Better Foundation Models:

Self-supervised Learning for Generalisation

A25 - Computer Vision by Learning

Hi, I'm Yuki

- Full Professor and head of Fundamental AI Lab at UTN
  - Self-supervised Learning
  - Multimodal Learning
  - Large Model Adaptation

- More info: https://fundamentalailab.github.io/,
- yuki.asano@utn.de

Self-Supervised Representation Learning

- Novel SSL algorithms
- Better visual Foundation Models
- Synthetic data and generative models

Vision-language Learning

- Vision-Language Models
- Data-efficient training
- Fundamental understanding
- Bias, Privacy and Fairness

Fundamental AI Lab

Video and Temporal Learning

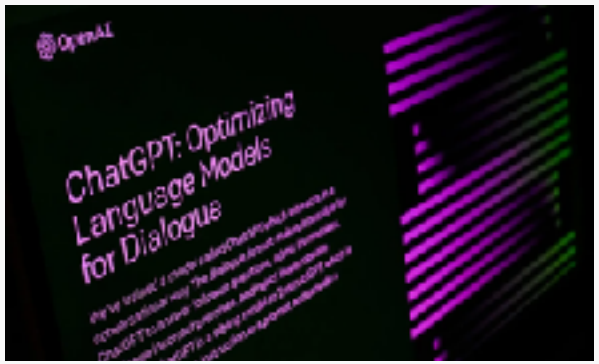
- Learning image models from video signals
- Cross-modal and multimodal learning frameworks
- Better video architectures and tasks

Large Language Models

- Instruction-Tuning
- Reasoning, planning
- Parameter-Efficient Finetuning
- Bias, Fairness

Nuremburg image via Unsplash

Currently we have ~three types of key Foundation Models



(Multimodal) Large Language Models

writing, coding, assistive tech, etc.

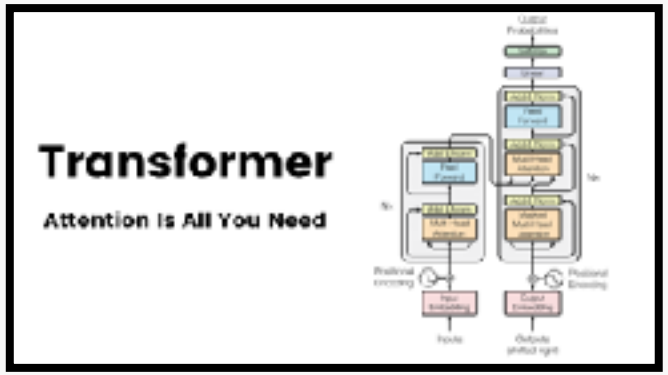
Text-conditional Generative Models

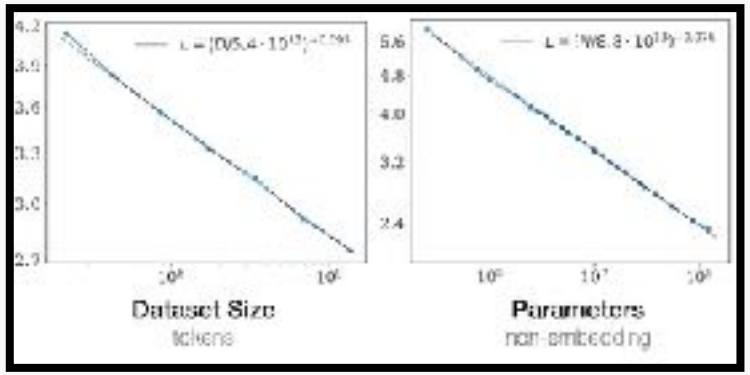
Image/video generation, editing, control

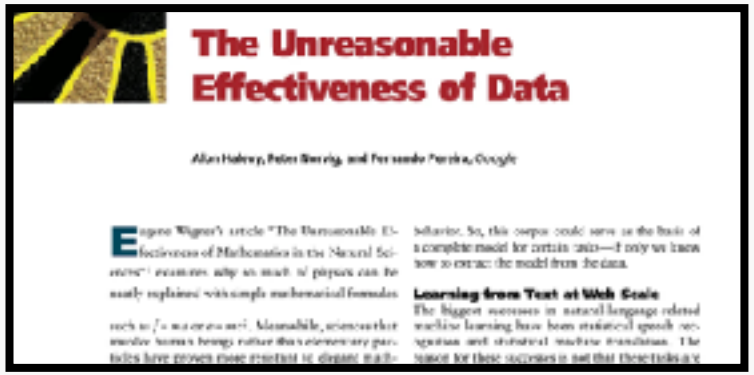
General Image Understanding Models

Transfer learning, vision component in MLLMs etc.

What happened in the last ~2 years?



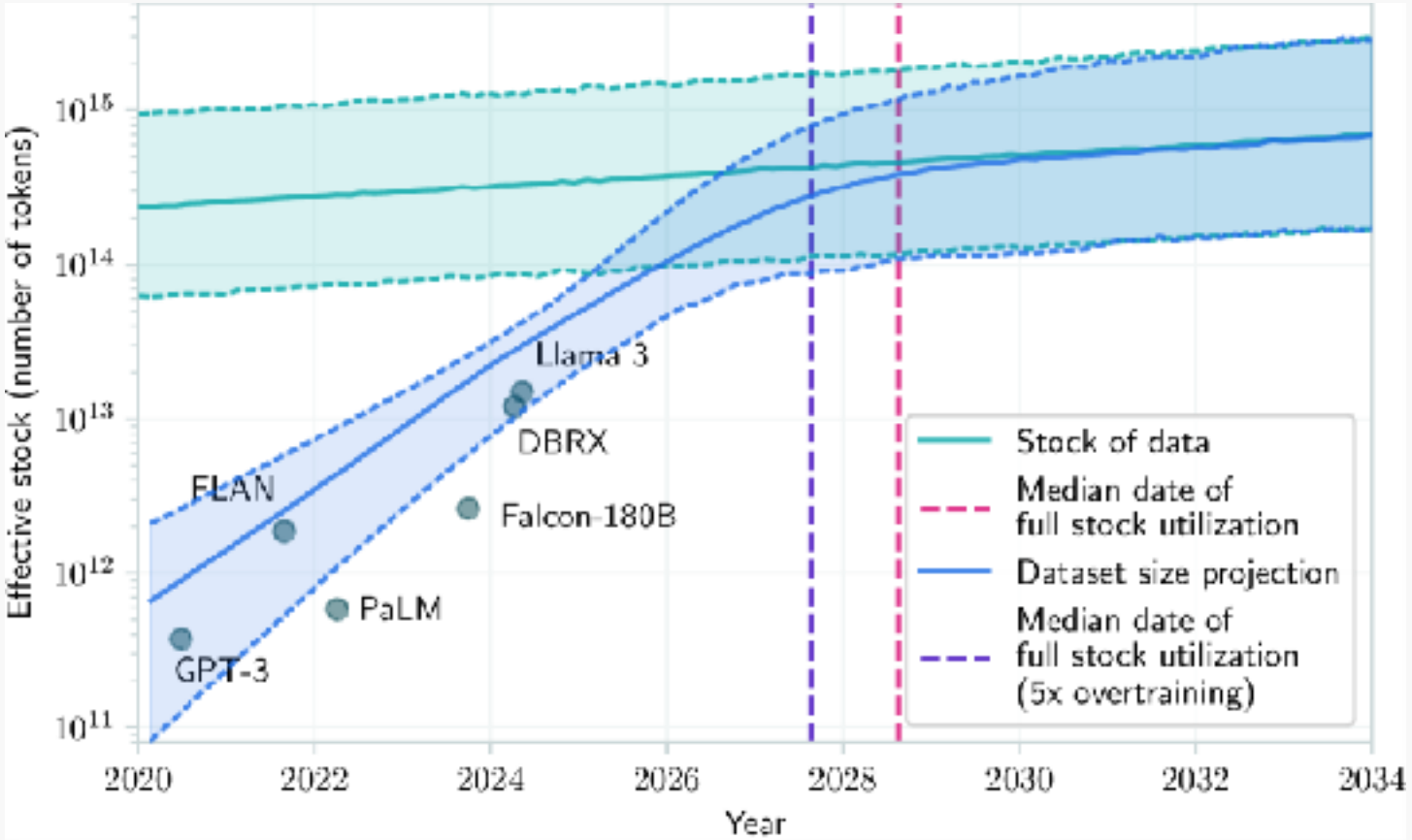




Flexible "model" + Predictable behavior + More (unsupervised) data

Attention Is All You Need. Vaswani et al. NeurIPS 2017
Training Compute-Optimal Large Language Models. Hoffman et al. NeurIPS 2022
The Unreasonable Effectiveness of Data. Halevy et al. 2009

Current speed of progress requires exponential increase in dataset size



... and for text-data, we might run out soon

"end of pretraining"

Will we run out of data? Limits of LLM scaling based on human-generated data. Villalobos et al. ICML 2024

Effect of compute in video generation

Self-supervised Learning

Manual data annotations for supervised learning are limiting.

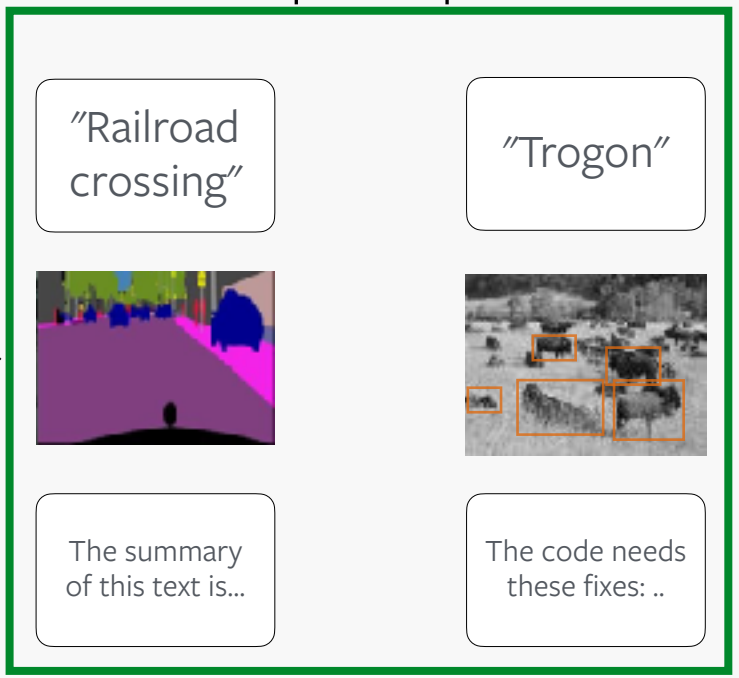
Data is cheap & ubiquitous

data

Supervised Learning

signal

But annotations are expensive and often require experts



ImageNet: A Large-Scale Hierarchical Image Database. Dong et al. CVPR 2009
The Cityscapes Dataset for Semantic Urban Scene Understanding. Cordts et al. CVPR 2016
Scene parsing through ADE20K dataset. Zhou et al. CVPR 2017.

Self-supervised learning solves the problem of annotations.

data

signal

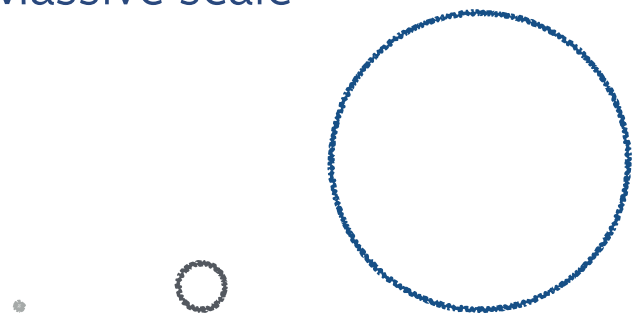
Self-Supervised Learning

Self-supervision

Extract a supervisory signal from the raw data alone

Self-supervised Learning has benefits besides scalability

Massive scale



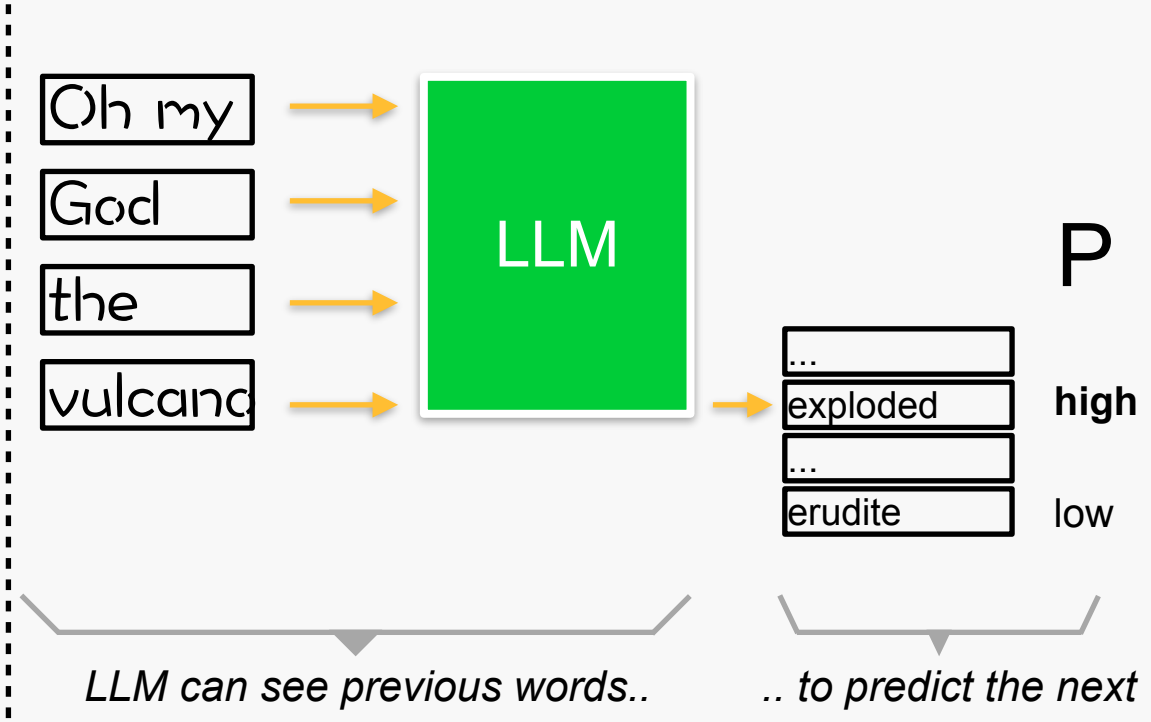
sup. << weak sup. << raw

No cost of relabelling

No language bias

Fundamentals

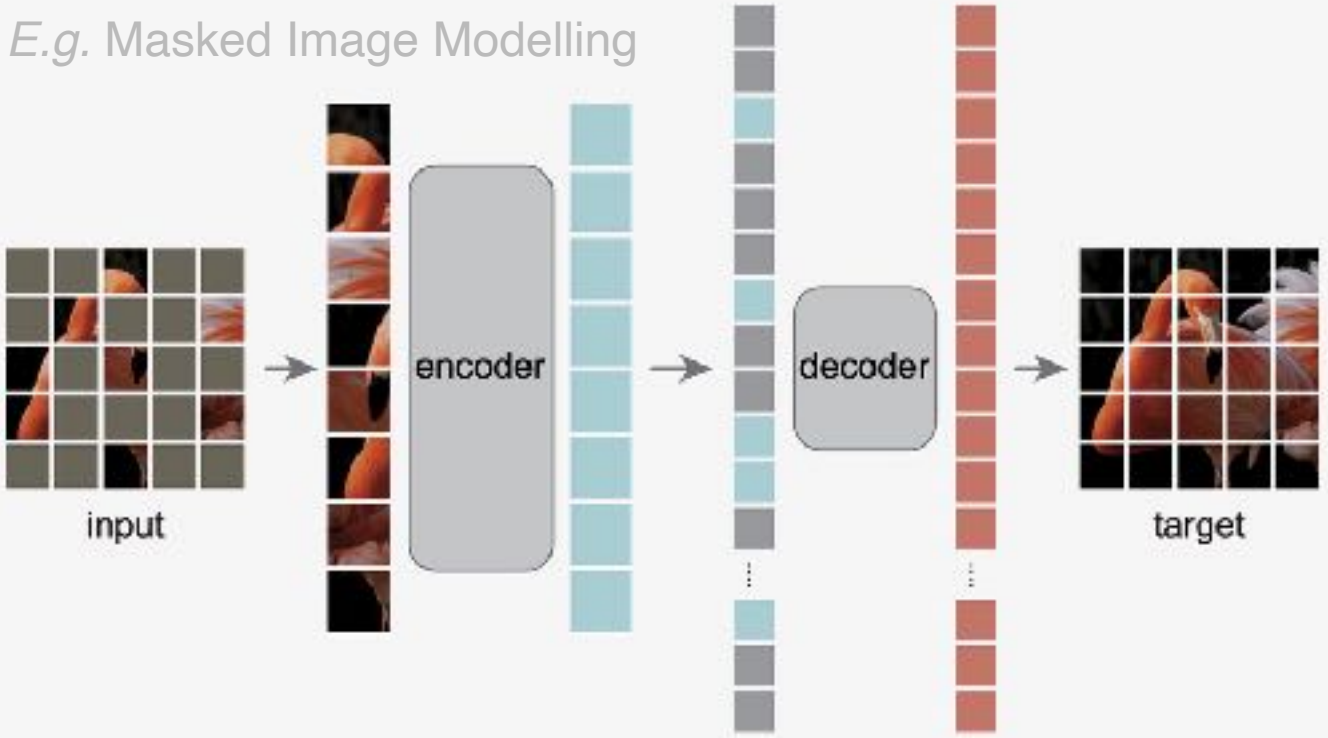
How does self-supervised learning work?



adapted from https://what-if.xkcd.com/34/

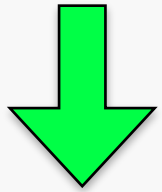
How does self-supervised learning work for vision?

E.g. Masked Image Modelling



But:

- Images don't have natural word "units"
- Images don't have fixed a vocabulary
- Image generation ≠ Image understanding



Diverse Self-Supervision Methods in Computer Vision

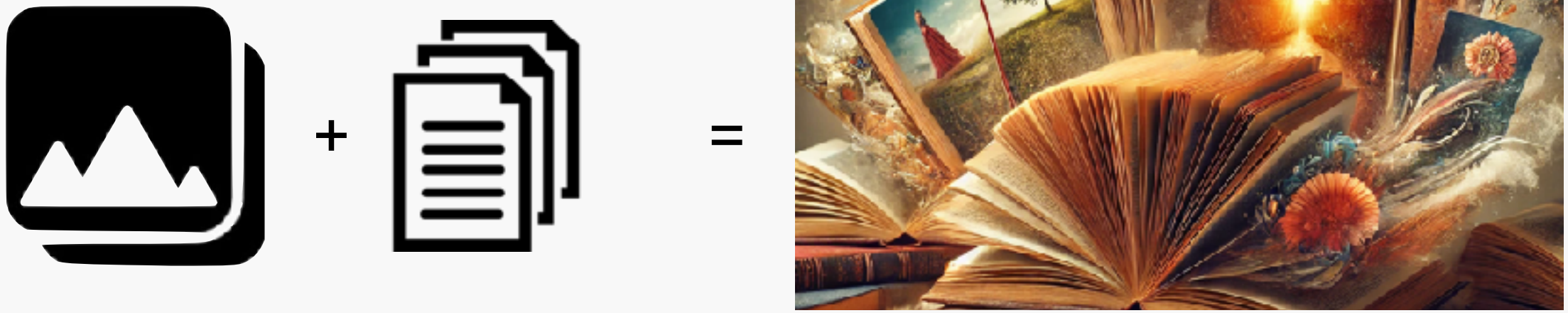
Masked Autoencoders Are Scalable Vision Learners. He et al. CVPR 2022

Today we'll talk about two research topics

Improving vision neural networks

Understanding visio-lingual models

Today we'll talk about two research topics



Understanding visio-lingual models

Multi-modal Learning

+ captions/ thoughts?

What makes multimodal learning interesting?

Text is like an "augmentation" / broader description

The man at bat readies to swing at the pitch while the umpire looks on.

The meaning depends on both modalities (rarer)

What makes multimodal learning interesting?

Interfaces: e.g. for visually impaired people

Ground common sense knowledge in real-world

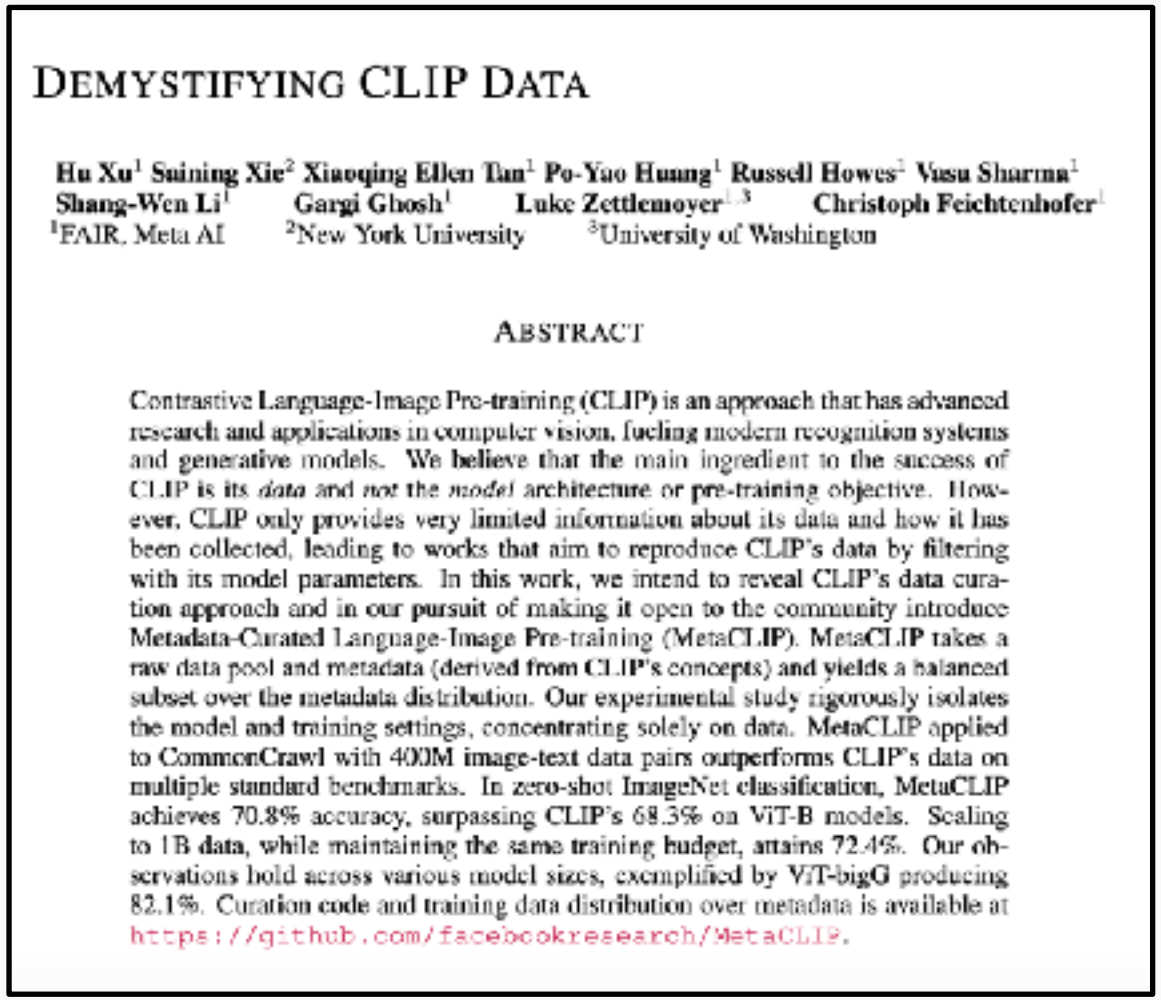
Visual Alignment in Text-Only LLMs: New Frontiers in Data Efficiency

Jona Ruthardt, Gertjan J. Burghouts, Serge Belongie, Yuki M. Asano

arxiv

Fundamental AI Lab UTN TNO

Motivation



DEMYSTIFYING CLIP DATA

Hu Xu[1] Saining Xie[2] Xiaoqing Ellen Tan[1] Po-Yao Huang[1] Russell Howes[1] Vasu Sharma[1]
Shang-Wen Li[1] Gargi Ghosh[1] Luke Zettlemoyer[1,3] Christoph Feichtenhofer[1]
[1]FAIR, Meta AI [2]New York University [3]University of Washington

ABSTRACT

Contrastive Language-Image Pre-training (CLIP) is an approach that has advanced research and applications in computer vision, fueling modern recognition systems and generative models. We believe that the main ingredient to the success of CLIP is its *data* and *not* the *model* architecture or pre-training objective. However, CLIP only provides very limited information about its data and how it has been collected, leading to works that aim to reproduce CLIP's data by filtering with its model parameters. In this work, we intend to reveal CLIP's data curation approach and in our pursuit of making it open to the community introduce Metadata-Curated Language-Image Pre-training (MetaCLIP). MetaCLIP takes a raw data pool and metadata (derived from CLIP's concepts) and yields a balanced subset over the metadata distribution. Our experimental study rigorously isolates the model and training settings, concentrating solely on data. MetaCLIP applied to CommonCrawl with 400M image-text data pairs outperforms CLIP's data on multiple standard benchmarks. In zero-shot ImageNet classification, MetaCLIP achieves 70.8% accuracy, surpassing CLIP's 68.3% on ViT-B models. Scaling to 1B data, while maintaining the same training budget, attains 72.4%. Our observations hold across various model sizes, exemplified by ViT-bigG producing 82.1%. Curation code and training data distribution over metadata is available at https://github.com/facebookresearch/MetaCLIP.

