Grounding Everything: Emerging Localization Properties in Vision-Language Transformers

Supplementary Material

The supplementary material is organized as follows: We provide a link to a demo in Section A. We then cover additional implementation details and present the rollout of one block in Section B. We further provide additional experimental ablation results in Section C. In Section D, we give more details on the analysis of localization properties and provide additional studies about those properties and Section E provides additional details about the grouping factors. Finally, we close with more qualitative examples and their analysis in Section F.

A. Demo

We provide a HuggingFace demo at: https://huggingface.co/spaces/WalidBouss/GEM

B. Additional Implementation Details

GEM is built in parallel to the vision transformer by processing input features coming from the vision transformer through a series of ensembled iterative-temperature regularized self-self attention. We fix the number of iterations of self-self attention to one for all layers, i.e., we apply one step of self-self attention to the normalized projected features **and** one step of self-self attention to the values using the temperature heuristic as proposed section 3.1. Figure 7 shows the rolled-out processing pipeline for self-self attention with one iteration and ensembled over queue-queue, key-key, and value-value attention. In the first iteration step self-self attention is computed on the respective query, key, or value projection following Equation 5 (main paper), followed by self-self attention of the respective projection applied to the value projection following Equation 6 (main paper). Finally, all three projections are ensembled following Equation 7 (main paper).

C. Additional Ablation

To gain a deeper understanding of the factors influencing the performance of our method, we provide two additional ablations. Namely, we disentangle GEM's performance for the depth of the vision transformer at which we apply self-self-attention and evaluate the effect of adding the MLPs from the vision transformer encoder after the self-self attention in the alternative pathway.

Impact of path length: In Table 7 we evaluate the segmentation performance of GEM applied to CLIP for two model sizes (ViT-B/16 and ViT-B/32) for different starting

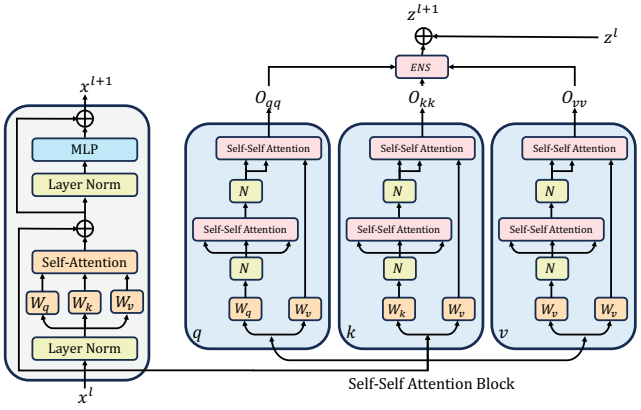

Figure 7. Detailed Illustration of GEM for a number of iterations for the iterative self-self attention equal to 1, where the block N corresponds to L^2 normalization.

layers. We report the mIoU on PascalVOC. For both architectures, the performance remains significantly stable as long as GEM is applied before the last layers with best performance at a depth of three to five layers. We attribute the performance stability to the fact that the skip connections are essentially an exponential moving average applied at each layer. Therefore, the influence on the output features of the first layers decays exponentially. In general, we fix the depth d of GEM to equal to $d = 4$ for all reported experiments.

Impact of MLP: Originally, the studied vision-language models were trained using MLPs in their transformer blocks. While MaskCLIP [8] and CLIPSurgery [17] already showed the negative impact of the MLP, we further assess the influence of these MLPs on the downstream performance for the GEM architecture. Table 7 reports the mIoU on PascalVOC for ViT-B/16 and ViT-B/32 for different depths with and without the MLPs. We can see that adding MLPs have a slight negative effect on the downstream performance. While this is not a significant drop, it still shows that omitting MLPs will in general lead to better results.

Projection impact on different backbones: Table 6 present the results of using different projections using different pretrained ViT-B/16 backbones. We observe that the conclusions drawn section 3 indeed still hold for other pretrained weights than CLIP, i.e., that any projection let it be

VLM	Proj.	VOC	Context
CLIP	v-v	45.14	**32.7**
	k-k	45.9	30.4
	q-q	<u>46.0</u>	31.2
	qkv	**46.2**	<u>32.6</u>
MetaCLIP	v-v	45.7	<u>34.0</u>
	k-k	<u>46.4</u>	32.2
	q-q	44.15	31.6
	qkv	**46.8**	**34.5**
OpenCLIP	v-v	40.7	<u>31.4</u>
	k-k	<u>42.5</u>	30.3
	q-q	40.7	<u>31.4</u>
	qkv	**43.1**	**31.7**

Table 6. Additional evaluation of v-v, k-k, q-q, and qkv for different ViT-B/16 backbones.

v-v, k-k or q-q produces competitive performance within our framework but qkv-ensemble produces overall better performance without the need to choose a specific projection.

