Supplementary Material for
Location-Aware Self-Supervised Transformers for Semantic Segmentation

Contents

A . Implementation and Evaluation Details . . . 1

A.1 LOCA pretraining details 1

A.2 Semantic segmentation datasets . . 1

A.3 Evaluation protocol 2

B . Additional Ablations 2

C . Additional Results 3

C.1 Comparison on 11 semantic segmentation tasks 3

C.2 More localization/classification trade-off results 3

C.3 Scaling Study 3

C.4 Interaction with DINO-v2. 3

D . Visualizations 4

E . Potential Negative Societal Impact 5

This appendix to the main paper provides further details on implementation and evaluation (Sec. A), additional ablation studies (Sec. B), additional quantitative (Sec. C) and qualitative results (Sec. D), and a discussion on potential negative societal impact (Sec. E)

A. Implementation and Evaluation Details

A.1 LOCA pretraining details

We train our models with a base learning rate of 0.001 (linearly ramped up during the first 15 epochs before cosine decay), a batch size of 1024 and a weight decay of 0.1 with adamw optimizer [13]. Models for ablations and analyses are trained during 100 epochs while checkpoints for main results are trained for 600 epochs. 100 epochs of training on 16 TPUv2 accelerators take 29 hours. We use $\eta = 0.8$ for masking. For data augmentation we apply random resized crop, horizontal flipping and color jittering (following the parameters from BYOL [10]). Momentum parameter is set to 0.996 and increased with a cosine schedule to 1 during training [4, 10, 29]. We typically use 10 queries per reference view. We follow MSN pipeline for generating query views [2]. In particular, we restrain the spatial extent of the queries thanks to token dropping. Specifically, one query undergoes random token dropping while the other queries have focal random token dropping. Results are reported with the weights from the momentum branch [4, 29]. We implement LOCA in Jax using the open-sourced SCENIC library [8]. Code and models to reproduce our results will be made publicly available as a SCENIC project.

			Classif.		Loc.		Both	
Method	Data	Sup.	A	P	A	P	A	P
ViT-Base/16								
CLIP [19]	WIT	Text	58.3	67.1	66.4	73.2	45.9	52.8
AugReg [21]	Im21k	Labels	**60.7**	**66.1**	67.4	75.0	48.1	**55.7**
LOCA	Im21k	∅	50.2	63.9	**68.5**	**76.5**	**48.5**	**55.7**
ViT-Large/16								
AugReg [21]	Im21k	Labels	**60.3**	**65.8**	68.0	75.4	50.7	56.5
LOCA	Im21k	∅	51.6	63.3	**71.0**	**78.9**	**52.3**	**60.3**

Table A.1. **Comparison with supervised pretrainings** by disentangling localization and classification on semantic segmentation. We report classification only with a frozen backbone ("Classif.": mAP), localization only ("Loc": mIoU) and semantic segmentation end-to-end finetunings ("Both": mIoU) on ADE20k ("A") and Pascal Context ("P"). Results for ADE20k are also presented in the main paper. LOCA yields excellent locality and good semantic understanding. It is behind supervised image-level pretraining on the pure semantic axis (classification) but better on segmentation ("Both").

A.2 Semantic segmentation datasets

In this paper, we report results on the following diverse semantic segmentation benchmarks: ADE20k [28], Pascal Context ("P.Cont") [16], Pascal VOC ("P.VOC") [9], Cityscapes ("Citys.") [6], Berkeley Deep Drive ("BDD") [26], CamVid [3], India Driving Dataset ("IDD") [25], KITTI [1], SUN-RGB-D ("SUN") [20], ISPRS [14] and SUIM [12]. We detail the main four datasets used in this paper here and refer to corresponding papers and to Mensink *et al.* [15] for details on the remaining other datasets.

ADE20K [28]. It is a dataset containing scenes with fine-grained labels with 150 semantic classes and is one of the most challenging semantic segmentation datasets. The training split is composed of 20,210 images. We report re-