We've heard that "CLIP generalises because of language"

BUT:

*Does CLIP **actually** generalise?*

Appendix p.14, Table 11:

	ImageNet	Food-101	CIFAR10	CIFAR100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	MNIST	FER-2013	STL-10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMemes	SST2
MetaCLIP (400M) ViT-L	76.2	90.7	95.5	77.4	75.9	70.5	84.7	40.4	62.0	93.7	94.4	76.4	61.7	46.5	99.3	59.7	71.9	47.5	29.9	30.9	70.1	75.5	57.1	35.1	56.6	65.6
# of cls. w/ nor-zero counts	703/998	52/101	10/10	93/100	1/200	193/397	0/196	8/100	40/47	15/37	86/102	61/102	10/10	12/12	10/10	2/10	32/45	1/43	0/4	190/211	1/2	5/101	122/700	8/8	1/2	2/2

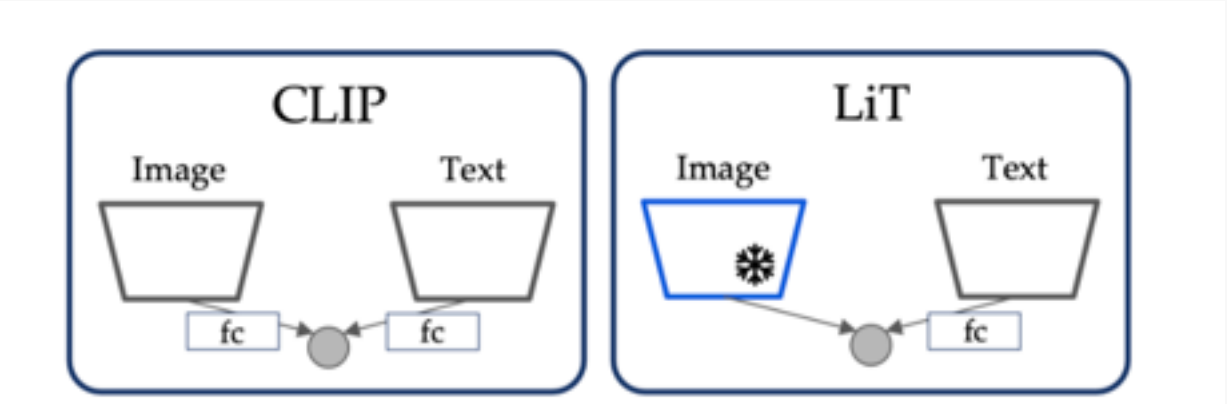
Table 11: Measuring task-alignment. First row: MetaCLIP (400M) ViT-L/14 accuracy, second row: number of classes matched in metadata

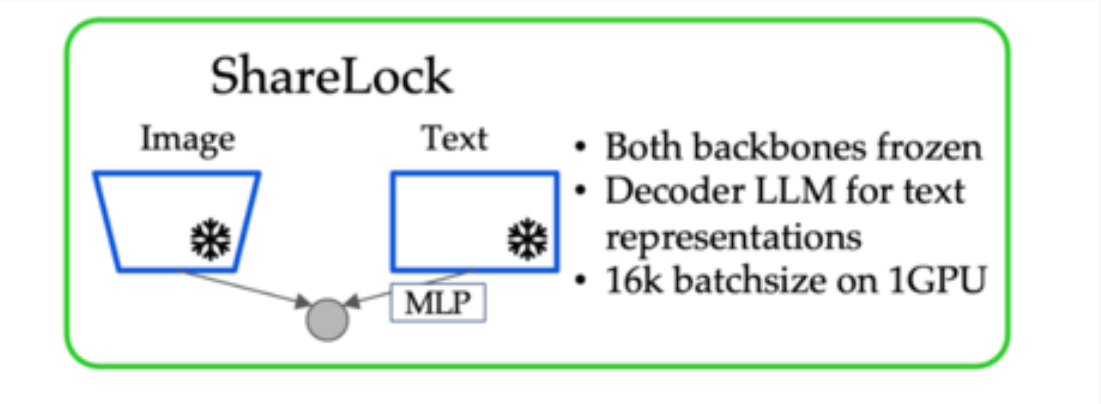
"Interestingly, there seems to be a correlation with the accuracy and the number of classes matched in the metadata."

CLIP, for the most part, is evaluated within-domain (it's just a big domain)

But surely language features, e.g. from pretrained models should help generalise?

Setup of new benchmark & method: **Share**d Vision-Language-**Lock**ed Tuning





Data: supervised classification datasets, split into mutually exclusive categories.

--> Train with "a photo of a {class name}"

[AWA2, CUB, FGVCAircraft, and ImageNet+]

Text features are obtained from the last layer's feature of the last token

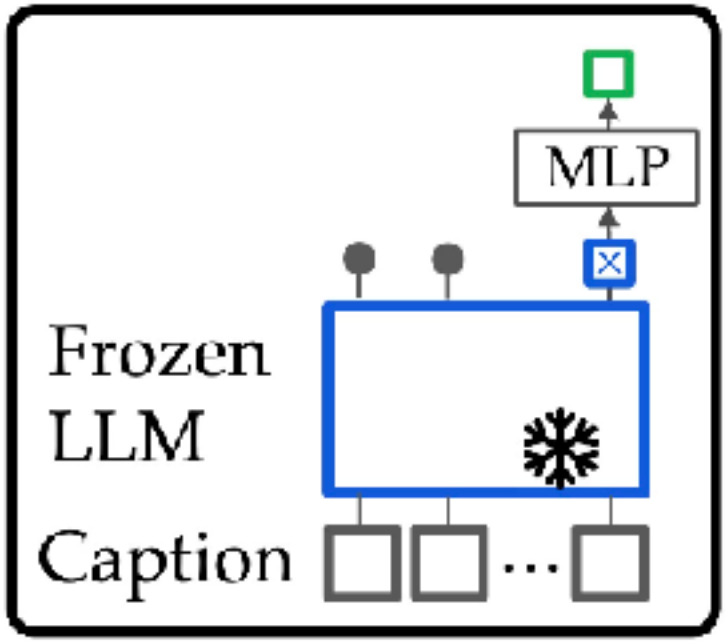


Figure 3. **Text features.** We obtain the final text features by processing the last caption token with an MLP. This allows avoiding expensive forward passes of the LLM during training by precomputing and storing the features (×).

Decoder representations are actually really good.

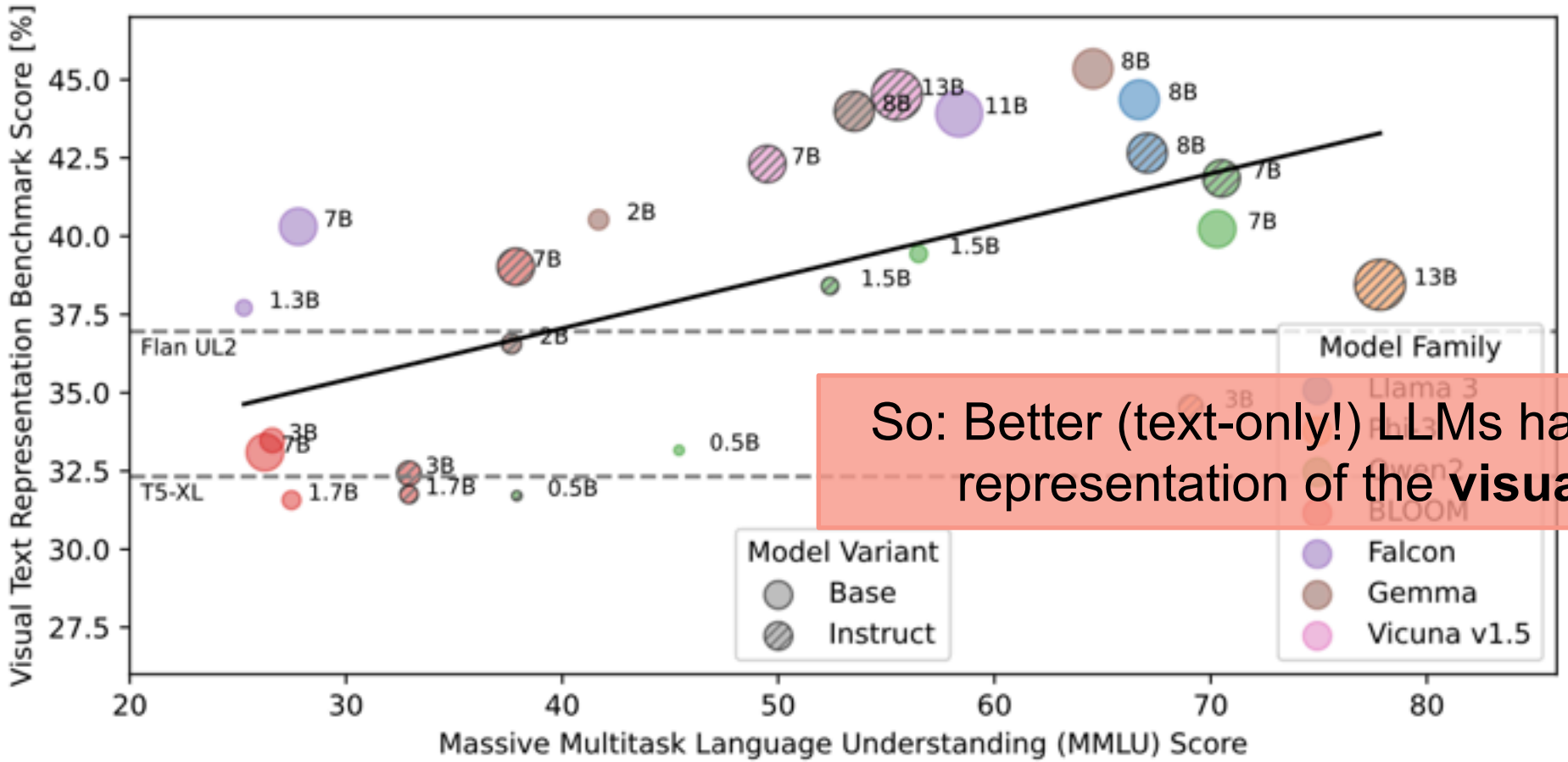
Type	Language Model	Class Names
Enc.	BERT-Large [9]	18.3
	T5-XL [47]	33.6
	Flan-UL2 [55]	37.0
	SentenceT5-XXL [39]	39.5
Dec.	Gemma 7B [16]	39.7
	Llama-3 8B [11]	<u>40.2</u>
	NV-Embed [31]	**40.5**

What people previously used

New billion-scale LLMs

LLMs contain knowledge that helps visual zero-shot classification

Moreover: LLM's ShareLock performance correlates with (text-only) MMLU evaluation!



And what if we train with actual image-caption datasets?

Fundamental AI Lab

UTN

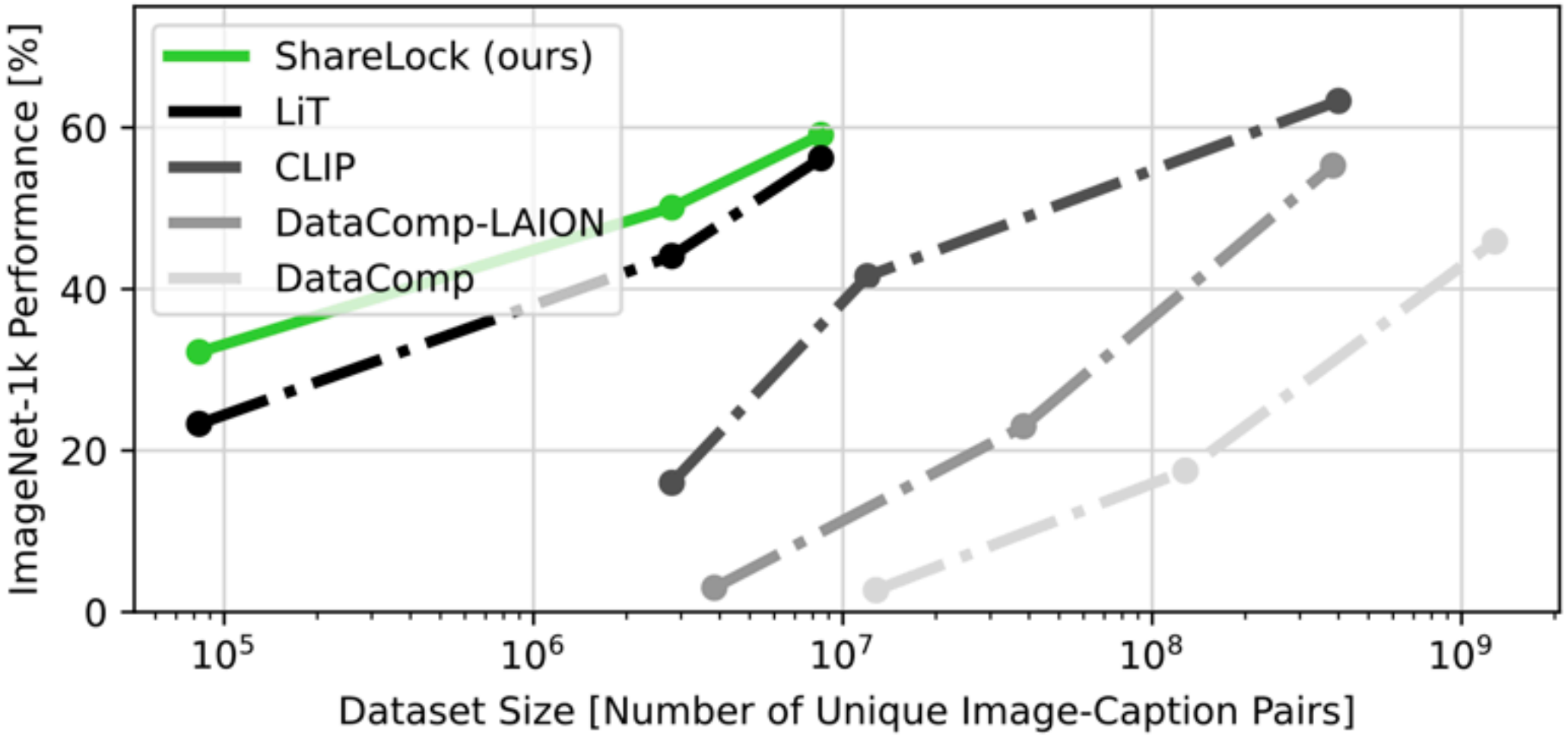
Strong SotA for datasets 100k-12M

Model	Dataset	[Size]	IN-1k	IN-V2	IN-R	IN-A	IN Sketch	ObjectNet	Avg
LiT	COCO	83k	23.3	20.8	34.4	21.1	18.4	29.2	24.5
ASIF	COCO	83k	9.4	8.7	14.4	8.8	6.9	16.1	10.7
ShareLock	COCO	83k	**32.2**	**28.6**	**36.6**	**22.8**	**22.4**	**30.4**	**28.8**
LiT	CC3M Subset	563k	41.7	37.5	59.2	44.4	32.4	40.7	42.6
ASIF	CC3M Subset	563k	21.6	20.5	27.7	24.4	14.9	21.5	21.8
ShareLock	CC3M Subset	563k	**50.5**	**45.8**	**60.5**	**47.0**	**36.9**	**41.1**	**47.0**
CLIP [12]	CC3M	2.8M	16.0	13.2	17.6	3.6	6.4	8.2	10.8
SLIP [38]	CC3M	2.8M	23.5	20.2	26.8	6.8	12.1	14.3	17.3
LaCLIP [12]	CC3M	2.8M	21.3	18.6	23.5	5.0	10.6	10.2	14.9
LiT	CC3M	2.8M	44.1	39.3	62.7	45.6	34.8	**43.3**	45.0
ShareLock	CC3M	2.8M	**52.1**	**47.1**	**64.1**	**50.9**	**39.0**	43.1	**49.4**
DataComp [14]	CPool-S	3.84M	3.0	2.7	4.4	1.5	1.3	3.7	2.8
CLIP [12]	CC12M	12M	41.6	35.4	52.6	10.7	28.8	24.0	32.2
SLIP [38]	CC12M	12M	41.7	35.9	55.2	13.8	30.7	29.3	34.4
LaCLIP [12]	CC12M	12M	49.0	43.3	63.8	14.7	39.4	28.1	39.7
LiT	CC12M	8.5M	56.2	49.9	**70.3**	52.8	43.9	**47.8**	53.5
ShareLock	CC12M	8.5M	**59.1**	**53.2**	68.8	**53.4**	**44.5**	46.7	**54.3**
DataComp [14]	CPool-M	38.4M	23.0	18.9	28.0	4.3	15.1	17.7	17.8
DataComp [14]	CPool-L	384M	55.3	47.9	65.0	20.2	43.2	46.5	46.3
CLIP [46]	Proprietary	400M	68.4	61.8	77.6	50.1	48.2	55.4	60.2

Thanks to the frozen LLM, we excel in multi-lingual evaluations

Model	Dataset	[Size]	EN	CN	JP	IT
LiT	COCO	83k	23.3	0.2	0.2	4.5
ShareLock	COCO	83k	**32.2**	**11.3**	**1.9**	**15.6**
CLIP [12]	CC12M	12M	41.6	0.1	0.1	7.9
LiT	CC12M	8.5M	56.2	0.2	0.2	11.6
ShareLock	CC12M	8.5M	**59.1**	**25.1**	**1.9**	**35.8**
DataComp [14]	CPool-M	38.4M	23.0	0.2	0.3	4.7
DataComp [14]	CPool-L	384M	55.3	0.7	1.5	15.2
CLIP [46]	Proprietary	400M	68.4	1.4	4.1	21.7

Scaling curves



TLDR:

ShareLock is an ultra-lightweight vision-language model that

TLDR:

ShareLock is an ultra-lightweight vision-language model that

achieves competitive multimodal performance by leveraging frozen features from state-of-the-art unimodal models.

TLDR:

ShareLock is an ultra-lightweight vision-language model that

achieves competitive multimodal performance by leveraging frozen features from state-of-the-art unimodal models.

Trained on just 563k image-caption pairs, it achieves **51% zero-shot accuracy on ImageNet** and outperforms existing methods

TLDR:

ShareLock is an ultra-lightweight vision-language model that

achieves competitive multimodal performance by leveraging frozen features from state-of-the-art unimodal models.

Trained on just 563k image-caption pairs, it achieves **51% zero-shot accuracy on ImageNet** and outperforms existing methods

in low-data regimes, with a total training time of <**15 GPU hours**.

PIN: Positional Insert unlocks object localisation abilities in VLMs.
Michael Dorkenwald, Nimrod Barazani, Cees G. M. Snoek, and Yuki M Asano.
CVPR, 2024

Fundamental AI Lab

Vision-Language Models are great at many things, but not localisation.

Prompt 1: Provide a bounding box around the cat

Prompt 2: Localise the cat in the image

Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Our solution: *unlock* localisation abilities in frozen VLMs

VLMs are bad at localising and cannot handle the bbox detection task

But (somewhat noisy) localisation does emerge in some VLMs

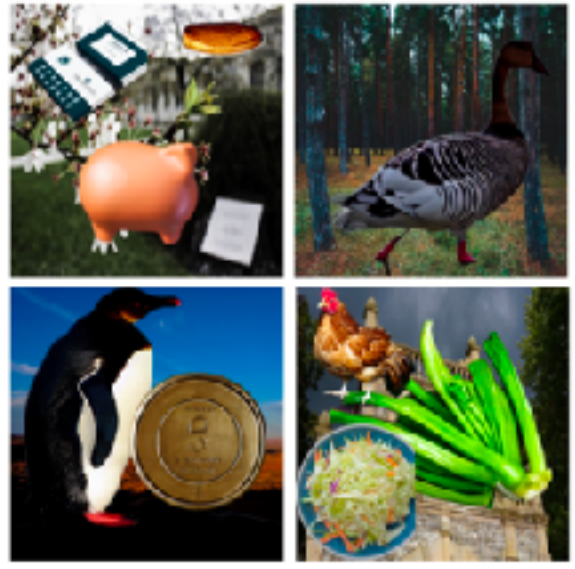
Try to **unlock** the forgotten localisation abilities in **frozen VLMs**

Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Our approach

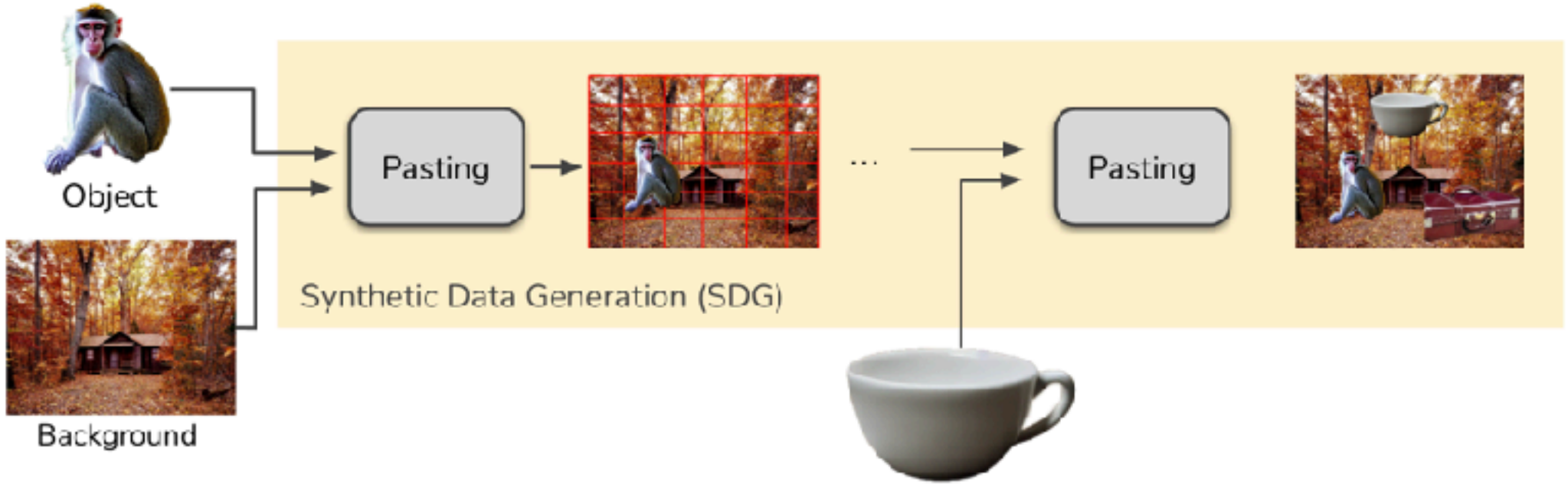
frozen VLM, e.g. Flamingo

Positional Insert (PIN) module



Synthetic, unlabeled data

The data



Zhao et al. X-Paste: Revisiting Scalable Copy-Paste for Instance Segmentation using CLIP and StableDiffusion. ICML 2023

