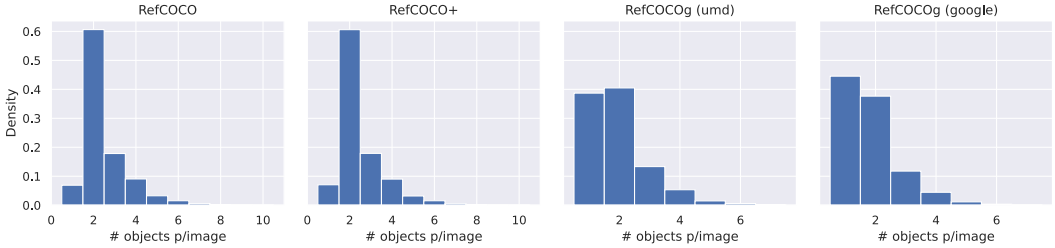

Figure 6: **Object Instances per Image**: histograms of the distribution of the number of object instances referenced in each image within the training sets of the studied datasets.

A Constrained Greedy Matching Algorithm

The details of our constrained greedy matching algorithm are presented in 1. For the sake of simplicity of notation, the image index i is dropped as this matching only occurs for all objects within an image.

Algorithm 1 Constrained Greedy Matching

Input: mask choices $\{\mathbf{m}^c_{j,k}\}_c$, model predictions $\hat{\mathbf{m}}_{j,k}$
Result: greedy matching $\delta^*_{j,k,c}$
$\delta^*_{j,k,c} \leftarrow 0$
$\mathcal{C} \leftarrow \emptyset$ ▷ Initialize the exclusion set
$\mathcal{M} \leftarrow \left\{ j,k,c : \ell_{\text{IoU}}\left(\hat{\mathbf{m}}_{j,k}, \mathbf{m}^c_{j,k}\right) \right\}$ ▷ Compute pseudo-IoU for all mask & prediction pairs
SORT($\mathcal{M}$) ▷ Sort them in descending order
while $\mathcal{M} \neq \emptyset$ **do**
    $j', k', c' \leftarrow$ POP($\mathcal{M}$) ▷ Get the next highest IoU mask choice
    if $c \in \mathcal{C}$ or $j \in \mathcal{C}$ **then** ▷ If either the object or mask has been matched, skip it
        continue
    end if
    for k **do** ▷ Match it for all expressions of the same object
        $\delta^*_{j',k,c'} \leftarrow 1$
    end for
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(j', c')\}$ ▷ Add object & mask to the exclusion set to avoid re-matching them
end while
return $\delta^*_{j,k,c}$

B Number of Object Instances per Image

In Figure 6 we show the distribution of the number of objects per image in the training sets of RefCOCO, RefCOCO+ and RefCOCOg (umd and google splits). Note that this average is higher for RefCOCO and RefCOCO+ than for RefCOCOg. The higher the average number of object instances per image, the more effective we expect the loss from Stage 3 to be in correcting the zero-shot mistakes due to the matching mechanism. This intuition is backed by the experimental results presented in Section 4.1 of the main paper, where the improvement in RefCOCO and RefCOCO+ from S+S+C is higher than the one in RefCOCOg (umd and google partitions).

C Pixel-level contrastive ablation

On top of the constrained greedy matching results, we also experimented with a pixel-dense contrastive loss term. This contrastive term, $\mathcal{L}_{\leftrightarrow}$ has the goal of regularizing the output by explicitly leveraging the insight that references to the same object (positive examples) in an image should have the same output, and that other objects in the same image (negative examples) should have a different output.

Table 5: **Zero-shot and Weakly-Supervised Comparison with $\mathcal{L}_{\leftrightarrow}$**: oIoU (top) and mIoU (bottom) results on benchmark datasets for our zero-shot method (S+S), our full method (S+S+C), the zero-shot bootstrapped models from Stage 3 (ZSBOOTSTRAP), and an ablation with the proposed pixel-dense contrastive loss $\mathcal{L}_{\leftrightarrow}$, along with existing baselines and ablations. The first column refers to the type of method: zero-shot (ZS), weakly-supervised (WS) or fully-supervised (FS). Best zero-shot results are highlighted in **purple**, and the best weakly-supervised ones in **green**. For RefCOCOg, U refers to the UMD partion, and G refers to the Google partion.

