



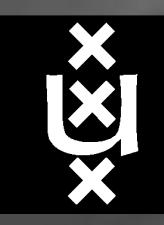
UNIVERSITY OF AMSTERDAM

# Vision Foundation Models (with academic compute)

YUKI M ASANO

ECCV 2024

SSLWIN



UNIVERSITY OF AMSTERDAM

# Vision Foundation Models (with academic compute)

YUKI M ASANO  
ECCV 2024

SSLWIN

# Instruction tuning: the difference between GPT-3 and

giant training corpora



(raw) LLMs

(e.g. GPT-3)

- ✗ requires industrial compute
- ✗ model is relatively useless

small IT datasets



Instruction tuning



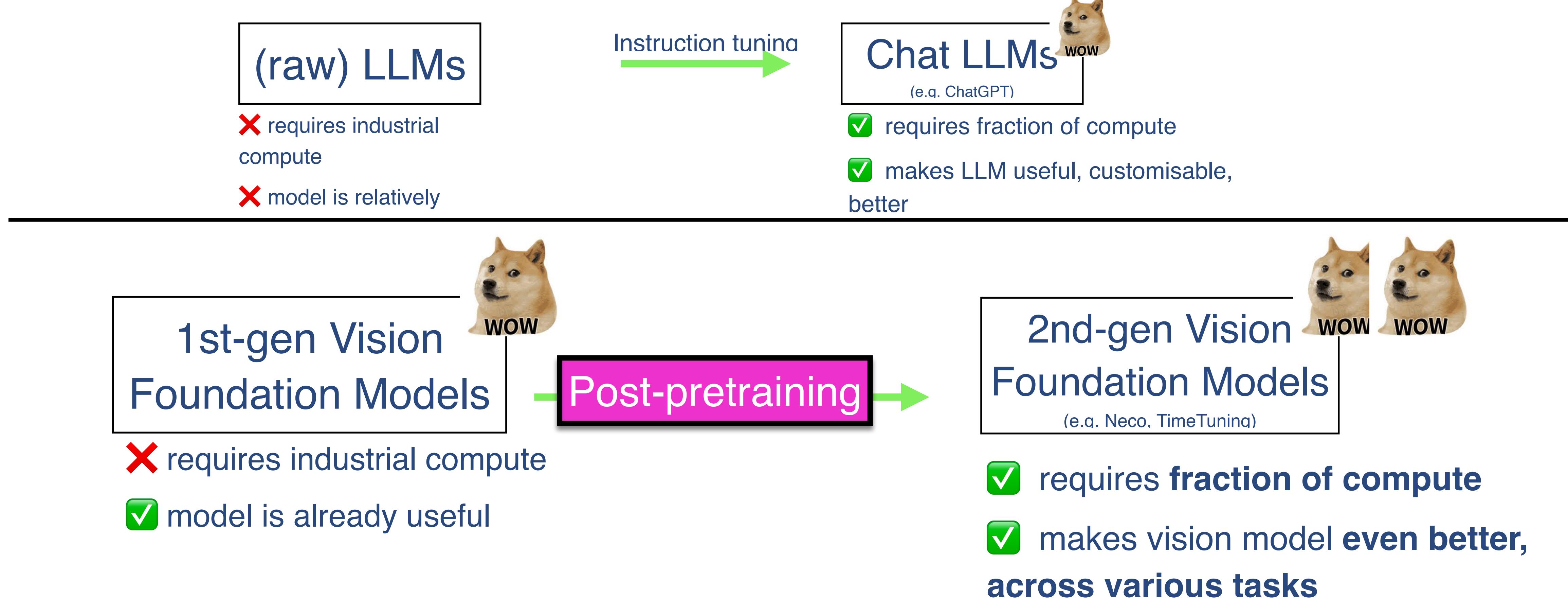
Chat LLMs

(e.g. ChatGPT)



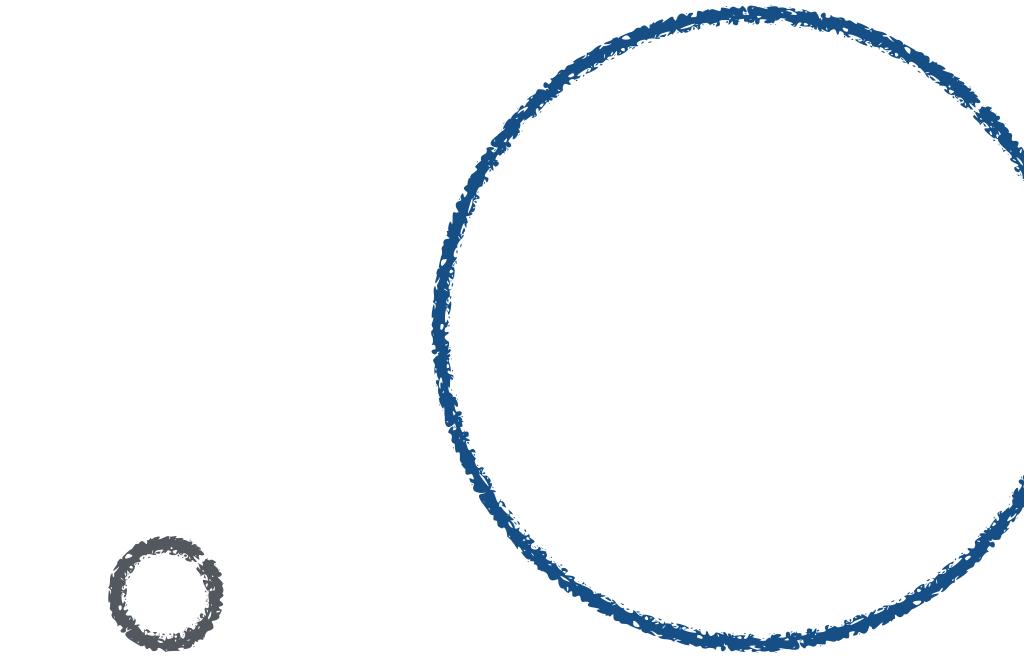
- ✓ requires fraction of compute
- ✓ makes LLM useful, customisable, better

# Computer vision needs more *post-pretraining*



# Why Self-supervised Learning still matters, despite CLIP and

Massive scale



sup. << weak sup. << raw

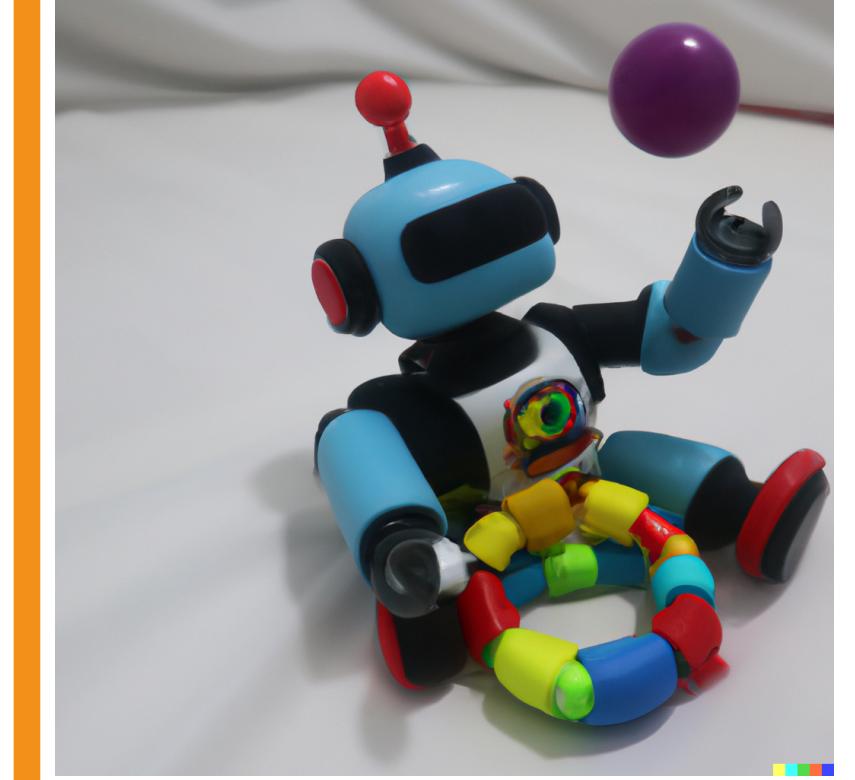
Cost of (re)labelling



Problems of labels



Fundamentals





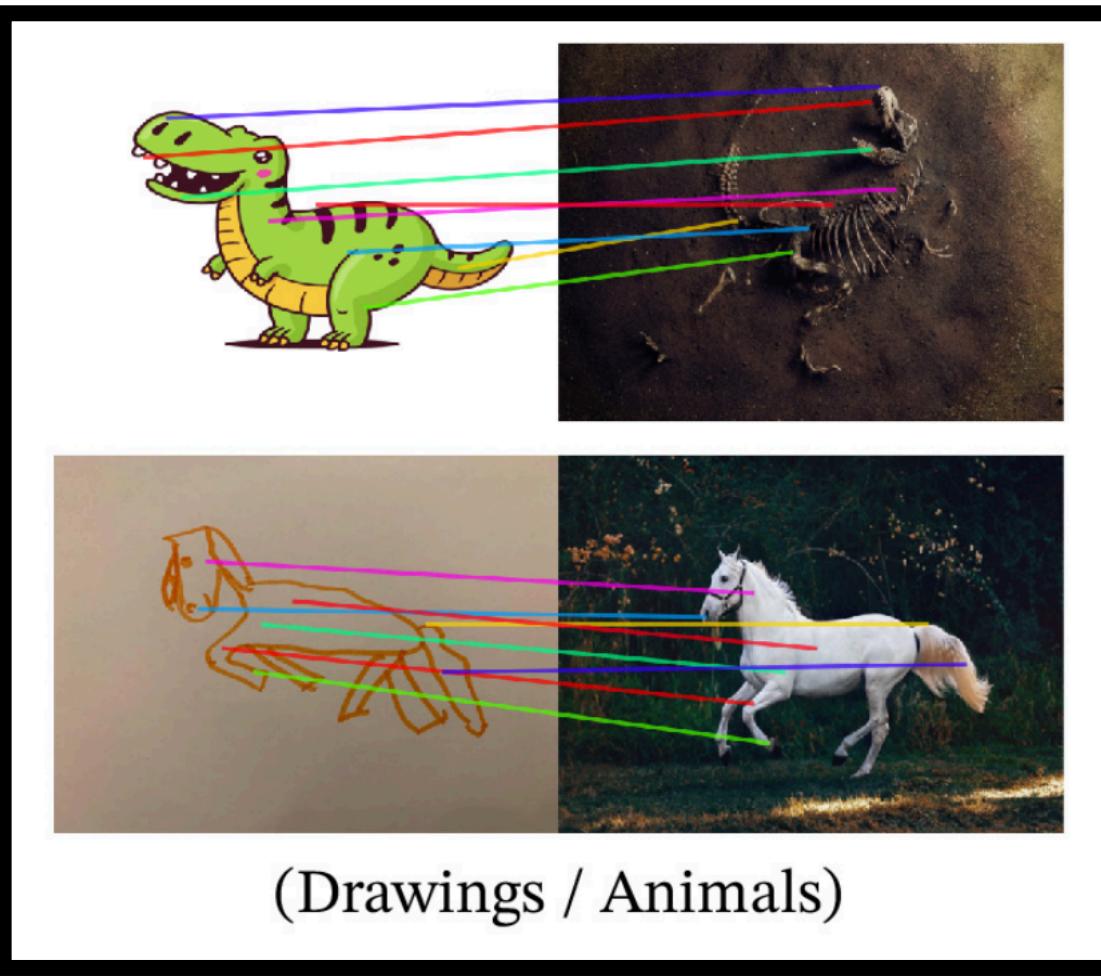
*NeCo: Improving DINov2's spatial representations in 19 GPU hours with Patch Neighbor Consistency.*

Valentinos Pariza, Mohammadreza Salehi, Gertjan Burghouts, Francesco Locatello, Yuki M. Asano.  
<https://www.taste.com.au/recipes/creamy-bacon-carbonara/12c27b1e-5fb6-48c9-ac42-82a37cc8ab30?nk=3b0399578618abc93c6fc9abab4a14a7-1727548620>

# How semantic are patch representations?

Which patch from the whole dataset is the closest?

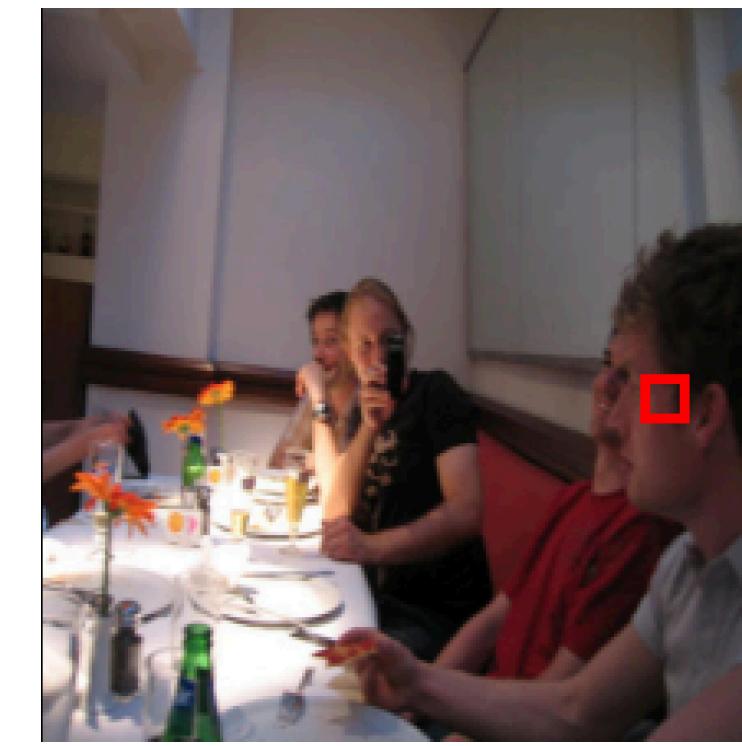
Qualitative results in DINOv2



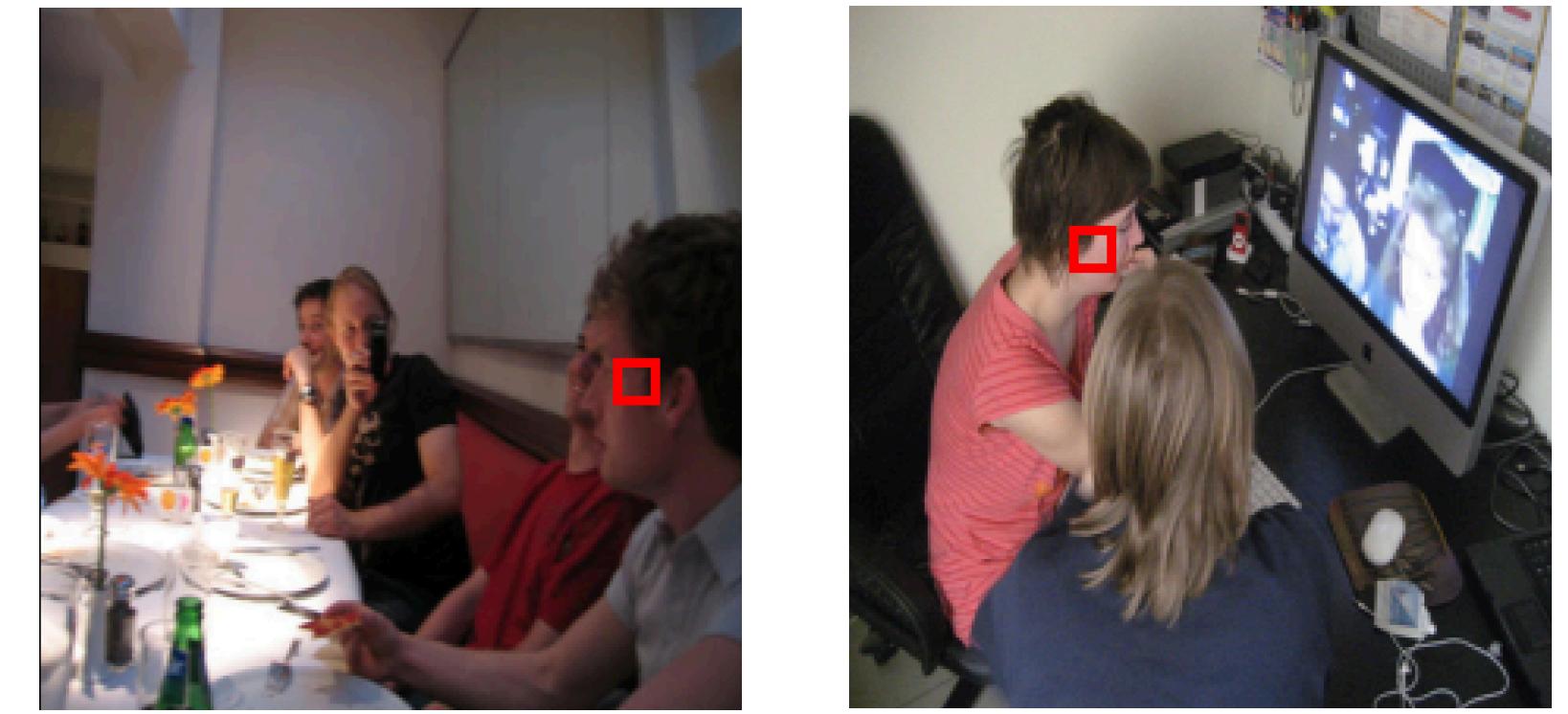
But often...



How it should be



How it is



with SoTA DINOv2-R model

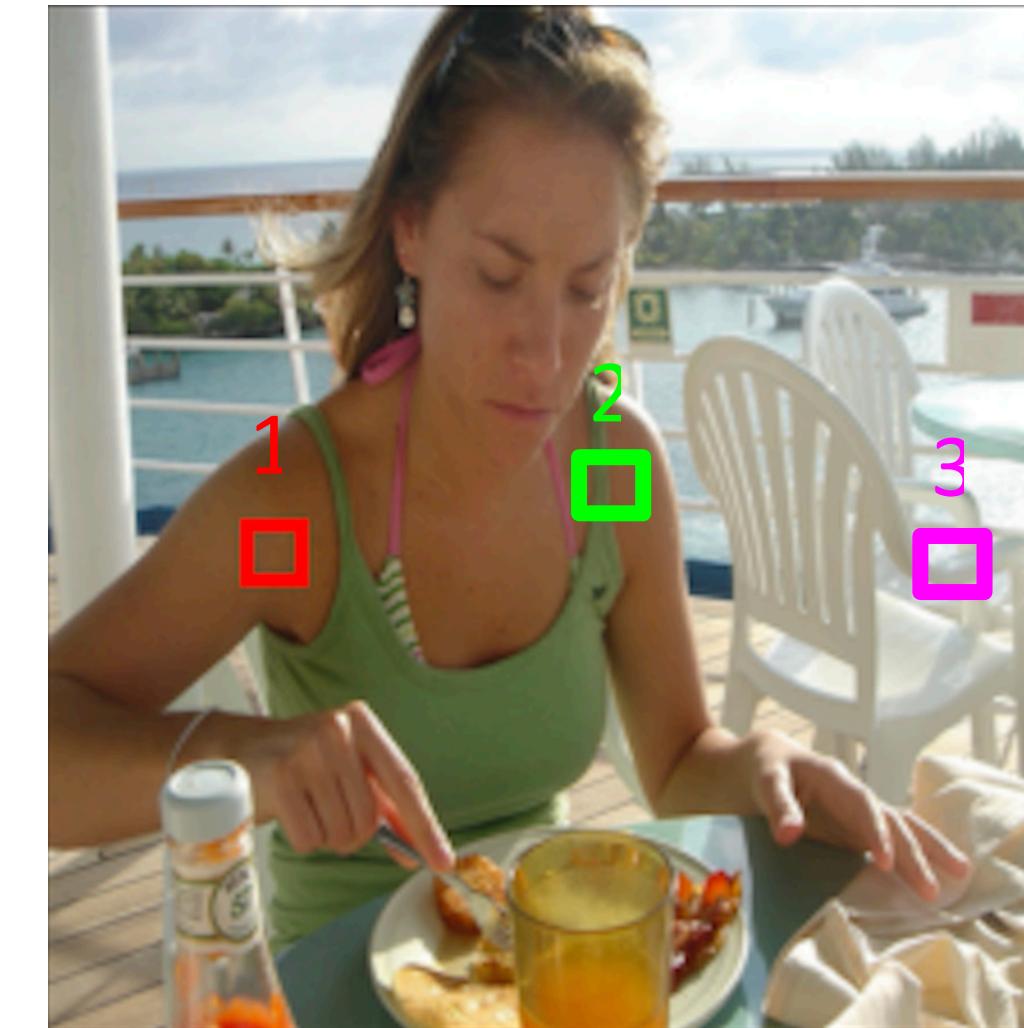
# Idea of Patch Nearest Neighbor Consistency: intuitive

