the following configurations to run Agglomerative clustering: *linkage*=Ward, *affinity*=Euclidean. Additionally, for Cluster-CLIP, AG-T, we set *connectivity* to be a grid.

- Adaptive K-Means - The method attempts to select best number of clusters out of 5, 10, 15, and 20 clusters.

- Anchors - We evaluated the following different divisions of the embeddings space: (1) 1×1, 2×2 (2) 2×2, 3×3 (3) 3×3, 4×4, 5×5 (4) 2×2, 3×3, 5×5, 7×7 (5) 2×2, 3×3, 4×4, 5×5, 7×7 resulting in 5, 13, 50, 87 and 103 clusters respectively.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11618–11628. Computer Vision Foundation / IEEE, 2020.

[2] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5356–5364. Computer Vision Foundation / IEEE, 2019.

[3] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[5] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.

[6] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, page 728–755. Springer-Verlag, 2022.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[9] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[10] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu,

Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.