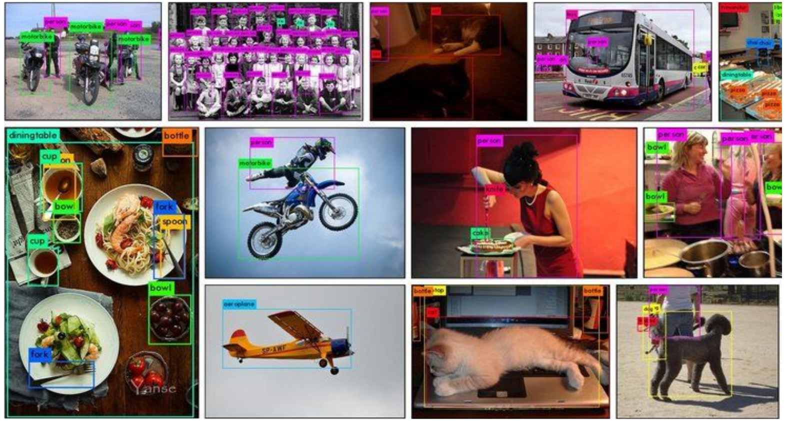
Exploiting self-supervised features: unsupervised object localization

Oriane Siméoni
valeo.ai

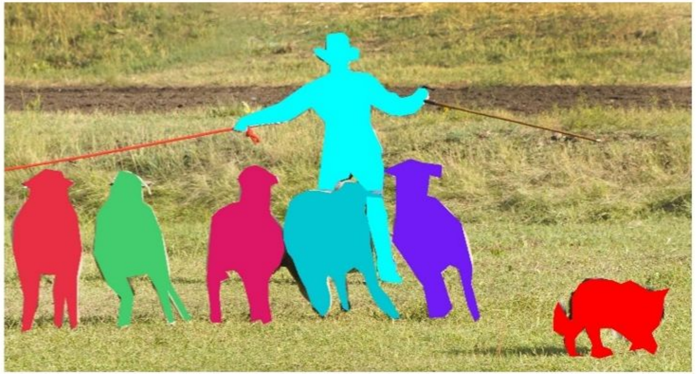
Object localization

Object detection



COCO [Lin et al. ECCV'14]

Instance segmentation



COCO [Lin et al. ECCV'14]

Training a model for those tasks requires

- a lot of **annotation** ✍️
- the definition of a **finite set of classes**

Unsupervised object localization

Object detectio

stance segmentation

Segment anything [Kirillov et al., arxiv'23]

Training a model for those tasks requires

- a lot of **annotation** ✍️
- the definition of a **finite set of classes**

How to perform **object localization** with **no annotation ?**

Unsupervised object localization

Unsupervised object localization

single-object/mask

Unsupervised **object discovery**

Metric: **corloc**
→ the percentage of correct boxes

Unsupervised **saliency detection**

Metric: **IoU**, **Accuracy**

multi-object

Unsupervised class-agn. **object detection**

Metric: **AP**

Unsupervised class-agn. **instance segmentation**

Metric: **AP**

Some background

- **Region proposals**
  - Generate numerous class-agnostic bounding boxes with **high recall** but **low precision** eg. EdgeBoxes [Zitnick et al., ECCV'14], Selective Search [Uijlings et al., IJCV'13]

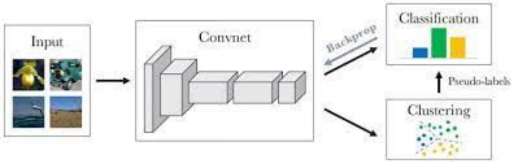
- **Methods based on inter-image similarity**
  - Explore an **entire dataset** [Cho et al. CVPR'15; Vo et al. CVPR'19; ECCV'20; NeurIPS'21]
  - Often requires external box proposals
  - **Quadratic costs** (except for [Vo et al. NeurIPS'21])

Vo et al. CVPR'19

Powerful self-supervision and transformers

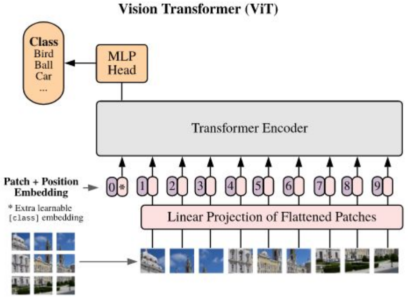
- **Self-supervision** has shown to be very powerful

RotNet [Gidaris et al. ICLR'18]



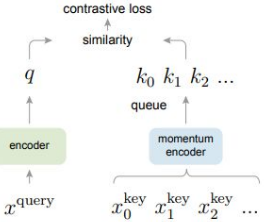
DeepCluster [Caron et al. ECCV'18]

- **Transformers** applied to vision become prevalent

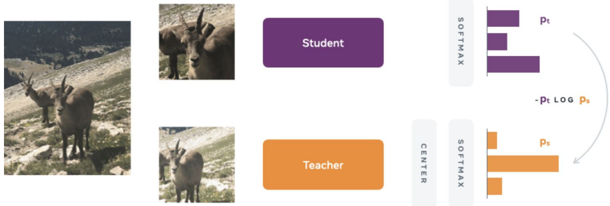


ViT [Dosovitskiy et al. ICLR'20]

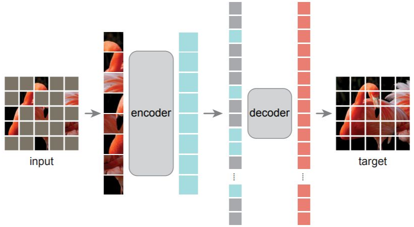
- **Transformers** & **self-supervision**



MOCO [He et al. CVPR'20]



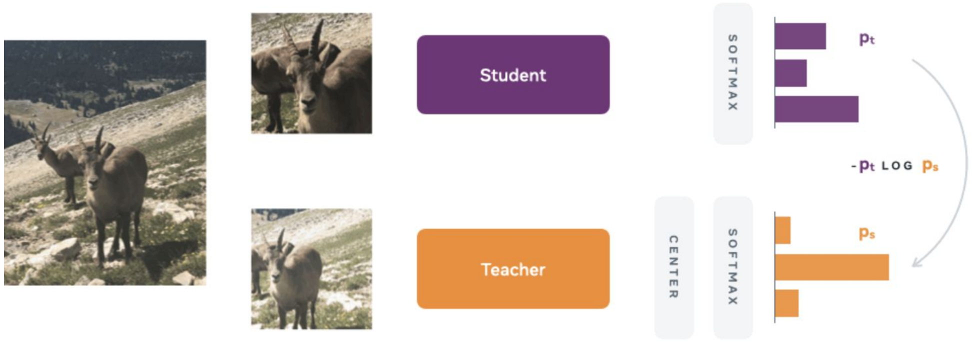
DINO [Caron et al. ICCV'21]

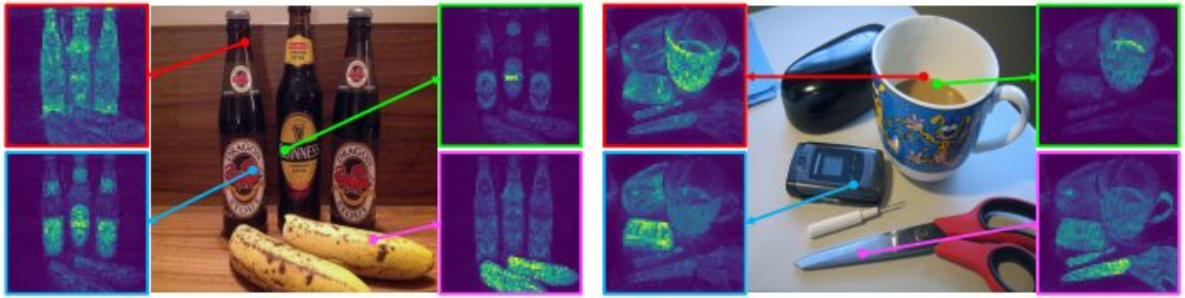


MAE [He et al. CVPR'22]

Exploiting existing powerful self-supervised features

ViT models pre-trained in a **self-supervised** manner have **good localization properties**

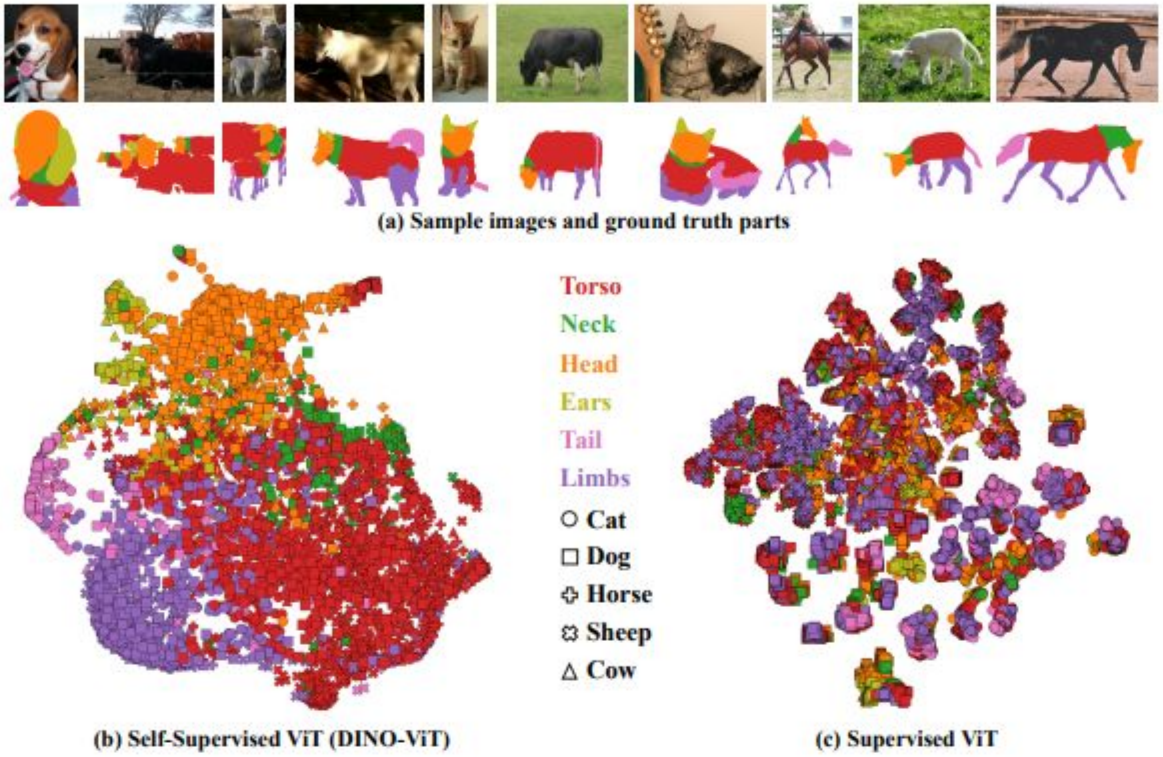




DINO [Caron et al. ICCV'21]

Exploiting existing powerful self-supervised features

ViT models pre-trained in a **self-supervised** manner have **good localization properties**



t-SNE visualization

Deep ViT Features as Dense Visual Descriptors [Amir et al. ECCVW'22]

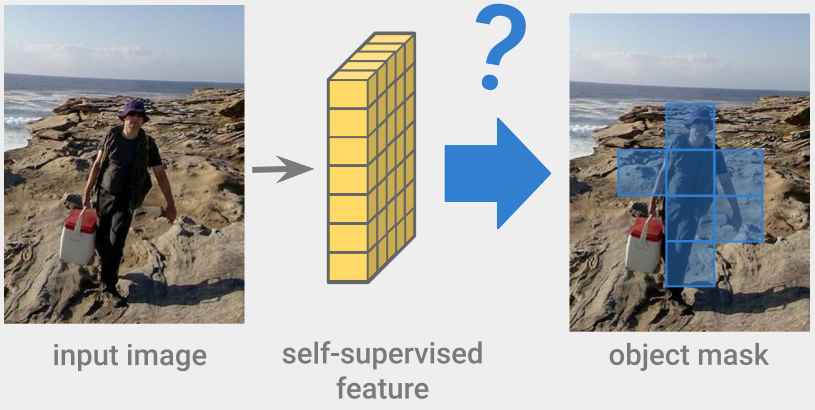
Presentation outline

Unsupervised object discovery
Generation of initial masks

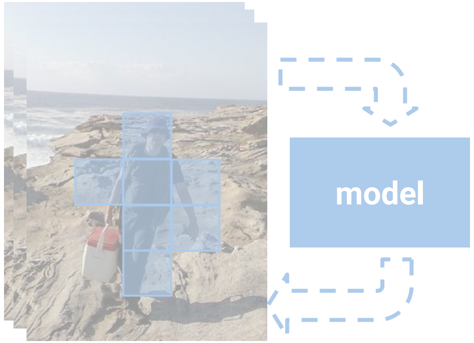
Improving localization through learning

Class-aware ?

Presentation outline



Unsupervised object discovery
Generation of initial masks



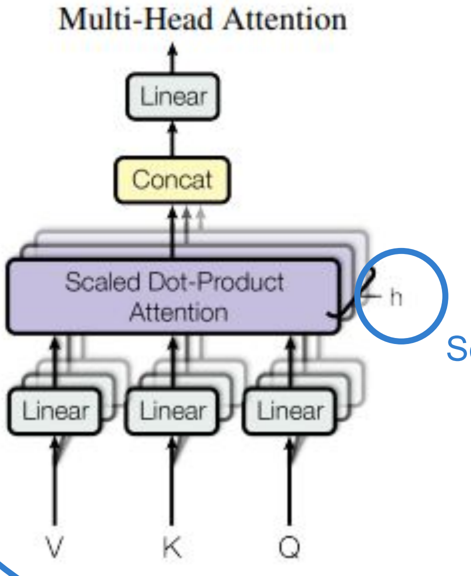
Improving localization
through learning

Class-aware ?

Using the self-attention maps

[CLS] self-attention maps

DINO [Caron et al. ICCV'21]



Several heads

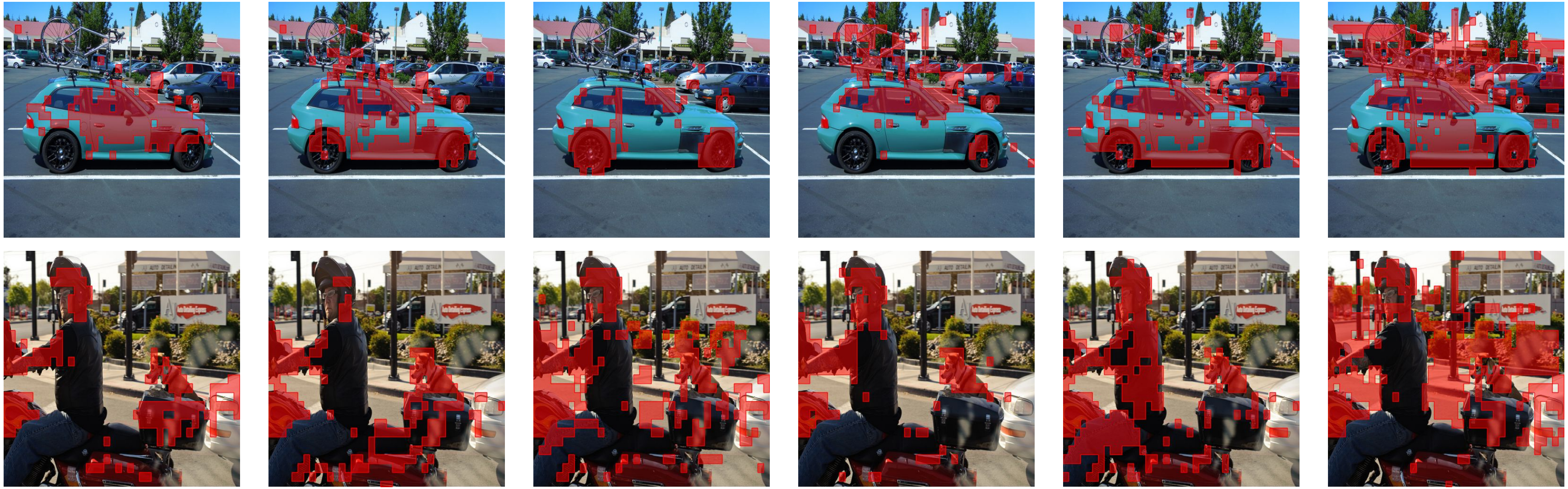
$$\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Attention is all you need [Vaswani et al. NeurIPS'17]

Using the self-attention maps

- But, **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

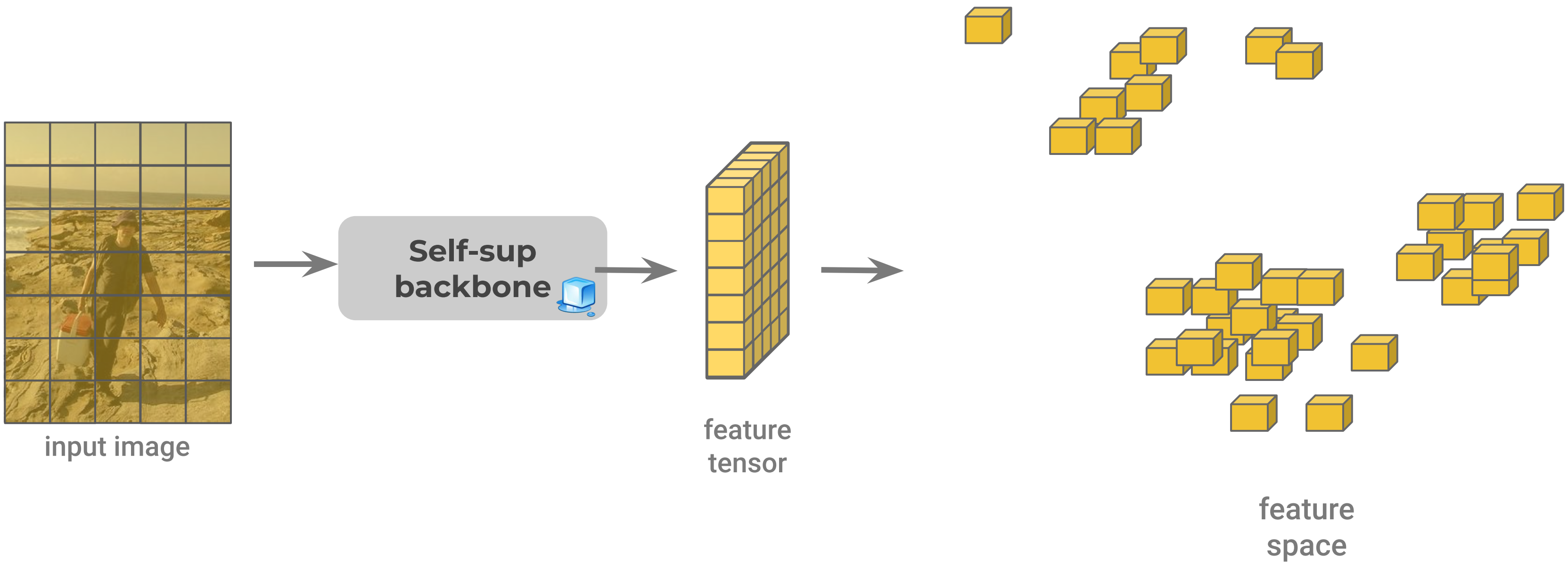
How to exploit the self-supervised features?