Given a **query patch** of a right shoulder, top neighbors should be in the following order:

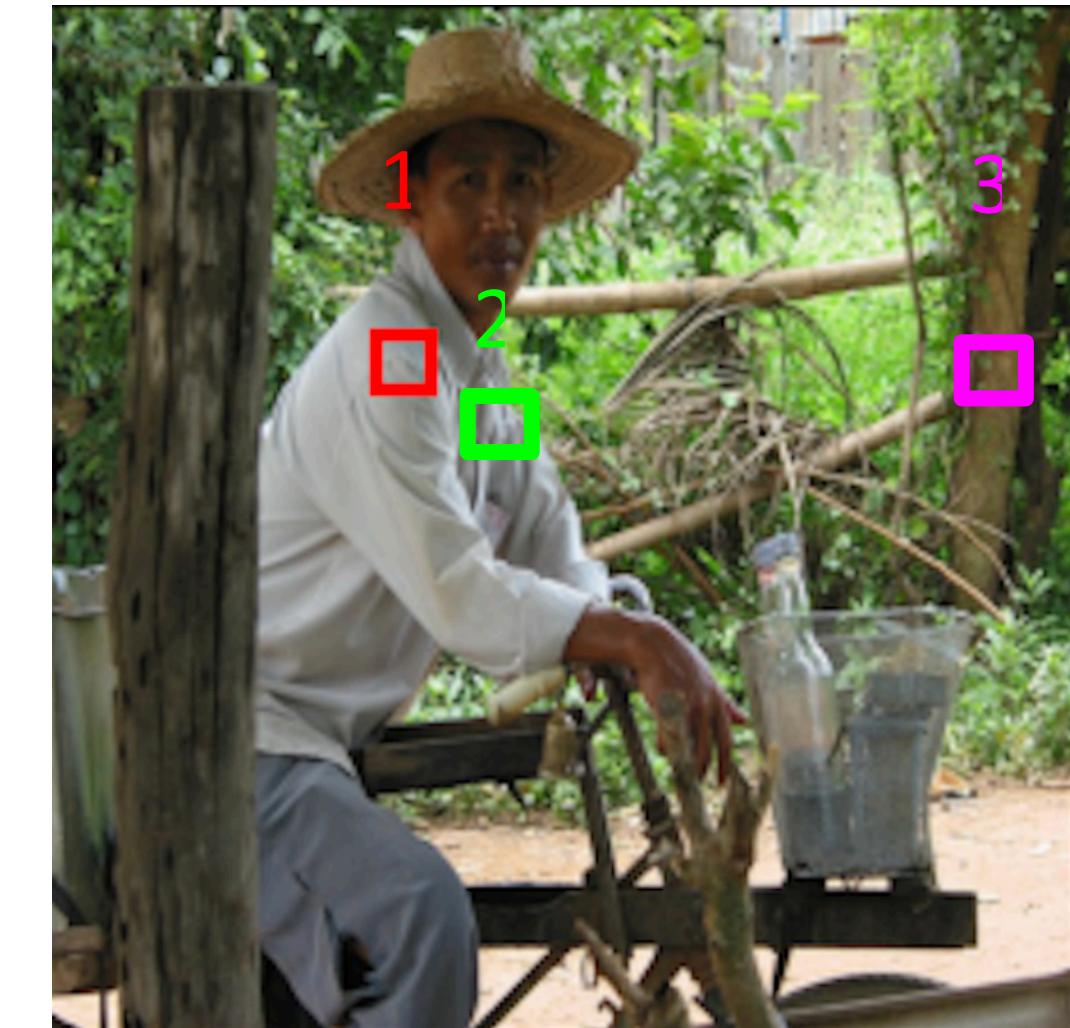
- (1) All Right Shoulder Patches, (2) All Left Shoulder Patches, (...) (3) Everything Else**



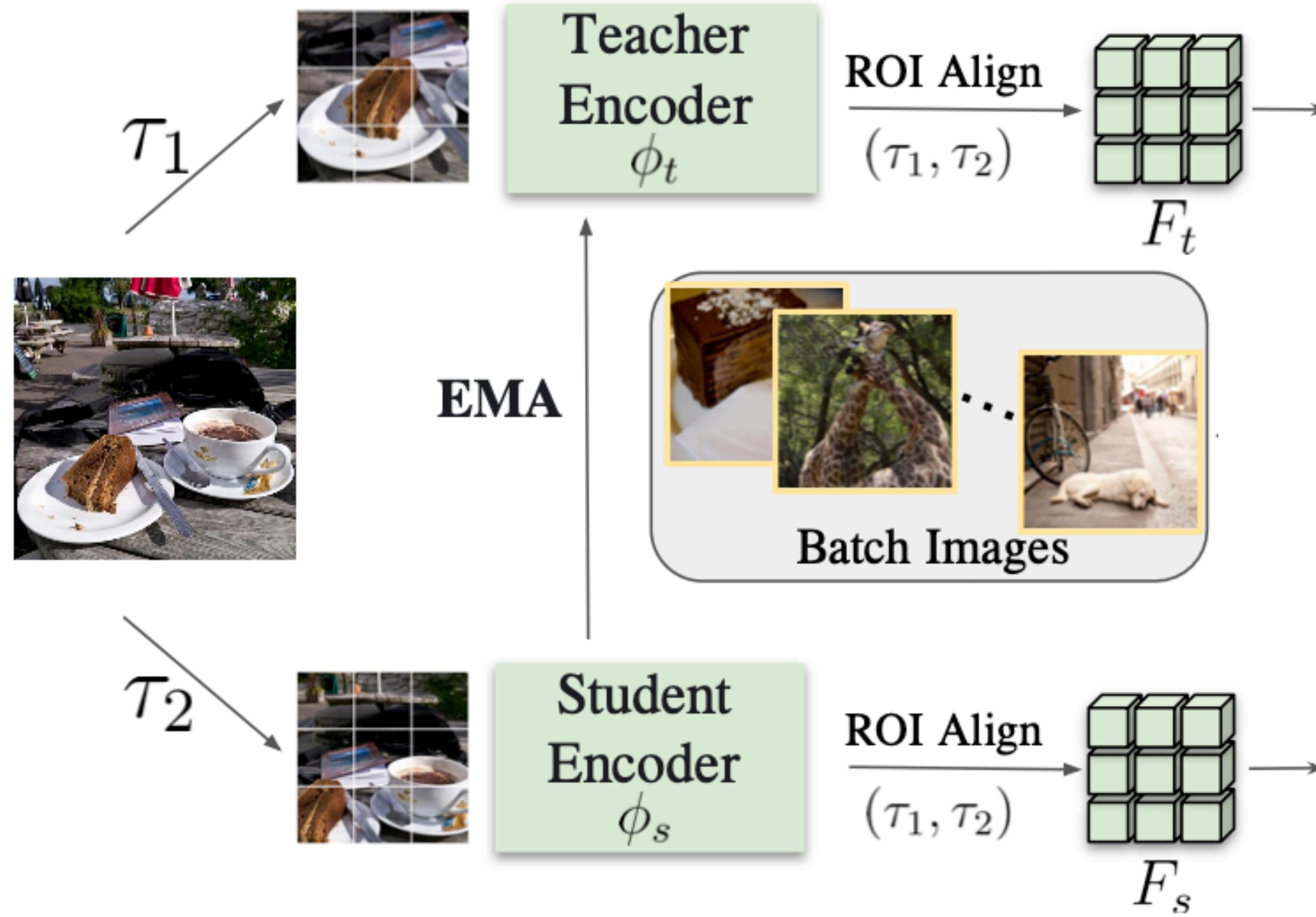
Query Patch



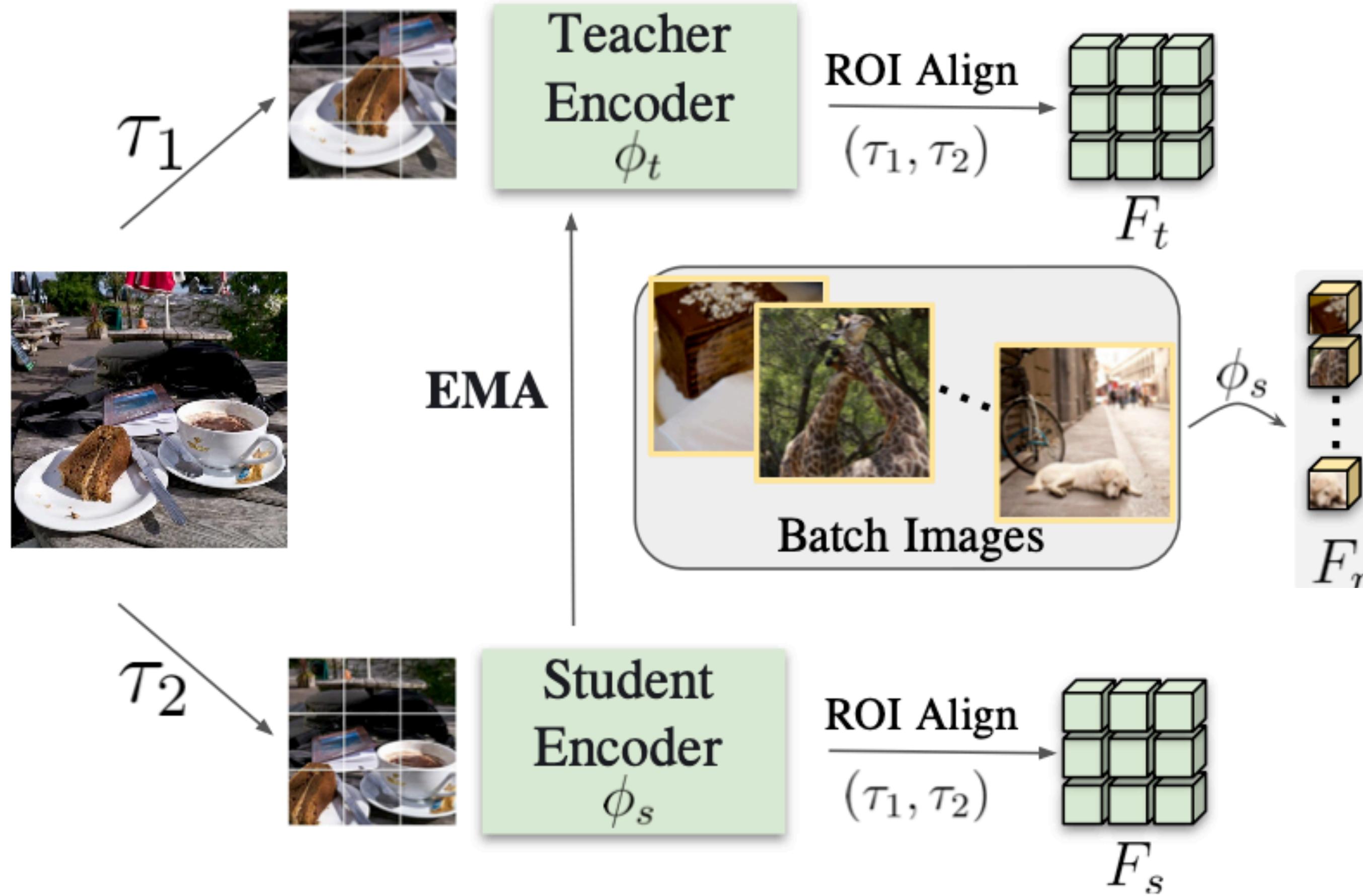
Example Patches



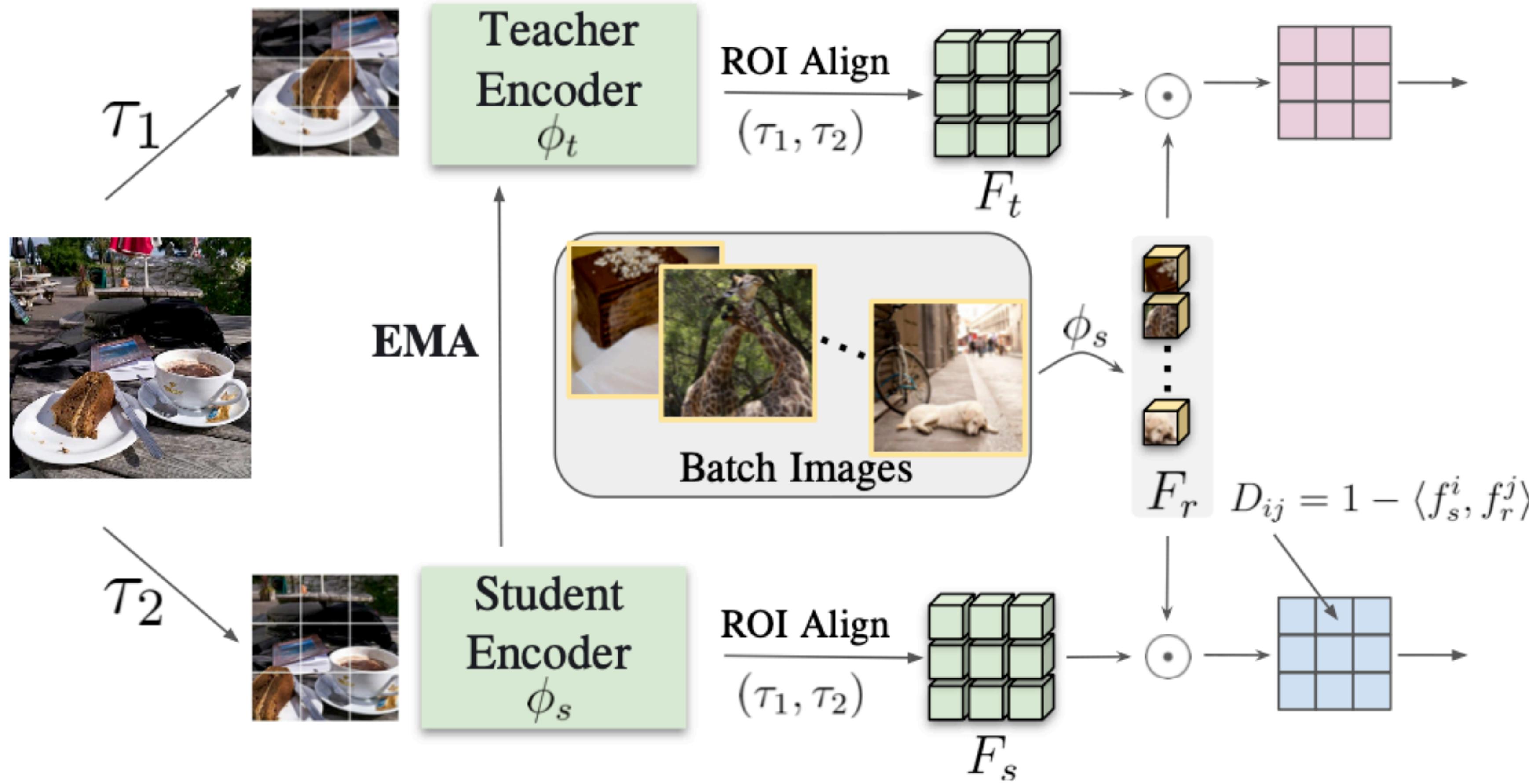
# PaNeCo: Patch Nearest neighbor Consistency



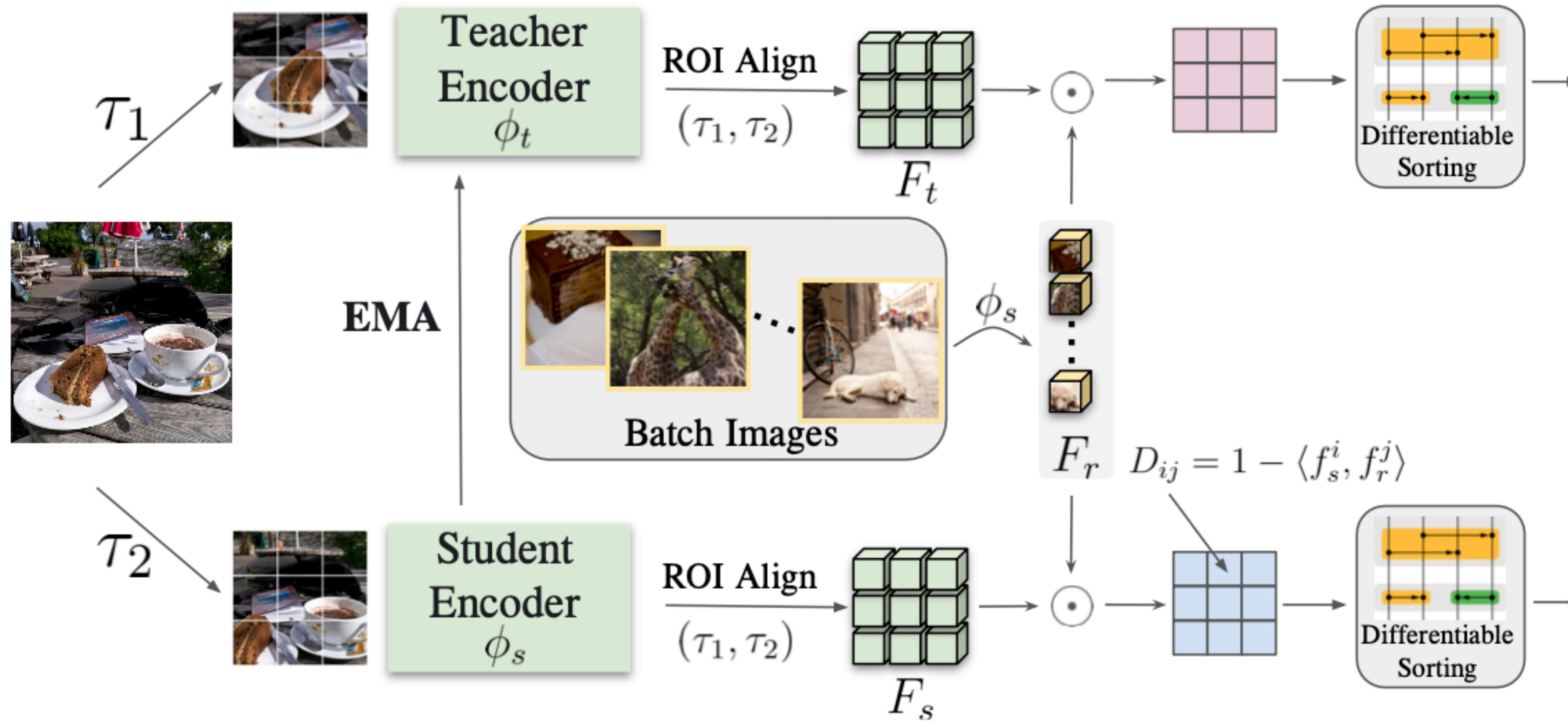
# PaNeCo: Patch Nearest neighbor Consistency



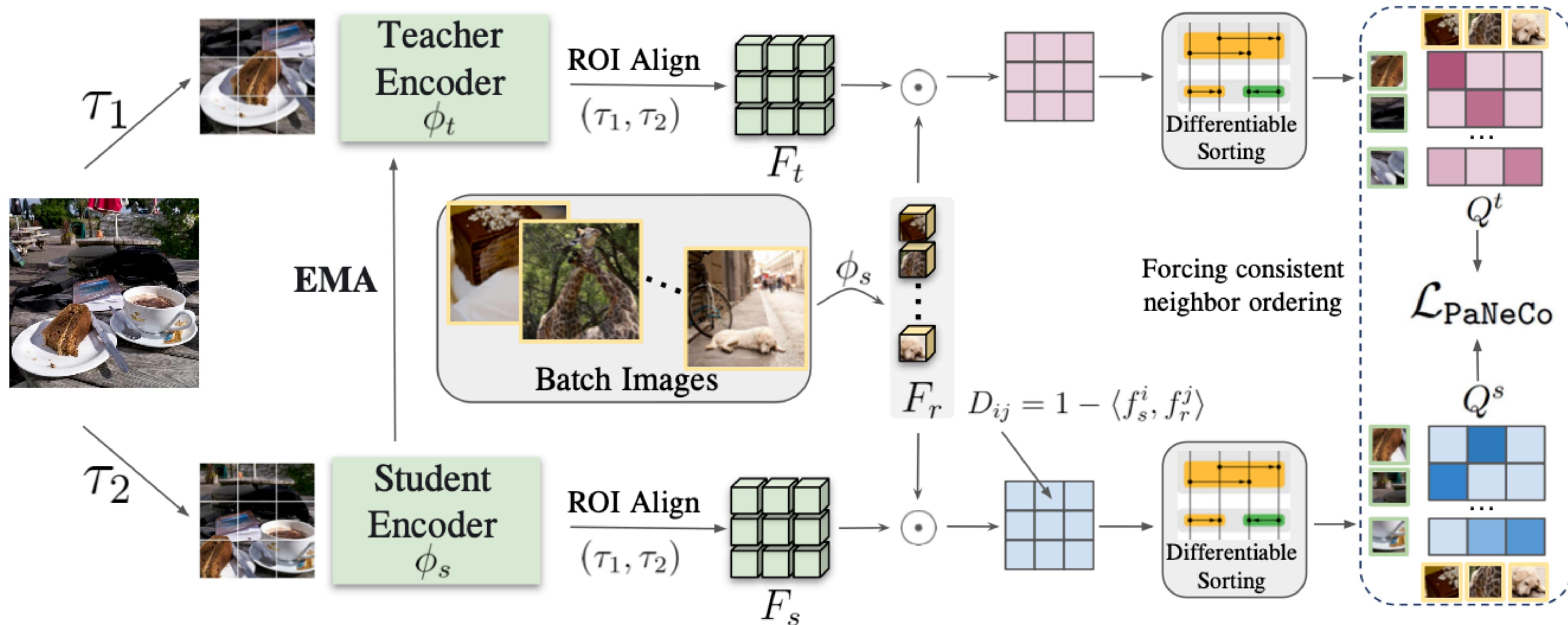
# PaNeCo: Patch Nearest neighbor Consistency



