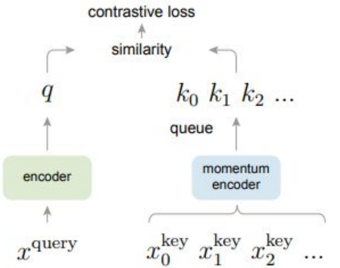
From **Unsupervised Object Localization** to **Open-Vocabulary Semantic Segmentation**

Oriane Siméoni
Meta FAIR
(previously valeo.ai)

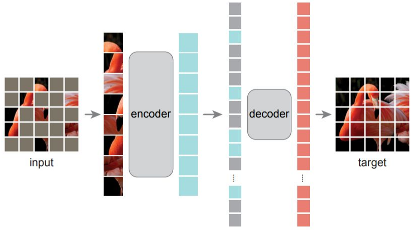
All works presented were done at valeo.ai

Self-Supervised Learning

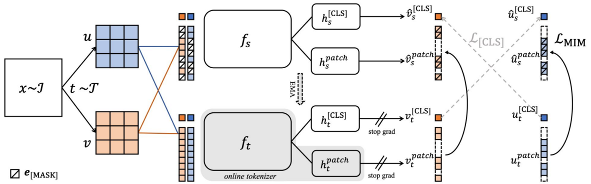
RotNet [Gidaris et al. ICLR’18]



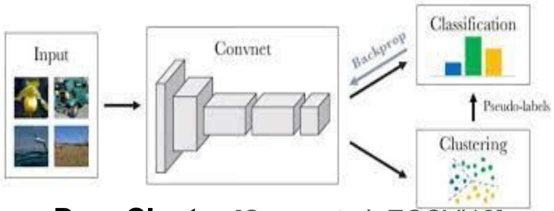
MOCO [He et al. CVPR’20]



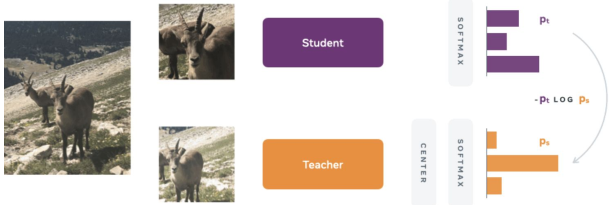
MAE [He et al. CVPR’22]



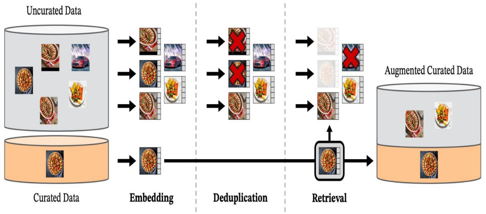
iBOT [Zhou et al. ICLR’22]



DeepCluster [Caron et al. ECCV’18]



DINO [Caron et al. ICCV’21]

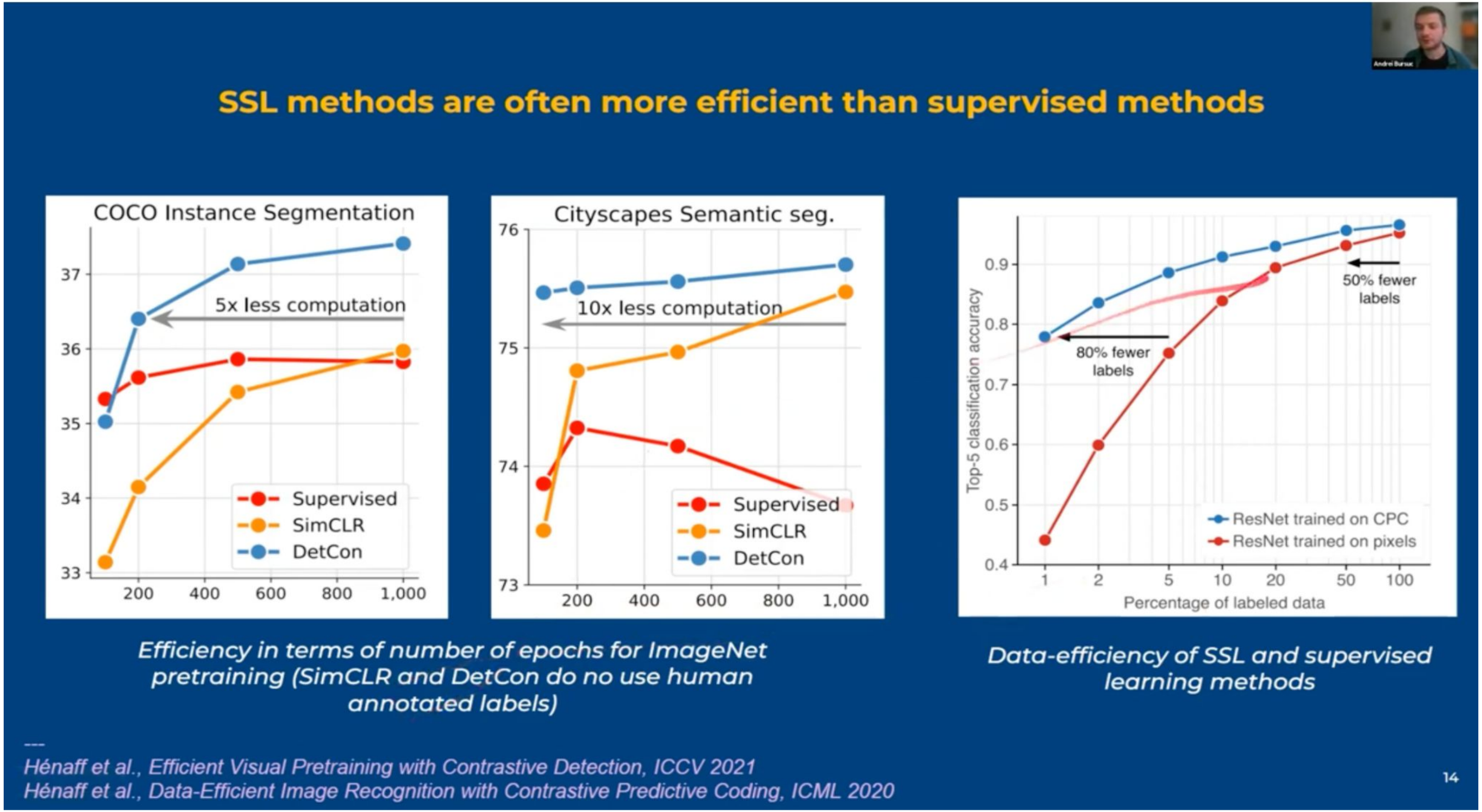


DINOv2 [Oquab et al. TMLR’23]

Learn image features **with no human-made annotation** using a **proxy task**

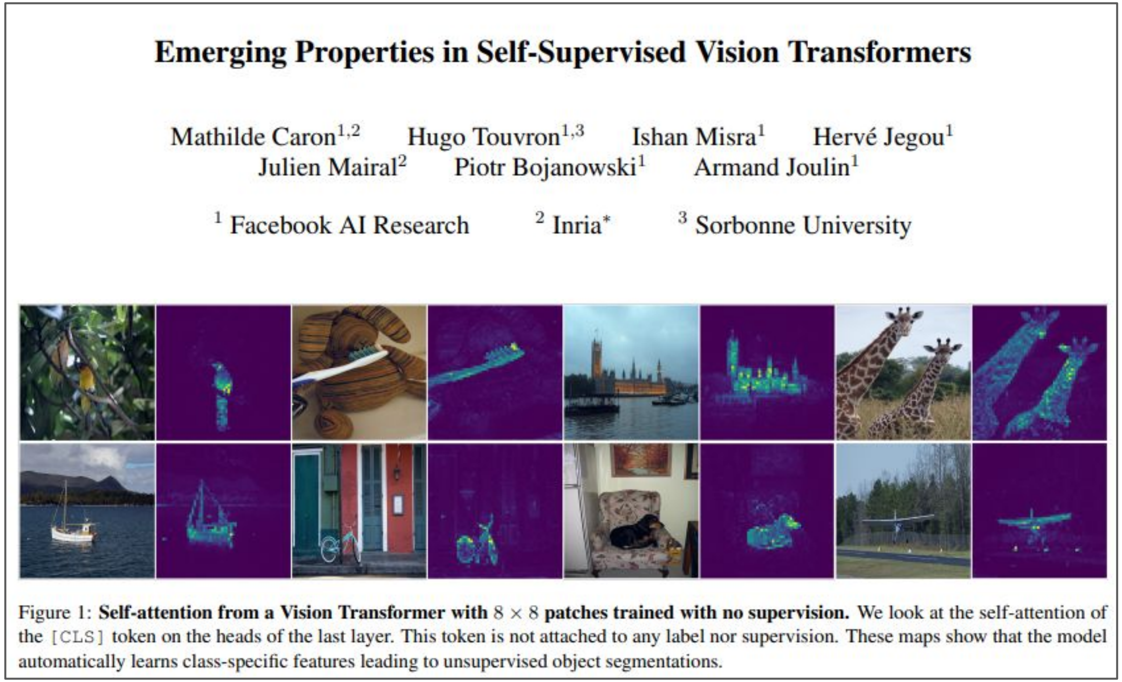
Self-supervised learning is great for **pre-training**

Stolen from Andrei Bursuc from *ECCV'22 Tutorial: Self-Supervision on Wheels*



But not only

[CLS] self-attention maps



Emerging Properties in Self-Supervised Vision Transformers

Mathilde Caron[1,2] Hugo Touvron[1,3] Ishan Misra[1] Hervé Jegou[1]
Julien Mairal[2] Piotr Bojanowski[1] Armand Joulin[1]

[1] Facebook AI Research [2] Inria* [3] Sorbonne University

Figure 1: **Self-attention from a Vision Transformer with 8×8 patches trained with no supervision.** We look at the self-attention of the [CLS] token on the heads of the last layer. This token is not attached to any label nor supervision. These maps show that the model automatically learns class-specific features leading to unsupervised object segmentations.