D. Analysis of Localization Properties

In Section 4.5 in the main paper, we examine the factors contributing to the localization performance of the proposed method. In the following we provide details on the metrics used, a further discussion of the results as well as an analysis of those characteristics with respect to the depth of the GEM path. We assume that for localization in vision-language models, two essential properties must be fulfilled: *visual distinctiveness*, which refers to the meaningful grouping of visual feature representations, and *vision-language alignment*, which refers to the alignment of these groups with their respective textual descriptions encoded by the language model. In the case of CLIP, vision-language alignment translates to aligning patch tokens with the ViT [CLS] token, as the [CLS] token was trained to correlate with text embeddings through contrastive learning.

D.1. Visual Distinctiveness

For visual distinctiveness, we consider two metrics:

Patch-Patch Similarity. This captures the similarity among patches within each layer. We define an overall path-patch similarity as $S_{pp} = \frac{1}{n(n-1)} \sum_{\substack{i,j \\ i \neq j}} x_i \cdot x_j^T$.

An increase in patch-patch similarity indicates a higher tendency for tokens to share similar characteristics. However, high global path-patch similarity can also indicate that all patch tokens are near-identical, thus reducing localization effectiveness.

Object-Background Contrast. We, therefore, further consider the object-background contrast. A critical characteristic of a model's localization proficiency is the ability to ensure similarity among patch tokens representing the same object while maintaining separation between those representing distinct objects. This characteristic permits the formation of semantically coherent clusters within the embedding space. To this end, we adapt the Michelson contrast to measure the contrast in the similarity between foreground and background patch tokens. For this evaluation, we leverage the segmentation masks of the training set of the PascalVOC dataset [9]. For a given segmentation mask M of an object, we first compute the overall inside-to-inside similarity (noted $S^M_{in,in}$) and inside-to-outside ($S^M_{in,out}$):

$$\begin{aligned} S^M_{in,in} &= \frac{1}{m(m-1)} \sum_{\substack{i,j \in M \\ i \neq j}} \cos(x_i, x_j)^+, \\ S^M_{in,out} &= \frac{1}{m(n-m)} \sum_{\substack{i \in M \\ k \notin \mathcal{M}}} \cos(x_i, x_k)^+ \end{aligned} \tag{9}$$

Here, $m = |M|$ is the area covered by the mask, and the positive part function is employed to clamp negative similarities to zero, *i.e.* $\cdot^+ = \max(0, \cdot)$. The object-background contrast (C^M) for an object mask M is then defined as:

$$C^M = \frac{S^M_{in,in} - S^M_{in,out}}{S^M_{in,in} + S^M_{in,out}} \tag{10}$$

We average across all the masks in the dataset: $MC^M = \frac{1}{|\mathcal{M}|} \sum_{M \in \mathcal{M}} CS^M$, with $|\mathcal{M}|$ being the total number of masks. Note that the ground truth masks are only used for analysis here.

D.2. Vision-Language Alignment

Second, we consider the problem of vision-language alignment. Here, we aim to measure the contrast between the similarity of the text embedding representation of the class and the foreground patch embeddings, compared to the similarity of the text embedding and the background patches.

Text-Object-Background contrast. Let $p \in \mathbb{R}^{n \times d}$ be the patch token outputted by the vision transformer, where n is the number of patches. For a segmentation mask M, the associated class name is denoted as $c(M)$, and we denote $t_{c(M)} \in \mathbb{R}^{1 \times d}$ the text embedding of that class. We compute the overall text-object similarity (noted $TS^M_{txt,obj}$) and text-background similarity ($S^M_{txt,bg}$):

$$\begin{aligned} TS^M_{txt,obj} &= \frac{1}{m} \sum_{i \in M} \cos(t_{c(M)}, p_i)^+, \\ TS^M_{txt,bg} &= \frac{1}{n-m} \sum_{k \notin \mathcal{M}} \cos(t_{c(M)}, p_k)^+ \end{aligned} \tag{11}$$