# PaNeCo: Patch Nearest neighbor Consistency

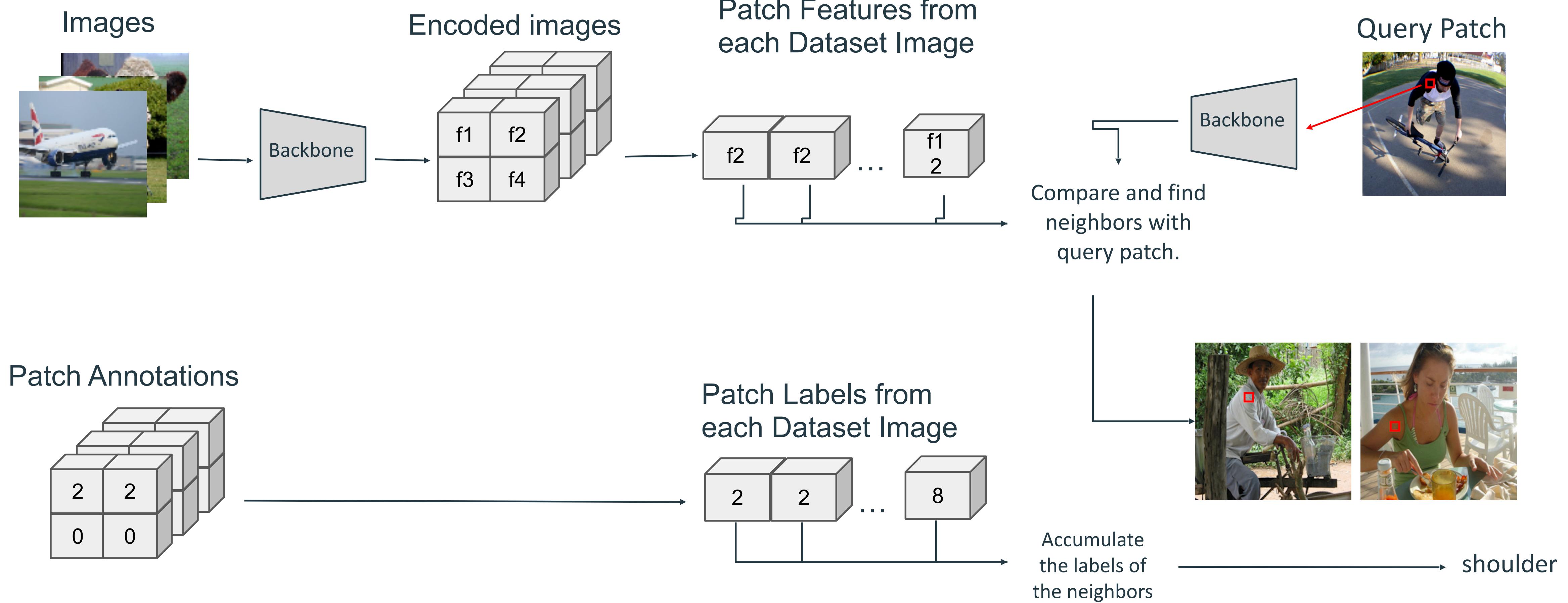


# PaNeCo: Patch Nearest neighbor Consistency

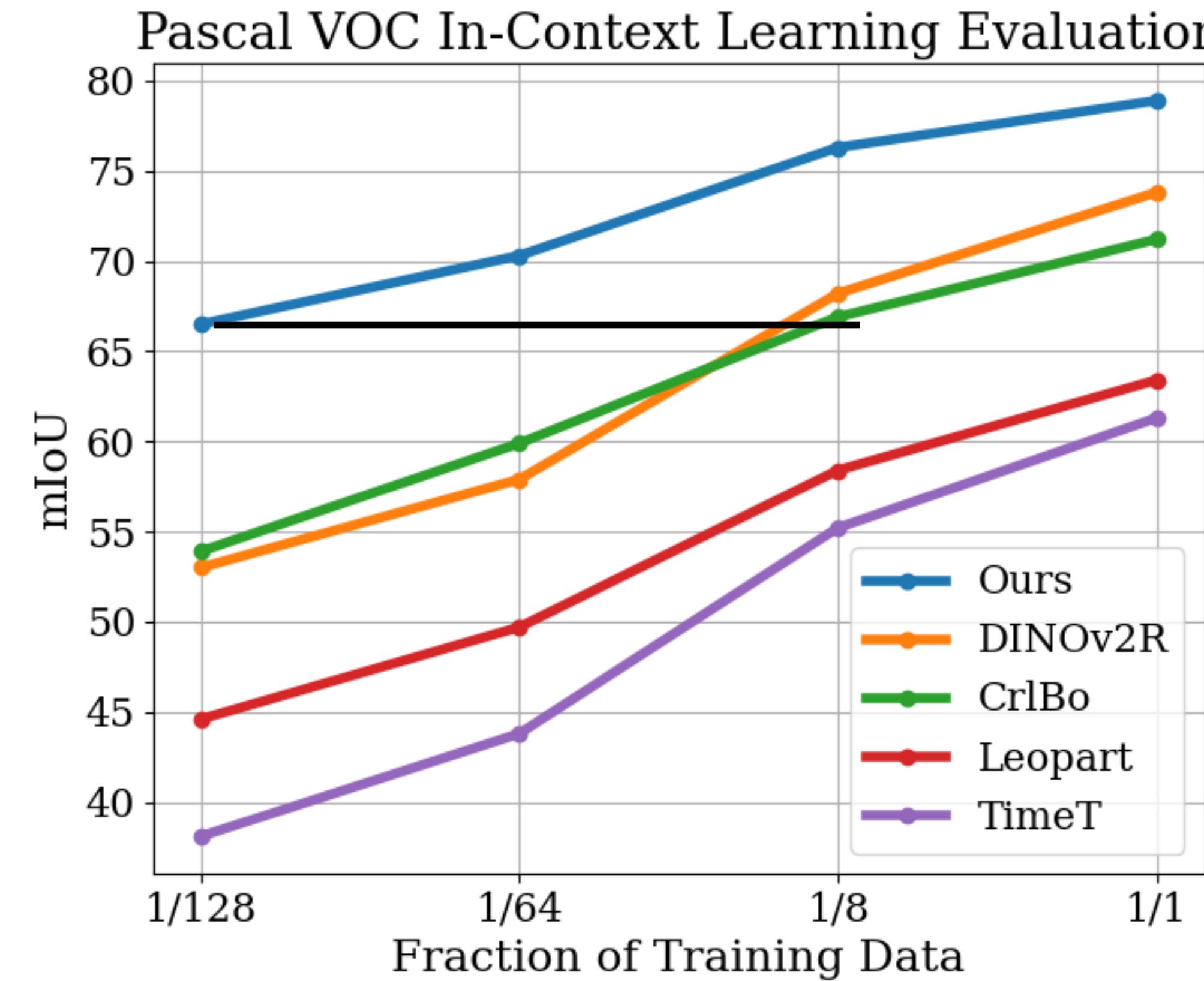


# Results

# Evaluation 1: Visual in-context segmentation via dense NN retrieval

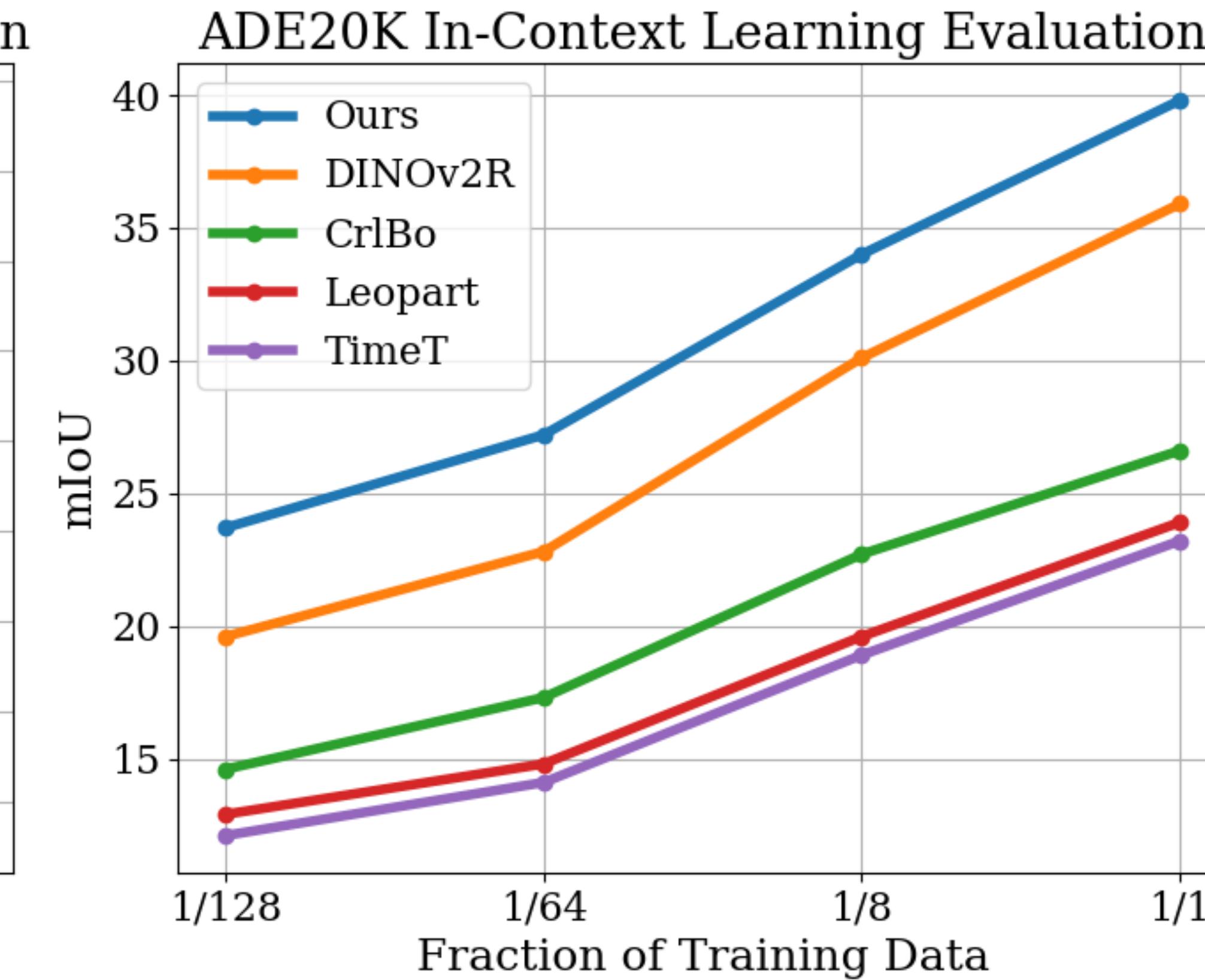
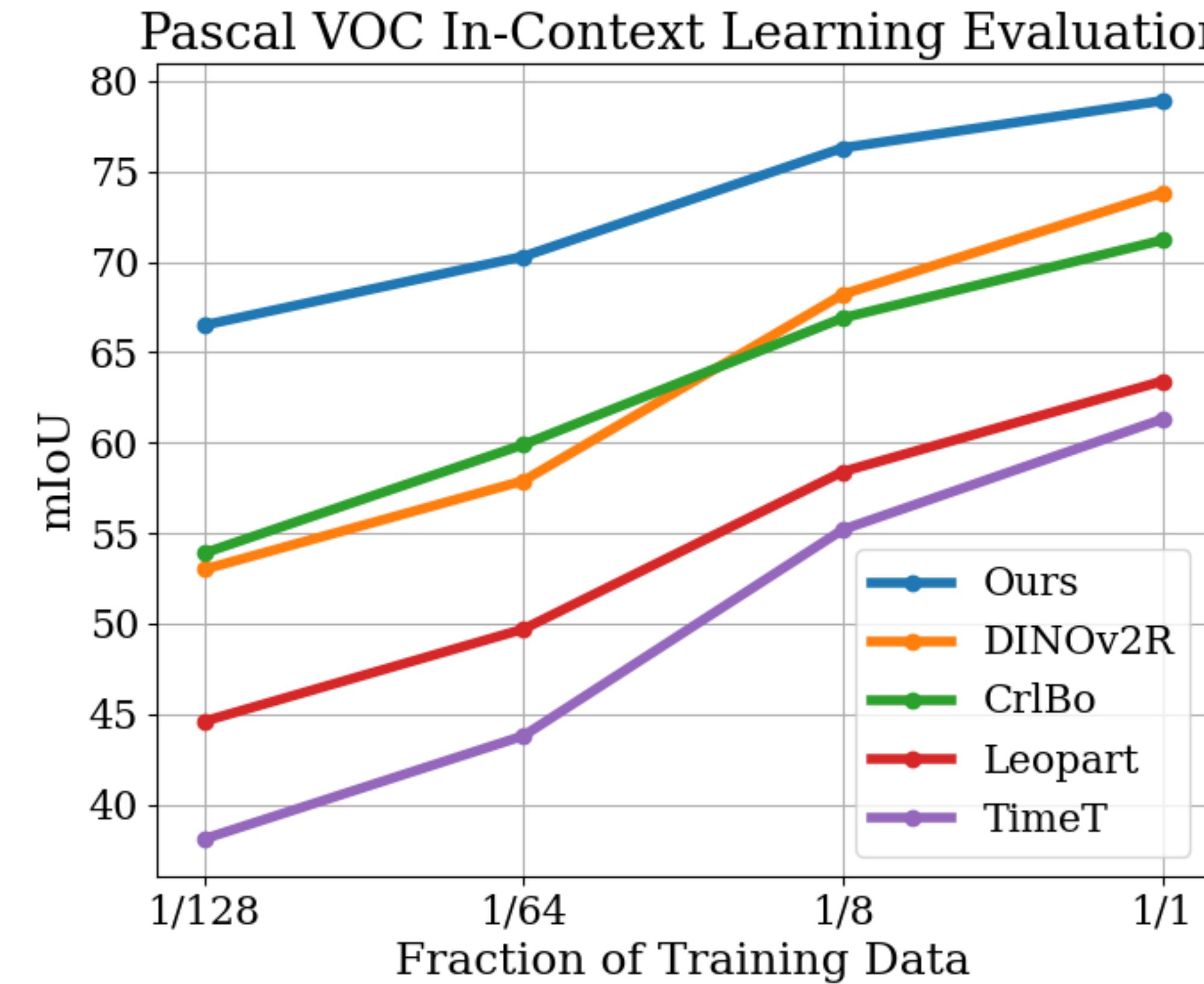


# In-context scene understanding benchmark

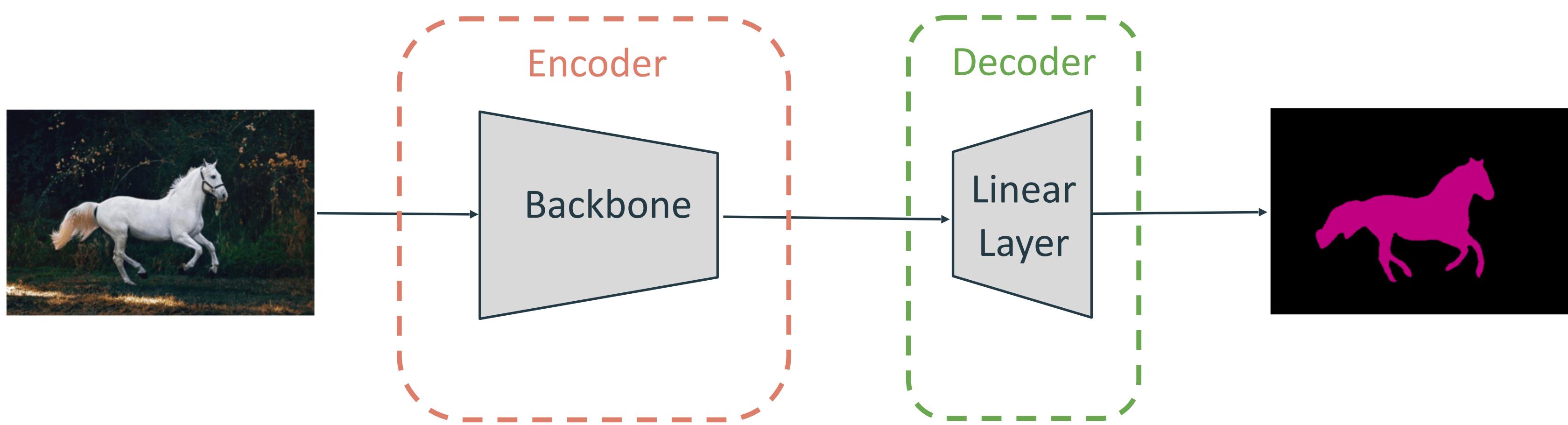


matches performances of  
DINOv2-R with ~15x less  
data

# In-context scene understanding benchmark



# Linear Segmentation Evaluation



- Encode Image to patch-level features,
- Decode with a linear layer the per pixel semantic labels of the image,
- Supervised training of the linear layer of the decoder for this task.

# Linear segmentation performance

Method	Backbone	Params	COCO-Things	COCO-Stuff	Pascal VOC	ADE20K
DINO	ViT-S/16	21M	43.9	45.9	50.2	17.5
TimeT	ViT-S/16	21M	58.2	48.7	66.3	20.7
iBOT	ViT-S/16	21M	58.9	51.5	66.1	21.8
CrOC	ViT-S/16	21M	64.3	51.2	67.4	23.1
CrlBo	ViT-S/16	21M	64.3	49.1	71.6	22.7
DINOv2R	ViT-S/14	21M	75.3	56.0	74.2	35.0
PaNeCo	ViT-S/14	21M	<b>82.3</b>	<b>62.0</b>	<b>81.3</b>	<b>40.1</b>
DINO	ViT-B/16	85M	55.8	51.2	62.7	23.6
MAE	ViT-B/16	85M	38.0	38.6	32.9	5.8
iBOT	ViT-B/16	85M	69.4	55.9	73.1	30.1
CrIBo	ViT-B/16	85M	69.6	53.0	73.9	25.7
DINOv2R	ViT-B/14	85M	78.3	57.6	79.8	40.3
PaNeCo	ViT-B/14	85M	<b>85.5</b>	<b>63.3</b>	<b>83.3</b>	<b>44.9</b>

A linear segmentation head is trained on top of the frozen spatial features obtained from different feature extractors. We report the mIoU scores achieved on the validation sets of 4 different datasets.