DINO [Caron et al. ICCV’21]

- They have good localization properties
- Suffer fewer shortcuts than their fully-supervised counterparts

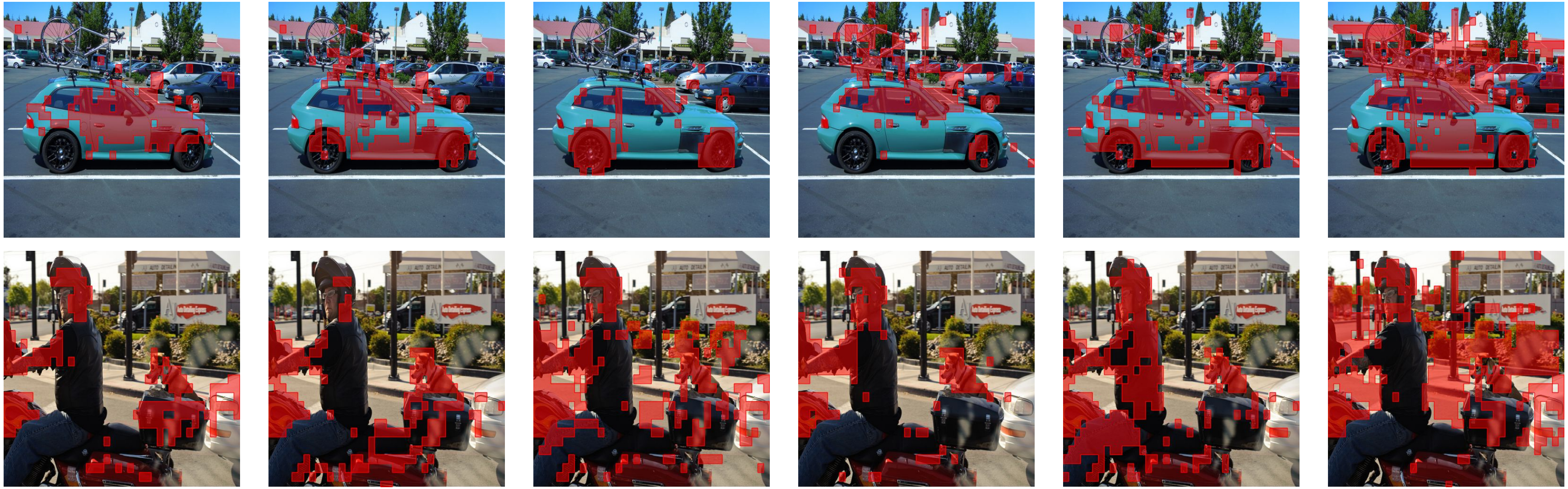
From **Unsupervised Object Localization** to **Open-Vocabulary Semantic Segmentation**

From **Unsupervised Object Localization** to **Open-Vocabulary Semantic Segmentation**

Self-attention maps

- The **6 heads** attend to **different parts** of an image
- Without supervision hard to distinguish **what is important** and is an object

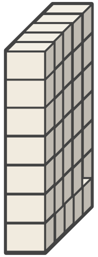
[CLS] self-attention maps



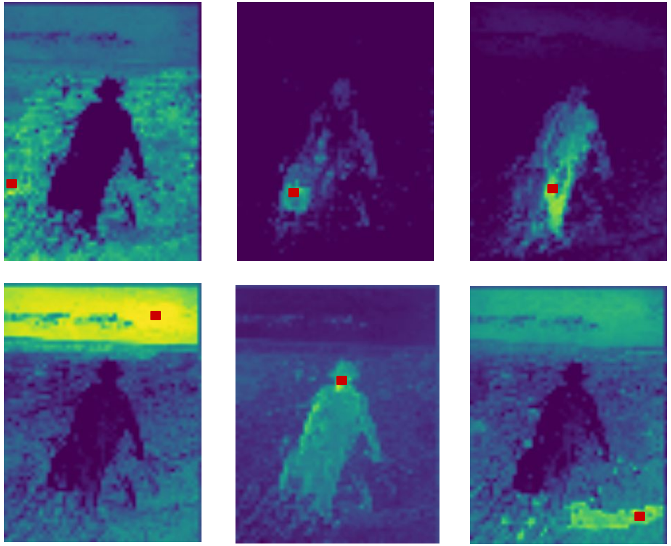
Head 1 | Head 2 | Head 3 | Head 4 | Head 5 | Head 6

Object localization in SSL similarity graph

SSL backbone



patch features
(here the keys of the last layer of DINO)



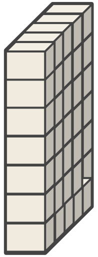
Patch **correlations** to seed

Observations

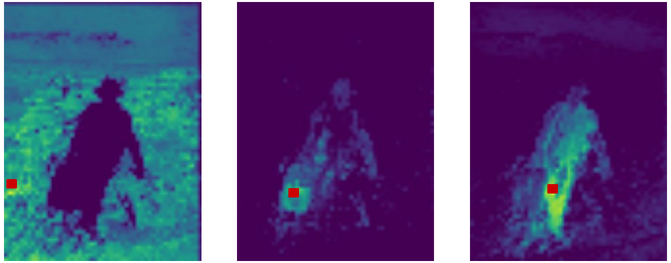
- Features correlate semantically

Object localization in SSL similarity graph

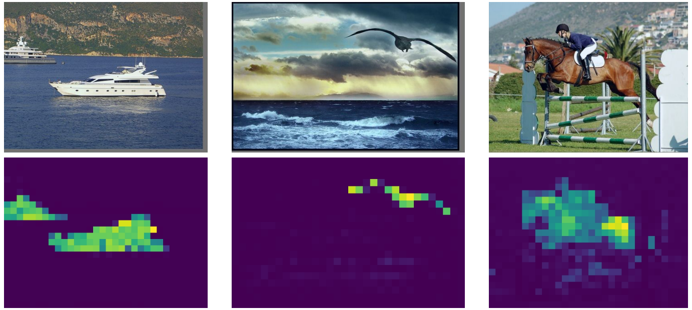
SSL backbone



patch features
(here the keys of the last layer of DINO)



Patch **correlations** to seed

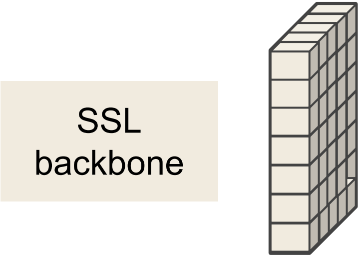


Patch **degree**
low (yellow) to high (blue)

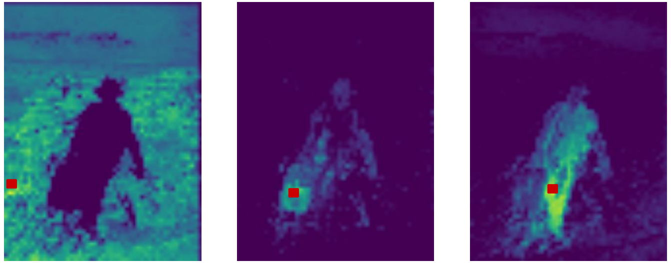
Observations

- Features correlate semantically
- When compute a binary similarity graph
  (nodes connected if cosine similarity >0)
  - **object patches are less connected than background**

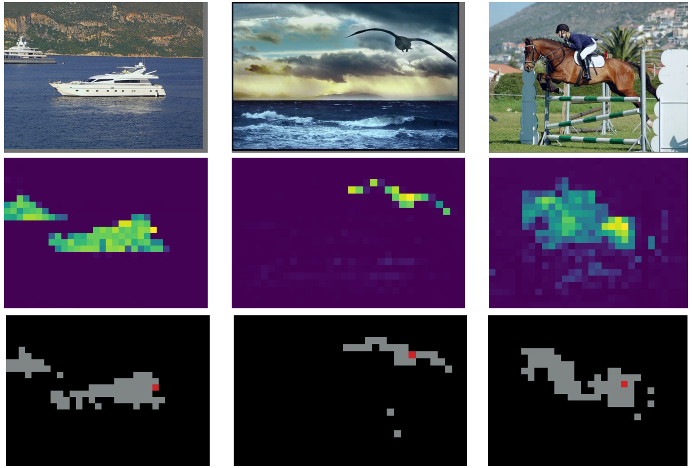
That's basically **LOST** [Siméoni et al., BMVC'21]



patch features
(here the keys of the last layer of DINO)



Patch **correlations** to seed



Patch **degree**
low (yellow) to high (blue)