		RefCOCO			RefCOCO+			RefCOCOg		
		val	testA	testB	val	testA	testB	val(U)	test(U)	val(G)
oIoU										
ZS	GL CLIP	24.88	23.61	24.66	26.16	24.90	25.83	31.11	30.96	30.69
	GL CLIP (SAM)	24.50	26.00	21.00	26.88	29.95	22.14	28.92	30.41	28.92
	S+S (*Ours*)	**33.31**	**40.35**	**26.14**	**34.84**	**43.16**	**28.22**	**35.71**	**42.10**	**41.70**
WS	TRIS	31.17	32.43	29.56	30.90	30.42	**30.80**	36.00	36.19	36.23
	ZSBOOTSTRAP (*Ours*)	33.61	42.20	26.12	34.13	42.03	26.60	38.27	40.09	37.03
	S+S+C (*Ours*)	50.13	60.70	**43.46**	**40.61**	**49.68**	29.54	**41.96**	**42.59**	42.18
	S+S+C + $\mathcal{L}_{\leftrightarrow}$ (*Ours*)	**50.43**	**61.66**	43.28	39.47	48.97	**30.08**	41.62	42.48	**42.26**
FS	LAVT	72.73	75.82	68.79	62.14	63.38	55.10	61.24	62.09	60.50
mIoU										
ZS	GL CLIP	26.20	24.94	26.56	27.80	25.64	27.84	33.52	33.67	33.61
	GL CLIP (SAM)	30.79	33.08	27.51	32.99	37.17	**29.47**	39.45	40.85	40.66
	S+S (*Ours*)	**36.95**	**43.77**	**27.97**	**37.68**	**46.24**	29.31	**41.41**	**47.18**	**47.57**
WS	TSEG	25.95	–	–	22.26	–	–	23.41	–	–
	Shatter&Gather	34.76	34.58	35.01	28.48	28.60	27.98	–	–	28.87
	ZSBOOTSTRAP (*Ours*)	37.29	44.18	28.43	38.84	46.13	29.60	43.41	44.78	42.72
	S+S+C (*Ours*)	**56.03**	**64.73**	**38.64**	**46.89**	55.45	33.88	48.18	48.61	49.41
	S+S+C + $\mathcal{L}_{\leftrightarrow}$ (*Ours*)	55.46	64.45	38.54	46.56	**55.96**	**34.61**	**48.53**	**48.71**	**49.84**
FS	LAVT	74.46	76.89	70.94	65.81	70.97	59.23	63.34	63.62	63.66

However, note that for negative examples considered pairwise, the output of the network should only be distinct in locations where the instance chosen masks from Equation 3 are active, as it should be 0 elsewhere for both. Given the full matching δ^*, we can obtain the chosen mask for each input as $\mathbf{m}_{i,j,k} = \sum_c \mathbf{m}^c_{i,j,k}\delta^*_{j,k,c}$. Now consider a pair of negative examples for an object j, k and j^-, k^-. The "active" pixels in either of the chosen, pseudo-ground-truth masks as $\mathbf{A}(i,j,k,j^-,k^-) = \mathbf{m}_{i,j,k} \cup \mathbf{m}_{i,j^-,k^-}$. These are then used to slice each of the masks such that $\tilde{\mathbf{m}}_{i,j,k} = \hat{\mathbf{m}}_{i,j,k}|_{A(i,j,k,j^-,k^-)}$ (and similarly for j^-, k^-). We can then write the contrastive loss as:

$$\mathcal{L}_{\leftrightarrow}(i,j,k) = \sum_{k^+ \neq k} \mathrm{KL}(\hat{\mathbf{m}}_{i,j,k} \,\|\, \hat{\mathbf{m}}_{i,j,k+}) + \sum_{j^- \neq j,k^-} [\gamma \mathrm{KL}(\tilde{\mathbf{m}}_{i,j,k} \,\|\, \tilde{\mathbf{m}}_{i,j-,k-})]^{-1}, \tag{4}$$

where $\gamma \in \mathbb{R}^+$ is a hyperparameter to tune the balance of negative and positive examples in the contrastive term.

We present the full results in Table 5. As can be observed, the $\mathcal{L}_{\leftrightarrow}$ loss does not lead to an clear improvement of the results, which is why we have decided to omit it from the main method.