# PaNeCo starting with different pretrained weights.

Pretrain	Pascal VOC						COCO-Things					
	At Init			+PANeCo			At Init			+PANeCo		
	K=GT	K=500	Lin.	K=GT	K=500	Lin.	K=21	K=500	Lin.	K=21	K=500	Lin.
iBOT [92]	4.4	31.1	66.1	15.4 <sup>↑11.0</sup>	51.2 <sup>↑20.1</sup>	68.6 <sup>↑2.5</sup>	7.6	28.0	58.9	20.4 <sup>↑12.8</sup>	52.8 <sup>↑24.8</sup>	67.7 <sup>↑8.8</sup>

# PaNeCo starting with different pretrained weights.

Pretrain	Pascal VOC						COCO-Things					
	At Init			+PANeCo			At Init			+PANeCo		
	K=GT	K=500	Lin.	K=GT	K=500	Lin.	K=21	K=500	Lin.	K=21	K=500	Lin.
iBOT [92]	4.4	31.1	66.1	15.4 <sup>↑11.0</sup>	51.2 <sup>↑20.1</sup>	68.6 <sup>↑2.5</sup>	7.6	28.0	58.9	20.4 <sup>↑12.8</sup>	52.8 <sup>↑24.8</sup>	67.7 <sup>↑8.8</sup>
DINO [15]	4.3	17.3	50.2	14.5 <sup>↑10.2</sup>	47.9 <sup>↑30.6</sup>	61.3 <sup>↑11.1</sup>	5.4	19.2	43.9	16.9 <sup>↑11.5</sup>	50.0 <sup>↑30.8</sup>	62.4 <sup>↑18.5</sup>

# PaNeCo starting with different pretrained weights.

Pretrain	Pascal VOC						COCO-Things					
	At Init			+PANeCo			At Init			+PANeCo		
	K=GT	K=500	Lin.	K=GT	K=500	Lin.	K=21	K=500	Lin.	K=21	K=500	Lin.
iBOT [92]	4.4	31.1	66.1	15.4 <sup>↑11.0</sup>	51.2 <sup>↑20.1</sup>	68.6 <sup>↑2.5</sup>	7.6	28.0	58.9	20.4 <sup>↑12.8</sup>	52.8 <sup>↑24.8</sup>	67.7 <sup>↑8.8</sup>
DINO [15]	4.3	17.3	50.2	14.5 <sup>↑10.2</sup>	47.9 <sup>↑30.6</sup>	61.3 <sup>↑11.1</sup>	5.4	19.2	43.9	16.9 <sup>↑11.5</sup>	50.0 <sup>↑30.8</sup>	62.4 <sup>↑18.5</sup>
TimeT [66]	12.2	46.2	66.3	17.9 <sup>↑5.7</sup>	52.1 <sup>↑5.9</sup>	68.5 <sup>↑2.2</sup>	18.4	44.6	58.2	20.6 <sup>↑2.2</sup>	54.3 <sup>↑9.7</sup>	64.8 <sup>↑6.6</sup>

# PaNeCo starting with different pretrained weights.

Pretrain	Pascal VOC						COCO-Things					
	At Init			+PANECo			At Init			+PANECo		
	K=GT	K=500	Lin.	K=GT	K=500	Lin.	K=21	K=500	Lin.	K=21	K=500	Lin.
iBOT [92]	4.4	31.1	66.1	15.4 <sup>↑11.0</sup>	51.2 <sup>↑20.1</sup>	68.6 <sup>↑2.5</sup>	7.6	28.0	58.9	20.4 <sup>↑12.8</sup>	52.8 <sup>↑24.8</sup>	67.7 <sup>↑8.8</sup>
DINO [15]	4.3	17.3	50.2	14.5 <sup>↑10.2</sup>	47.9 <sup>↑30.6</sup>	61.3 <sup>↑11.1</sup>	5.4	19.2	43.9	16.9 <sup>↑11.5</sup>	50.0 <sup>↑30.8</sup>	62.4 <sup>↑18.5</sup>
TimeT [66]	12.2	46.2	66.3	17.9 <sup>↑5.7</sup>	52.1 <sup>↑5.9</sup>	68.5 <sup>↑2.2</sup>	18.4	44.6	58.2	20.6 <sup>↑2.2</sup>	54.3 <sup>↑9.7</sup>	64.8 <sup>↑6.6</sup>
Leopard [93]	15.4	51.2	66.5	21.0 <sup>↑5.6</sup>	55.3 <sup>↑4.1</sup>	68.3 <sup>↑1.8</sup>	14.8	53.2	63.0	18.8 <sup>↑4.0</sup>	53.9 <sup>↑0.7</sup>	65.4 <sup>↑2.4</sup>

# PaNeCo starting with different pretrained weights.

Pretrain	Pascal VOC						COCO-Things					
	At Init			+PANECO			At Init			+PANECO		
	K=GT	K=500	Lin.	K=GT	K=500	Lin.	K=21	K=500	Lin.	K=21	K=500	Lin.
iBOT [92]	4.4	31.1	66.1	15.4 <sup>↑11.0</sup>	51.2 <sup>↑20.1</sup>	68.6 <sup>↑2.5</sup>	7.6	28.0	58.9	20.4 <sup>↑12.8</sup>	52.8 <sup>↑24.8</sup>	67.7 <sup>↑8.8</sup>
DINO [15]	4.3	17.3	50.2	14.5 <sup>↑10.2</sup>	47.9 <sup>↑30.6</sup>	61.3 <sup>↑11.1</sup>	5.4	19.2	43.9	16.9 <sup>↑11.5</sup>	50.0 <sup>↑30.8</sup>	62.4 <sup>↑18.5</sup>
TimeT [66]	12.2	46.2	66.3	17.9 <sup>↑5.7</sup>	52.1 <sup>↑5.9</sup>	68.5 <sup>↑2.2</sup>	18.4	44.6	58.2	20.6 <sup>↑2.2</sup>	54.3 <sup>↑9.7</sup>	64.8 <sup>↑6.6</sup>
Leopart [93]	15.4	51.2	66.5	21.0 <sup>↑5.6</sup>	55.3 <sup>↑4.1</sup>	68.3 <sup>↑1.8</sup>	14.8	53.2	63.0	18.8 <sup>↑4.0</sup>	53.9 <sup>↑0.7</sup>	65.4 <sup>↑2.4</sup>
CrIBo [49]	18.3	54.5	71.6	21.7 <sup>↑3.4</sup>	59.6 <sup>↑5.1</sup>	72.1 <sup>↑0.5</sup>	14.5	48.3	64.3	21.1 <sup>↑6.6</sup>	54.0 <sup>↑5.7</sup>	68.0 <sup>↑3.7</sup>

frozen clustering and linear segmentation results on Pascal VOC and COCO-Things.

→ PaNeCo considerably boosts ( $\uparrow$ ) the performance of **different backbones**

# Qualitative Results

# Nearest Neighbors of Patches from representations

Query



Retrieved Nearest Neighbors

DINOv2R



PaNeCo



DINOv2R



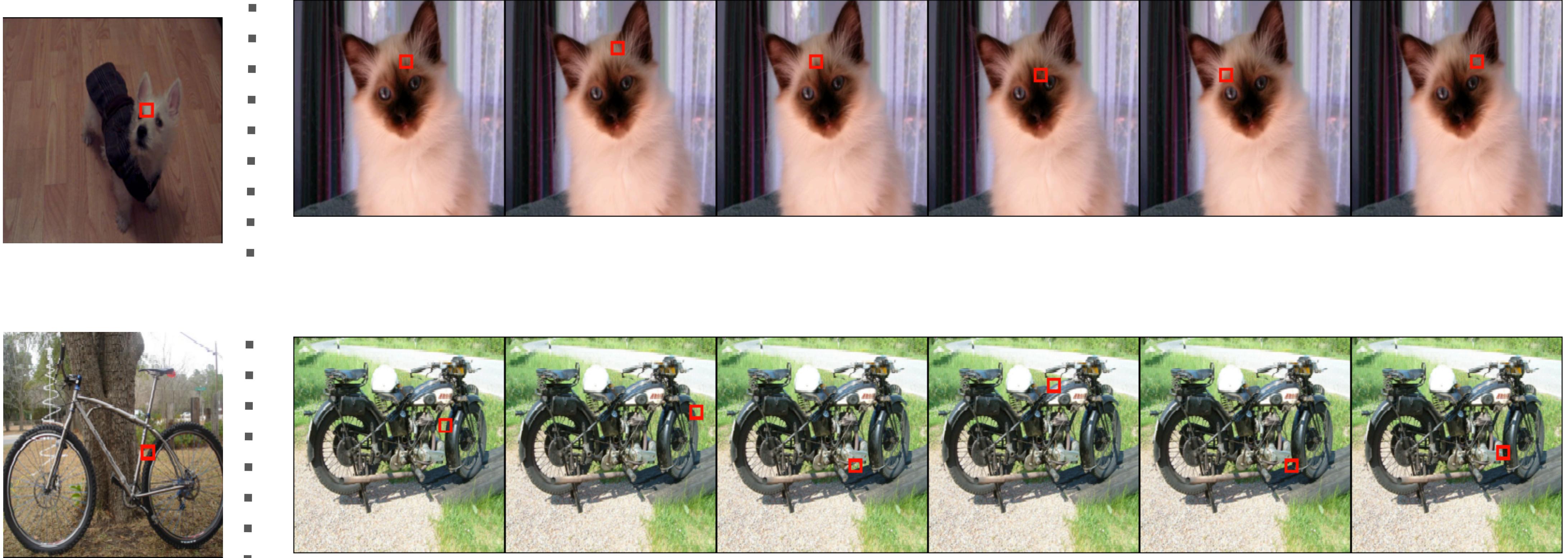
PaNeCo



NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency. Valentinos Pariza, Mohammadreza Salehi, Gertjan Burghouts, Francesco Locatello, Yuki M. Asano. arxiv 2024

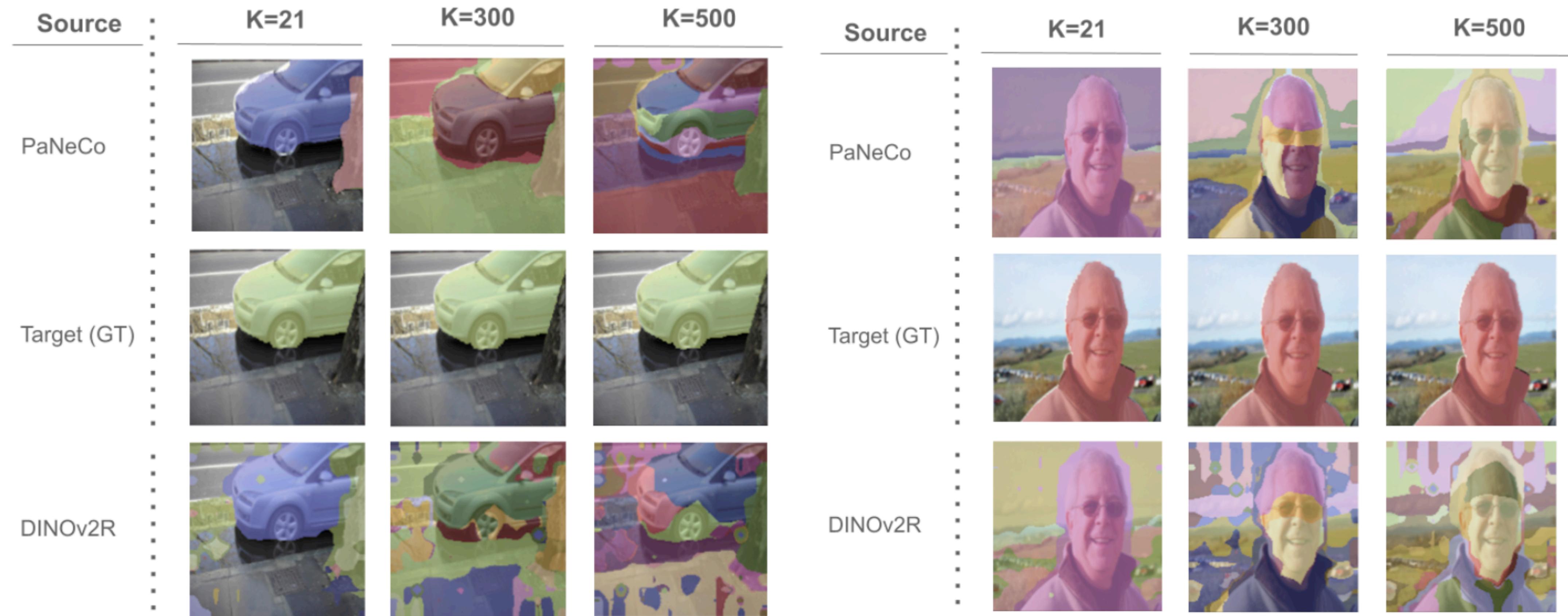
# PaNeCo rarely confuses semantically close patches

## Query Retrieved Nearest Neighbors



On average such cases appear around 6% of the times from Pascal VOC retrieval cases.

# k-Means Semantic Segmentation



# What's the sauce?



- **Dense Patch-ordering** is less well suited for post-pretraining
- We can **improve upon (very strong) DINO/ DINOv2R** models
- Strongest improvements in in-context semantic segmentation and even full-finetuning
- also: code/models now available!



*Time does tell: self-supervised time-tuning of dense image representations.*

# Current Vision Foundation Models are trained with images. Videos can enable new directions



Visual development for AI

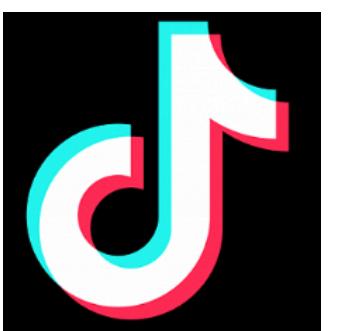


"Get" physics



Embodied AI

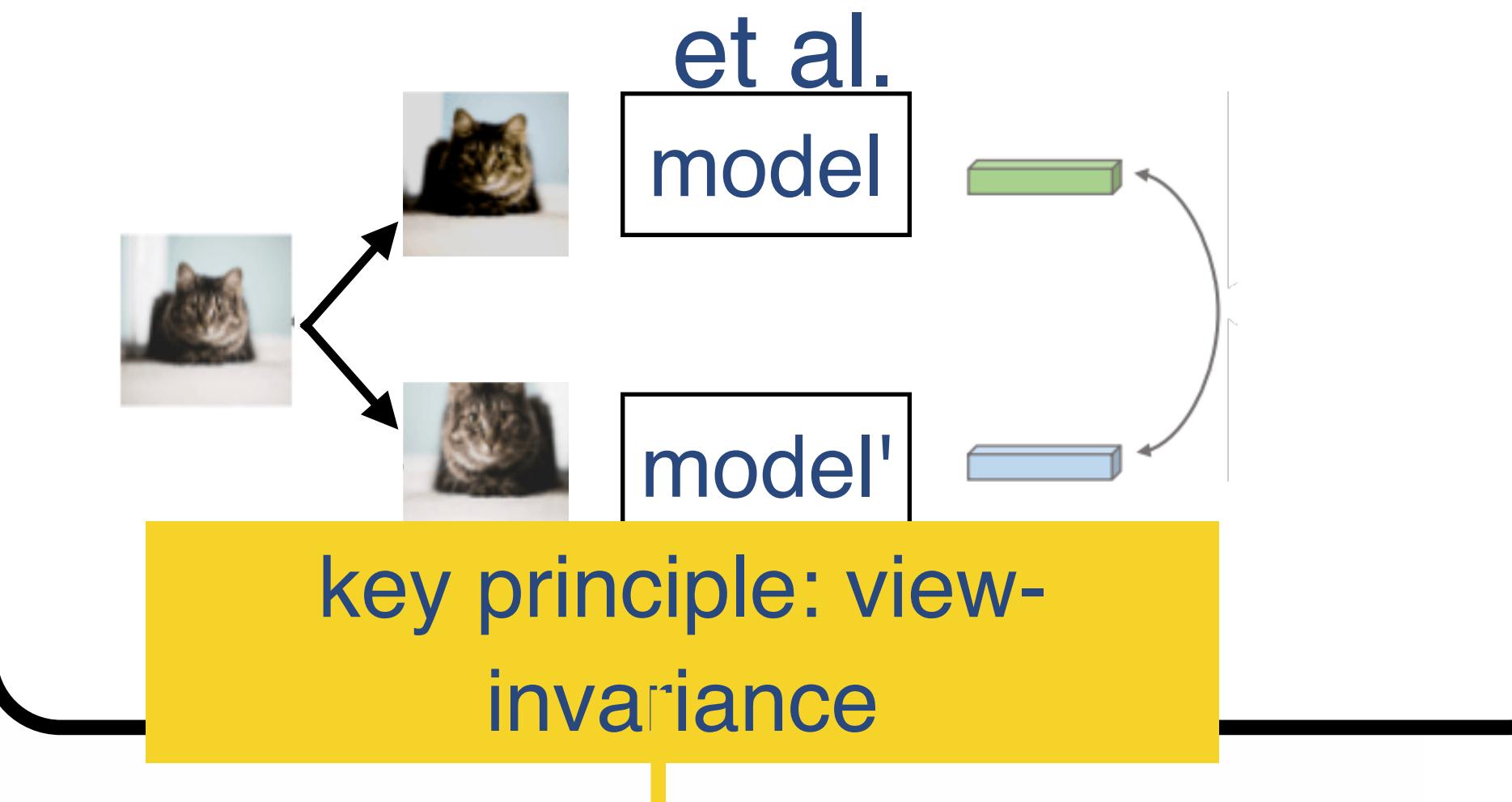
+ their *insane* scal



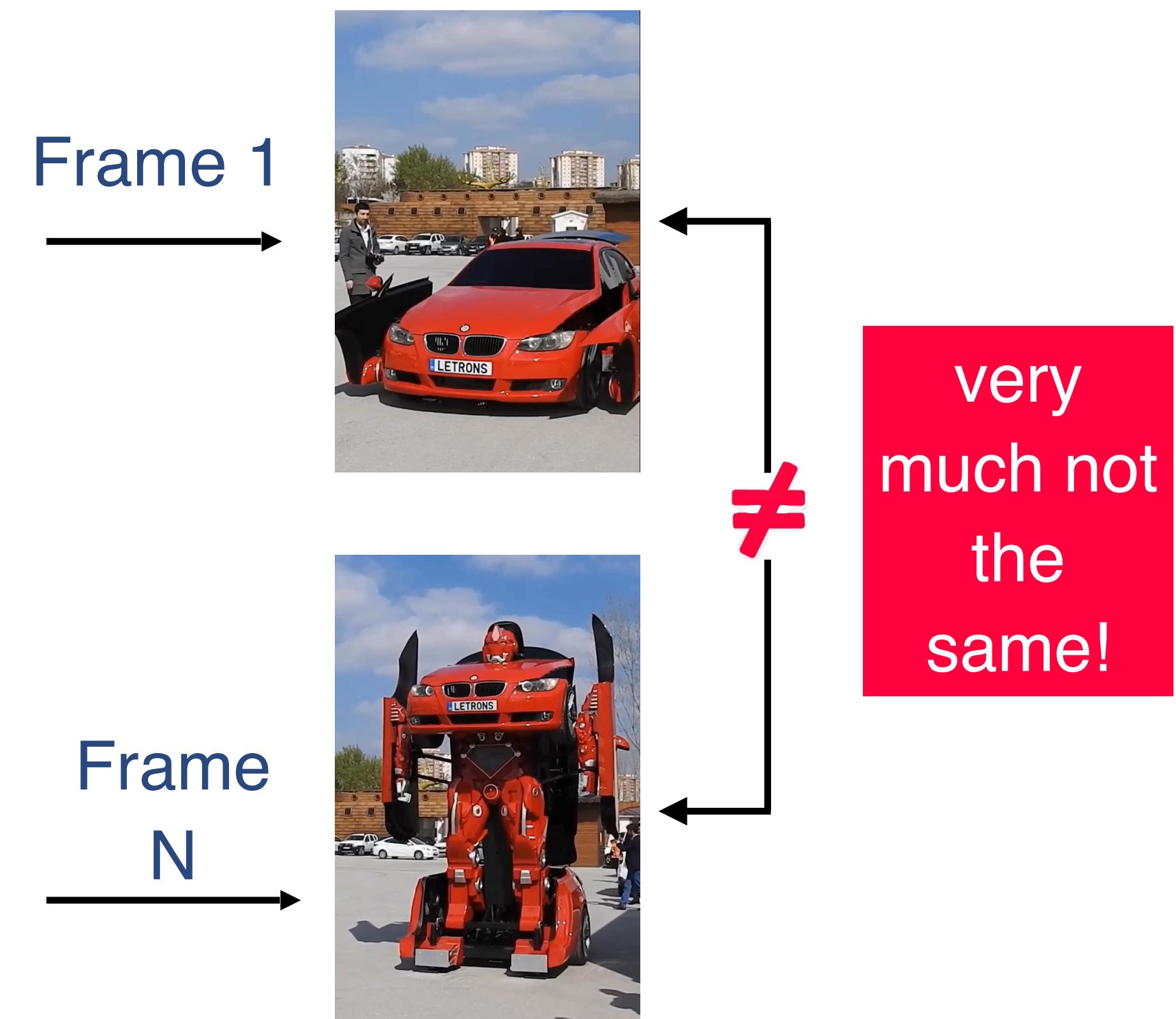
YouTube

# Augmentations are crucial in classic image-SSL, but forcing frames to be invariant is limiting

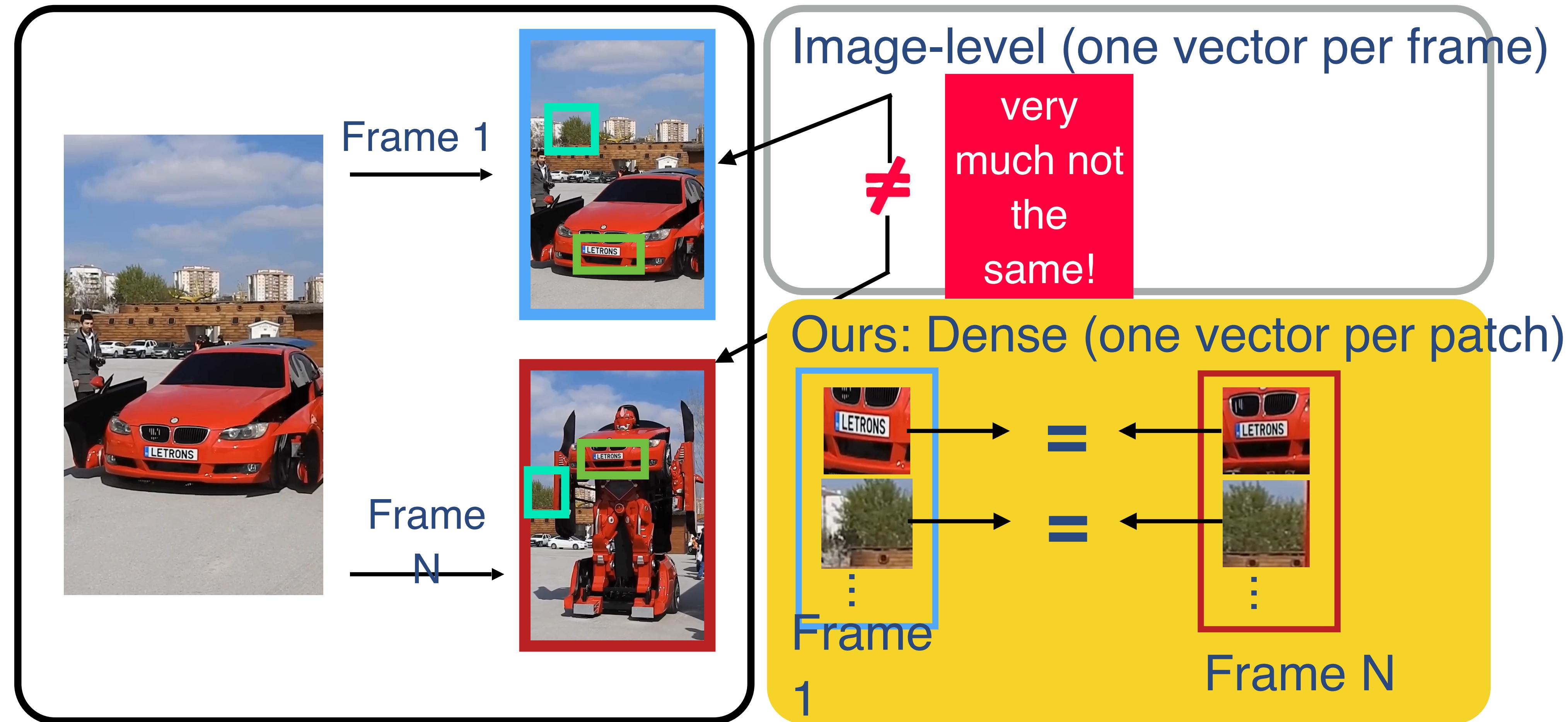
Images: SimCLR, MoCo, SwaAV  
et al.



