Improving Foundation Models (with academic compute)

YUKI M ASANO

ML IN PL 2024

Fundamental AI Lab

UTN

Improving Foundation Models (with academic compute)

YUKI M ASANO

Fundamental AI Lab

UTN

ML IN PL 2024

Instruction tuning: the difference between GPT-3 and ChatGPT

giant training corpora

↓

(raw) LLMs
(e.g. GPT-3)

❌ requires industrial compute

❌ model is relatively useless

Instruction tuning: the difference between GPT-3 and ChatGPT

giant training corpora

↓

(raw) LLMs
(e.g. GPT-3)

❌ requires industrial compute

❌ model is relatively useless

small IT datasets

↓

Instruction tuning →

Instruction tuning: the difference between GPT-3 and ChatGPT

giant training corpora

↓

(raw) LLMs
(e.g. GPT-3)

❌ requires industrial compute

❌ model is relatively useless

small IT datasets

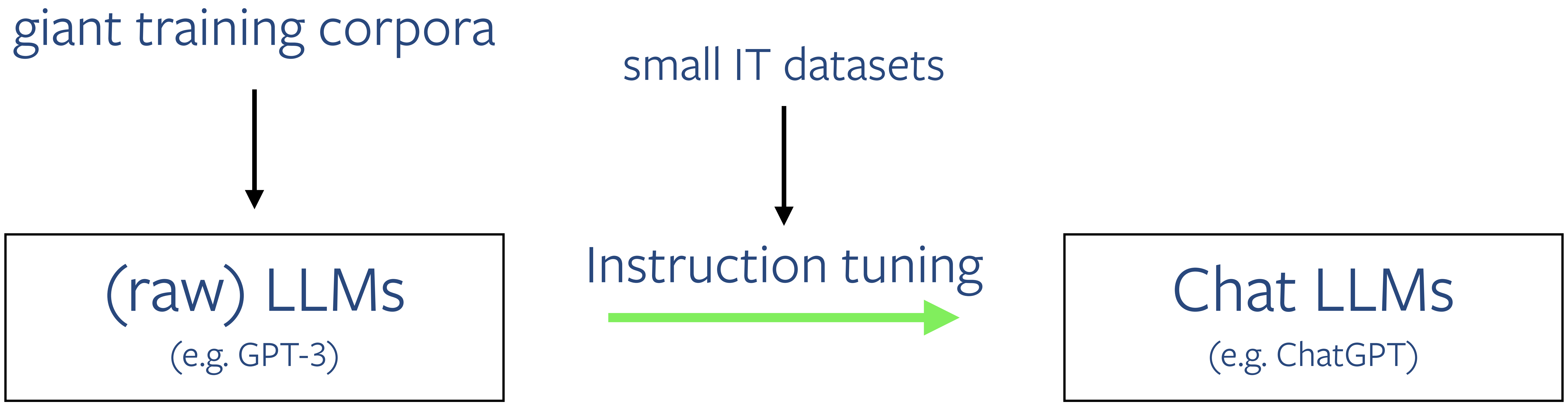
↓

Instruction tuning →

Chat LLMs
(e.g. ChatGPT)

Fundamental AI Lab

Instruction tuning: the difference between GPT-3 and ChatGPT



❌ requires industrial compute

❌ model is relatively useless

✅ requires fraction of compute

✅ makes LLM useful, customisable, better

Instruction tuning: the difference between GPT-3 and ChatGPT

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 (raw) LLMs

(e.g. GPT-3)

❌ requires industrial compute

❌ model is relatively useless

Instruction tuning

Chat LLMs

(e.g. ChatGPT)

✅ requires fraction of compute

✅ makes LLM useful, customisable, better

1st-gen Vision Foundation Models

(e.g. DINO, CLIP)

❌ requires industrial compute

✅ model is already useful

(raw) LLMs
(e.g. GPT-3)

❌ requires industrial compute

❌ model is relatively useless

Instruction tuning →

Chat LLMs
(e.g. ChatGPT)

✅ requires fraction of compute

✅ makes LLM useful, customisable, better

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

❌ requires industrial compute

✅ model is already useful

→

2nd-gen Vision Foundation Models
(e.g. Neco, TimeTuning)

✅ requires **fraction of compute**

✅ makes vision model **even better, across various tasks**

(raw) LLMs
(e.g. GPT-3)

❌ requires industrial compute

❌ model is relatively useless

Instruction tuning →

Chat LLMs
(e.g. ChatGPT)

✅ requires fraction of compute

✅ makes LLM useful, customisable, better

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

❌ requires industrial compute

✅ model is already useful

? →

2nd-gen Vision Foundation Models
(e.g. Neco, TimeTuning)

✅ requires **fraction of compute**

✅ makes vision model **even better, across various tasks**

Fundamental AI Lab

Computer vision needs more *post-pretraining*

(raw) LLMs
(e.g. GPT-3)

❌ requires industrial compute

❌ model is relatively useless

Instruction tuning →

Chat LLMs
(e.g. ChatGPT)

✅ requires fraction of compute

✅ makes LLM useful, customisable, better

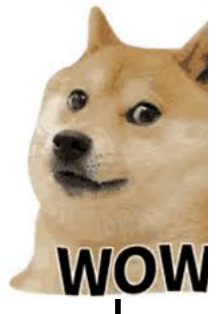
1st-gen Vision Foundation Models
(e.g. DINO, CLIP)

❌ requires industrial compute

✅ model is already useful

Post-pretraining →

2nd-gen Vision Foundation Models
(e.g. Neco, TimeTuning)

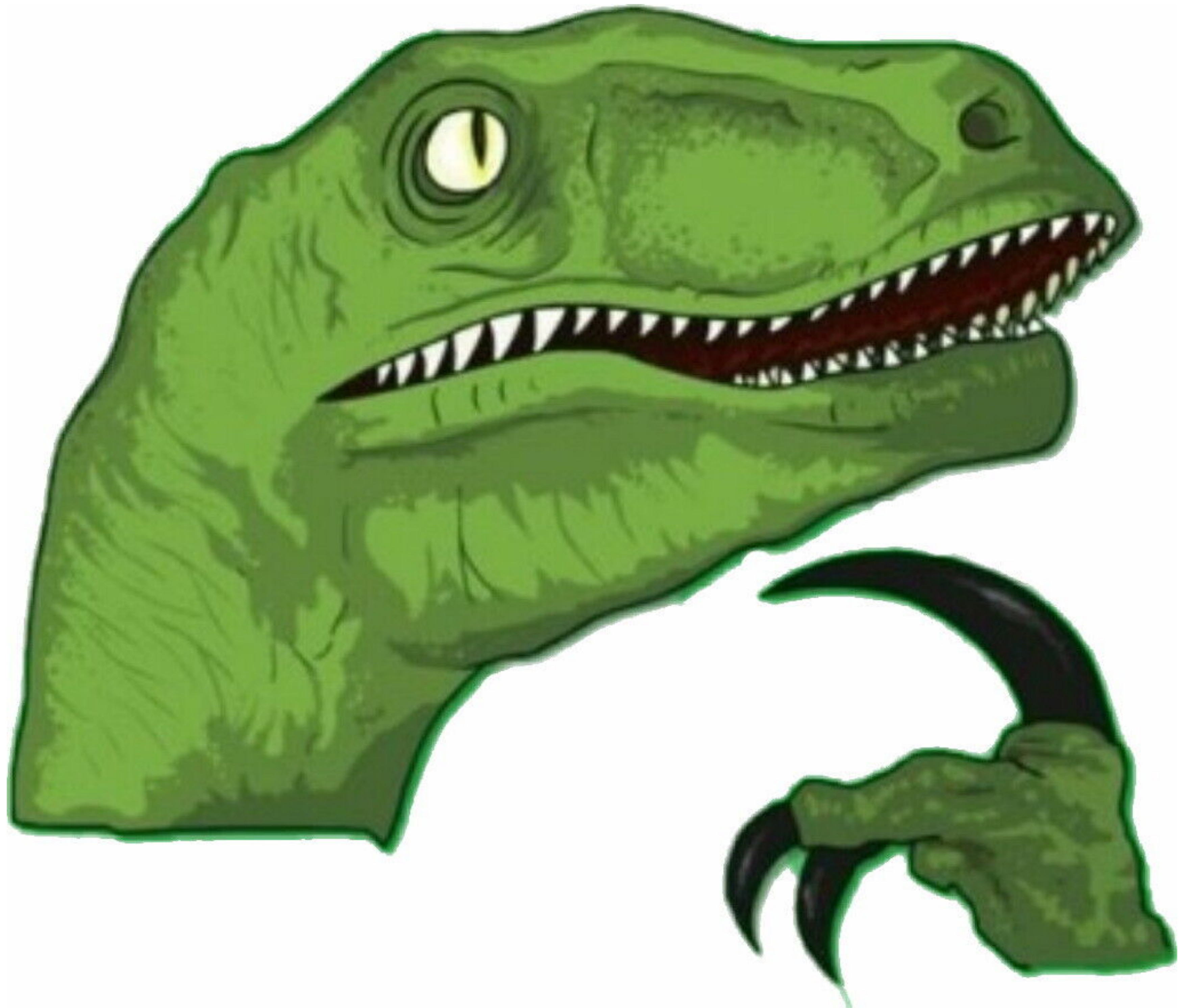


✅ requires **fraction of compute**

✅ makes vision model **even better, across various tasks**

Post-pretraining

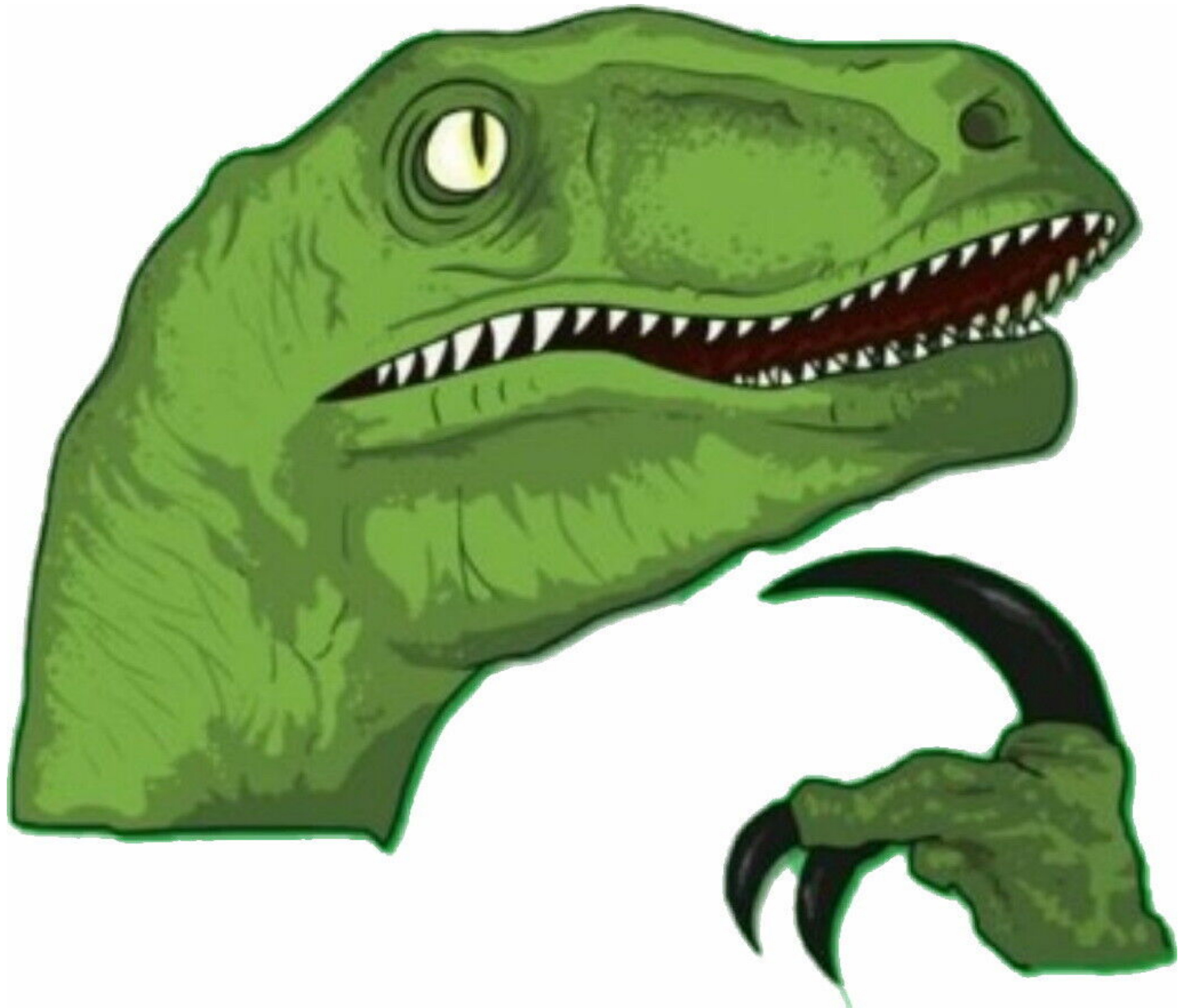
Post-pretraining



Fundamental
AI Lab

Post-pretraining

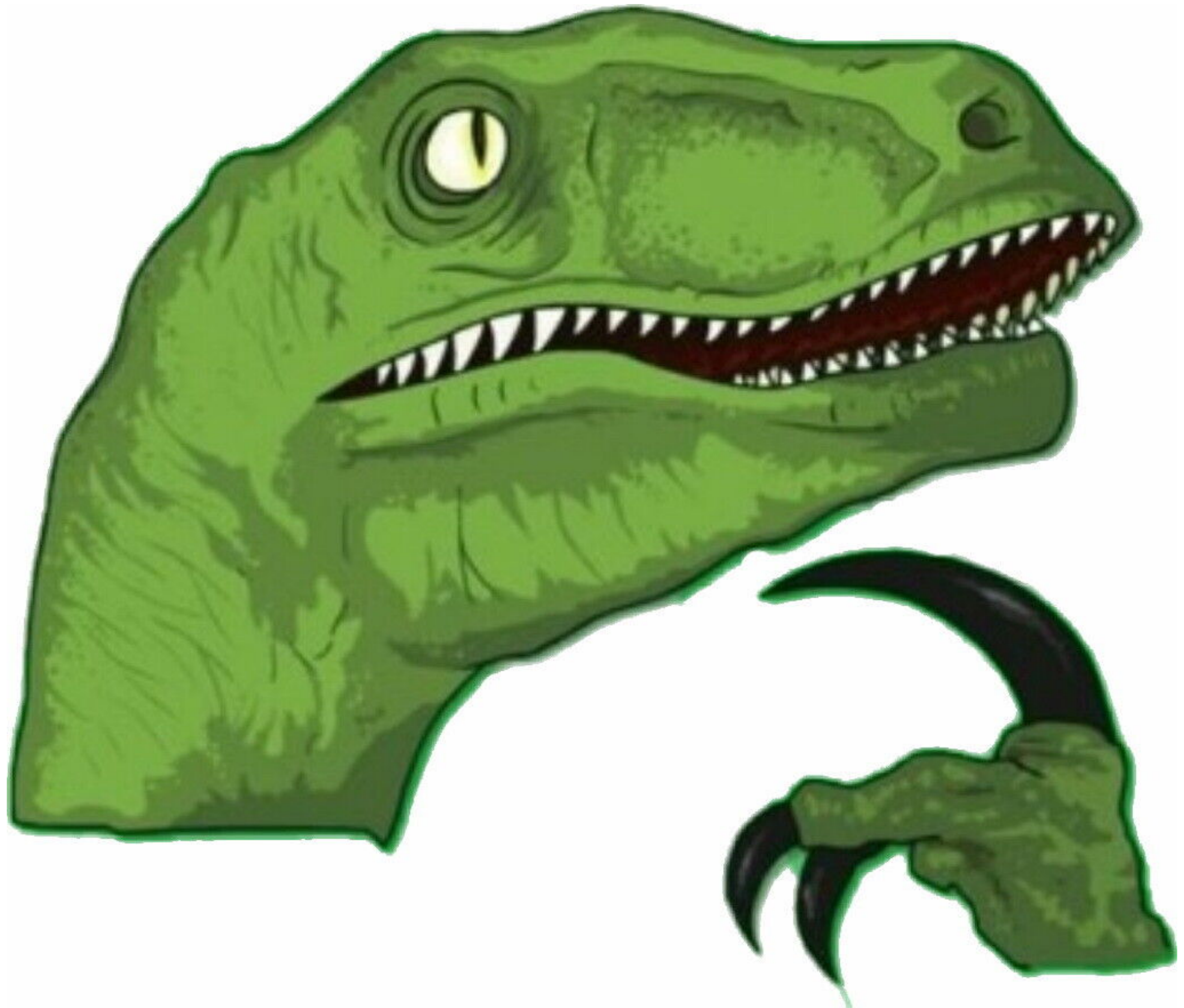
Question 1: What losses should I use?



Post-pretraining

Question 2: What kind of data?

Question 1: What losses should I use?



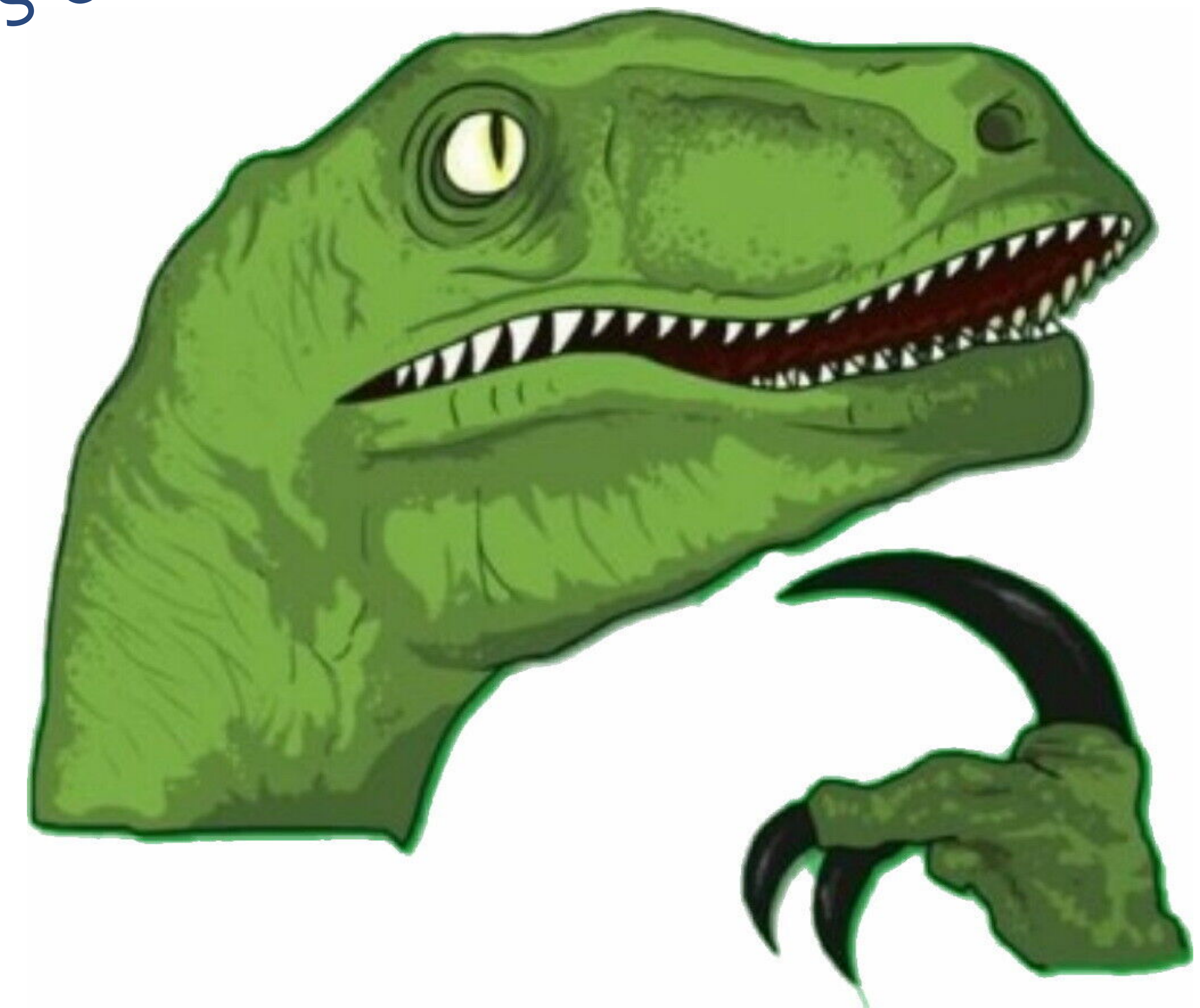
Fundamental AI Lab

Post-pretraining

Question 2: What kind of data?

Question 1: What losses should I use?

Question 3: How can I be efficient in terms of number of adapter parameters?



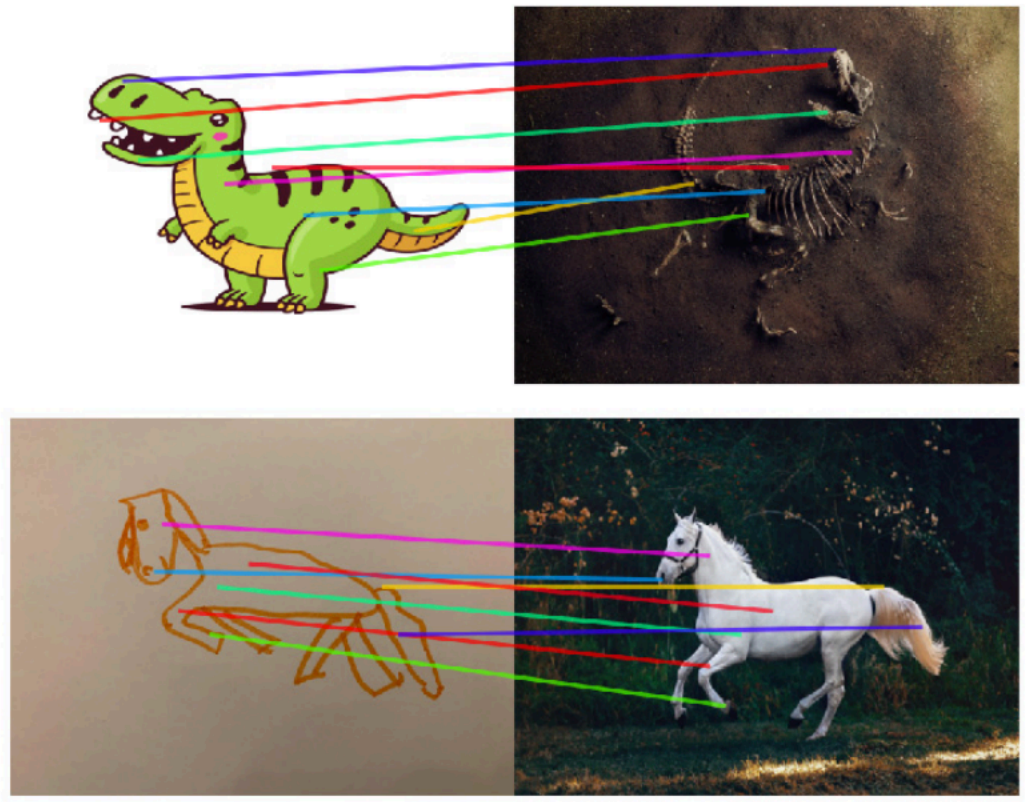
Fundamental AI Lab

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

How semantic are patch representations?

Qualitative results in DINOv2

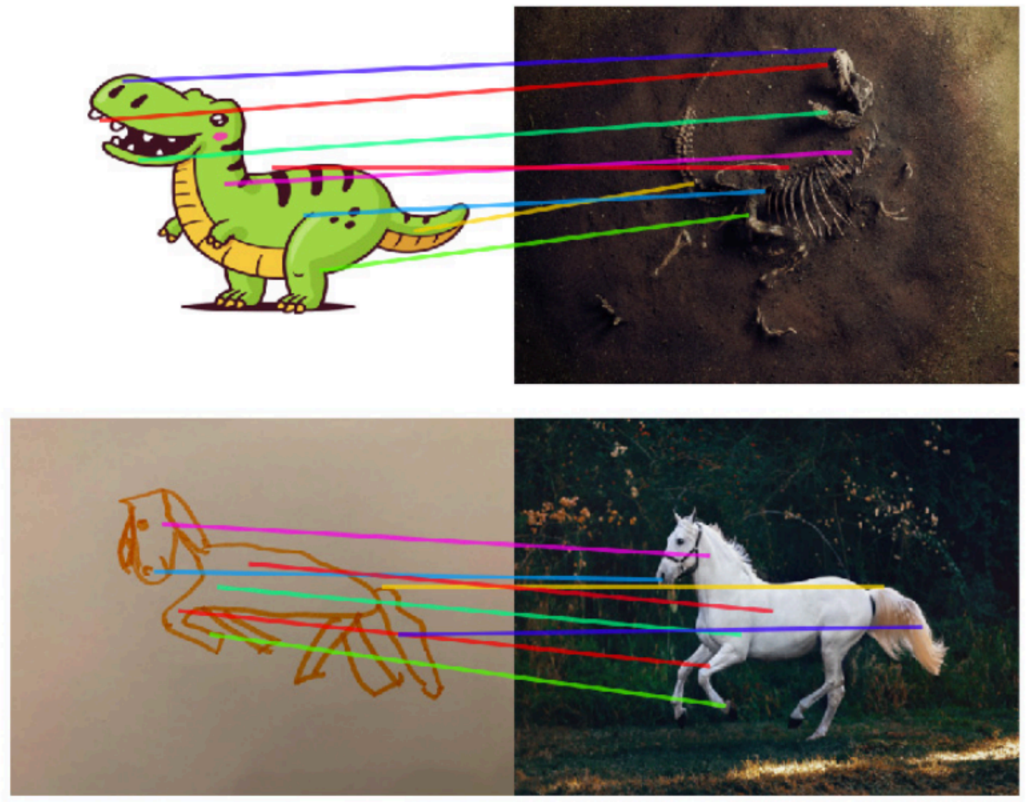


(Drawings / Animals)

Oquab et al. DINOv2: Learning Robust Visual Features without Supervision. TMLR 2023
Darcet et al. Vision Transformers Need Registers. ICLR 2024

How semantic are patch representations?