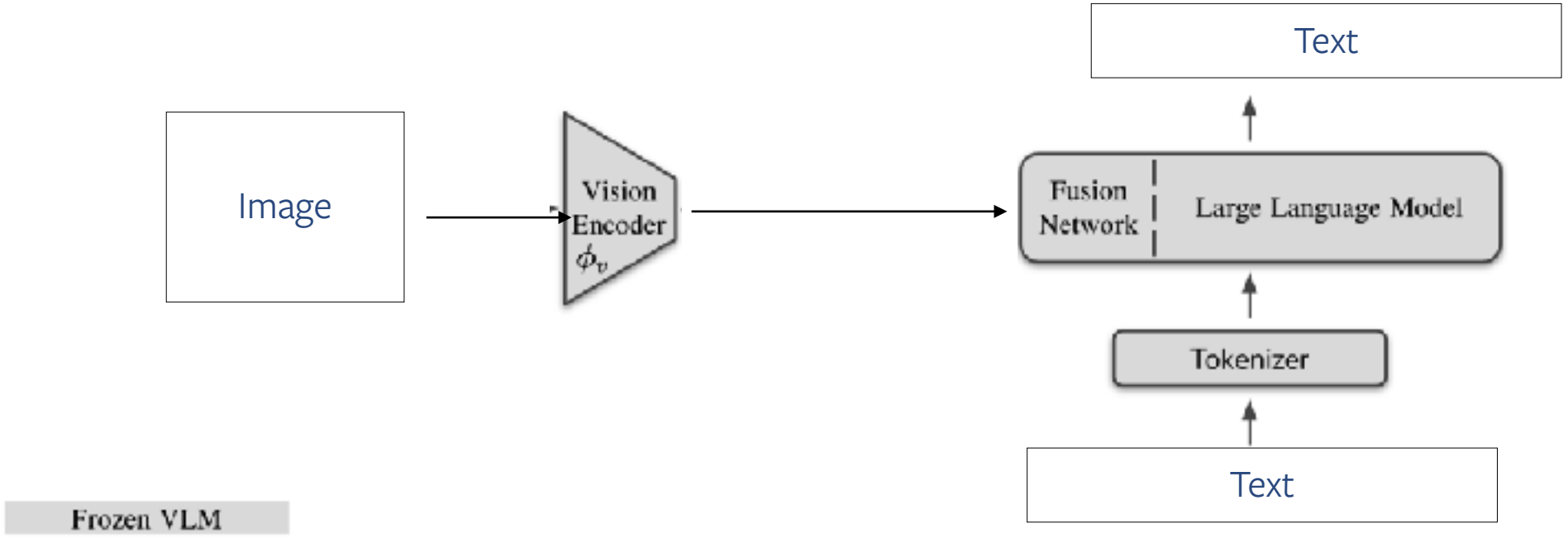
Example generated data



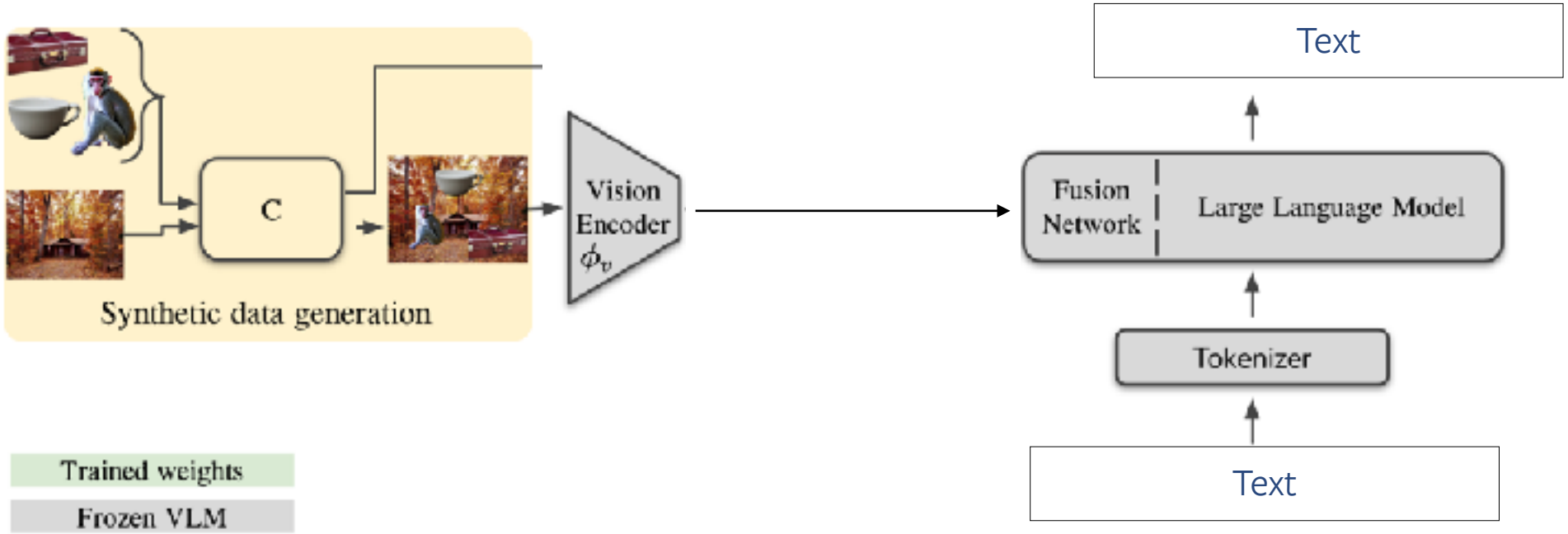


Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Default Flamingo

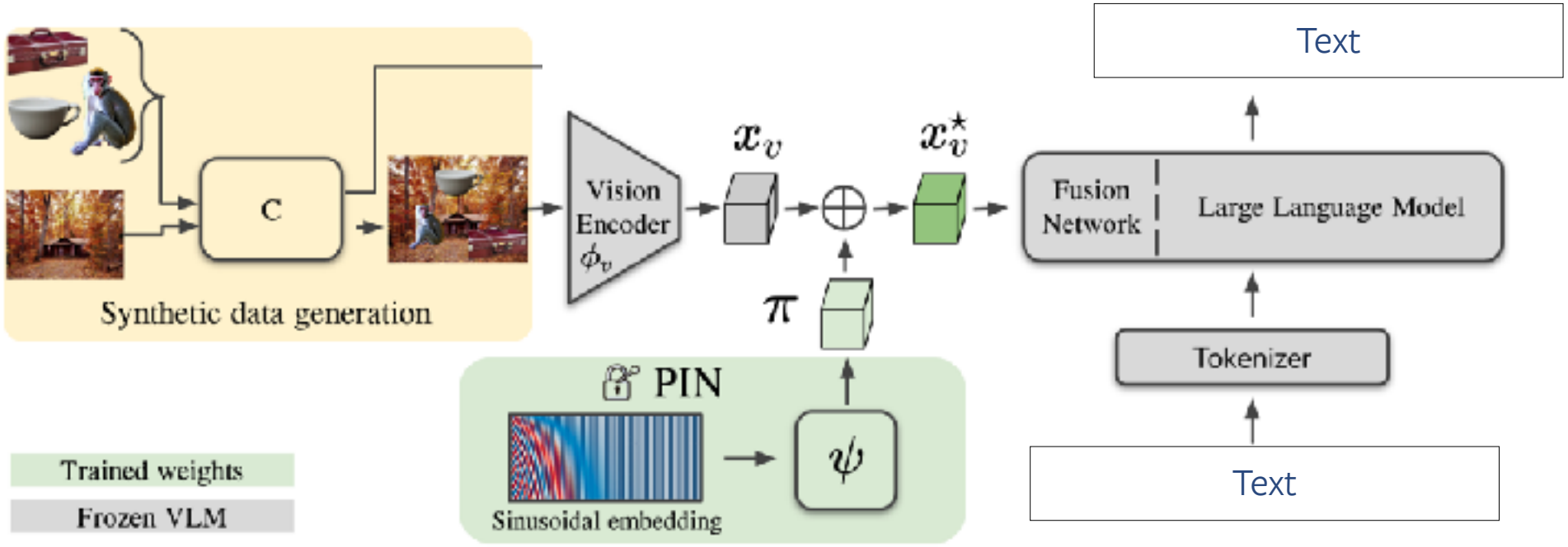


Our method 1: feed the frozen vision encoder synthetic data



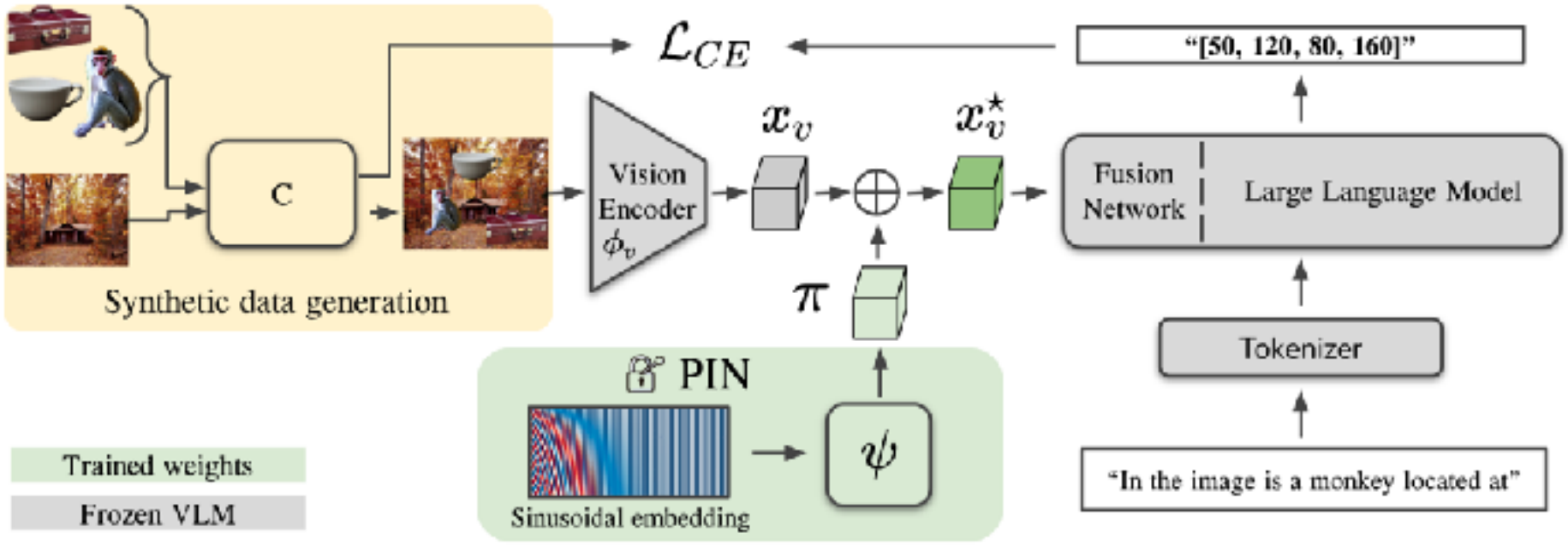
Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Our method 2: provide VLM spatial learning capacity



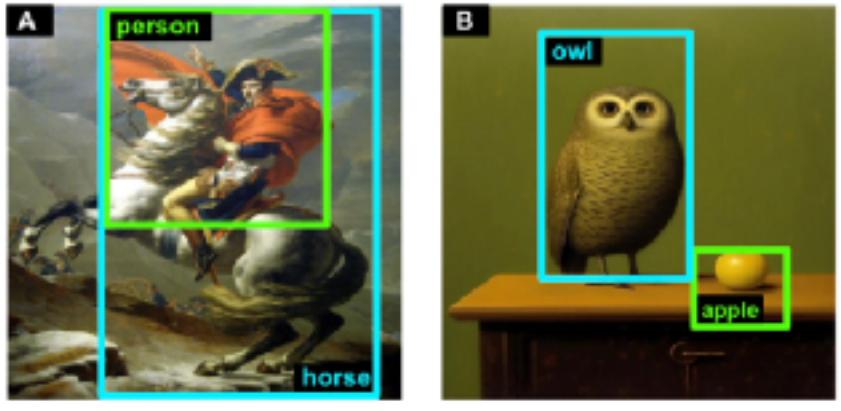
Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Our method 3: train using pasted obj locations via next-word prediction



Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Results

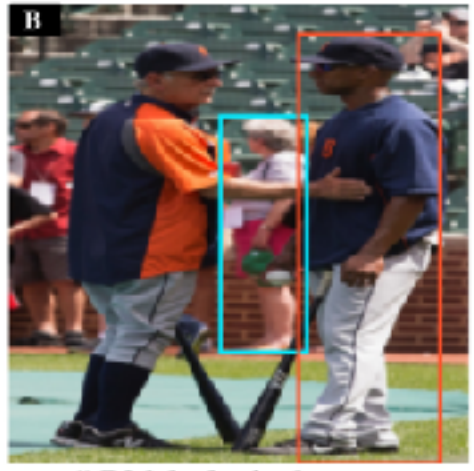


We beat common PEFT methods

	Method	PVOC$_{\leq 3 \text{ Objects}}$ mIoU	PVOC$_{\leq 3 \text{ Objects}}$ mIoU$_M$	PVOC$_{\leq 3 \text{ Objects}}$ mIoU$_L$	COCO$_{\leq 3 \text{ Objects}}$ mIoU	COCO$_{\leq 3 \text{ Objects}}$ mIoU$_M$	COCO$_{\leq 3 \text{ Objects}}$ mIoU$_L$	LVIS$_{\leq 3 \text{ Objects}}$ mIoU	LVIS$_{\leq 3 \text{ Objects}}$ mIoU$_M$	LVIS$_{\leq 3 \text{ Objects}}$ mIoU$_L$
OpenFlamingo [5]	*Baselines*									
	raw	0	0	0	0	0	0	0	0	0
	random	0.22±0.04	0.10±0.02	0.33±0.06	0.12±0.04	0.07±0.02	0.22±0.08	0.07±0.03	0.06±0.02	0.18±0.09
	2 context	0.19±0.11	0.08±0.05	0.30±0.18	0.10±0.08	0.06±0.04	0.18±0.16	0.04±0.06	0.03±0.04	0.10±0.15
	5 context	0.19±0.09	0.07±0.04	0.31±0.15	0.10±0.08	0.06±0.04	0.20±0.16	0.06±0.05	0.04±0.03	0.17±0.13
	10 context	0.20±0.11	0.06±0.03	0.32±0.18	0.09±0.07	0.05±0.04	0.17±0.14	0.05±0.05	0.03±0.03	0.15±0.14
	PEFT									
	CoOp on LLM	0.28	0.11	0.43	0.22	0.10	0.39	0.13	0.07	0.40
	VPT on F	0.34	0.16	0.51	0.26	0.15	0.47	0.19	0.14	0.48
	VPT on ϕ_V	0.42	0.21	0.61	0.33	0.22	0.57	0.23	0.19	0.56
	LoRA on ϕ_V	0.44	0.26	**0.62**	0.33	0.23	0.58	0.23	0.19	0.55
	PIN (ours)	**0.45**	**0.27**	**0.62**	**0.35**	**0.26**	**0.59**	**0.26**	**0.24**	**0.61**
BLIP-2 [32]	*PEFT*									
	VPT on F	0.33	0.12	0.51	0.27	0.12	0.50	0.18	0.11	0.47
	VPT on ϕ_V	0.32	0.12	0.50	0.26	0.11	0.48	0.17	0.10	0.46
	PIN (ours)	**0.44**	**0.24**	**0.63**	**0.34**	**0.22**	**0.60**	**0.26**	**0.23**	**0.60**

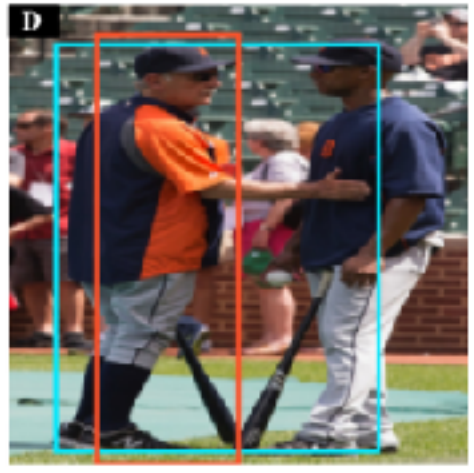
Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

"Left black shirt"



"Old lady in between the players"

"A guy in red on left"



"Guy in orange"

"Right player"

"Top left apron strings"

"Pizza squares left"

"Pizza right front piece in middle"

"A man black"

"A right person"

Predictions Ground Truth

Today we'll talk about two research topics. Topic 2:

Improving vision neural networks

NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency.

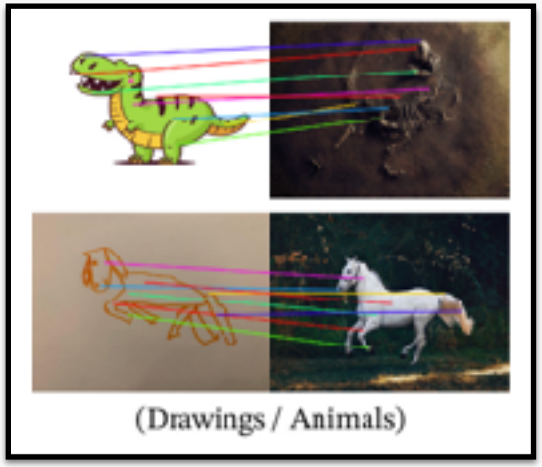
Valentinos Pariza, Mohammadreza Salehi, Gertjan Burghouts, Francesco Locatello, Yuki M. Asano.
arxiv 2024

Fundamental AI Lab

UTN

How semantic are patch representations?

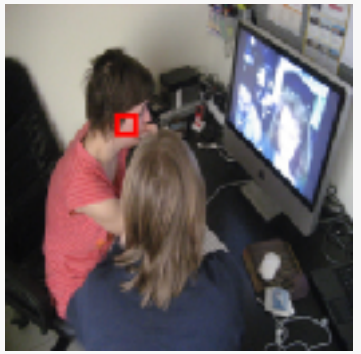
Qualitative results in DINOv2



But often...

How it should be

Which patch from the whole dataset is the closest?



How it is

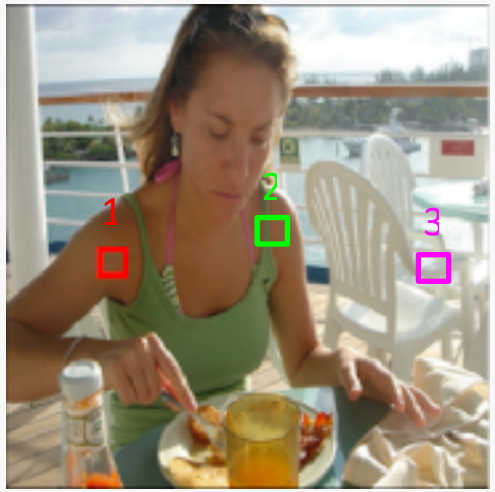
with SoTA DINOv2-R model

Idea of Patch Nearest Neighbor Consistency: intuitive to us

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

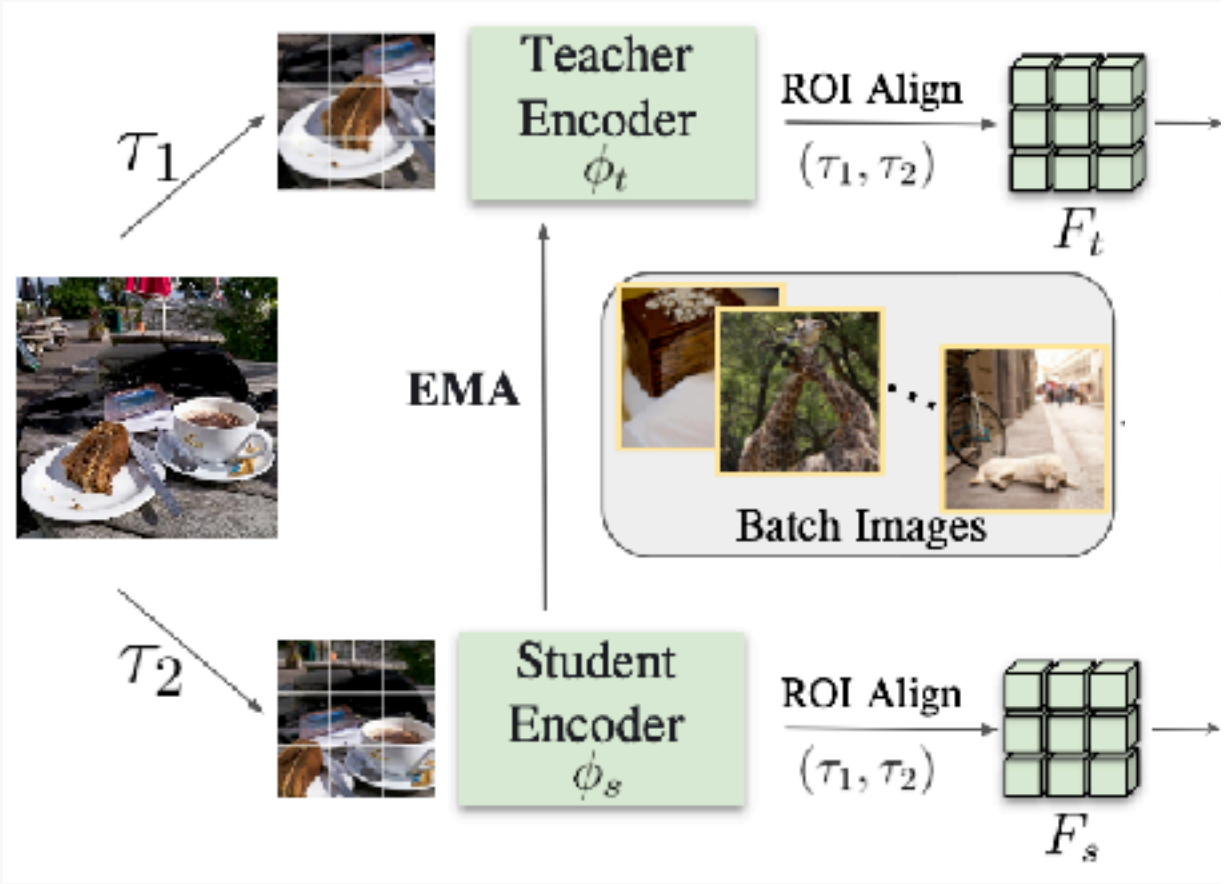
(1) All Right Shoulder Patches, (2) All Left Shoulder Patches, (...) (3) Everything Else

Query Patch



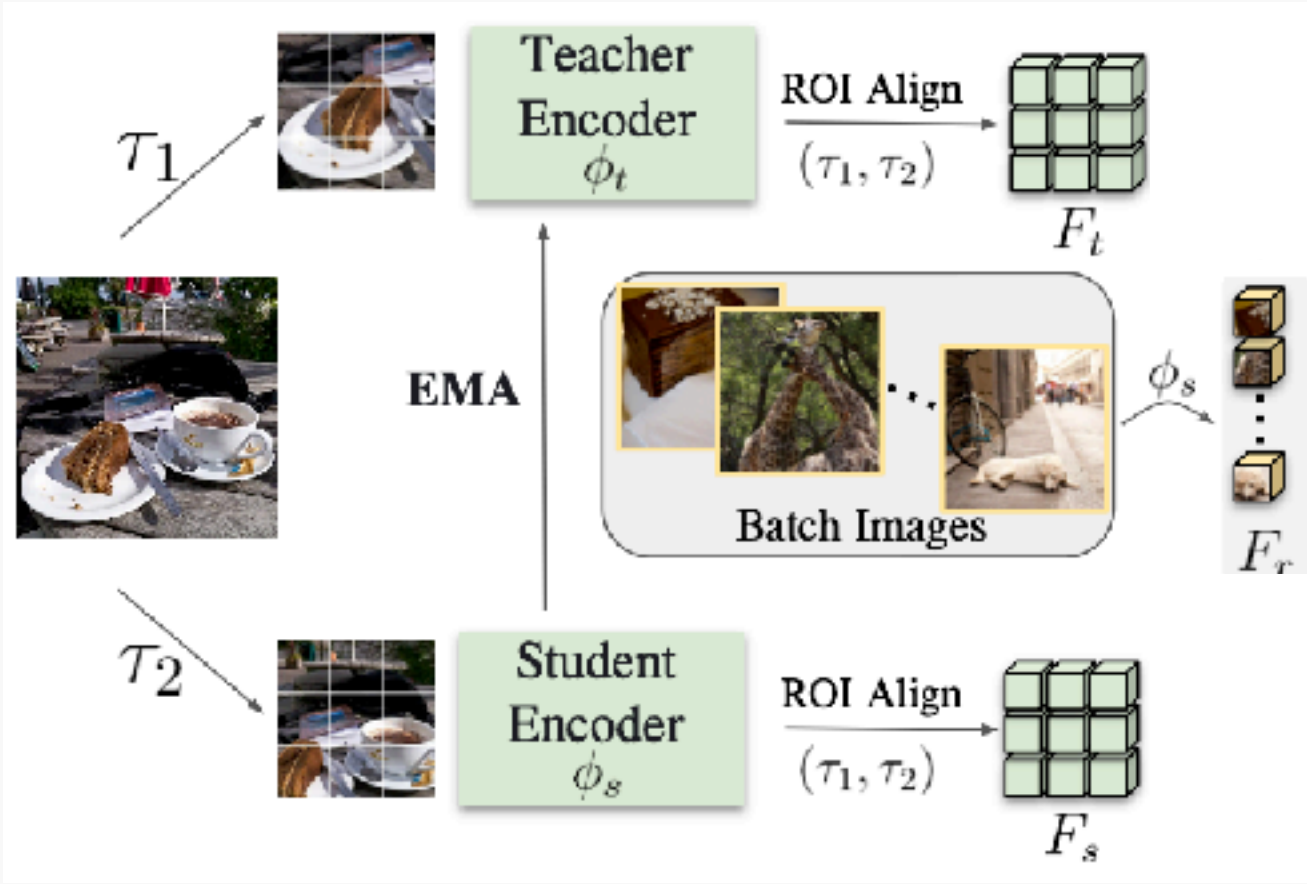
Example Patches

PaNeCo: Patch Nearest neighbor Consistency



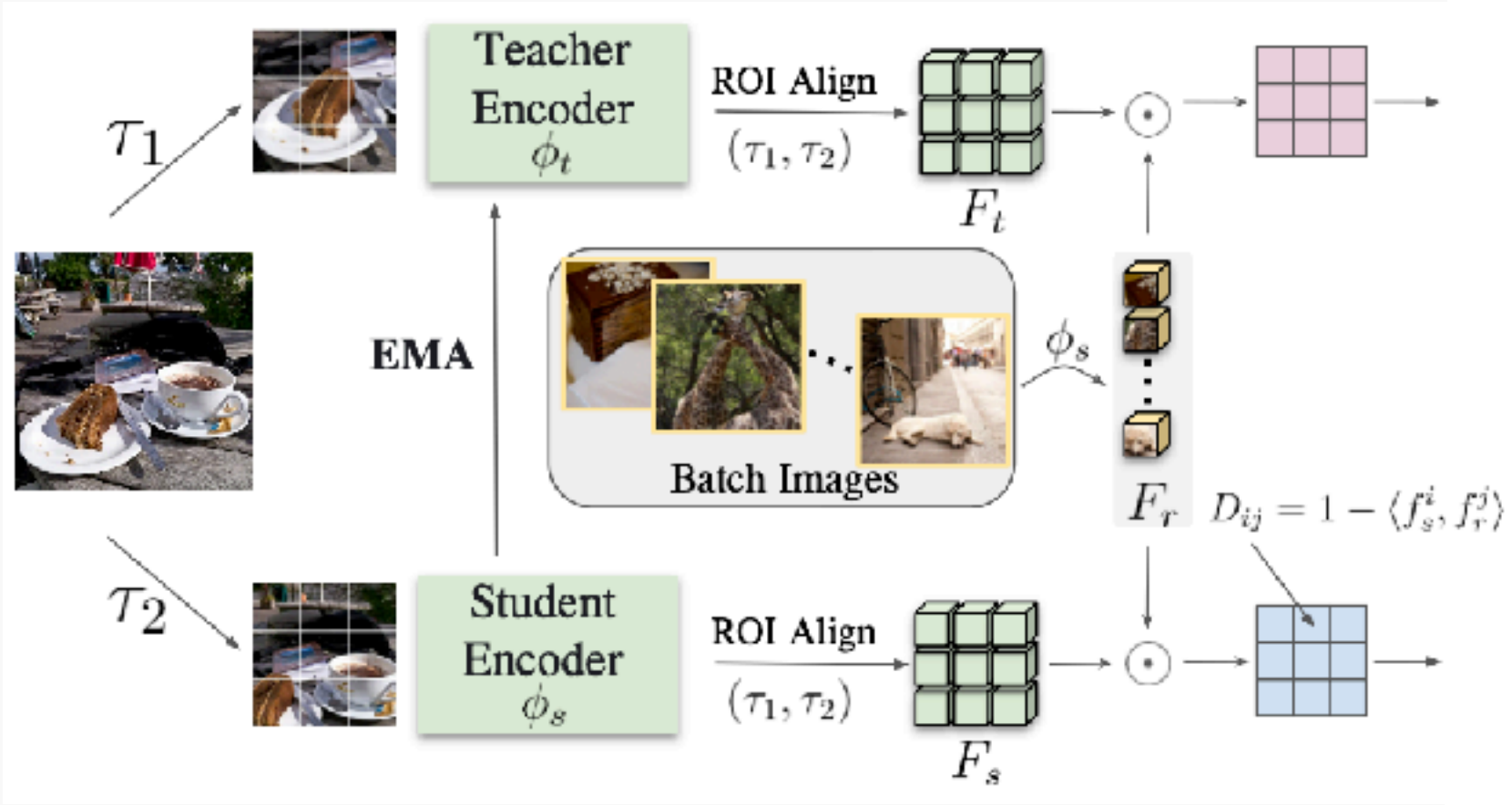
NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency. Pariza, Salehi, Burghouts, Locatello, Asano. arxiv 2024