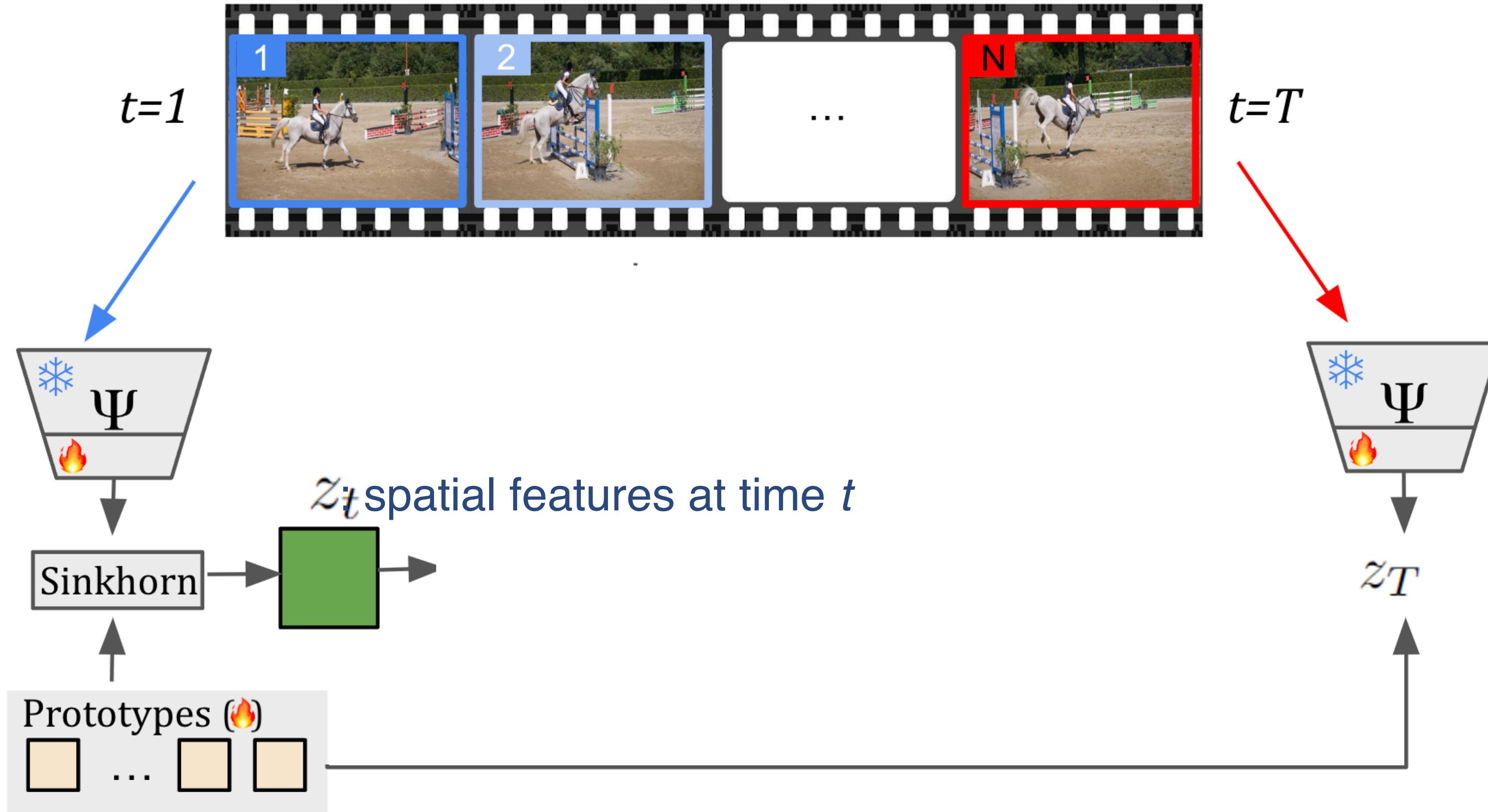
But does this generally make sense?



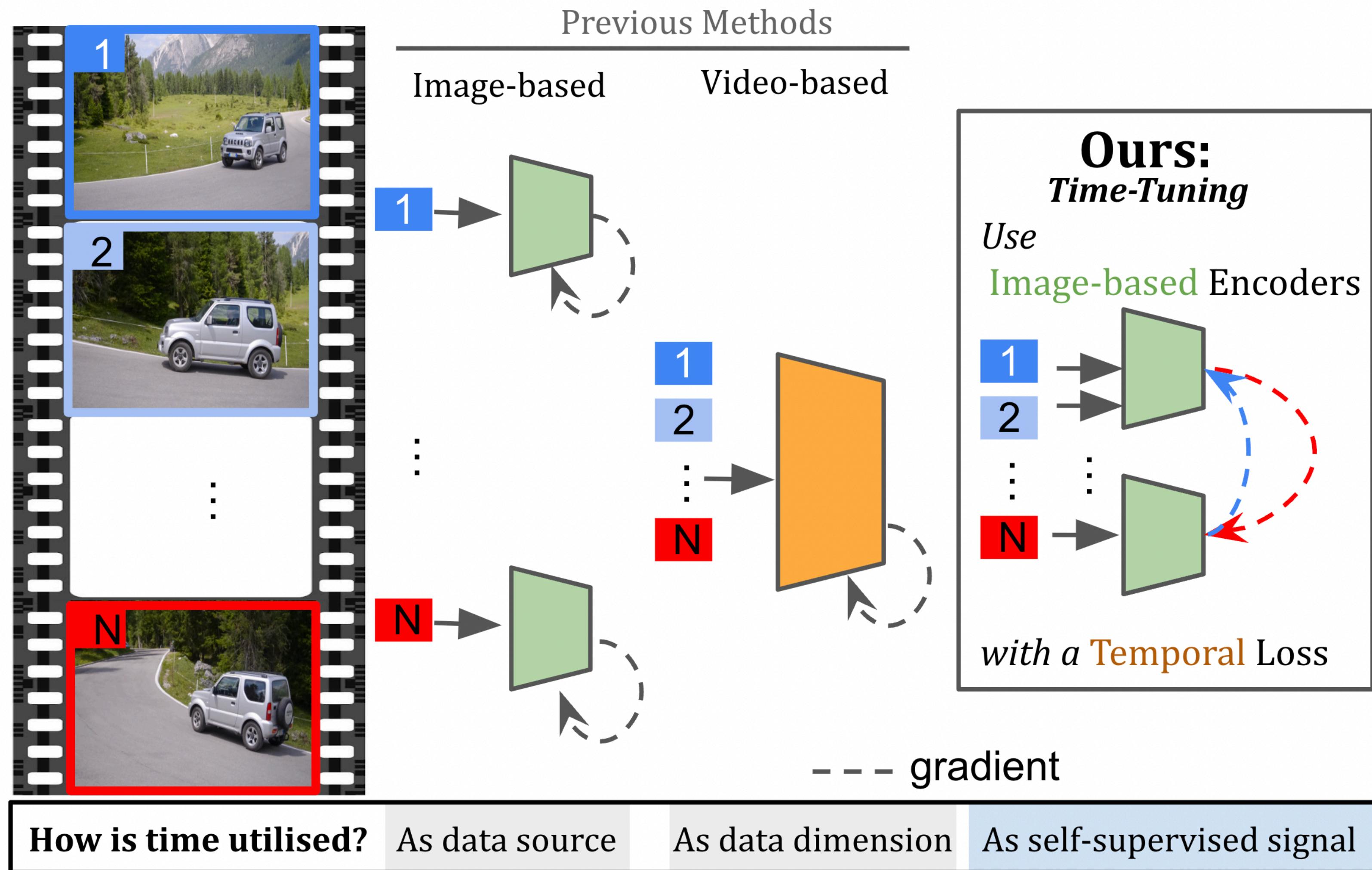
# Solution is obvious



We model a video by tracking image patches,  
and aligning their clustered features

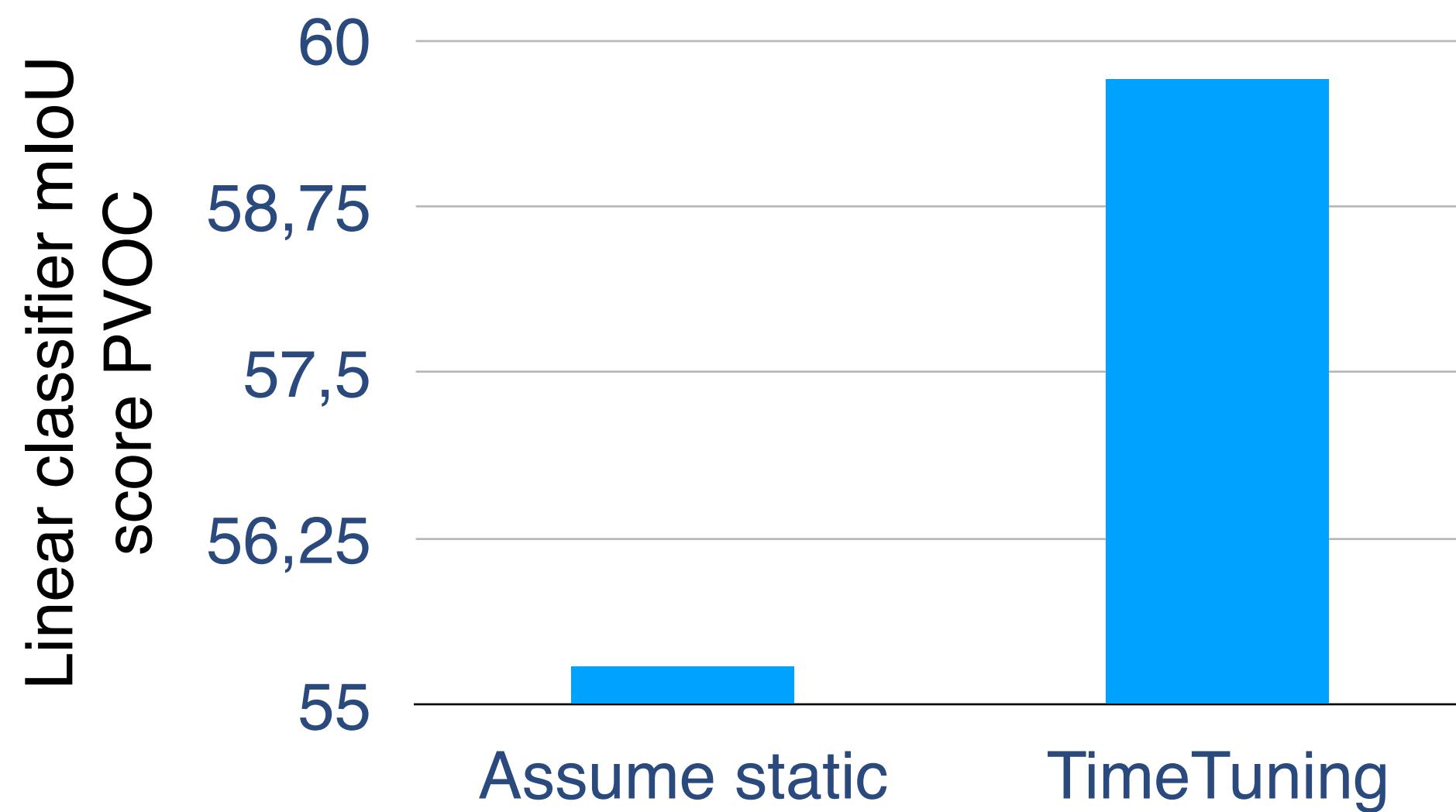


# Using videos to learn self-supervised image encoders

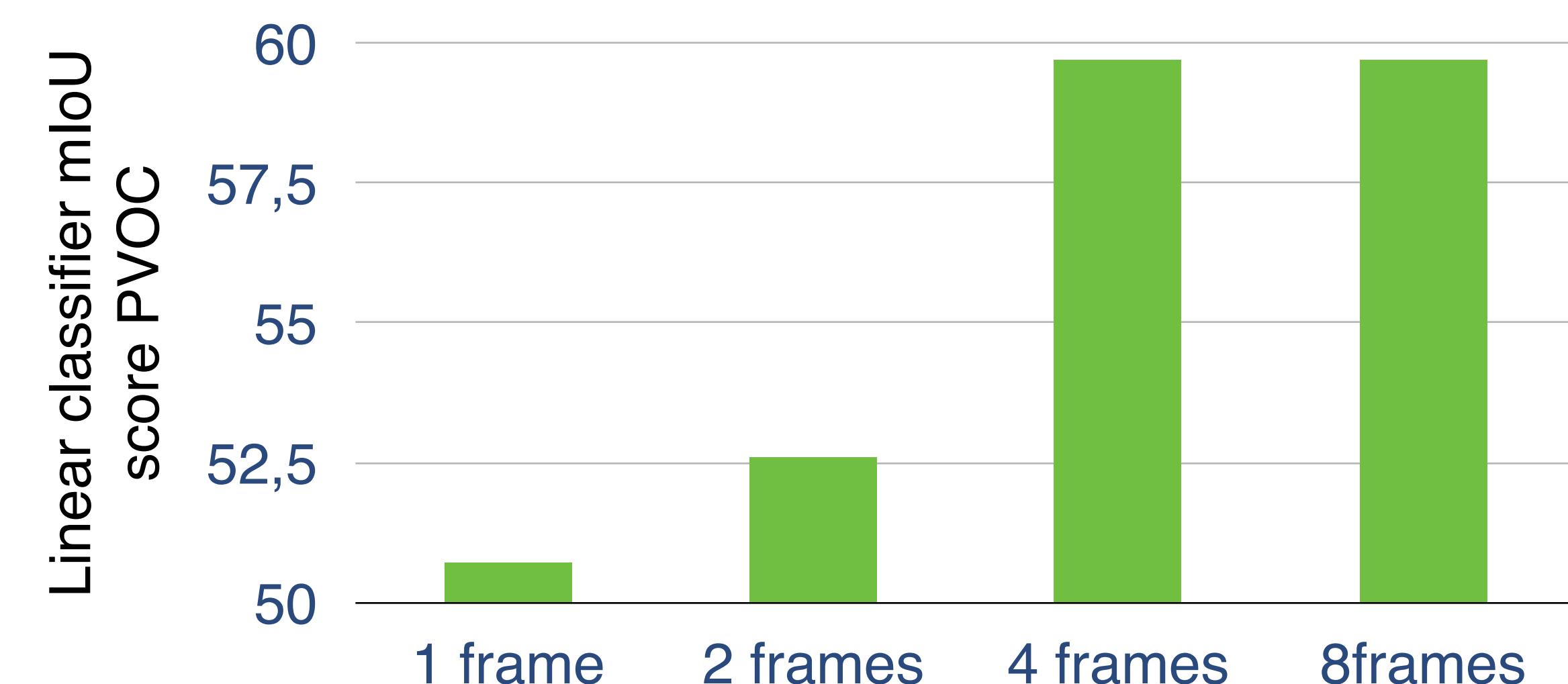


# Ablations demonstrate using time helps learn better features

Modelling time is  
esential



Model learns from temporal  
info



# Results

## SoTA on unsupervised video segmentation

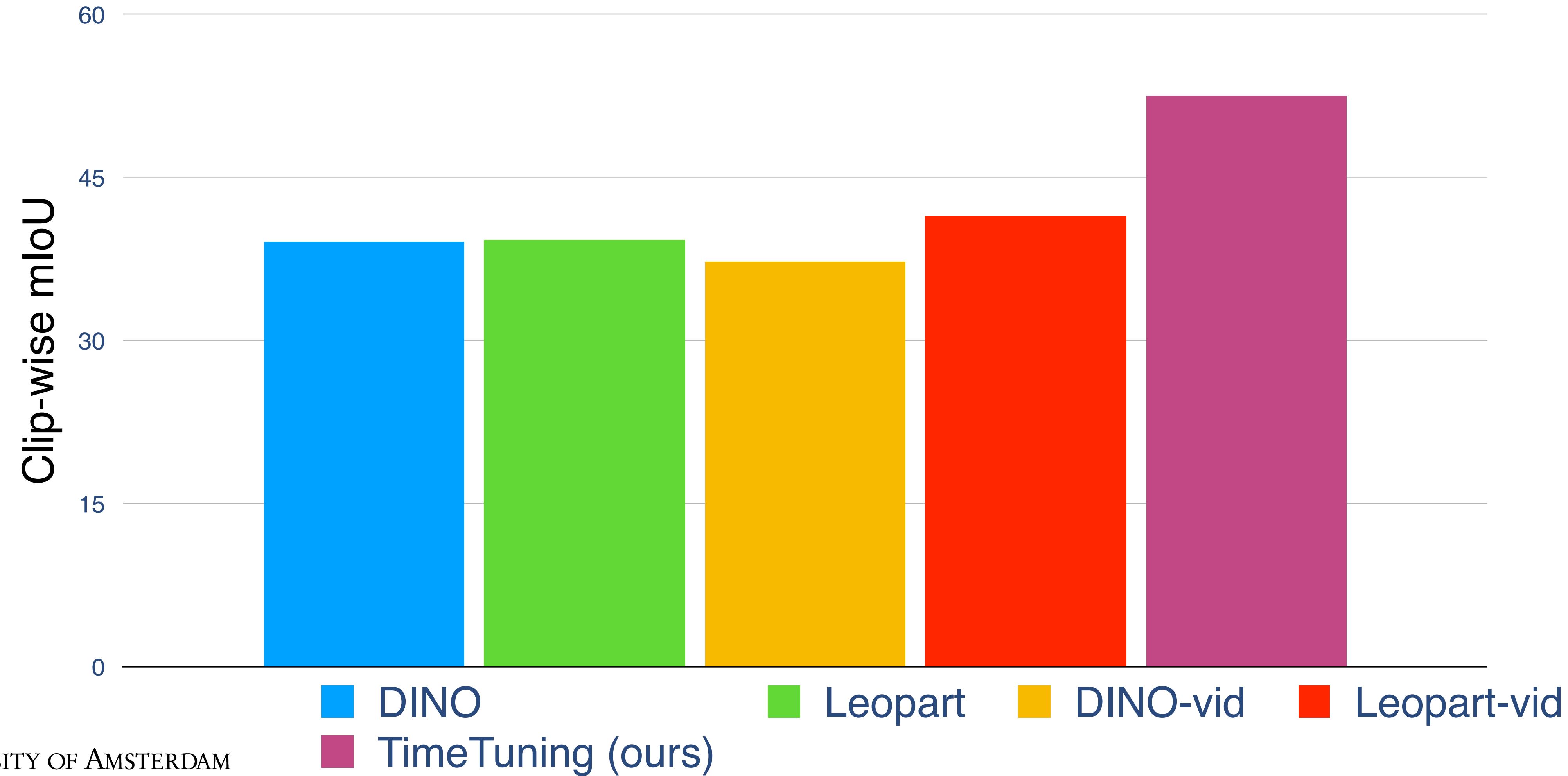
	Clustering					
	YTVOS			DAVIS		
	F	C	D	F	C	D
<i>Trained on Images</i>						
Resnet50	44.0	43.4	1.7	39.3	37.4	4.2
SwAV [8]	39.5	38.2	3.2	32.0	29.6	7.3
DINO [9]	39.1	37.9	1.9	30.2	31.0	1.6
Leopard [74]	39.2	37.9	11.7	30.3	30.2	16.5
<i>Trained on Videos</i>						
STEGO*	41.5	40.3	2.0	31.9	31.0	3.2
DINO*	37.2	36.1	1.2	29.3	29.2	2.4
Leopard*	41.5	40.5	7.7	37.5	36.5	12.6
<b>TIMET(ours)</b>	<b>52.5</b>	<b>51.3</b>	<b>13.3</b>	<b>53.7</b>	<b>53.0</b>	<b>20.5</b>