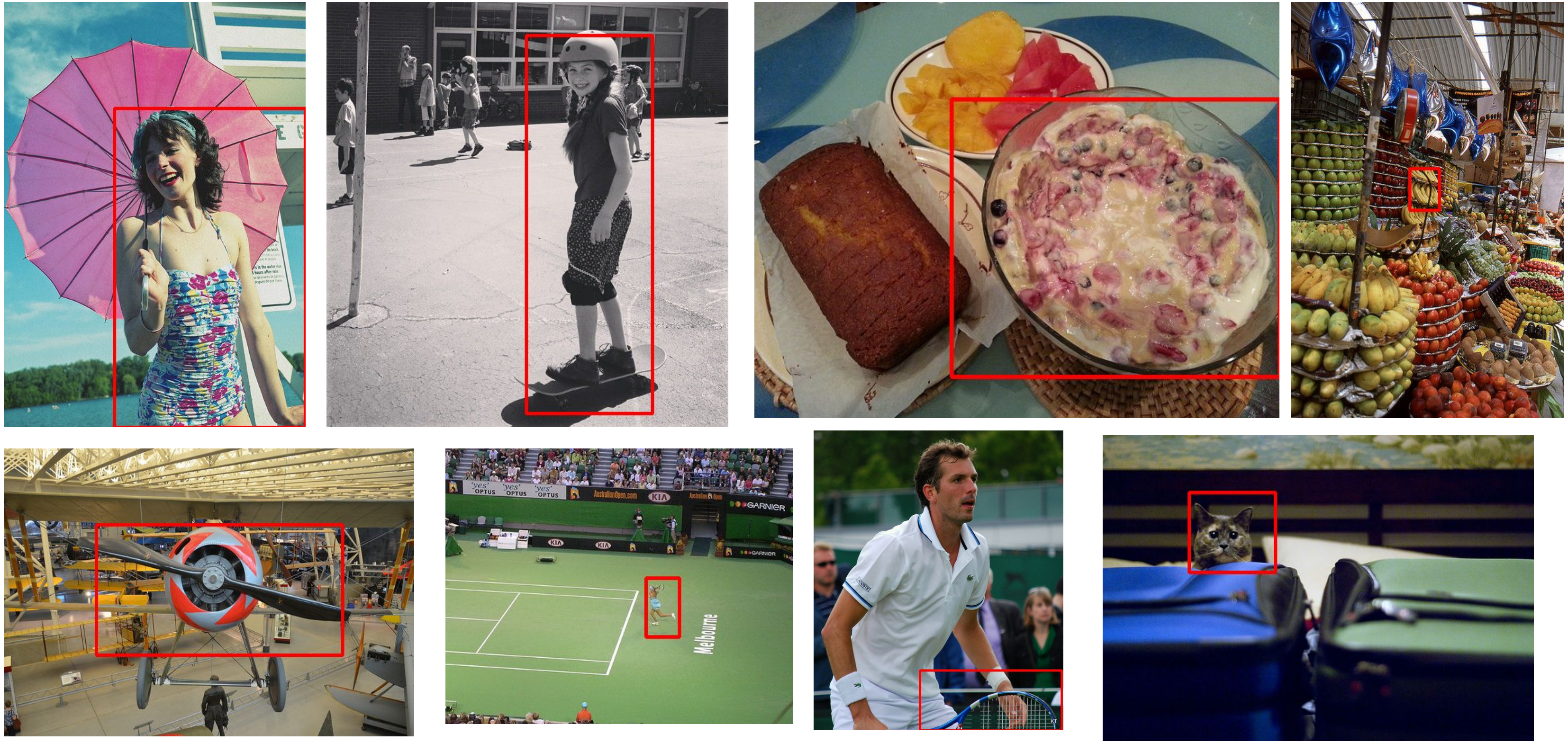
Initial **seed**

LOST [Siméoni et al., BMVC'21]

- Compute a binary similarity graph
  (nodes connected if cosine similarity >0)
- **Object =** patch with the lowest degree
  & connected correlated patches
- Additional expansion step

Siméoni et al., Localizing Objects with Self-Supervised Transformers and no Labels, BMVC'21

LOST qualitative results



Siméoni et al., Localizing Objects with Self-Supervised Transformers and no Labels, BMVC'21

LOST quantitative results

Method	VOC07_trainval	VOC12_trainval	COCO_20k
Selective Search [65]	18.8	20.9	16.0
EdgeBoxes [84]	31.1	31.6	28.8
Kim *et al*. [38]	43.9	46.4	35.1
Zhang *et al*. [80]	46.2	50.5	34.8
DDT+ [72]	50.2	53.1	38.2
rOSD [68]	54.5	55.3	48.5
LOD [69]	53.6	55.1	48.5
DINO-seg (w. ViT-S/16)	45.8	46.2	42.1
LOST (ours)	**61.9**	**64.0**	**50.7**
	+ 7.4	**+ 8.7**	**+ 2.2**

Corloc metric = % of correct boxes
→ a predicted box is correct if has IoU>0.5 with one of gt boxes

Previous **SoTA** were:

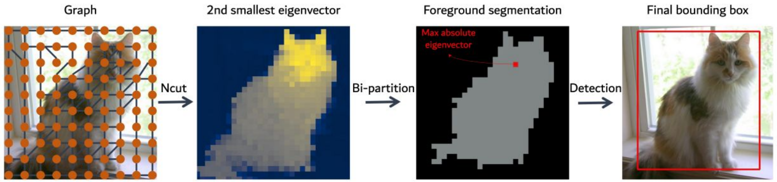
- **Region proposals** method (high recall, low precision)
- Methods based on **inter-image similarity**: dataset exploration often with quadratic costs

Siméoni et al., Localizing Objects with Self-Supervised Transformers and no Labels, BMVC'21

Then came more powerful algorithms

TokenCut [Wang et al. CVPR’22], **Deep Spectral Methods** [Melas-Kyriazi et al. CVPR’22], **SelfMask** [Shi et al. CVPRW’22]

- Same features, similar graph
- Solve a normalized graph-cut problem with **spectral clustering** → improved localization



CutLer [Wang et al. CVPR’23]

- Detect several objects
- Remove already discovered nodes from the graph and **repeat the operation**

More details/discussion in our recent **survey**:
Unsupervised Object Localization in the Era of Self-Supervised ViTs: A Survey, Siméoni et al., IJCV’24

Foreground / background unsupervised segmentation

FOUND [Siméoni et al., CVPR'23]

- **Look for the background instead of objects**
- No hypotheses about objects

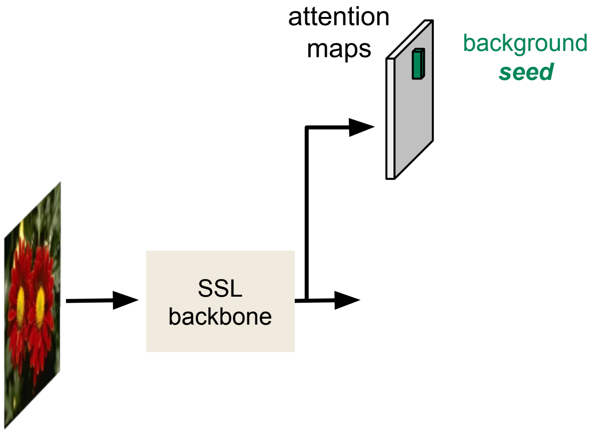
Siméoni et al., Unsupervised Object Localization: Observing the Background to Discover Objects, CVPR'23

Foreground / background unsupervised segmentation

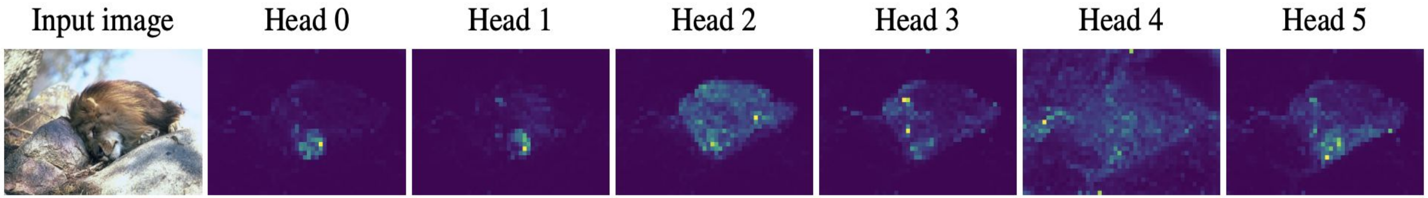
FOUND [Siméoni et al., CVPR'23]

- **Look for the background instead of objects**
- No hypotheses about objects

Background mask:
- Seed = patch receiving least attention



[CLS] self-attention maps



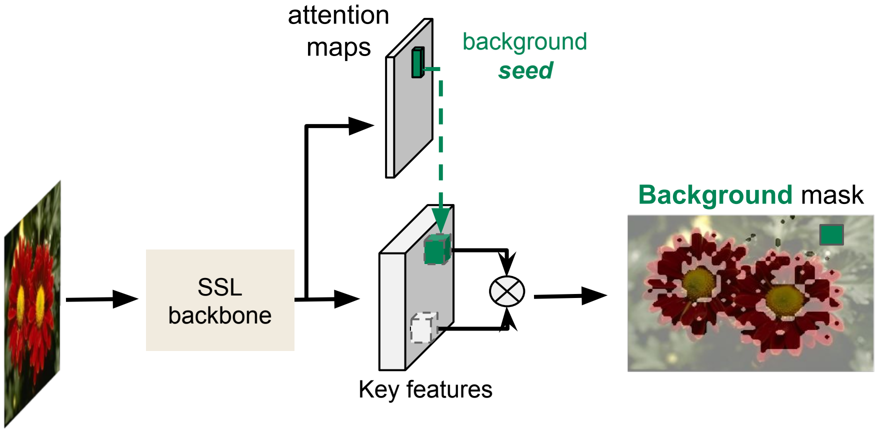
Siméoni et al., Unsupervised Object Localization: Observing the Background to Discover Objects, CVPR'23

Foreground / background unsupervised segmentation

FOUND [Siméoni et al., CVPR'23]

- **Look for the background instead of objects**
- No hypotheses about objects

Background mask:
- Seed = patch receiving least attention
- Mask = correlated patches to seed



Siméoni et al., Unsupervised Object Localization: Observing the Background to Discover Objects, CVPR'23