Qualitative results in DINOv2

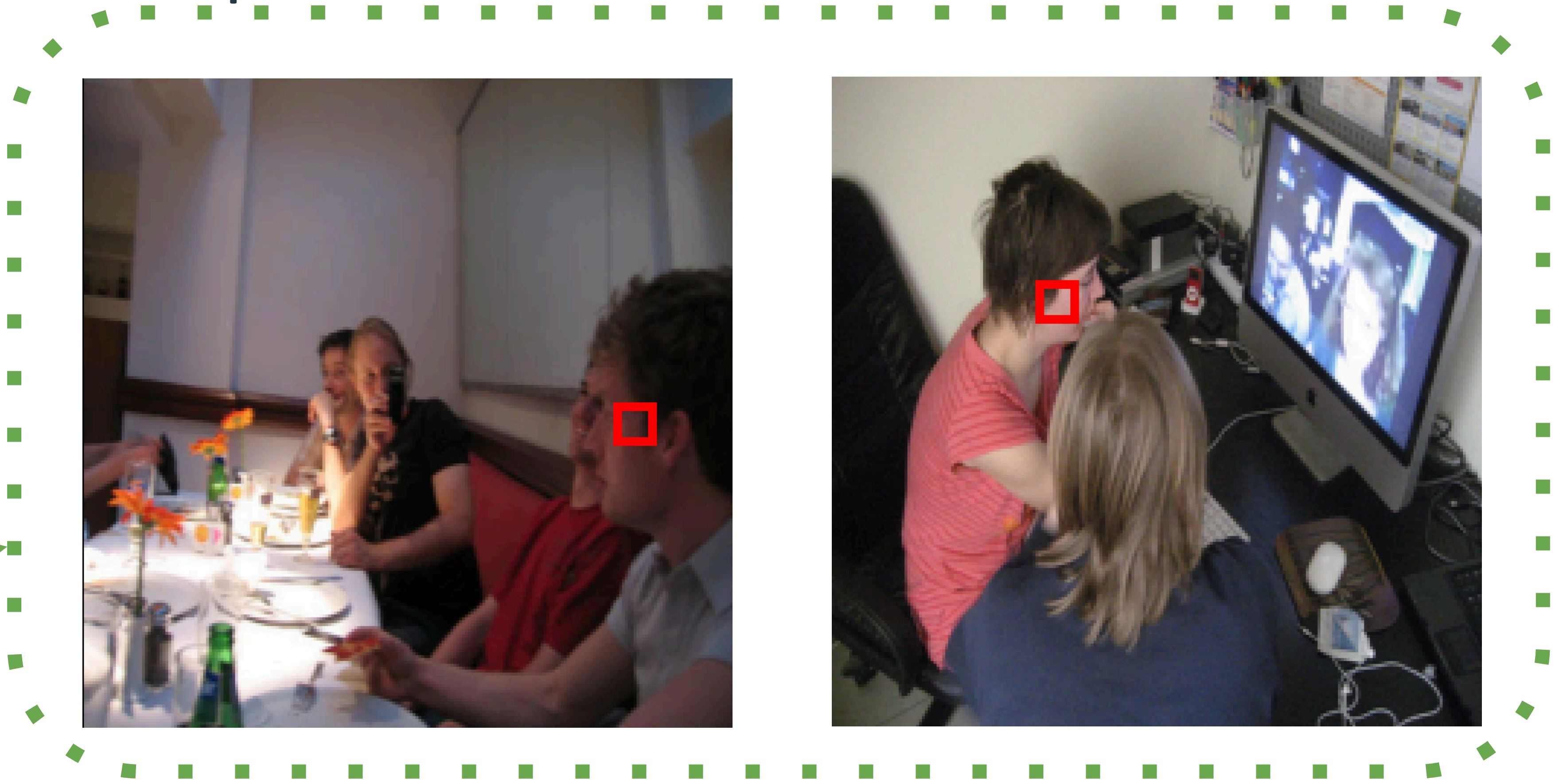


(Drawings / Animals)

But often...

How it should be

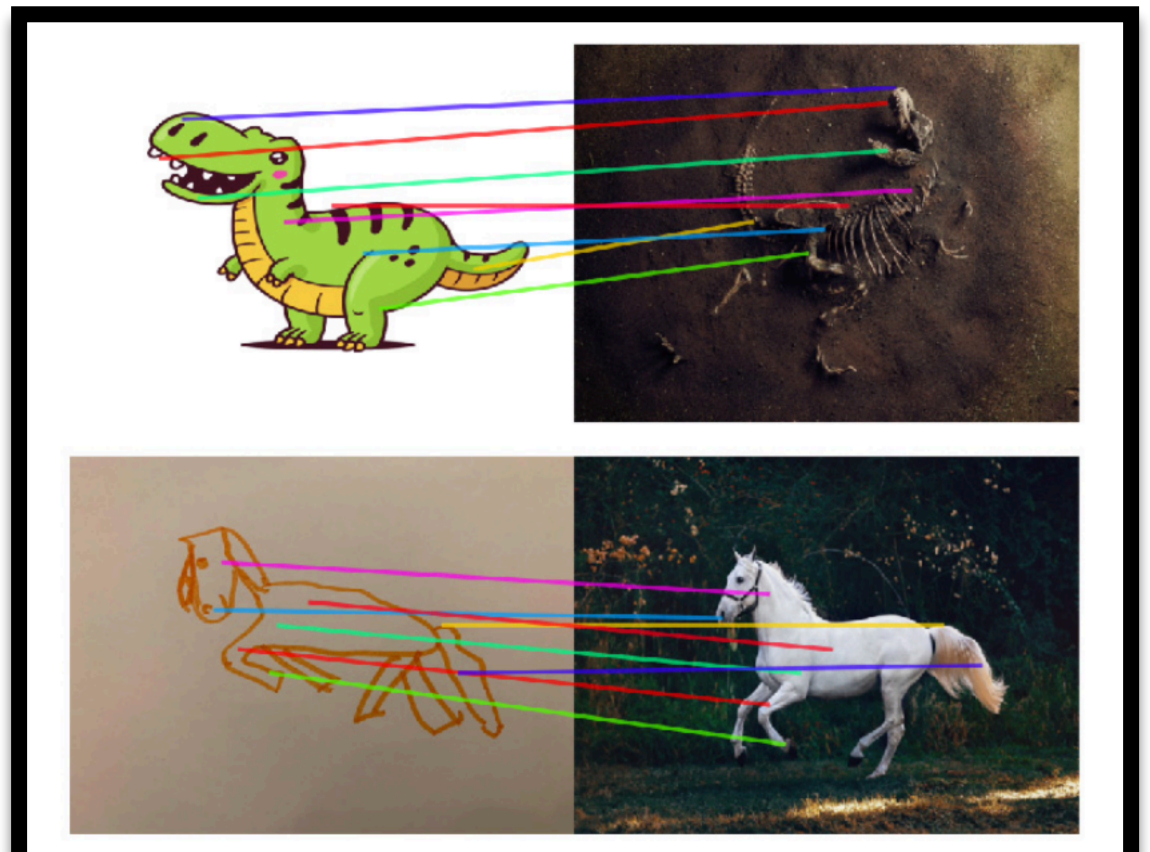
Which patch from the whole dataset is the closest?



Oquab et al. DINOv2: Learning Robust Visual Features without Supervision. TMLR 2023
Darcet et al. Vision Transformers Need Registers. ICLR 2024

How semantic are patch representations?

Qualitative results in DINOv2



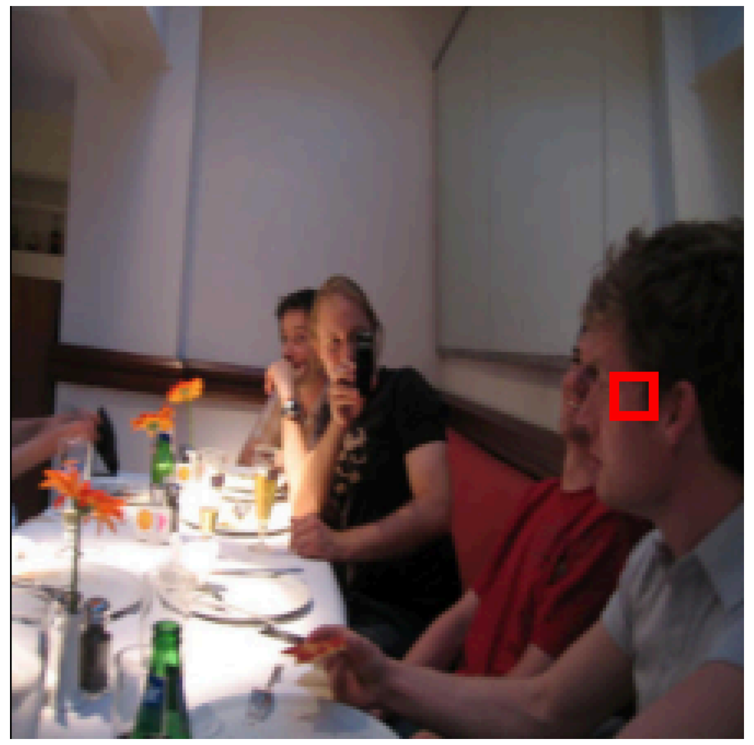
(Drawings / Animals)

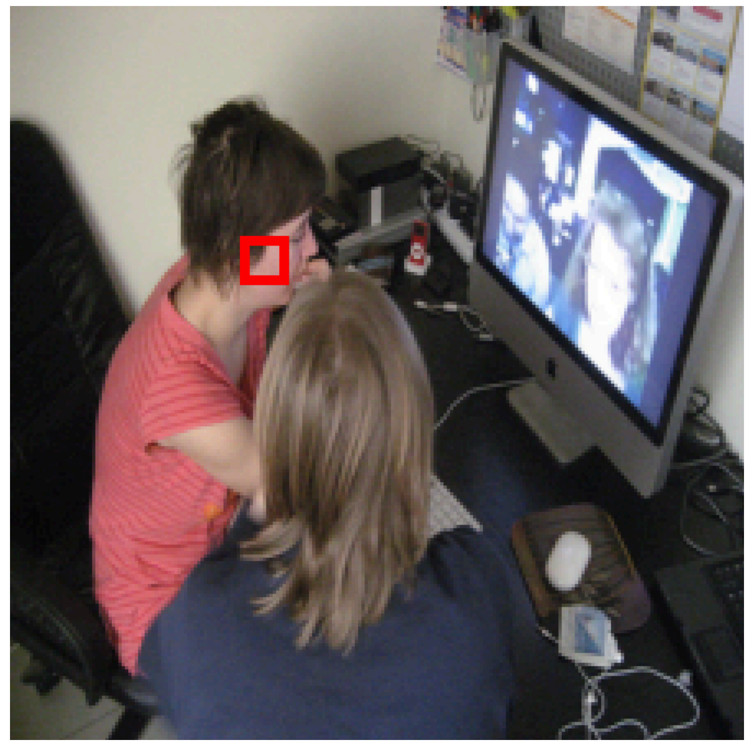
But often...

How it should be

How it is

Which patch from the whole dataset is the closest?





with SoTA DINOv2-R model

Fundamental
AI Lab

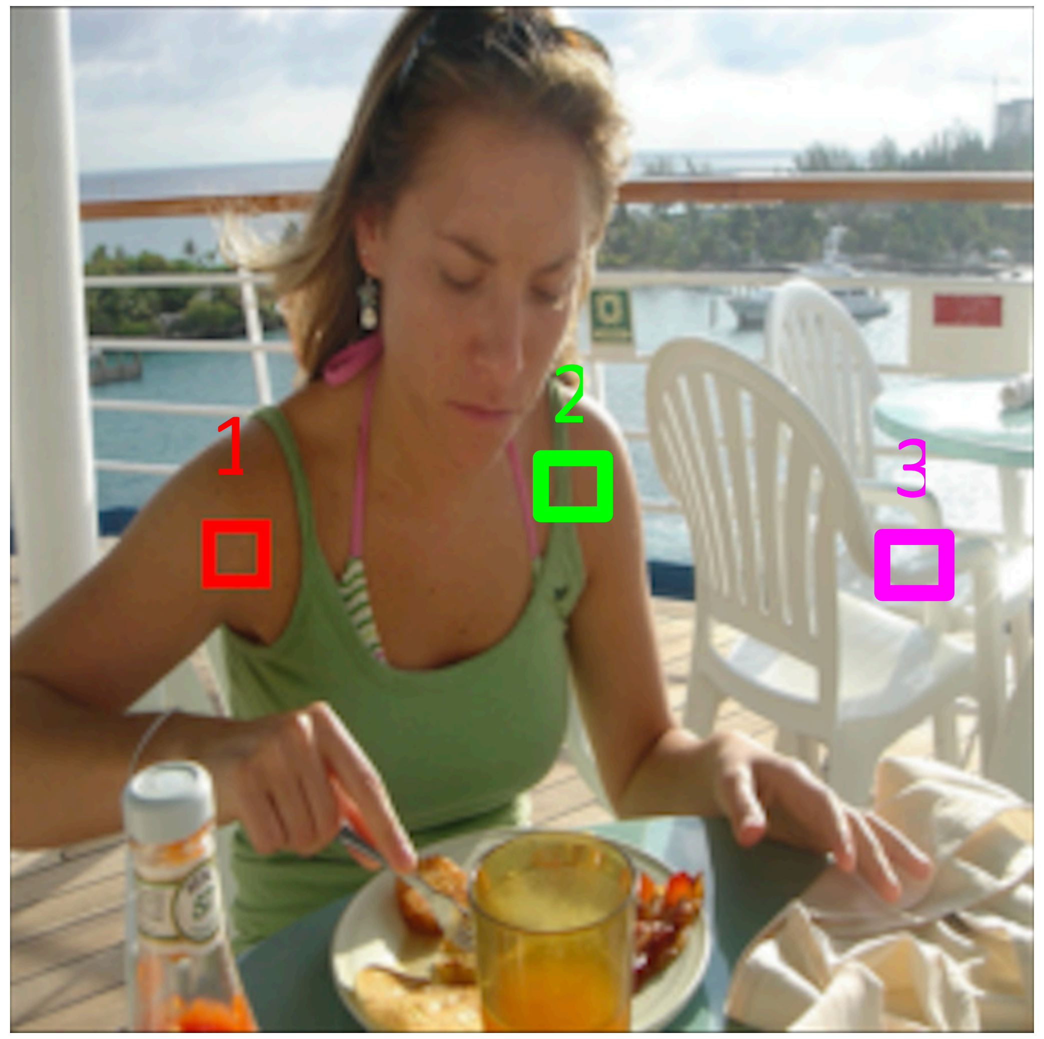
Oquab et al. DINOv2: Learning Robust Visual Features without Supervision. TMLR 2023
Darcet et al. Vision Transformers Need Registers. ICLR 2024

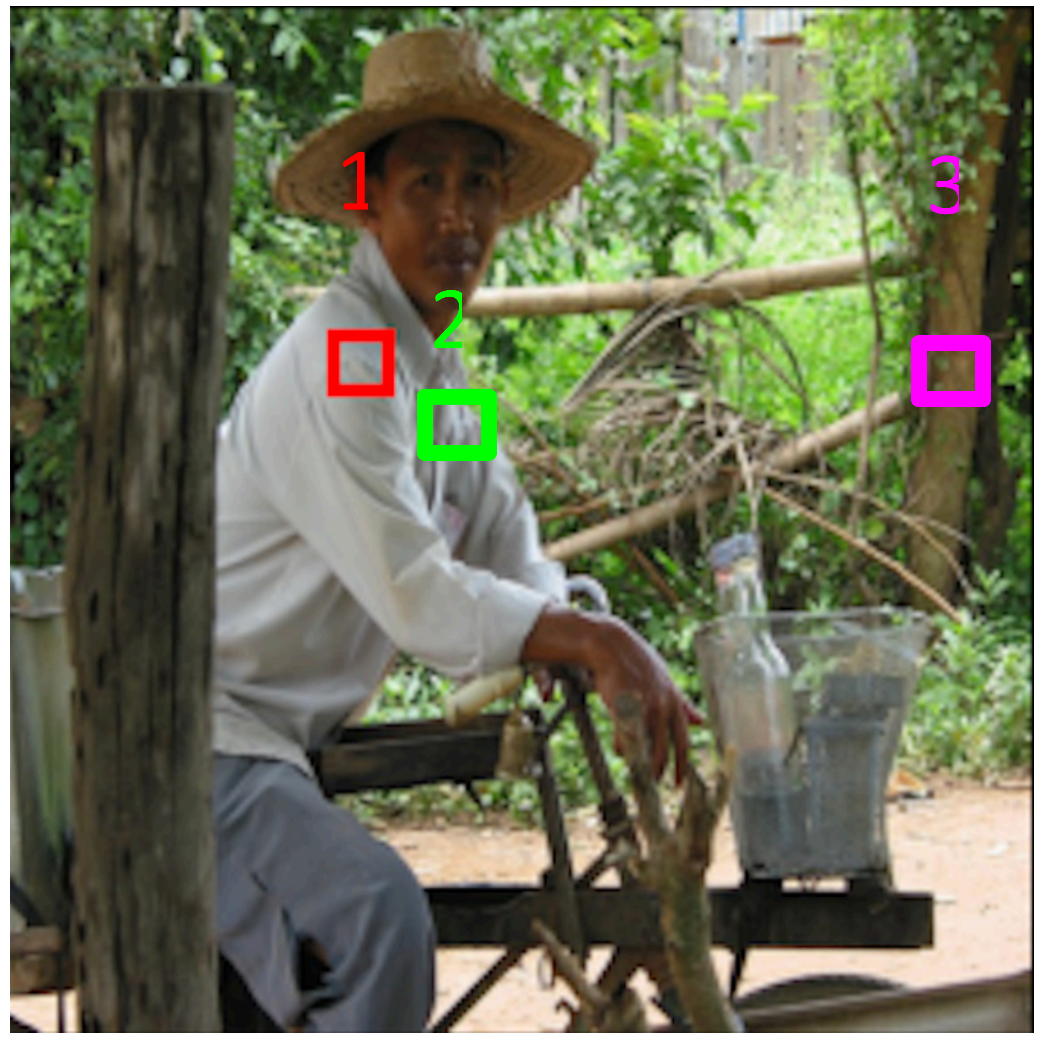
Idea of Patch Nearest Neighbor Consistency: intuitive to us

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

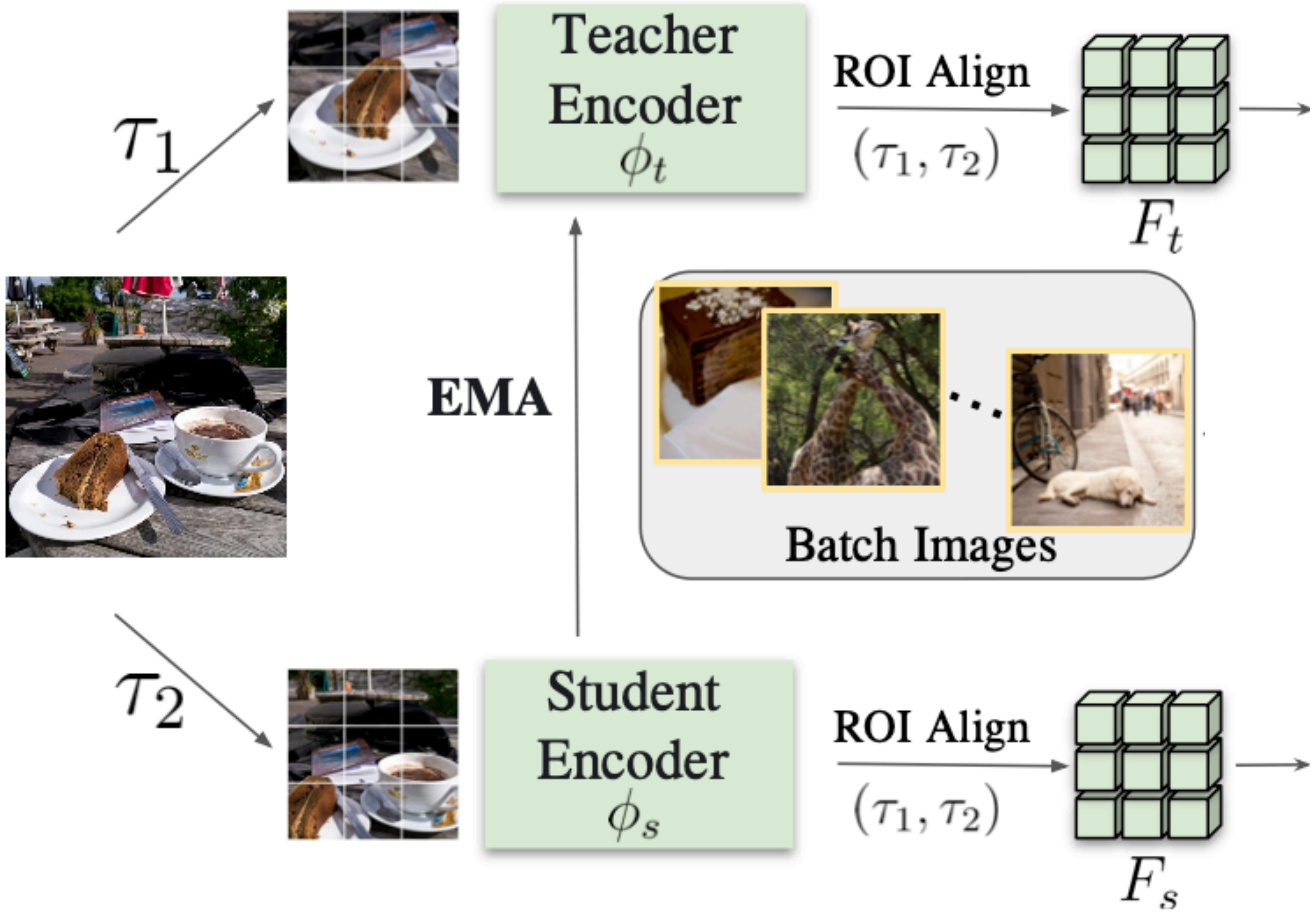




Example Patches

Fundamental AI Lab

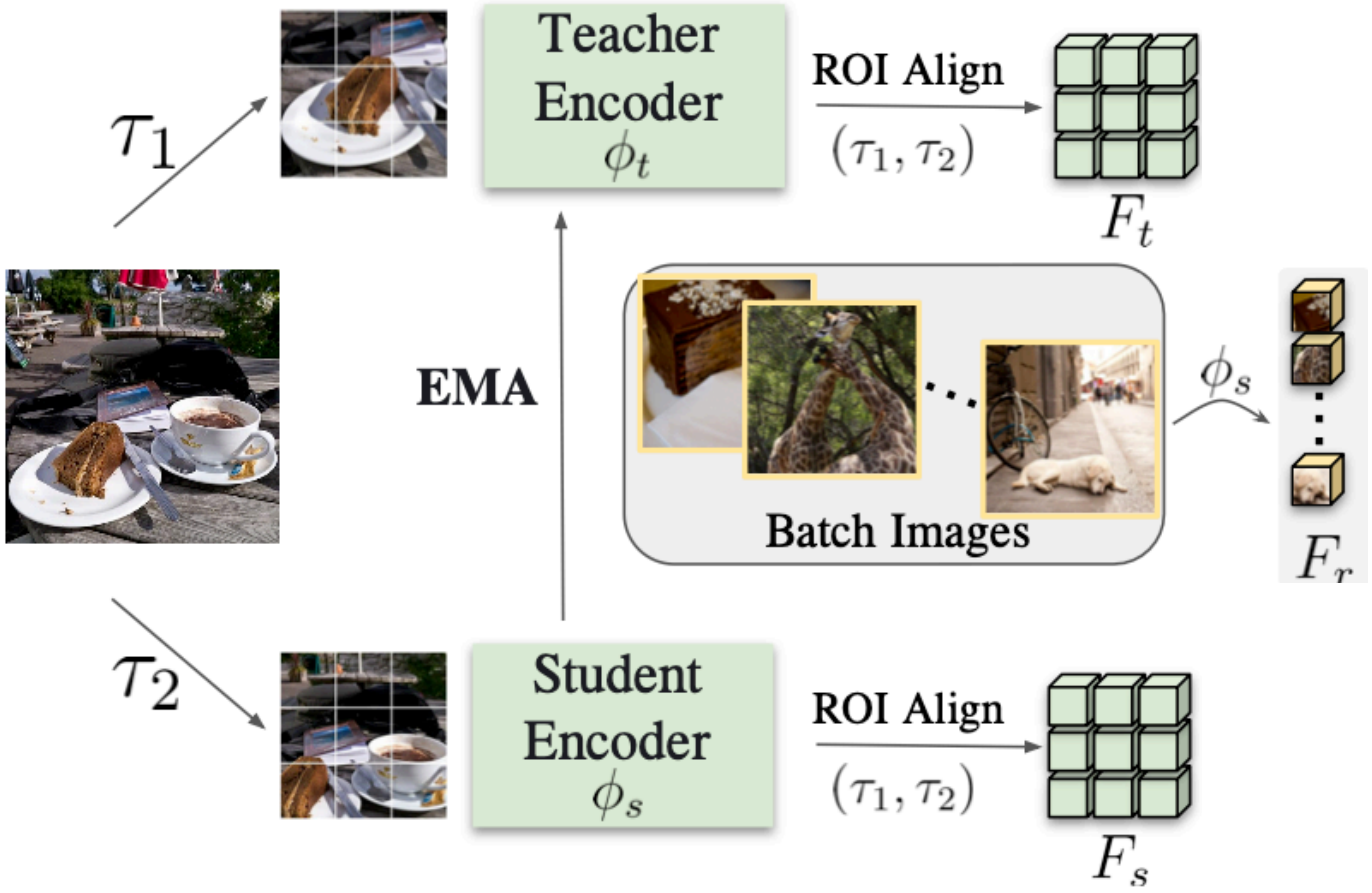
PaNeCo: Patch Nearest neighbor Consistency