		Consumer			*Driving*					*Indoor*	*Aerial*	*Underwater*	Avg. rel.
Pretraining method	Labels	ADE20k	P.Cont	P.VOC	Citys.	BDD	CamVid	IDD	KITTI	SUN	ISPRS	SUIM	Δ (%)
ImageNet-1k / ViT-Base/16													
Random init.		21.1	19.6	29.1	51.4	40.2	43.3	45.2	39.0	19.7	28.1	53.0	0
DeiT [22, 23]	✓	47.1	–	–	–	–	–	–	–	–	–	–	–
DeiT-III [24]	✓	47.3	53.9	76.1	79.7	62.7	53.8	55.4	47.2	47.5	42.1	73.5	+79.0
DINO [4]		44.1	50.7	74.1	78.4	60.7	51.5	54.3	46.4	44.4	41.5	71.2	+71.9
MoCo-v3 [5]		45.4	51.6	74.5	78.6	60.4	51.1	53.7	45.7	45.6	42.1	72.6	+73.6
iBOT [29]		47.0	54.6	75.0	**79.8**	62.1	51.5	55.5	47.0	46.3	42.2	73.2	+77.7
MAE [11]		45.5	51.7	75.0	79.7	62.1	**57.8**	**55.8**	48.3	45.9	44.6	72.4	+77.8
LOCA (Ours)		**47.9**	**54.9**	**76.7**	**79.8**	**62.8**	56.1	55.6	**48.5**	**47.7**	**45.6**	**74.0**	**+82.1**
ImageNet-21k / ViT-Large/16													
Random init.		21.2	20.1	31.1	44.9	39.7	43.7	45.4	39.7	19.2	26.7	48.3	0
Augreg [21, 22]	✓	50.7	56.5*	77.5	80.7*	62.3	51.2	54.9	47.6	48.5	43.8	73.7	+84.8
LOCA (Ours)		**52.3**	**60.3**	**78.7**	**81.5**	**65.3**	**56.0**	**57.5**	**50.3**	**51.3**	**49.7**	**73.7**	**+93.9**

Table A.2. **Comparison with previous results on 11 semantic segmentation datasets.** We report mean IoU on the validation set of different semantic segmentation benchmarks. Backbones are pretrained using different self-supervised and supervised methods. We consider two settings: (i) pretraining on ImageNet-1k with ViT-Base/16 and (ii) pretraining on ImageNet-21k with ViT-Large/16. We follow the experimental setup of Segmenter [22] for end-to-end finetuning with linear decoder. We report official numbers from [22] when available and run the evaluation from official released checkpoints when not available. We report the average over 5 runs with single-scale mode (*: with multi-scale evaluation). Finally, we report in the last column the relative improvement over starting from random initialization averaged over the 11 datasets ("avg.rel Δ").

sults on the validation set, composed of 2,000 images.

Pascal Context [16]. The training split is composed of 4,998 images with 59 semantic classes and a background class (hence a total of 60 classes). The validation set has 5,105 images.

Pascal VOC [9]. This dataset has a training set of 10,582 images and counts 21 classes (with background class). We report results on the validation set, it has 1,449 images.

Cityscapes [6]. The dataset contains 5,000 images from 50 different cities. We consider the setup with 19 classes as in [22]. There are 2,975 images in the training set, 500 images in the validation set and 1,525 images in the test set (not used). We report results on the validation set.

A.3 Evaluation protocol

We hope to use a simple decoder for semantic segmentation for better investigating the effectiveness of pretraining. We precisely follow the experimental setup of Segmenter [22] for end-to-end finetuning of Vision Transformer with linear decoder. The data augmentation used during training is normalization, random resizing of the image to a ratio between 0.5 and 2.0, photometric jittering and random horizontal flipping. We randomly crop images and use padding to preserve aspect ratio. We use the 512×512 resolution for all datasets and 768×768 on Cityscapes. On ADE20k, we train for 127 epochs with minibatch size of 16 (resulting in 160k iterations). On Pascal, we train for 256 epochs with minibatch size of 16 (resulting in 80k iterations). On Cityscapes, we train for 215 epochs with minibatch size of 8 (resulting in 80k iterations). On all other datasets, we train with minibatch size of 16 and 160k iterations. We use the "poly" learning rate decay schedule and sweep the base learning rate in $\{8e-5, 1e-4, 3e-4, 8e-4\}$ for all of our runs. Weight-decay is kept fixed at 0.01. At evaluation time, we use the sliding-window mechanism with window resolution matching the resolution used during training (i.e. 512×512 for all datasets and 768×768 for Cityscapes) to handle varying image sizes during inference. Table 3 row 6 in Segmenter paper [22] reports 48.06 mIoU (single scale) for finetuning from ViT-B/16 AugReg checkpoint [21]. The average of 3 runs in the same setup in our codebase gives 48.07 mIoU (run 1: 48.41, run 2: 48.08, run 3: 47.70). This validates our reproduction of the linear decoder presented in the Segmenter work [22].