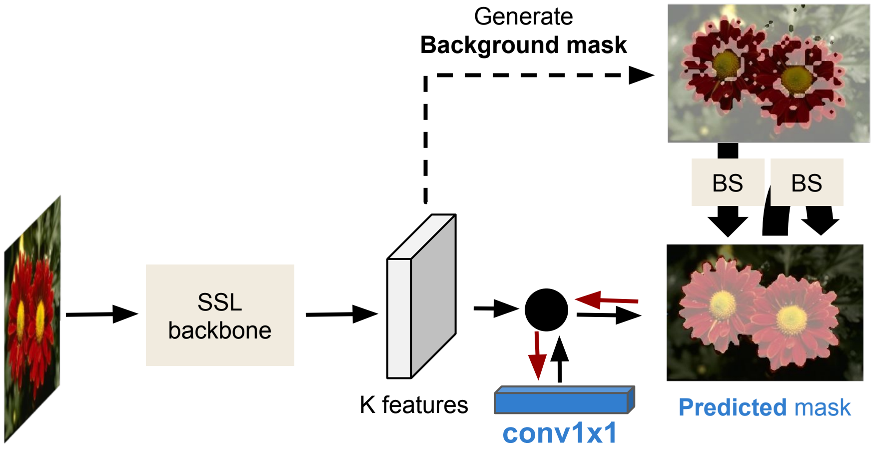
Foreground / background unsupervised segmentation

FOUND [Siméoni et al., CVPR'23]

- **Look for the background instead of objects**
- No hypotheses about objects

Background mask:
- Seed = patch receiving least attention
- Mask = correlated patches to seed

FOUND = a single conv 1x1
- Trained using background masks as pseudo-labels
- **Bilateral Solver** (BS) used to refine masks along pixel edges



Background mask | **Foreground** mask | **Foreground** mask + BS | **Predicted** mask

Siméoni et al., Unsupervised Object Localization: Observing the Background to Discover Objects, CVPR'23

Out-of-domain predictions (*no post-processing)*

FOUND [Siméoni et al., CVPR'23]

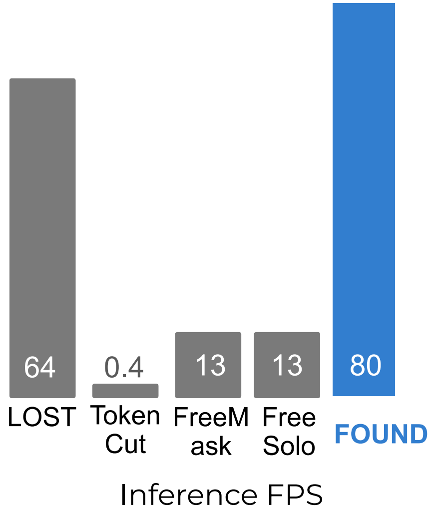
- **Single conv 1x1** layer trained with pseudo-labels
- Trained for 500 it. on DUTS-TR [Wang et al, CVPR17] (10k images) ~ **2h with a single GPU**
- Inference at **80 FPS** on a V100

Siméoni et al., Unsupervised Object Localization: Observing the Background to Discover Objects, CVPR'23

Quantitative results

Method	Learning	DUT-OMRON [65]			DUTS-TE [55]			ECSSD [43]		
		Acc	IoU	max F_β	Acc	IoU	max F_β	Acc	IoU	max F_β
— Without post-processing bilateral solver —										
HS [63]		.843	.433	.561	.826	.369	.504	.847	.508	.673
wCtr [73]		838	.416	.541	.835	.392	.522	.862	.517	.684
WSC [28]		.865	.387	.523	.862	.384	.528	.852	.498	.683
DeepUSPS [36]		.779	.305	.414	.773	.305	.425	.795	.440	.584
BigBiGAN [54]		.856	.453	.549	.878	.498	.608	.899	.672	.782
E-BigBiGAN [54]		.860	.464	.563	.882	.511	.624	.906	.684	.797
Melas-Kyriazi et al. [33]		.883	.509	—	.893	.528	-	.915	.713	—
LOST [45] ViT-S/16 [6]		.797	.410	.473	.871	.518	.611	.895	.654	.758
DSS [34] [59]		—	.567	—	—	.514	—	—	.733	—
TokenCut [59] ViT-S/16 [6]		.880	533	.600	.903	576	.672	.918	712	.803
SelfMask [44]	✓	.901	.582	—	.923	626	—	.944	781	—
FOUND — single ViT-S/8 [6]	✓	**.920**	**.586**	**.683**	**.939**	637	**.733**	.912	793	.946
FOUND — multi ViT-S/8 [6]	✓	.912	.578	.663	.938	**645**	.715	**.949**	**807**	**.955**

- **Inference at 80 FPS** on a V100
- **<1000** learned **parameters**



Siméoni et al., Unsupervised Object Localization: Observing the Background to Discover Objects, CVPR'23

From **Unsupervised Object Localization** to **Open-Vocabulary Semantic Segmentation**

From **Unsupervised Object Localization** to **Open-Vocabulary Semantic Segmentation**

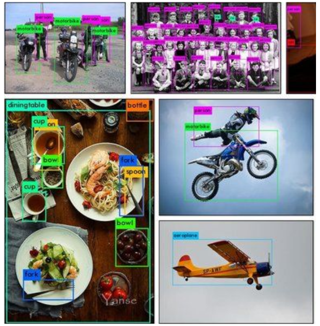
Limits in the object localization task

Classic benchmarks
Closed vocabulary setup

Limitation in the **definition** of the problem
- Requires the definition of a finite set of **classes**

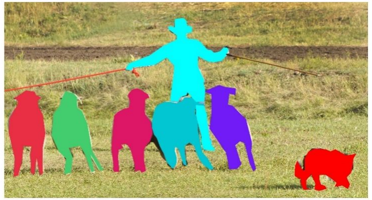
Fully-supervised training

High costs
- Expensive in money/time to get annotation
- **For each new class**: need new annotation + re-training



Object detection

COCO [Lin et al. ECCV’14]



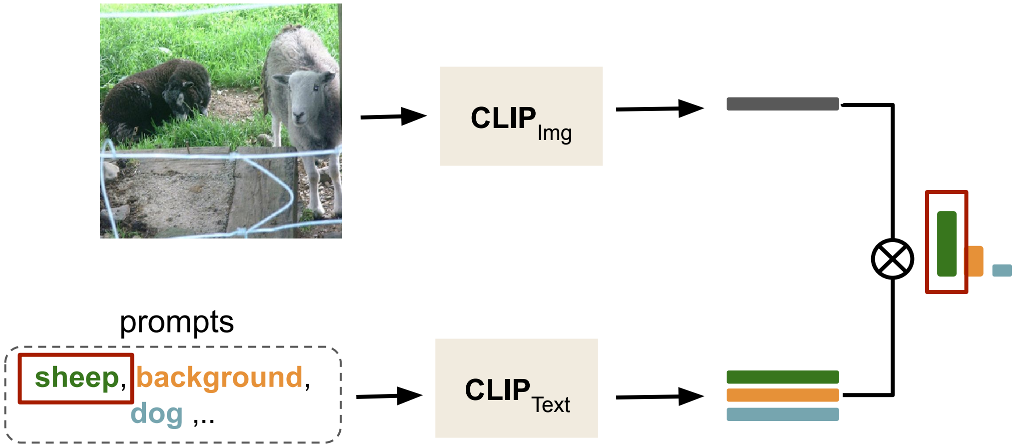
Instance segmentation

Global text/image alignment

- Powerful VLMs which **align text and images**
- **CLIP** [Ilharco et al. 21] trained with a **global** objective to **align** *text to images*
  → great zero-shot classification

However, going **from global to dense pixel classification** is **not obvious**

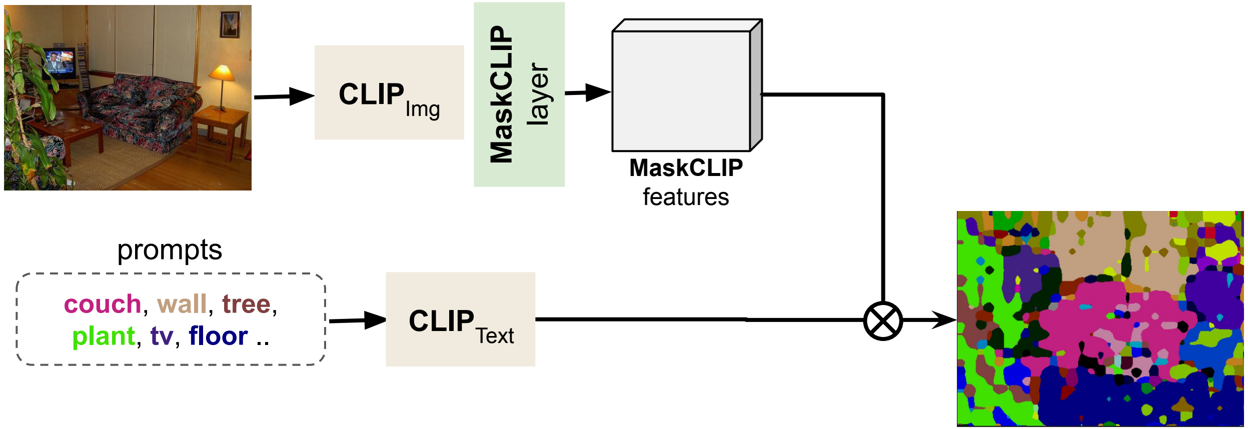
- very noisy (**MaskCLIP** [Zhou et al. ECCV'22]),
- require training (**TCL** [Cha et al. CVPR'23], **CLIPpy** [Ranasinghe et al. ICCV'23]), extra annotation, etc..



MaskCLIP: pixel-level CLIP-like features

MaskCLIP [Zhou et al. ECCV’22]

- No training
- Drops the global pooling layer of CLIP
- Matches the projected features directly to text via a 1×1 convolution layer.