Fundamental AI Lab

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

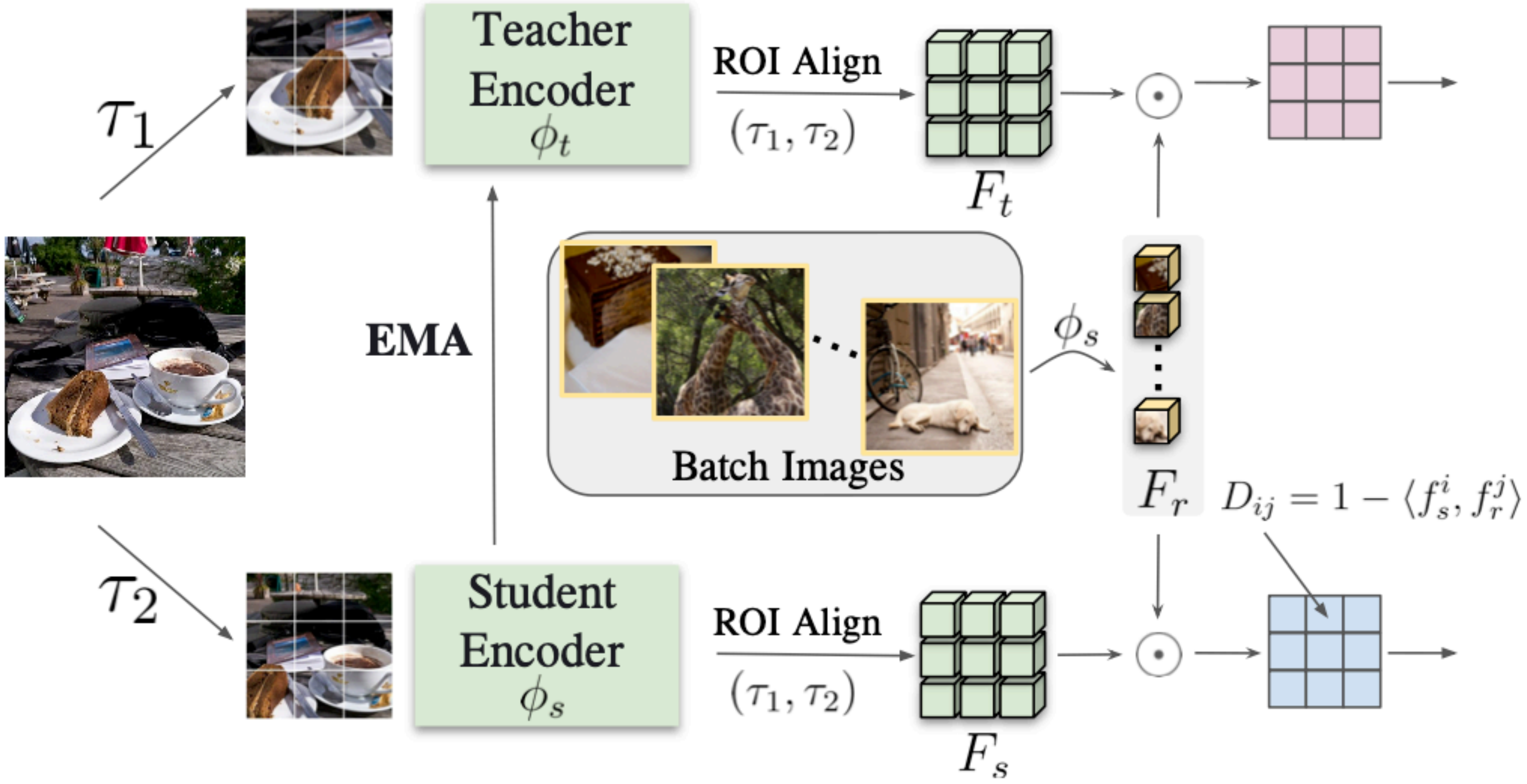
PaNeCo: Patch Nearest neighbor Consistency



Fundamental AI Lab

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

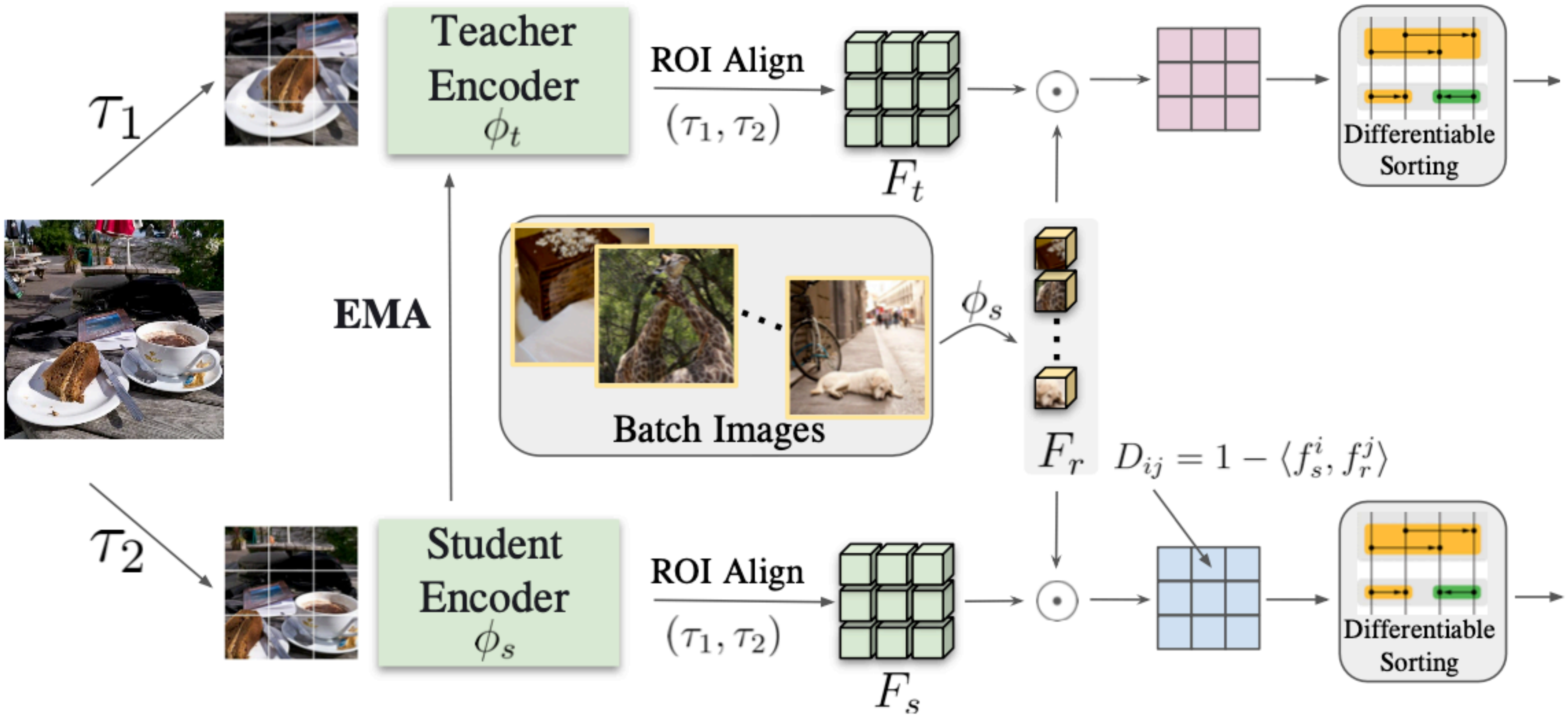
PaNeCo: Patch Nearest neighbor Consistency



Fundamental AI Lab

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

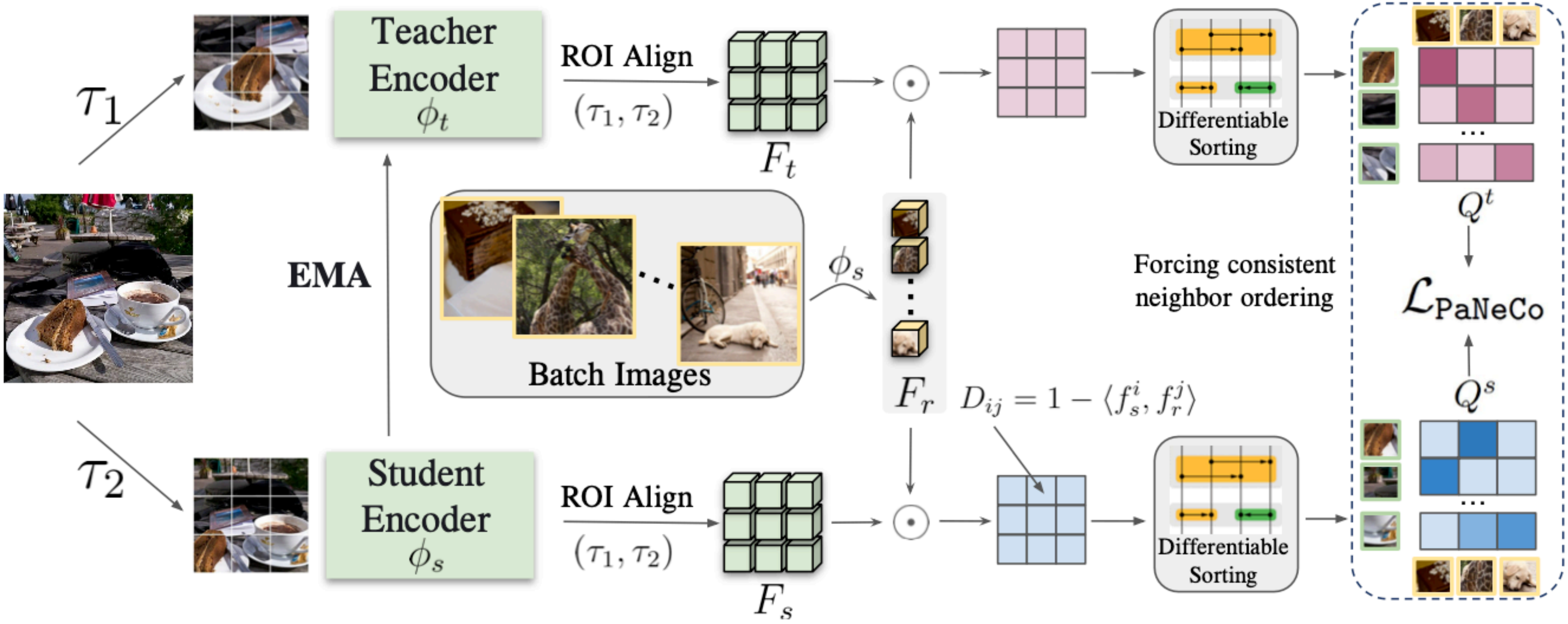
PaNeCo: Patch Nearest neighbor Consistency



Fundamental AI Lab

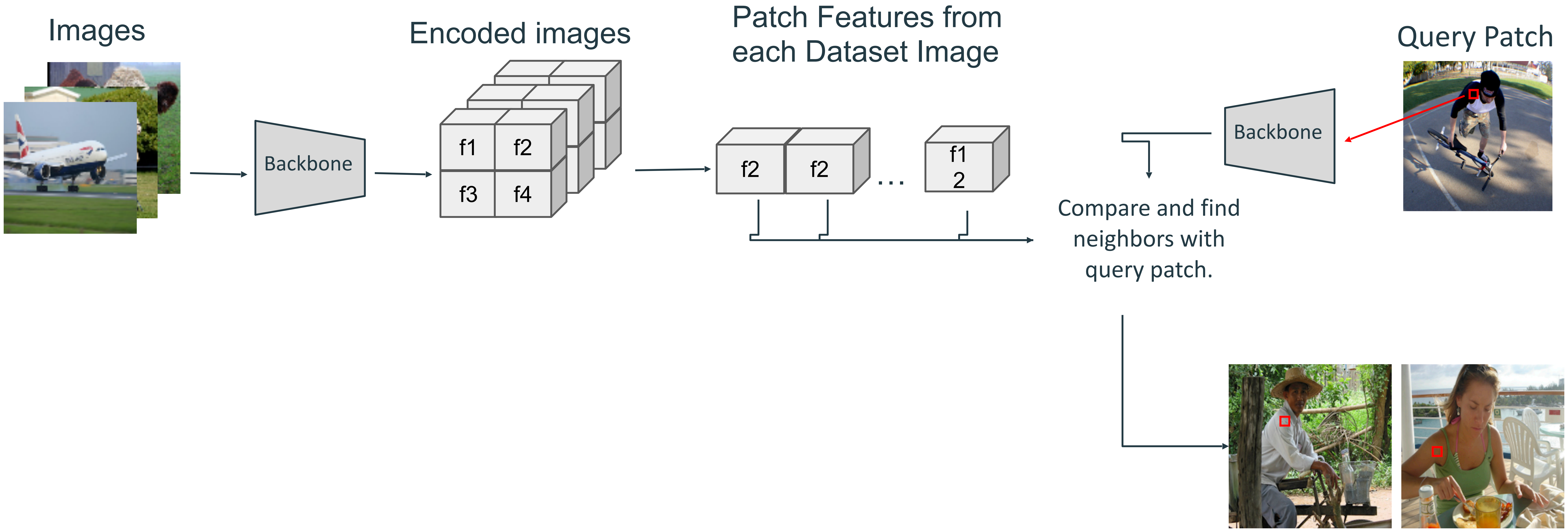
NeCo: Improving DINOv2's spatial representations in 19 GPU hours with Patch Neighbor Consistency. Pariza, Salehi, Burghouts, Locatello, Asano. arxiv 2024

PaNeCo: Patch Nearest neighbor Consistency



Results

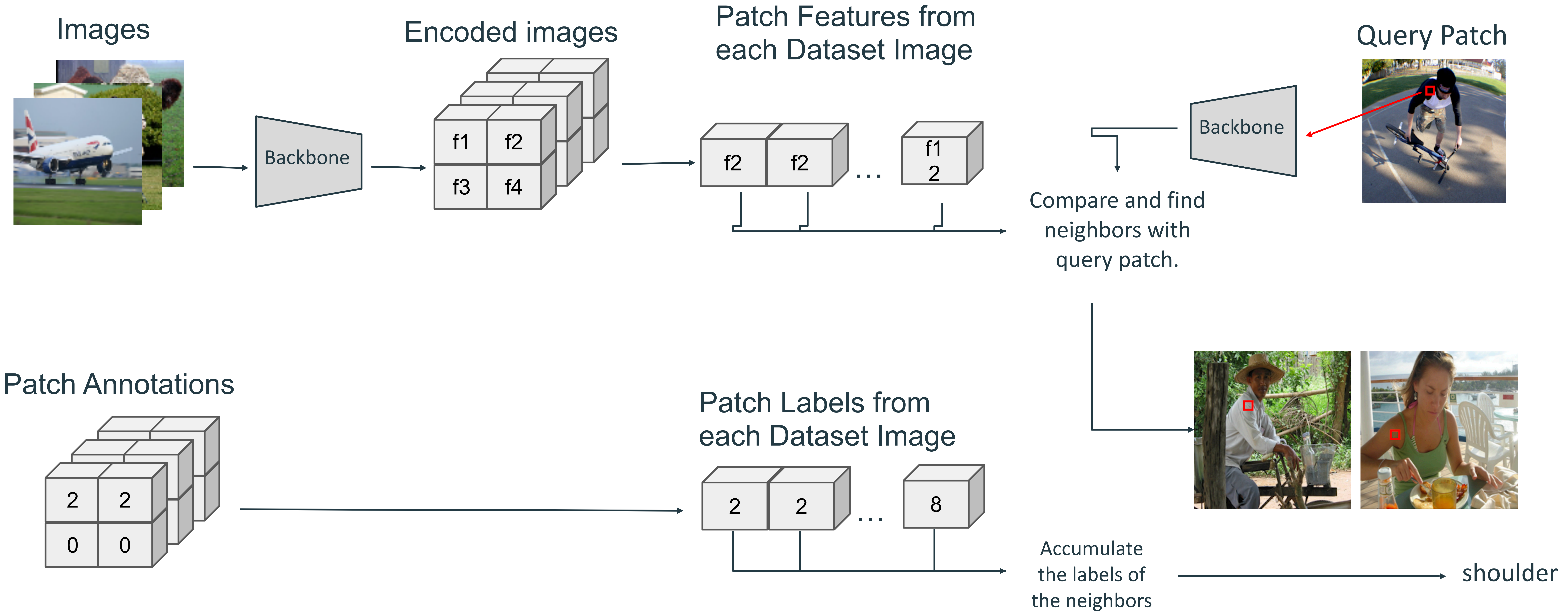
Evaluation 1: Visual in-context segmentation via dense NN retrieval



Towards In-context Scene Understanding. Ivana Balažević, David Steiner, Nikhil Parthasarathy, Relja Arandjelović, Olivier J. Hénaff. NeurIPS 2023

Fundamental AI Lab

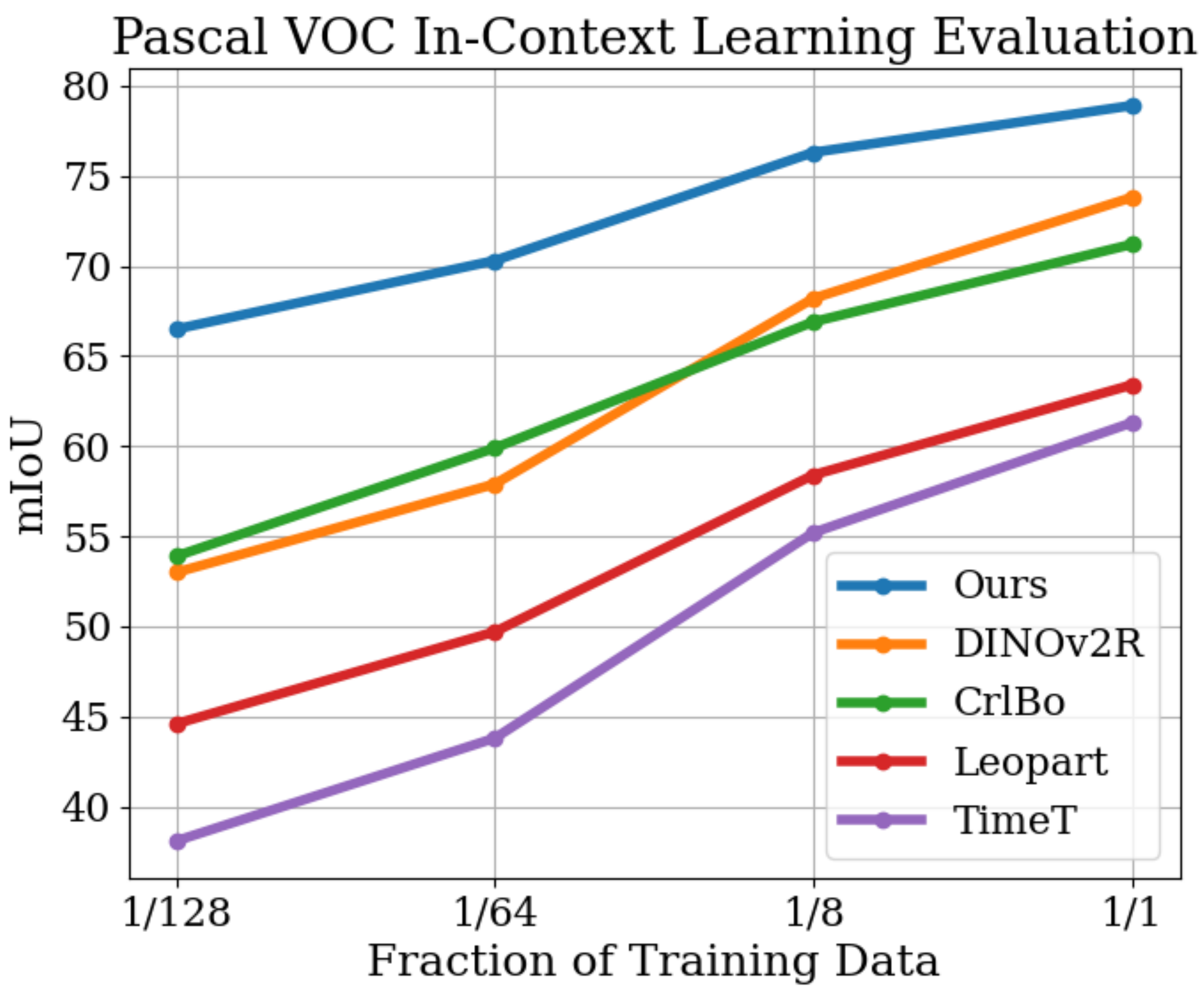
Evaluation 1: Visual in-context segmentation via dense NN retrieval



Towards In-context Scene Understanding. Ivana Balažević, David Steiner, Nikhil Parthasarathy, Relja Arandjelović, Olivier J. Hénaff. NeurIPS 2023

Fundamental AI Lab

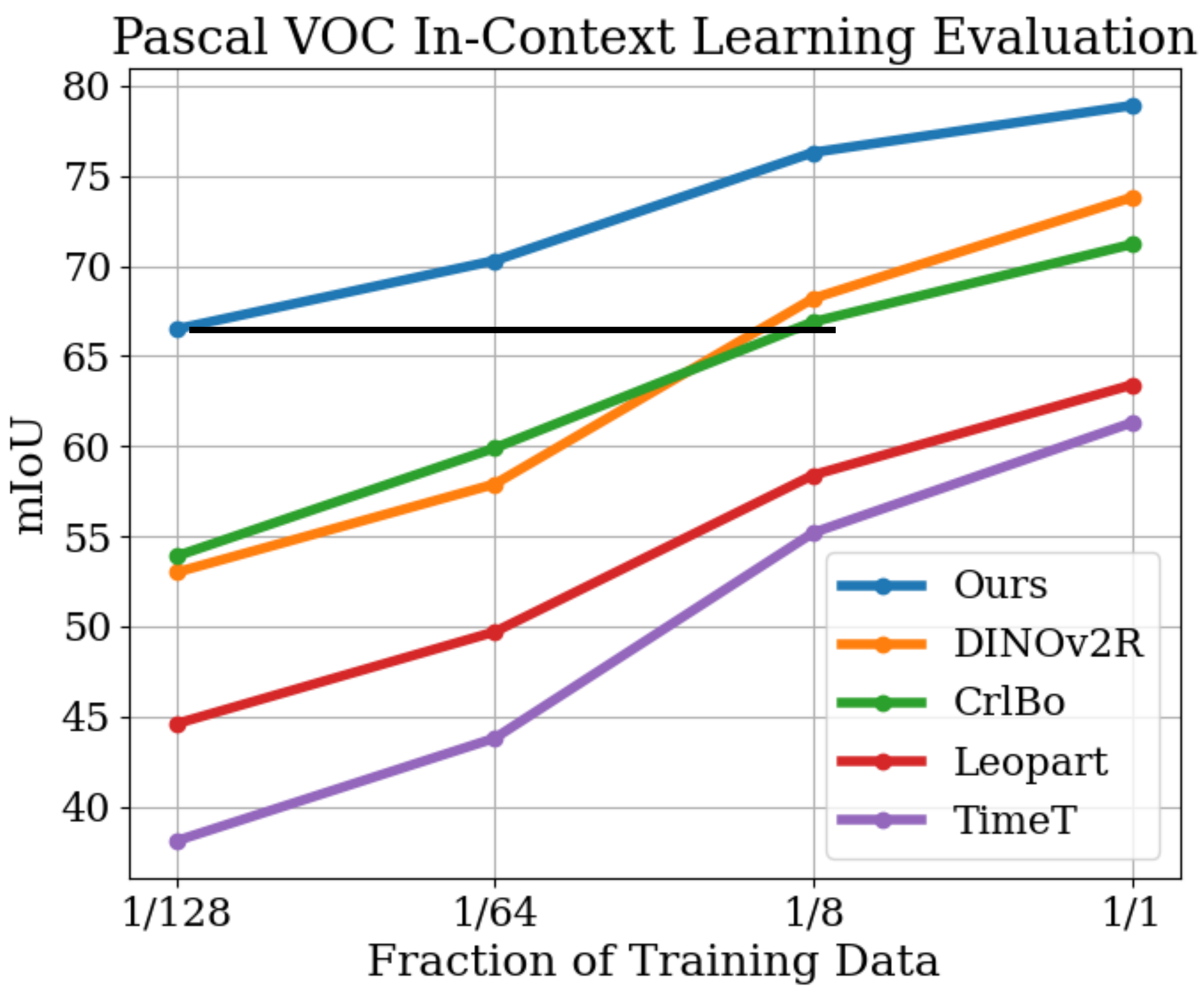
In-context scene understanding benchmark