- The **K,Q,V** features are interesting and do not require decision on the head

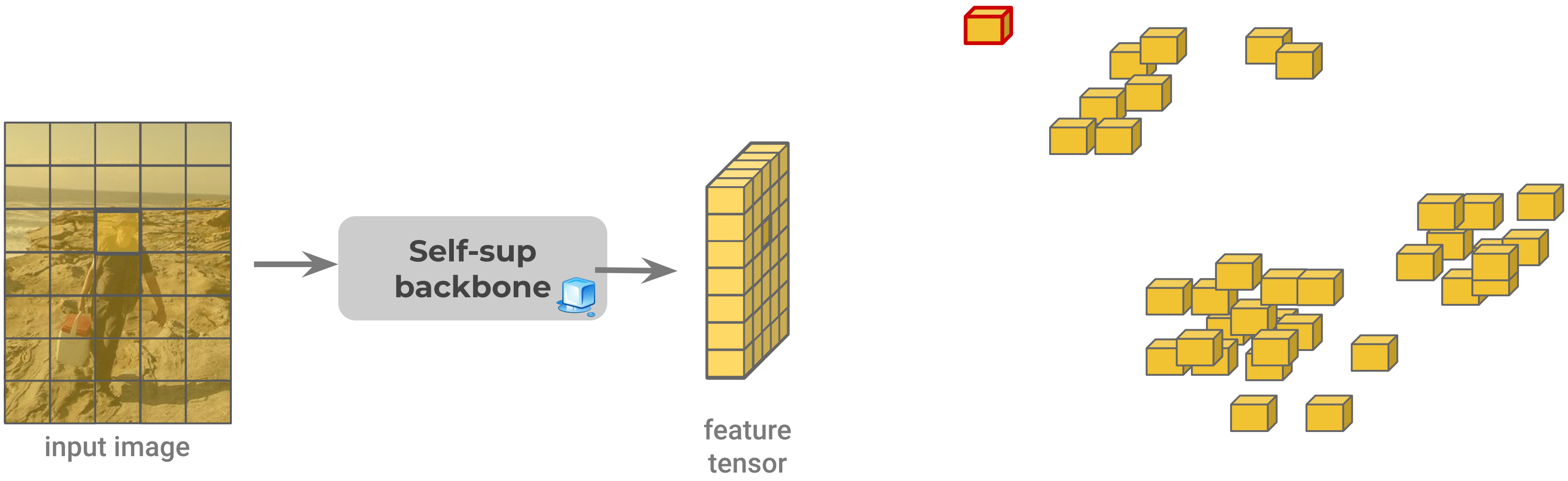
- **Good correlation** properties of the features
- Features for object patches are more **discriminative** than for the background
  → Object patches are less correlated to other patches

- Most methods require to compute a **graph** of patch features

Build a similarity graph

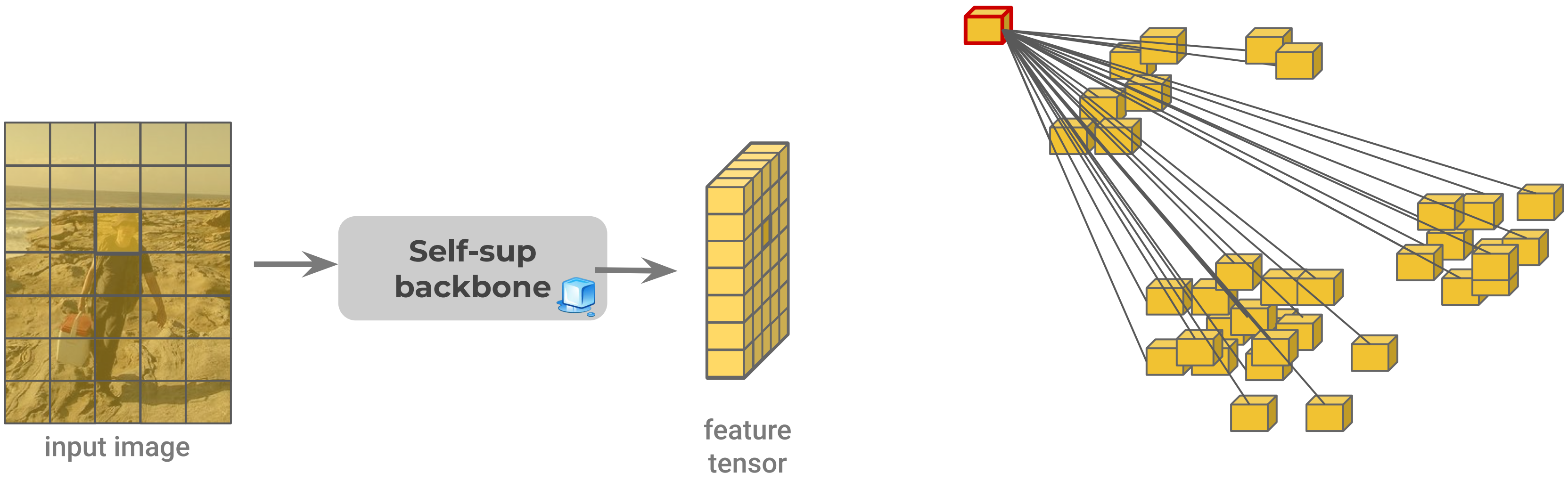


Build a similarity graph



Building graph:
- nodes: patches

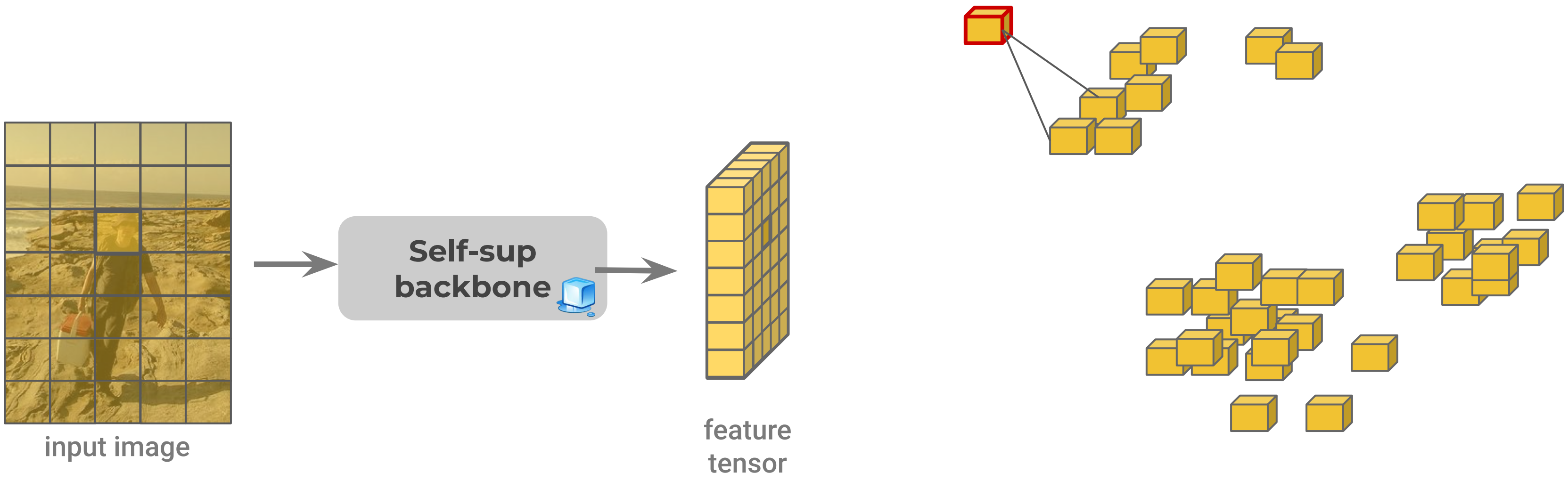
Build a similarity graph



Building graph:

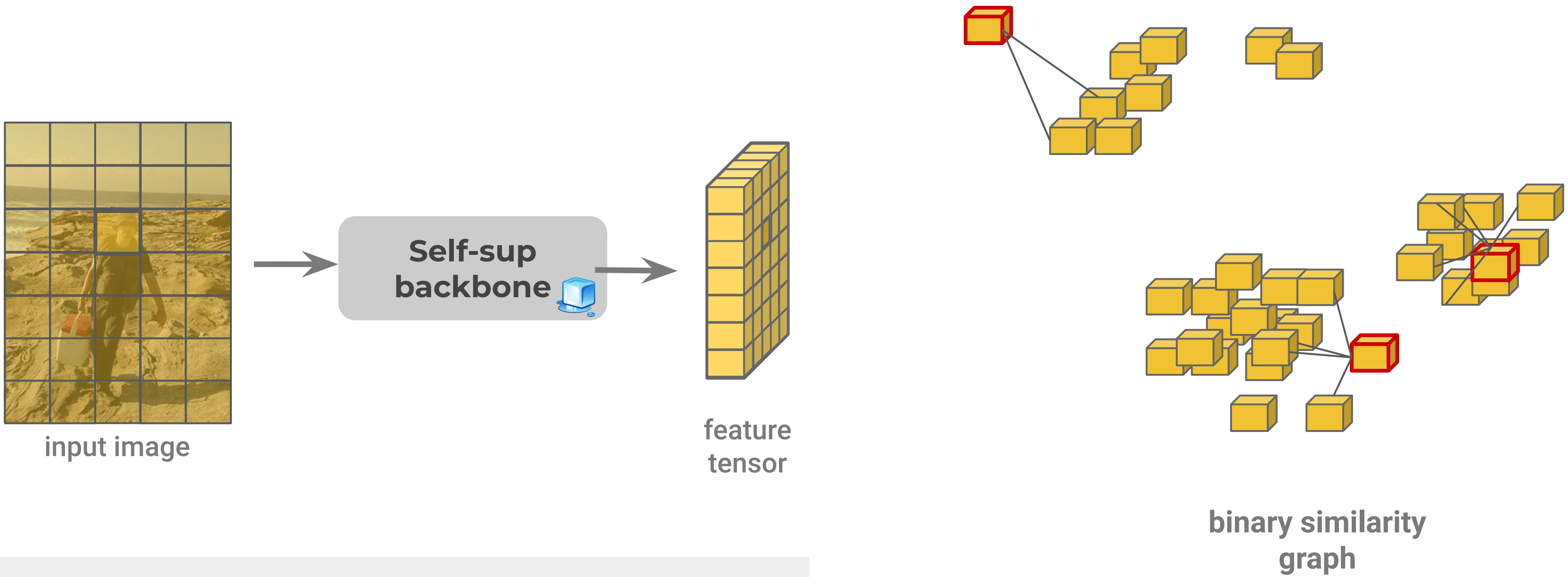
- nodes: patches
- edges: cosine similarity

Build a similarity graph



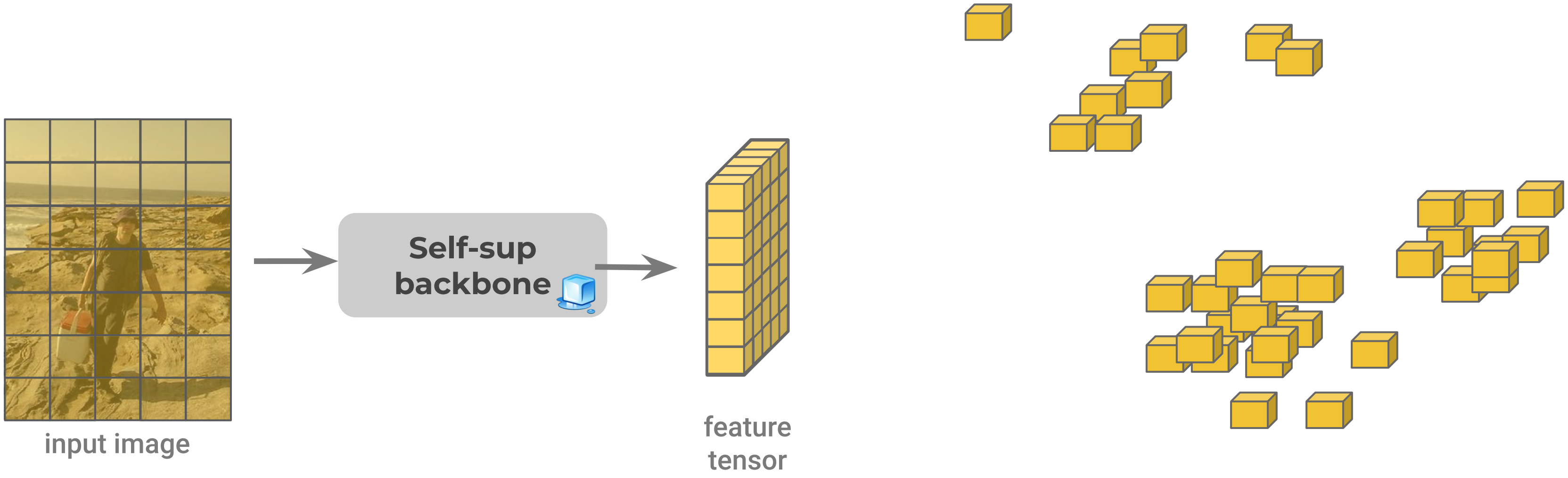
Building graph:
- nodes: patches
- edges: cosine similarity
- connect patches with edges **above a threshold**

Build a similarity graph



Building graph:
- nodes: patches
- edges: cosine similarity
- connect patches with edges **above a threshold**

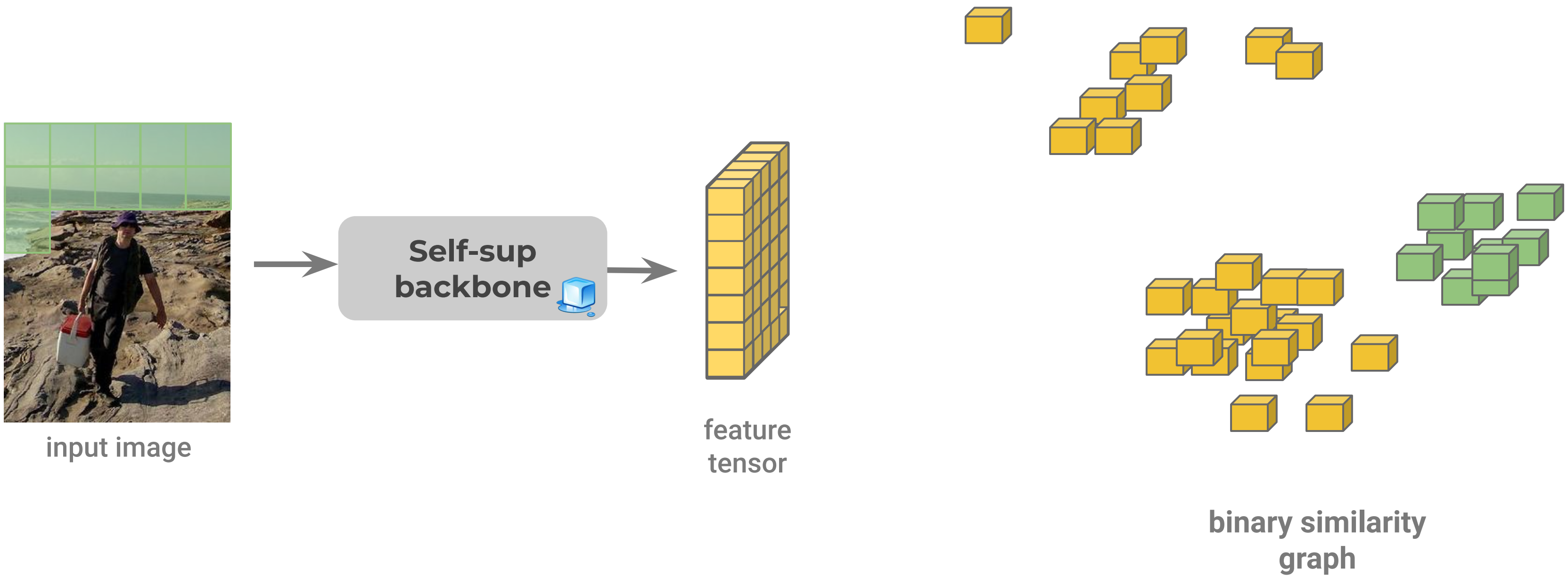
Build a similarity graph



Building graph:
- nodes: patches
- edges: cosine similarity
- connect patches with edges **above a threshold**