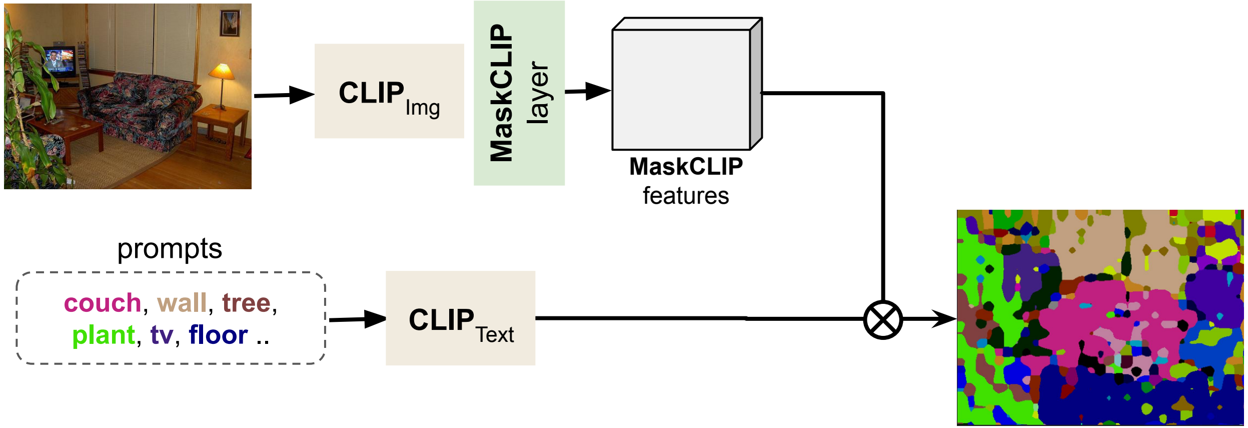


Any way to **leverage SSL** ?

MaskCLIP: pixel-level CLIP-like features

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation



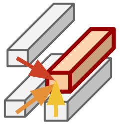
Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

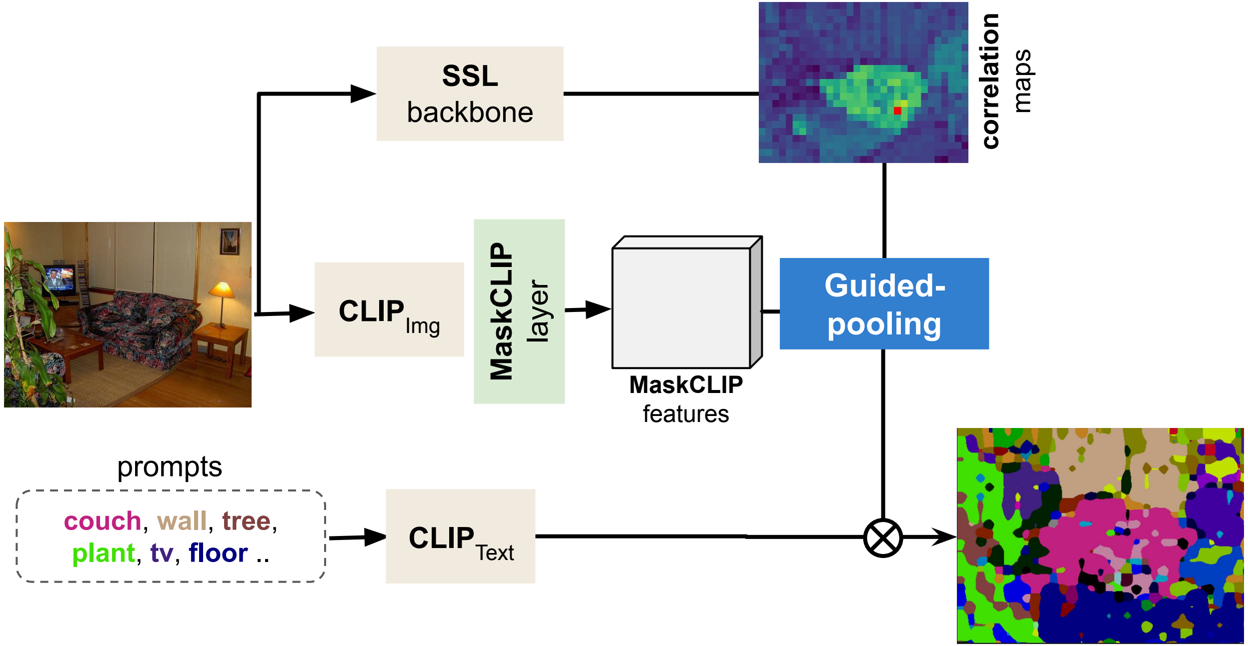
Leveraging SSL patch correlation information

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation

- **Guided pooling** = weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold



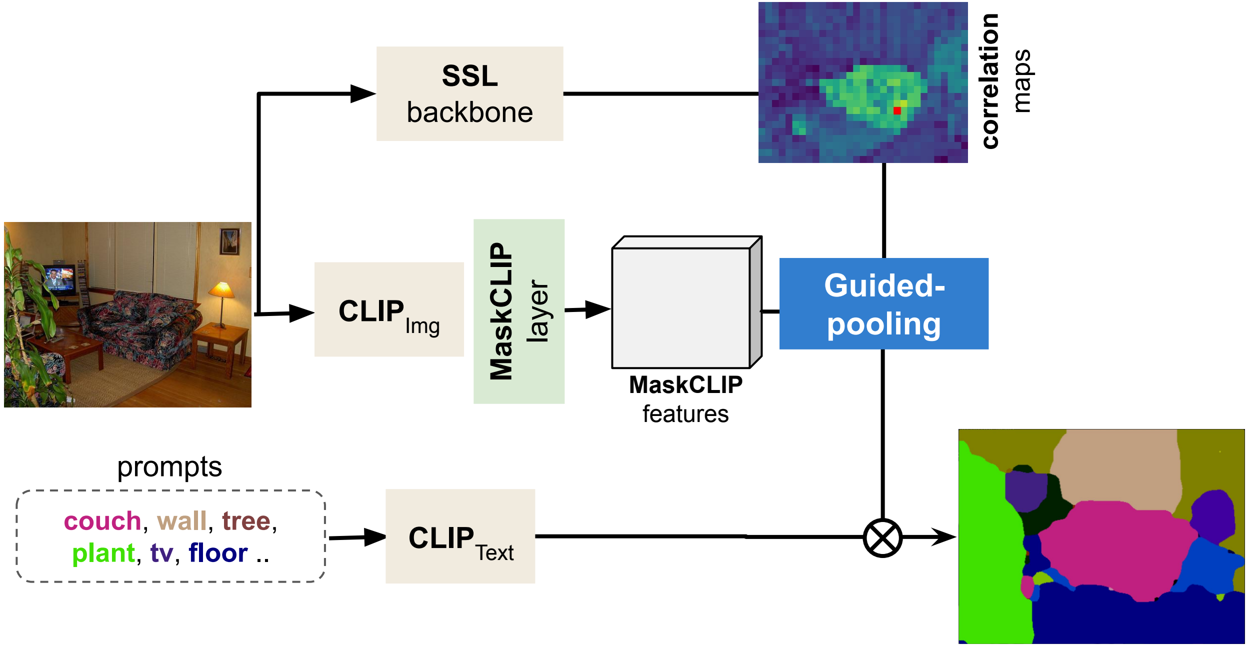


Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Leveraging SSL patch correlation information

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation
- **Guided pooling** = weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold

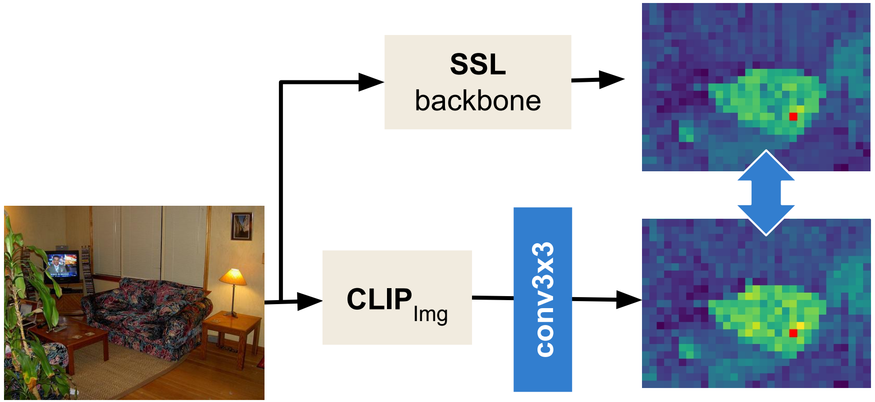


Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Teaching CLIP a **first DINO trick**

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation
- **Guided pooling = weighted average of** pixel features
  - weights = SSL correlations
  - only correlation > threshold
- Teach **CLIP a first trick**
  - Single **conv3x3** trained to produce features w/ correlations *alike DINO's*
  - Trained with a **BCE**
  - ~40 mins on 1 NVIDIA A5000 and **1.5k images** (PASCAL VOC train)