Fundamental AI Lab

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

In-context scene understanding benchmark

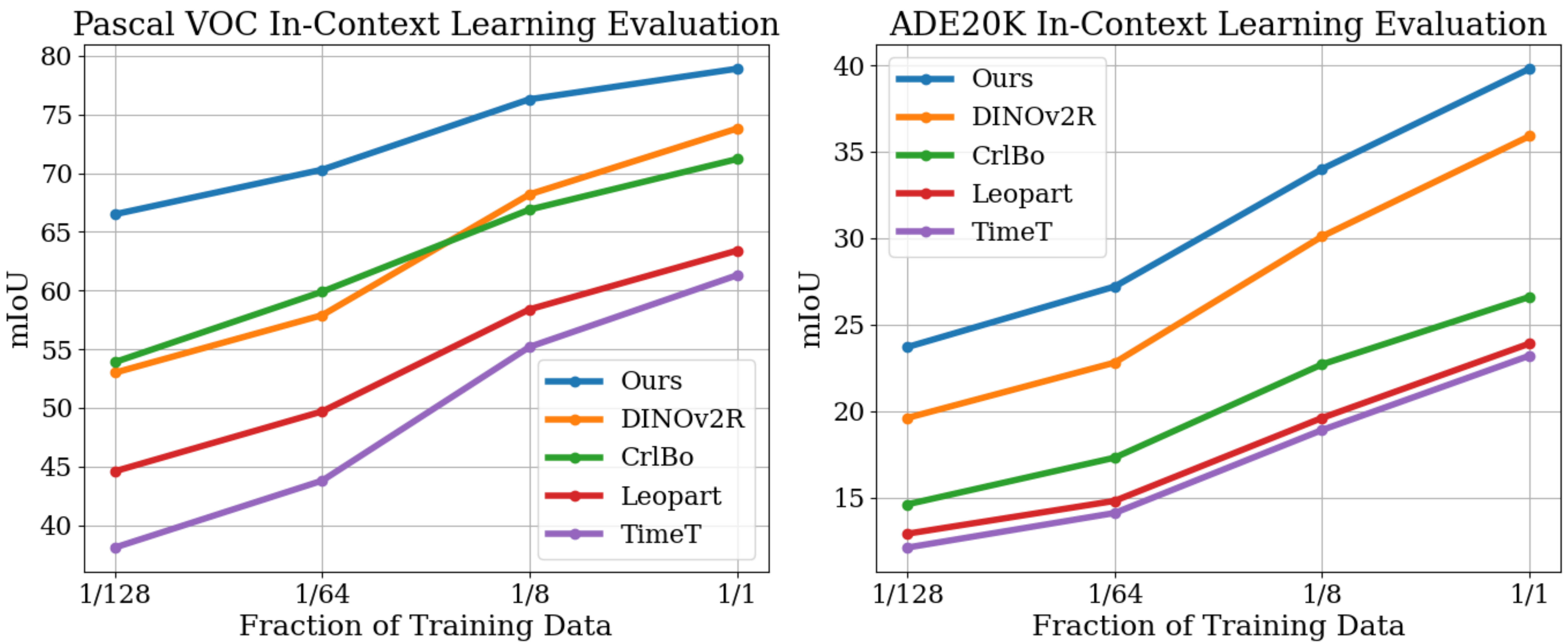


matches performances of DINOv2-R with ~15x less data

Fundamental AI Lab

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

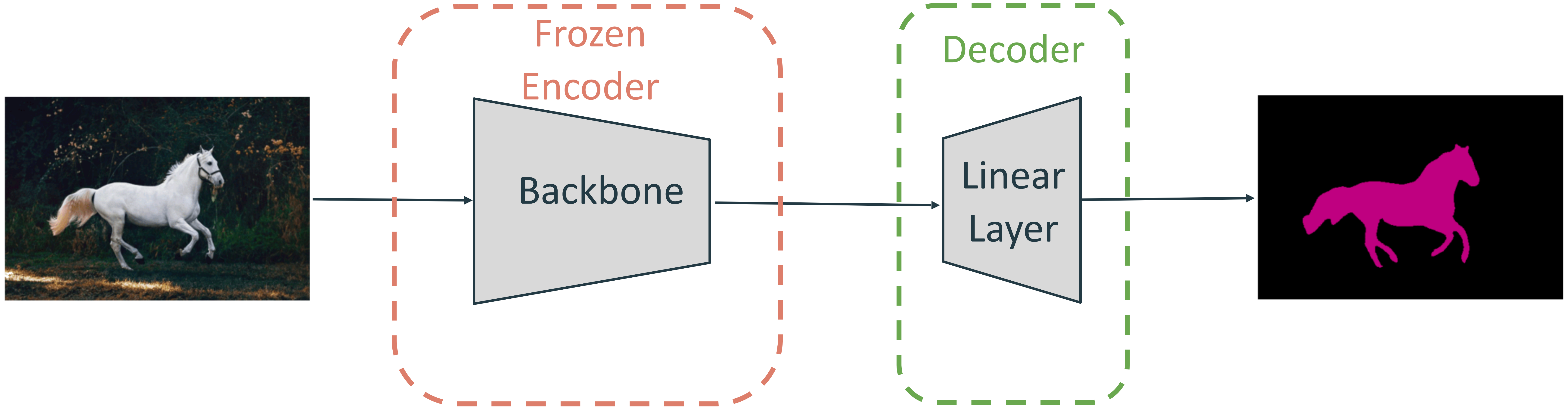
In-context scene understanding benchmark



Fundamental AI Lab

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

Evaluation #2: Linear Segmentation



- 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.

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

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	**82.3**	**62.0**	**81.3**	**40.1**
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	**85.5**	**63.3**	**83.3**	**44.9**

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.

Eval #3: Fully unsupervised semantic segmentation

	mIoU
DINOv2R	12.2
+ PaNeCo	17.8 (+5.6%)
+ CBFE	41.3 (+23.3%)
+ CD	55.1 (+13.8%)

<u>Semantic Segmentation on Pascal VOC for 21 clusters</u>

k-Means Overclustering

Cluster-based Foreground Extraction (CBFE)

Community Detection (CD)

Method	mIoU
MaskConstrast [74]	35.1
DINOv2R [58]	35.1
DeepSpectral [54]	37.2
DINOSAUR [67]	37.2
Leopart [93]	41.7
COMUS [87]	50.0
PaNeCo	**55.1**

Other State of the Art Semantic Segmentation Performances for 21 clusters as the 21 target semantic labels in the dataset.

Fundament
AI Lab

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 $^{\uparrow 8.8}$

Fundamental AI Lab

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	$16.9^{\uparrow 11.5}$	$50.0^{\uparrow 30.8}$	$62.4^{\uparrow 18.5}$

Fundamental AI Lab

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.

→ PaNeCo 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

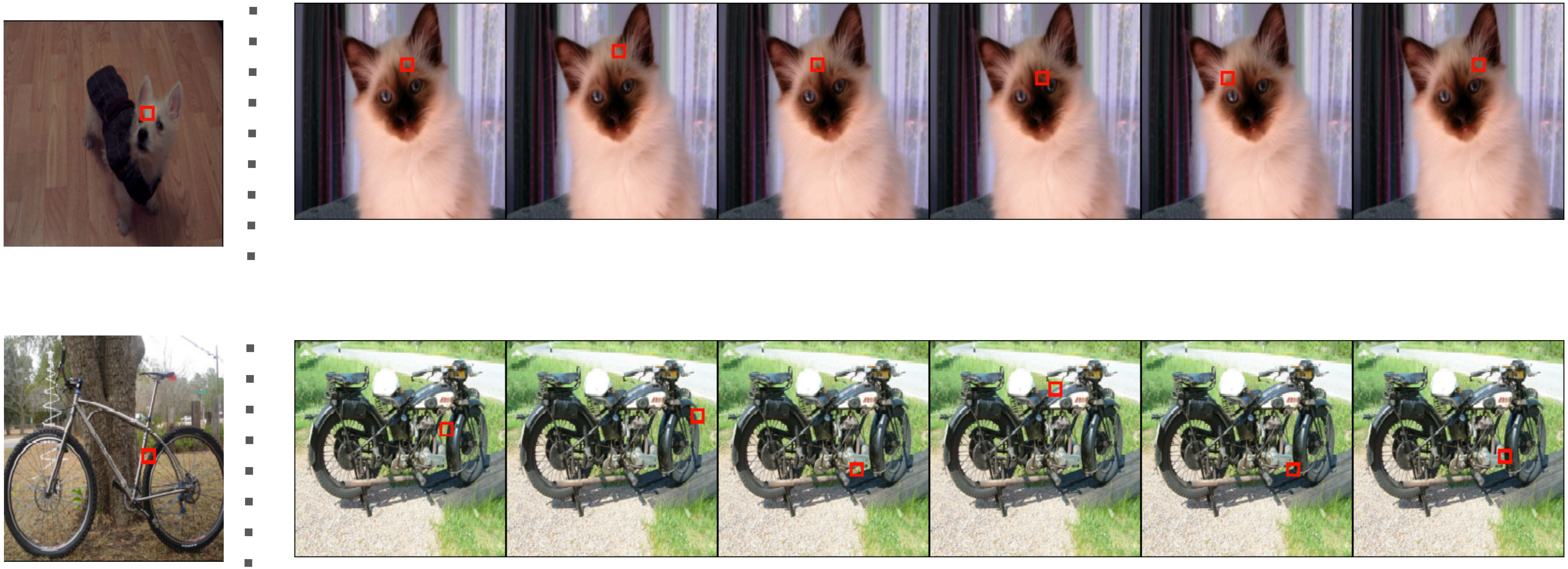
Qualitative Results

Nearest Neighbors of Patches from representations

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

Fundamental AI Lab

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

Key takeaways

Key takeaways

- **Dense Patch-ordering** is loss well suited for post-pretraining

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

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

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

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

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

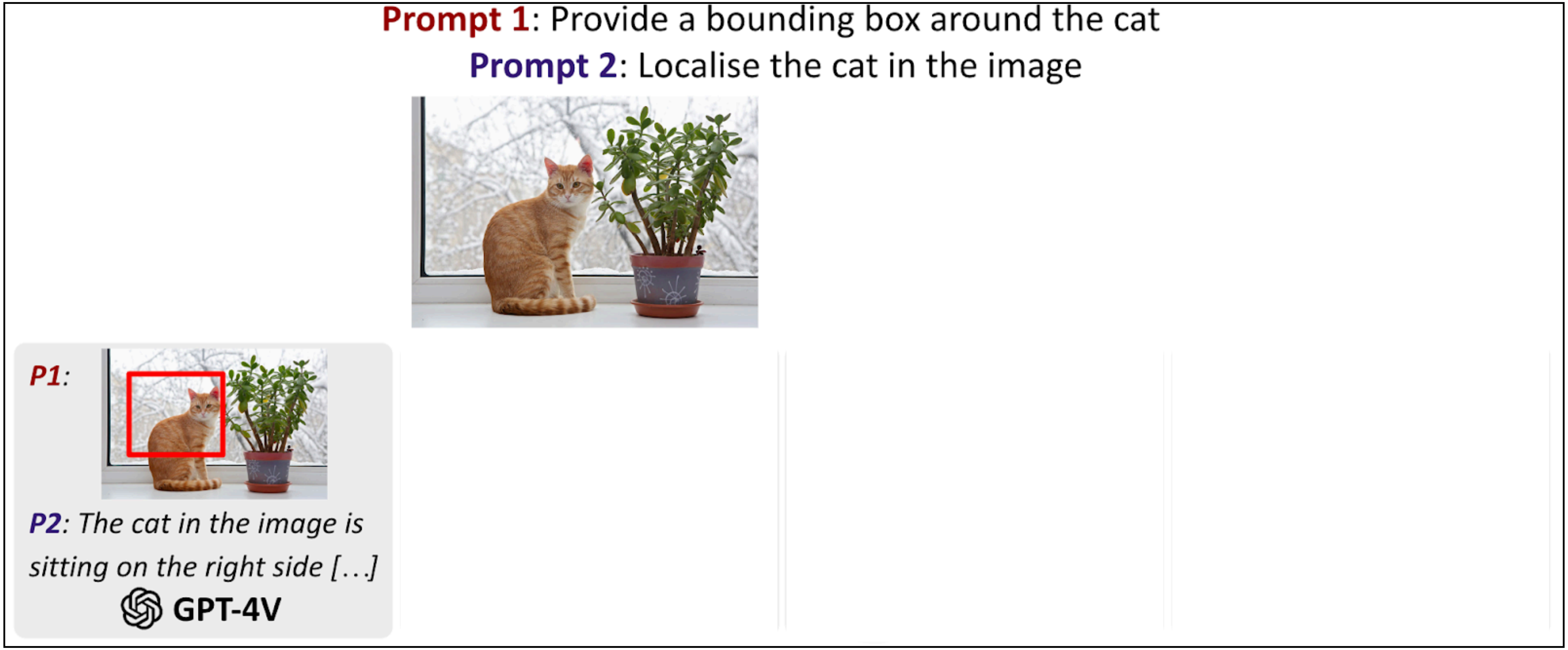
Prompt 1: Provide a bounding box around the cat

Prompt 2: Localise the cat in the image

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

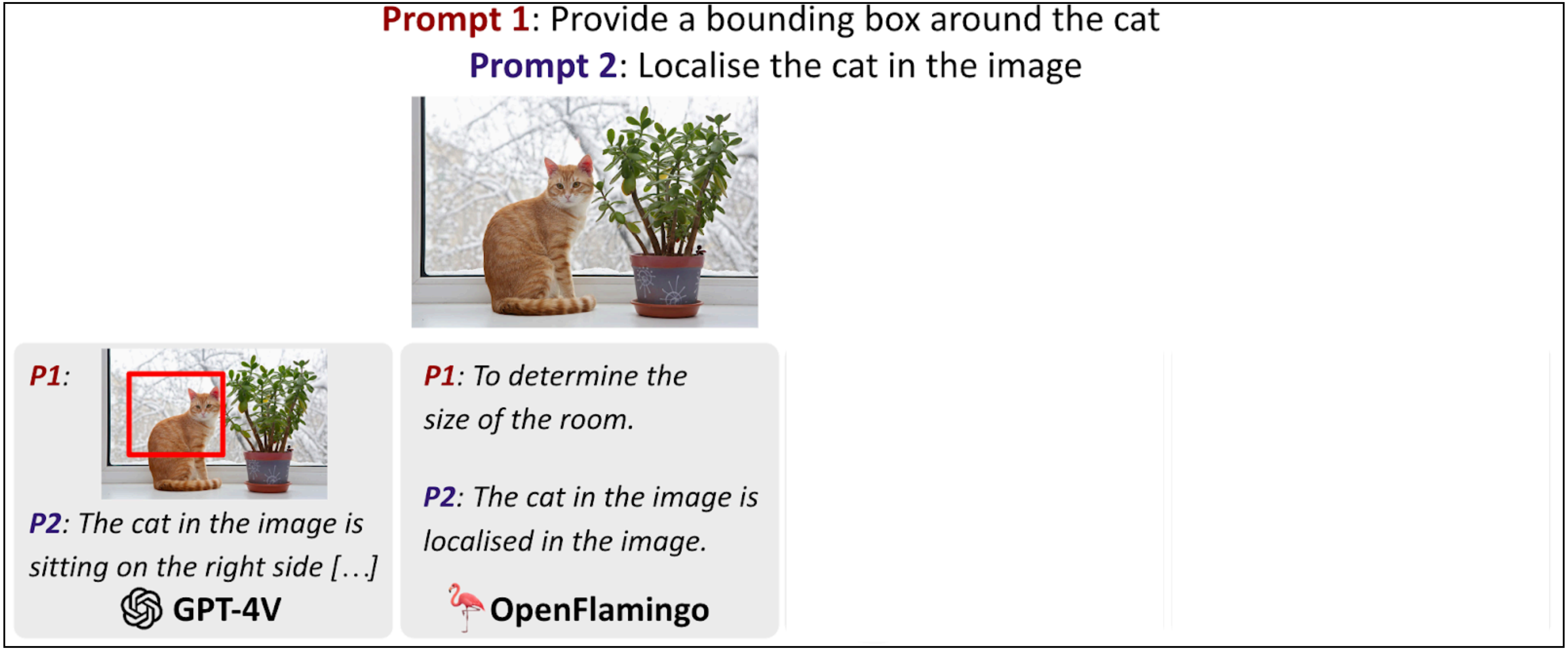
Fundamental AI Lab

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



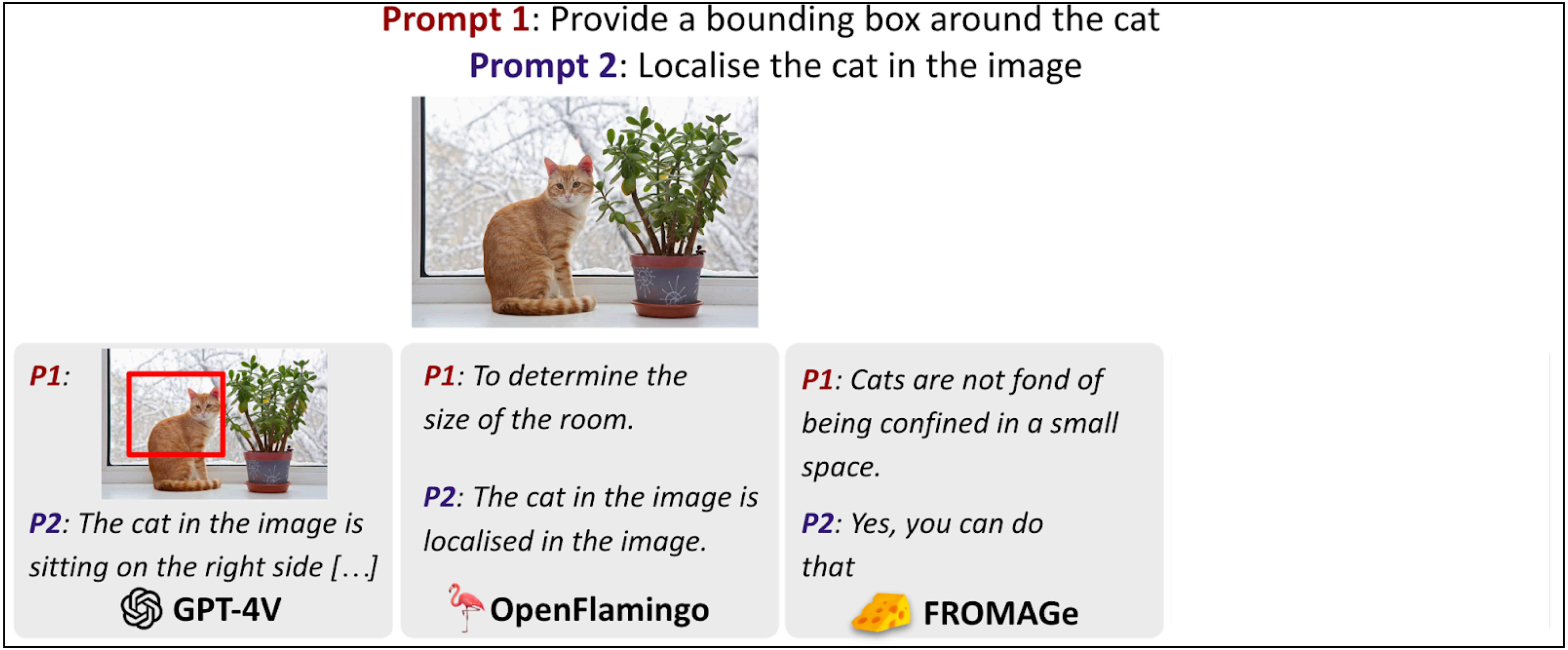
Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

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



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

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



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

P1:

