Study of the Emerging Properties of Self-Supervised Vision Transformers and Semantic Segmentation

RECVIS 2021: Final Project Report

January 24, 2022

Clément Apavou
MVA Master's student
École Normale Supérieure Paris-Saclay
clement.apavou@ens-paris-saclay.fr

Arthur Zucker
MVA Master's student
École Normale Supérieure Paris-Saclay
arthur.zucker@ens-paris-saclay.fr

Abstract

Self-supervised learning using transformers [4] has shown interesting emerging properties [2] and learn rich embeddings without annotations. Most recently, Barlow Twins [12] proposed an elegant self-supervised learning technique using a ResNet-50 backbone which achieved competitive results when fine-tuned on downstream tasks. In this paper, we propose to study Vision Transformers trained using the Barlow Twins self-supervised method, and compare the results with [2]. We demonstrate the effectiveness of the Barlow Twins method by showing that networks pretrained on the small PASCAL VOC 2012 dataset are able to generalize well while requiring less training and computing power than the DINO method. Finally we propose to leverage self-supervised vision transformers and their semantically rich attention maps for semantic segmentation tasks.

1. Related work and Motivations

The following section introduces key notions and the motivations behind our project.

1.1. Self-supervised learning

In recent years, deep learning models have been increasingly growing, along with the need for far more and denser annotations. This is most apparent in semantic segmentation where the labels are pixel-wise and are thus very costly to produce. Self-supervised learning leverages unlabeled data and particular training methods to learn general features on a pretext task. The various state of the art papers have shown that networks trained in such manner achieve competitive results in Natural Language Processing [3] and Computer Vision [2].

We propose to study self-supervised learning as a pretraining to semantic segmentation tasks. Various self-supervised training methods exist, but current state of the art use two different augmented versions of the same image.

1.2. Barlow Twins and DINO

Barlow Twins [12] is a self-supervised learning method relying on a single neural network trained to produce embeddings from two images. Two random parts of an image are augmented and fed to the network's backbone while the loss is an empirical cross-correlation between the predicted embeddings and the identity matrix. The following augmentations are applied: random cropping, resizing to 224×224, horizontal flipping, color jittering, converting to grayscale, Gaussian blurring, and solarization [12]. These transformations mostly affect the textures and colors of the image, but keep the semantic intact. This formulation of self-supervision is resilient to collapse (a common problem in self-supervised and contrastive learning where the networks outputs constants features).

DINO [2] uses a similar training method, were various local and global augmented crops of the same image are fed to a student and a teacher network. The loss is a cross-correlation between the output of the student and the teacher network. The student updates his weights based on the loss while the teacher relies on an EMA (Exponential Moving Average) of the student's weights. DINO demonstrated that vision transformers' attention heads trained using this technique learned semantically meaningful information which could be assimilated to soft semantic labels. This is especially interesting given the previously mentioned high cost of semantic labels.

Our goal was to reproduce the emerging properties using the elegant and lightweight Barlow Twins method. Previously mentioned methods learn to extract similar features from two different augmented version of the same image, and mostly differ in the objective function. Indeed, they have similar image augmentations which force the network to be invariant to the transformations and thus pay "attention" to what is unchanged and meaningful in the image. Barlow Twins' loss is more resilient to collapse while also being visually interpretable. Our intuition is that although Barlow Twins has access to less images when comparing the extracted features (the official implementation of DINO uses 8 different crops while Barlow twins uses 2), it compensates with more augmentations and a more stable loss. To the best of our knowledge, Barlow Twins has not been trained on vision transformers.

1.3. ViTs for Semantic Segmentation

State of the art ViT based Semantic Segmentation methods [10, 13, 11] proposed various decoder heads to exploit the features extracted by transformers, however, to the best of our knowledge, attention maps have not been used as ad-

ditional features. DINO's [2] learned attention maps provide meaningful global features which gives the attention of each patch w.r.t the other patches. This provides global context and semantic information. We propose to extract and convert the per-patch attention to feature maps and then fuse them with the patch embeddings. We however did not have the time to implement a new decoder head.

1.4. Dataset

We evaluated the quality of the pretrained models on the Pascal VOC segmentation (2012) dataset [5] which contains a small amount of annotated data. It has 20 categories, a small training size of 1,464 and a validation of 1,449 images. The testing set is private so we used the validation as testing as it is done in the literature. We decided to use a small training data in order to demonstrate the generalizing power of the Barlow Twins pretrained networks. Our choice was also influenced by our low computing power, and our time constraints.