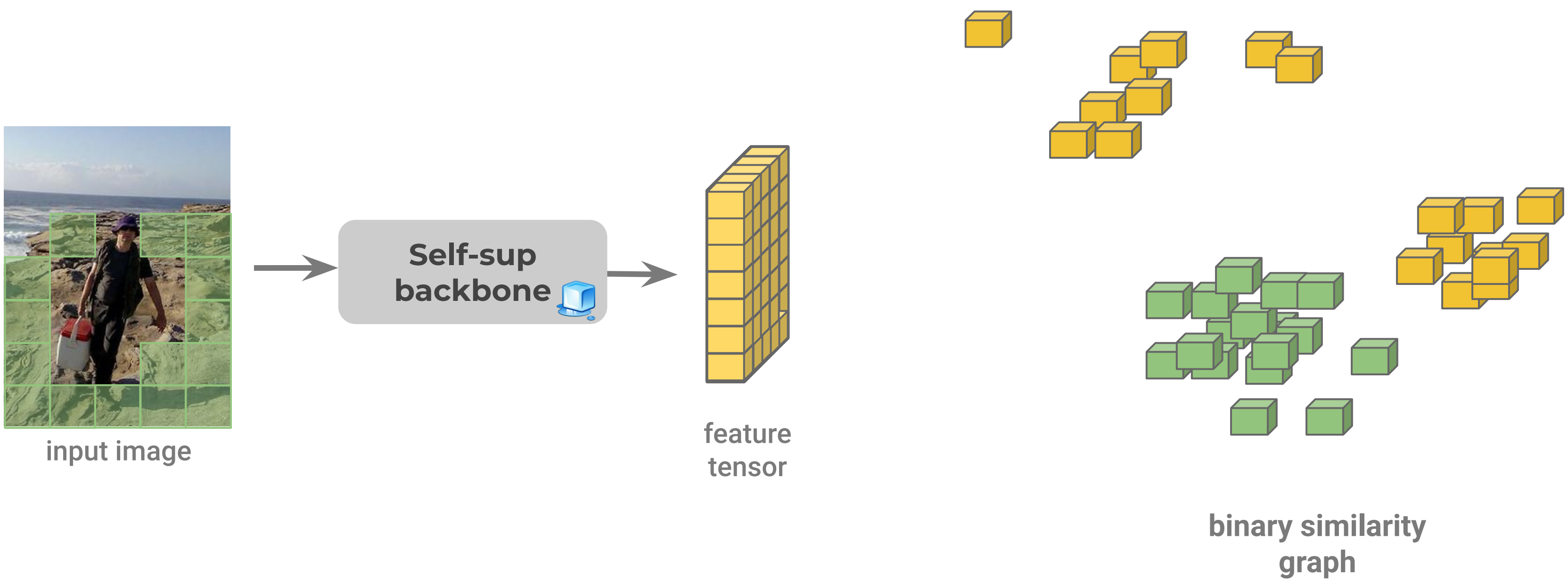
NB: for visualization purposes the edges are not represented

Build a similarity graph

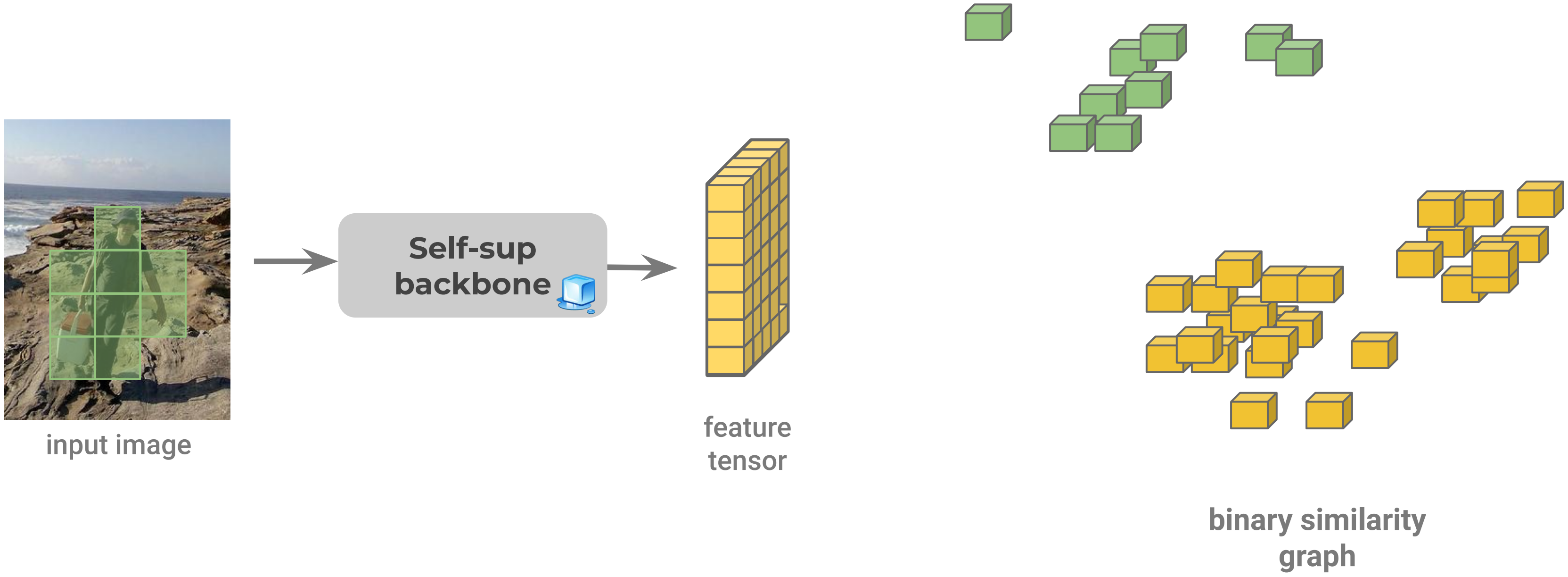


21

Build a similarity graph



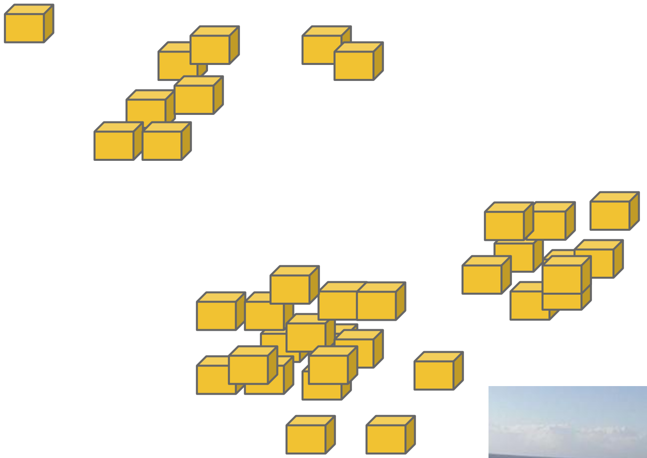
Build a similarity graph



LOST [Siméoni et al. BMVC'21]

Assumptions

- **Foreground** patches are **less correlated** than **background** patches
- **Less patches** of object than background



input image

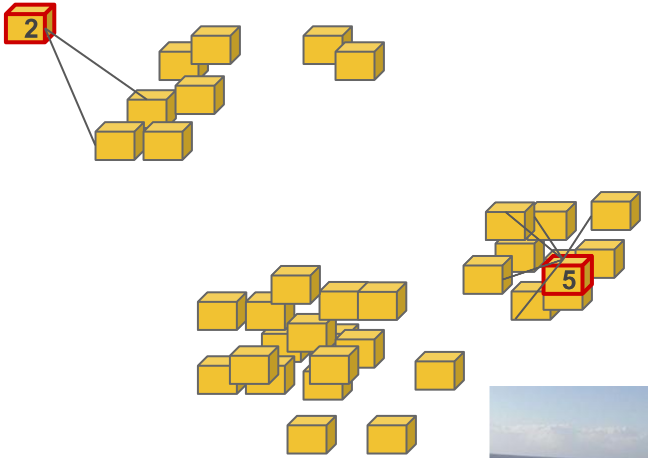
LOST [Siméoni et al. BMVC'21]

Assumptions

- **Foreground** patches are **less correlated** than **background** patches
- **Less patches** of object than background

Concept

- Use the information of **degree**



Degree of a vertex
# of edges that are incident to the vertex

input image

LOST [Siméoni et al. BMVC'21]

Assumptions

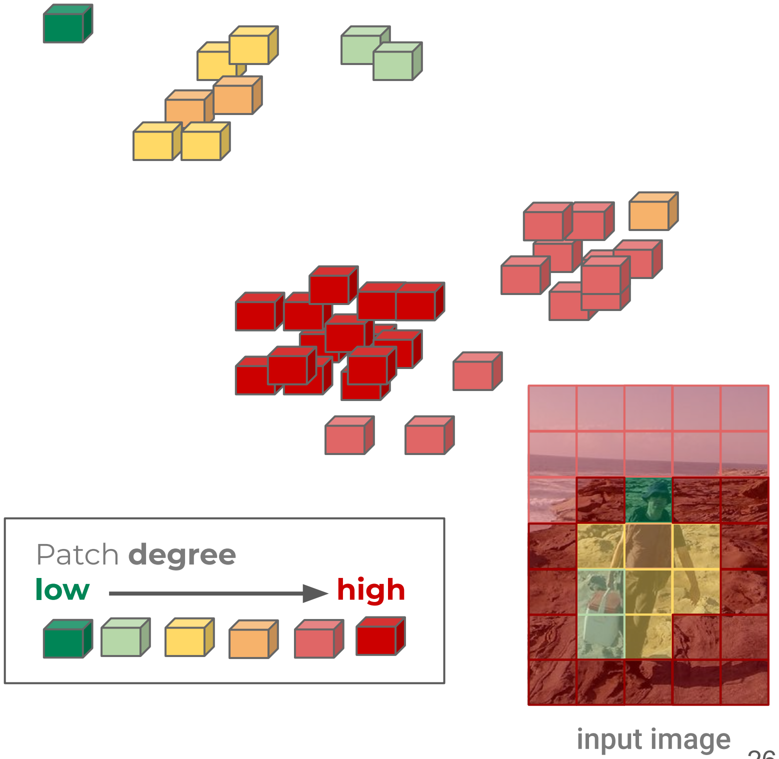
- **Foreground** patches are **less correlated** than **background** patches
- **Less patches** of object than background

Concept

- Use the information of **degree**

Degree of a vertex

# of edges that are incident to the vertex



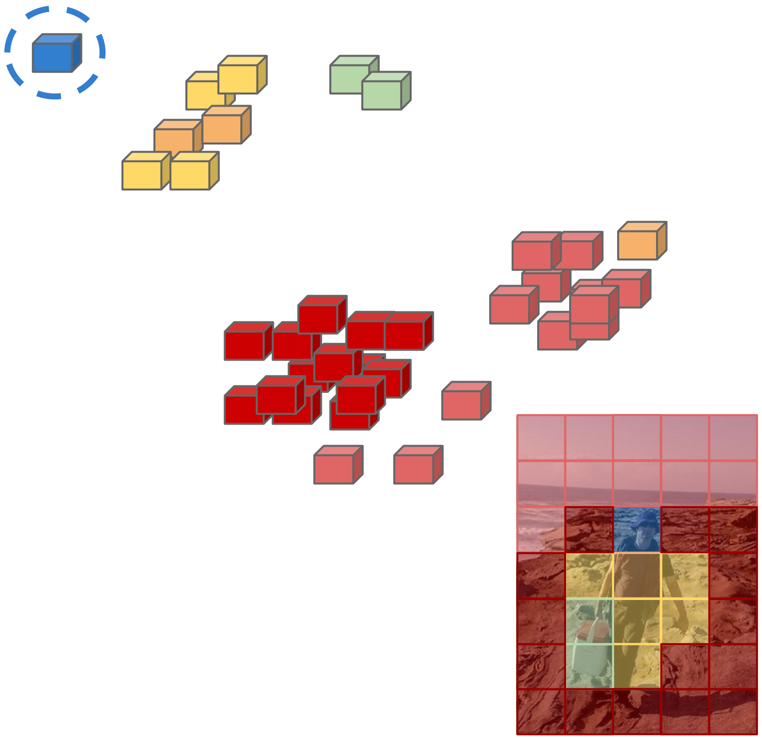
LOST [Siméoni et al. BMVC'21]

Assumptions

- **Foreground** patches are **less correlated** than **background** patches
- **Less patches** of object than background

Concept

- Use the information of **degree**
- **Object seed**: patch with the lowest degree



input image

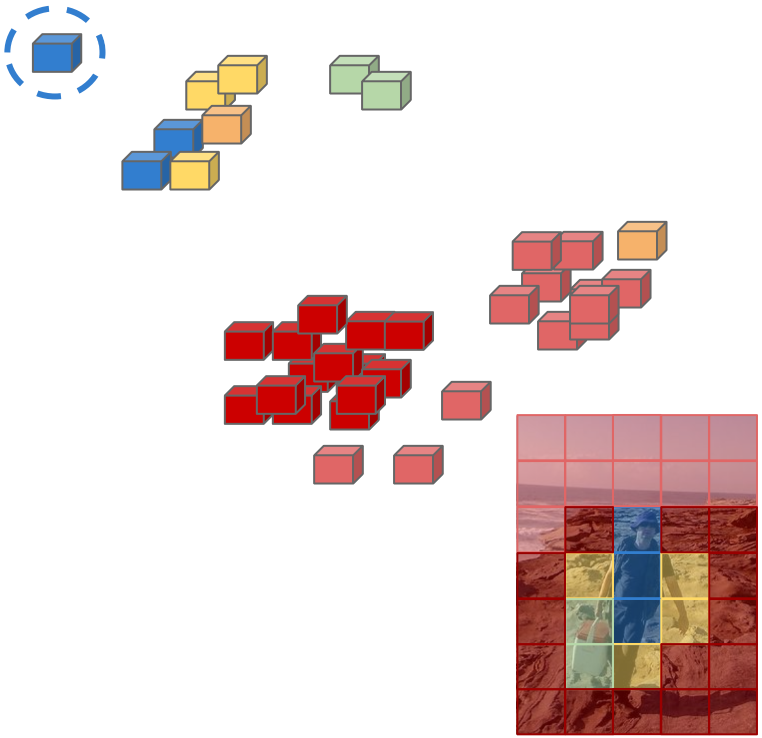
LOST [Siméoni et al. BMVC'21]

Assumptions

- **Foreground** patches are **less correlated** than **background** patches
- **Less patches** of object than background

Concept

- Use the information of **degree**
- **Object seed**: patch with the lowest degree
- Select **similar** patches



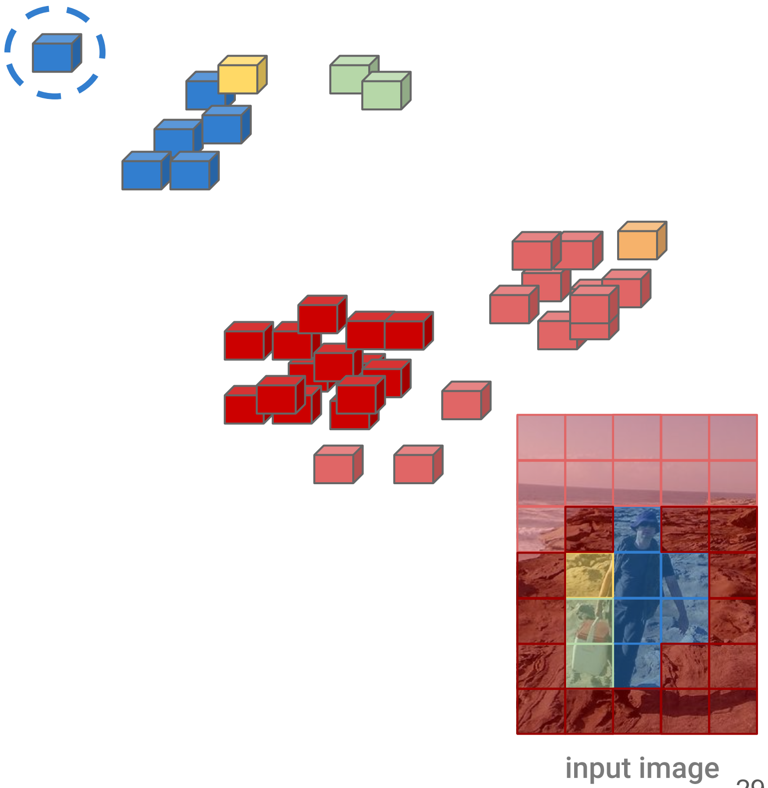
LOST [Siméoni et al. BMVC'21]

Assumptions

- **Foreground** patches are **less correlated** than **background** patches
- **Less patches** of object than background

Concept

- Use the information of **degree**
- **Object seed**: patch with the lowest degree
- Select **similar** patches
- Further **expand** patch region to **similar** patches



LOST [Siméoni et al. BMVC'21]

Assumptions

- **Foreground** patches are **less correlated** than **background** patches
- **Less patches** of object than background

Concept

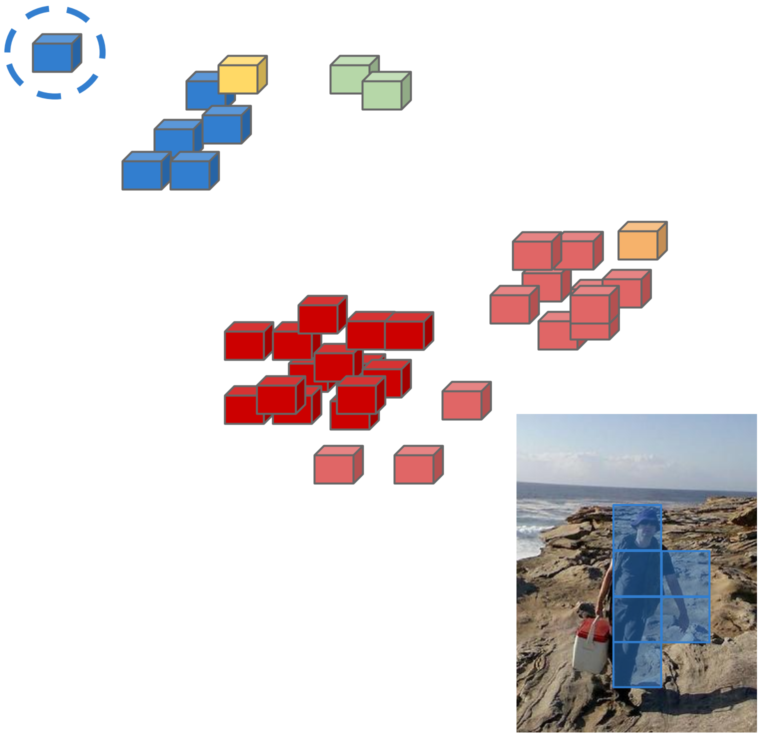
- Use the information of **degree**
- **Object seed**: patch with the lowest degree
- Select **similar** patches
- Further **expand** patch region to **similar** patches