***P2**: The cat in the image is sitting on the right side [...]*

GPT-4V

***P1**: To determine the size of the room.*

***P2**: The cat in the image is localised in the image.*

OpenFlamingo

***P1**: Cats are not fond of being confined in a small space.*

***P2**: Yes, you can do that*

FROMAGe

***P1**: Provide a bounding box around the cat's plant*

***P2**: <empty string>*

BLIP-2

Fundamental AI Lab

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

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

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**

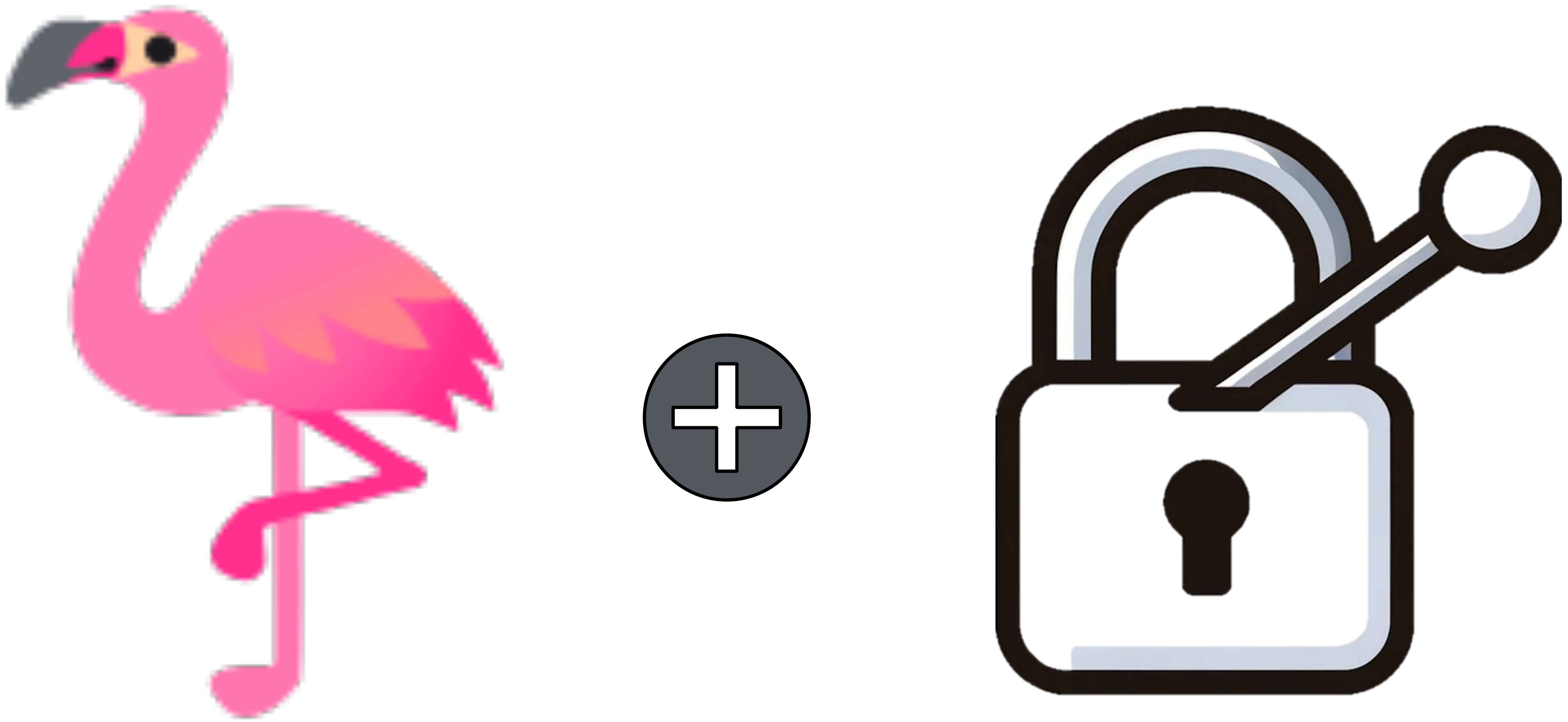
Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Our approach

frozen VLM, e.g. Flamingo

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

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

Our approach

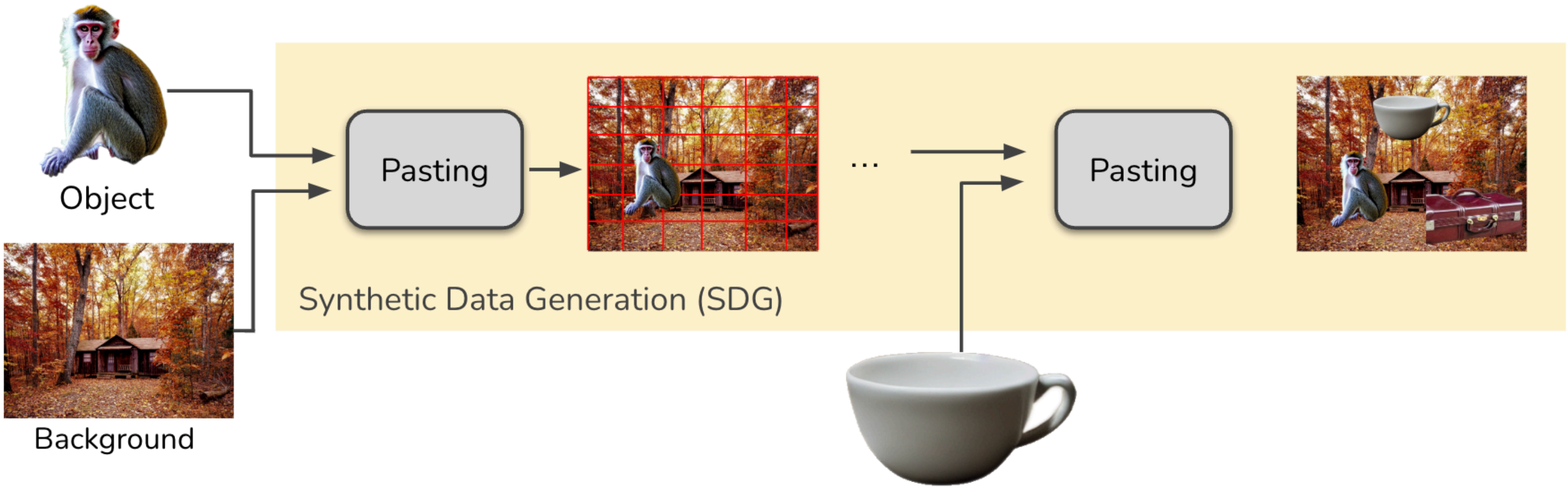
frozen VLM, e.g. Flamingo

Positional Insert (PIN) module

Synthetic, unlabeled data

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

The data



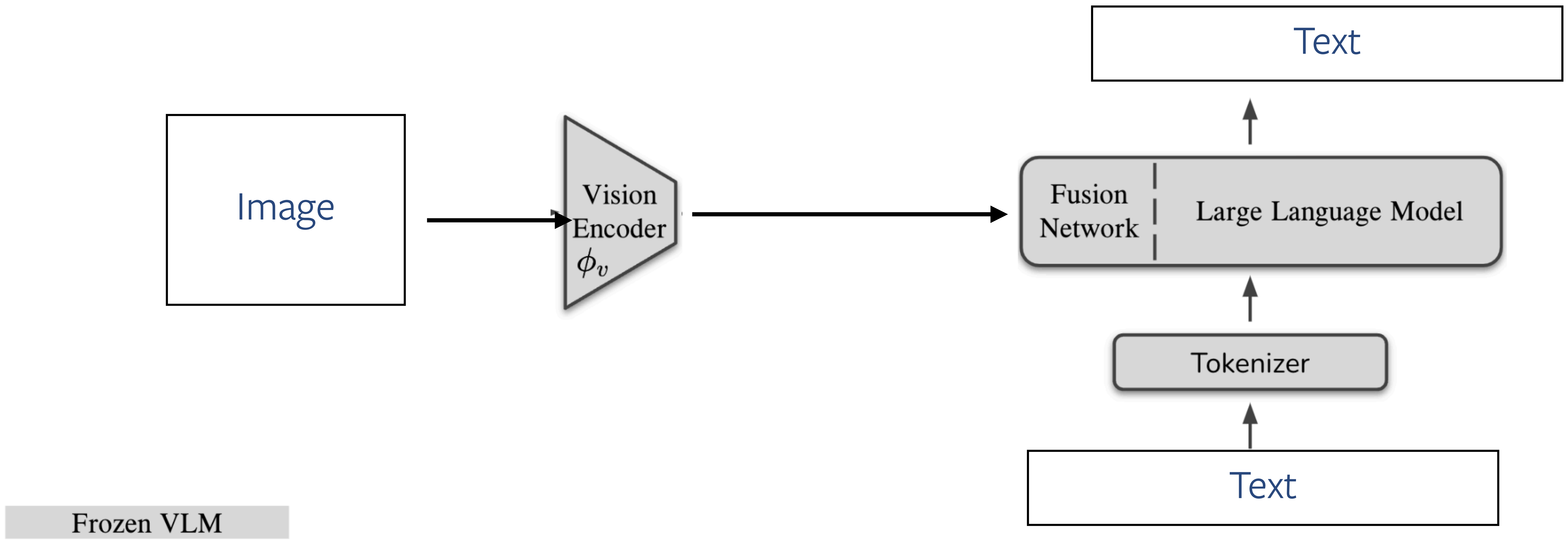
Zhao et al. X-Paste: Revisiting Scalable Copy-Paste for Instance Segmentation using CLIP and StableDiffusion. ICML 2023

Example generated data

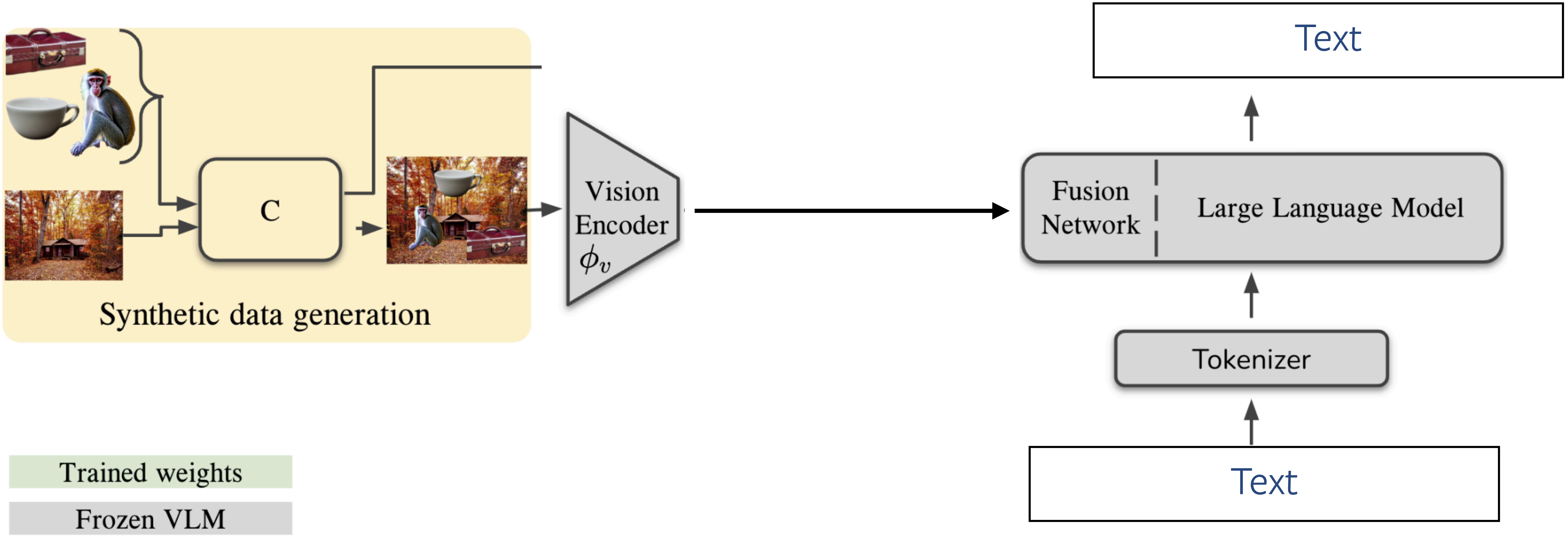
    

Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Default Flamingo



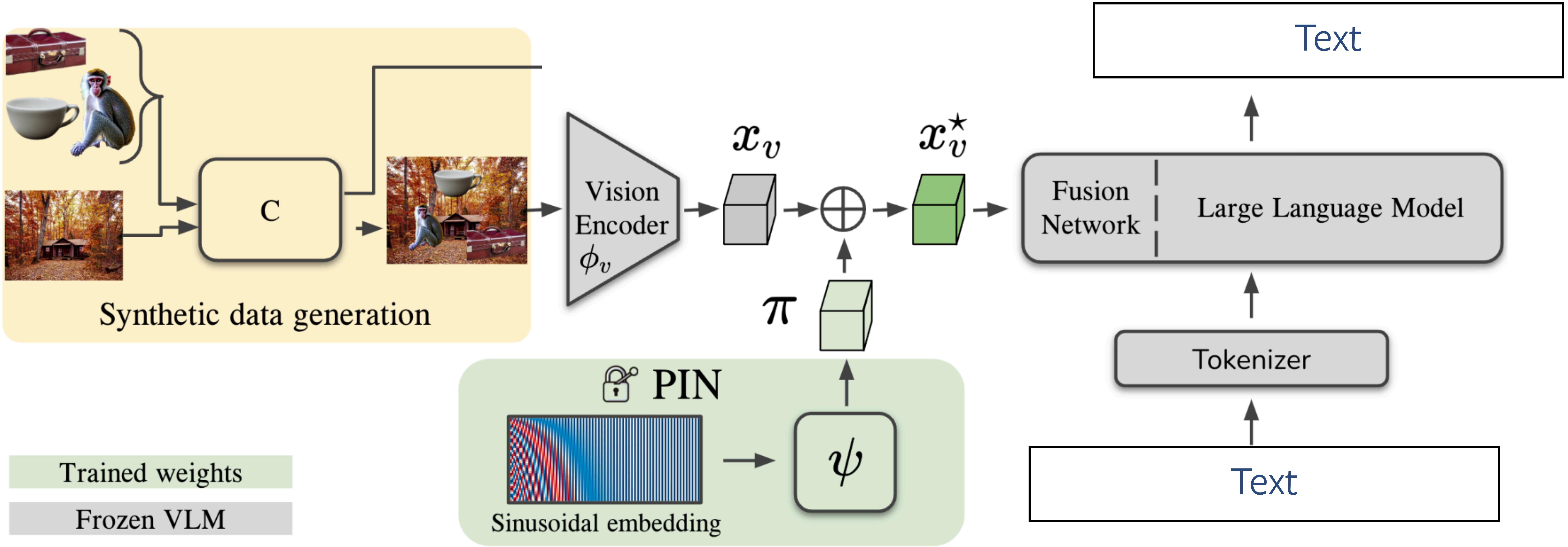
Our method 1: feed the frozen vision encoder synthetic data



Fundamental AI Lab

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

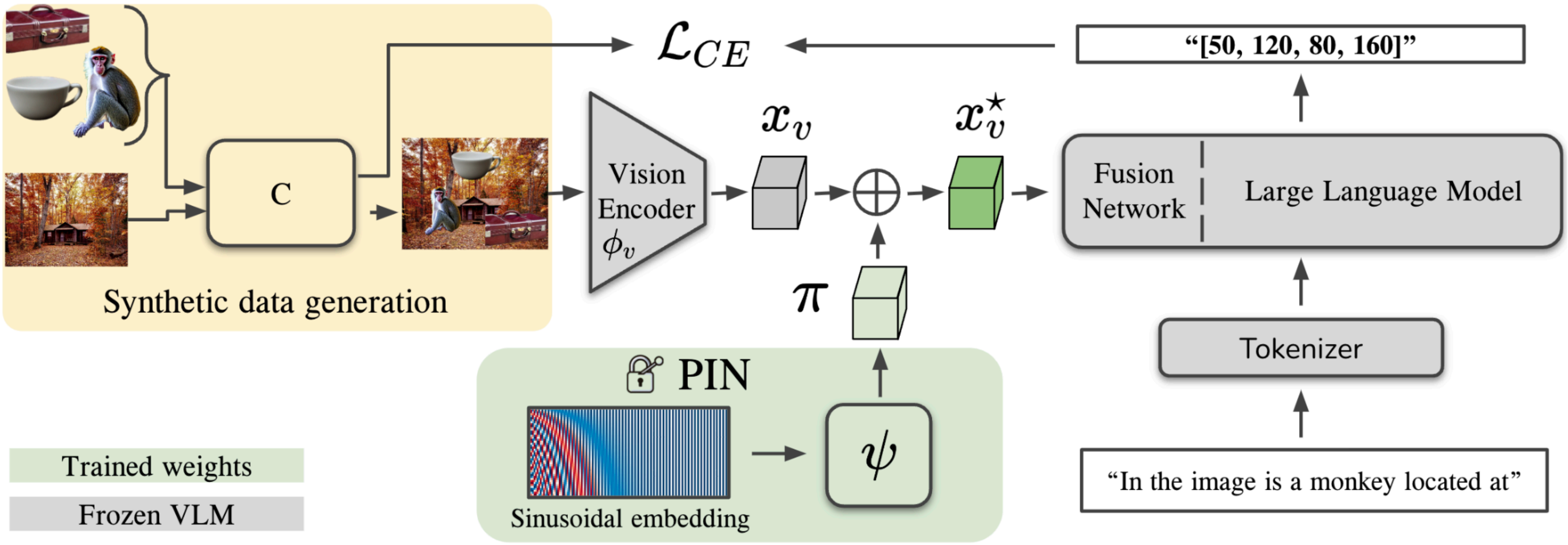
Our method 2: provide VLM spatial learning capacity



Fundamental AI Lab

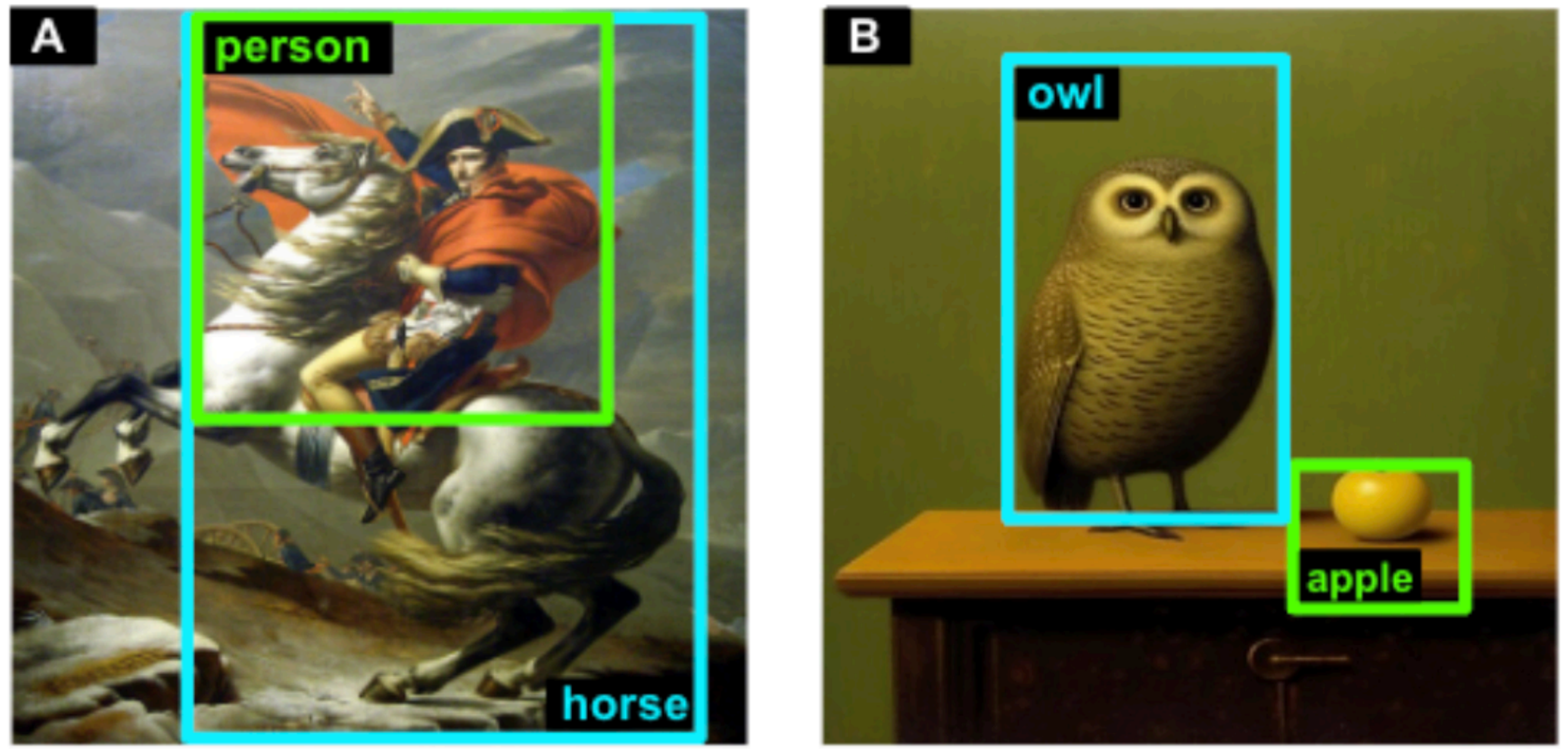
Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

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

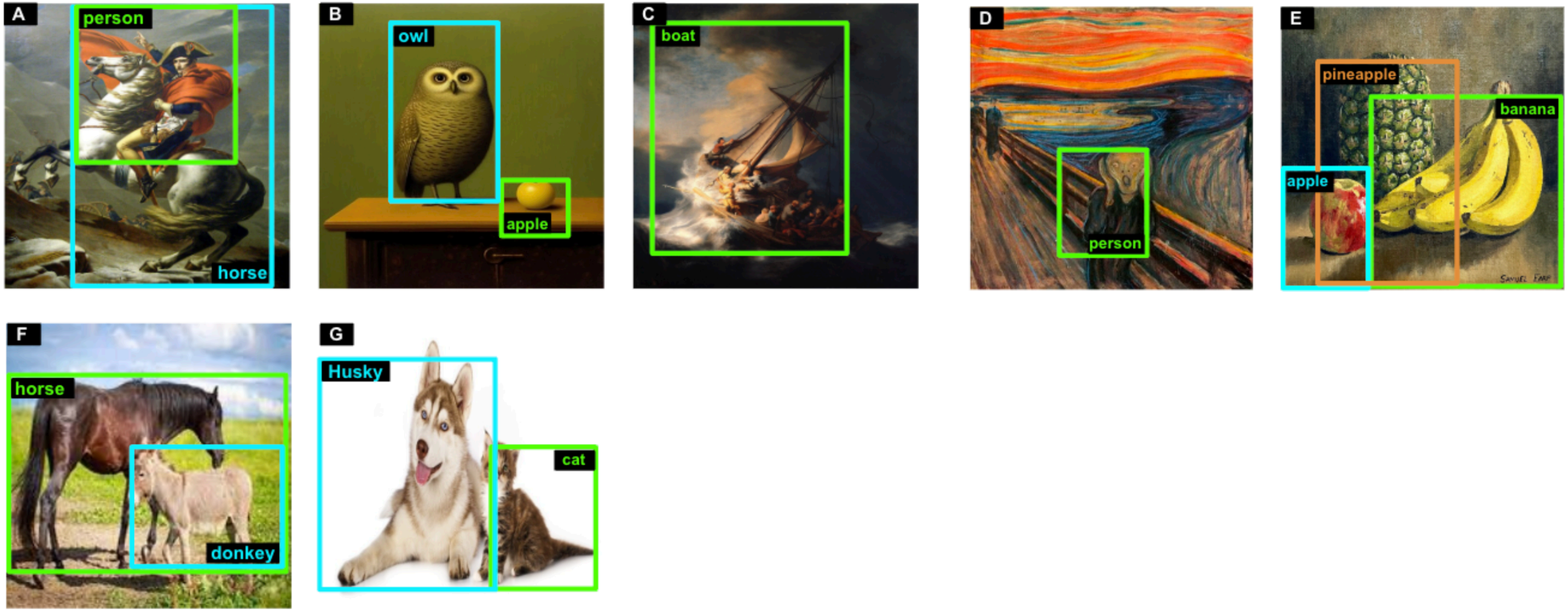


Dorkenwald, Snoek, Asano. PINs: Positional Insert unlocks object localisation abilities in VLMs, CVPR'24.

Results

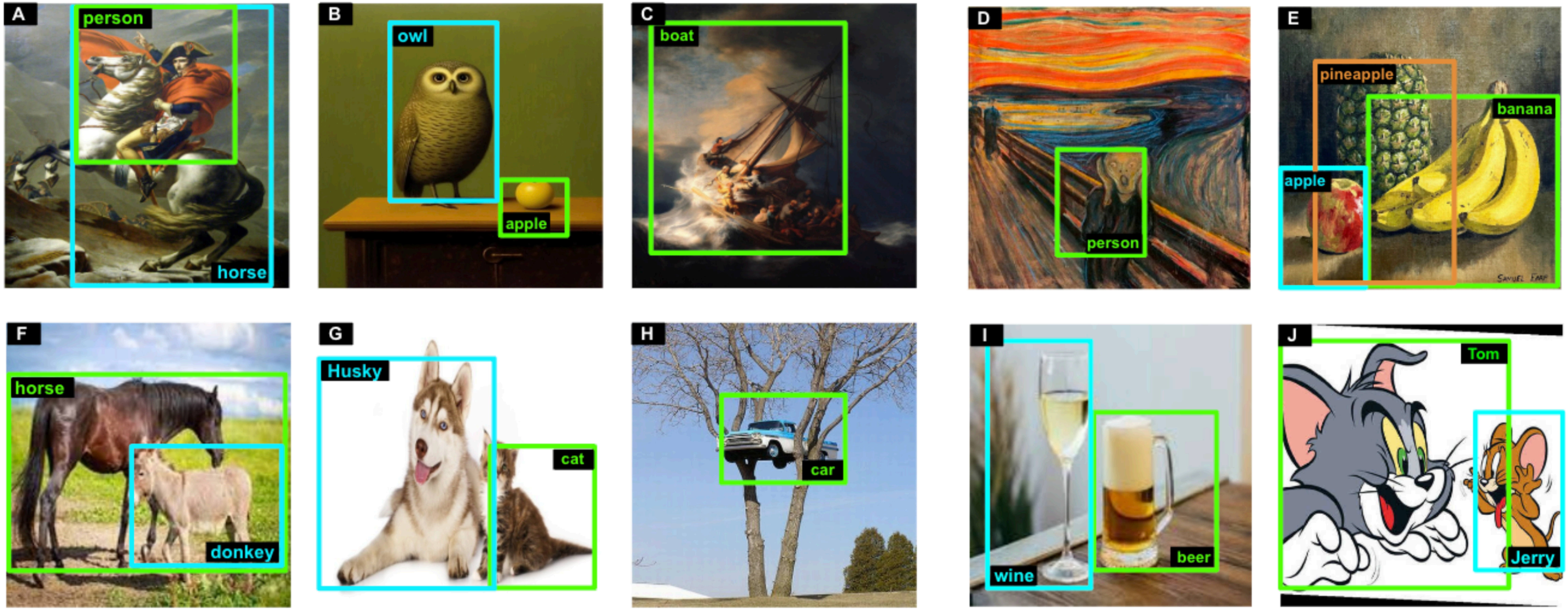

A
person
horse
B
owl
apple
C
boat
D
person
E
pineapple
banana
apple

Results



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

Results



Fundamental AI Lab

We beat common PEFT methods

Model	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$\pm$0.04	0.10$\pm$0.02	0.33$\pm$0.06	0.12$\pm$0.04	0.07$\pm$0.02	0.22$\pm$0.08	0.07$\pm$0.03	0.06$\pm$0.02	0.18$\pm$0.09
	2 context	0.19$\pm$0.11	0.08$\pm$0.05	0.30$\pm$0.18	0.10$\pm$0.08	0.06$\pm$0.04	0.18$\pm$0.16	0.04$\pm$0.06	0.03$\pm$0.04	0.10$\pm$0.15
	5 context	0.19$\pm$0.09	0.07$\pm$0.04	0.31$\pm$0.15	0.10$\pm$0.08	0.06$\pm$0.04	0.20$\pm$0.16	0.06$\pm$0.05	0.04$\pm$0.03	0.17$\pm$0.13
	10 context	0.20$\pm$0.11	0.06$\pm$0.03	0.32$\pm$0.18	0.09$\pm$0.07	0.05$\pm$0.04	0.17$\pm$0.14	0.05$\pm$0.05	0.03$\pm$0.03	0.15$\pm$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**