Backbone	MLP	depth: / layer:	11 L2	10 L3	9 L4	8 L5	7 L6	6 L7	5 L8	4 L9	3 L10	2 L11	1 L12
ViT-B/16	✗		45.1	45.4	45.4	45.5	45.5	45.3	**45.6**	45.5	45.2	43.8	4.8
ViT-B/16	✓		41.6	42.0	41.7	41.6	41.9	42.3	42.2	42.1	**42.4**	38.8	26.2
ViT-B/32	✗		41.0	41.0	41.1	41.2	41.2	41.3	41.4	**41.5**	40.3	26.1	5.1
ViT-B/32	✓		39.7	39.6	39.7	40.0	40.1	40.1	40.2	**40.3**	38.4	21.6	4.3

Table 7. Evaluation of depth and impact of MLP on PascalVOC. We report mIoU performance depending on the depth resp. the starting layer of the self-self attention pipeline. It shows that starting at the middle layers provides best results, but also that higher layers can provide good results. In general, self-self attention without MLP outperforms self-self attention with MLP.

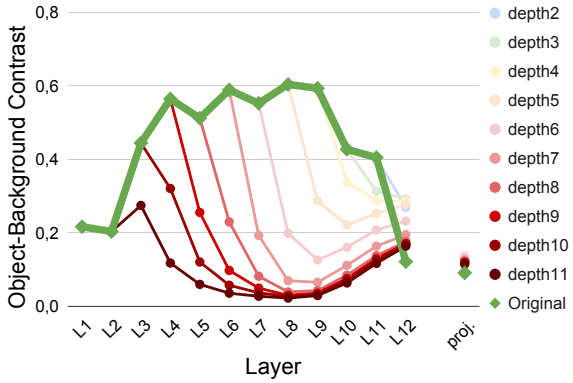

Figure 8. Object-Background contrast of CLIP (original) compared to GEM for different starting depth on PascalVOC for CLIP-ViT-B/16.

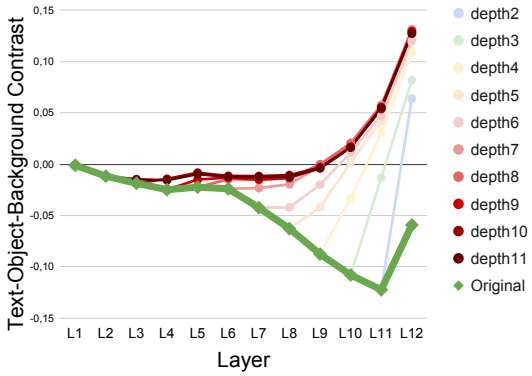

Figure 9. Text-Object-Background contrast of CLIP (original) compared to GEM for different starting depth on PascalVOC for CLIP-ViT-B/16.

The text-object-background contrast for mask M is then defined as: $TC^M = \frac{TS^M_{txt,obj} - TS^M_{txt,bg}}{TS^M_{txt,obj} + TS^M_{txt,bg}}$

This metric is subsequently averaged across all masks in the dataset to derive the global text-object-background contrast $MTC = \frac{1}{|\mathcal{M}|} \sum_{M \in \mathcal{M}} TC^M$.

A higher positive value for MTC signifies that foreground patch embeddings are closer to their corresponding text embeddings than background patch embeddings. A negative value would indicate an inverse relationship.

D.3. Analysis

Figure 5 in the main paper shows the results for the described metrics for CLIP, CLIPSurgery, and GEM for different numbers of iterations. The observed increase in patch-patch similarity from CLIP to CLIPSurgery, in Figure 5a, is due to the clustering induced by the self-self attention. We contribute the slight decrease for GEM to the added normalization and temperature. This is recovered by the higher object-background contrast of GEM over CLIPSurgery and CLIP, pointing to the effective grouping of visual tokens and their ability to distinguish between distinct objects. Further, the analysis of text-object similarity demonstrates improved alignment between visual tokens and text embeddings, enhancing vision-language integration. Notably, CLIP, while exhibiting similar levels of visual distinctiveness in terms of patch-patch similarity and object-background contrast, significantly lags in terms of vision-language alignment, showing a negative text-object contrast, which means that background patches tend to align more closely with object-class text embeddings. This aligns with earlier findings in Li et al. [17] and Mukhoti et al. [26].

We further analyze the impact of GEM with respect to the depth of the self-self attention as well as in comparison to the original model for a CLIP ViT/B-16 model on VOC. We show the object-background contrast (Figure 8) as well as the text-object-background contrast (Figure 9) after each layer as well as for different depths. While the object-background contrast first drops by applying self-self attention, it also shows that it usually recovers after 3-4 layers, while the original CLIP architecture keeps a higher contrast, but significantly drops in the last three layers. Comparing this with the behavior of the text-object-background contrast (Figure 9), we can see that the patch-language alignment of the original CLIP backbone drops significantly after layer six and only recovers in the last layer while the alignment of the self-self attention module consistently increases. Note that the original CLIP backbone always shows a negative text-object contrast, which means that background patches are more closely aligned to the object-class text embedding than the objects themselves while GEM reaches a positive alignment in the last layers.