Benefits

+ **Quick** (60 FPS)
+ Better than inter-images methods

Limits

- **Single** object detection
- Issues when object covers most of image
- Coarse



input image

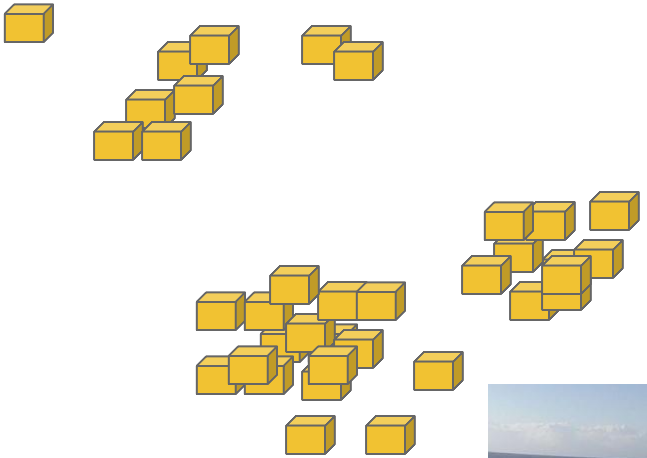
TokenCut [Wang et al. CVPR'22]

Assumptions

- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem



input image

Normalized graph-cut

Find **two sets** in a graph with

- **min degree of similarity** between two sets
- each set with **max degree of similarity** to the **whole** graph

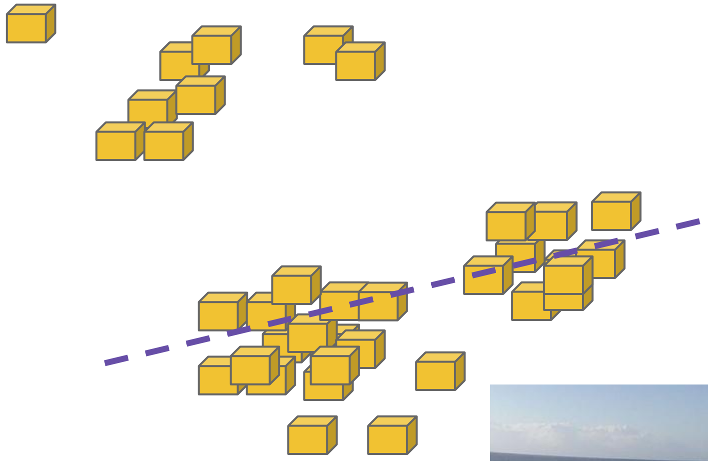
TokenCut [Wang et al. CVPR'22]

Assumptions

- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem



input image

Normalized graph-cut

Find **two sets** in a graph with

- **min degree of similarity** between two sets
- each set with **max degree of similarity** to the **whole graph**

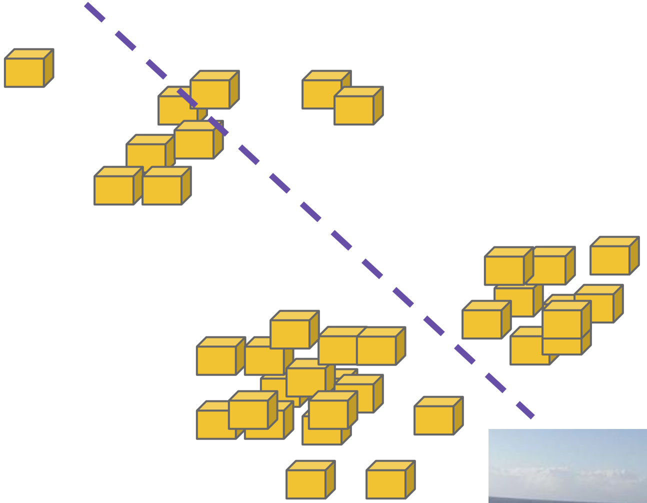
TokenCut [Wang et al. CVPR’22]

Assumptions

- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem



input image

Normalized graph-cut

Find **two sets** in a graph with

- **min degree of similarity** between two sets
- each set with **max degree of similarity** to the **whole** graph

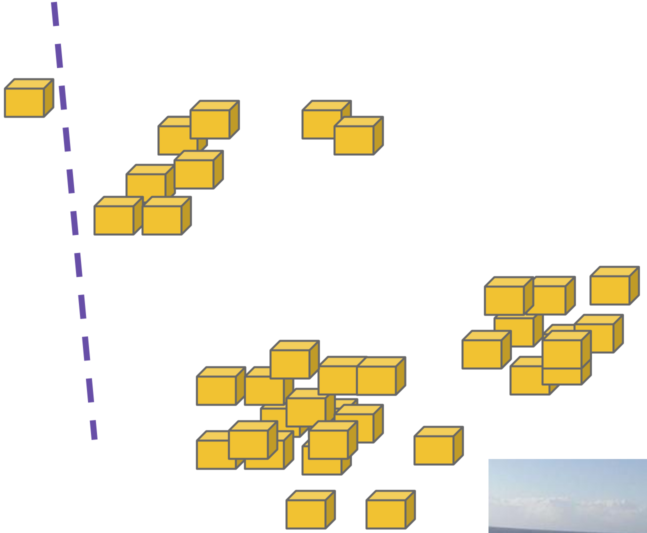
TokenCut [Wang et al. CVPR'22]

Assumptions

- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem



input image

Normalized graph-cut

Find **two sets** in a graph with

- **min degree of similarity** between two sets
- each set with **max degree of similarity** to the **whole** graph

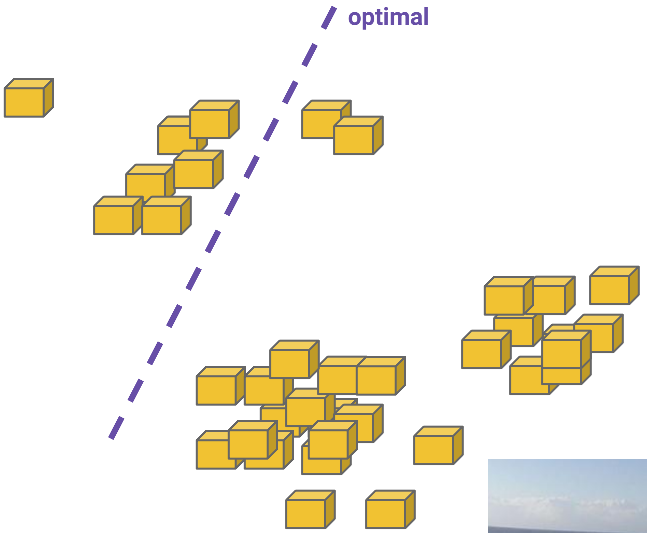
TokenCut [Wang et al. CVPR'22]

Assumptions

- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem



Normalized graph-cut

Find **two sets** in a graph with

- **min degree of similarity** between two sets
- each set with **max degree of similarity** to the **whole graph**

input image

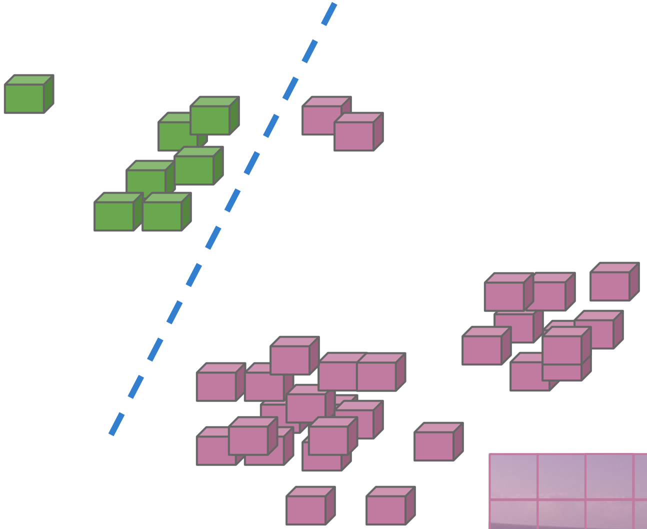
TokenCut [Wang et al. CVPR'22]

Assumptions

- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem
  - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?

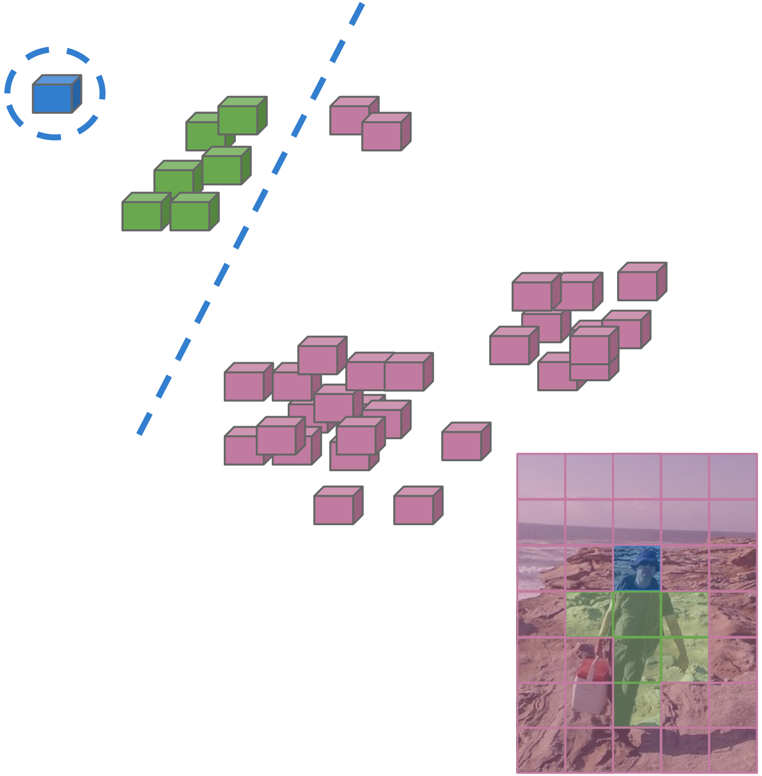



input image

TokenCut [Wang et al. CVPR’22]

Assumptions
- Foreground objects can then be segmented to **group self-similar region**

Concept
- Solve a **normalized graph-cut** problem
  - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?
- **Select** the **set** containing the **least connected patch**



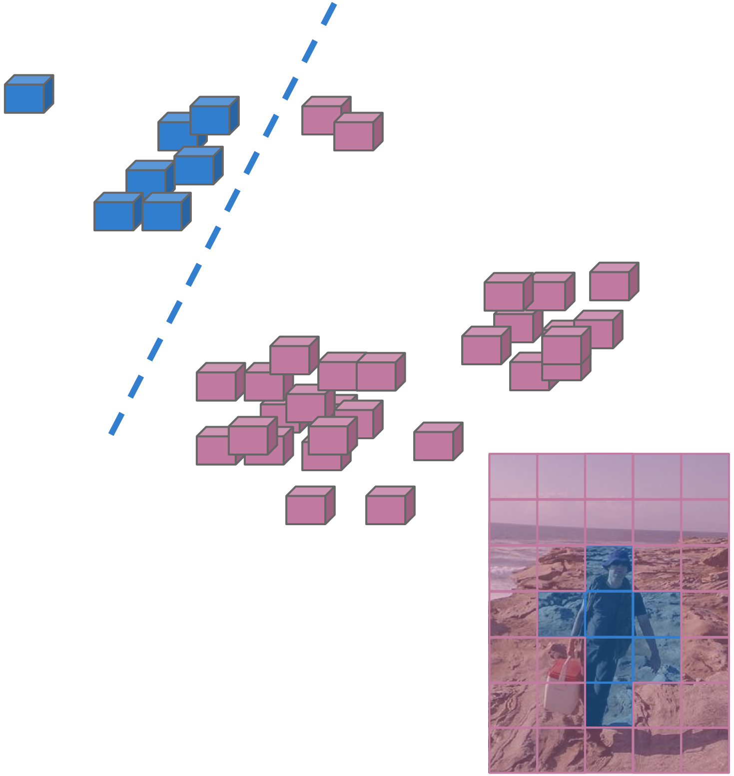
TokenCut [Wang et al. CVPR'22]

Assumptions

- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem
    - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?
- **Select** the **set** containing the **least connected patch**

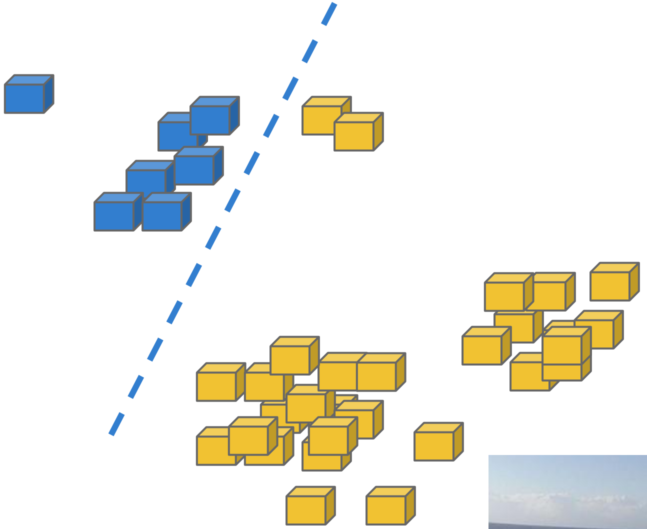


input image

TokenCut [Wang et al. CVPR'22]

Assumptions
- Foreground objects can then be segmented to **group self-similar region**

Concept
- Solve a **normalized graph-cut** problem
  - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?
- **Select** the **set** containing the **least connected patch**



input image

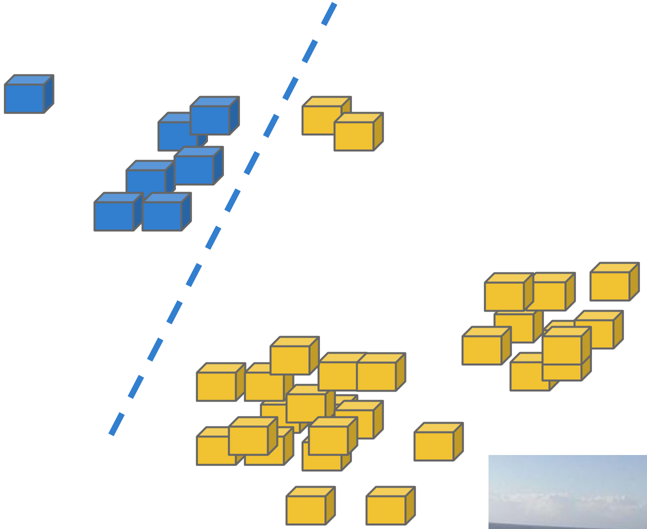
TokenCut [Wang et al. CVPR'22]

Assumptions

- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem
  - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?
- **Select** the **set** containing the **least connected patch**



input image

Benefits

+ More refined **localization**
+ Better than inter-images methods
+ Generalizability to =/= feats

Limits

- **Single** object detection
- Requires to compute eigenvector → slower

Related work

- **Deep Spectral Methods** [Melas-Kyriazi et al. CVPR'22]

Additional idea: integrate pixel-level features in the graph

TokenCut [Wang et al. CVPR'22] ⟶ MaskCut [X. Wang et al. CVPR'23]

Assumptions

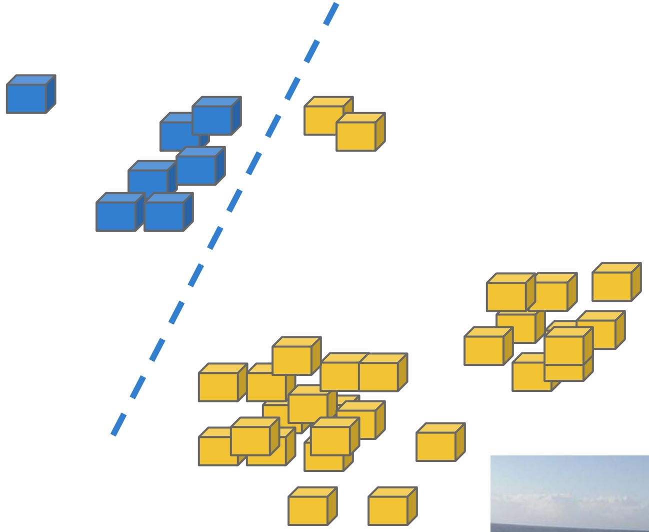
- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem
  - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?
- **Select** the **set** containing the **least connected patch**

Extension

- More than one object can be found
- **Remove the already discovered** nodes from the graph and **repeat the operation**



input image

TokenCut [Wang et al. CVPR'22] ⟶ MaskCut [X. Wang et al. CVPR'23]

Assumptions

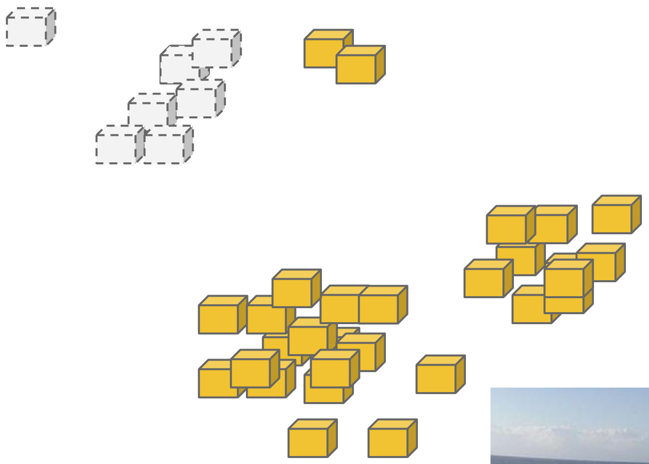
- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem
  - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?
- **Select** the **set** containing the **least connected patch**