Fundamental AI Lab

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


A
B
C
D
E
"Left black shirt"
"Old lady in between the players"
"A guy in red on left"
"Guy in orange"
"Right player"
F
G
H
I
J
"Top left apron strings"
"Pizza squares left"
"Pizza right front piece in middle"
"A man black"
"A right person"
Predictions
Ground Truth

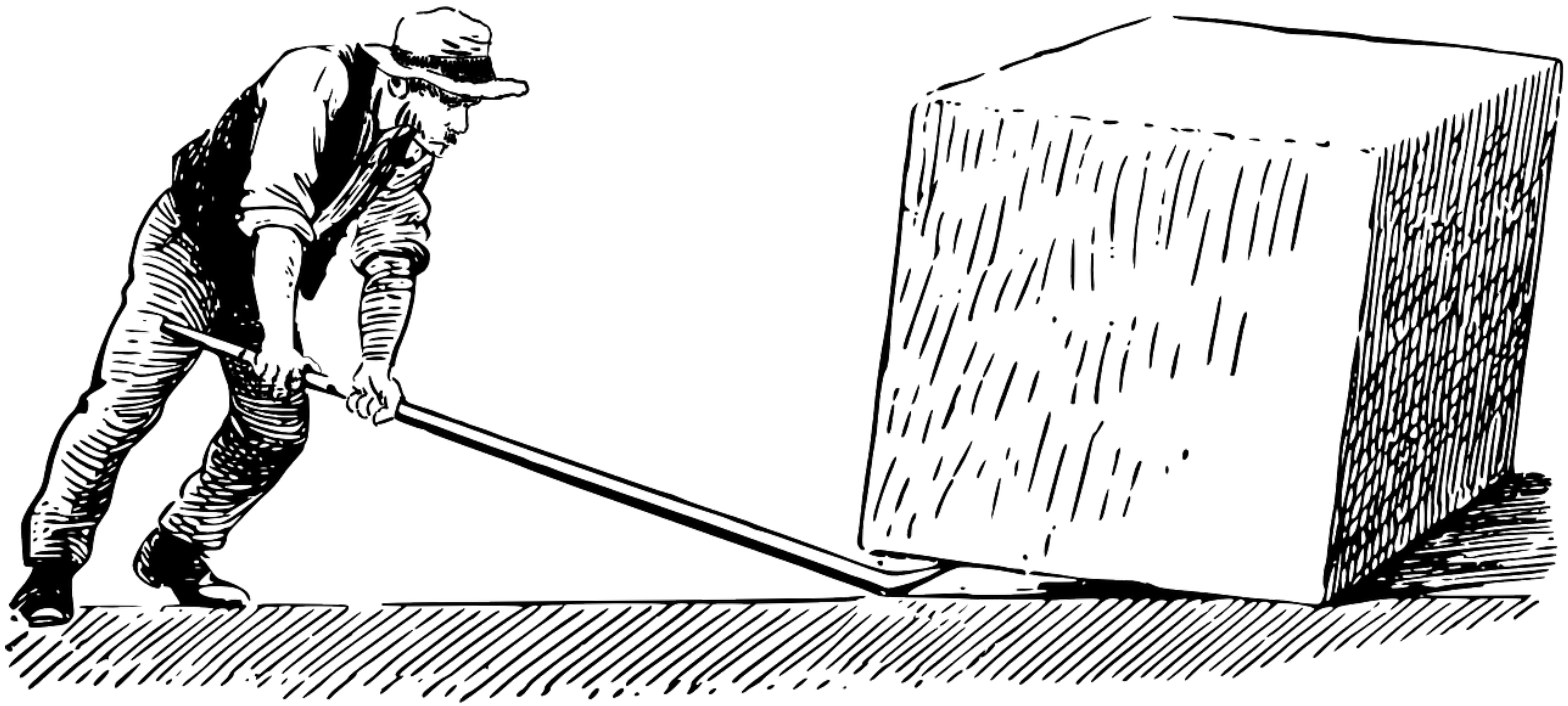
VeRA: Vector-based Random Matrix Adaptation

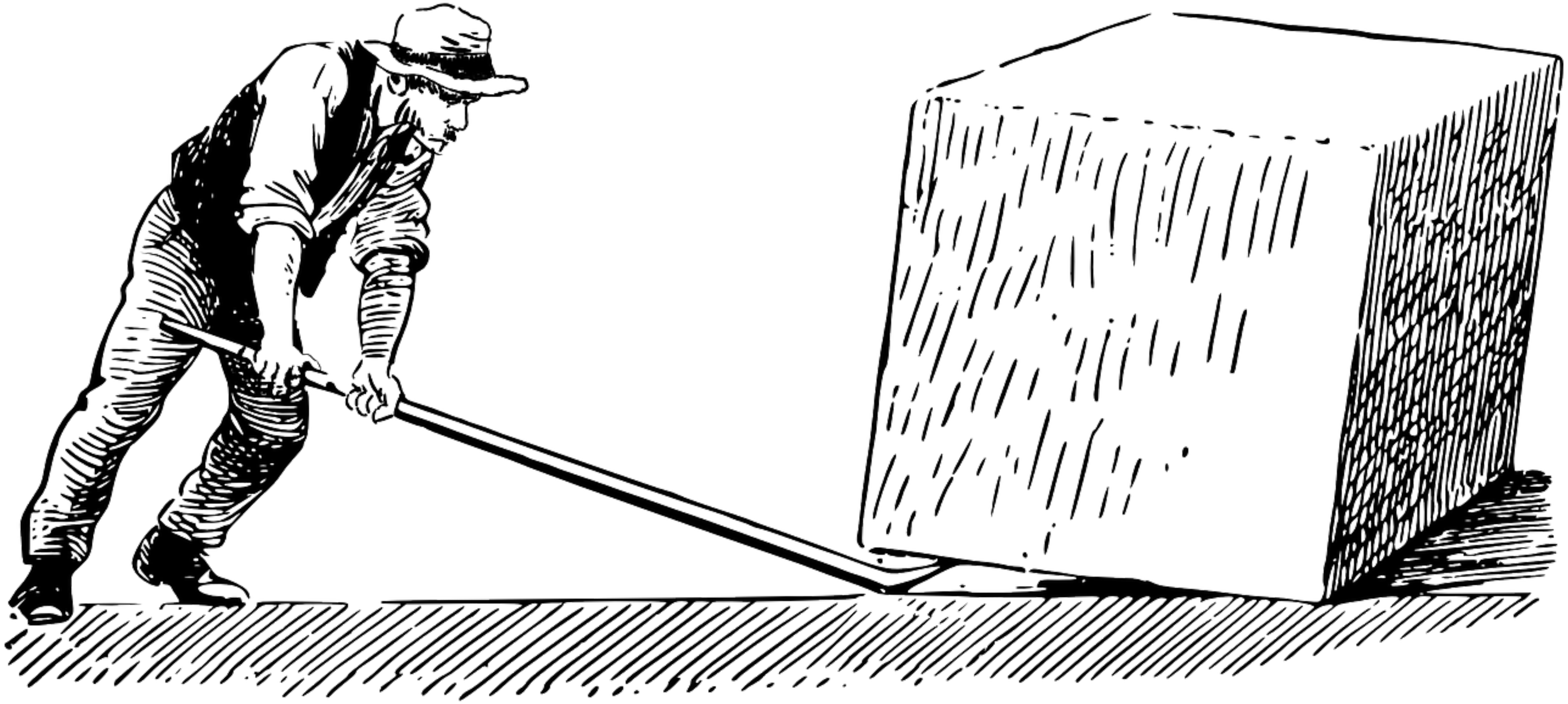
Dawid J. Kopiczko, Tijmen Blankevoort, Yuki M. Asano