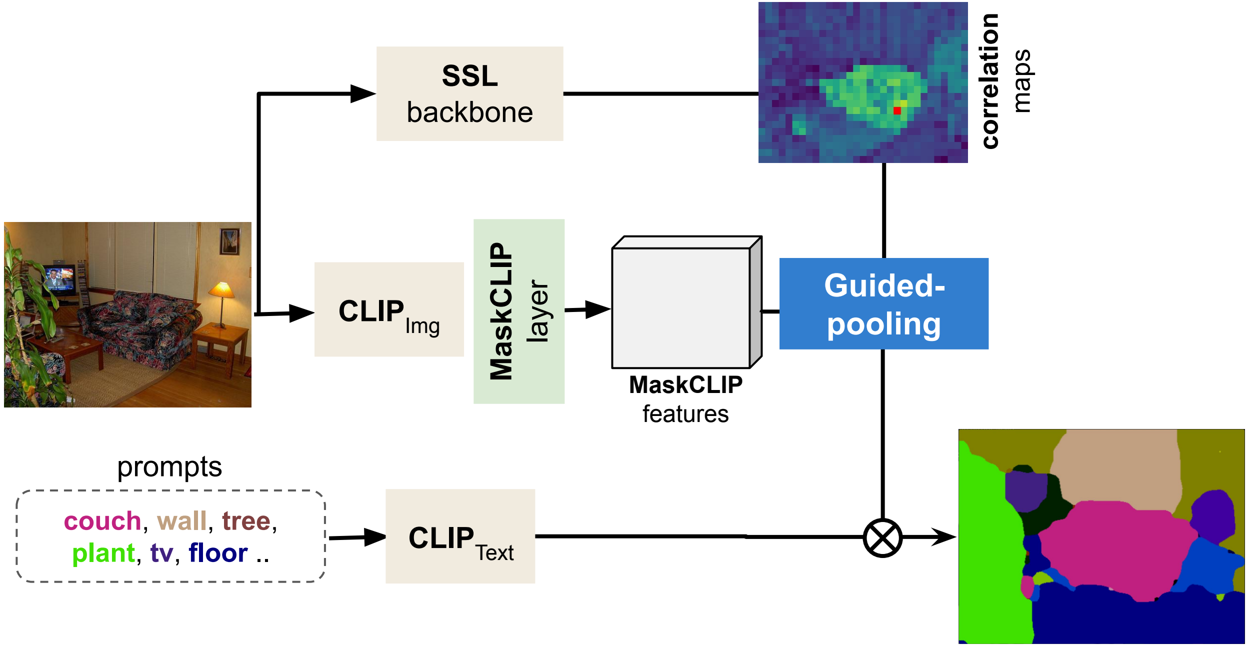
CLIP already contains **good localization properties**

Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Teaching CLIP a **first DINO trick**

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation
- **Guided pooling** = weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold
- Teach **CLIP a first trick**
  - Single **conv3x3** trained to produce features w/ correlations *alike DINO's*

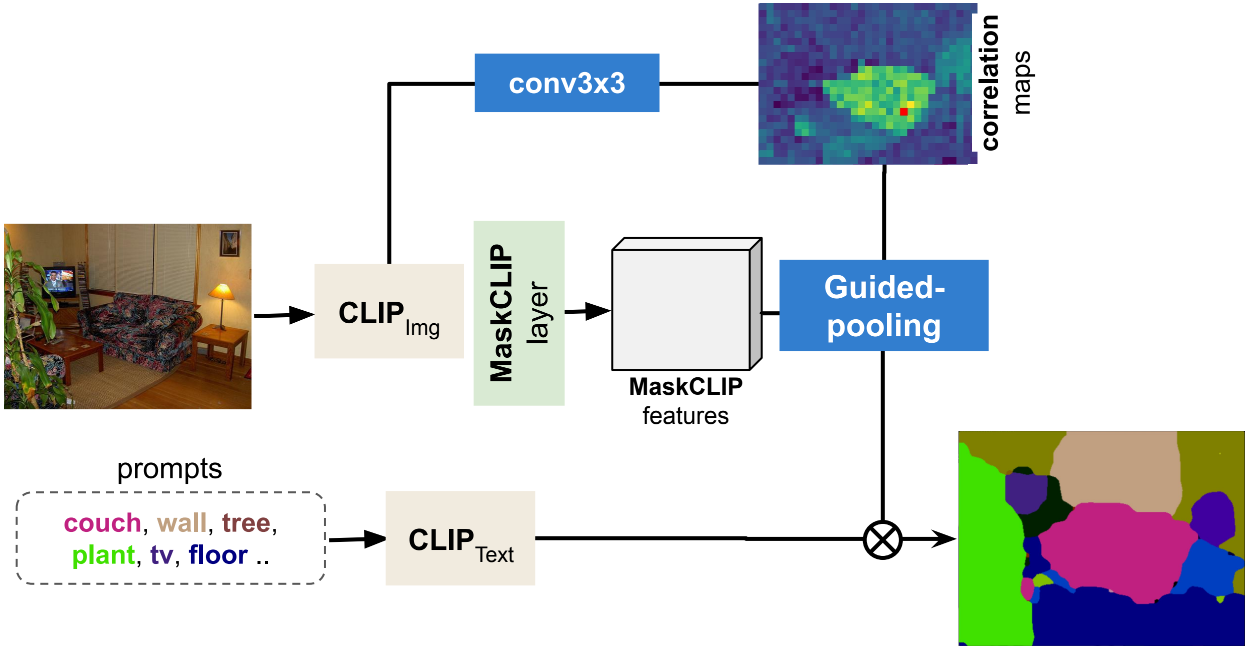


Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Teaching CLIP a **first DINO trick**

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation
- **Guided pooling** = weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold
- Teach **CLIP a first trick**
  - Single **conv3x3** trained to produce features w/ correlations *alike DINO's*

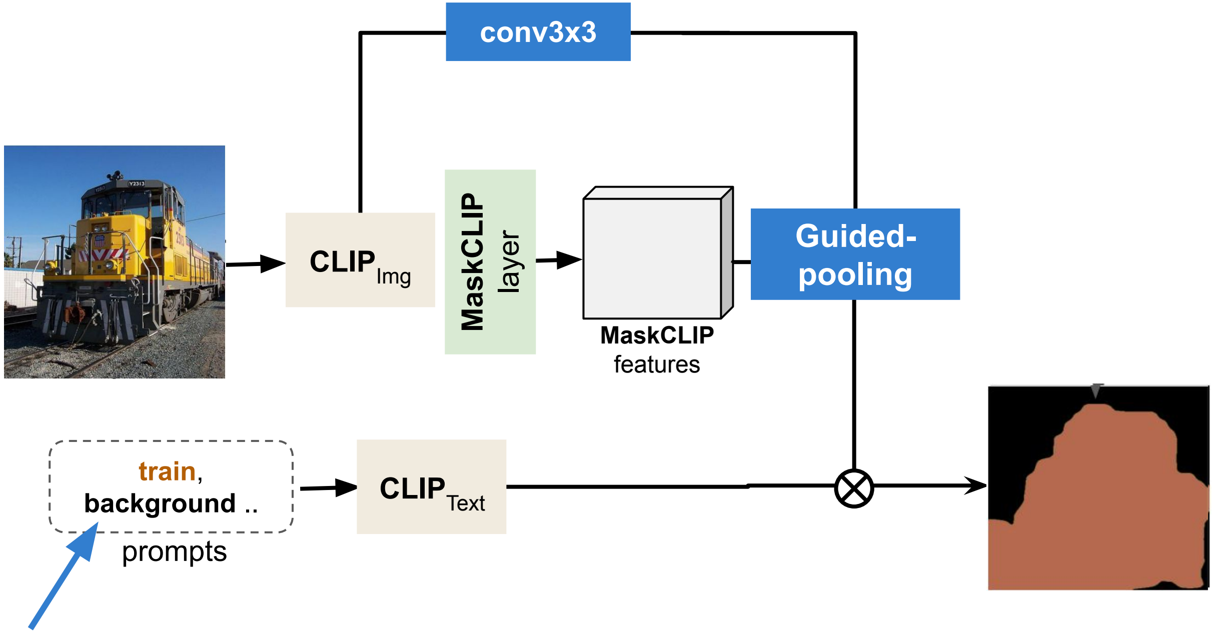


Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Leveraging SSL patch correlation information

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation
- **Guided pooling** = weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold
- Teach **CLIP a first trick**
  - Single **conv3x3** trained to produce features w/ correlations *alike DINO's*

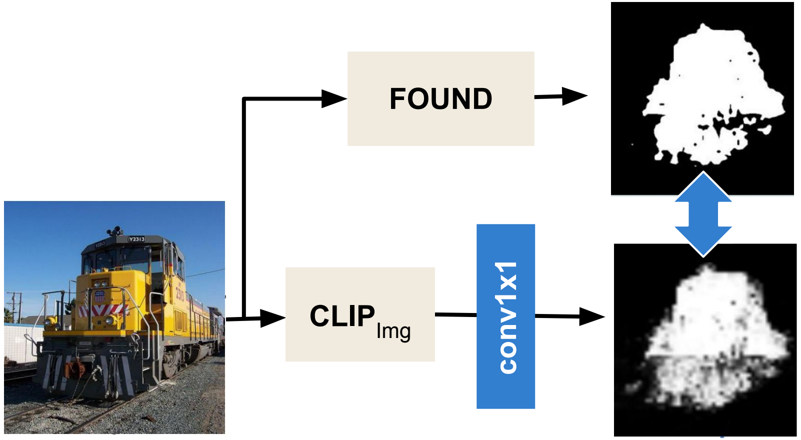


Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Leveraging SSL patch correlation information

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation
- **Guided pooling =** weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold
- Teach **CLIP a first trick**
  - Single **conv3x3** trained to produce features w/ correlations *alike DINO's*
- Teach **CLIP a second trick**
  - Foreground segmentation w/ **conv1x1** trained to mimic FOUND