B. Additional Ablations

Table A.3 shows an additional ablation study of several components of our model. In Tab. A.3, we observe that our model is quite robust to the ablation of position embeddings (1) or color jittering augmentation (3). We also see that reducing the number of queries per reference, for example 5 instead of 10, allows to speed up pretraining time by ×1.6 while inducing a loss of only 0.7% in performance (see rows 4 and 2).

	Variant	ADE20k	Δ
	LOCA	46.2	
1	w/o positional embeddings	45.9	– 0.3
2	w/ 5 queries instead of 10	45.5	– 0.7
3	w/o color data augmentation	44.6	– 1.6
4	w/ 2 queries instead of 10	43.4	– 2.8

Table A.3. **Ablation study.** We ablate different component of LOCA, one at a time. All variants are run during 100 epochs and we report their absolute and relative (Δ) performance (mIoU) after finetuning on semantic segmentation.

C. Additional Results

C.1 Comparison on 11 semantic segmentation tasks

In Table A.2, we compare LOCA pre-training to different self-supervised and supervised methods on eleven semantic segmentation benchmarks with diverse properties and domains. The datasets and evaluation protocols are detailed in Sections A.2 and A.3. With ViT-Base/16 architecture and ImageNet-1k dataset, the relative improvement over starting from random initialization averaged over the 11 datasets for LOCA features is +82.1%. This is +4.8 points above the best self-supervised competitor, MAE, and +3.1 points above supervised pretraining with DeiT-3. With ViT-Large/16, LOCA features transfer even better to semantic segmentation. They reach a relative improvement over random initialization of +93.9%, which is 9.1 points higher than the results obtained with AugReg checkpoint [21] in the Segmenter paper [22]. This validates our location-aware pretraining for transferring on semantic segmentation downstream tasks compared to using checkpoints pretrained with a supervised, global task such as AugReg [21].

C.2 More localization/classification trade-off results

Semantic segmentation is the coupling of classification and localization, where these two tasks can have different feature preferences. In this section, we propose to disentangle classification and localization performance on semantic segmentation benchmarks which require both. First, we discard local information and evaluate classification only by training a linear layer with a multi-label binary cross-entropy loss. Second, we evaluate localization only by reporting the performance of an already finetuned semantic segmentation model in presence of a class oracle. Specifically, the oracle replaces the label of each mask by the label of the ground truth mask it has the best IoU with. This evaluation allows to assess the shape and localization of the predictions but not their class.

Comparison with image-level supervised pretrainings. We compare our self-supervised location-aware pretraining to two powerful image-level pretraining paradigms: (i) image classification (i.e. label supervision) as in [21, 27] and (ii) image-text alignment as in CLIP [19]. We present the results by disentangling localization and classification on semantic segmentation. Note that we report classification with a frozen backbone as typically done in self-supervised learning literature (coined as the "linear probing" evaluation protocol). In Table 5 of the main paper, we have reported results only with ADE20k dataset. We show in Table A.1 that observations and conclusions are consistent when considering other datasets, namely Pascal Context and Cityscapes. On Pascal Context, we interestingly observe in Table A.1 that the final performance on semantic segmentation is the same for AugReg and LOCA ViT-B/16 checkpoints pretrained on ImageNet-21k (i.e. 55.7 mIoU). However, this performance can be explained by different factors for the two checkpoints: (i) good classification performance for AugReg (i.e. 66.1 for AugReg *vs* 63.9 for LOCA) and (ii) acute localization performance for LOCA (i.e. 75.0 for AugReg *vs* 76.5 for LOCA).

Comparison with different supervised and self-supervised pretrainings. In Table A.4, we compare the behavior of models pretrained with an image-level versus spatially-aware objective with ViT-B/16 on ImageNet-1k. Unlike previous experiment in Table A.1, we report end-to-end finetuning for classification only in this experiment. Indeed, we have observed that freezing the backbone and training a linear classifier on top of MAE features perform very poorly [11]. In Table A.4, we observe that models pretrained with a global, image-level objective such as DeiT-III or MoCo-v3 tend to be better on the classification aspect. By contrast, models trained with a spatially-aware objective such as MAE or LOCA produce more spatially accurate predictions. Overall, LOCA yields excellent locality and good class-level understanding (while not beating representations learned with label classification pretraining [24] on the pure classification axis). This results in strong semantic segmentations which require both locality and semantic features.