## SoTA on unsupervised image segmentation

	Pascal VOC			
	K=21	K=500	LC	FCN
<i>Trained on Images</i>				
ResNet-50	4.5	36.5	53.8	-
DINO [9]	5.5	17.4	50.6	60.6
SwAV [8]	11.6	35.7	50.7	-
MaskContrast [57]	35.0	45.4	49.2	-
DenseCL [61]	-	43.6	49.0	69.4
STEGO [21]	7.0	19.5	59.1	63.5
CrOC [52]	20.6	-	61.6	-
Leopard [74]	<b>36.6</b>	50.5	<b>68.0</b>	70.1
<i>Trained on Videos</i>				
STEGO*	4.0	15.5	51.1	55.5
Leopard*	14.9	21.2	53.2	63.2
Flowdino <sup>†</sup> [70]	-	-	59.4	-
<b>TIMET (ours)</b>	34.5	<b>53.2</b>	<b>68.0</b>	<b>70.6</b>



# Results on unsupervised video semantic segmentation



# Unsupervised Semantic Segmentation on videos

[simply running k-means on a couple of videos' spatial features, k=10]

DINO



✗ part-centric maps

STEGO



✗ noisy maps

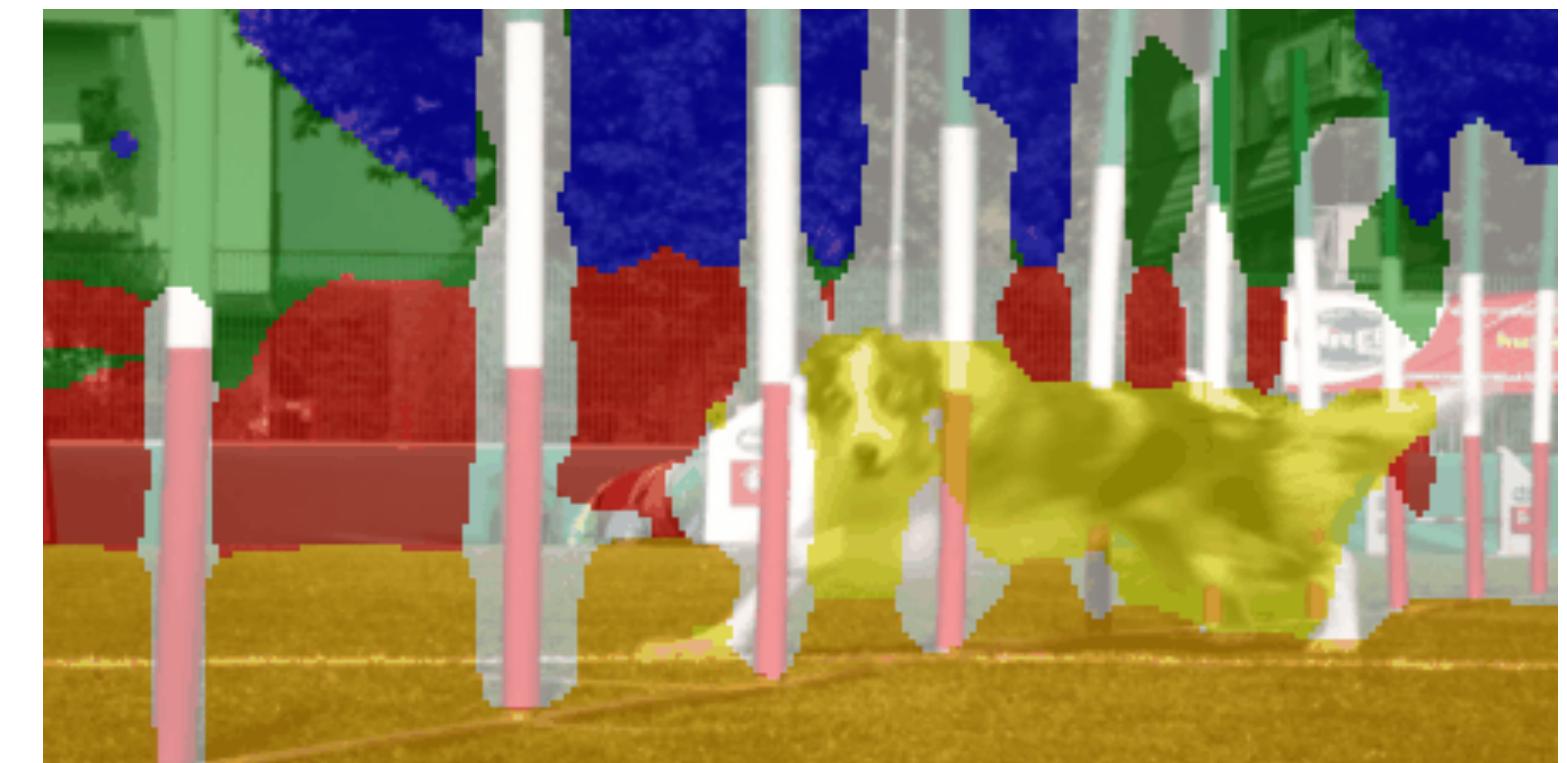
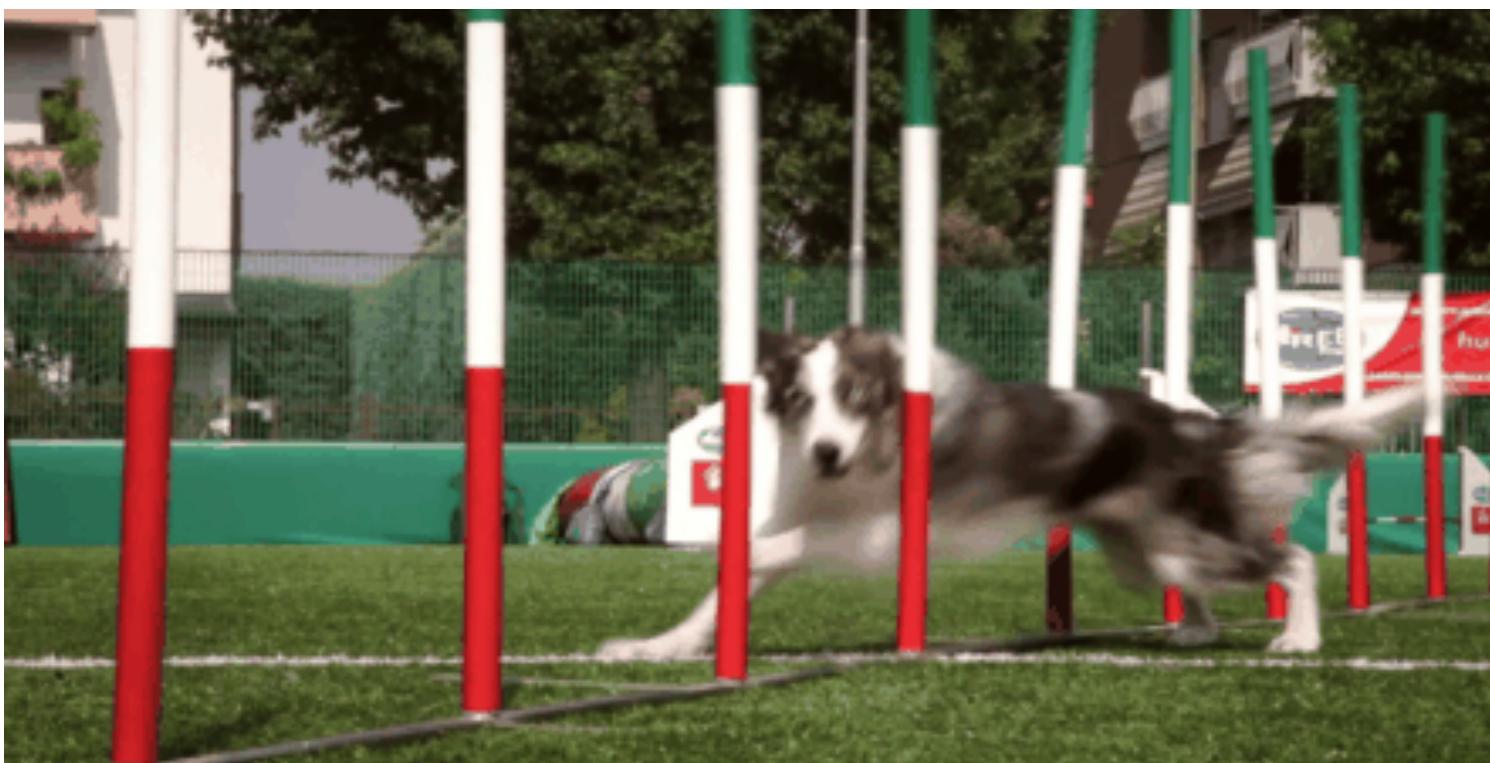
Ours



✓ crisp semantic  
maps

# Unsupervised Semantic Segmentation on videos

[here: running k-means on the whole video's spatial features, k=5]



# What's the sauce?



- **Videos** provide rich supervision signal
- Don't use frame-wise invariance across time, but instead patch-level invariance
- We can **improve upon the (strong) DINO model**
- Strongest improvements in unsupervised segmentation



*Is ImageNet worth 1 video? Learning strong image encoders from 1 long unlabelled video.*

Shashanka Venkataraman, Mamshad Nayeem Rizve, João Carreira, Yannis Avrithis\*,  
Yuki M. Asano\*

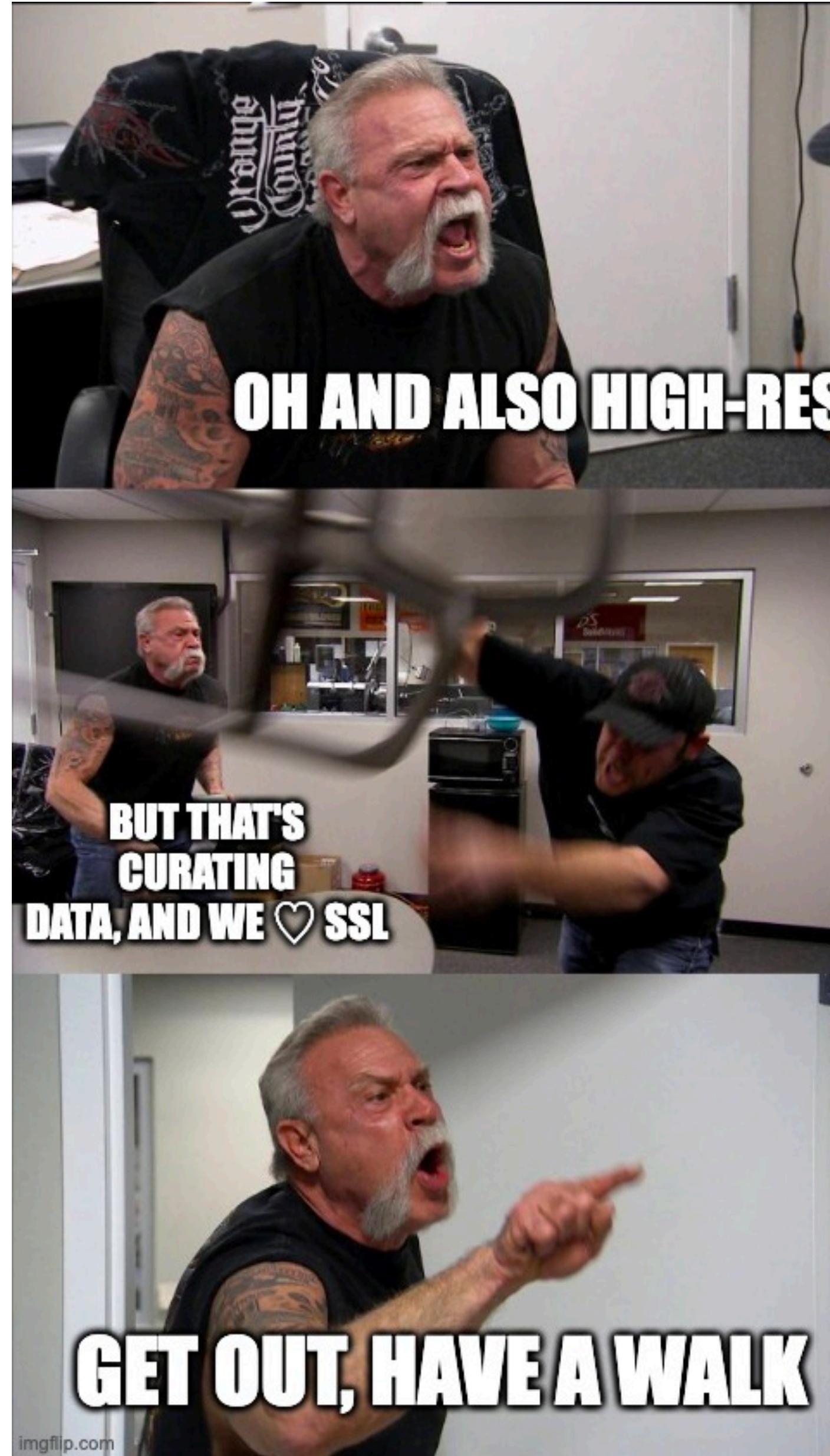
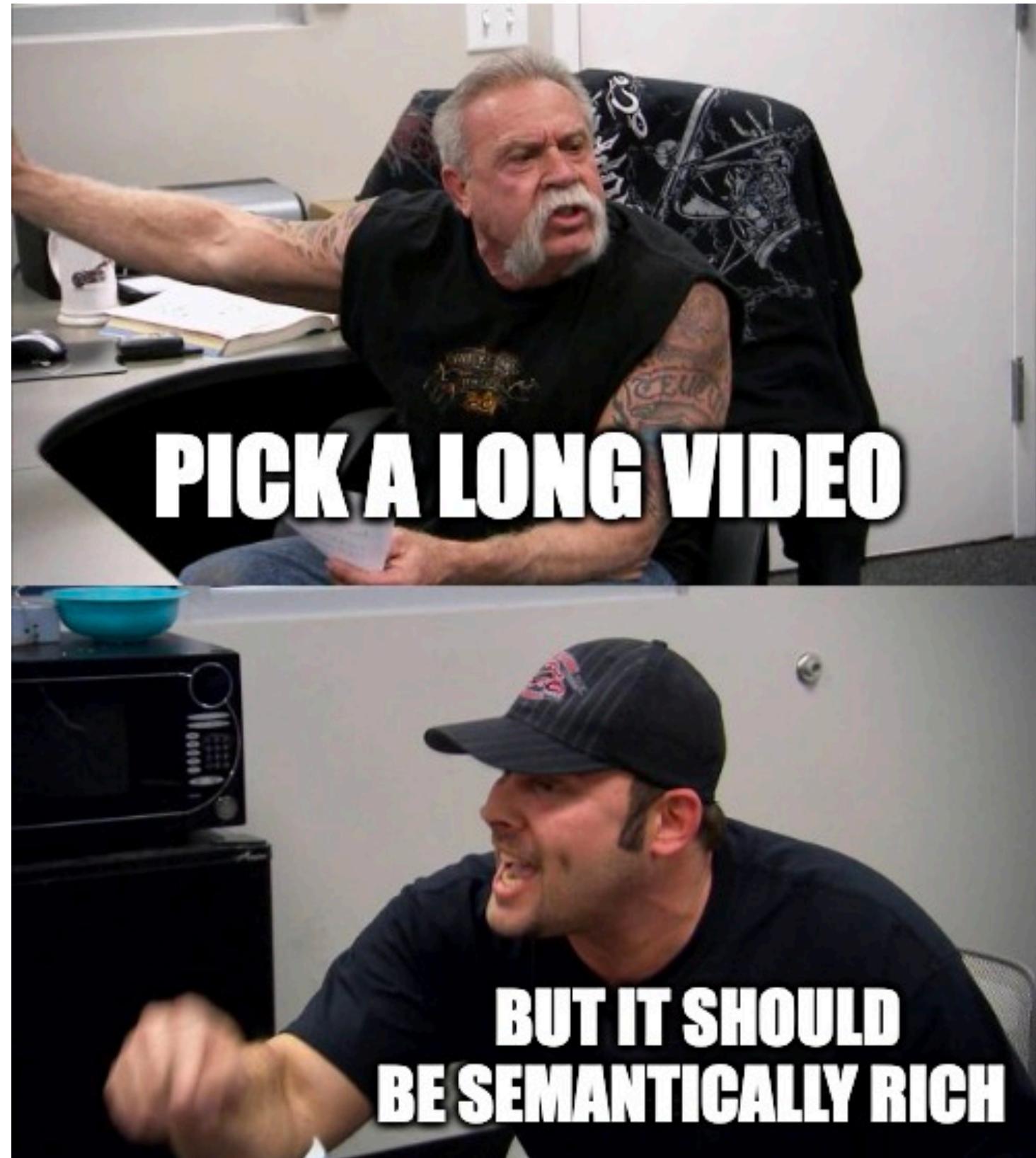
<https://www.barilla.com/it-it/ricette/tutte/farfalle-con-fave-e-pesto-ricotta-e-noci>

**TimeTuning:**  
DINO as init &  
use temporal info  
of videos.

~~How powerful is time~~  
without image-

Study the  
extreme: try to  
learn from a  
**single video,**  
**from scratch.**