PaNeCo: Patch Nearest neighbor Consistency



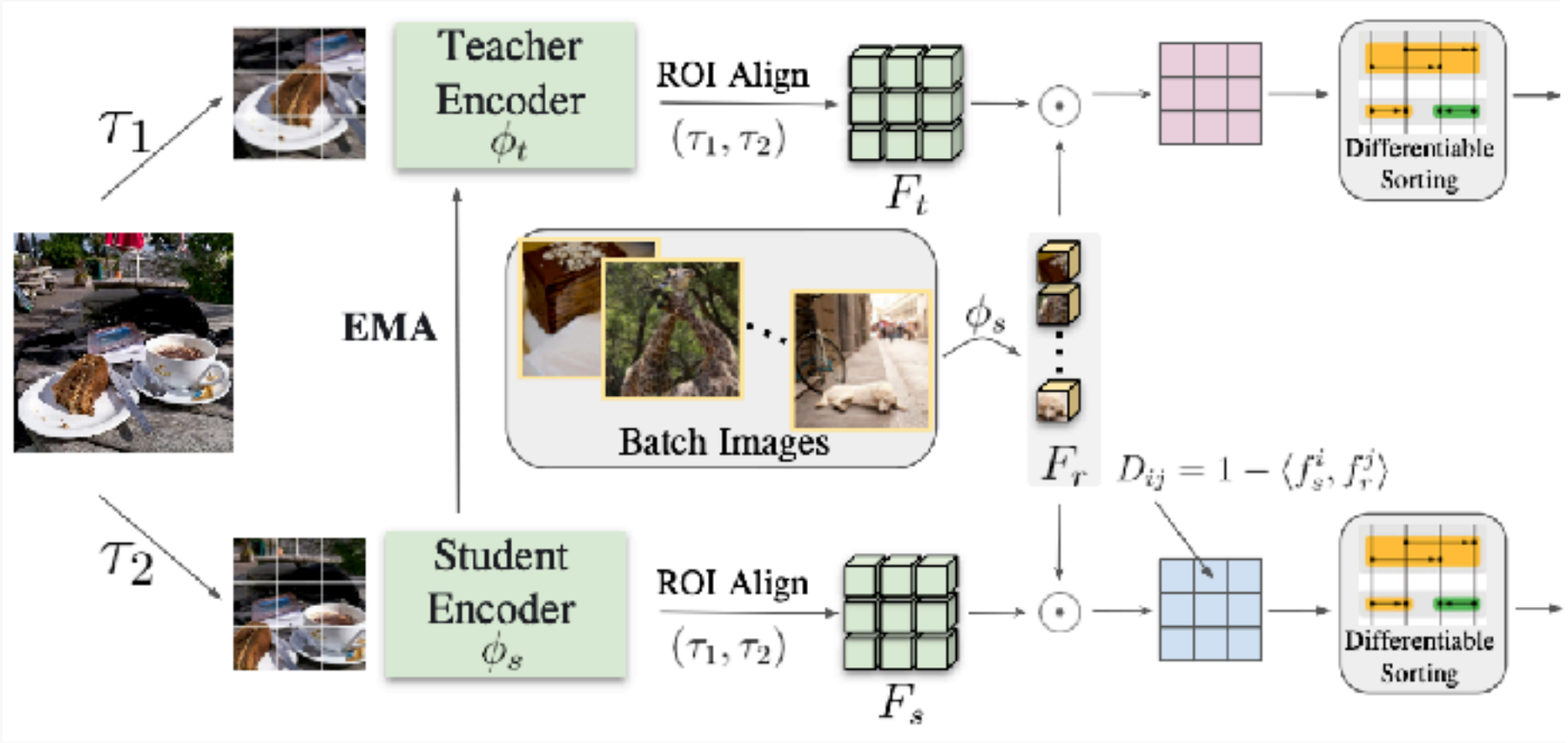
NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency. Pariza, Salehi, Burghouts, Locatello, Asano. arxiv 2024

PaNeCo: Patch Nearest neighbor Consistency



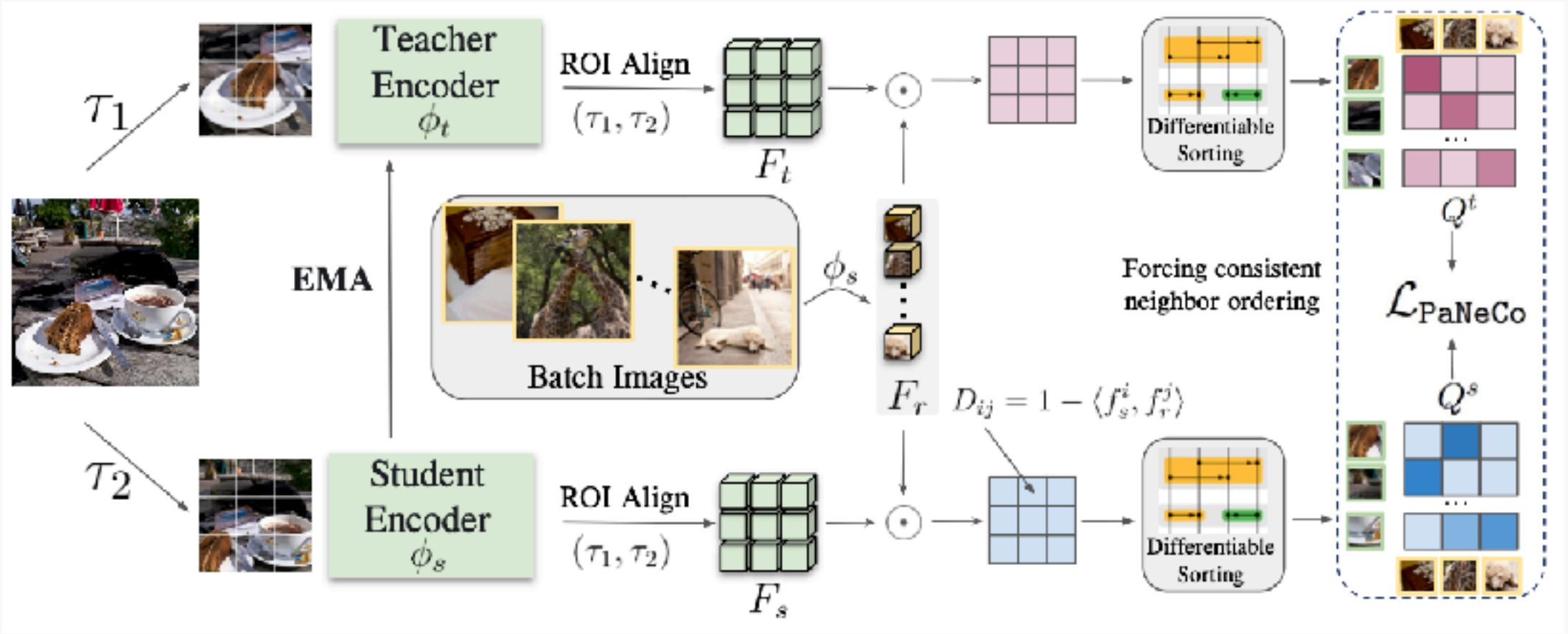
NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency. Pariza, Salehi, Burghouts, Locatello, Asano. arxiv 2024

PaNeCo: Patch Nearest neighbor Consistency



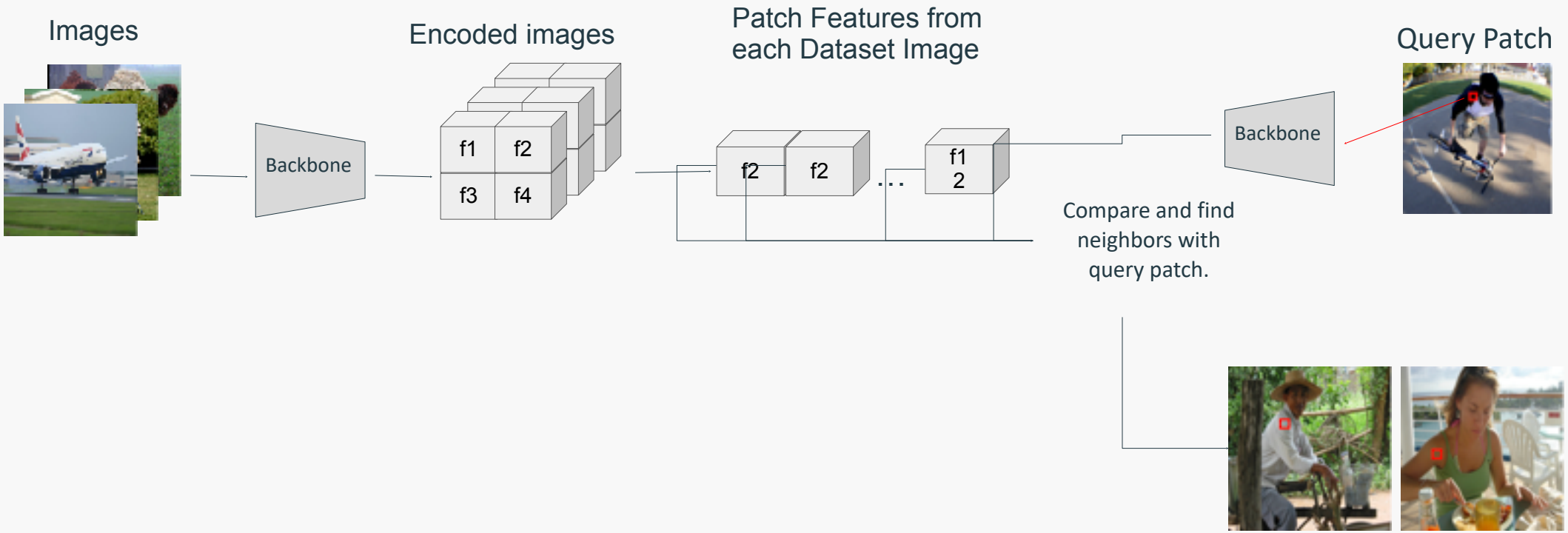
NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency. Pariza, Salehi, Burghouts, Locatello, Asano. arxiv 2024

PaNeCo: Patch Nearest neighbor Consistency

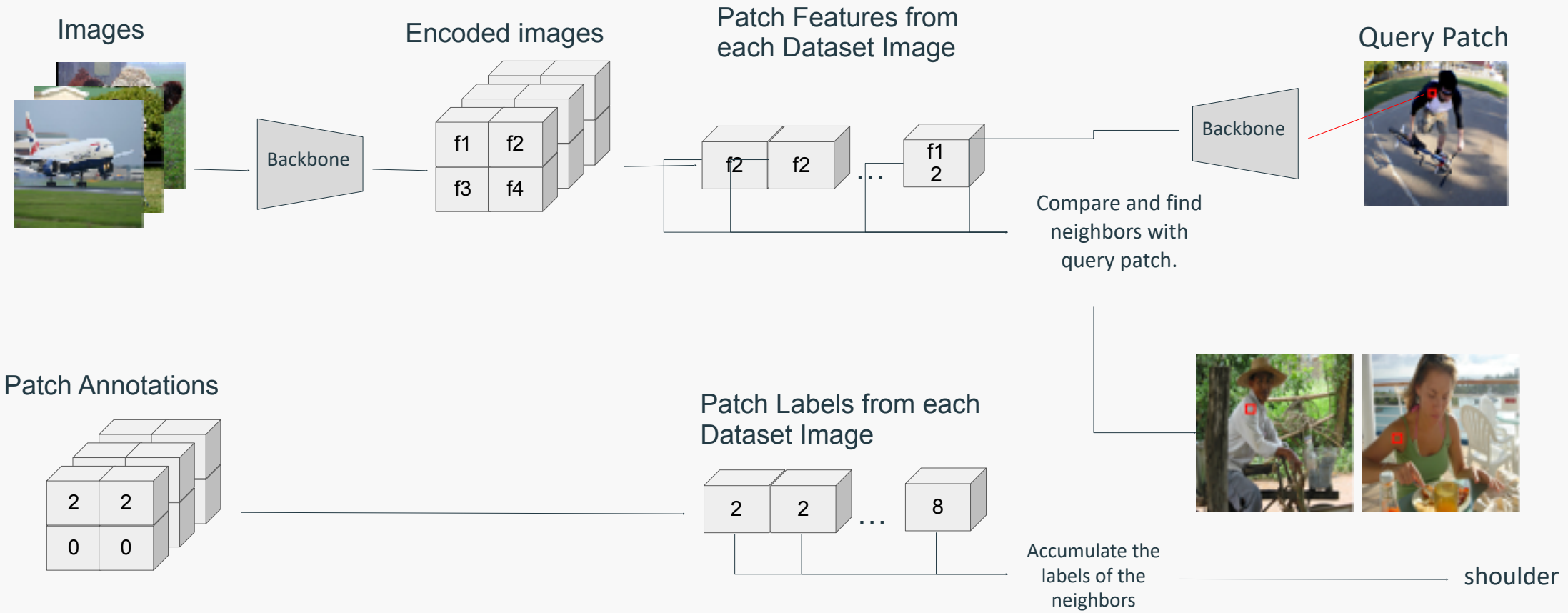


NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency. Pariza, Salehi, Burghouts, Locatello, Asano. arxiv 2024

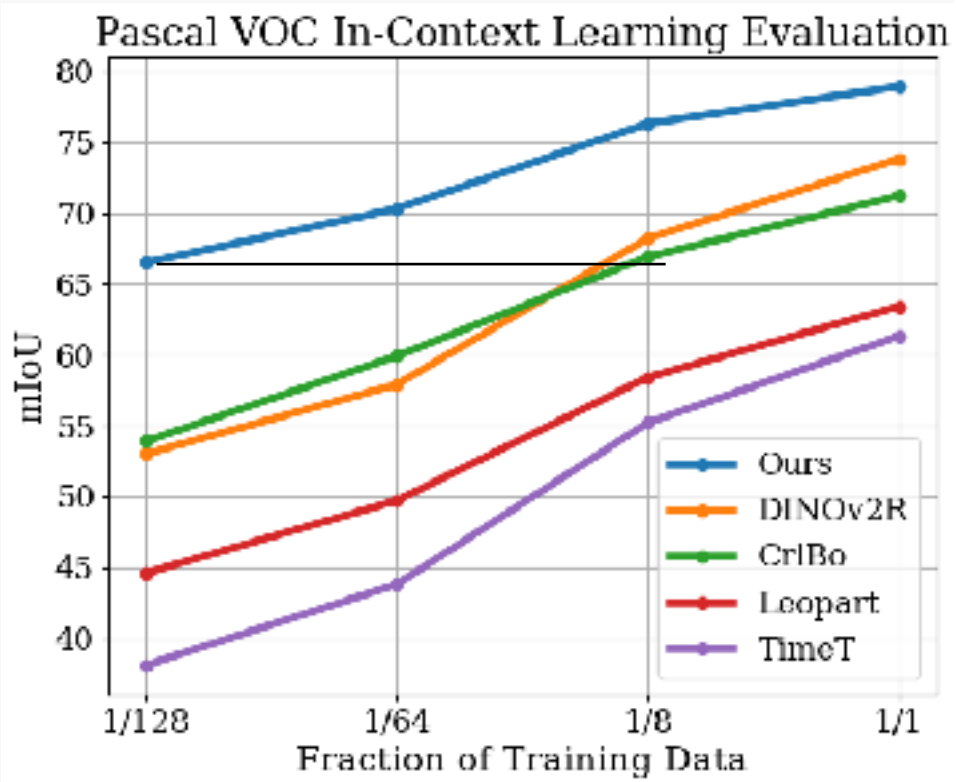
Evaluation 1: Visual in-context segmentation via dense NN retrieval



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



In-context scene understanding benchmark



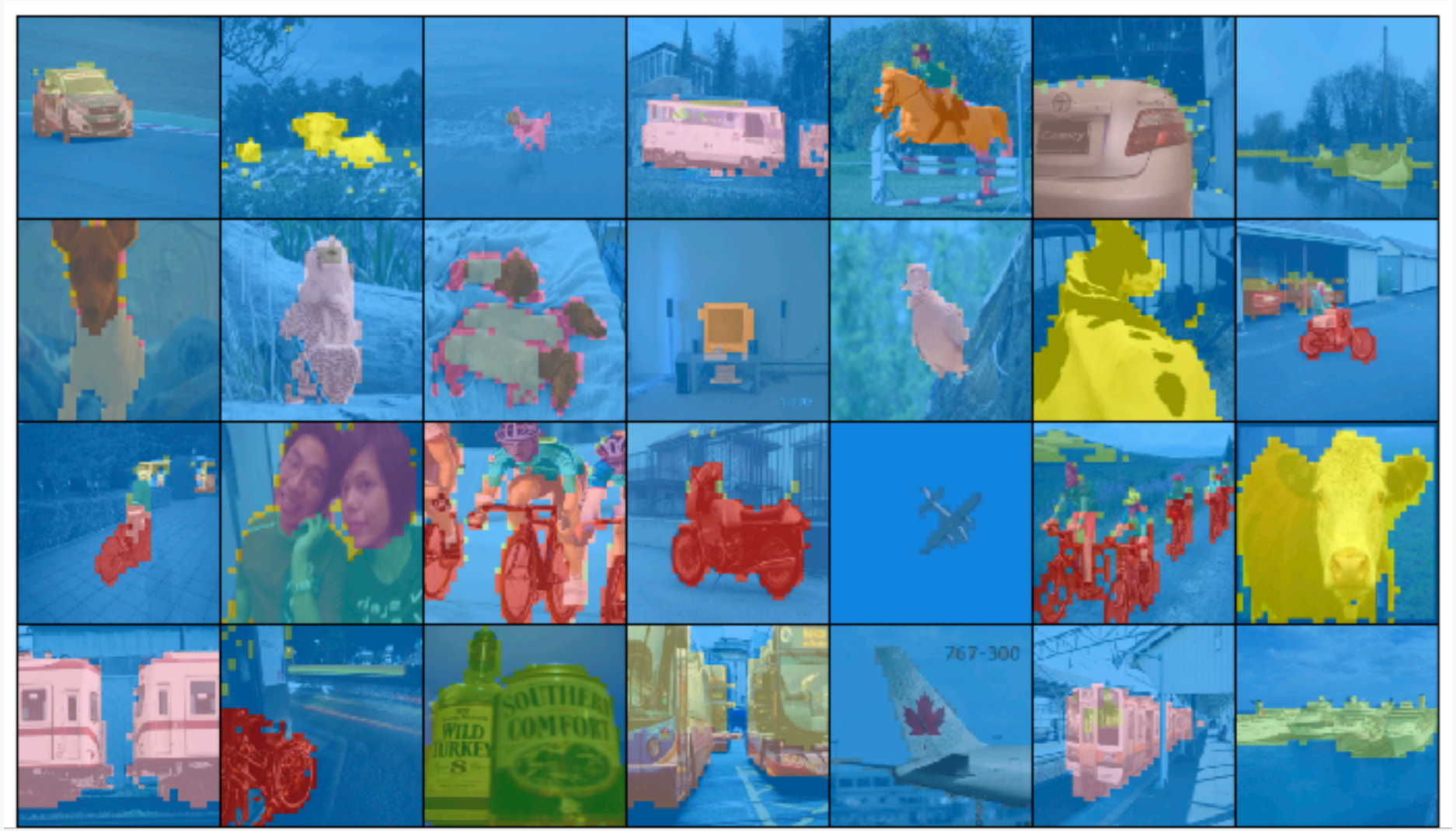
matches performances of DINOv2-R with ~15x less data

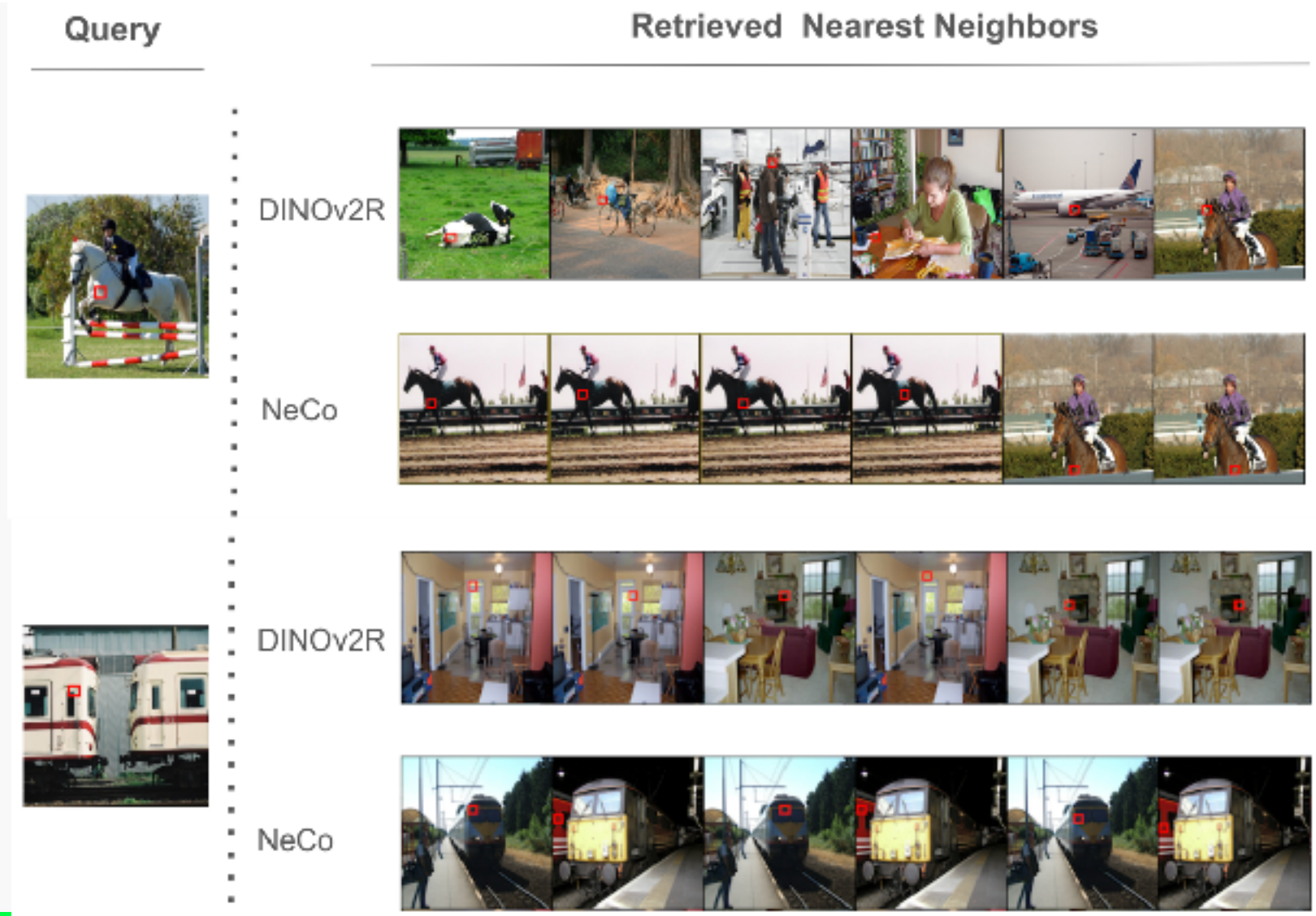
NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency. Pariza, Salehi, Burghouts, Locatello, Asano. arxiv 2024

(without any training)

Fundamental AI Lab

UTN




Query
Retrieved Nearest Neighbors
DINOv2R
NeCo
DINOv2R
NeCo

PaNeCo starting with different pretrained weights.