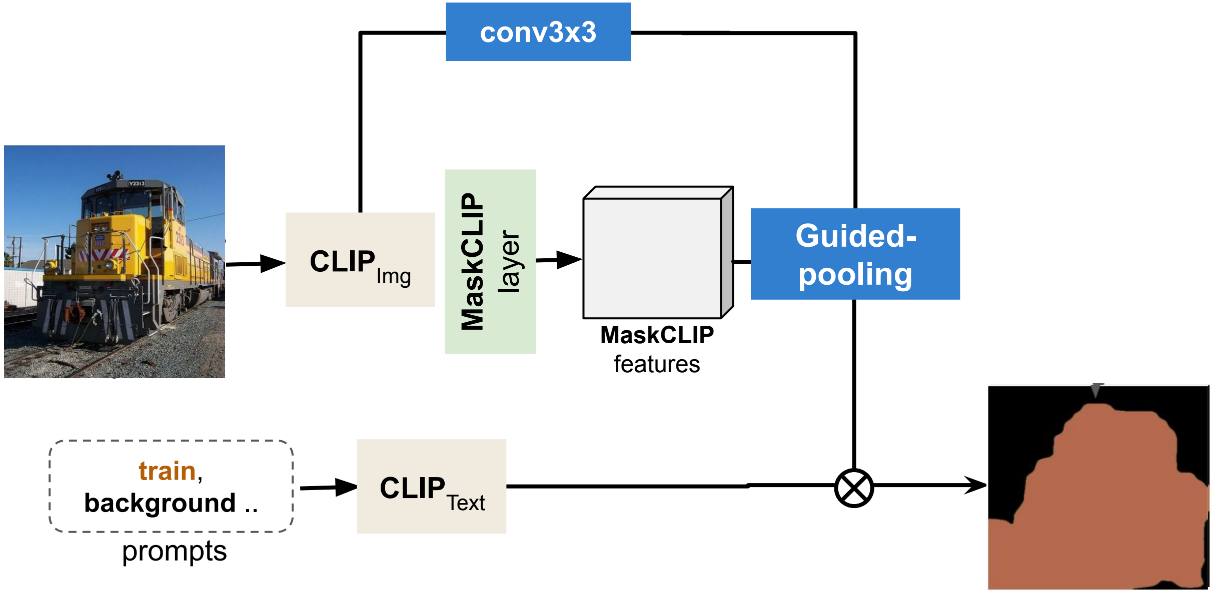


Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Leveraging SSL patch correlation information

CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation
- **Guided pooling** = weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold
- Teach **CLIP a first trick**
  - Single **conv3x3** trained to produce features w/ correlations *alike DINO's*
- Teach **CLIP a second trick**
  - Foreground segmentation w/ **conv1x1** trained to mimic FOUND



Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Leveraging SSL patch correlation information

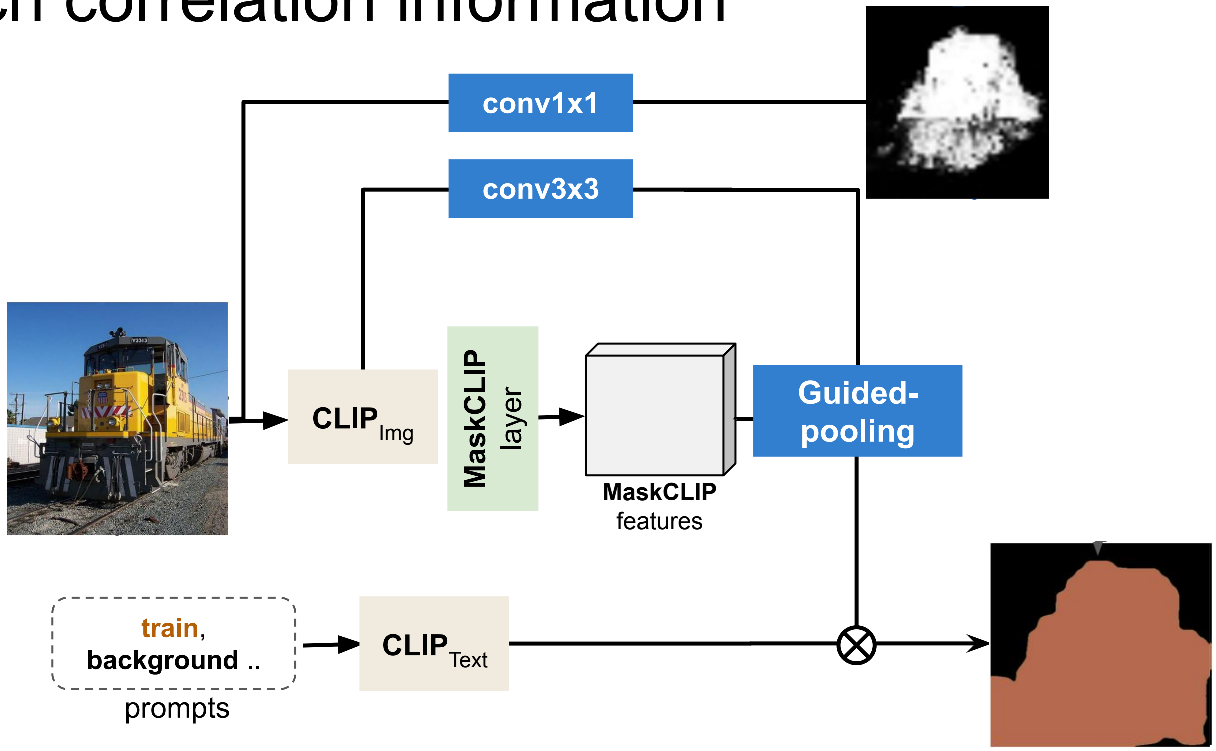
CLIP-DINOiser [Wysoczanska et al., ECCV'24]

- Idea: Strengthen **MaskCLIP** using SSL correlation

- **Guided pooling** = weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold

- Teach **CLIP a first trick**
  - Single **conv3x3** trained to produce features w/ correlations *alike DINO's*

- Teach **CLIP a second trick**
  - Foreground segmentation w/ **conv1x1** trained to mimic FOUND

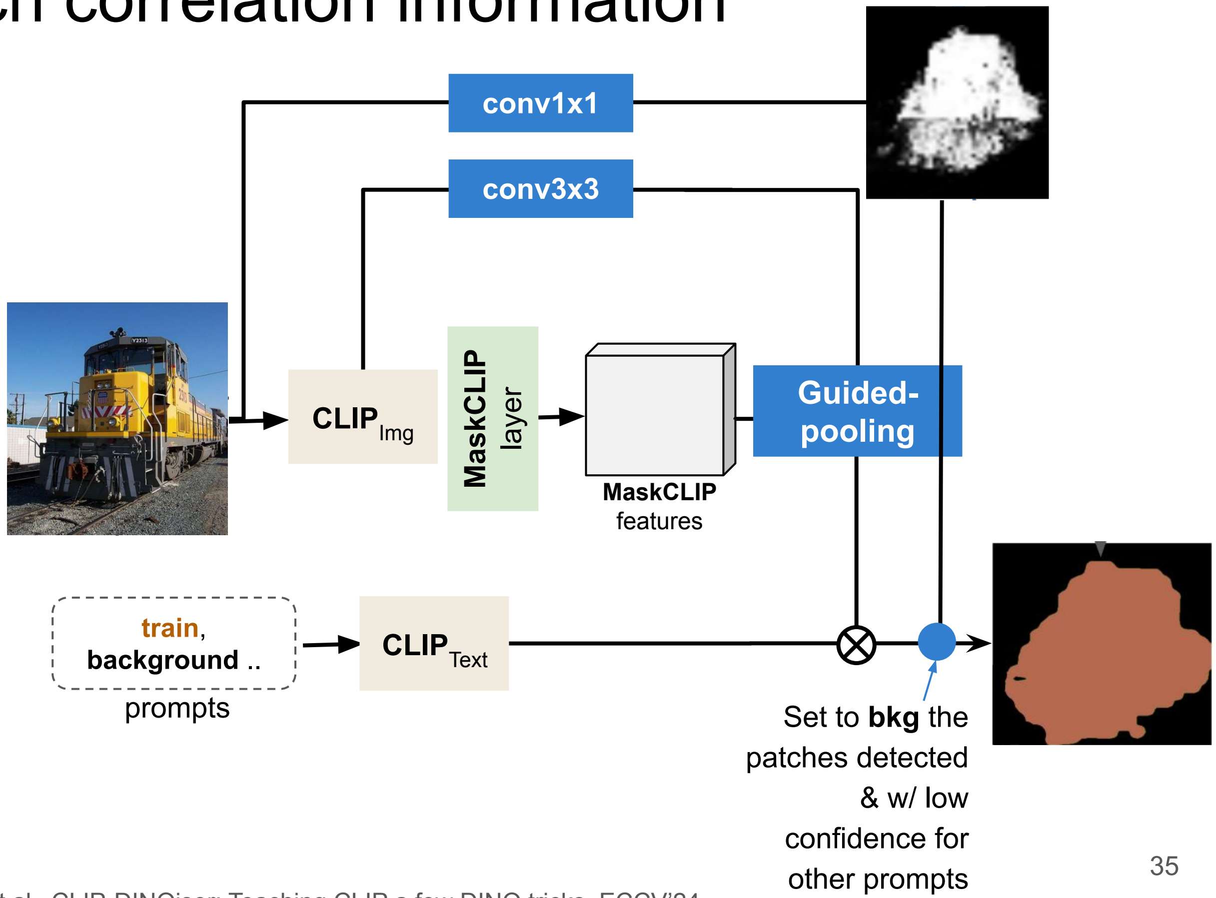


Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Leveraging SSL patch correlation information