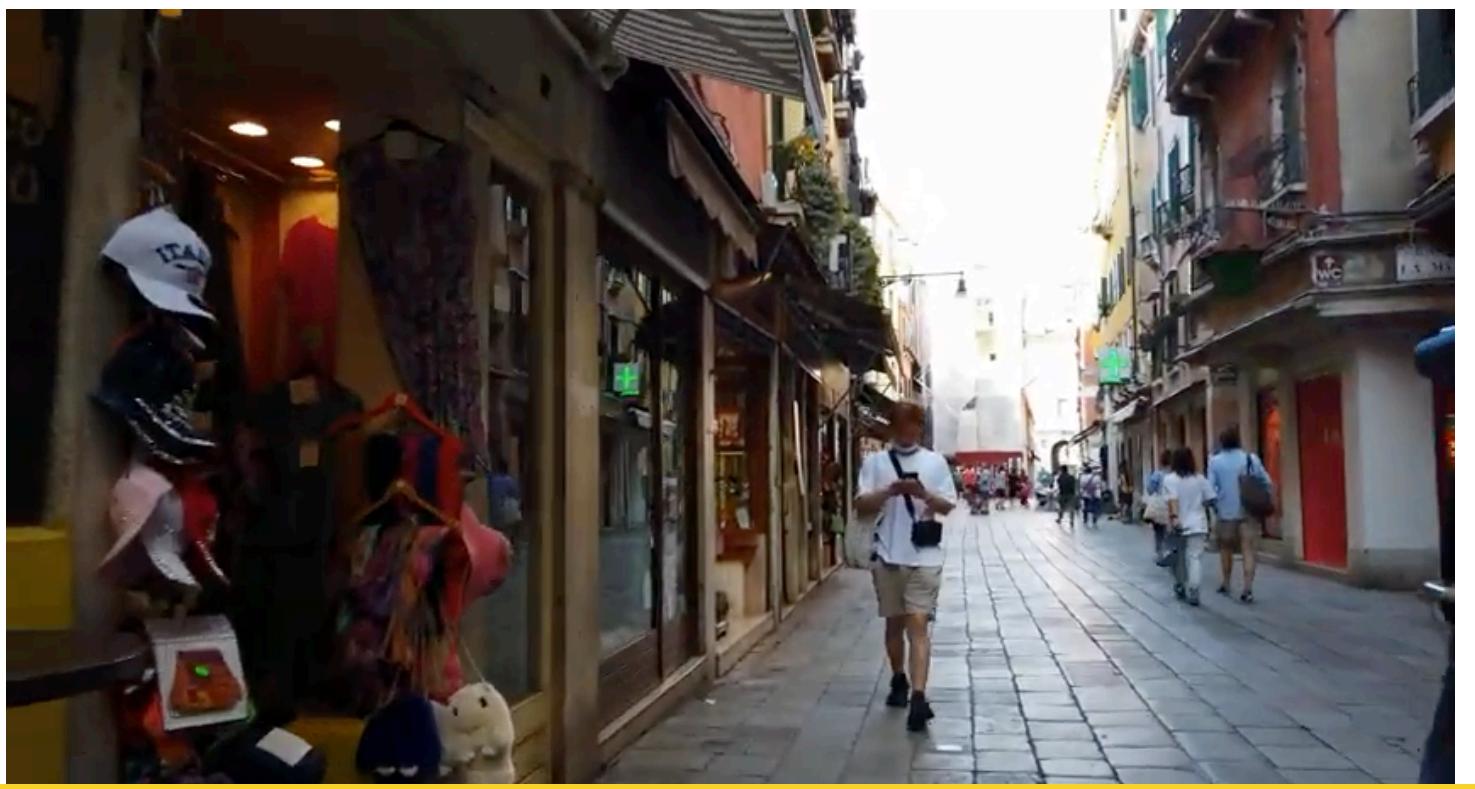
# us figuring out



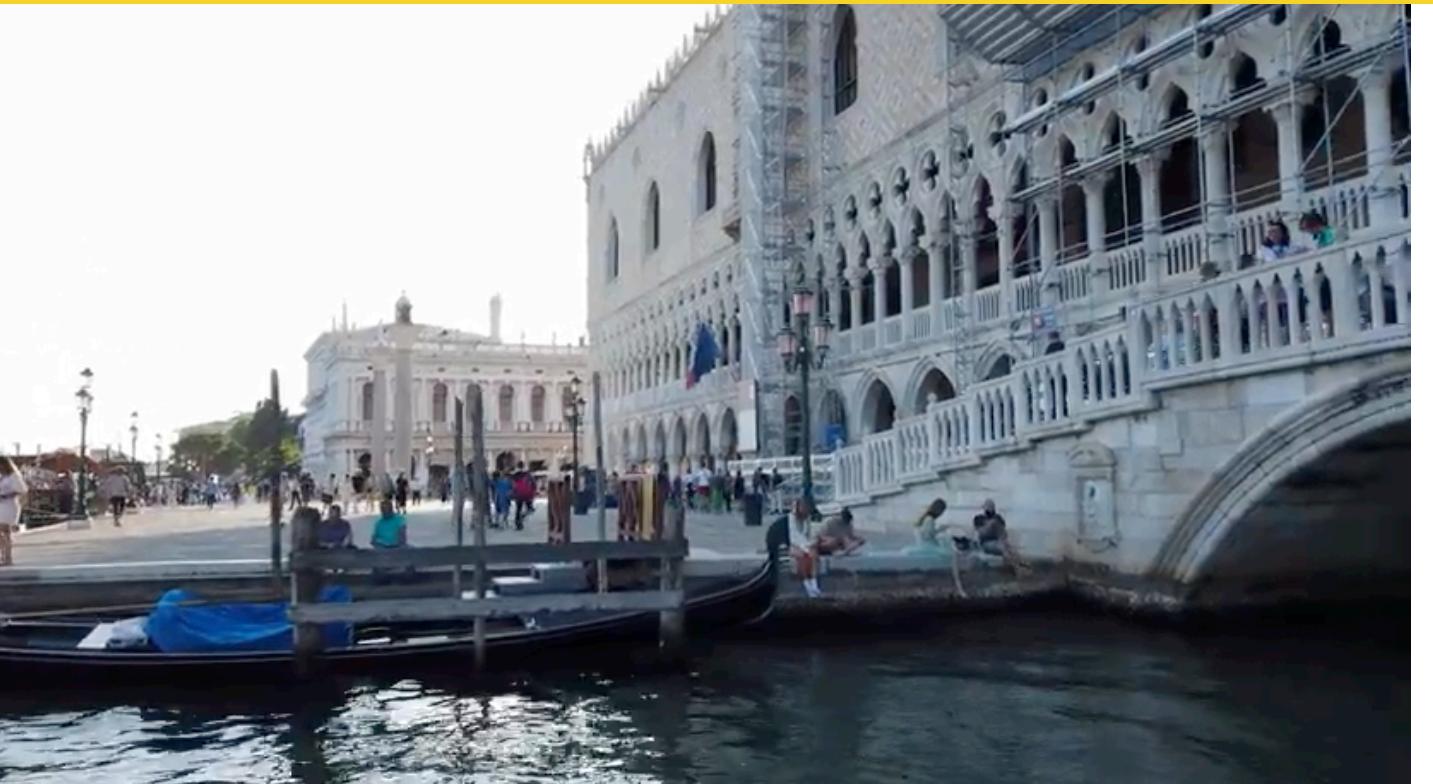
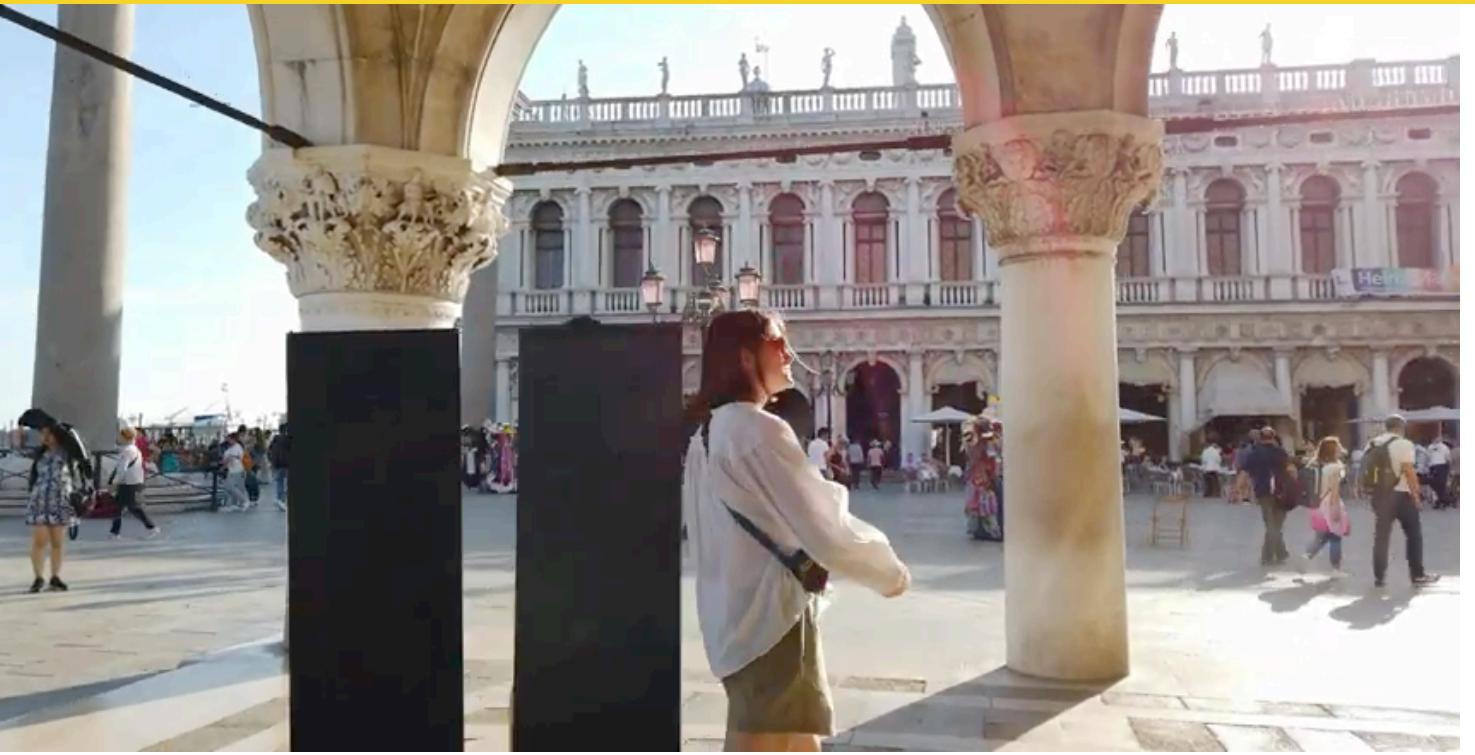
- ✓ Long
- ✓ High-res, smooth
- ✓ Semantically rich
- ✓ Scalable (we ❤ SSL)



## Walking Tours



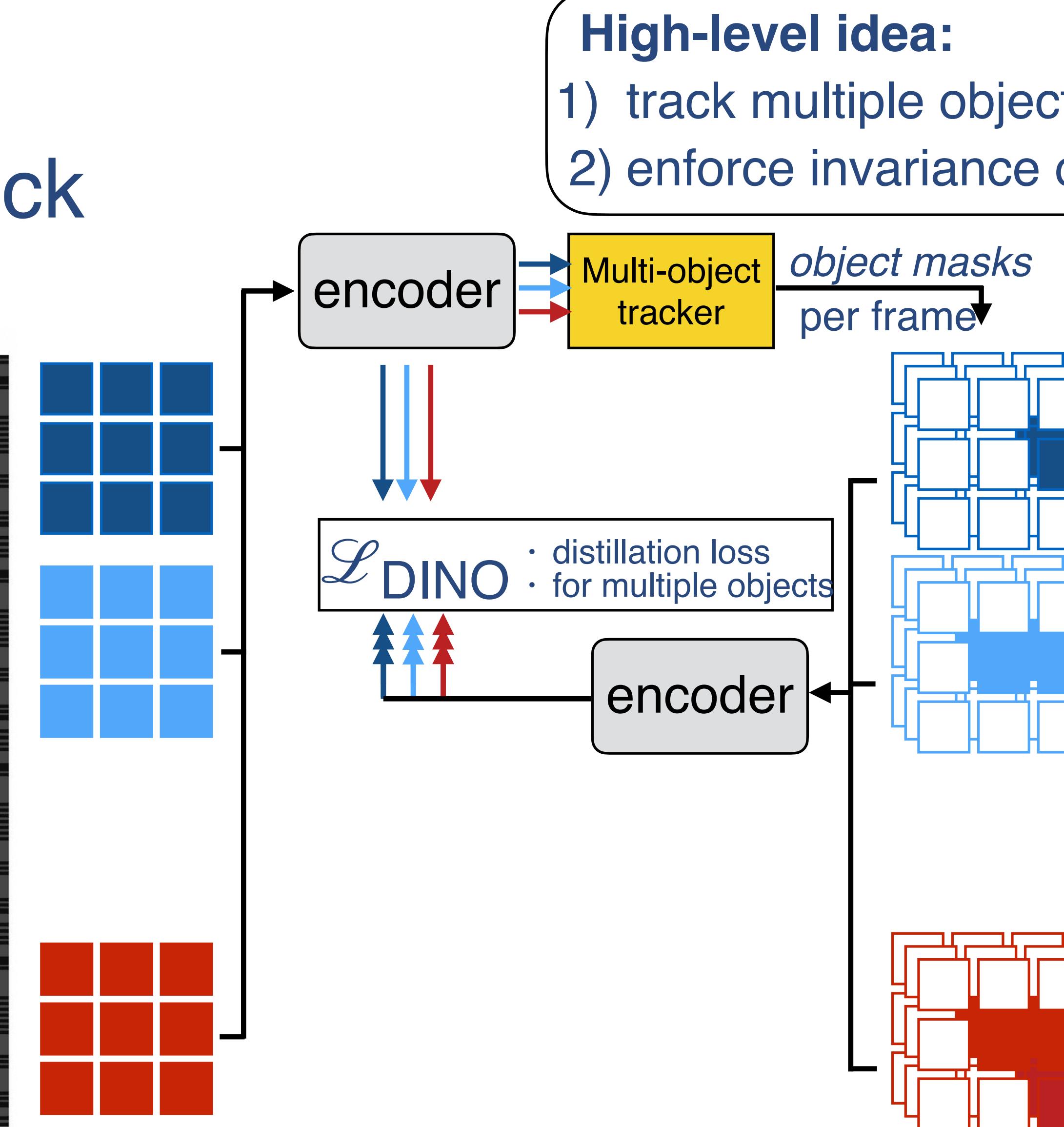
The dataset consists of 10x 4K videos of different cities' Walking Tours.



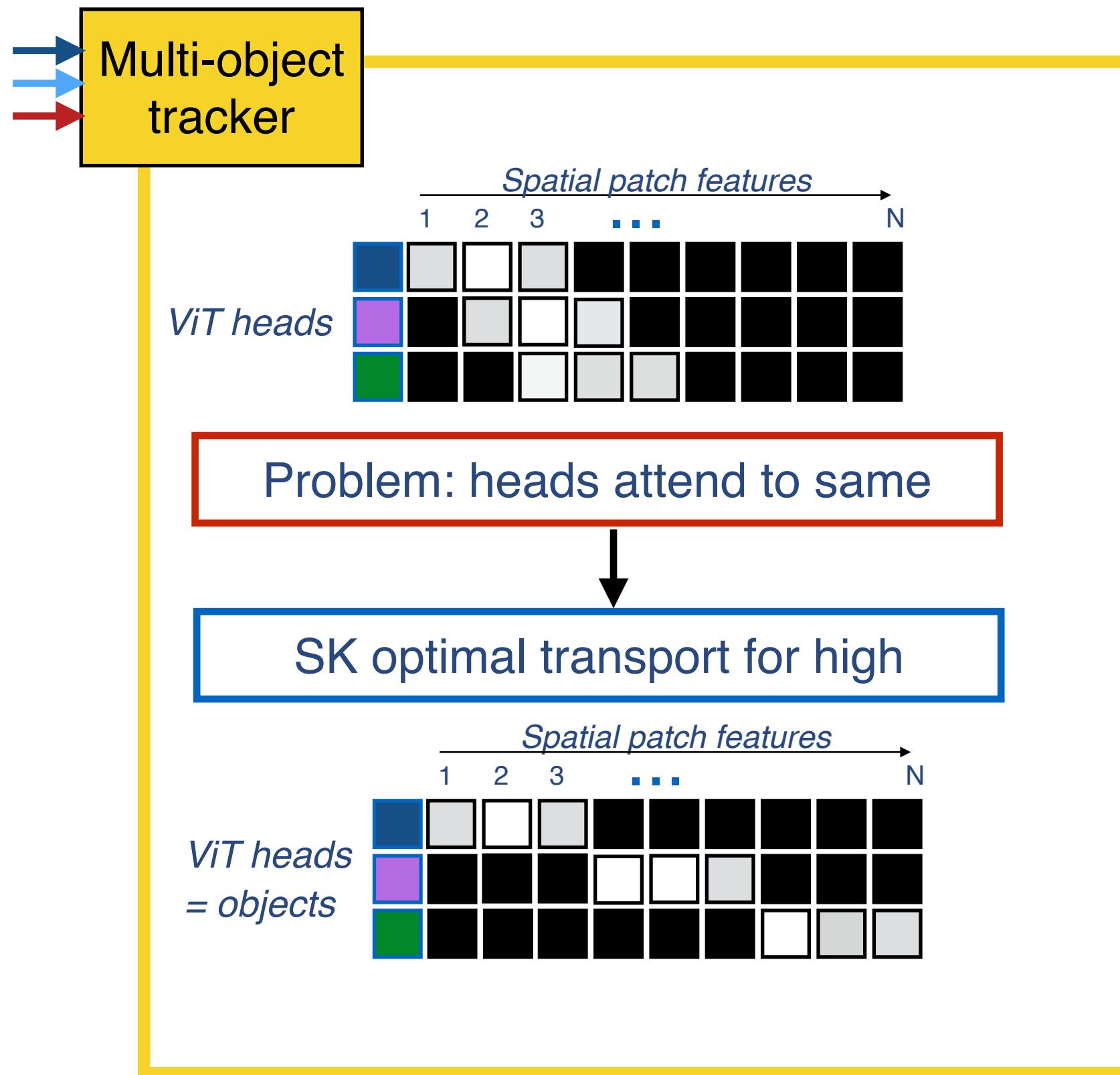
# Dora: Discover and Track



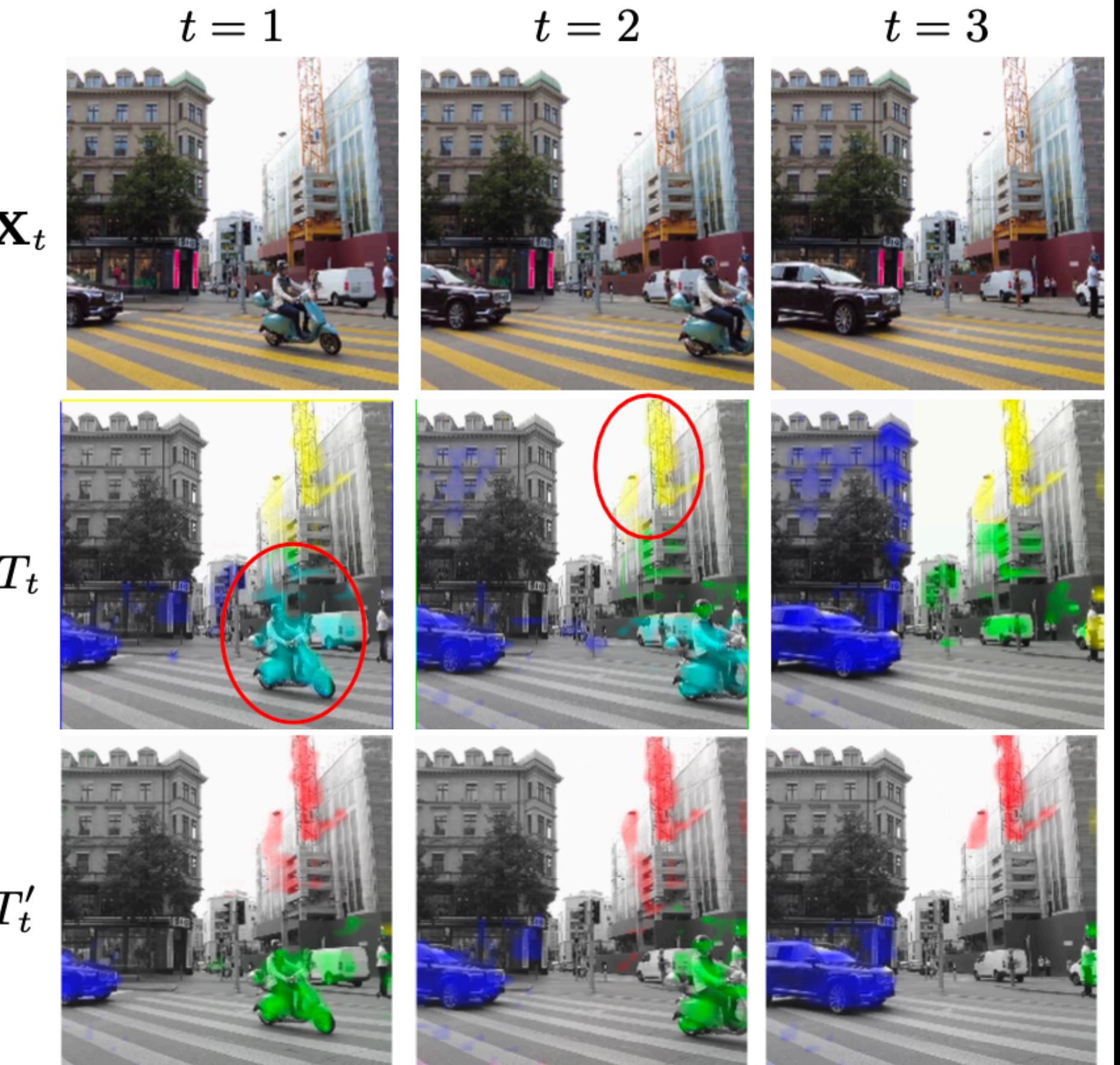
Much like Dora, we walk around and learn from what we see.