ICLR 2024

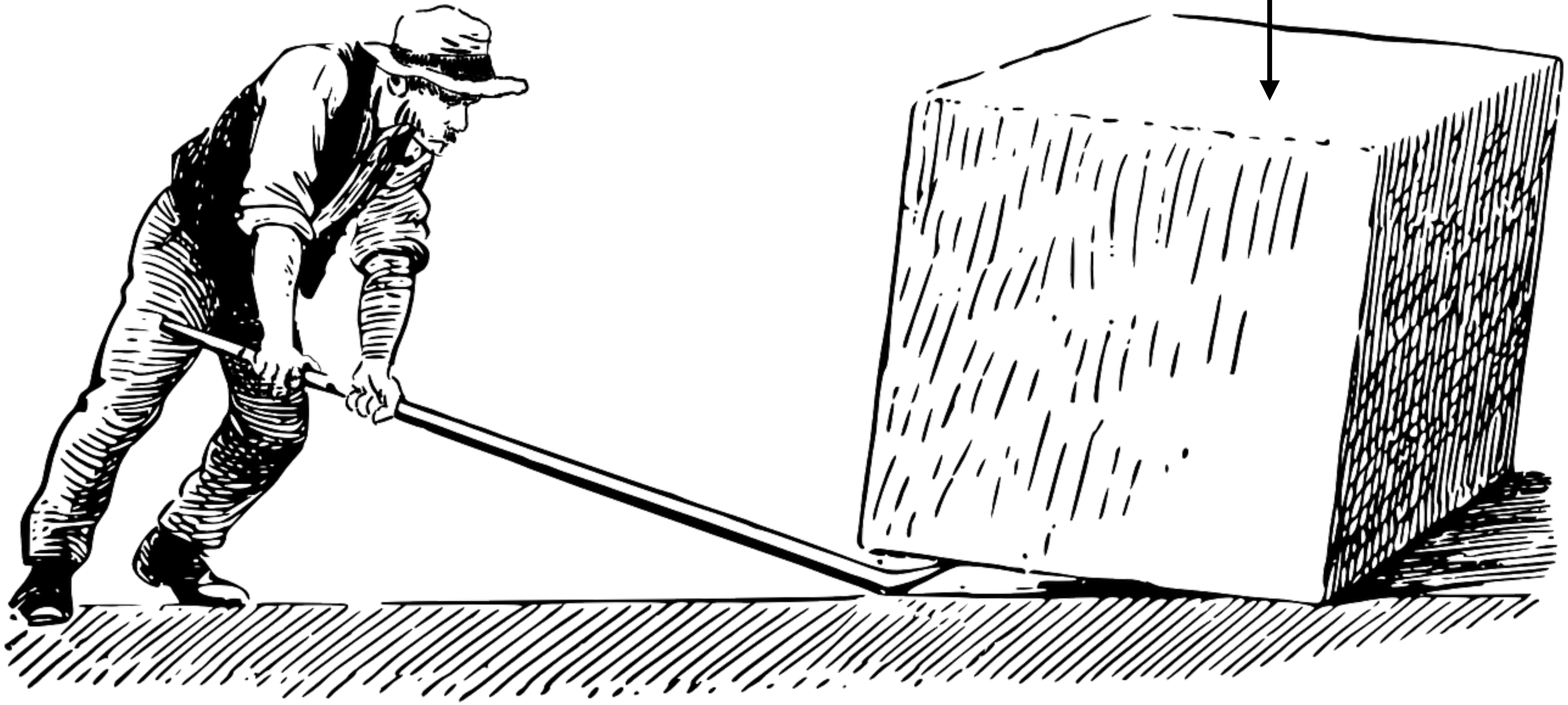
Fundamental AI Lab

Qualcomm

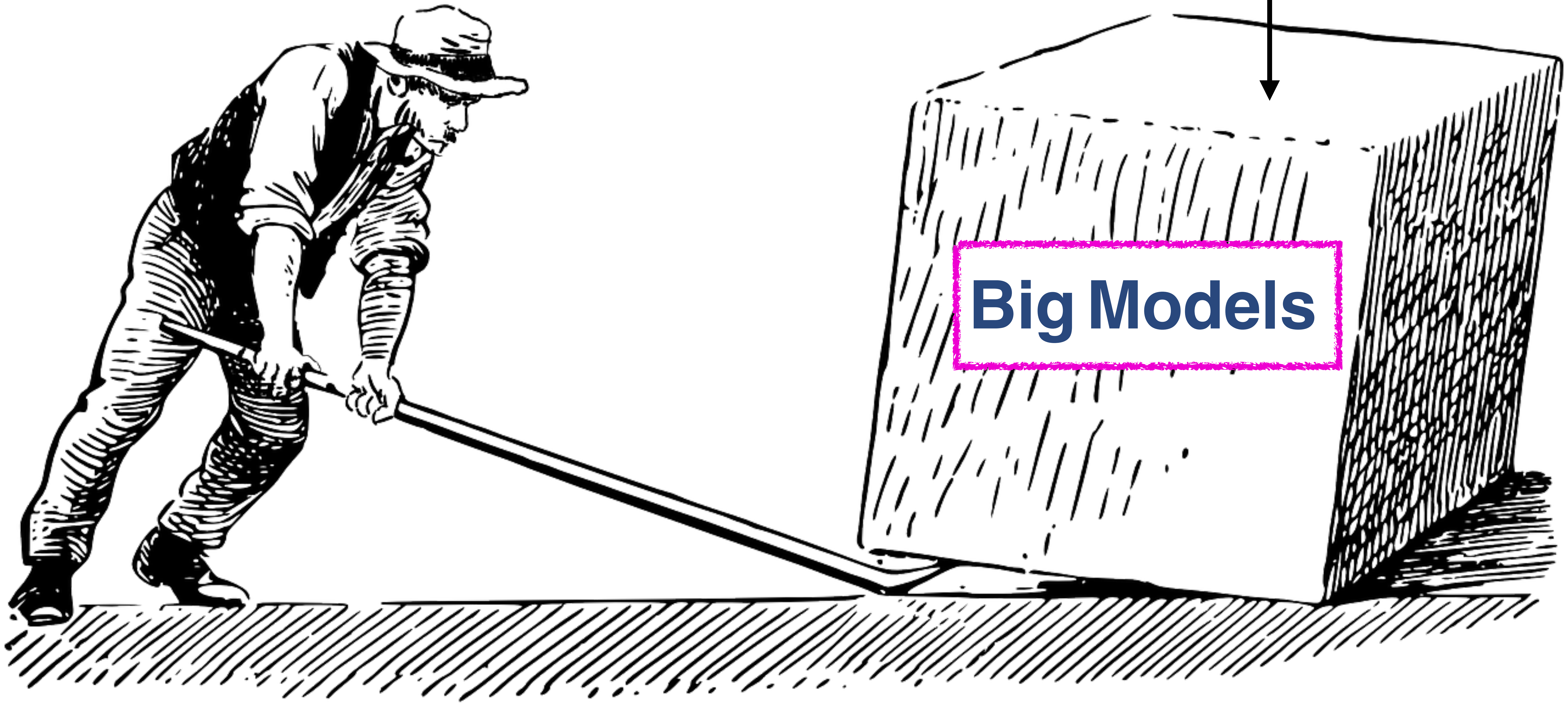




GPUs, data, $$$

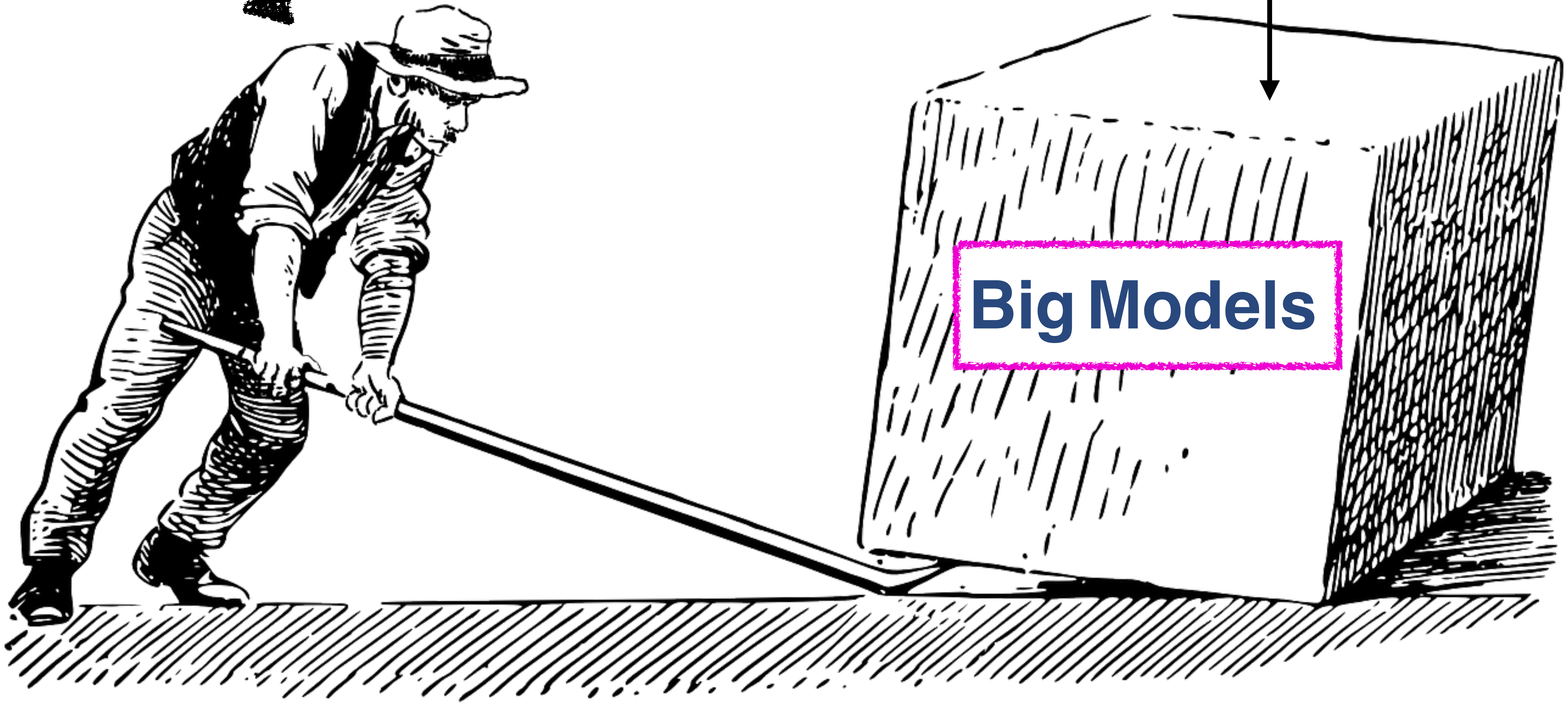


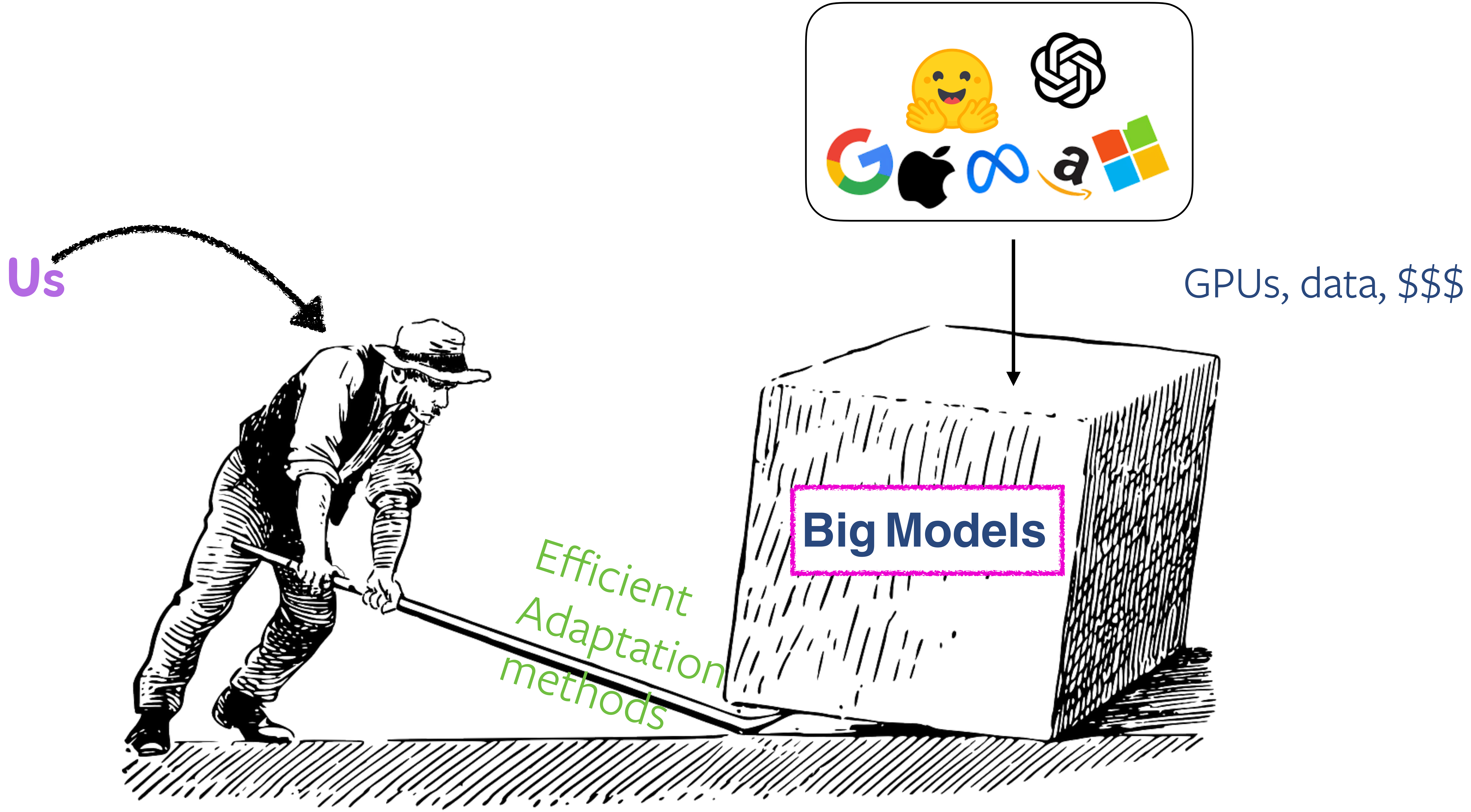
GPUs, data, $$$

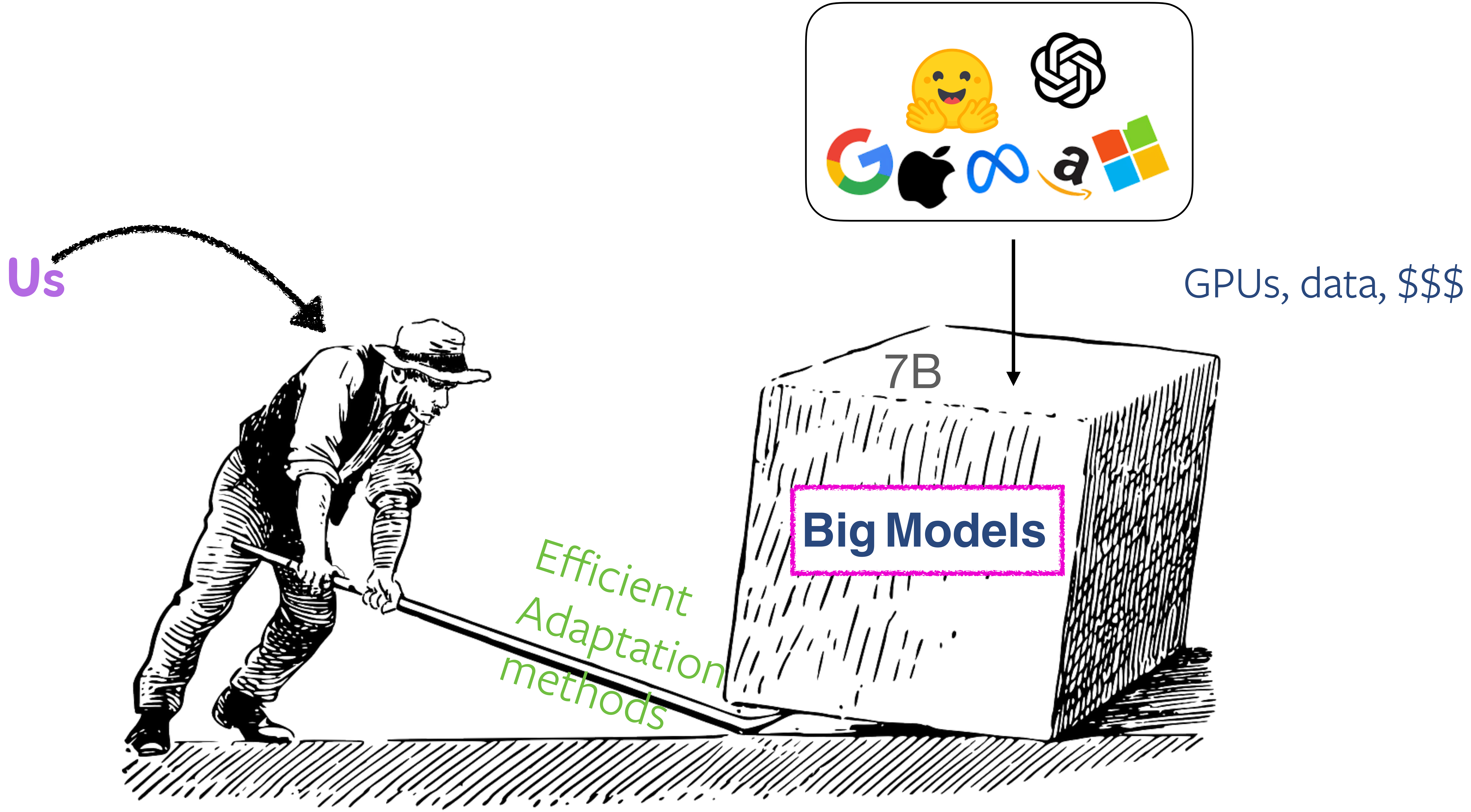


Us

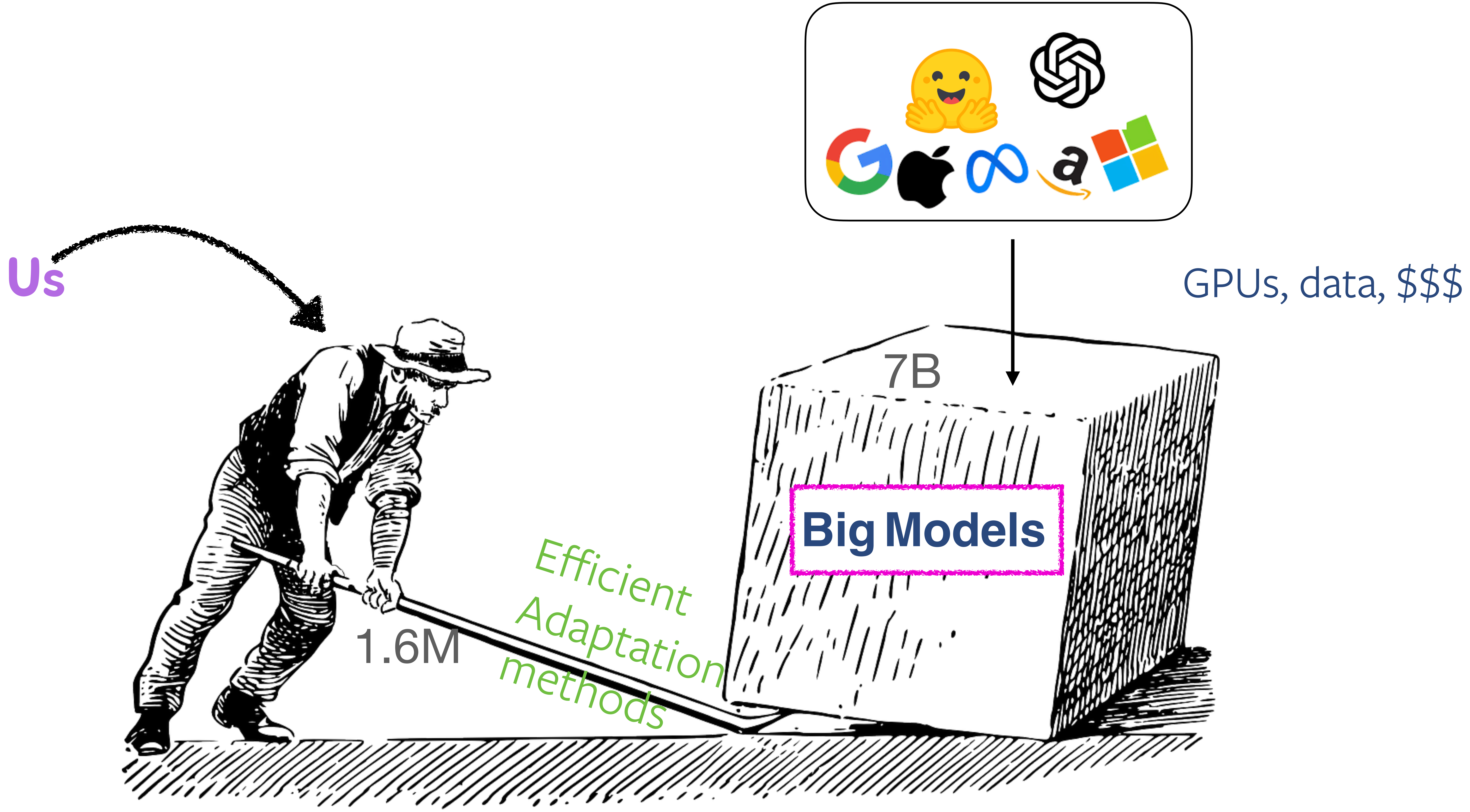
GPUs, data, $$$



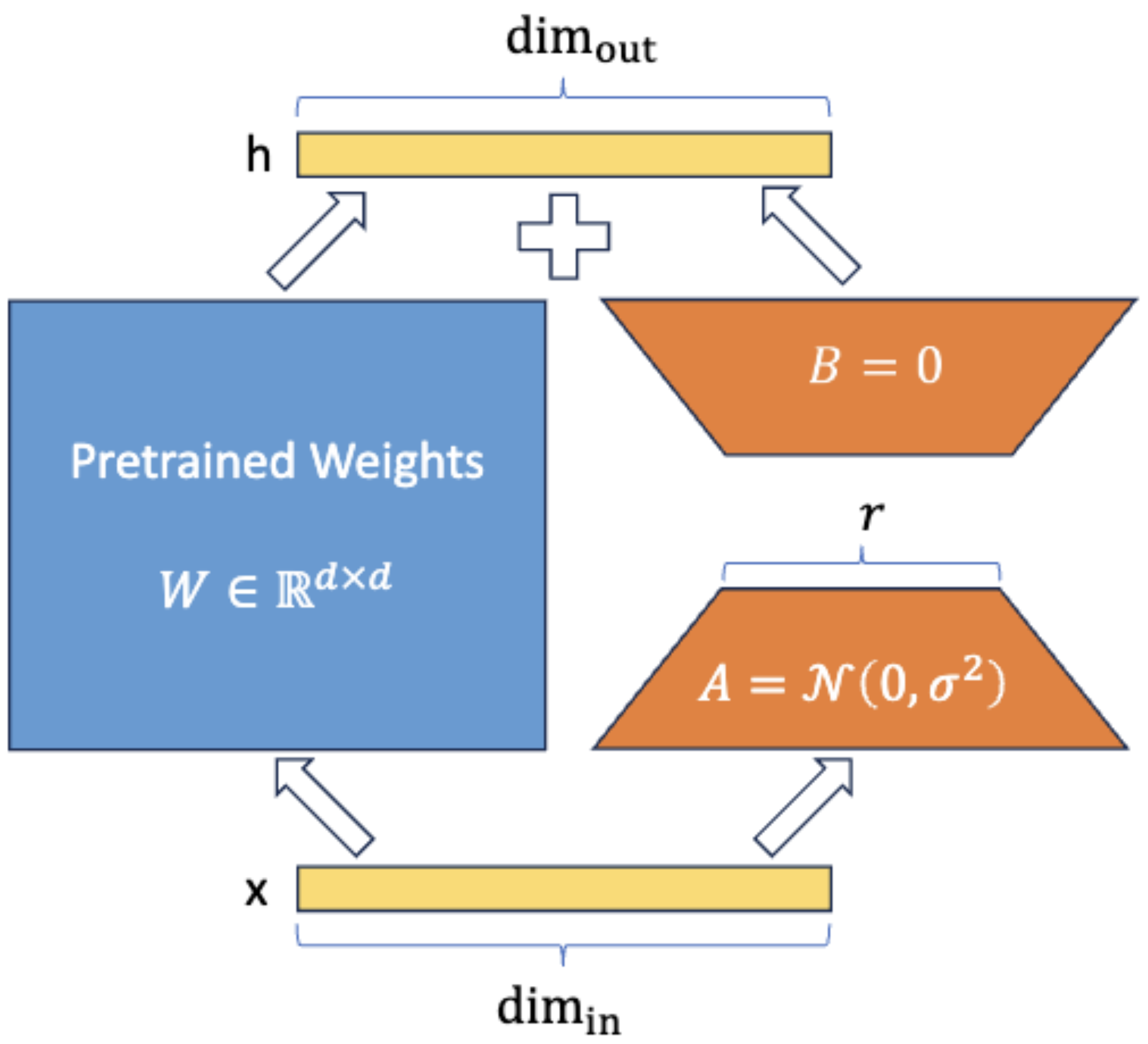

Us
GPUs, data, $$$
Big Models
Efficient Adaptation methods



Fundamental
AI Lab


Us
GPUs, data, $$$
7B
Big Models
1.6M
Efficient
Adaptation
methods

We make LoRA more efficient

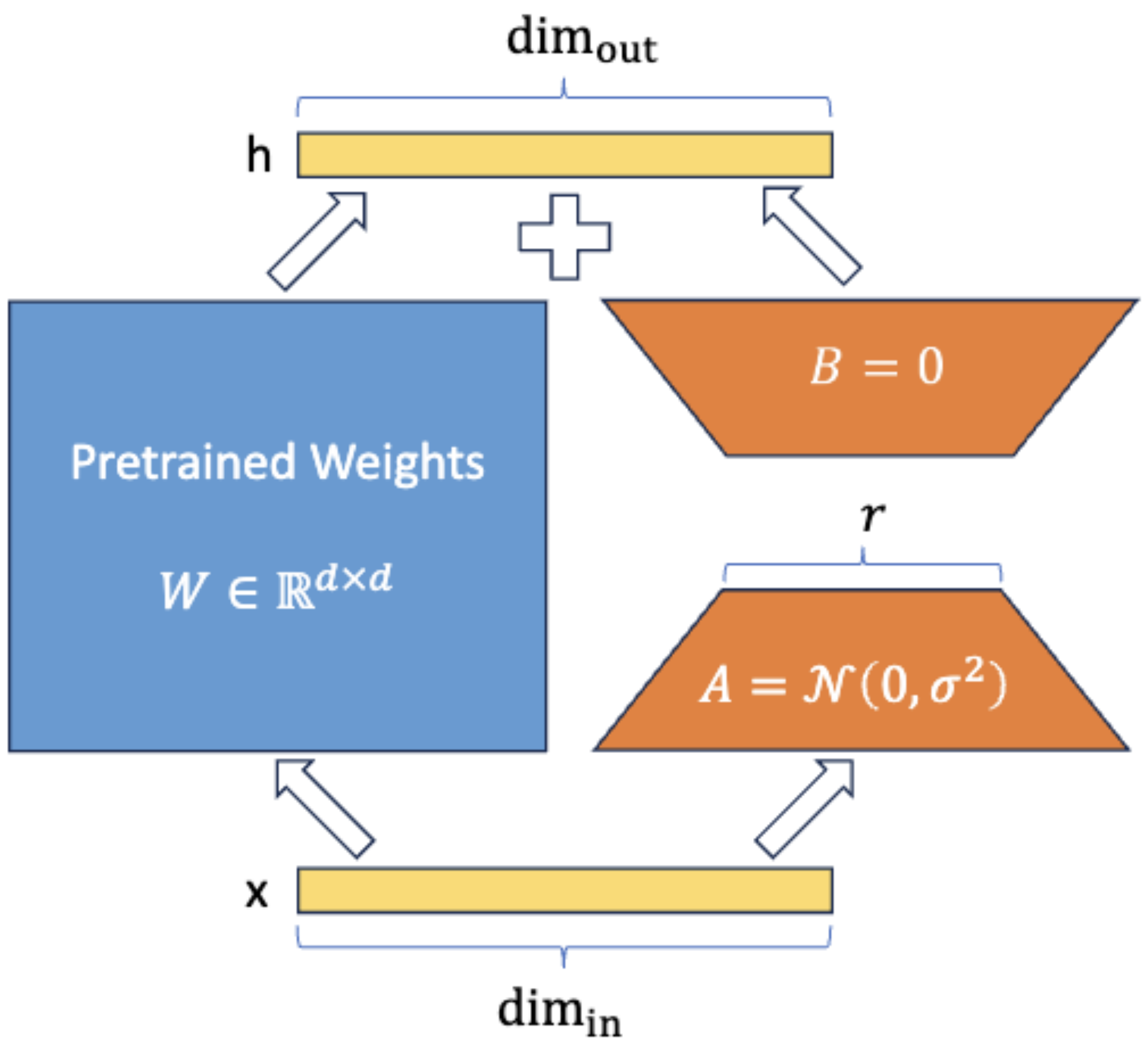


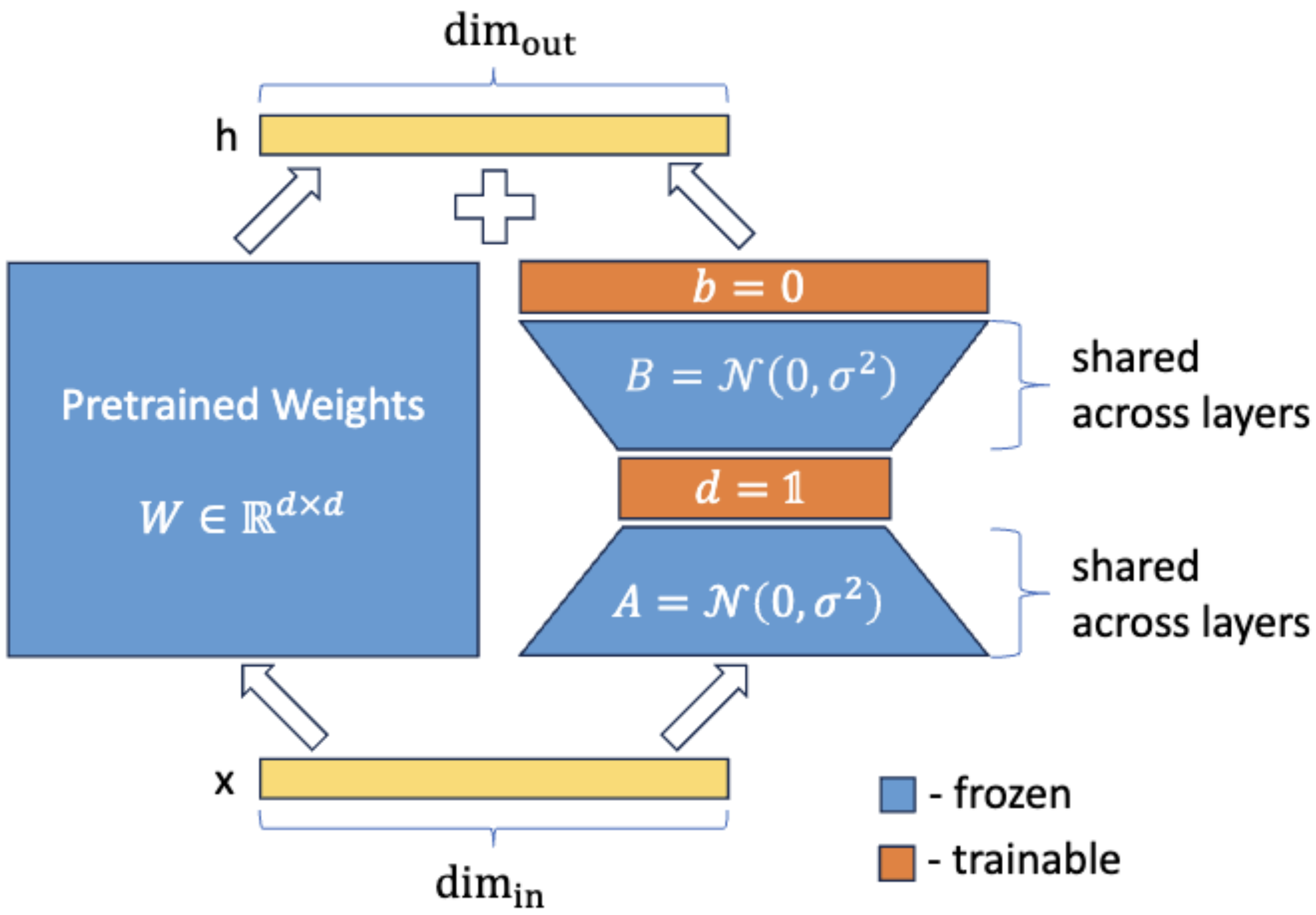
Low-Rank Adaptation (LoRA)

W′ = W + AB,
where A,B are low-rank,
learned per-layer

Fundamental AI Lab

We make LoRA more efficient





Low-Rank Adaptation (LoRA)

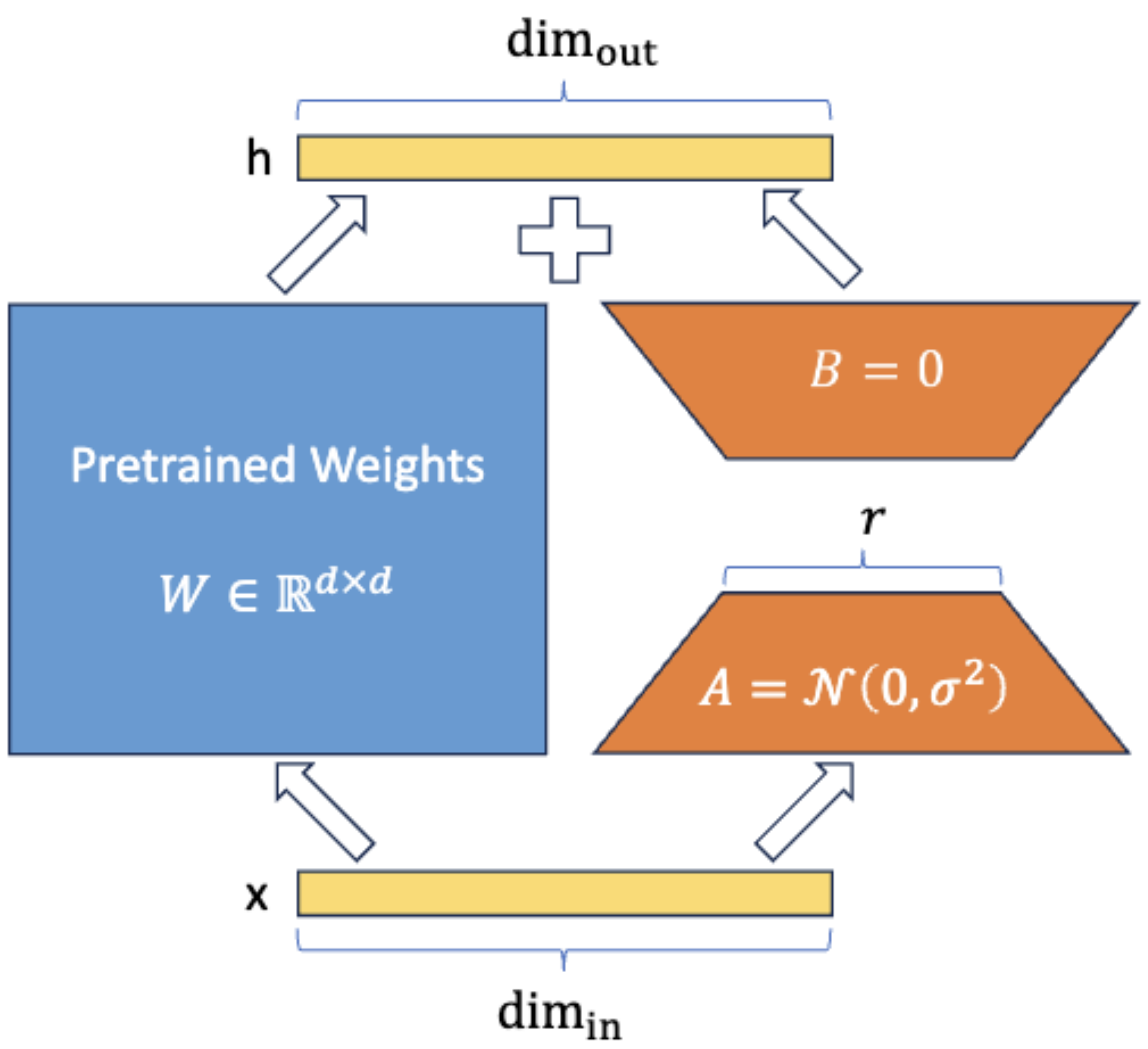
W′ = W + AB,

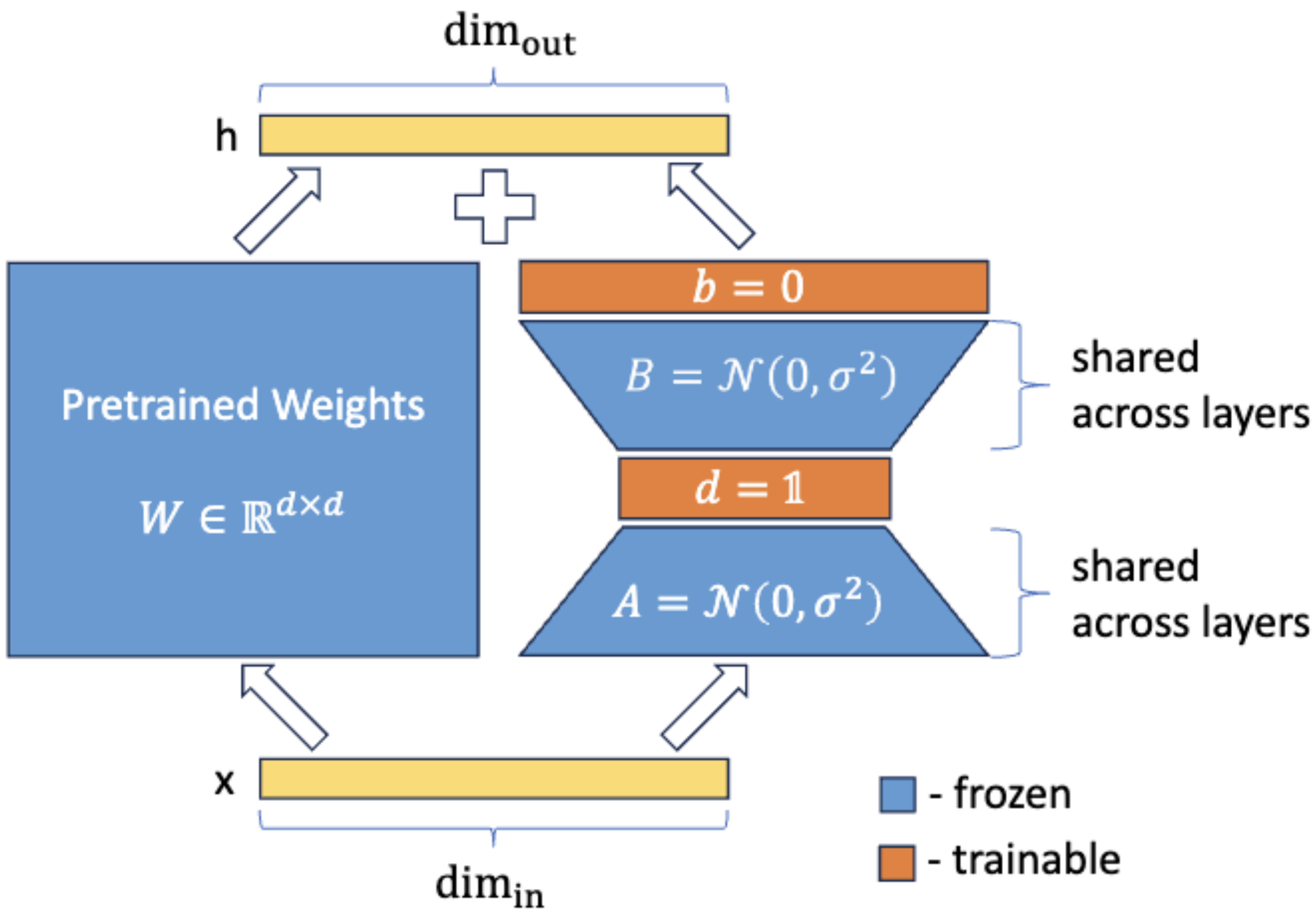
where A,B are low-rank,

learned per-layer

Fundamental AI Lab

We make LoRA more efficient





Low-Rank Adaptation (LoRA)