	Pascal VOC						COCO-Things					
	At Init			+PaNeCo			*At Init*			+PaNeCo		
Pretrain	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$^{\uparrow 11.0}$	51.2$^{\uparrow 20.1}$	68.6$^{\uparrow 2.5}$	7.6	28.0	58.9	20.4$^{\uparrow 12.8}$	52.8$^{\uparrow 24.8}$	67.7$^{\uparrow 8.8}$

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

PaNeCo starting with different pretrained weights.

	Pascal VOC						COCO-Things					
	At Init			+PaNeCo			*At Init*			+PaNeCo		
Pretrain	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$^{\uparrow 11.0}$	51.2$^{\uparrow 20.1}$	68.6$^{\uparrow 2.5}$	7.6	28.0	58.9	20.4$^{\uparrow 12.8}$	52.8$^{\uparrow 24.8}$	67.7$^{\uparrow 8.8}$
DINO [15]	4.3	17.3	50.2	14.5$^{\uparrow 10.2}$	47.9$^{\uparrow 30.6}$	61.3$^{\uparrow 11.1}$	5.4	19.2	43.9	16.9$^{\uparrow 11.5}$	50.0$^{\uparrow 30.8}$	62.4$^{\uparrow 18.5}$

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

PaNeCo starting with different pretrained weights.

	Pascal VOC						COCO-Things					
	At Init			+PaNeCo			*At Init*			+PaNeCo		
Pretrain	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$^{\uparrow 11.0}$	51.2$^{\uparrow 20.1}$	68.6$^{\uparrow 2.5}$	7.6	28.0	58.9	20.4$^{\uparrow 12.8}$	52.8$^{\uparrow 24.8}$	67.7$^{\uparrow 8.8}$
DINO [15]	4.3	17.3	50.2	14.5$^{\uparrow 10.2}$	47.9$^{\uparrow 30.6}$	61.3$^{\uparrow 11.1}$	5.4	19.2	43.9	16.9$^{\uparrow 11.5}$	50.0$^{\uparrow 30.8}$	62.4$^{\uparrow 18.5}$
TimeT [66]	12.2	46.2	66.3	17.9$^{\uparrow 5.7}$	52.1$^{\uparrow 5.9}$	68.5$^{\uparrow 2.2}$	18.4	44.6	58.2	20.6$^{\uparrow 2.2}$	54.3$^{\uparrow 9.7}$	64.8$^{\uparrow 6.6}$

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

PaNeCo starting with different pretrained weights.

	Pascal VOC						COCO-Things					
	At Init			+PaNeCo			*At Init*			+PaNeCo		
Pretrain	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$^{\uparrow 11.0}$	51.2$^{\uparrow 20.1}$	68.6$^{\uparrow 2.5}$	7.6	28.0	58.9	20.4$^{\uparrow 12.8}$	52.8$^{\uparrow 24.8}$	67.7$^{\uparrow 8.8}$
DINO [15]	4.3	17.3	50.2	14.5$^{\uparrow 10.2}$	47.9$^{\uparrow 30.6}$	61.3$^{\uparrow 11.1}$	5.4	19.2	43.9	16.9$^{\uparrow 11.5}$	50.0$^{\uparrow 30.8}$	62.4$^{\uparrow 18.5}$
TimeT [66]	12.2	46.2	66.3	17.9$^{\uparrow 5.7}$	52.1$^{\uparrow 5.9}$	68.5$^{\uparrow 2.2}$	18.4	44.6	58.2	20.6$^{\uparrow 2.2}$	54.3$^{\uparrow 9.7}$	64.8$^{\uparrow 6.6}$
Leopart [93]	15.4	51.2	66.5	21.0$^{\uparrow 5.6}$	55.3$^{\uparrow 4.1}$	68.3$^{\uparrow 1.8}$	14.8	53.2	63.0	18.8$^{\uparrow 4.0}$	53.9$^{\uparrow 0.7}$	65.4$^{\uparrow 2.4}$

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

PaNeCo starting with different pretrained weights.

	Pascal VOC						COCO-Things					
	At Init			+PaNeCo			*At Init*			+PaNeCo		
Pretrain	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$^{\uparrow 11.0}$	51.2$^{\uparrow 20.1}$	68.6$^{\uparrow 2.5}$	7.6	28.0	58.9	20.4$^{\uparrow 12.8}$	52.8$^{\uparrow 24.8}$	67.7$^{\uparrow 8.8}$
DINO [15]	4.3	17.3	50.2	14.5$^{\uparrow 10.2}$	47.9$^{\uparrow 30.6}$	61.3$^{\uparrow 11.1}$	5.4	19.2	43.9	16.9$^{\uparrow 11.5}$	50.0$^{\uparrow 30.8}$	62.4$^{\uparrow 18.5}$
TimeT [66]	12.2	46.2	66.3	17.9$^{\uparrow 5.7}$	52.1$^{\uparrow 5.9}$	68.5$^{\uparrow 2.2}$	18.4	44.6	58.2	20.6$^{\uparrow 2.2}$	54.3$^{\uparrow 9.7}$	64.8$^{\uparrow 6.6}$
Leopart [93]	15.4	51.2	66.5	21.0$^{\uparrow 5.6}$	55.3$^{\uparrow 4.1}$	68.3$^{\uparrow 1.8}$	14.8	53.2	63.0	18.8$^{\uparrow 4.0}$	53.9$^{\uparrow 0.7}$	65.4$^{\uparrow 2.4}$
CrIBo [49]	18.3	54.5	71.6	21.7$^{\uparrow 3.4}$	59.6$^{\uparrow 5.1}$	72.1$^{\uparrow 0.5}$	14.5	48.3	64.3	21.1$^{\uparrow 6.6}$	54.0$^{\uparrow 5.7}$	68.0$^{\uparrow 3.7}$

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

→ NeCo considerably boosts (↑) the performance of **different backbones**

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

Key takeaways

- **Dense Patch-ordering** is loss 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!

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

No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations

Walter Simoncini, Spyros Gidaris, Andrei Bursuc, Yuki M. Asano

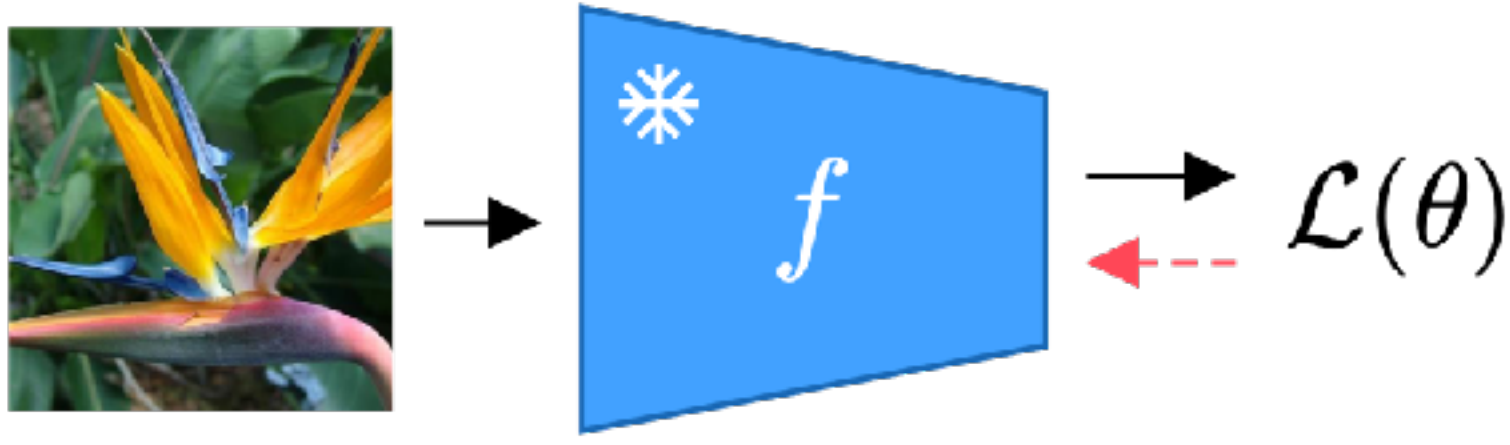
NeurIPS 2024

Fundamental AI Lab

Valeo

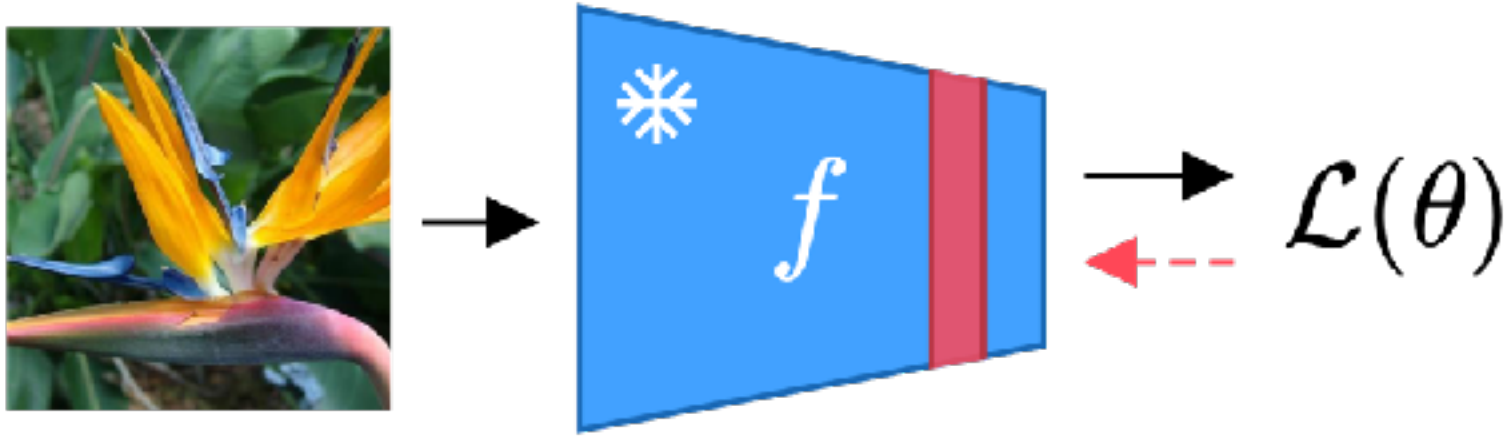
Idea

The **loss** indicates how the network output should **change** to solve a task



Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Idea



Gradients carry **information** about the **network, task** and **data**

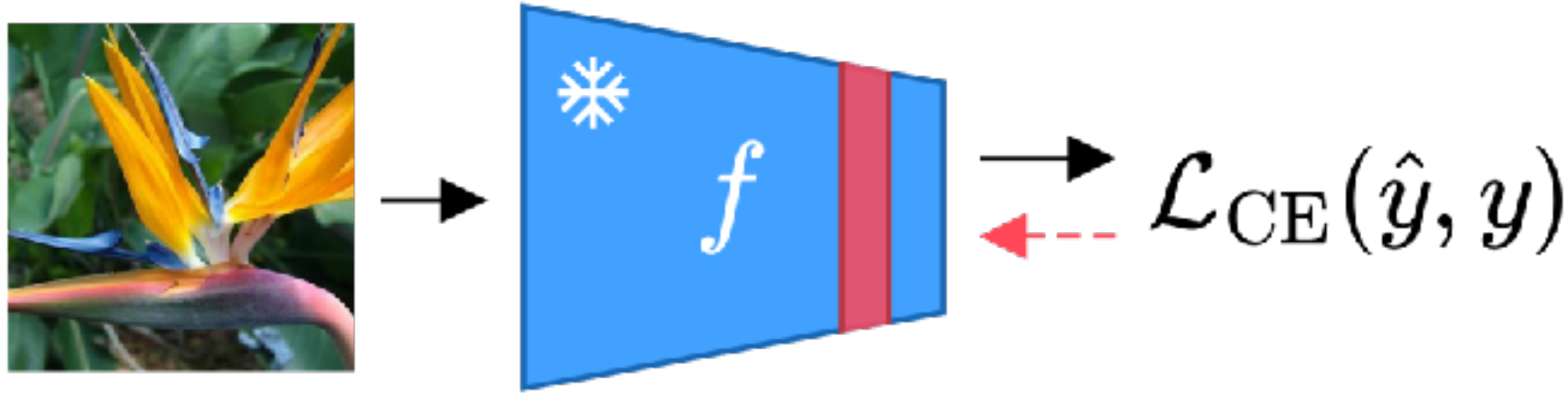
Why not use them as **features too?**

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Idea

Traditionally, vision models are trained with **supervision**

Labels are needed to compute gradients 😢



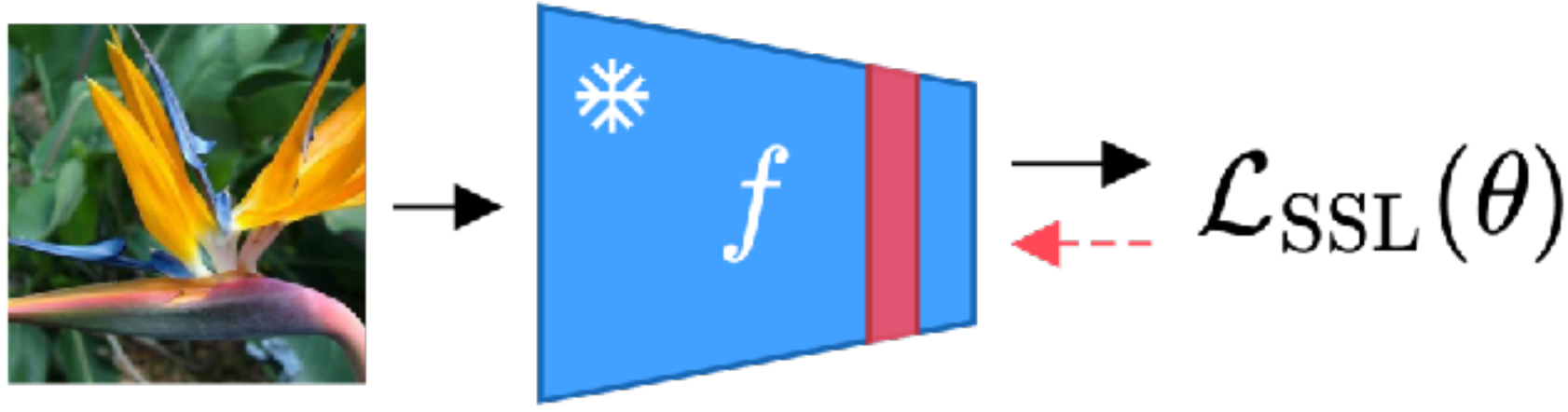
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Idea

Self Supervised Learning to the rescue! 🎉

No Labels

Several Proxyllosses



Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

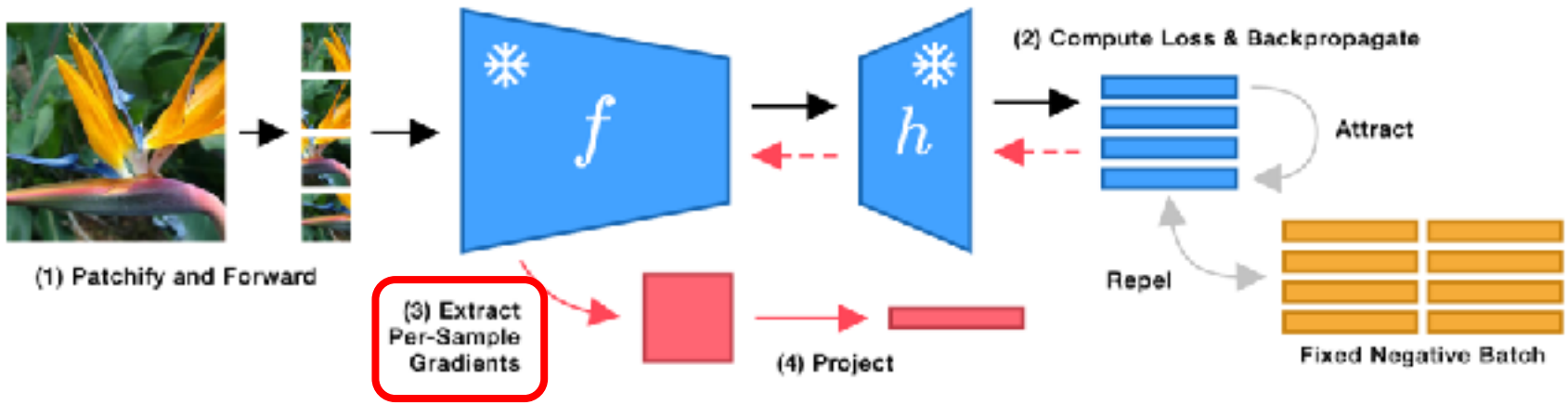
Method

Given a pre-trained vision transformer we

Forward an image (or multiple views of it).

Compute a self-supervised loss & backpropagate.

Extract the **gradients** wrt the **weights** of a layer and downsample them.



Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Method

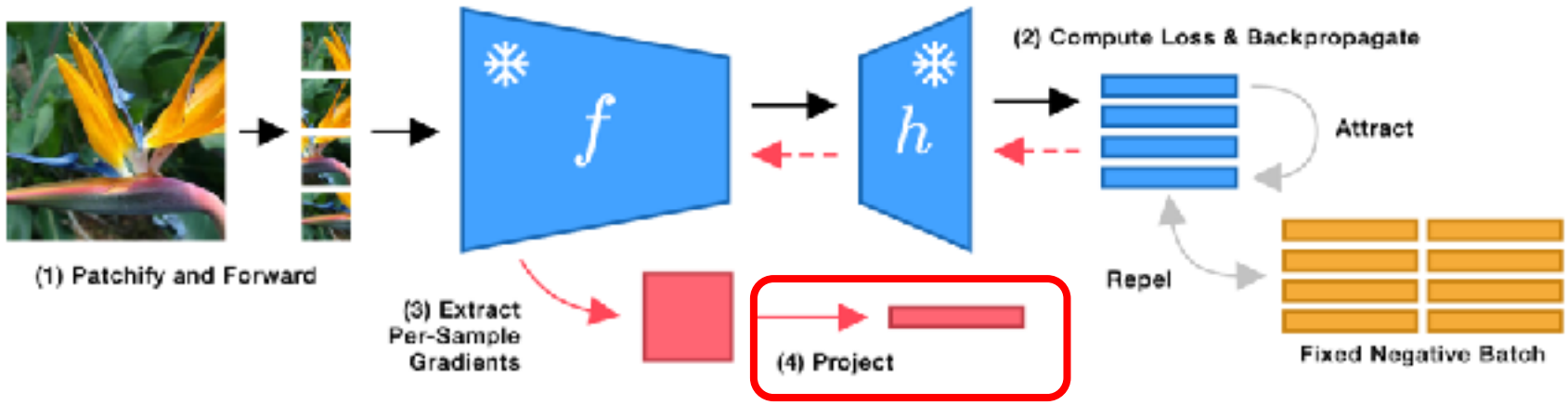
Given a pre-trained vision transformer we

Forward an image (or multiple views of it).

Compute a self-supervised loss & backpropagate.

Extract the **gradients** wrt the **weights** of a layer and downsample them.

Project gradients and obtain a FUNGI (Feature from UNsupervised GradIents).



Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

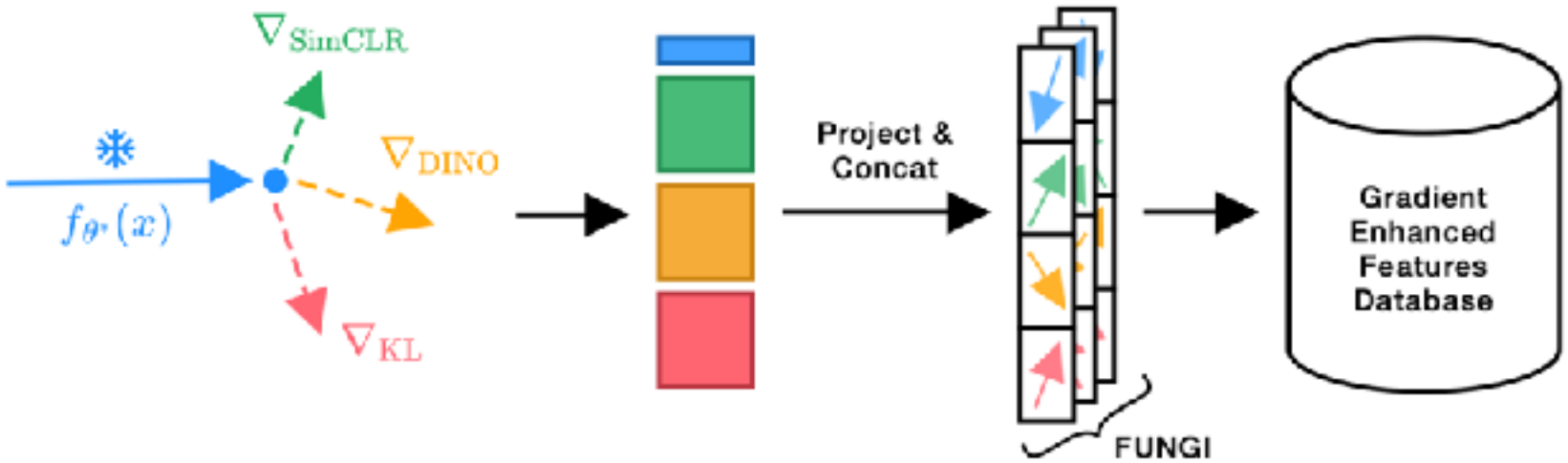
Self-Supervised Objectives

Three objectives: DINO, SimCLR and KL.

We concatenate (multiple) gradients and the model embeddings.

More **powerful**, as they contain information from multiple objectives.

More **robust**, as the other features can counteract a bad local gradient approximation



Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Code Implementation

```python
# Wrap the model using the FUNGI feature extractor
wrapper = FUNGIWrapper(
    model=model,
    # (1) Select a layer
    target_layer="blocks.11.attn.proj",
    device=device,
    # (2) Choose the SSL objectives
    extractor_configs=[
        KLConfig(),
        DINOConfig()
    ]
)

# (3) Extract FUNGI
fungi = wrapper(PIL.Image.open("image.jpg"))
```

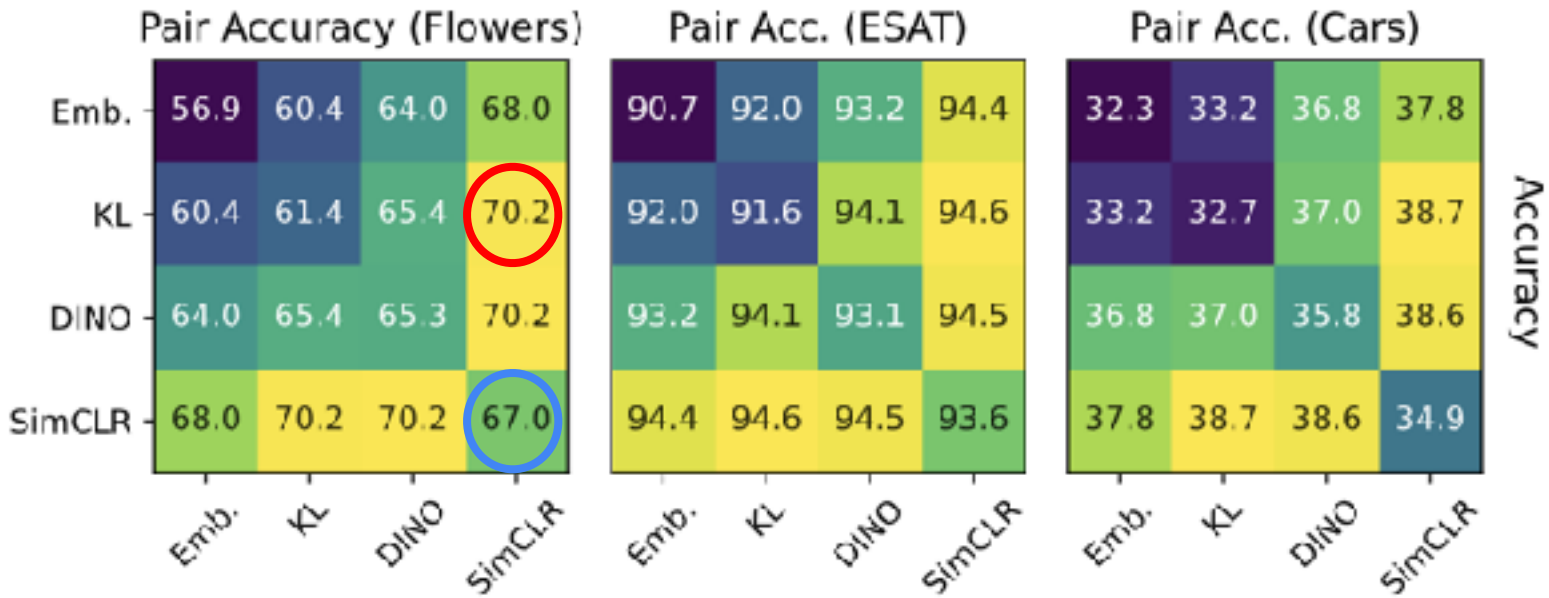
https://github.com/WalterSimoncini/fungivision

Properties

Gradient features can **enhance** the **retrieval** performance

When **combined** with other gradient features or the embeddings, they improve further

Gradients encode **different** and **complementary** information to each other



Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Experiments

We evaluate **FUNGI** across 20 backbones, 22 datasets and 3 modalities (vision, language and audio), for a total of **~1000 experiments**.