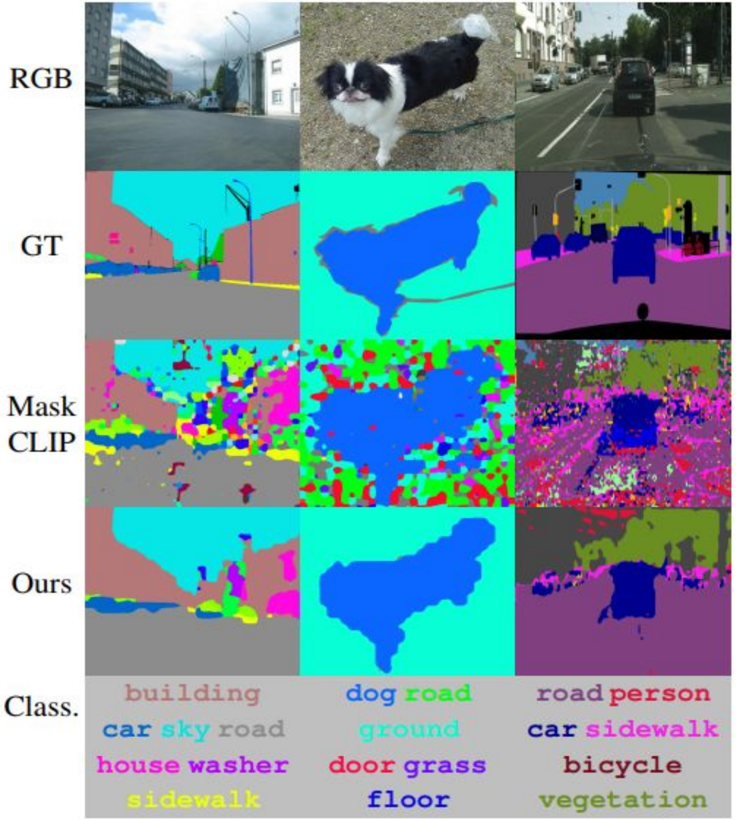
CLIP-DINOiser [Wysoczanska et al., ECCV’24]

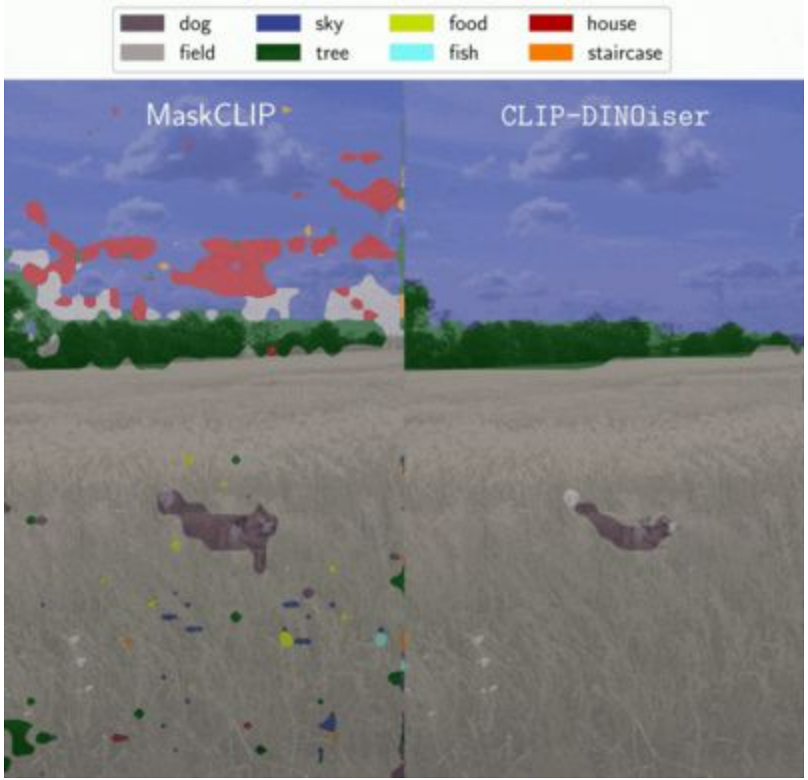
- Idea: Strengthen **MaskCLIP** using SSL correlation
- **Guided pooling** = weighted average of pixel features
  - weights = SSL correlations
  - only correlation > threshold
- Teach **CLIP a first trick**
  - Single **conv3x3** trained to produce features w/ correlations *alike DINO’s*
- Teach **CLIP a second trick**
  - Foreground segmentation w/ **conv1x1** trained to mimic FOUND



Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV’24

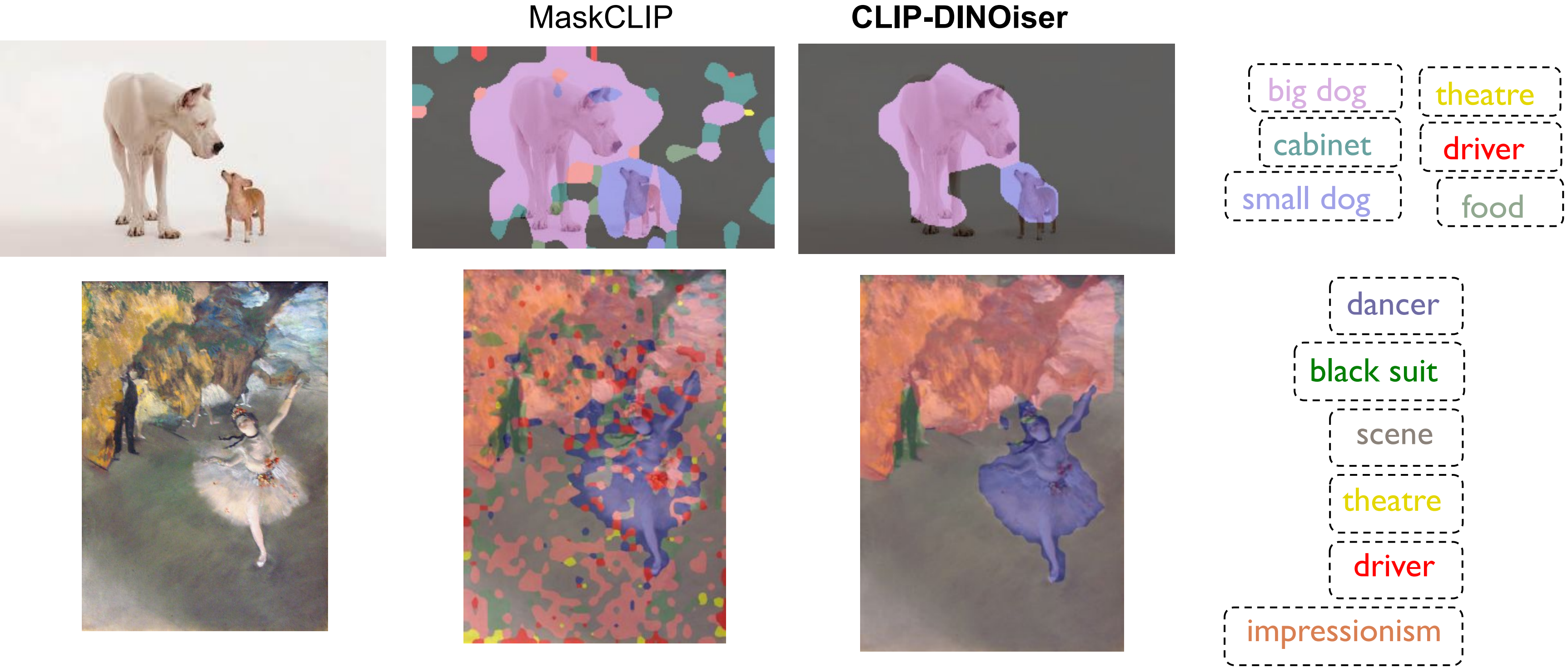
CLIP-DINOiser's qualitative results





Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

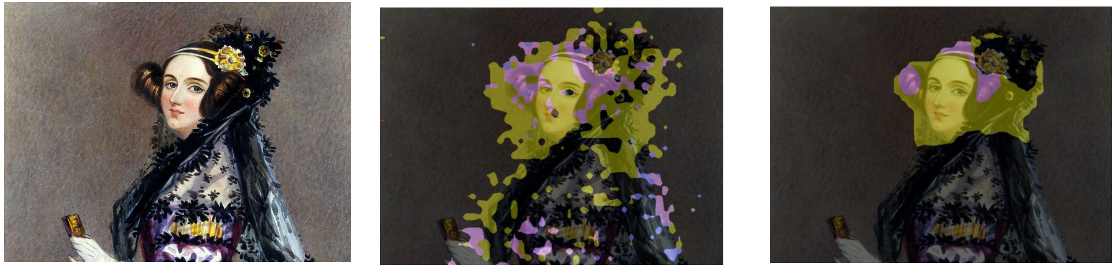
CLIP-DINOiser's qualitative results



Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

CLIP-DINOiser's qualitative results

MaskCLIP | **CLIP-DINOiser**



Princess Leia

Ada Lovelace

Luke Skywalker

rusted van
green trees
clouds mountains

french pastries
wooden table
plate

sky sports car
strange turtle
city water

white horse
dark horse

leather bag
vintage bike

Wysoczanska et al., CLIP-DINOiser: Teaching CLIP a few DINO tricks, ECCV'24

Going further

A Study of Test-time Contrastive Concepts for Open-world, Open-vocabulary Semantic Segmentation

Monika Wysoczańska[1*] **Antonin Vobecky**[2,3,4] **Amaia Cardiel**[2,8]

Tomasz Trzciński[1,5,6] **Renaud Marlet**[2,7] **Andrei Bursuc**[2] **Oriane Siméoni**[2]

[1]Warsaw University of Technology [2]valeo.ai [3]CIIRC CTU Prague [4]FEE CTU Prague
[5]Tooploox [6]IDEAS NCBR [7]LIGM, Ecole des Ponts, Univ Gustave Eiffel
[8]Université Grenoble Alpes

Rethink the **evaluation paradigm** of the open-vocabulary semantic segmentation: new metric and removing access to an exhaustive set of classes

Where do we go from here?

Why do **we like self-supervision?**

- It requires **no annotation**

- Learns **strong representation**
  - For **pre-training**
  - Good **localization** properties

- No need to know the end task (often ill-defined)
- Not impacted by annotation biases

- Can be exploited at little cost eg. with **cheap convolutional layers**
- Localization of objects is possible and **classes can come later**

Remaining challenges

- How to handle the ill-definition of an object?
- Multi-instance?
- Handling granularity?
- Different representation for **end usage/tasks**?

Questions?