2. Contributions

2.1. Github repository

Our GithubRepository contains our implementation for this project. We implemented the global structure and the Barlow Twins method from scratch in PyTorch Lightning, our visualization of the attention maps is inspired from the official DINO [2] repository. Our `trainer` module takes care of initializing the lightning module and the datamodule, both of which can be chosen in our configuration file (`config/hparams.py`). *simple parsing* package extracts and parses the configuration file and allows us to switch between the two tasks: Barlow Twins training and Semantic Segmentation fine-tuning. We used the very practical Weights & Biases (wandb) [1] library to log all of our experiments.

2.2. Visualizations

We implemented two very efficient and easy-to use callbacks to visualize the effective receptive fields and the attention maps at train and validation time. Examples are shown in 1c and 2a. Both rely on pytorch `hooks` and provide more interpretation to the training. Both were implemented from scratch, and the visualization of the effective receptive fields is based on the theory from [8].

We also logged the evolution of the cross-correlation matrix which is fare more interpretable than the value of the loss. As various training showed, a decreasing loss can have a cross-correlation matrix far from the identity. We used a heatmap to represent the empirical cross correlation matrix were values close to 1 are red and values close to zeros are cyan blue. An example of the cross-correlation matrix of a fully converged model can be found in 1a.

Finally we logged the semantic segmentation predictions and various metrics such as *mIoU*, *F1-score* and *accuracy*. The *wandb* experiments can be found here and here.

2.3. Experiments

We provide the global seed used for every experiments, as well as the state of the code corresponding to the run. We used 5 different GPUs, 4 of which are Tesla P100-PCIE-16GB, while the other one is an NVIDIA GeForce RTX 3080. We did not use the google credits as the computing engines do not perform as well, and do not provide the same scalability (drivers have to be manually installed, etc).

3. Experiments and Results

All experiments were run for 400 epochs (unless mentioned otherwise), using AdamW [7], a Cosine Annealing scheduler [6], and the biggest batch size possible given the hardware and the architecture (`tune_batch_size` argument in our code finds the correct batch size). We often used the learning rate finder [9] available in pytorch-lightning, which automatically sets the learning rate.

3.1. Barlow Twins

ResNet : We trained a ResNet-50 (the link takes you to the wandb run with the various logs) to have a baseline, and check our implementation. The obtained cross-correlation matrix can be found in 1c. We trained it with a batch size of 64, which is the minimum batch size to properly train Barlow Twins. The network started to converged but has not finished training. The trained backbone was later used as a baseline in our semantic segmentation experiments (see robust-gorge-60).

ViT : We found that the appropriate projection dimension for the Barlow Twins projection layer was 512. When using 2048 or 1024, the networks took too much time to train and while the loss decreased, the cross-correlation matrix was irrelevant. On top of that, we found out that larger projection dimension lead to sparser diagonal entries (this is also related to the hyperparameters of the loss).

We sweeped through various hyperparameters of the ViT which showed us that a small depth (less than 12) did not allow the emergence of semantic learning as it can be seen in 2a. We were not able to reproduce the emerging properties in most of our runs, this is due to various factors: we used batch size smaller than 64 which slows down and damp the learning process, the image to patch ratio used for Barlow Twins is low. Emerging properties were found in various runs, see 5 for more details.

3.2. Semantic Segmentation

3.2.1. The abilities of Barlow Twins to transfer knowledge

The first experiments was to determine whether Barlow Twins is able to extract relevant features for semantic segmentation downstream tasks. To do so, we used the DeepLabV3 architecture with a various pretrained ResNet-50 backbone (from torchvision models). The DeepLabV3 model was pre-trained on COCO train2017 which contains the same classes as Pascal VOC 2012, but has more data.

In 1, the DeepLabV3 model with a Barlow Twins pretrained (on ImageNet) ResNet-50 from the official repository achieves comparable results to the ResNet-50 pretrained (supervised) for an image classification task on ImageNet. This is important as Barlow twins does not require annotated data for the pretraining. This result shows that Barlow Twins allows the backbone to extract general relevant features, that can be transferred for a semantic segmentation task. Our ResNet-50 pretrained with Barlow Twins on VOC2012 did not reach convergence and thus is not as competitive.

3.2.2. Self-supervised ViTs and semantic heads

Our second experiments was to compared fine-tuned Vision Transformers pretrained using Barlow Twins and DINO. We also evaluated the performance of different semantic segmentation heads using different patch size. **All backbones were freezed** during training to evaluate the generalization power of the self-supervised pretrained backbones. Moreover, the same set of hyperparameters were used in order to remove possible experimental biases. Results can be found in the table 2.