We evaluate **FUNGI** in

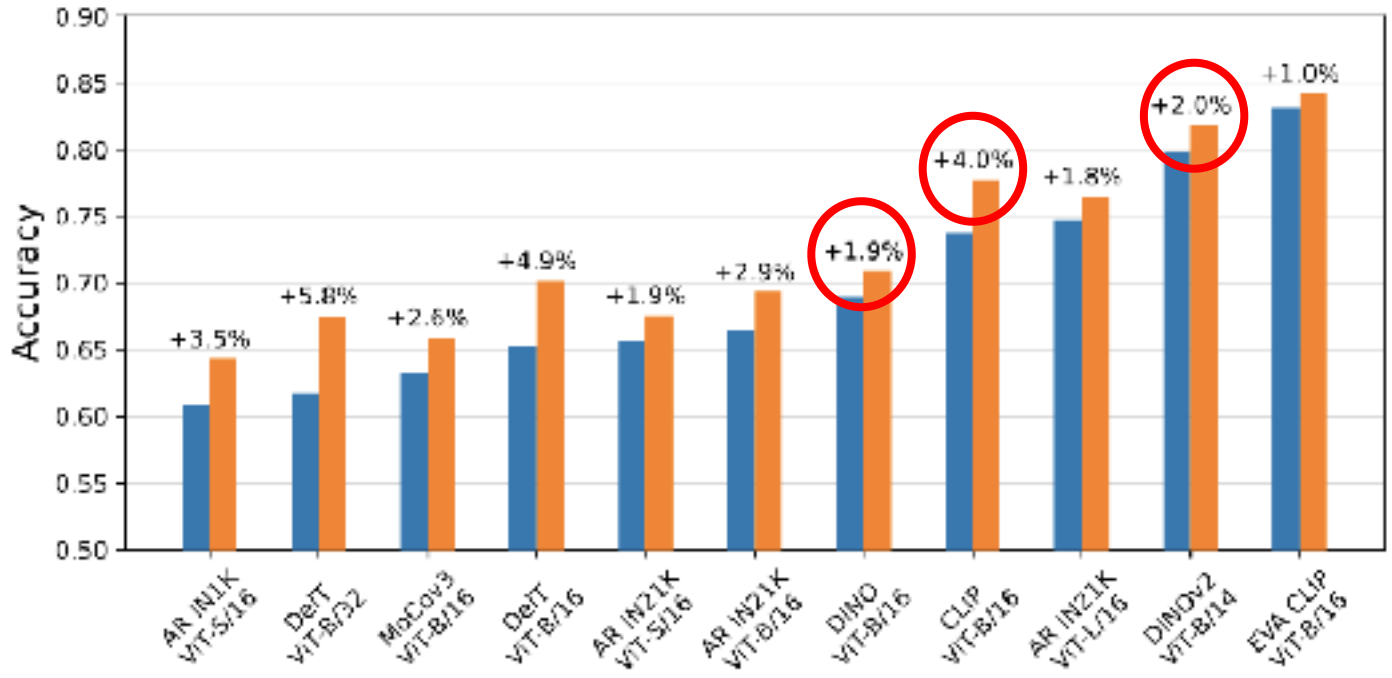
- Retrieval & k-nearest neighbor (k-nn) classification
- Linear classification
- k-means clustering

Retrieval-Based Tasks

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

k-nn classification (vision)

Large improvements in k-nn, even for DINO v1/2 and CLIP

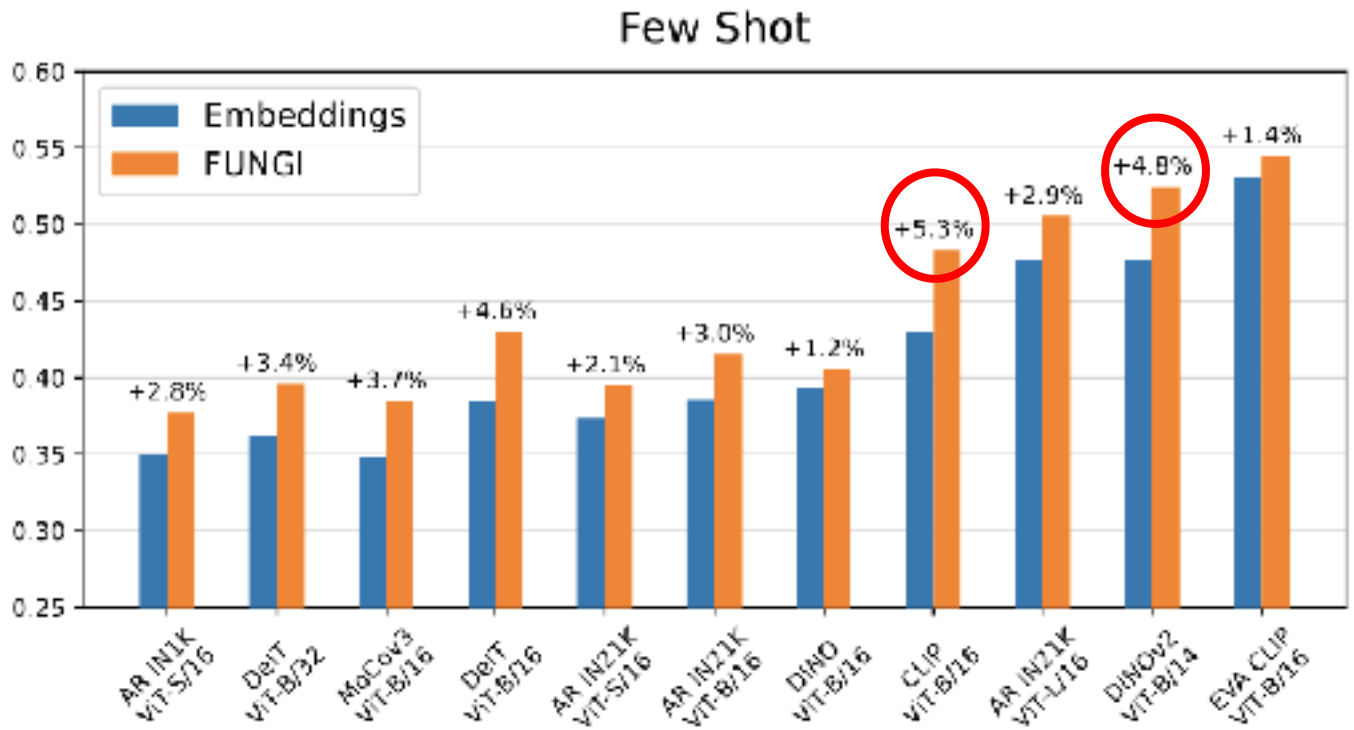


Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

k-nn classification (vision)

Up to **5.3%** better for CLIP and **4.8%** for DINOv2 few-shot

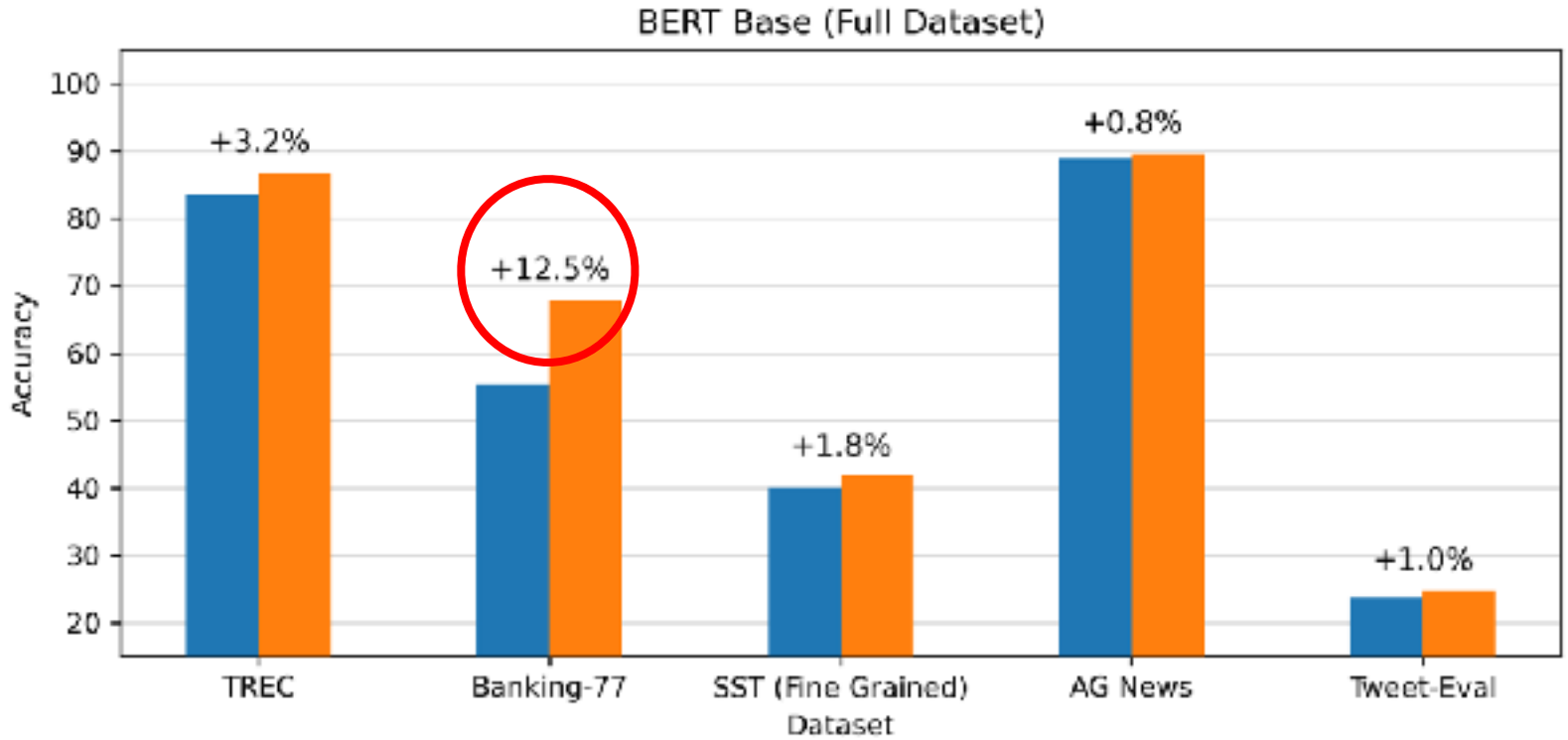


Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

k-nn classification (language)

Up to **12.5%** better using BERT Base

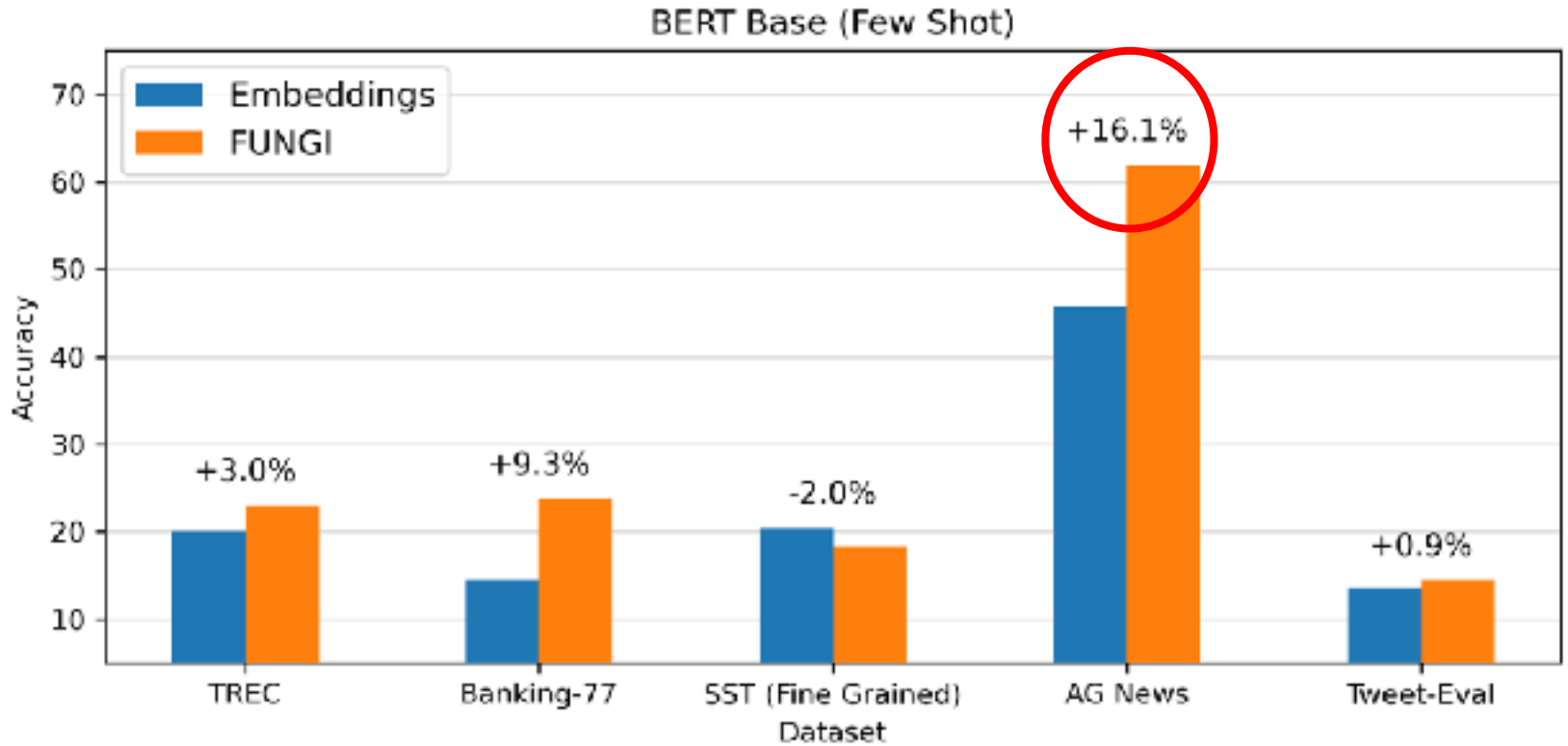


Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

k-nn classification (language)

Up to **16%** better in few shot classification using BERT Base

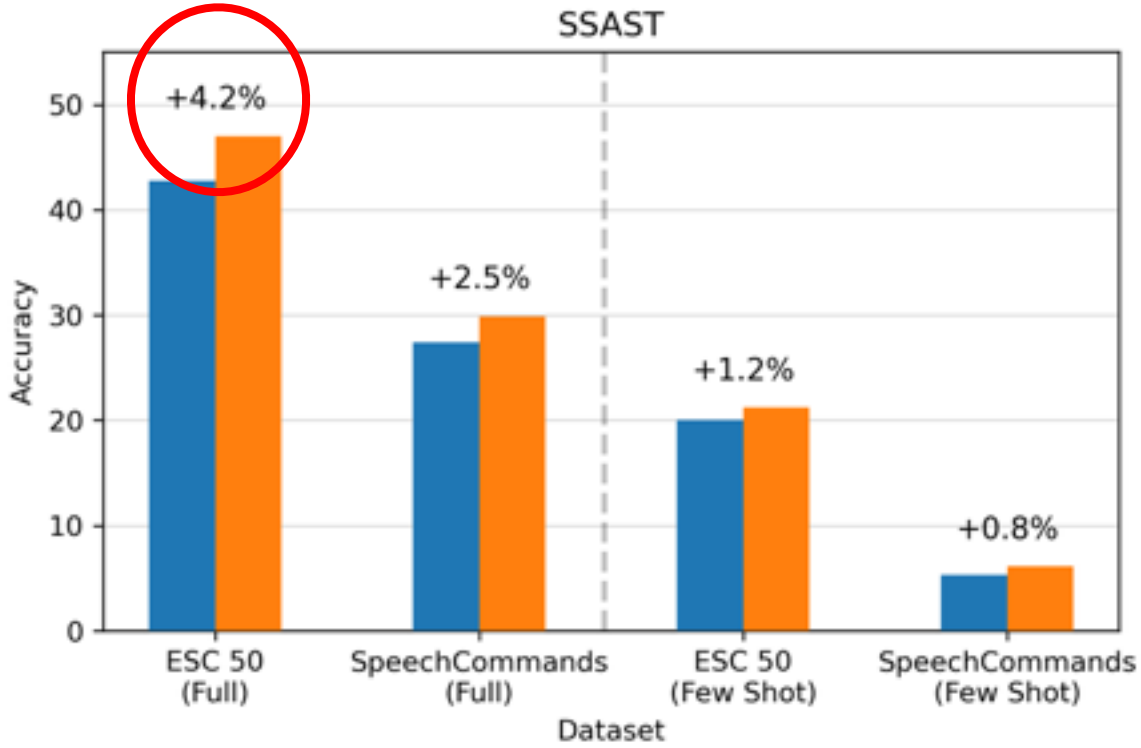


Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

k-nn classification (audio)

Up to **4.2%** better using a SSAST backbone



Fundamental AI Lab

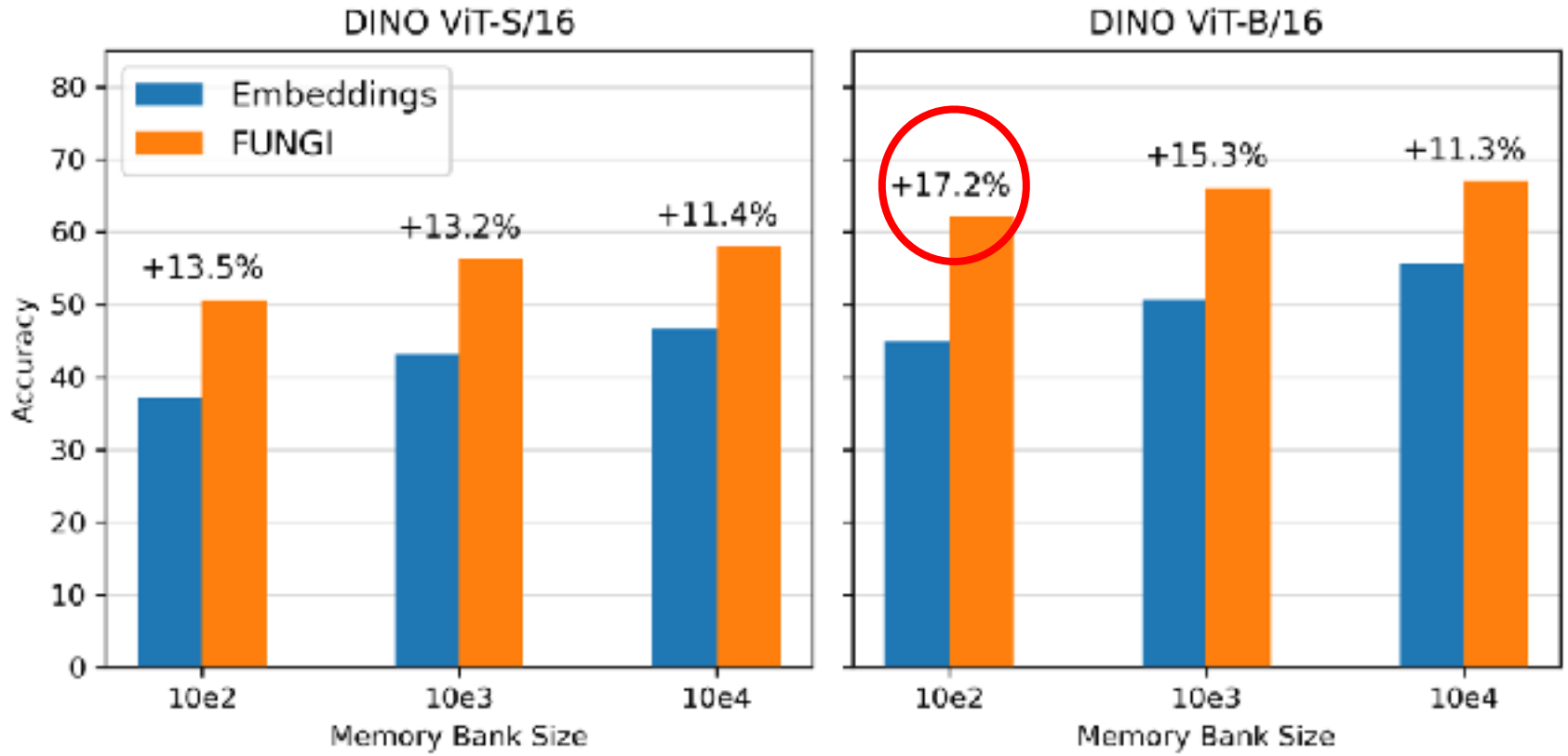
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Visual In-Context Segmentation

Fundamental AI Lab

In-Context Semantic Segmentation (Hummingbird) on Pascal VOC

Up to **17%** improvement over **DINOv1**

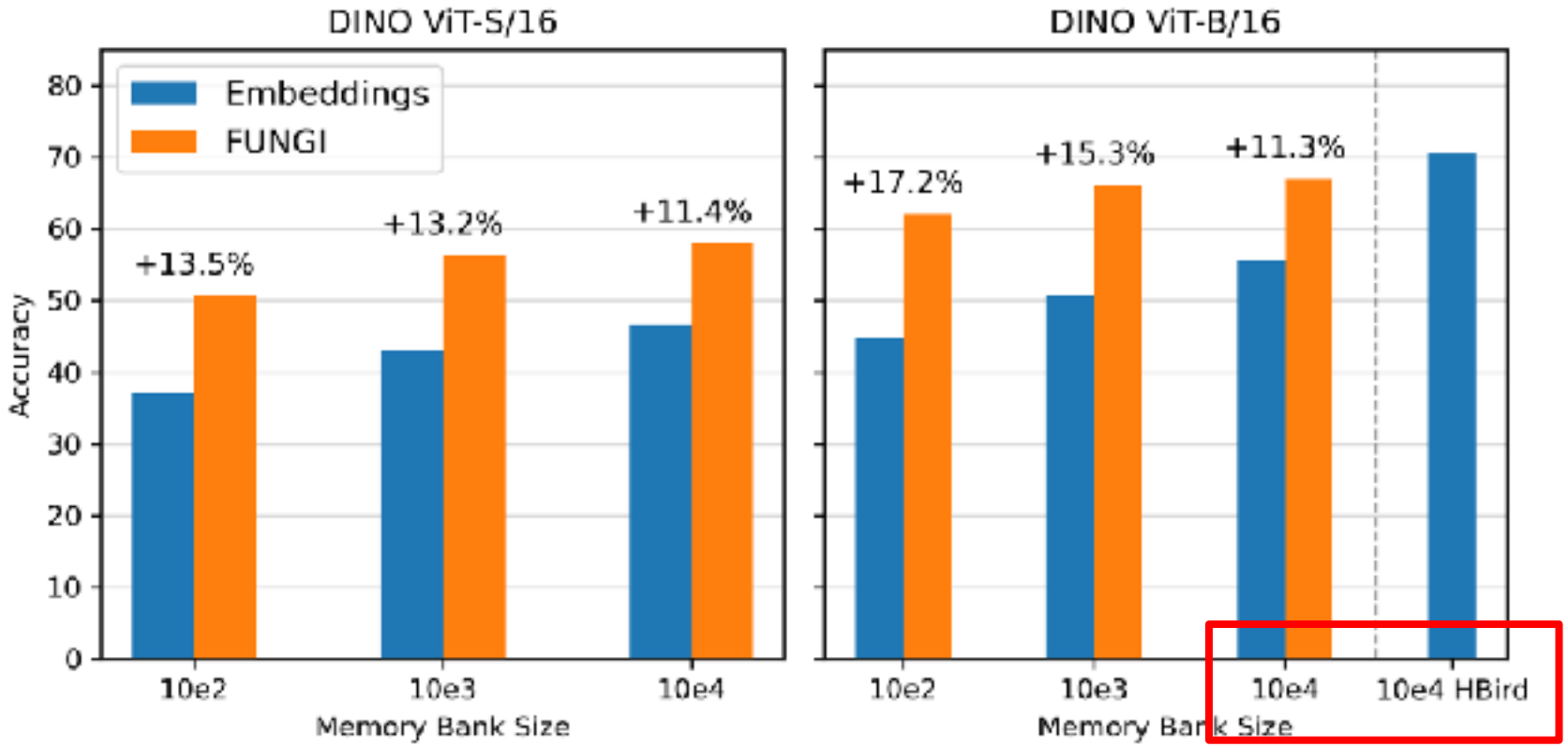


Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

In-Context Semantic Segmentation on Pascal VOC

Close to SoTA, **without any training!**



Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Fundamental AI Lab

In-Context Semantic Segmentation [8] on Pascal VOC

Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Language

Intent classification on banking-77 with GPT 4o mini

Examples selected with **FUNGI** improve accuracy by **+2.5%**!

```
You have to annotate banking-related queries
with an appropriate intent. You must choose a
single class in the following comma-separated
list:

{list of classes}

You must only output the class, nothing more.
Examples follow:

{20 (text, label) training pairs}

The test sample is: {text}
```