Extension

- More than one object can be found
- **Remove the already discovered** nodes from the graph and **repeat the operation**



input image

TokenCut [Wang et al. CVPR'22] ⟶ MaskCut [X. Wang et al. CVPR'23]

Assumptions

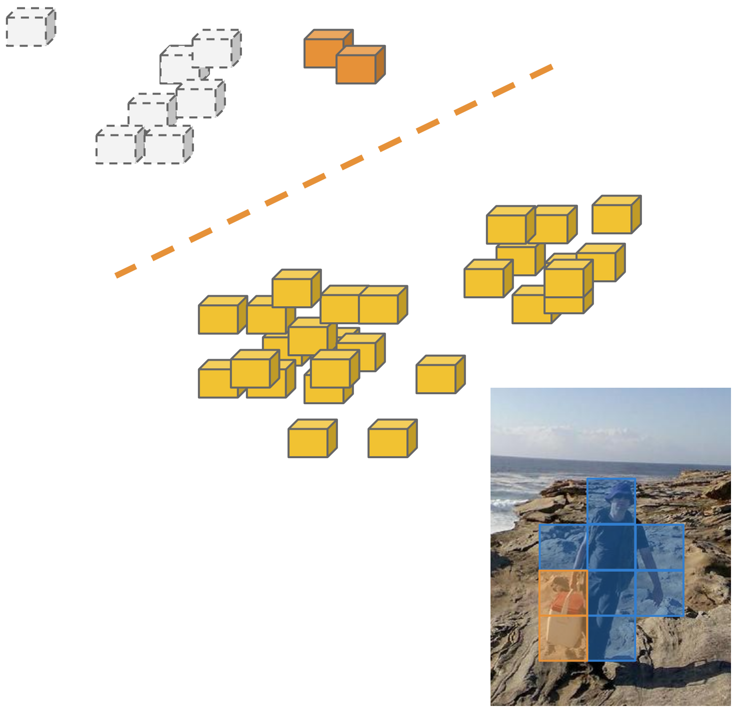
- Foreground objects can then be segmented to **group self-similar region**

Concept

- Solve a **normalized graph-cut** problem
  - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?
- **Select** the **set** containing the **least connected patch**

Extension

- More than one object can be found
- **Remove the already discovered** nodes from the graph and **repeat the operation**



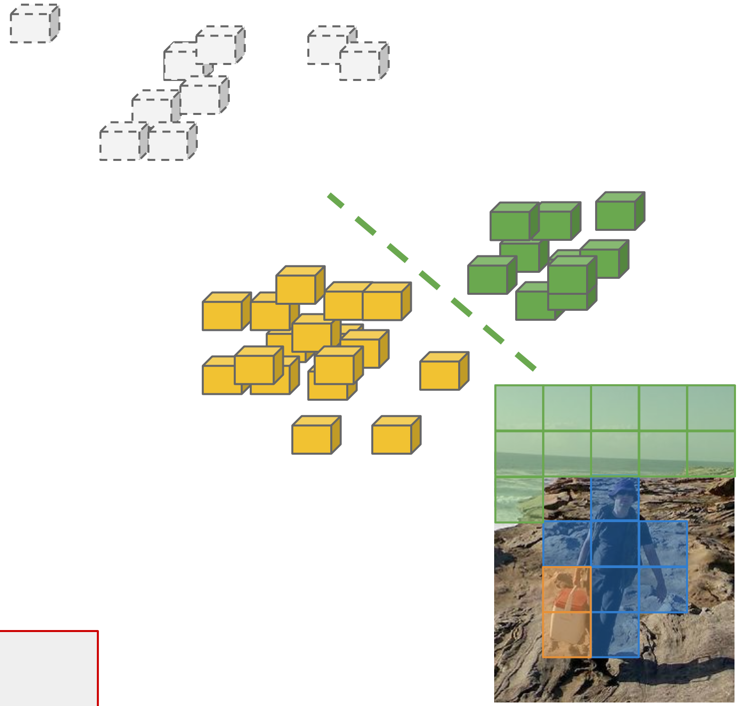
input image

TokenCut [Wang et al. CVPR'22] ⟶ MaskCut [X. Wang et al. CVPR'23]

Assumptions
- Foreground objects can then be segmented to **group self-similar region**

Concept
- Solve a **normalized graph-cut** problem
  - Solved with **spectral clustering**
- Given the bi-partition, **which is the object** ?
- **Select** the **set** containing the **least connected patch**

Extension
- More than one object can be found
- **Remove the already discovered** nodes from the graph and **repeat the operation**



Benefits
+ **Several** objects
+ More refined **localization**
+ Better than inter-images methods

Limits
- Are they **all objects ?**
- Requires to compute eigenvector + iterative process → slower

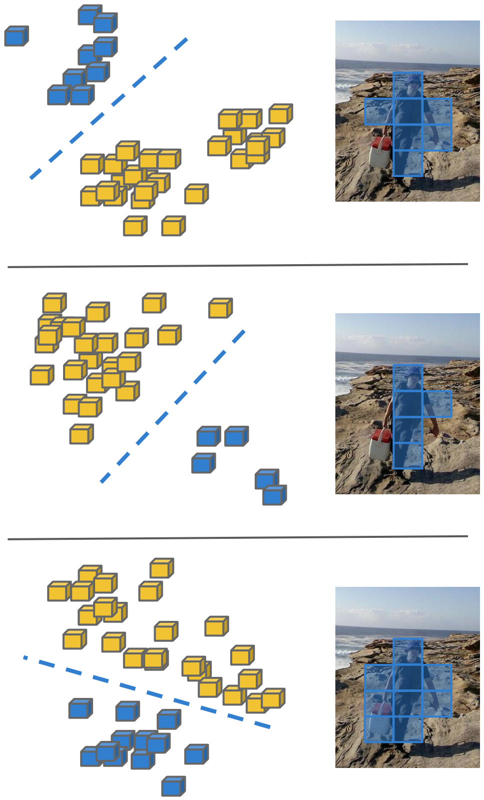
SelfMask [Shin et al. CVPRW’22]

Assumptions

- **Different self-supervised features** entangle different information about foreground/background

Concept

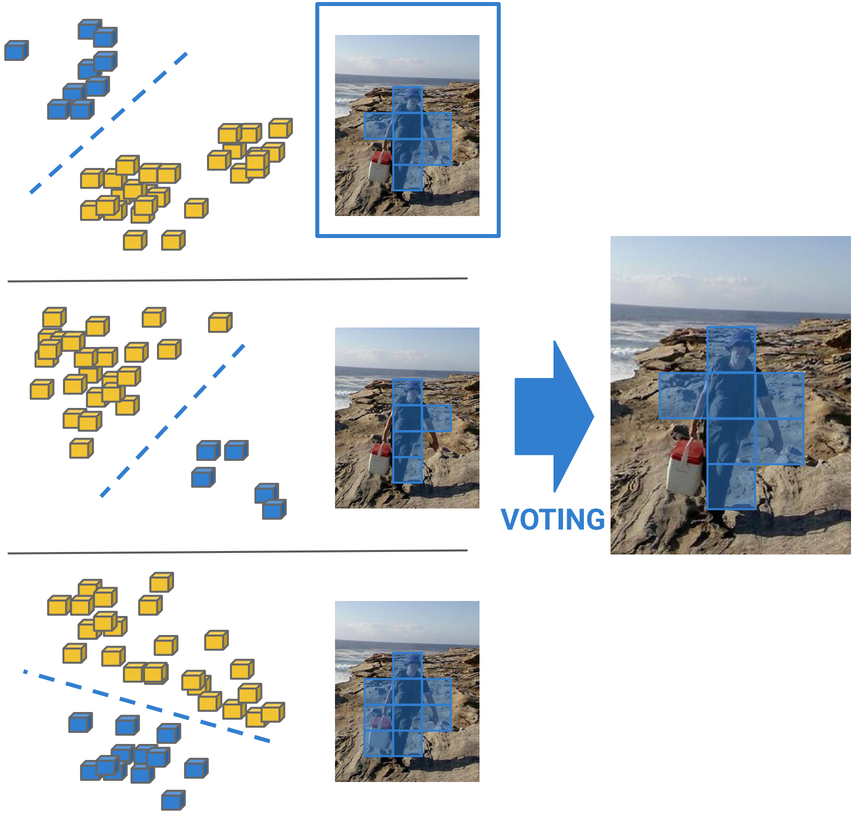
- Consider different self-supervised features
- Use **spectral clustering** to produce masks with different number of clusters



SelfMask [Shin et al. CVPRW'22]

Assumptions
- **Different self-supervised features** entangle different information about foreground/background

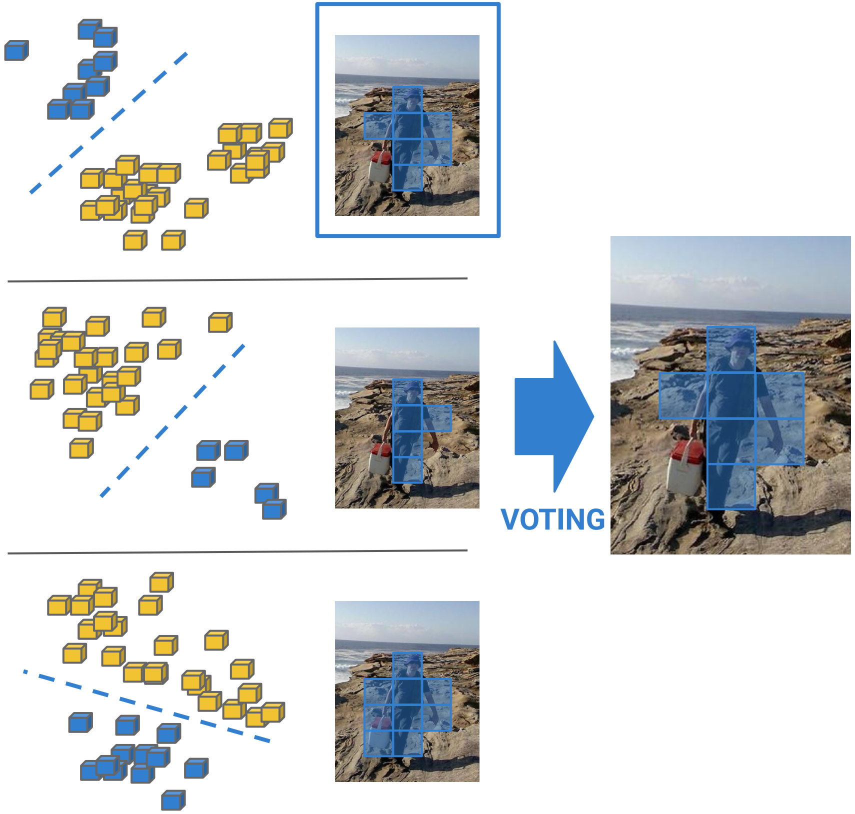
Concept
- Consider different self-supervised features
- Use **spectral clustering** to produce masks with different number of clusters
- Vote for the **best** candidate
    - Mask with **highest IoU similarity** to all others



SelfMask [Shin et al. CVPRW'22]

Assumptions
- **Different self-supervised features** entangle different information about foreground/background

Concept
- Consider different self-supervised features
- Use **spectral clustering** to produce masks with different number of clusters
- Vote for the **best** candidate
  - Mask with **highest IoU similarity** to all others



Benefits
+ Leverages **several self-supervised features**
+ Better than inter-images methods

Limits
- **Single** object detection
- **Several** forward passes
- Requires to compute eigenvector → slower

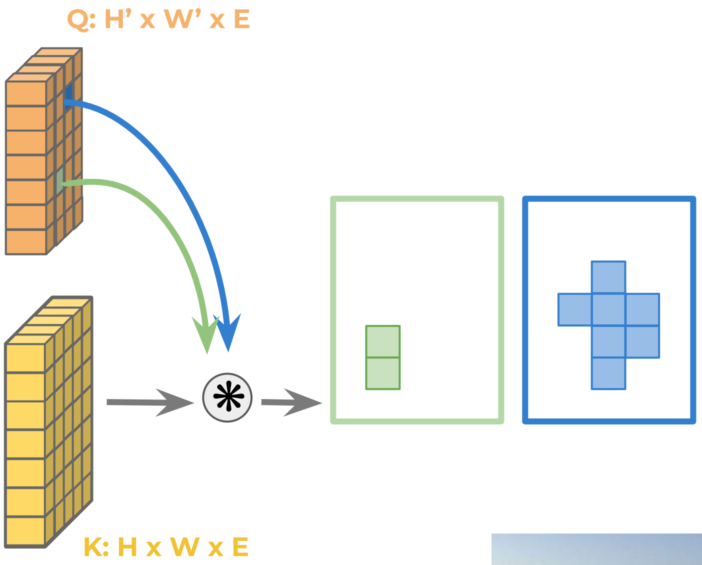
FreeMask [Xinlong Wang et al. CVPR'22]

Assumptions

- Attention can be directly used to produce masks

Concept

- Generate **many coarse masks** (one per query)
- Then sort and select using **NMS-like function**
- Use different **scales**



input image

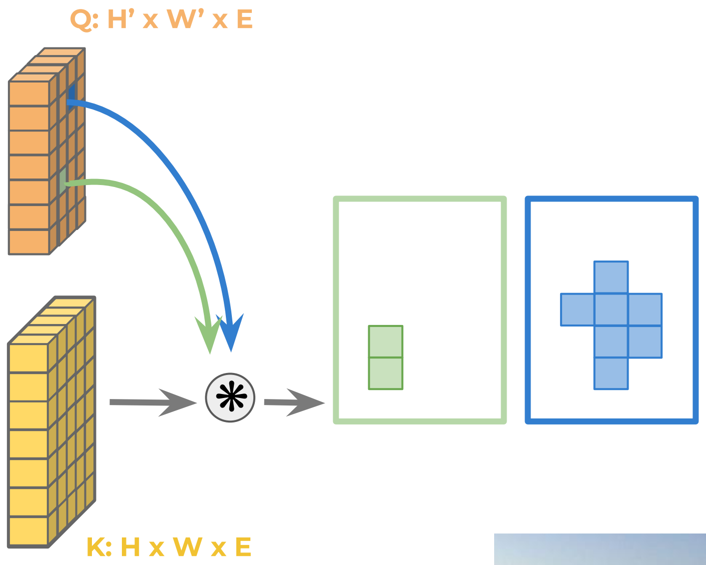
FreeMask [Xinlong Wang et al. CVPR'22]

Assumptions
- Attention can be directly used to produce masks

Concept
- Generate **many coarse masks** (one per query)
- Then sort and select using **NMS-like function**
- Use different **scales**

Benefits
+ **Several** objects
+ Getting closer to **instance**
+ Better than inter-images methods

Limits
- Masks for **not objects**
- Hard to filter out the bad masks




input image

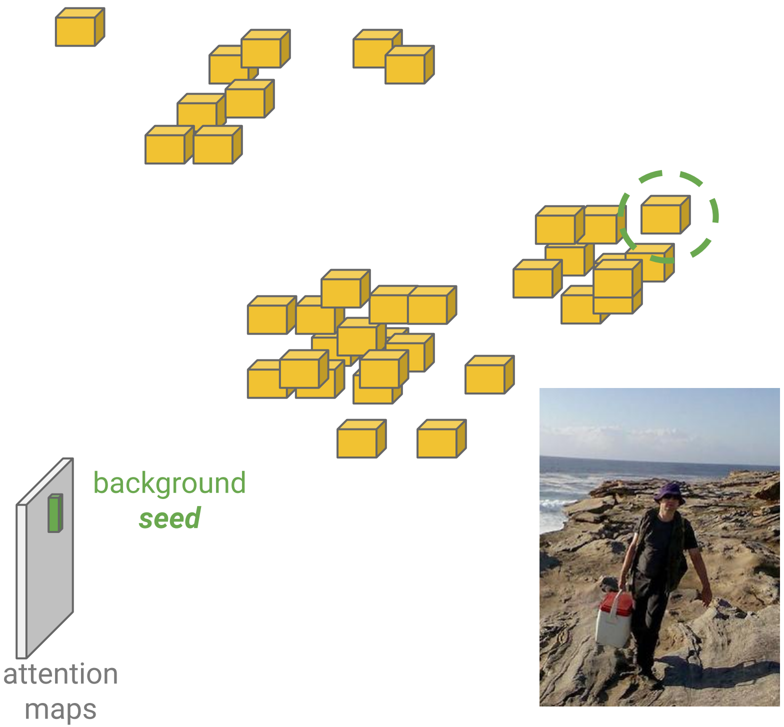
FOUND [Siméoni et al. CVPR’23]

Assumptions

- **Look for the background** instead of objects → no hypothesis needed about objects
- Background receives **little attention** in SSL features