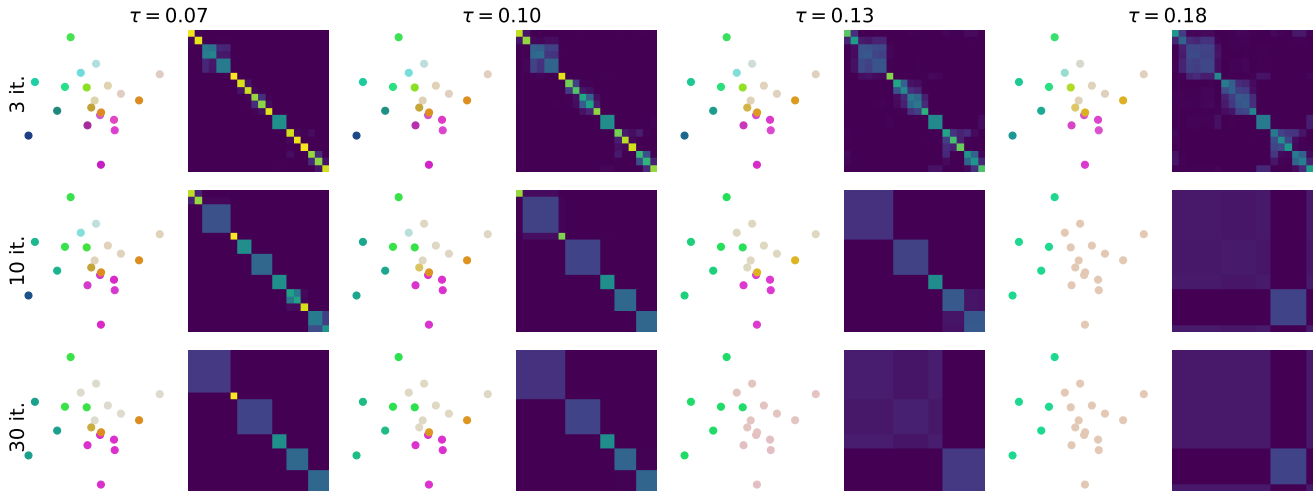

Figure 10. Visualization of self-self attention on a set of 20 vectors: In the top 3 rows, a set of 20 vectors undergoing self-self attention for iterations $K = \{3, 10, 30\}$ and temperatures $\tau = \{0.07, 0.1, 0.13, 0.18\}$. Displayed are the 20 data points (reduced to two dimensions via PCA) and their color represents a smooth cluster membership (the vector into which they are transformed is translated into a color value.) We further show the attention matrix for each configuration (the points were manually ordered for visual simplicity.) It shows that as the number of iterations and/or the temperature increases, self-self attention produces larger fewer clusters.

E. Further Details on Cluster Analysis

Section 3.2 discusses the idea that self-self attention acts as a form of clustering. In Figure 10 we extend the simulation presented in Section 3.2 to more iterations and temperatures. We further add the point-cluster associations reduced to two dimensions via PCA to further visualize the cluster formation. In general, we can observe that increasing the number of iterations (from top to bottom) leads to fewer, larger clusters. The same holds for the temperature parameter where a higher temperature also leads to larger, fewer clusters.

Comparison to Kmeans: We compared our self-self attention method with the Kmeans clustering algorithm. The detailed comparison is shown in Table 8. In this comparison, we looked at both the standard Kmeans and an enhanced version we call Multi-Head Kmeans, where we run separate Kmeans for each attention head to try and get better results. Although the performances of the Kmeans variants are fair, they underperform our self-attention method. Additionally, Kmeans was slower, even when using a GPU-optimized version (`kmeans_pytorch`[2]), and required fine-tuning for each dataset to decide on the number of clusters (K). This comparison highlights the advantages of our method in terms of performance and flexibility. It also shows how our self-self attention approach relates to traditional clustering methods like Kmeans