C.3 Scaling Study

We report in Table A.5 (resp. in Table A.6) the numbers corresponding to Figure 6 (left) (resp. (right)) of the main paper. We observe that the performance boost from increasing the pretraining dataset size increases when considering bigger architectures.

C.4 Interaction with DINO-v2.

We find in Table A.7 that DINO-v2 [17] with the ViT-B/14 architecture distilled from a ViT-giant teacher outperforms LOCA ViT-Base/16 (trained on ImageNet-1k only)

	Classification only (mAP)				Localization only (mIoU)				Both (mIoU)			
Method	ADE20k	P. Cont.	P. VOC	Citysc.	ADE20k	P. Cont.	P. VOC	Citysc.	ADE20k	P. Cont.	P. VOC	Citysc.
Image-level pretrainings												
DINO [4]	61.6	67.7	89.9	81.5	64.5	71.6	78.7	79.6	44.1	50.7	74.1	78.4
MoCo-v3 [5]	61.1	69.3	93.6	82.1	66.2	73.7	79.0	79.9	45.4	51.6	74.5	78.6
Supervised (DeiT-III [24])	**64.8**	**71.5**	**94.6**	84.0	66.5	73.6	80.1	80.7	47.3	53.9	76.1	79.7
Spatially-aware pretrainings												
MAE [11]	59.0	67.6	92.8	**84.3**	67.0	74.3	79.9	81.1	45.5	51.7	75.0	79.7
LOCA (Ours)	62.2	69.9	93.7	83.6	**67.9**	**75.4**	**80.5**	**81.4**	**47.9**	**54.9**	**76.7**	**79.8**

Table A.4. **Disentangling localization and classification on semantic segmentation.** We report end-to-end finetuning on classification only (with a multi-label classification loss) and localization only (with an oracle giving the class of the segmentation masks) evaluations on 4 popular semantic segmentation benchmarks: ADE20k [28], Pascal Context ("P.Cont.") [16], Pascal VOC ("P.VOC") [9] and Cityscapes ("City.") [6]. Best number is in bold and second best is underlined. We report performance for different methods pretrained on ImageNet-1k (with or without labels) with ViT-B/16.

Arch / Data	Rand-130k	Rand-1.3M	Full 13M	INet-1k
ViT-Base/16	41.4	46.9	48.5	47.9
ViT-Large/16	39.1	48.5	52.3	49.6

Table A.5. **Scaling in data axis** on ImageNet-21k. We report performance (mean IoU on ADE20k - single scale evaluation) for different pretrained LOCA models. "Rand-*x*" means that we take a random subset of size *x* in ImageNet-21k. 'INet-1k" means that we use the ImageNet-1k dataset only for pretraining.

Data / Arch	Small/16	Base/16	Large/16	Huge/16
ImageNet-1k	44.8	48.0	49.6	48.9
ImageNet-21k	44.8 (+0.0)	48.5 (+0.5)	52.3 (+2.7)	54.3 (+5.4)

Table A.6. **Scaling in model axis** on ImageNet-21 and ImageNet-1k. We report performance (mean IoU on ADE20k - single scale evaluation) for different pretrained LOCA models. The performance boost from increasing the pretraining dataset size increases when considering bigger architectures.

Dataset	DINO-v2-ViT-B/14 (distilled) [17]	LOCA-ViT-B/16
ADE20k	51.8	47.9
BDD	64.8	62.8
CamVid	57.2	56.1
IDD	58.3	55.6
KITTI	50.2	48.5
SUN	49.4	47.7
SUIM	73.2	74.0

Table A.7. **Comparison with DINO-v2.**

on the different considered semantic segmentation benchmarks. Interestingly, we observe that the gap between DINO-v2 and LOCA is higher for datasets which are close to the training data of DINO-v2, such as ADE20k. It is expected since the dataset to train DINO-v2 contains retrieved images specially selected to be similar to ADE20k training images. The gap with DINO-v2 reduces for datasets out of the distribution of training data of DINO-v2 (like CamVid or SUIM for example).