Concept

- Find the **background seed** = patch with **least attention**



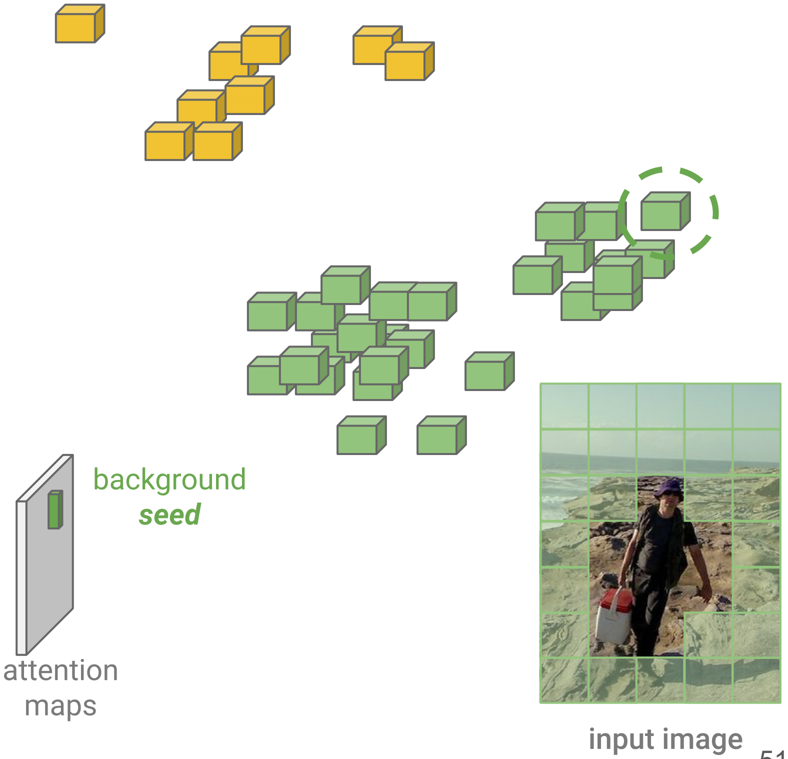
FOUND [Siméoni et al. CVPR'23]

Assumptions

- **Look for the background** instead of objects → no hypothesis needed about objects
- Background receives **little attention** in SSL features

Concept

- Find the **background seed** = patch with **least attention**
- Select all similar patches = **background mask**



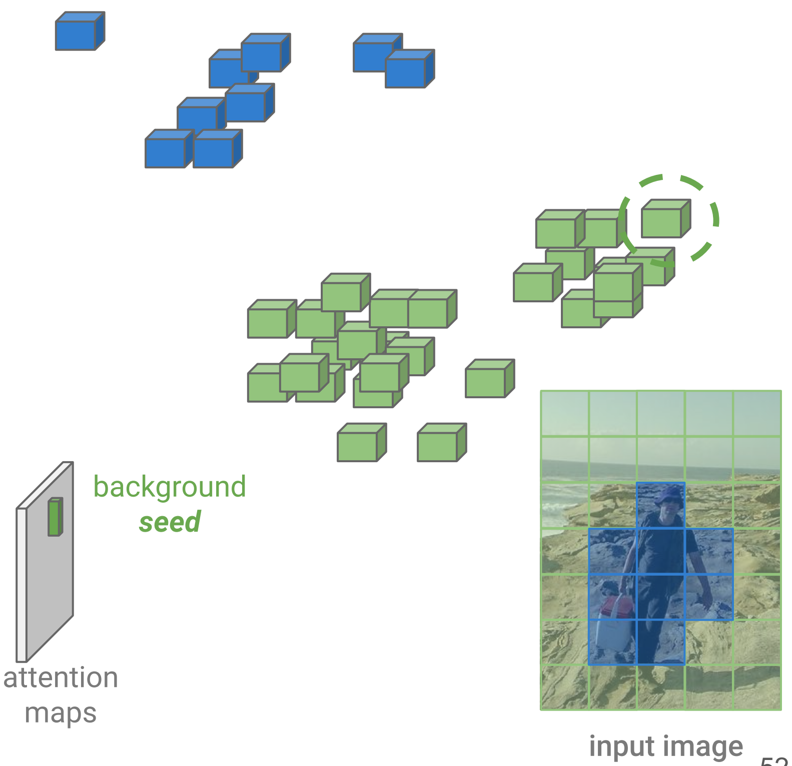
FOUND [Siméoni et al. CVPR'23]

Assumptions

- **Look for the background** instead of objects → no hypothesis needed about objects
- Background receives **little attention** in SSL features

Concept

- Find the **background seed** = patch with **least attention**
- Select all similar patches = **background mask**
- **Foreground** = complement of background



FOUND [Siméoni et al. CVPR'23]

Assumptions

- **Look for the background** instead of objects → no hypothesis needed about objects
- Background receives **little attention** in SSL features

Concept

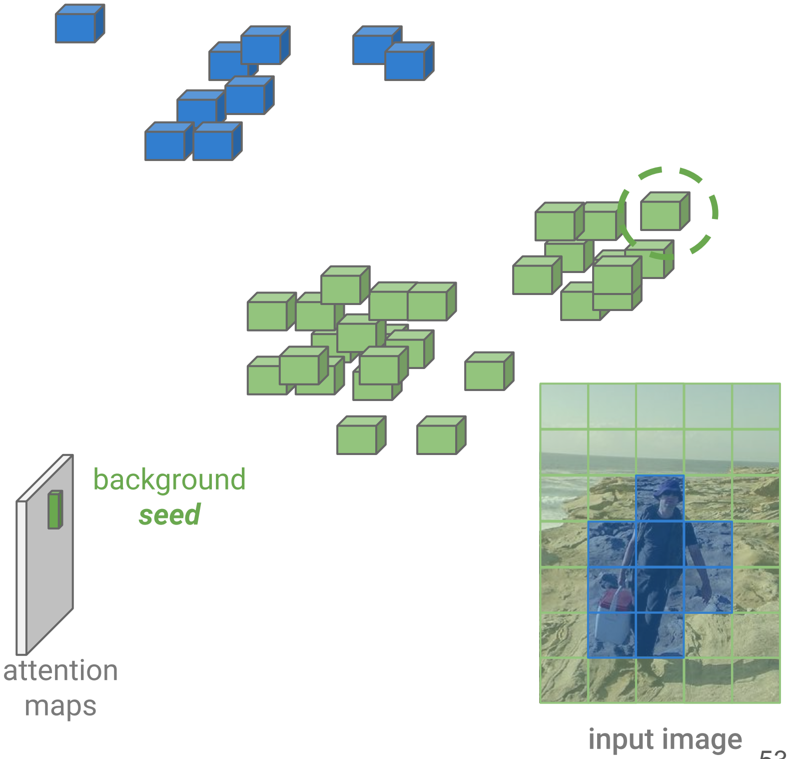
- Find the **background seed** = patch with **least attention**
- Select all similar patches = **background mask**
- **Foreground** = complement of background

Benefits

+ Localize **several object**
+ Quick to compute
+ Better than inter-images methods

Limits

- **No clear instance**
- Coarse



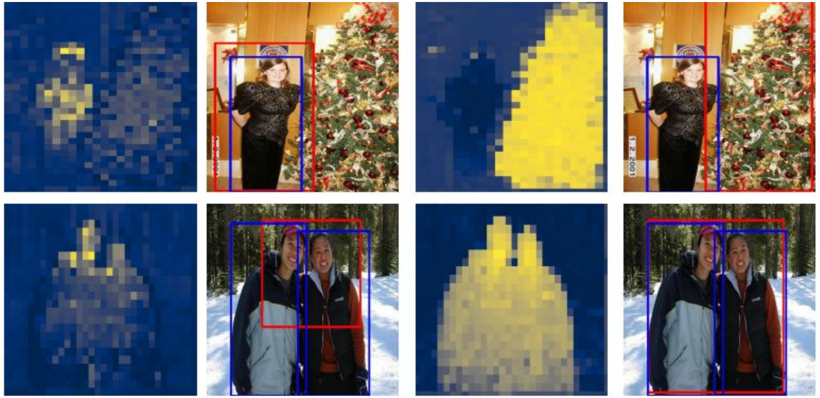
Some results

Is the box corresponding to a ground-truth box ?

Method	VOC07	VOC12	COCO20k
— No learning —			
Selective Search [47]	18.8	20.9	16.0
EdgeBoxes [76]	31.1	31.6	28.8
Kim et al. [26]	43.9	46.4	35.1
Zhang et al. [70]	46.2	50.5	34.8
DDT+ [60]	50.2	53.1	38.2
rOSD [50]	54.5	55.3	48.5
LOD [53]	53.6	55.1	48.5
DINO-seg [6] [45] (ViT-S/16 [6])	45.8	46.2	42.0
LOST [45] (ViT-S/8 [6])	55.5	57.0	49.5
LOST [45] (ViT-S/16 [6])	61.9	64.0	50.7
DSS [34] (ViT-S/16 [6])	62.7	66.4	52.2
TokenCut [59] (ViT-S/8 [6]) †	67.3	71.6	60.7
TokenCut [59] (ViT-S/16 [6])	68.8	72.1	58.8

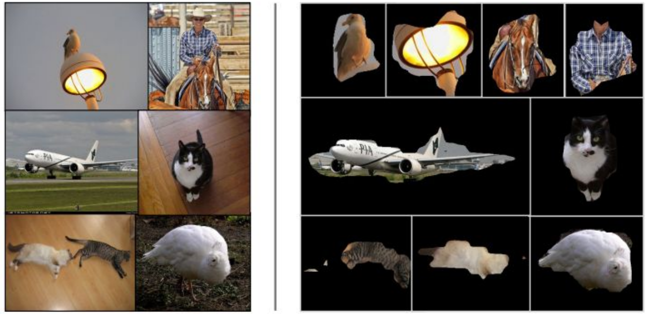
Using self-sup.

metric: corloc



(a) LOST Inverse Attn. (b) LOST Detection (c) Our Eigen Attention (d) Our Detection

TokenCut [Wang et al. CVPR'22]



Unlabeled images Free Mask output

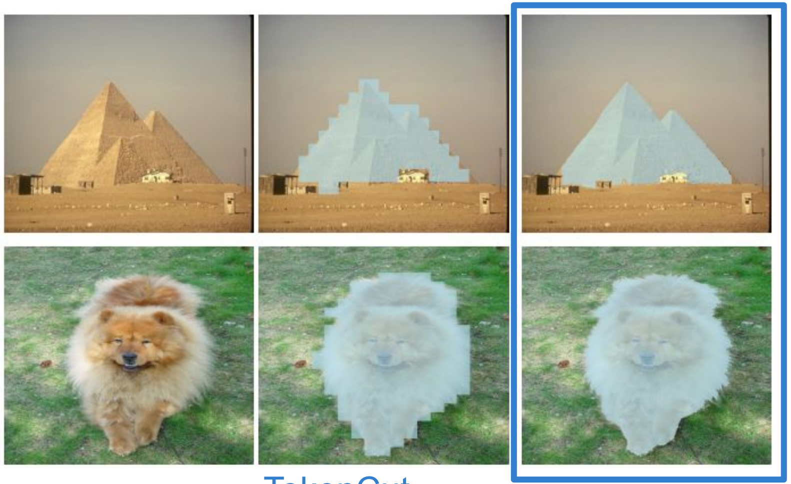
FreeMask [Xinlong Wang et al. CVPR'22]

Simple ways to refine prediction

- Fit the masks to **pixel-level** information
- Use **Bilateral Solver** (BS) or **Conditional Random Field** (CRF)
- Require **no-training**

Limits

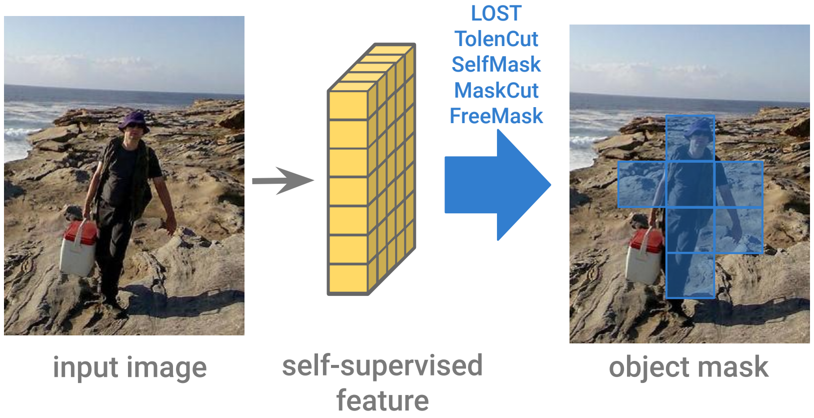
- **Rather** slow post-processing



TokenCut [X et al. CVPR'22]

+ BS

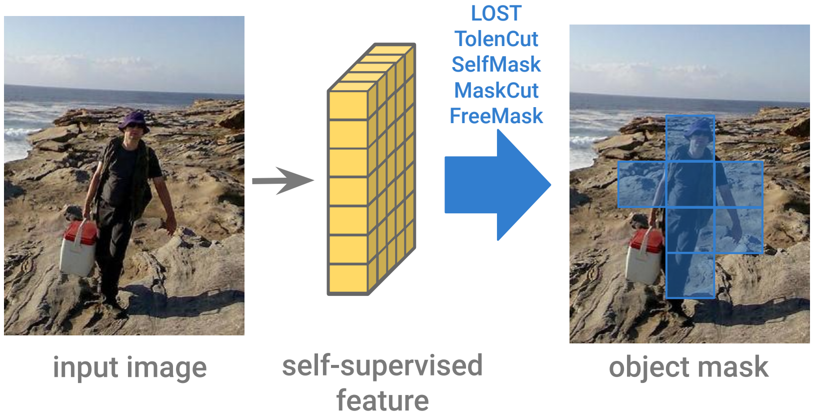
Take-away



Conclusions

- Possible to discover **objects** with **no annotation**
- Easy to discover *a single object*, generalizing to **several is harder**
- Interesting performances on VOC/COCO dataset

Take-away



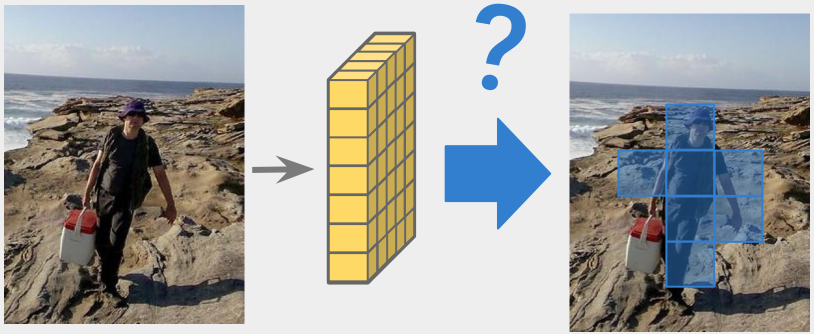
Conclusions

- Possible to discover **objects** with **no annotation**
- Easy to discover *a single object*, generalizing to **several is harder**
- Interesting performances on VOC/COCO dataset

Remaining issues

- How to successfully perform **multi-object detection ?**
- How to exchange information at a **dataset level** ?
- How to **refine** results ?

Presentation outline

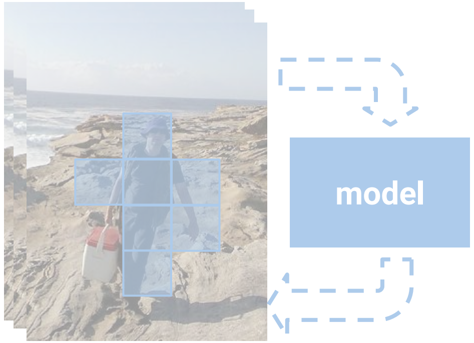


input image

self-supervised feature

object mask

Unsupervised object discovery
Generation of initial masks

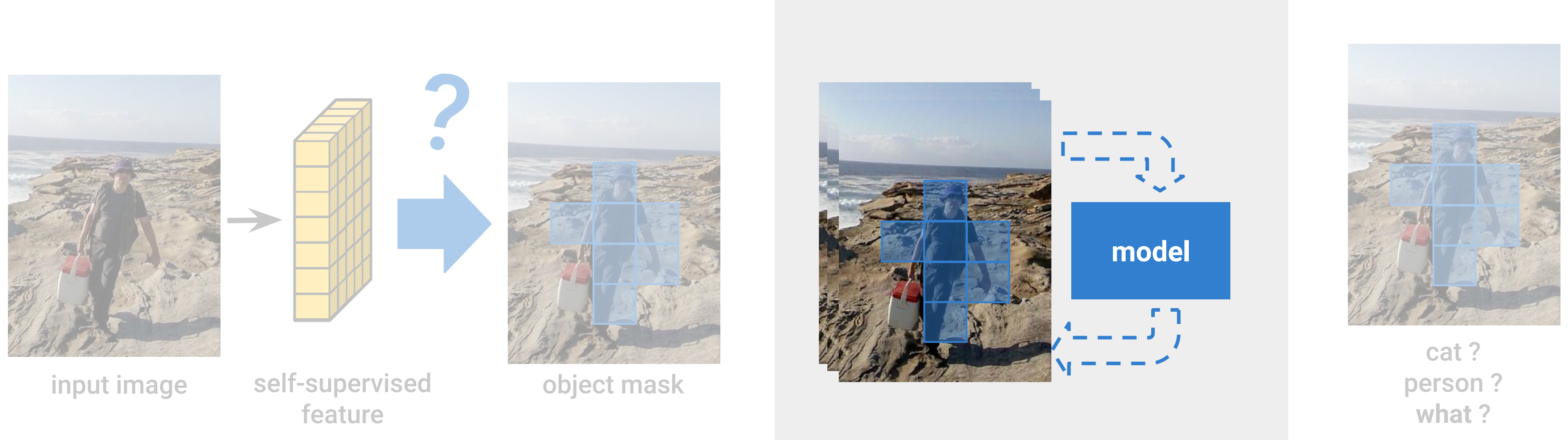


Improving through learning

cat ?
person ?
what ?

Class-aware ?

Presentation outline



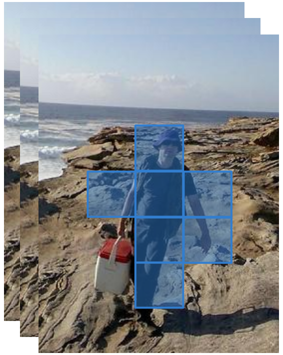
Unsupervised object discovery
Generation of initial masks

Improving through learning

Class-aware ?