[2] https://github.com/subhadarship/kmeans_pytorch

Method	K	VOC	Context	Ade20k	V7	fps
Vanilla Kmeans	2	43.0	25.0	8.8	46.0	2.9
	3	28.2	29.8	11.3	46.9	2.7
	5	42.6	30.1	12.9	47.4	2.4
	7	42.3	30.3	13.5	47.4	2.1
	10	41.8	30.3	13.9	47.5	1.6
Multi-Head Kmeans	2	42.9	27.4	9.8	46.9	2.8
	3	44.1	30.0	12.1	47.9	2.7
	5	43.5	31.0	13.7	48.2	2.3
	7	42.7	31.0	14.0	48.3	2.0
	10	42.9	30.8	14.3	48.3	1.5
GEM	-	**46.2**	**32.6**	**15.7**	**50.9**	**37.2**

Table 8. Comparison of GEM with vanilla Kmeans and multi-head Kmeans.

F. Qualitative Analysis

We finally provide additional qualitative results for GEM:

Comparison of vision-language models. Figure 11 compares the localization performance of GEM applied to different vision-language models, namely, CLIP, OpenCLIP, MetaCLIP and BLIP. Overall, MetaCLIP produces sharper and more accurate localization compared to other models. It is also able to better identify objects, e.g., only GEM-MetaCLIP was able to localize "Glove" (Figure 11 Row 2). Compared to that, GEM-BLIP, the only model trained with a multi-objective loss (contrastive, image-text matching and captioning) is still able to localize objects most of the time, but its segmentation mask is less precise.

Comparison to other methods. Figure 12 offers a qualitative comparison between different open-vocabulary segmentation methods. Included in the comparison are methods that use localization information (bounding box or mask) during training, e.g., GroundingSAM and OVSeg, that use a training strategy specifically tailored for segmentation, e.g., GroupViT and SegCLIP, and training-free methods, e.g., MaskCLIP, CLIPSurgery and our method GEM.

Figure 12 shows that methods that were trained with localization information output high-quality masks (see "Cat", "Squirrel" and "Jet Ski") when the object is correctly identified. However, they are not able to detect entities in images that usually don't appear in detection and segmentation datasets. For example, neither GroundingSAM nor OVSeg are able the localize the "Boxer" or the "Violin" in the cartoon (Figure 12 row 8 & 9). This shows the limitation of using handcrafted segmentation annotation during training as they require too much effort to annotate and hence cover a much-restricted scope of entities.

Methods that either fine-tune a pretrained Vision-Language like SegCLIP or train from scratch, are able to accurately segment common objects, e.g., "Cat" (Row 3), "Squirrel" (Row 6) and "Lizard" (Row 4) in Figure 12 – explaining the high performance they get on simply dataset like PascalVOC. However, these methods are unable to segment the rarest entities like the "Jet Ski" (Row 2), "Logo" (Row 7), or even the "Flag" (Row 11). We attribute this lack of diversity to their training strategy that involves the curation of the vocabulary of the used image-text pairs, therefore, reducing the size of the learned vocabulary.

Conversely, training-free methods like MaskCLIP, CLIPSurgery, and GEM benefit from the millions of image-text pairs that vision-language models are trained on, to be able to identify a diverse set of entities. While the segmentation masks of such models are not as sharp as the one outputted by GroundingSAM for example, they are able to localize objects like "Tattoo" (Row 1), "Television" (Row 4) and "Rope" (Row 10) that GroundingSAM is not able to localize. GEM outperforms its training-free counterparts in terms of segmentation sharpness (more defined contours and fewer holes) and is also able to localize objects missed by MaskCLIP and CLIPSurgery,e.g., "Logo" (Row 7).

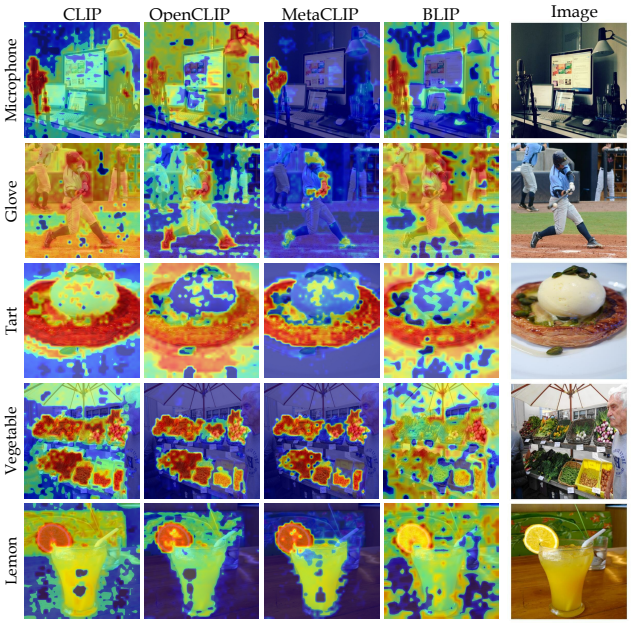

Figure 11. Qualitative comparison of GEM applied to different Vision-Language models.