Semantic head: SETR naive uses a per patch MLP to produce dense predictions. Unlike the results presented in [13], it achieved better results than the SETR-PUP head which relies on Progressive Up-samPling. This surprising result can be explained by the smaller number of parameters of the SETR PUP which requires more data to generalize.

Patch size: The patch size heavily influences the performances of the models. Indeed, with the same SETR naive head, a DINO pretrained ViT-S/8 achieves better results than a DINO pretrained ViT-S/16. Admittedly, smaller patches produce denser features which contain more information and allow the network to reconstruct a more fine grained segmentation. However, smaller patch size also requires more memory, and thus requires to use smaller batch size for the same computation.

Pretrain method: The experiments to evaluate the pretraining of Barlow twins was made with ViT-S/16 because, as previously mentioned, using bigger patch size reduced the possible batch size to less than 32, which does not allow the network to converge. We used a Barlow Twins pretrained network which was initialized with DINO weights and showed that it did not performed as well as the original pretrained DINO. However, the attention maps were not destroyed, and the Barlow Twins method was compatible with the learning of meaningful attention.

We can not directly compare ResNet-50 and ViT-S results as the dimension of their embeddings are different, and the up-sampling methods differ.

4. Discussion

In this project, we studied the emerging properties of self-supervised Vision Transformers using the Barlow Twins [12] method. We conducted various experiments in order to obtain semantically meaningful attention maps, but our results show that we lack the computing power necessary to train Barlow Twins with sufficient batch size which prevents the cross-correlation matrix to reach it's desired form. However, we demonstrated (see 5) that when using pretrained DINO weights, the Barlow Twins objective does not unlearns, and is very compatible with the DINO features. We also provided an explanation of why Barlow Twins should also learn semantically meaningful attention maps, and why the lack of computing power is apparent.

By evaluating the generalization power on PASCAL VOC 2012 of the various pretrained backbones, we also studied various semantic segmentation heads. Our experiments show that self-supervised training can be leverage to reduce the number of training samples in a semantic segmentation task as the Barlow Twins ResNet-50 performed comparatively to its supervised pretrained counterpart. While using our pretrained ViT did not yield exploitable results, a backbone which was initialized with DINO weights and trained using Barlow Twins achieved comparable results as the state of the art DINO.

We do not provide experiments with bigger batch size and smaller patch size which would greatly help our study. The limiting factor here is the computing power and the available time, as we were not able to come up with a custom head that would leverage the semantically rich attention maps.

Our approach focuses on the Barlow Twins method on only uses a very small dataset which is prone to over-fitting as we did not get the test set of PASCAL VOC 2012.

References

[1] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. 2

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerg-

ing properties in self-supervised vision transformers, 2021. 1, 2, 4, 5, 6

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 1

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1

[5] Mark Everingham, Luc van Gool, Chris Williams, John Winn, and Andrew Zisserman. Pascal voc 2012 dataset, 2022. Software available from wandb.com. 2

[6] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. 2

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 2

[8] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks, 2017. 2

[9] Leslie N. Smith. Cyclical learning rates for training neural networks, 2017. 2

[10] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation, 2021. 1

[11] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021. 1

[12] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *CoRR*, abs/2103.03230, 2021. 1, 3, 4, 5, 6

[13] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, 2021. 1, 3, 6

5. Additional resources

5.1. Additional information on ViT trainings (batch size problem)

We studied the influence of the batch size, which is coherent with [12]: bigger batch size allow better convergence, and batch size smaller than 64 make the cross-correlation matrix explodes, as it can be seen in the wandb runs gallant-universe-119-ccM and soft-haze-118. Since ViTs' memory consumption depends mostly on the patch size, we were not able to reach convergence with ViTs that had an image to patch size ratio of 16. We also studied the impact of the dimensions of the input size, and settled on using images of size 256 or 224. Smaller images allowed us to have bigger batches and patches, but the obtained attention maps were not as interpretable given their low resolution.