Improving through learning

Unsupervised object discovery



pseudo-labels

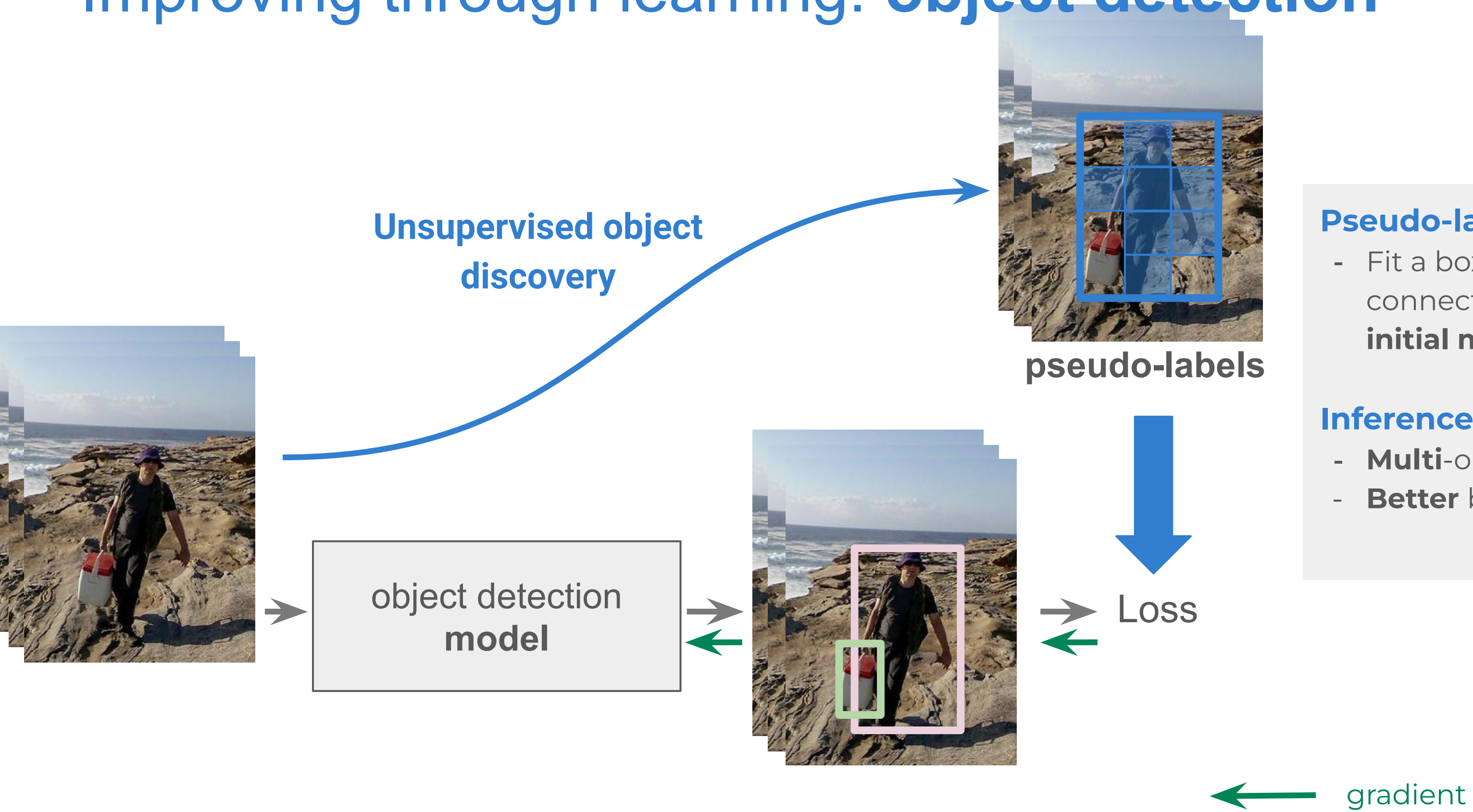
model

Loss

Assumptions

- Allows to go **from single** object discovery **to multiple** object localization
- Training helps to **smooth out mistakes** from initial localization

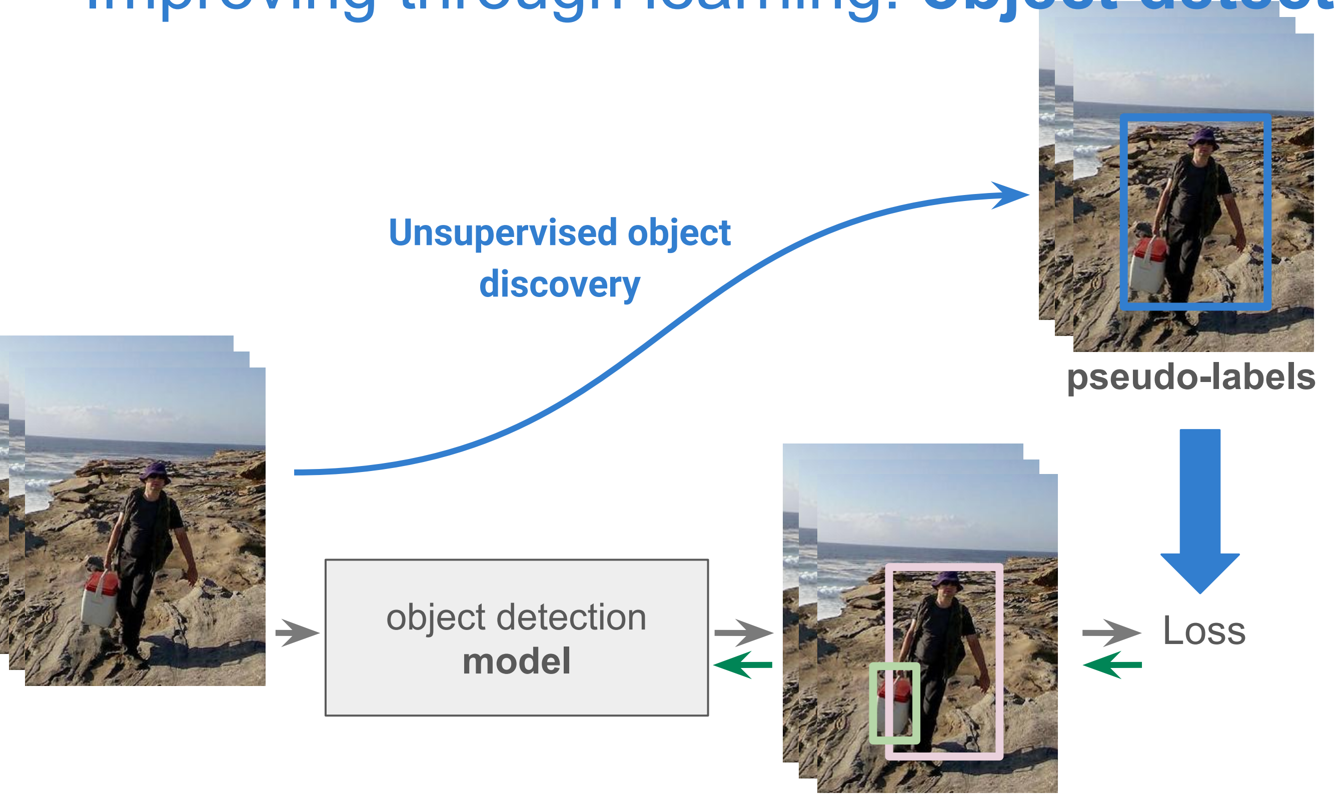
Improving through learning: **object detection**



Pseudo-labels:
- Fit a box to the biggest connected component in the **initial mask**

Inference:
- **Multi**-object detection
- **Better** boxes

Improving through learning: **object detection**



gradient

Pseudo-labels:
- Fit a box to the biggest connected component in the **initial mask**

Inference:
- **Multi**-object detection
- **Better** boxes

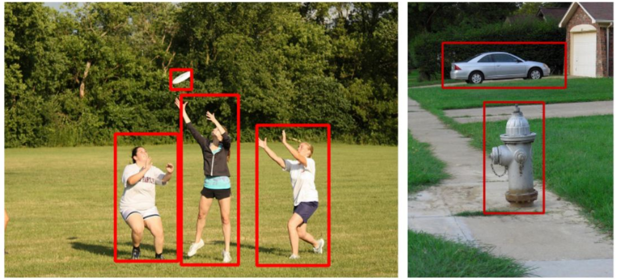
+CAD (Class-Agnostic Detector) [Siméoni et al. BMVC'21]

Concept

- Train an object detector model in a **class-agnostic** fashion (**+CAD**)
- Train a **Faster-RCNN** [Ren et al. NeurIPS'15] **without adaptation**

Benefit

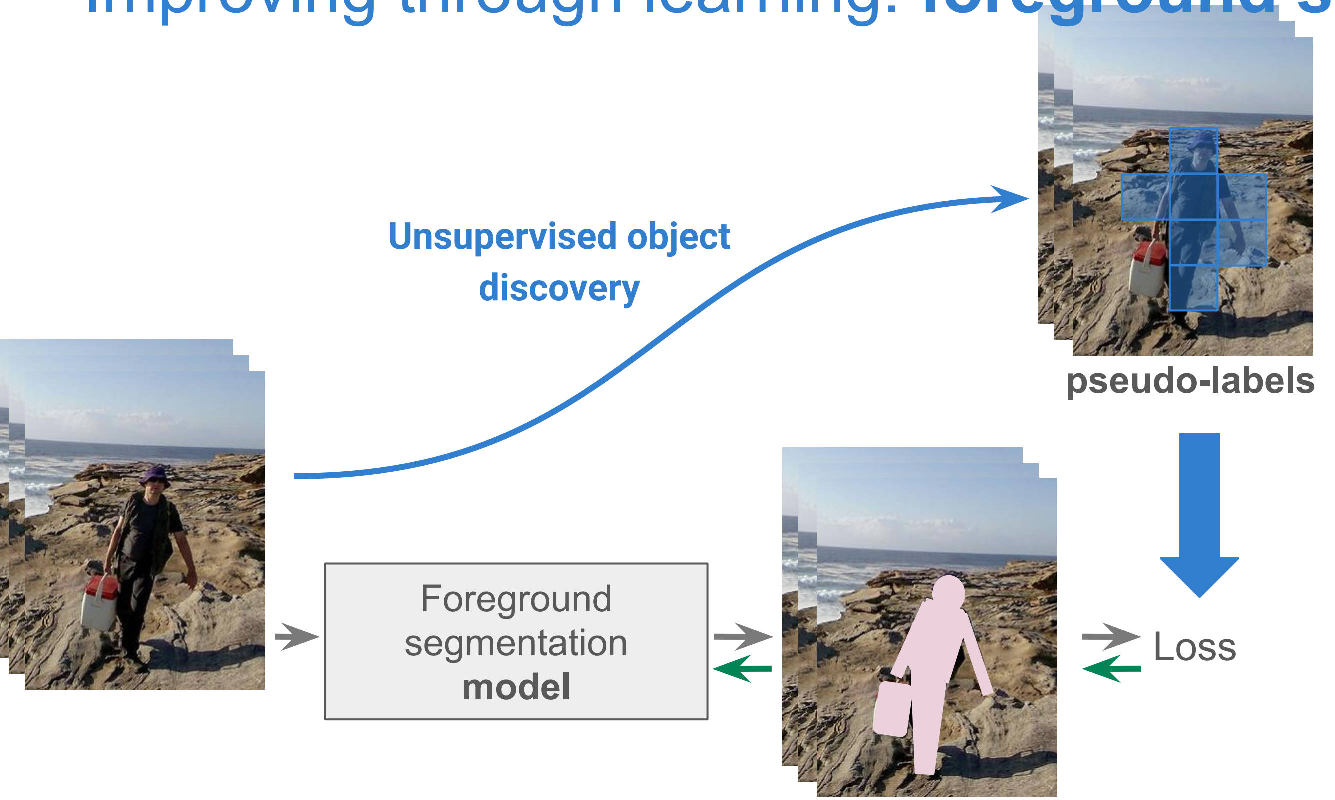
- From **single** to **multi-object** detection



+ CAD

Method	VOC07 [19]	VOC12 [20]	COCO20K [33, 56]
Selective Search [45, 51]	18.8	20.9	16.0
EdgeBoxes [45, 79]	31.1	31.6	28.8
Kim et al. [30, 45]	43.9	46.4	35.1
Zhange et al. [45, 74]	46.2	50.5	34.8
DDT+ [45, 61]	50.2	53.1	38.2
rOSD [45, 56]	54.5	55.3	48.5
LOD [45, 57]	53.6	55.1	48.5
DINO-seg [6, 45]	45.8	46.2	42.1
LOST [45]	61.9	64.0	50.7
TokenCut	**68.8** (↑ 6.9)	**72.1** (↑ 8.1)	**58.8** (↑ 8.1)
LOD + CAD* [45]	56.3	61.6	52.7
rOSD + CAD* [45]	58.3	62.3	53.0
LOST + CAD* [45]	65.7	70.4	57.5
TokenCut + CAD* [45]	**71.4** (↑ 5.7)	**75.3** (↑ 4.9)	**62.6** (↑ 5.1)
metric: corloc	+ 3.3/2.5	+ 6.4/3.2	+ 7.0/3.8

Improving through learning: **foreground segmentation**

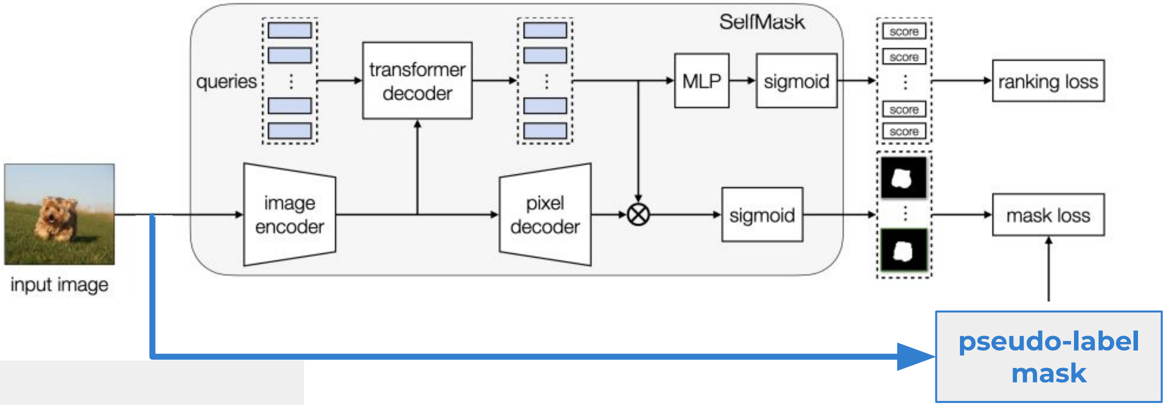


Pseudo-labels:
- The **initial mask**

Inference:
- Produce a foreground mask per image

SelfMask [Shin et al. CVPRW'22]

Concept:
- Learn an **encoder/decoder** architecture to produce masks
- Learning regularize results → **great boost**



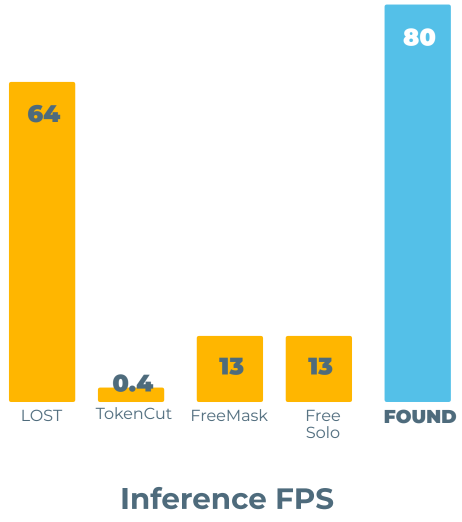
Architecture:
- **MaskFormer** [Cheng et al. CVPR'22] architecture
- Compose of an **image encoder** + a **pixel decoder** + a **transformer decoder**
- transformer decoder outputs **per-mask embeddings**

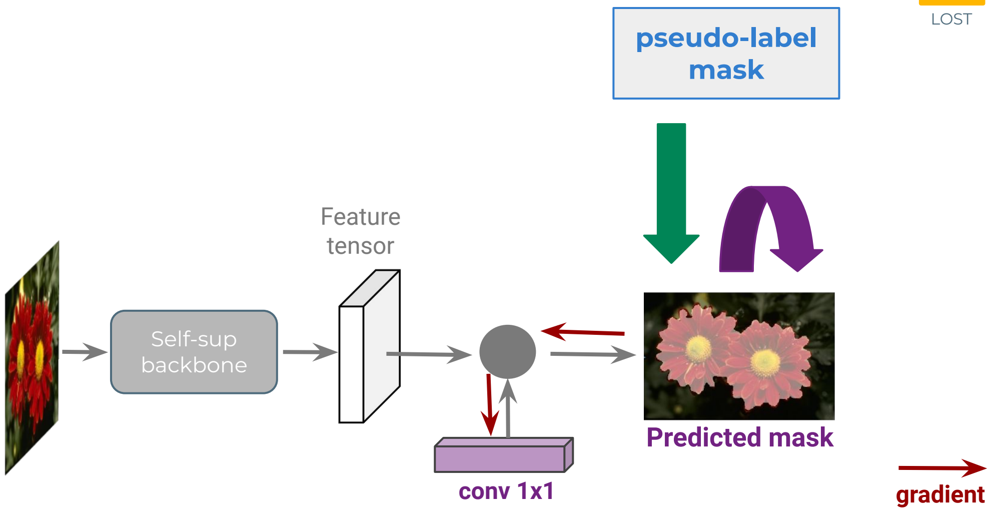
After training: + 14/16 IoU points

FOUND [Siméoni CVPR'23]

Concept

- Train a **single conv 1x1** layer with pseudo-labels
- Quick **2h training** on a single GPU with no annotation
- **Inference at 80 FPS** 🚀 on a V100





MOVE [Bielski et al. NeurIPS'22]

Related work

- **ReDO** [Chen et al. NeurIPS'19]

Idea: possible to change textures/colors of objects without changing the overall distribution of the dataset.
GAN-based method.