We report a comparison with DINO-v2 for informative purposes, but the reader might need to keep in mind that the comparison is unfair to LOCA for the following reasons: (i) different pretraining datasets (ImageNet-1k only *vs* in-house LVD-142M), (ii) DINO-v2 checkpoints are distilled from a ViT-g teacher with 1.1B parameters, (iii) use of smaller patches (/14 *vs* /16) which usually improves the performance of ViT models on dense tasks [4]. Plus, DINO-v2 is an unpublished, concurrently submitted work. However, we believe discussing the interaction with DINO-v2 is interesting because DINO-v2 training combines an image-level and a patch-level objective, where the latter is akin to iBOT [29]. Hence, given that LOCA outperforms iBOT for dense tasks like semantic segmentation (see Table A.2 for example) a promising direction could be to replace or complement iBOT in the DINO-v2 framework with explicit position prediction as in LOCA.

D. Visualizations

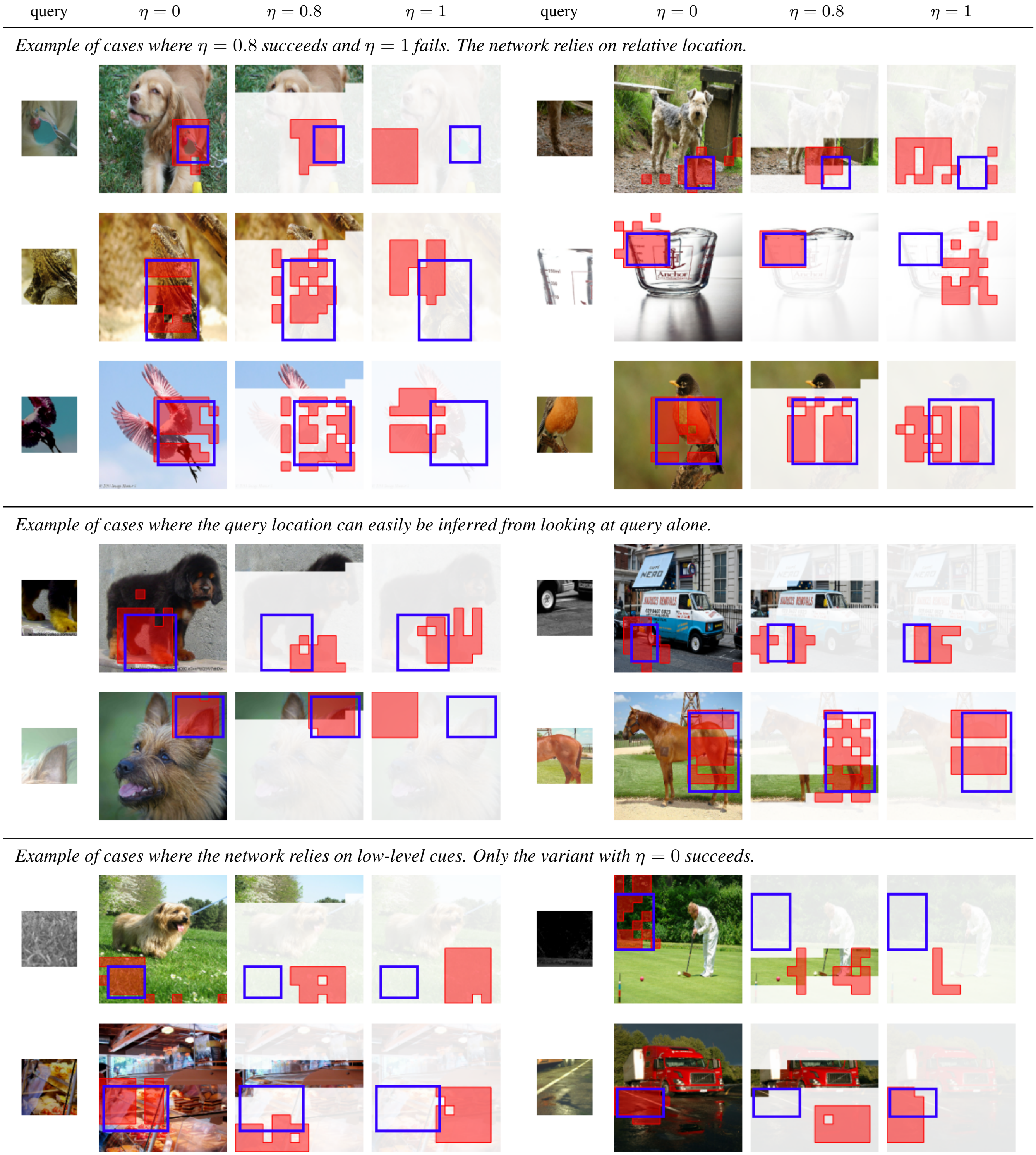
In this section, we visualize the output of the position prediction pretraining task. Specifically, in Figure A.1, we visualize query location prediction for different LOCA models. We compare models pretrained with different masking rates: (i) $\eta = 0$: no masking, the reference is entirely visible to the query; (ii) $\eta = 0.8$: default masking rate, only 40 reference patch tokens are visible to the query; (iii) $\eta = 1$: full masking, the reference is invisible to the query.