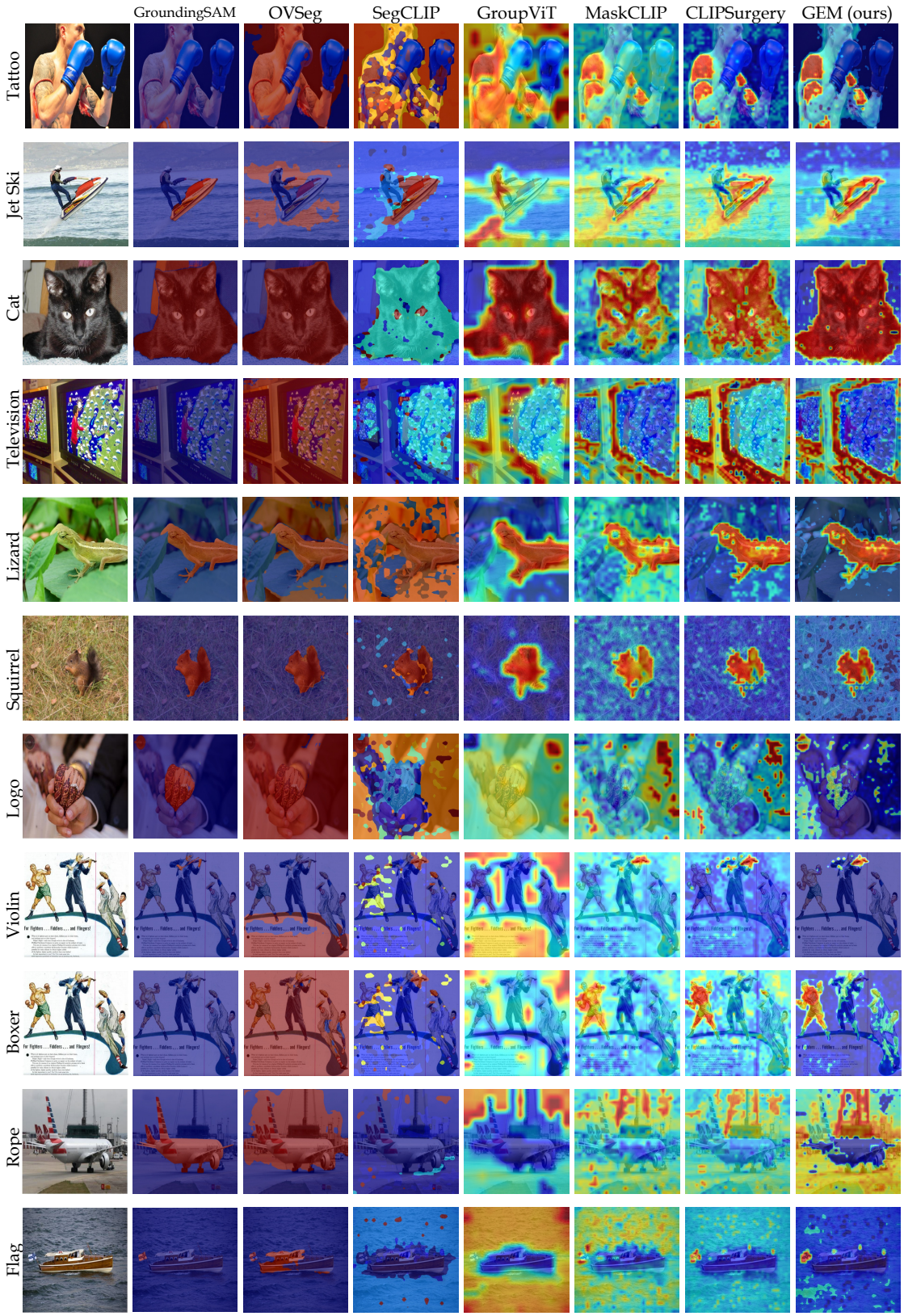

Figure 12. Qualitative comparison between different open-vocabulary segmentation methods, namely, GroundingSAM, OVSeg, SegCLIP, GroupViT, MaskCLIP, CLIPSurgery and GEM.