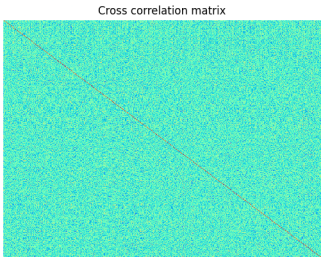
5.2. DINO and Barlow Twins training compatibility

By using Barlow Twins on a ViT with pretrained DINO weights, we demonstrated the similarity and compatibility between the two methods: the initial cross-correlation matrix of the pretrained DINO is very close to the identity (moreover, it has the lowest loss, and the best cross-correlation matrix of all our experiments), and after training for 20 epochs, the validation loss reached it's lowest peak. Then semantic properties of the attention maps displayed in 2b were kept and are still clearly visible. This proves that our intuition is well founded and the emerging properties can be reproduced using Barlow Twins. We only lack the computing power and time to train bigger networks until full convergence. Indeed, DINO [2] was trained for more than 3 days with two 8 GPUs servers, while we only used single GPU training, with 2 workers for 400 epochs. The loss went through a double decent, which means that we over-parameterized the network for the small dataset (which is in fact good). The same ViT trained from scratch did not reach full convergence as it can be seen in the 1b. The emerging properties and the loss did not converge properly due to the small batch size (32, and not 8 as it is reported in the configuration file, because we used set `tune_batch_size = True`).

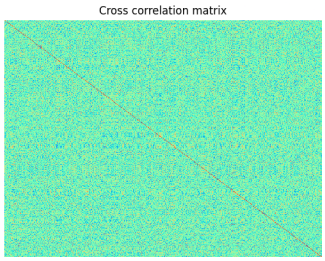
5.3. Code resources

We used the following implementations for the backbones and heads :

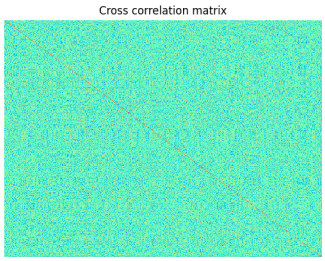
- DeepLabV3 from pytorch vision model
- ViT from pytorch image models
- ViT from vit-pytorch
- SETR adapted from setr-pytorch



(a) Best cross-correlation matrix obtained throughout our experiments: the diagonal is densely field with red dots corresponding to the value 1, and the rest has a cyan color, which corresponds to values close to 0. This corresponds to a ViT small trained for 20 epochs, using DINO [2] weights and Barlow Twins self supervision method and a patch size of 16 (brisk-valley).

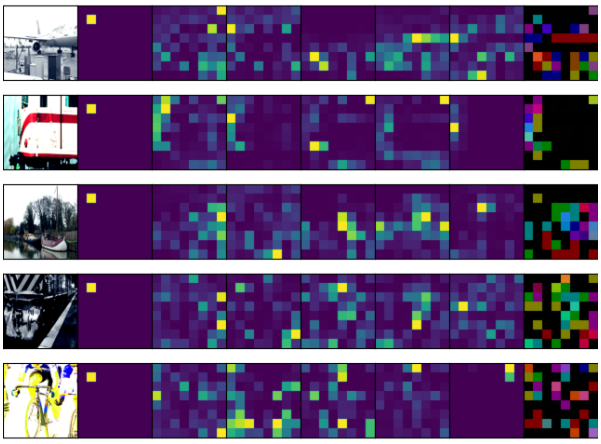


(b) Example of one cross-correlation matrix which starts to converge, but still has a green shade which indicates that it requires more training . This corresponds to the epoch 335 of rare-valley-110 experiment.

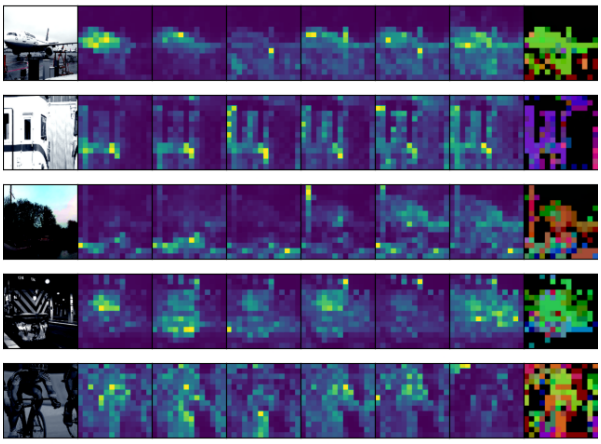


(c) Cross-correlation matrix after 400 epochs of our ResNet-50 trained using BarlowTwins. The projection dimension is set to 512, which gives a very dense diagonal entry (lefy-water).

Figure 1: Cross-correlation matrix obtained from ours experiments.