In the first rows of Figure A.1, we show examples where the network seems to effectively solve the task by *relative location*. In those cases, we observe that LOCA trained with masking rate $\eta = 0.8$ manages to locate the query based on the patches visible from the reference. For example we see that the network successfully manages to locate the leash joint based on seeing the patch representations of the head of the dog, or to locate the neck of the lizard based on the visible patches of its head. By contrast, the network which does not see the reference at all (i.e. $\eta = 1$) cannot successfully locate the query in those cases. Interestingly, we see that in some cases, this network ($\eta = 1$) can still locate the query by learning where things are typically located in natural images. For example, we observe in Figure A.1 that by recognizing a part such as "ear" it makes a guess that it is more likely to be at the top of the image rather than at the bottom. However, it cannot guess if it is left or right because we apply random horizontal flips between query and reference during training and so this patch is as likely to occur at the right than at the left of the image.

Lastly, we observe that the network trained with full access to the reference ($\eta = 0$) can almost always locate the query. This is because it can rely on low-level cues such as edge consistency or salient points. The last rows of Figure A.1 illustrate this phenomenon.

E. Potential Negative Societal Impact

In this work, we propose to pre-train neural networks on potentially large databases of images only before finetuning them on semantic segmentation tasks. Because we are not using any annotations during the pre-training dataset, this might reduce the biases present in the annotations of the datasets typically used for pre-training [18]. However, the models can still be affected by other sources of biases in the dataset, such as unfair distribution of images or biases against certain populations across the world [7, 18]. We acknowledge these potential caveats, and encourage the community to utilize more fair and responsible data collections.



Example of cases where $\eta = 0.8$ *succeeds and* $\eta = 1$ *fails. The network relies on relative location.*

Example of cases where the query location can easily be inferred from looking at query alone.

Example of cases where the network relies on low-level cues. Only the variant with $\eta = 0$ *succeeds.*

Figure A.1. **Visualizing LOCA's position predictions.** The query location is shown in blue in the reference and LOCA predictions are shown in red. Columns correspond to different reference masking rates and we show only patches visible to the query when it makes its prediction. Displayed images are not seen during training. See discussion in Section D.

References

[1] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 2018. 1

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022. 1

[3] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Patt. Rec. Letters*, 2009. 1

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 2, 4

[5] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 2, 4

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2, 4

[7] Terrance De Vries, Ishan Misra, Changhan Wang, and Laurens Van der Maaten. Does object recognition work for everyone? In *CVPR*, 2019. 5

[8] Mostafa Dehghani, Alexey Gritsenko, Anurag Arnab, Matthias Minderer, and Yi Tay. Scenic: A jax library for computer vision research and beyond. In *CVPR*, 2022. 1

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 2, 4

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 1

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2, 3, 4

[12] Md Jahidul Islam, Chelsey Edge, Yuyang Xiao, Peigen Luo, Muntaqim Mehtaz, Christopher Morse, Sadman Sakib Enan, and Junaed Sattar. Semantic segmentation of underwater imagery: Dataset and benchmark. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 1

[13] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 1

[14] Jochen Meidow, Melanie Pohl, Peter Solbrig, and Peter Wernerus. Theme section “urban object detection and 3d building reconstruction”. *ISPRS journal of photogrammetry and remote sensing*, 2014. 1

[15] Thomas Mensink, Jasper Uijlings, Alina Kuznetsova, Michael Gygli, and Vittorio Ferrari. Factors of influence for transfer learning across diverse appearance domains and task types. *IEEE TPAMI*, 2021. 1

[16] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 1, 2, 4

[17] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3, 4

[18] Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? *arXiv preprint arXiv:2006.16923*, 2020. 5

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. 2021. 1, 3

[20] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 1

[21] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 1, 2, 3

[22] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 2, 3

[23] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *preprint arXiv:2012.12877*, 2020. 2

[24] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022. 2, 3, 4

[25] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*, 2019. 1

[26] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 1

[27] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR, year=2022*. 3

[28] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 1, 4

[29] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 1, 2, 4