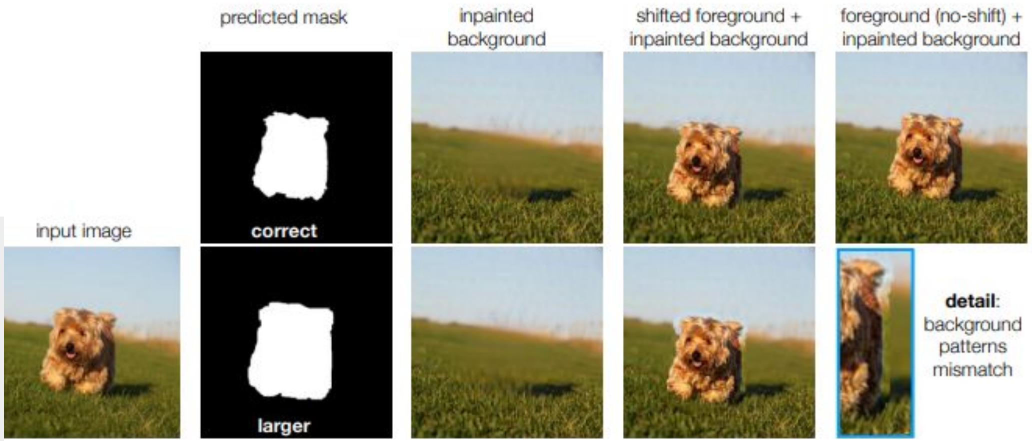
Assumptions

With a good mask:

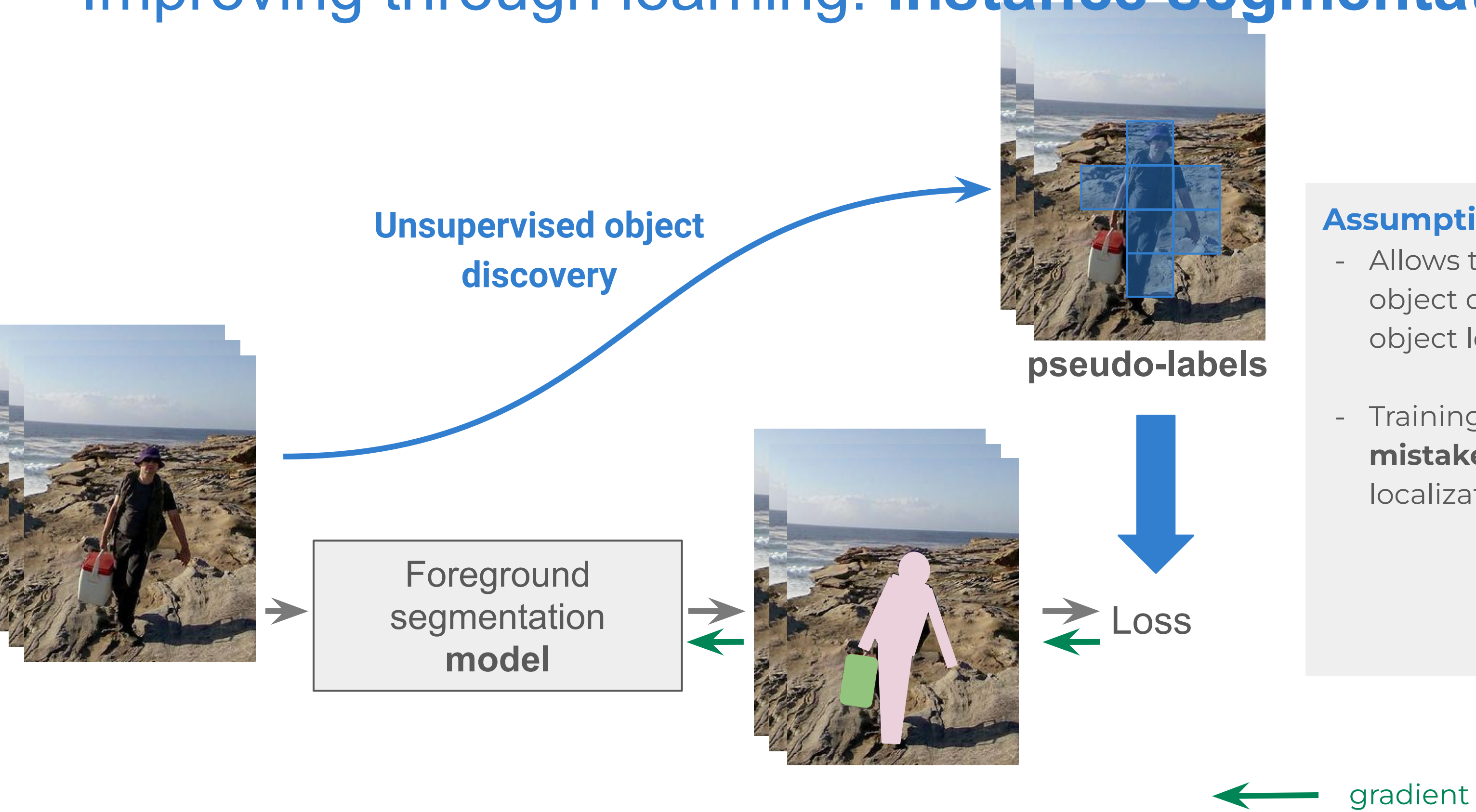
- can **remove** the object & **inpaint** the background
- can shift the extracted foreground object and **paste** it on top of the inpainted background
- if mask is not accurate → **see duplication artifacts**

Concept

- Train a **segmenter** model = **ViT+CNN head** to generate object masks
- Train a **discriminator** to predict if real or fake image



Improving through learning: **instance segmentation**



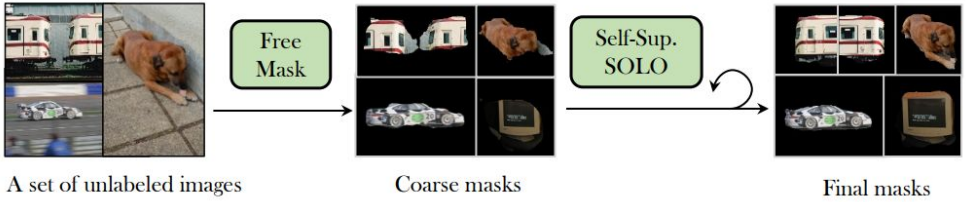
Assumptions

- Allows to go **from single** object discovery **to multiple** object localization
- Training helps to **smooth out mistakes** from initial localization

FreeSolo [Xinlong Wang et al. CVPR'22]

Concept
- Train an instance segmentation **SOLO** [Xinlong Wang et al. TPAMI'21] model

Tricks
- Use a **weakly-supervised loss** with boxes instead of masks with a loss for **min/max** of boxes and **avg** of boxes
- **Pairwise** affinity loss because **close pixels** are likely to be in the **same class**



CutLer [X. Wang et al. CVPR'23]

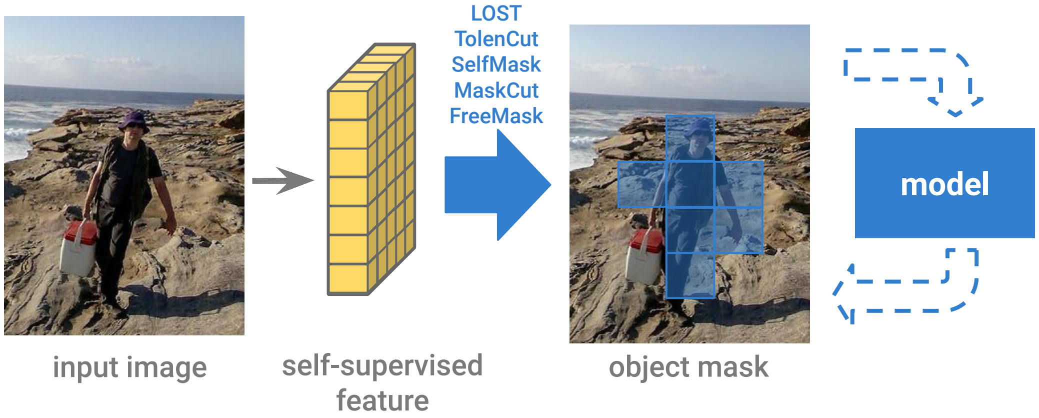
Concept

- Train instance segmentation models **Mask-RCNN** [He et al. ICCV'17] & **Cascade Mask-RCNN** [Cai et al. TPAMI'21]

Tricks

- **Drop the loss** for each predicted region that are matching any pseudo-masks
- **Copy/paste** augmentation
- Do **several rounds** of training **repetition** → **increase the number** of predicted instance

Take-away



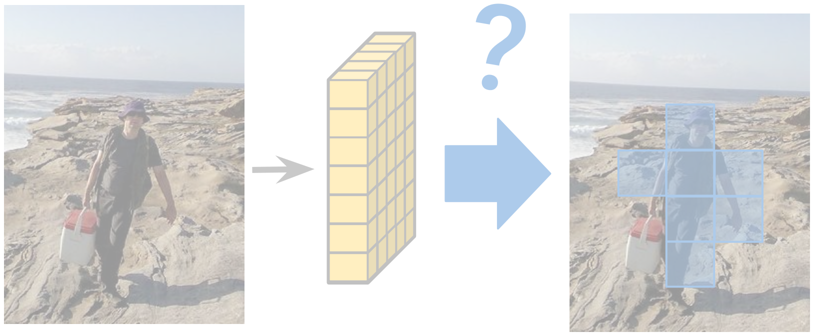
Conclusions

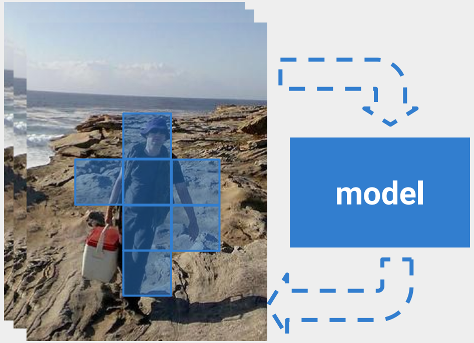
- **Training boosts** performances and regularize initial masks mistakes
- From single to **multi-object** localization

Remaining issues

- How to further improve results ?
- **Limited by the abilities** of the self-supervised features
- What about **classes**?

Presentation outline





Unsupervised object discovery
Generation of initial masks

Improving through learning

Class-aware ?

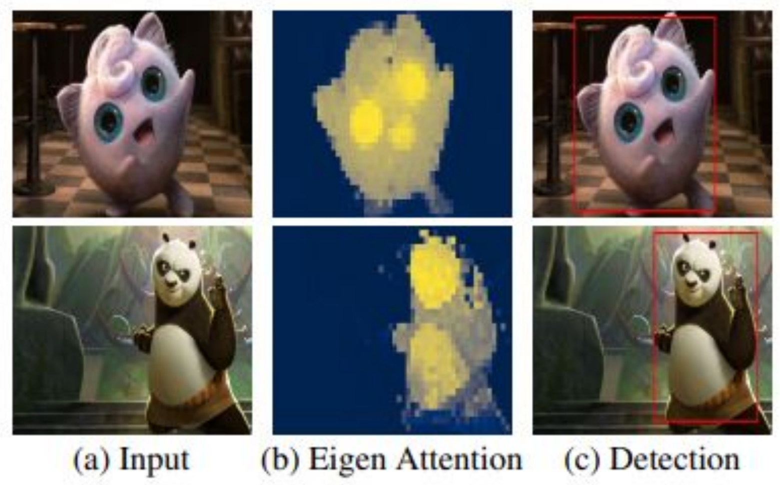
Presentation outline

Unsupervised object discovery
Generation of initial masks

Improving through learning

Class-aware ?

Out-of-domain localization



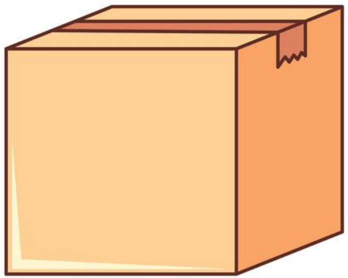
TokenCut [Wang et al. CVPR'22]

FOUND [Siméoni et al. CVPR'23]

From class-agnostic to class-aware?

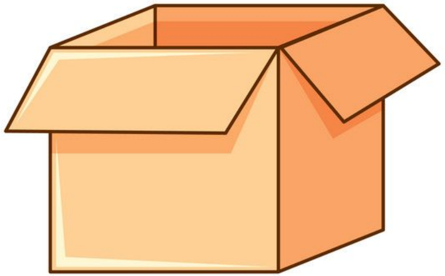
Closed-vocabulary

- Can build a **descriptor per mask** and compute **k-means clustering** at the level of the dataset [LOST, BMVC'21]
- But, **k is a hyper-parameter**

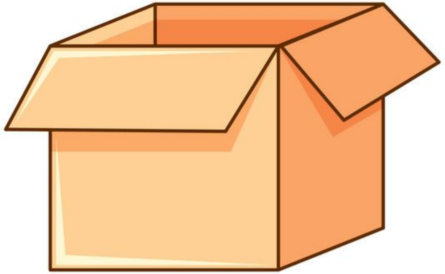


Open-vocabulary

- SSL features have at the moment **better dense-discrimativeness** than CLIP-like models
- But given mask computed using SSL features one can compute descriptor in an open-voc representation

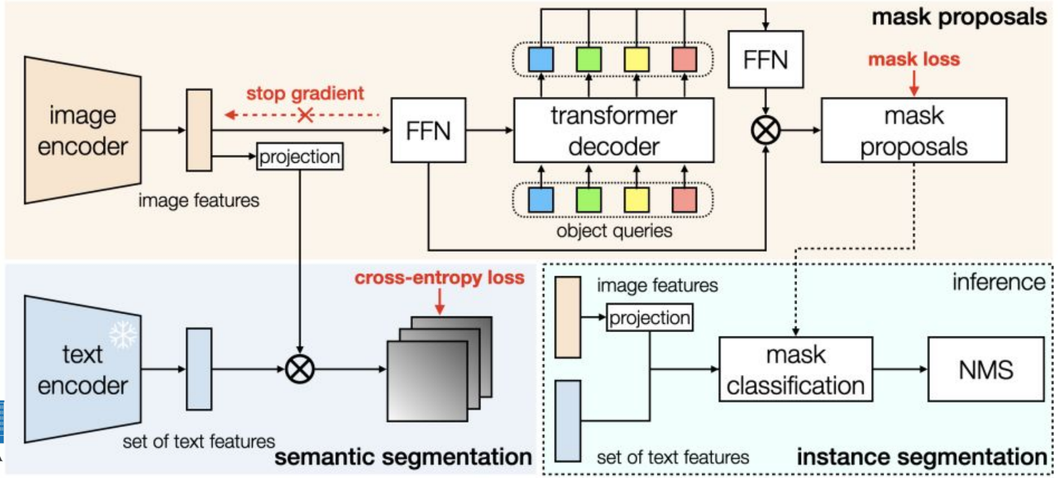


Open-vocabulary: **Zero-shot Unsupervised Transfer Instance Segmentation** [Shin et al., CVPRW'23]

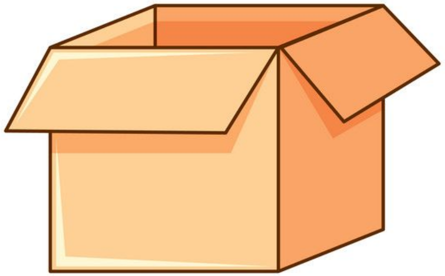


Concept

- A unified framework for **semantic** and **instance segmentation**
- Propose to match images to **set of text features**

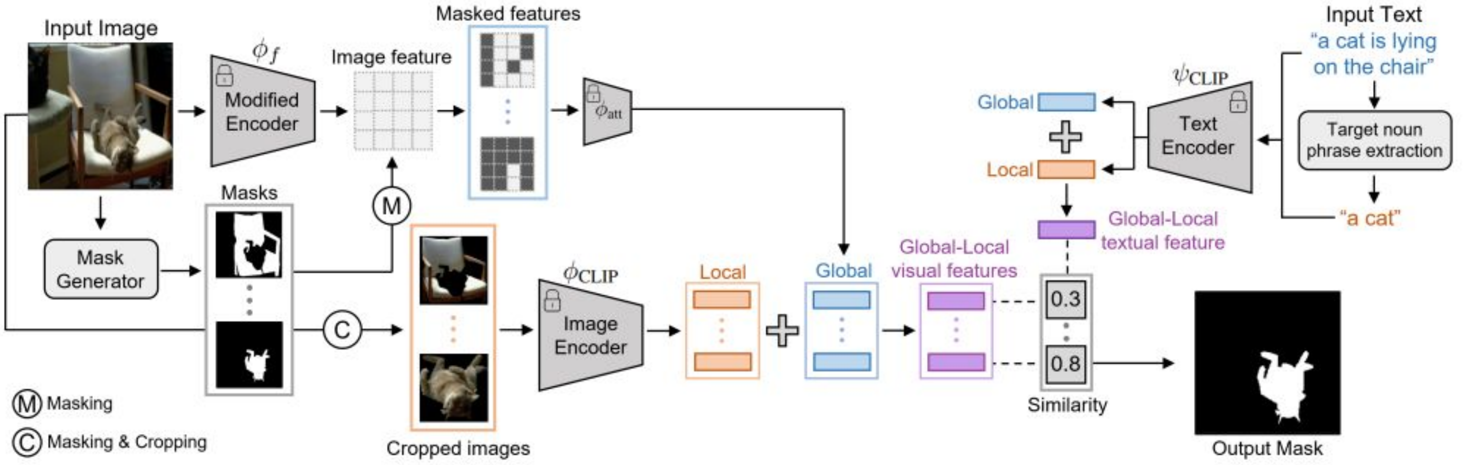


Open-vocabulary: **Zero-shot Referring Image Segmentation** [Yu et al., CVPR'23]



Concept

- **Zero-shot referring image segmentation**: find grounding region for a text input
- Use **FreeSolo** to generate masks
- Propose a **local/global** similarity



Take-away

Conclusions

- Possible to discover **objects** with **no annotation**
- Easy extraction method for *single object* localization
- Training allows to **boosts** performances and increase # localized objects
- Possible to assign **closed/open** classes to masks/boxes

Remaining issues

- How to further improve results ?
- **Limited by the abilities** of the self-supervised features
- Could we **learn image representation** specifically **designed** for the needs of **object localization** ?

Questions ?