	Banking-77
Embeddings	88.7
+ KL + SimCLR	**91.2** ↑2.5

Fundamental AI Lab

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Other Evaluations

Vision Linear Classification

Our features improve the performance of logistic regression for most backbones

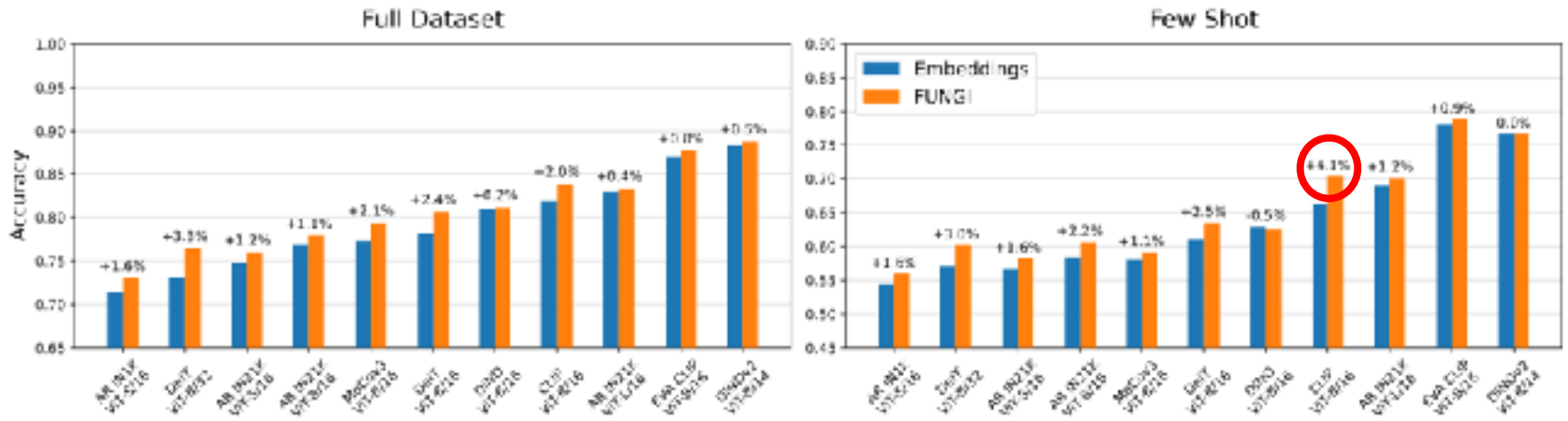


Figure 10: FUNGI works across backbones for linear probing. Accuracy in logistic regression-based image classification of embeddings versus FUNGI features on various ViT backbones, both for full dataset and few shot setups, averaged over 11 datasets. For the FUNGI features, we chose the best performing combination across datasets. "AR" indicates AugReg backbones (Steiner et al., 2022).

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Summary

Self-supervised **gradients can be used as features**, and can perform better than the embeddings

Combining gradients (and embeddings) produces **strong features** for **retrieval**, **linear classification** and **clustering**

FUNGI works **across modalities**

Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Learning to Count without Annotations

Lukas Knobel, Tengda Han, Yuki M. Asan

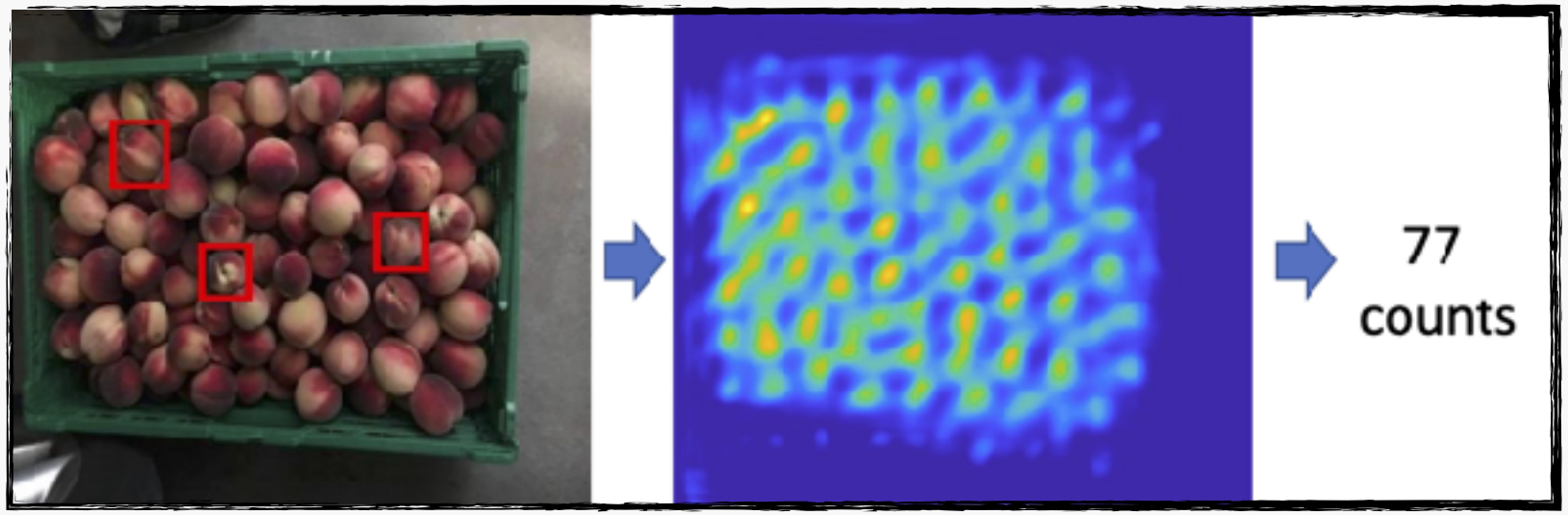
CVPR 2024

Fundamental AI Lab

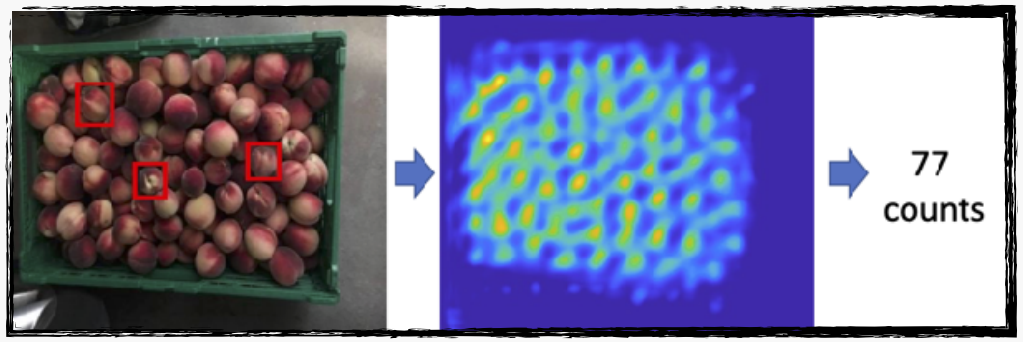
UTN

University of Amsterdam

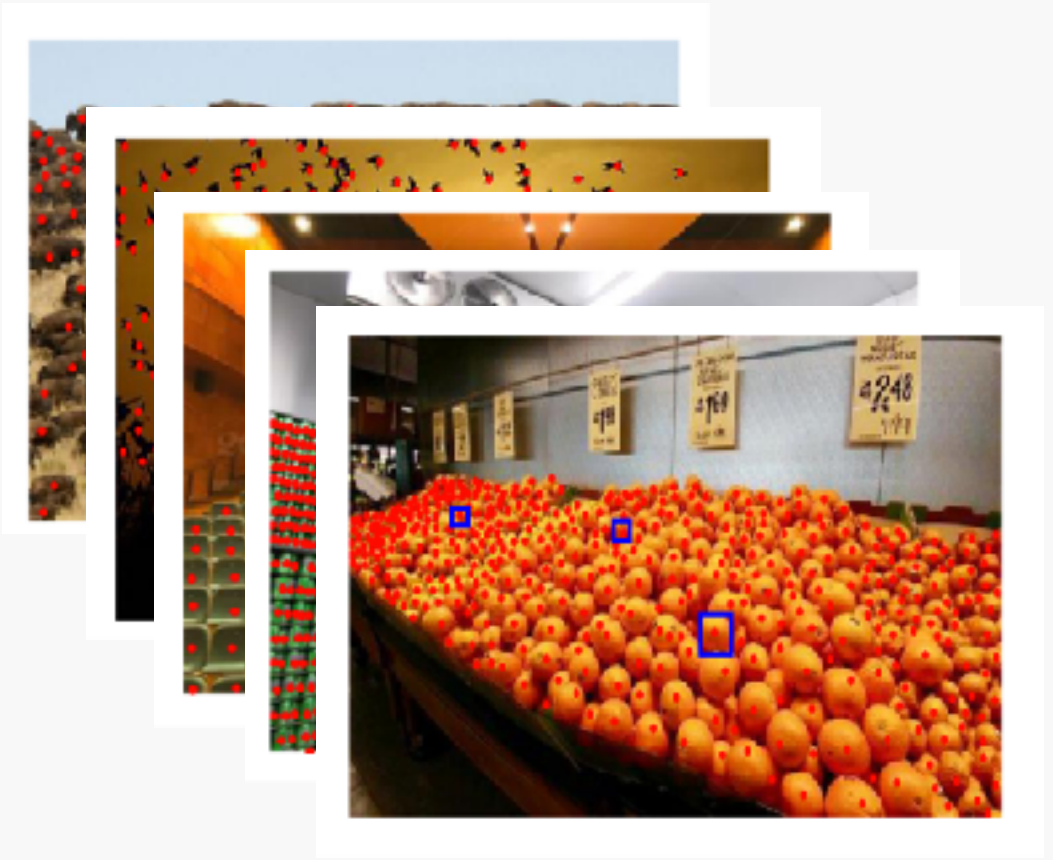
Referential Counting

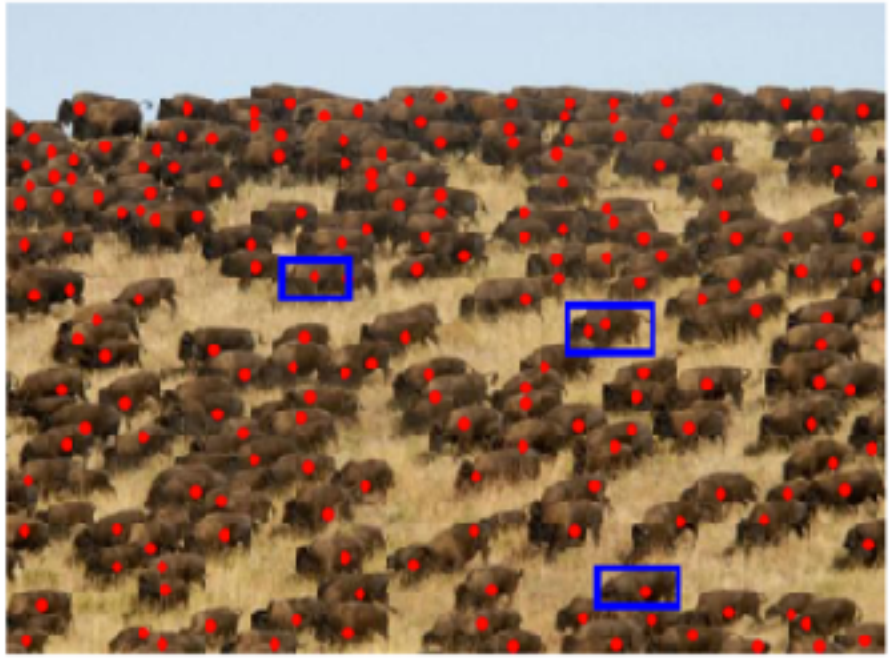


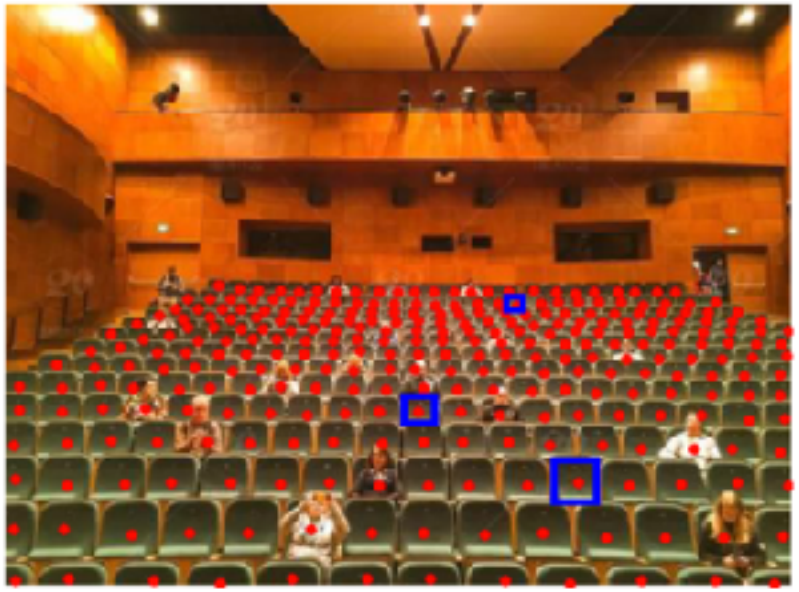
Referential Counting



...typically need counting supervision

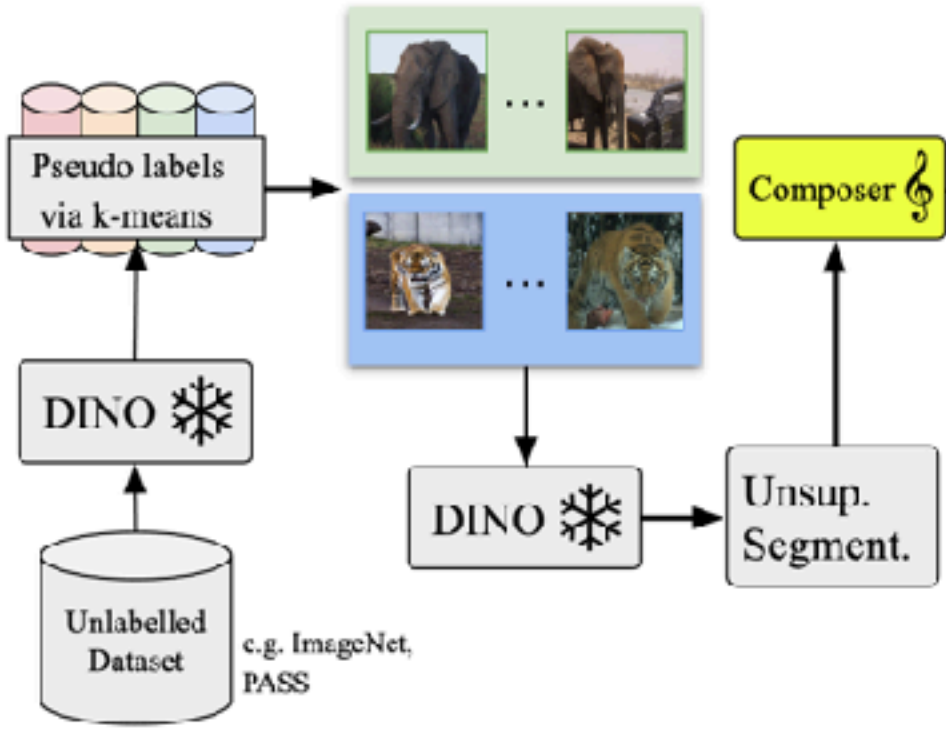




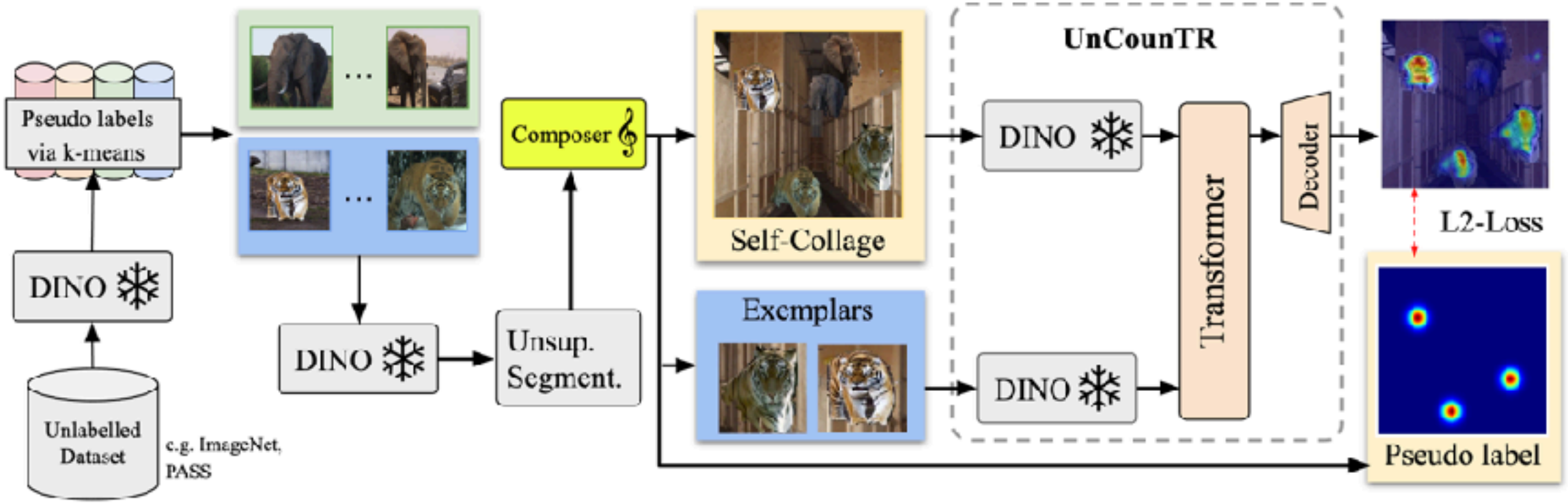


But pretrained & frozen models are *(very)* good.

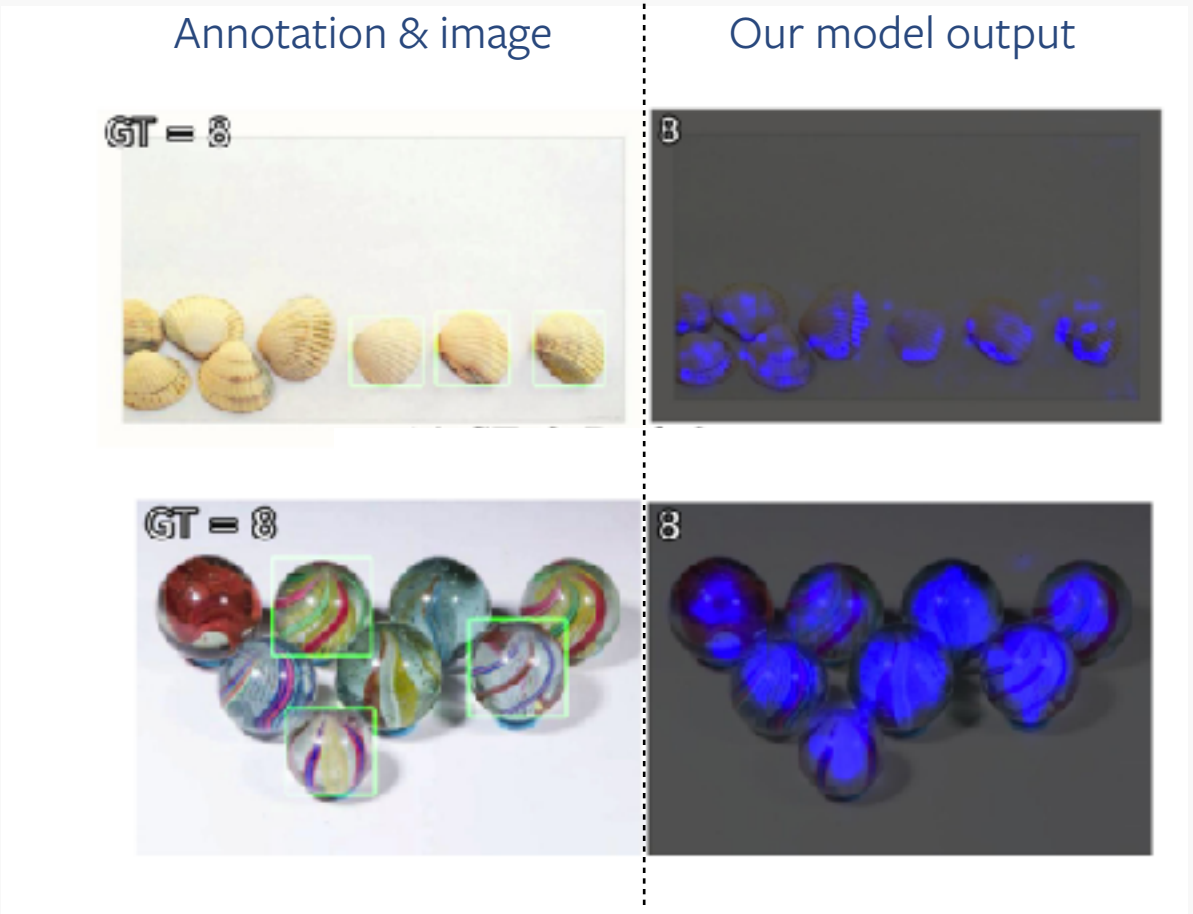
Good #1: unsupervised salient object segmentation



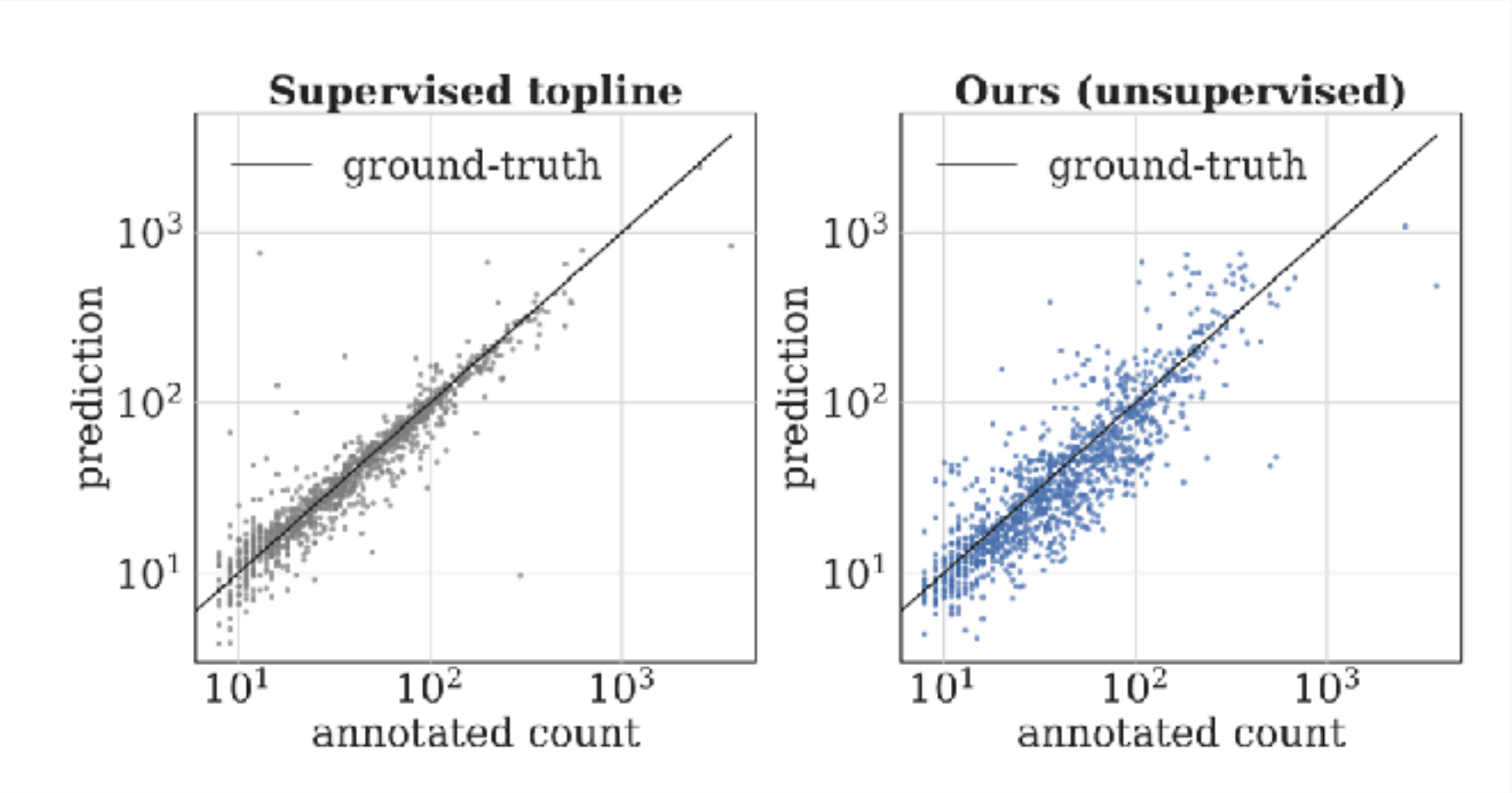
Good #2: strong & robust feature extraction



Result: we can learn object counting *without any* supervised data.