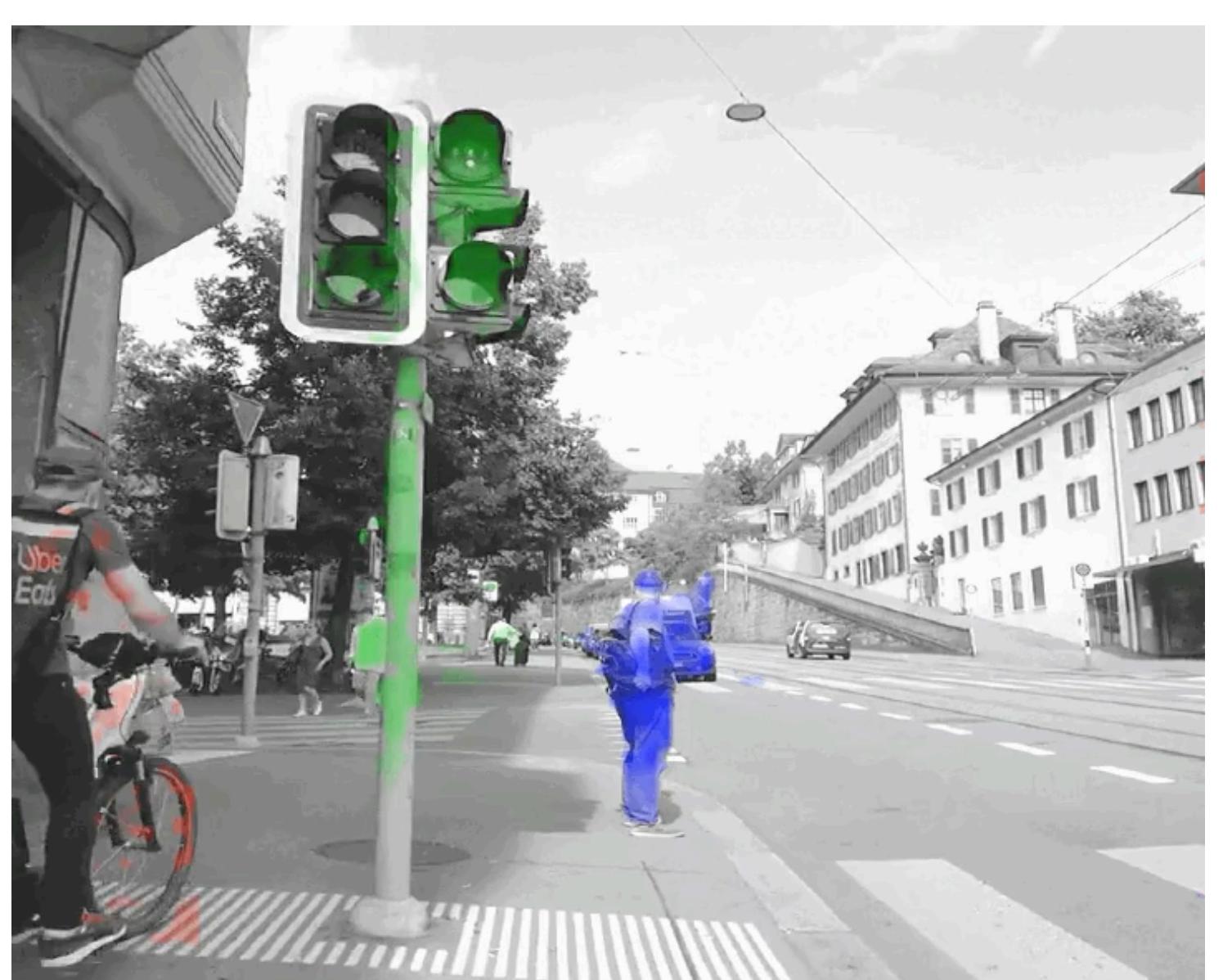
# Spreading attention with Sinkhorn-Knopp



Visualise attention of  
3 heads with colors  
R,G,B

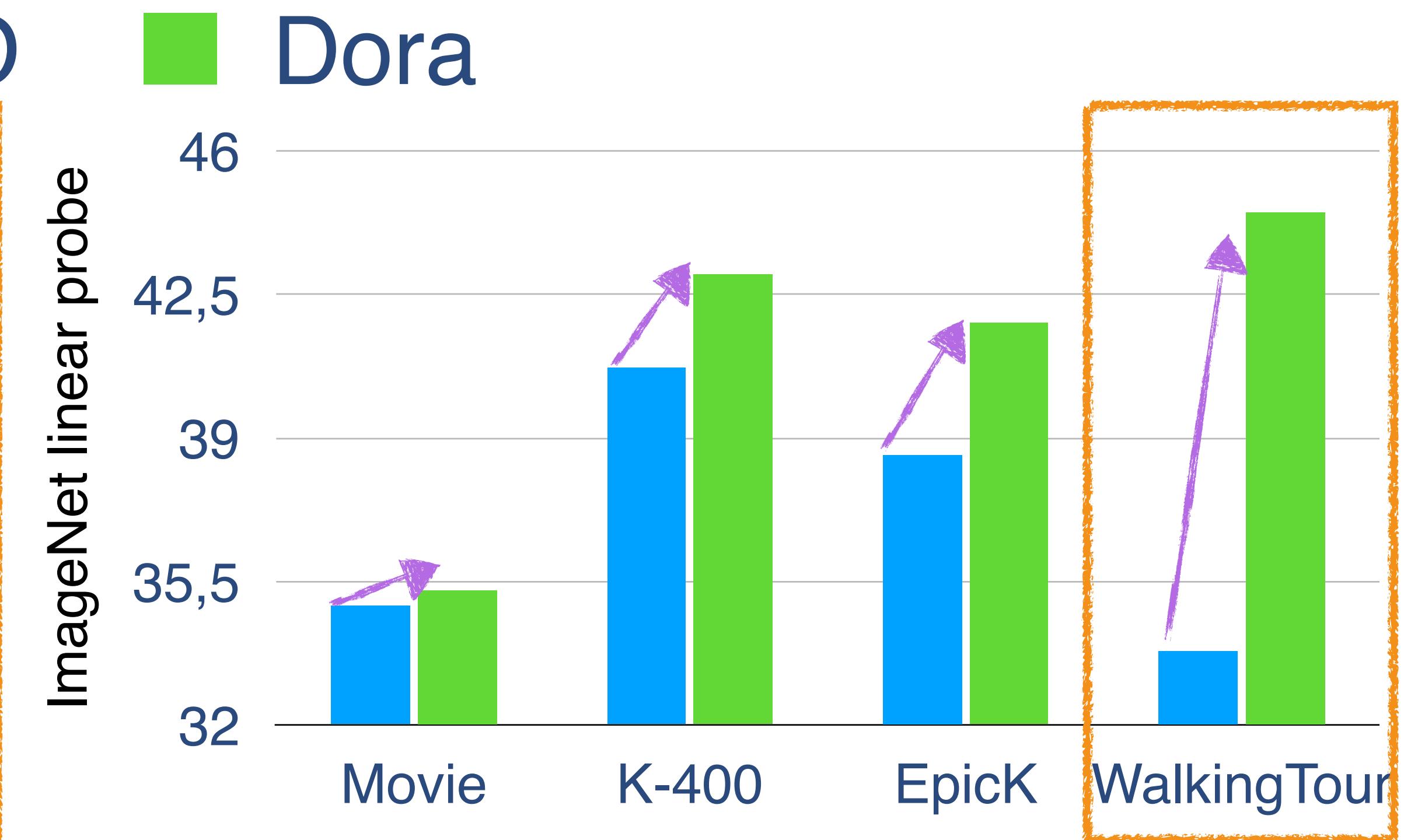
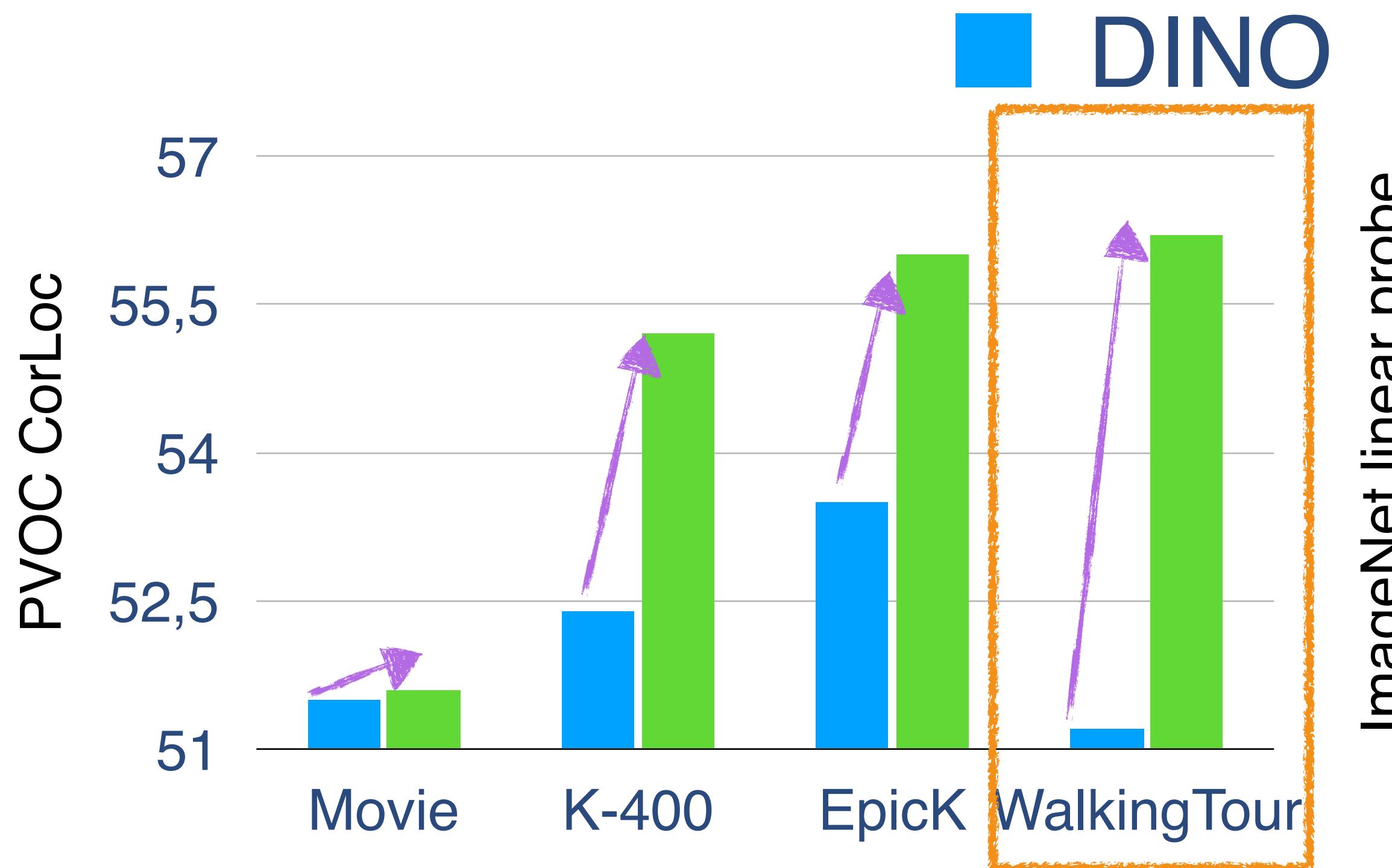


# More examples: multi-object tracking in a ViT emerges

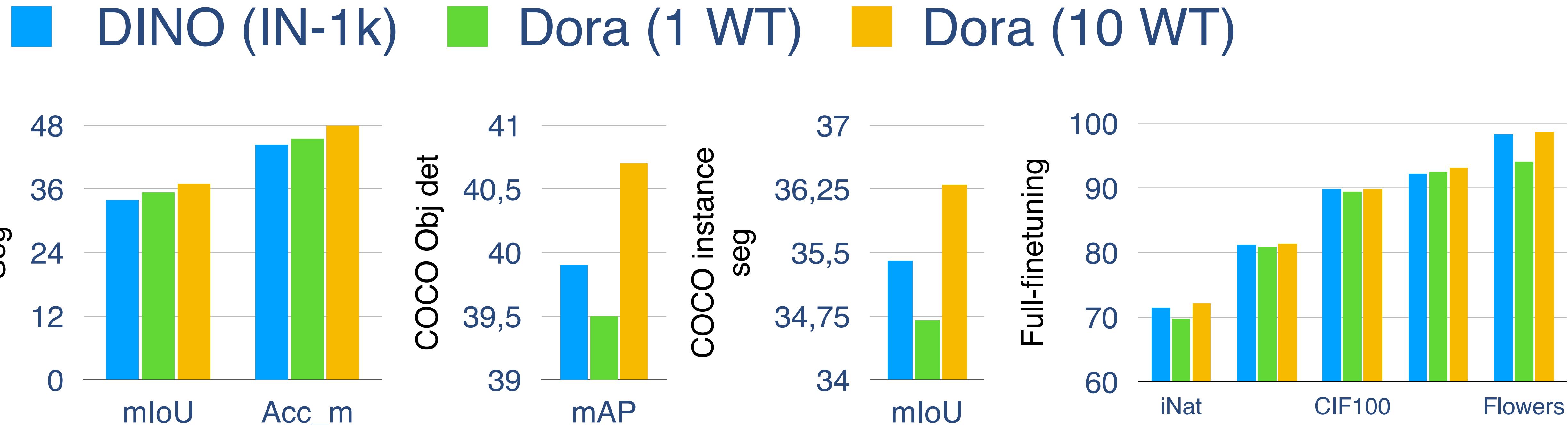


# Dora better than DINO

## WT+ Dora: great match



# But how does it compare against ImageNet pretraining?



Dora (1WT) ~ on par with DINO  
(IN-1k)

# What's the sauce?



- Training strong encoders **from scratch** with 1 video is possible
- Models match DINO (trained on ImageNet) in terms of performance
- The training loss is **spatially dense** and leverages **time**
- **Multi-object tracking** emerges
- **Walking videos** are great for training vision models

## Computer vision needs more post-pretraining

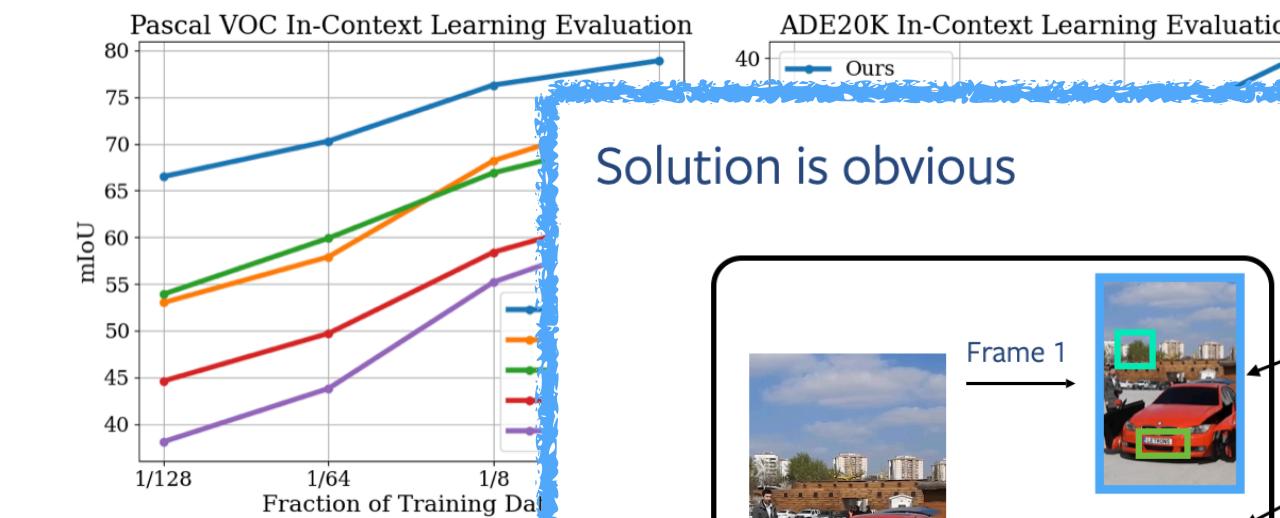


### 1st-gen Vision Foundation Models (e.g. DINO, CLIP)

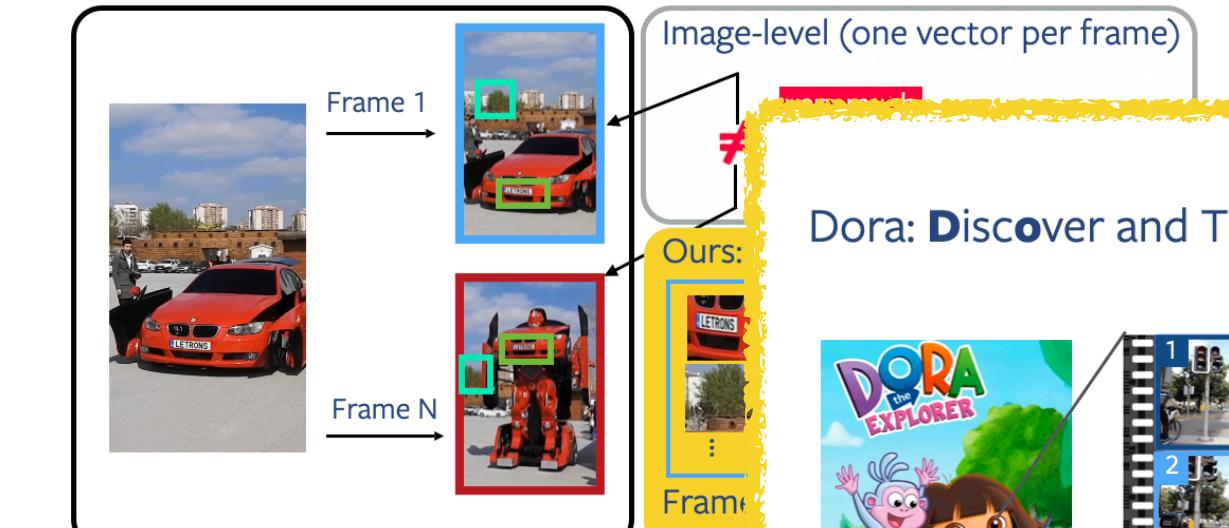
✗ requires industrial compute  
✓ model is already useful

UNIVERSITY OF AMSTERDAM

### In-context scene understanding benchmark



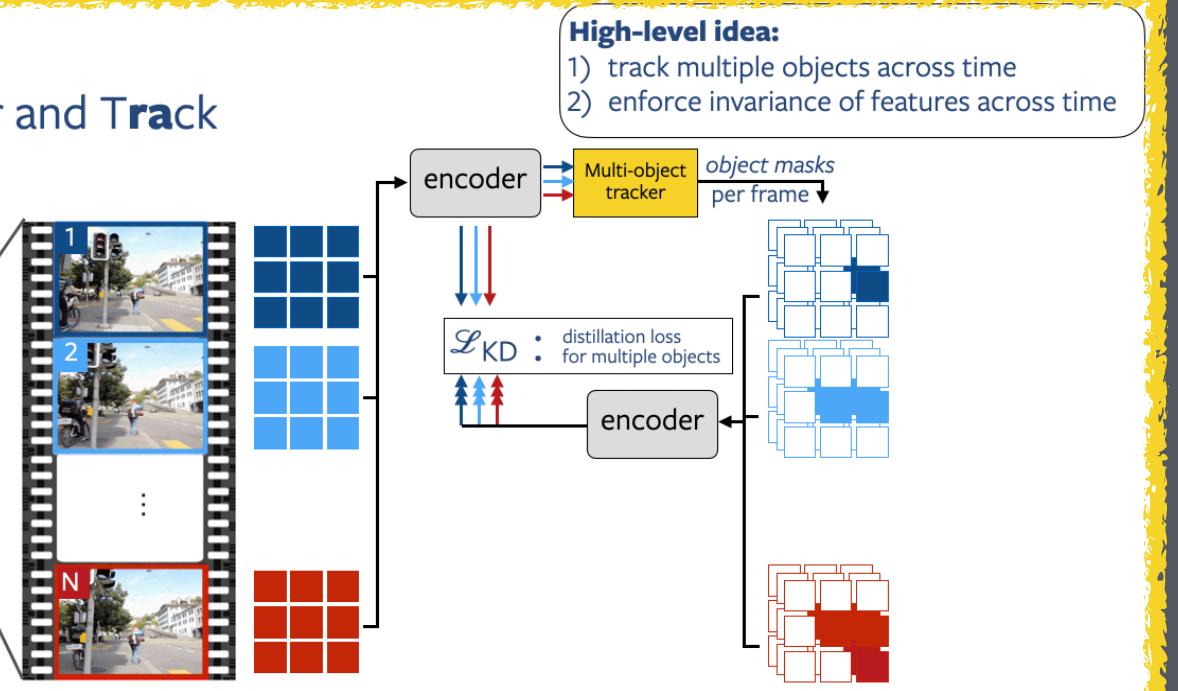
### Solution is obvious



### Dora: Discover and Track



Much like Dora, we walk around and learn from what we see.



Future Foundation Models will be massively pretrained with videos.  
Post-pretraining has a large potential that we're only beginning to exploit

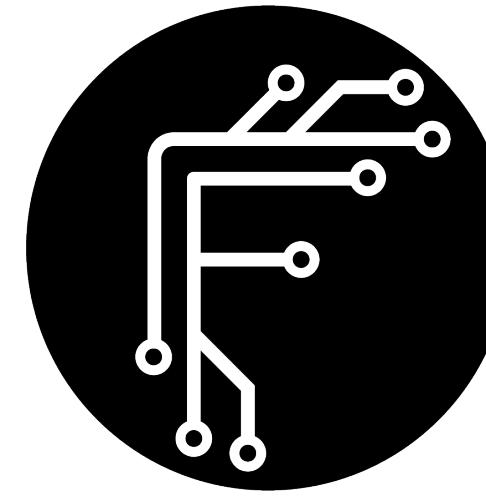
# Who gave the talk again?

Oh, hi, I'm Yuki

- Currently Assistant Professor with Video & Image Sense (VIS) Lab
- In two days:  
Full Professor and head of Fundamental AI Lab, University of Technology  
Nuremberg



@y\_m\_asano



**Fundamental  
AI Lab**

- Our focus:
  - Self-supervised Learning
  - Multimodal Learning
  - Large Language Models
- Let's collaborate!