(a) Attention maps at epoch 400 of the experiment called devout-energy-40. Using a batch size of 256, patch size of 32, the image to patch ratio is $256/32 = 8$. Convergence in loss and cross-correlation matrix is reached, but the attention maps are irrelevant. This comes from the small depth (4).



(b) brisk-valley-111, Barlow Twins on a ViT-S/16 pretrained DINO, this shows that Barlow Twins does not unlearn, and is compatible with learning semantically meaningful attention maps. The cross-correlation is instantly close to the identity matrix, which means that the features extracted by both method share common properties.

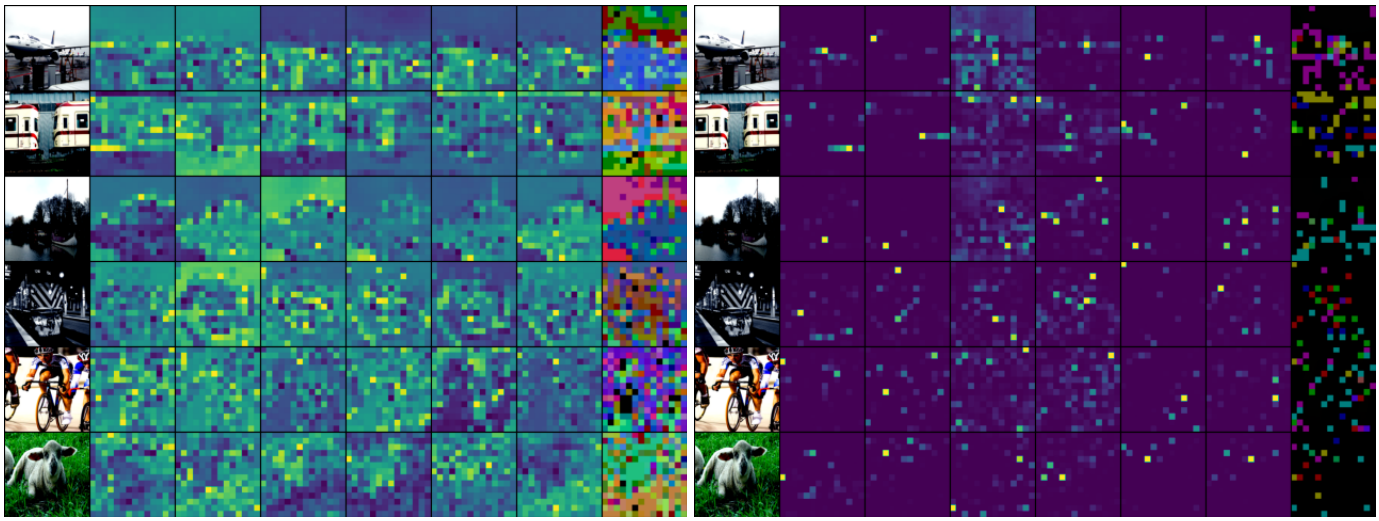
Figure 2: Attention maps of ViTs obtained from ours experiments.

Pretrain method	Pretrain	Weights	mIoU val	Run
Supervised				
Semantic Segmentation	COCO 2017	torchvision	0.6349	rural-valley
Image Classification	ImageNet	torchvision	0.5578	visionary-darkness
Self-supervised				
Barlow Twins [12]	ImageNet	BT repos.	0.5475	sage-cloud
Barlow Twins [12]	VOC 2012	leafy-water	0.1776	robust-gorge

Table 1: Fine tuning results on VOC 2012 using DeepLabV3 (DeepLabHead pretrained on COCO2017) and various pretrained ResNet-50.