Result: we can learn object counting *without any* supervised data.



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

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

ICLR (oral & outstanding paper) 2024

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

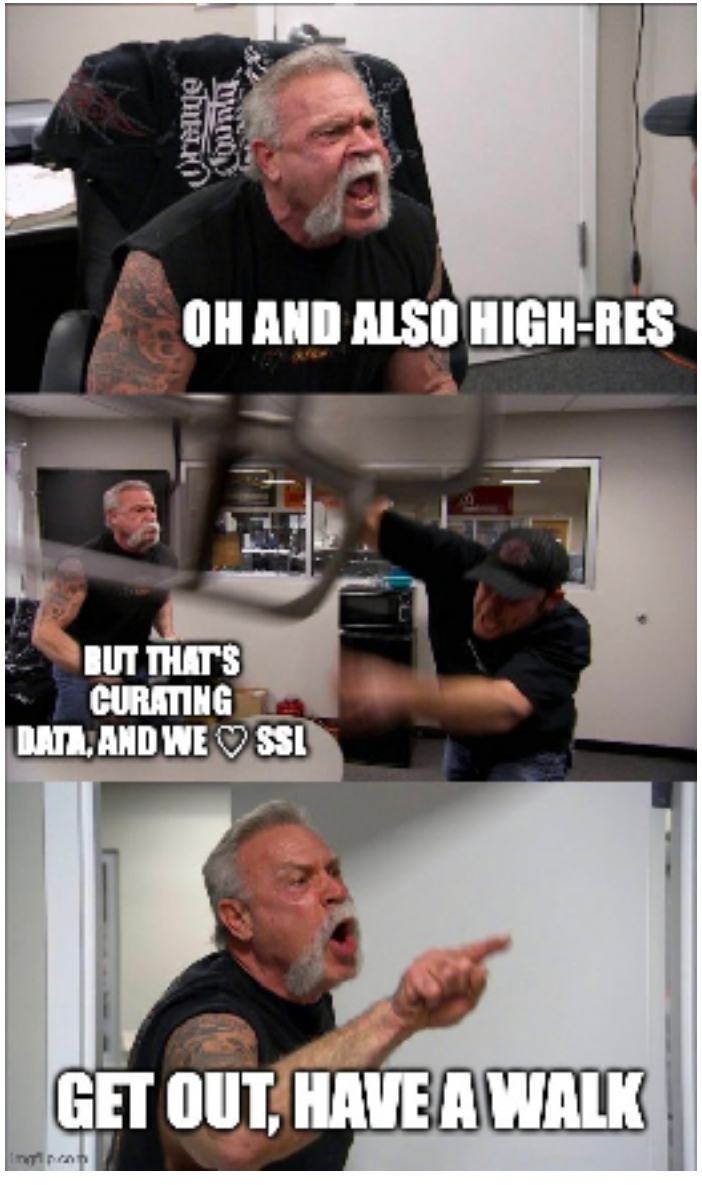
TimeTuning:
DINO as init & use temporal info of videos.

How powerful is time without image-pretraining?

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

Motivated by: Asano Rupprecht, Vedaldi. *A critical analysis of self-supervision, or what we can learn from a single image*. ICLR 2020

us figuring out which video to use



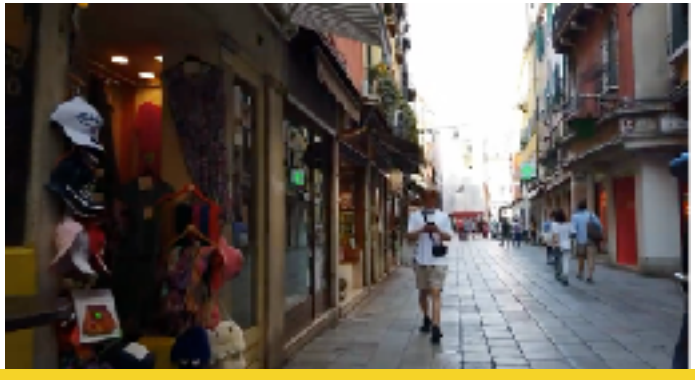
- ✓ Long
- ✓ High-res, smooth
- ✓ Semantically rich
- ✓ Scalable (we ❤️ SSL)

↓

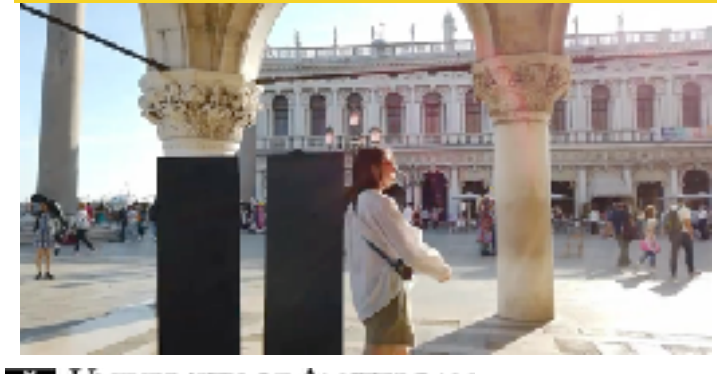
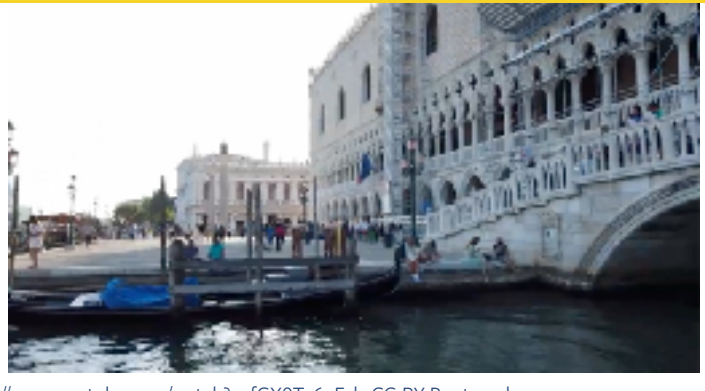
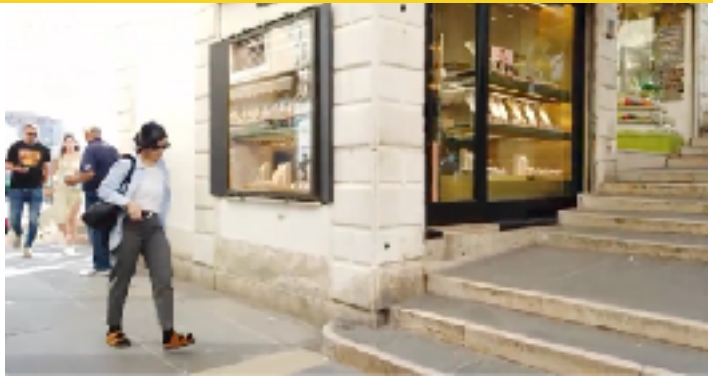
Walking Tours

WTours proposed for learning video compression in ACCV 2022: Wiles et al. Compressed Vision for Efficient Video Understanding.



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

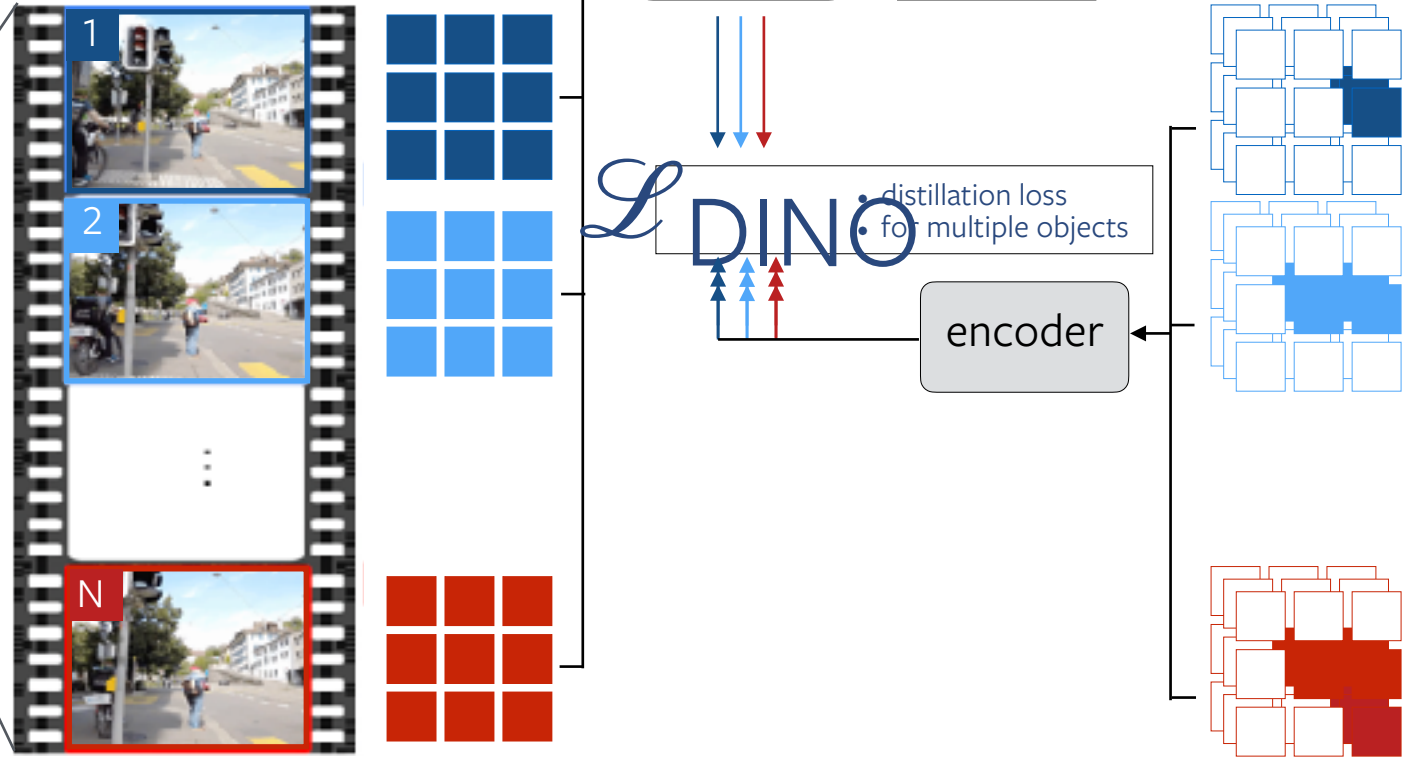




WT Venice: https://www.youtube.com/watch?v=fGX0Te6pFvk. CC-BY Poptravel.

Dora: **Disc**o**v**er and T**ra**ck

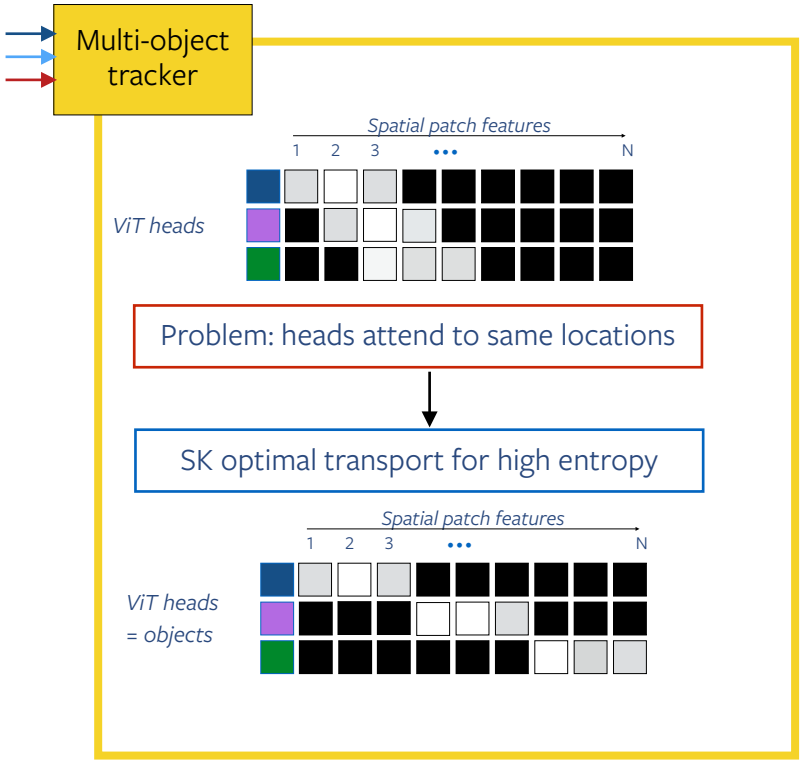
High-level idea:
1) track multiple objects across time
2) enforce invariance of features across time

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

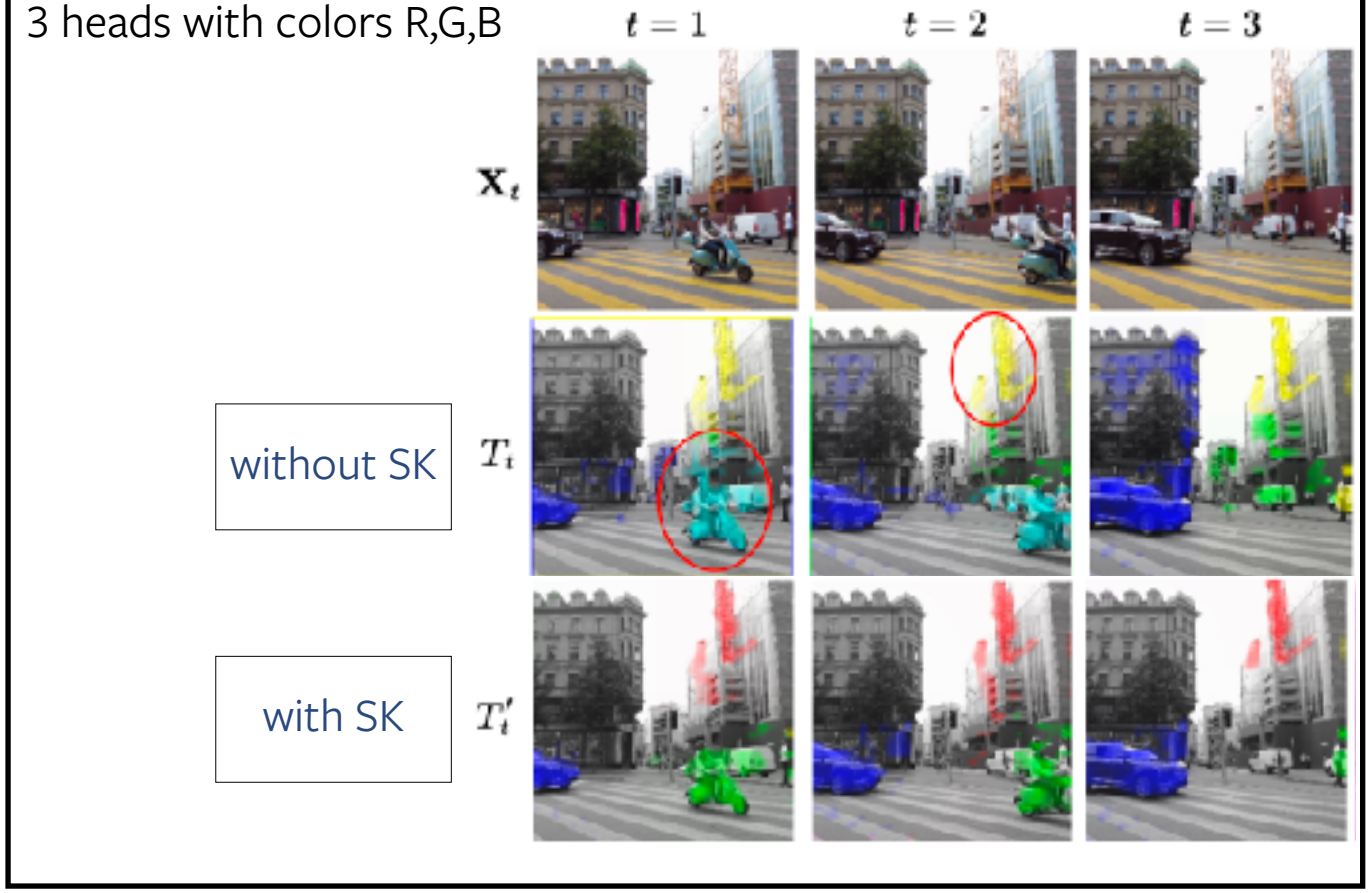


Venkataramanan, Rizve, Carreira, Asano*, Avrithis*. *Is ImageNet worth 1 video? Learning strong image encoders from 1 long unlabelled video.* ICLR 2024

Spreading attention with Sinkhorn-Knopp

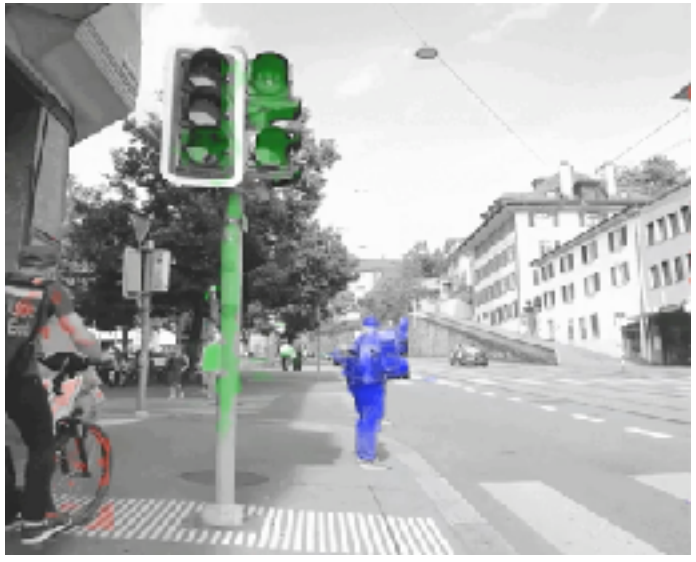


Visualise attention of
3 heads with colors R,G,B



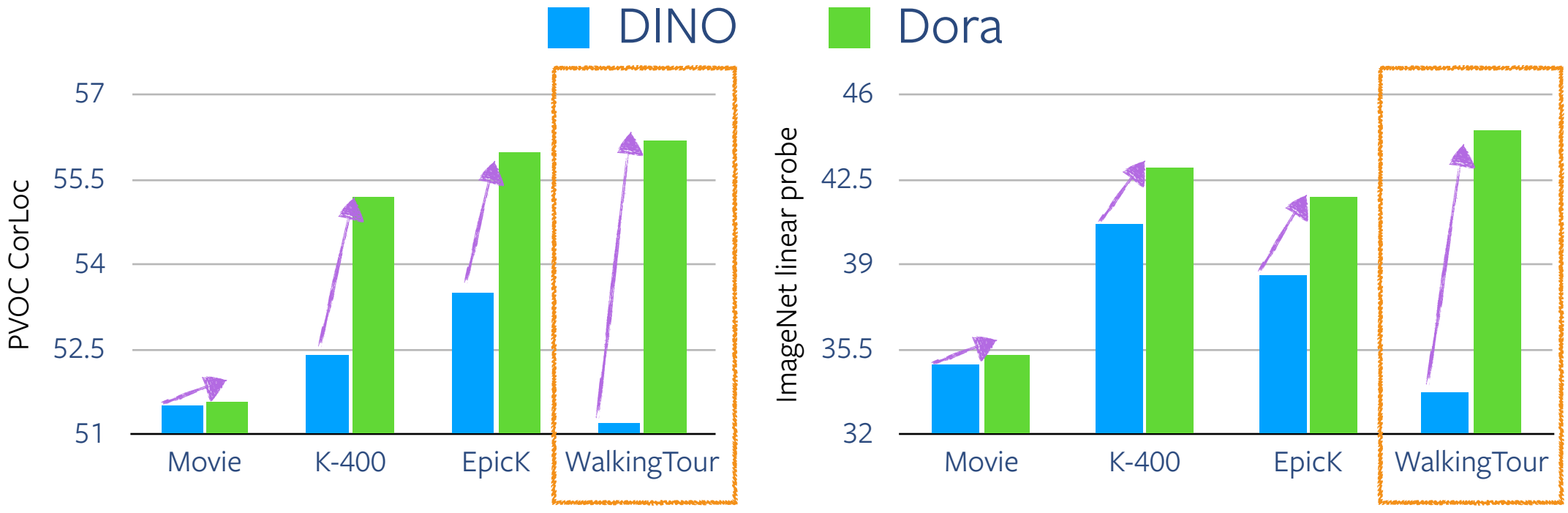
Venkataramanan, Rizve, Carreira, Asano*, Avrithis*. *Is ImageNet worth 1 video? Learning strong image encoders from 1 long unlabelled video.* ICLR 2024

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



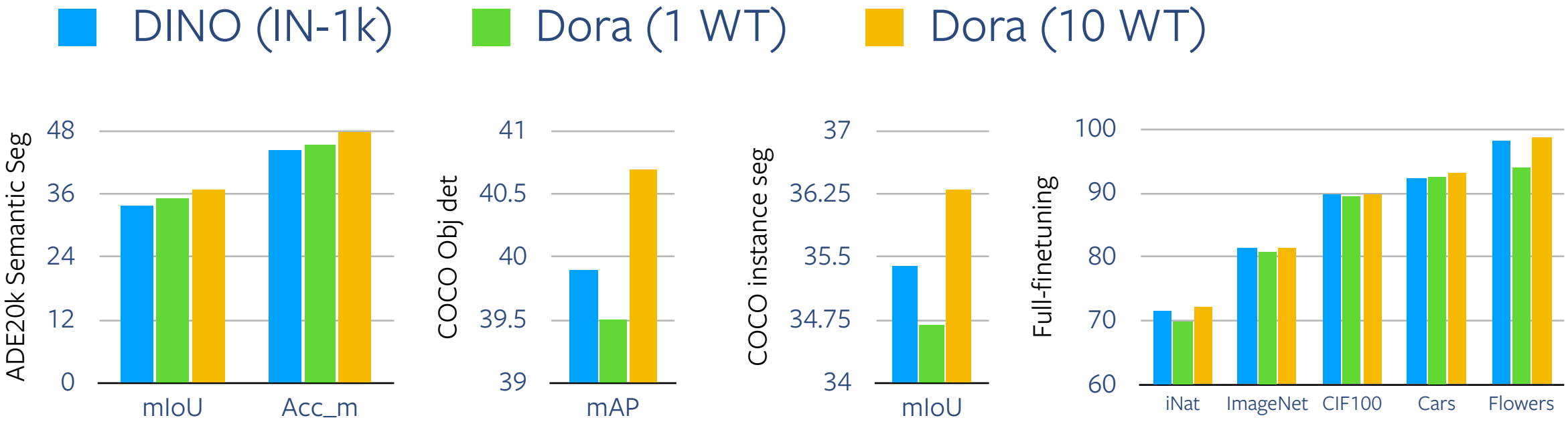
Venkataramanan, Rizve, Carreira, Asano*, Avrithis*. *Is ImageNet worth 1 video? Learning strong image encoders from 1 long unlabelled video*. ICLR 2024

Dora better than DINO
WT+ Dora: great match



Venkataramanan, Rizve, Carreira, Asano*, Avrithis*. *Is ImageNet worth 1 video? Learning strong image encoders from 1 long unlabelled video.* ICLR 2024

But how does it compare against ImageNet pretraining?



Dora (1WT) ~ on par with DINO (IN-1k)
Dora (10WT) > DINO (IN-1k) everywhere

Venkataramanan, Rizve, Carreira, Asano*, Avrithis*. *Is ImageNet worth 1 video? Learning strong image encoders from 1 long unlabelled video.* ICLR 2024

Key takeaways

- 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

Venkataramanan, Rizve, Carreira, Asano*, Avrithis*. *Is ImageNet worth 1 video? Learning strong image encoders from 1 long unlabelled video*. ICLR 2024