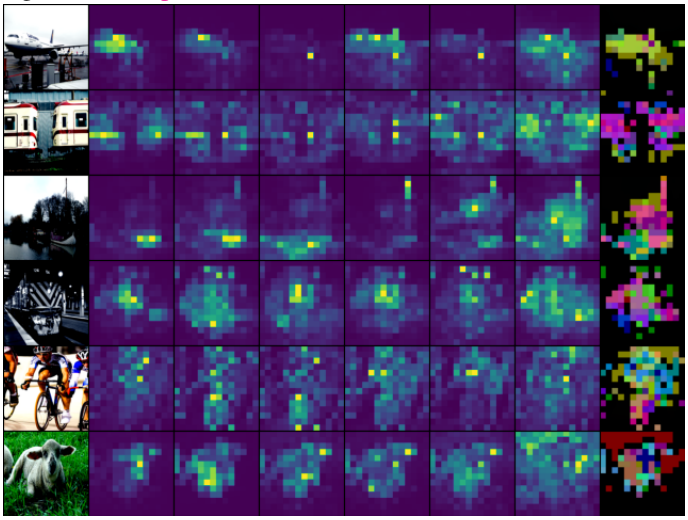
Method	Weights	Arch.	Head	mIoU val	Run
DINO [2]	DINO repo.	ViT-S/8	Linear	0.3955	devout-plant
DINO [2]	DINO repo.	ViT-S/8	SETR-PUP [13]	0.4297	exalted-darkness
DINO [2]	DINO repo.	ViT-S/8	SETR naive [13]	0.5861	misty-feather
Baseline	Random weights	ViT-S/16	SETR naive [13]	0.0845	valiant-puddle
DINO [2]	DINO repo.	ViT-S/16	SETR naive [13]	0.5273	effortless-gorge
Barlow Twins [12]	rare-valley	ViT-S/16	SETR naive [13]	0.1281	quiet-darkness
DINO [2] + Barlow Twins [12]	brisk-valley	ViT-S/16	SETR naive [13]	0.4654	brisk-puddle

Table 2: Semantic Segmentation results using different patch size, pretrain methods and semantic heads. The backbone was freezed and the input size is 224.

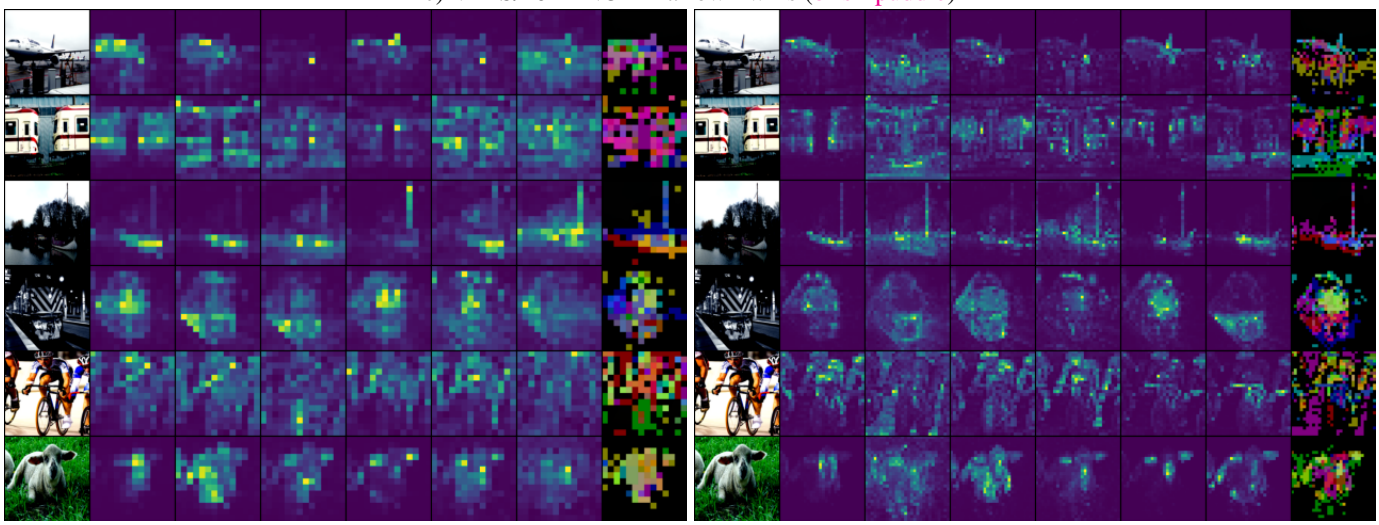


a) ViT-S/16 Random weights (valiant-puddle)

b) ViT-S/16 Barlow Twins (quiet-darkness)



c) ViT-S/16 DINO + Barlow Twins (brisk-puddle)



d) ViT-S/16 DINO (effortless-gorge)

e) ViT-S/8 DINO (misty-feather)

Figure 3: Attention maps of ViTs with different pretraining and patch sizes.

Figure 4: Predictions for a set of 4 images from the validation dataset using 4 different ResNet-50 backbones and a deeplabv3 head.

Figure 5: Predictions for a set of 4 images from the validation dataset using a ViT-S/8 backbones and 4 different heads.

Figure 6: Predictions for a set of 4 images from the validation dataset using different pretrained ViT-S/8 backbones and the naive SETR head.