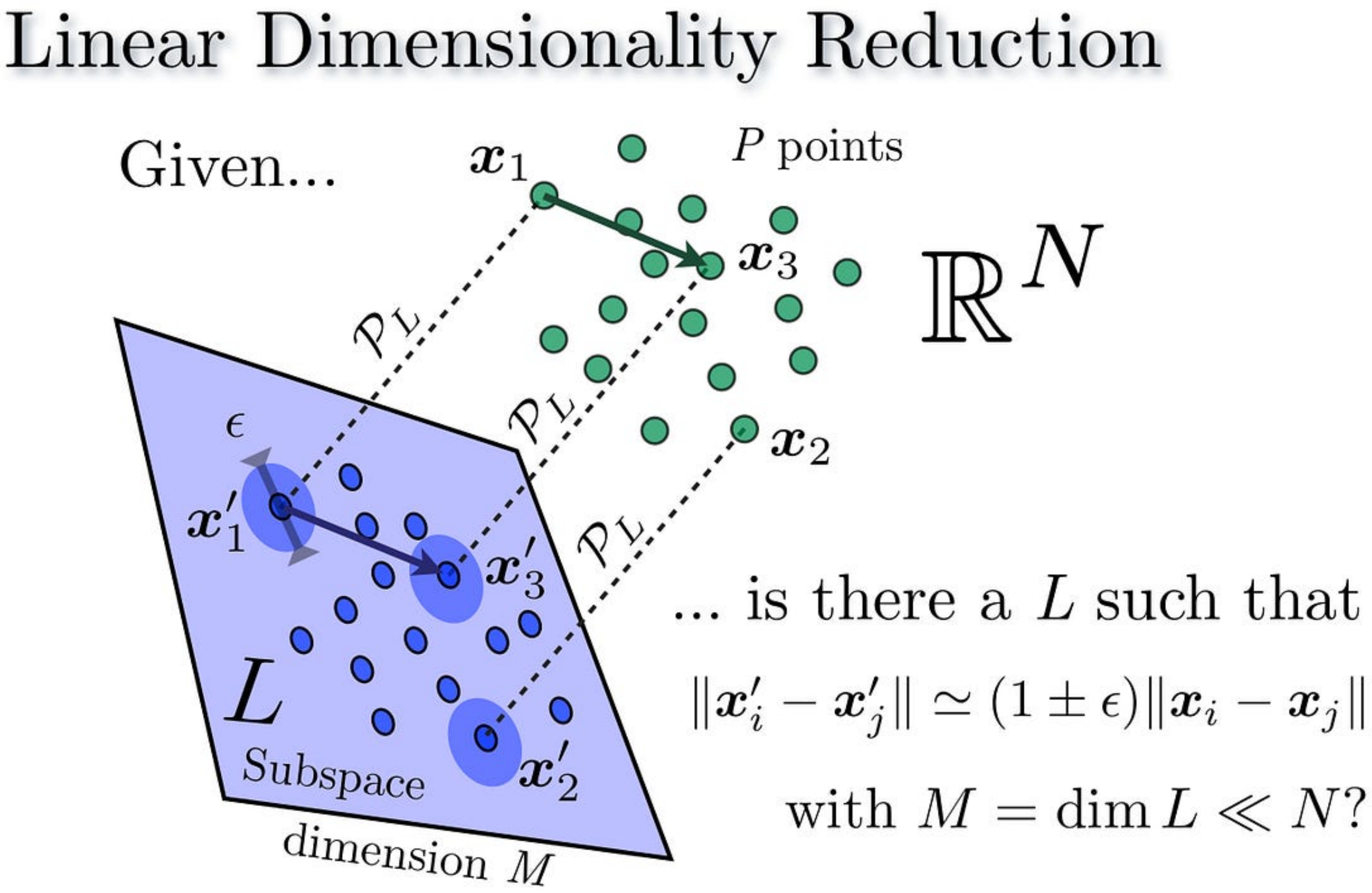
W′ = W + AB,
where A,B are low-rank,
learned per-layer

Vector-based Random Matrix Adaptation (VeRA)

W′ = W + AdBb,
where A,B are random & frozen, same across layers;
d,b are learned vectors

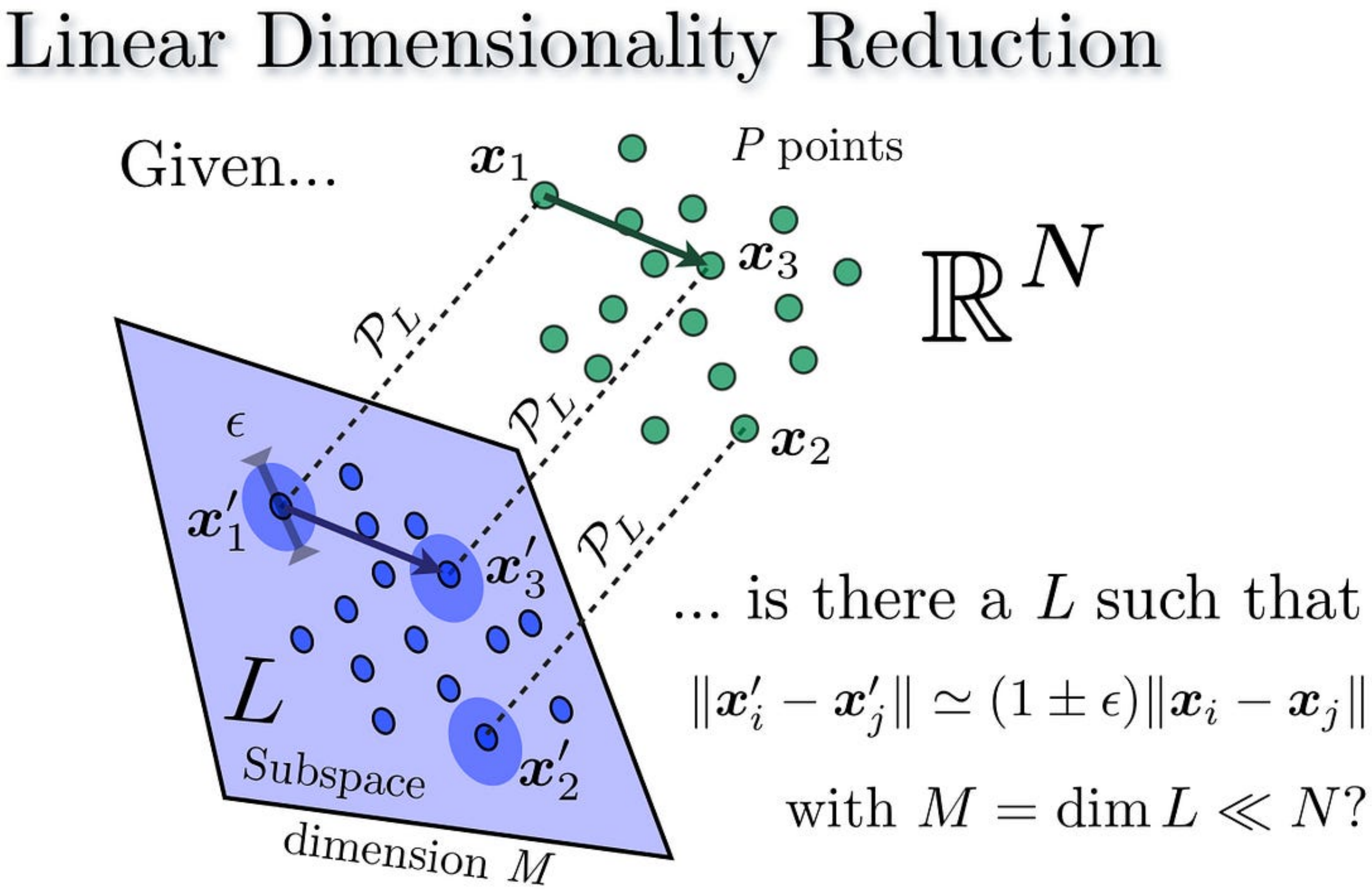
Fundamental AI Lab

Random matrices are powerful!



Random Features for Large-Scale Kernel Machines. Rahimi et al. NeurIPS 2007
Training BatchNorm and Only BatchNorm: On the Expressive Power of Random Features in CNNs. Frankle et al. ICLR 2021
HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models. Ruiz et al. ArXiv 2023

Random matrices are powerful!



Turns out: random projection is very good

Theorem 5 *Let $B_{n\times k}$ be a matrix whose entries are i.i.d. $N(0,1)$, and let $f : \mathbb{R}^n \to \mathbb{R}^k$ be given by $f(x) = \frac{Bx}{\sqrt{k}}$. Then with high probability (say $\geq \frac{1}{2}$ or $\geq 1 - \frac{1}{n}$), for all $x \neq y \in X$ we have $1 - \epsilon \leq \frac{\|f(x)-f(y)\|}{\|x-y\|} \leq 1 + \epsilon$.*

Proof It suffices to prove that for all $v \in \mathbb{R}^n$,

$$\Pr_f[1 - \epsilon \leq \frac{\|f(v)\|}{\|v\|} \leq 1 + \epsilon] > 1 - \frac{1}{n^3}. \quad (1)$$

This is enough because applying (1) to $v = x - y$ for $x, y \in \mathbb{R}^n$, we get $f(x,y) = f(x) - f(y)$, and by a union bound over $\binom{n}{2}$ pairs $x \neq y \in X$ the theorem follows.

In fact, it suffices to prove (1) for unit length v, since f is a linear transformation and $f\left(\frac{v}{\|v\|}\right) = \frac{f(v)}{\|v\|}$ for all v.

Assume $\|v\| = 1$. Each coordinate of $Bv = (\langle b_1, v\rangle, \ldots, \langle b_k, v\rangle)$ has distribution $N(0, \sigma^2 = \|v\|^2 = 1)$ by Fact 4, and they are clearly independent. Denoting $g_i = \langle b_i, v\rangle$, we have:

$$\mathbb{E}[\|f(v)\|^2] = \mathbb{E}\left[\frac{\|Bv\|^2}{k}\right] = \frac{1}{k}\mathbb{E}[g_1^2 + \ldots + g_k^2] = 1, \quad (2)$$

so $\mathbb{E}[\|Bv\|^2] = k$. We will bound $\Pr[\|f(v)\|^2 \geq (1+\epsilon)^2] = \Pr[\|Bv\|^2 \geq k(1+\epsilon)^2]$. A similar argument works for the event $\|Bv\|^2 \leq k(1+\epsilon)^2$.

Fundamental AI Lab

Random Features for Large-Scale Kernel Machines. Rahimi et al. NeurIPS 2007
Training BatchNorm and Only BatchNorm: On the Expressive Power of Random Features in CNNs. Frankle et al. ICLR 2021
HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models. Ruiz et al. ArXiv 2023

LoRA family

- LoRA: https://arxiv.org/abs/2106.09685
  - the OG of parameter-efficient finetuning
- **VeRA: https://arxiv.org/abs/2310.11454**
  - **10x more parameter-efficient than LoRA**
- QLoRA https://arxiv.org/abs/2305.14314
  - LoRA on a quantised LLM + tricks
- **LoRA-FA: https://arxiv.org/abs/2308.03303**
  - **2x more parameter-efficient than LoRA**
- **OFT: https://arxiv.org/abs/2306.07967**
  - **LoRA but with orthogonal matrices**
- **BOFT: https://arxiv.org/abs/2311.06243**
  - **Upgrade on OFT,**
- LoKr: https://arxiv.org/abs/2103.10385
  - Combines two LoRAs via Kronecker product
- LoHa: https://arxiv.org/abs/2108.06098
  - Hadamard product of two LoRA updates
- **NOLA: https://arxiv.org/abs/2310.02556**
  - **Uses learnable Kronecker products of random matrices**
- DyLoRA: https://arxiv.org/abs/2210.07558
  - trains LoRA with any ranks and then picks one
- KronA: https://arxiv.org/abs/2212.10650
  - adaptation based on Kronecker products
- Delta-LoRA: https://arxiv.org/abs/2309.02411
  - Incremental updates to the original fully connected layer
- AdaLoRA: https://arxiv.org/abs/2303.10512
  - adaptively allocates the rank of LoRA during training
- LoftQ: https://arxiv.org/abs/2310.08659
  - Initialise LoRA to minimise quantisation error of LLM
- DoRA: https://arxiv.org/abs/2402.09353
  - do the weight-norm trick on LoRA matrix (learn direction)
- PiSSA
  - start LoRA with SVD
- LoRA-XS
  - frozen SVD and learnable small matrix inbetween U,V

Results on GLUE with RoBERTa

	Method	# Trainable Parameters	SST-2	MRPC	CoLA	QNLI	RTE	STS-B	Avg.
BASE	FT	125M	94.8	90.2	63.6	92.8	78.7	91.2	85.2
	BitFit	0.1M	93.7	**92.7**	62.0	91.8	81.5	90.8	85.4
	Adpt^{D}	0.3M	$94.2_{\pm 0.1}$	$88.5_{\pm 1.1}$	$60.8_{\pm 0.4}$	$93.1_{\pm 0.1}$	$71.5_{\pm 2.7}$	$89.7_{\pm 0.3}$	83.0
	Adpt^{D}	0.9M	$94.7_{\pm 0.3}$	$88.4_{\pm 0.1}$	$62.6_{\pm 0.9}$	$93.0_{\pm 0.2}$	$75.9_{\pm 2.2}$	$90.3_{\pm 0.1}$	84.2
	LoRA	0.3M	$\mathbf{95.1}_{\pm 0.2}$	$89.7_{\pm 0.7}$	$63.4_{\pm 1.2}$	$\mathbf{93.3}_{\pm 0.3}$	$\mathbf{86.6}_{\pm 0.7}$	$\mathbf{91.5}_{\pm 0.2}$	**86.6**
	VeRA	**0.043M**	$94.6_{\pm 0.1}$	$89.5_{\pm 0.5}$	$\mathbf{65.6}_{\pm 0.8}$	$91.8_{\pm 0.2}$	$78.7_{\pm 0.7}$	$90.7_{\pm 0.2}$	85.2
LARGE	Adpt^{P}	3M	$96.1_{\pm 0.3}$	$90.2_{\pm 0.7}$	$\mathbf{68.3}_{\pm 1.0}$	$\mathbf{94.8}_{\pm 0.2}$	$83.8_{\pm 2.9}$	$92.1_{\pm 0.7}$	87.6
	Adpt^{P}	0.8M	$\mathbf{96.6}_{\pm 0.2}$	$89.7_{\pm 1.2}$	$67.8_{\pm 2.5}$	$\mathbf{94.8}_{\pm 0.3}$	$80.1_{\pm 2.9}$	$91.9_{\pm 0.4}$	86.8
	Adpt^{H}	6M	$96.2_{\pm 0.3}$	$88.7_{\pm 2.9}$	$66.5_{\pm 4.4}$	$94.7_{\pm 0.2}$	$83.4_{\pm 1.1}$	$91.0_{\pm 1.7}$	86.8
	Adpt^{H}	0.8M	$96.3_{\pm 0.5}$	$87.7_{\pm 1.7}$	$66.3_{\pm 2.0}$	$94.7_{\pm 0.2}$	$72.9_{\pm 2.9}$	$91.5_{\pm 0.5}$	84.9
	LoRA-FA	3.7M	96.0	90.0	68.0	94.4	86.1	92.0	87.7
	LoRA	0.8M	$96.2_{\pm 0.5}$	$90.2_{\pm 1.0}$	$68.2_{\pm 1.9}$	$\mathbf{94.8}_{\pm 0.3}$	$85.2_{\pm 1.1}$	$\mathbf{92.3}_{\pm 0.5}$	**87.8**
	VeRA	**0.061M**	$96.1_{\pm 0.1}$	$\mathbf{90.9}_{\pm 0.7}$	$68.0_{\pm 0.8}$	$94.4_{\pm 0.2}$	$\mathbf{85.9}_{\pm 0.7}$	$91.7_{\pm 0.8}$	**87.8**

Results on E2E benchmark with GPT2

	Method	# Trainable Parameters	BLEU	NIST	METEOR	ROUGE-L	CIDEr
MEDIUM	FT[1]	354.92M	68.2	8.62	46.2	71.0	2.47
	Adpt[L1]	0.37M	66.3	8.41	45.0	69.8	2.40
	Adpt[L1]	11.09M	68.9	8.71	46.1	71.3	2.47
	Adpt[H1]	11.09M	67.3	8.50	46.0	70.7	2.44
	DyLoRA[2]	0.39M	69.2	8.75	46.3	70.8	2.46
	AdaLoRA[3]	0.38M	68.2	8.58	44.1	70.7	2.35
	LoRA	0.35M	68.9	8.69	46.4	71.3	**2.51**
	VeRA	**0.098M**	**70.1**	**8.81**	**46.6**	**71.5**	2.50
LARGE	FT[1]	774.03M	68.5	8.78	46.0	69.9	2.45
	Adpt[L1]	0.88M	69.1	8.68	46.3	71.4	2.49
	Adpt[L1]	23.00M	68.9	8.70	46.1	71.3	2.45
	LoRA	0.77M	70.1	8.80	46.7	**71.9**	2.52
	VeRA	**0.17M**	**70.3**	**8.85**	**46.9**	71.6	**2.54**

Fundamental AI Lab

Instruction tuning: better than LoRA with 100x less parameters

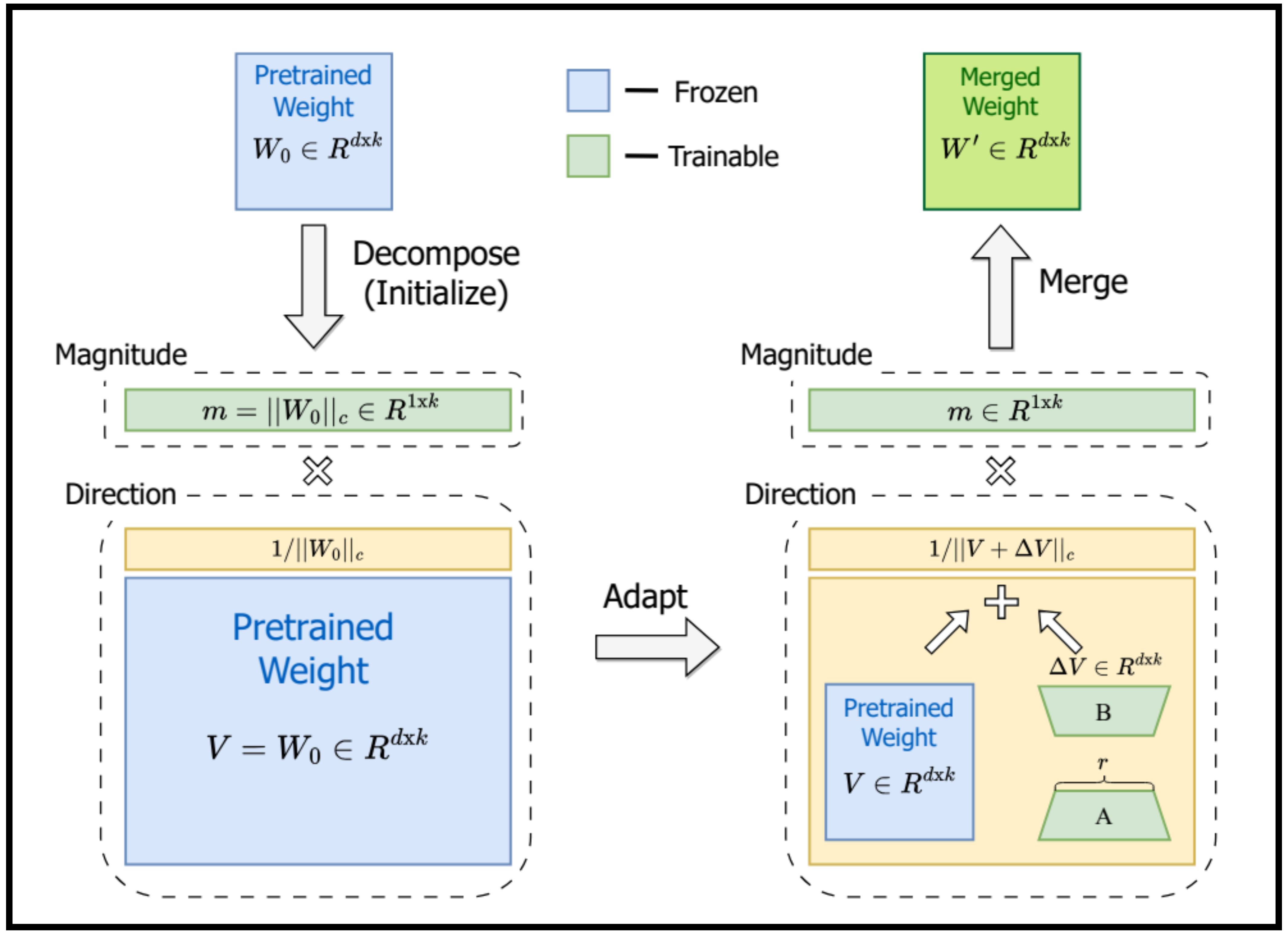
Model	Method	# Parameters	Score
Llama 13B	-	-	2.61
LLAMA 7B	LoRA	159.9M	5.03
	VeRA	1.6M	4.77
LLAMA 13B	LoRA	250.3M	5.31
	VeRA	2.4M	5.22
LLAMA2 7B	LoRA	159.9M	5.19
	VeRA	1.6M	5.08
LLAMA2 13B	LoRA	250.3M	5.77
	VeRA	2.4M	5.93

Fundamental AI Lab

Results on Image Classification with pretrained ViT

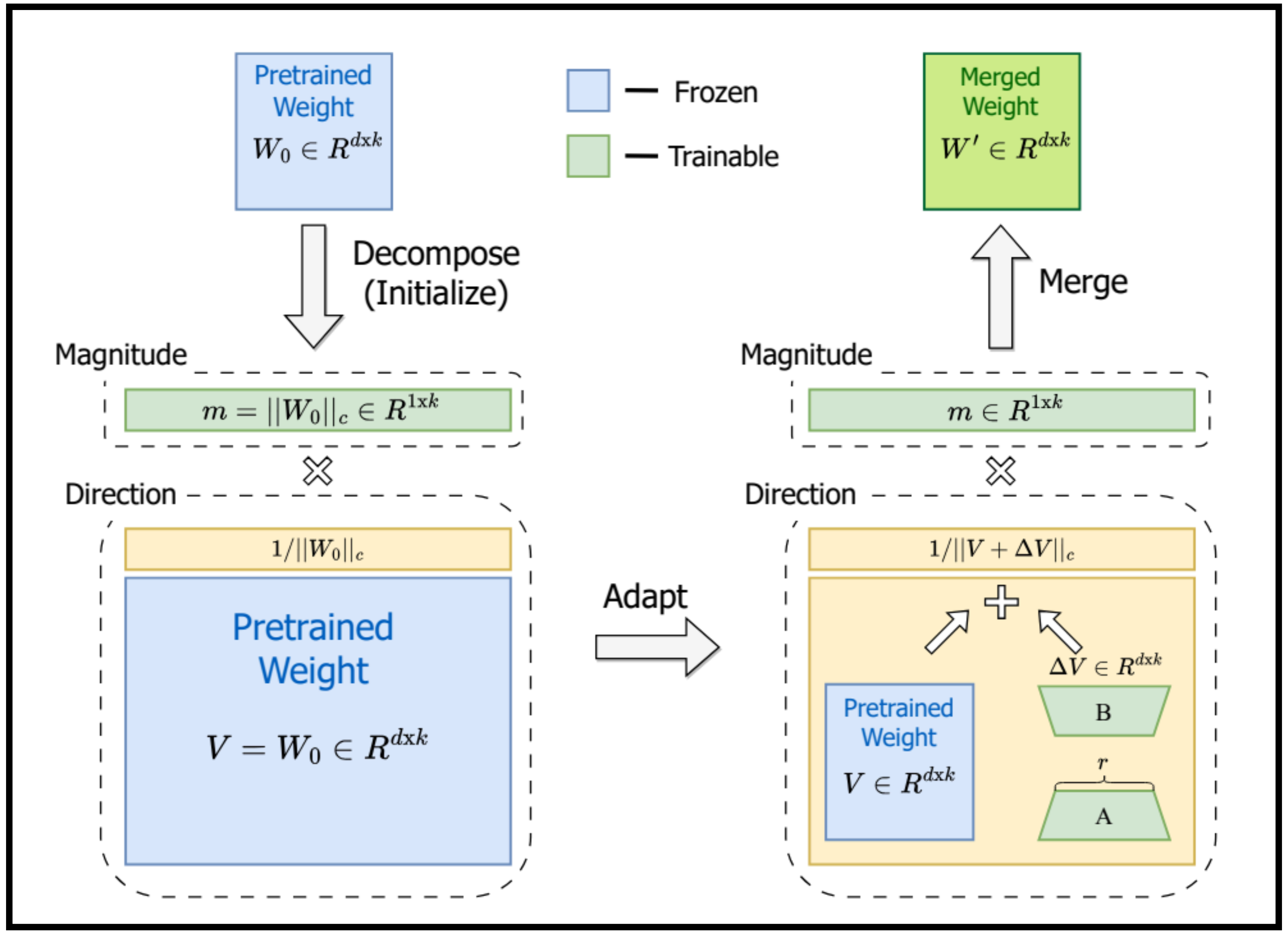
	Method	# Trainable Parameters	CIFAR100	Food101	Flowers102	RESISC45
ViT-B	Head	-	77.7	86.1	98.4	67.2
	Full	85.8M	**86.5**	**90.8**	98.9	**78.9**
	LoRA	294.9K	85.9	89.9	98.8	77.7
	VeRA	**24.6K**	84.8	89.0	**99.0**	77.0
ViT-L	Head	-	79.4	76.5	98.9	67.8
	Full	303.3M	86.8	78.7	98.8	**79.0**
	LoRA	786.4K	87.0	**79.5**	99.1	78.3
	VeRA	**61.4K**	**87.5**	79.2	**99.2**	78.6

DoRA: Weight-Decomposed Low-Rank Adaptation



DoRA: Weight-Decomposed Low-Rank Adaptation. Liu et al. 2024
Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. Salimans & Kingma. NeurIPS 2016

DoRA: Weight-Decomposed Low-Rank Adaptation

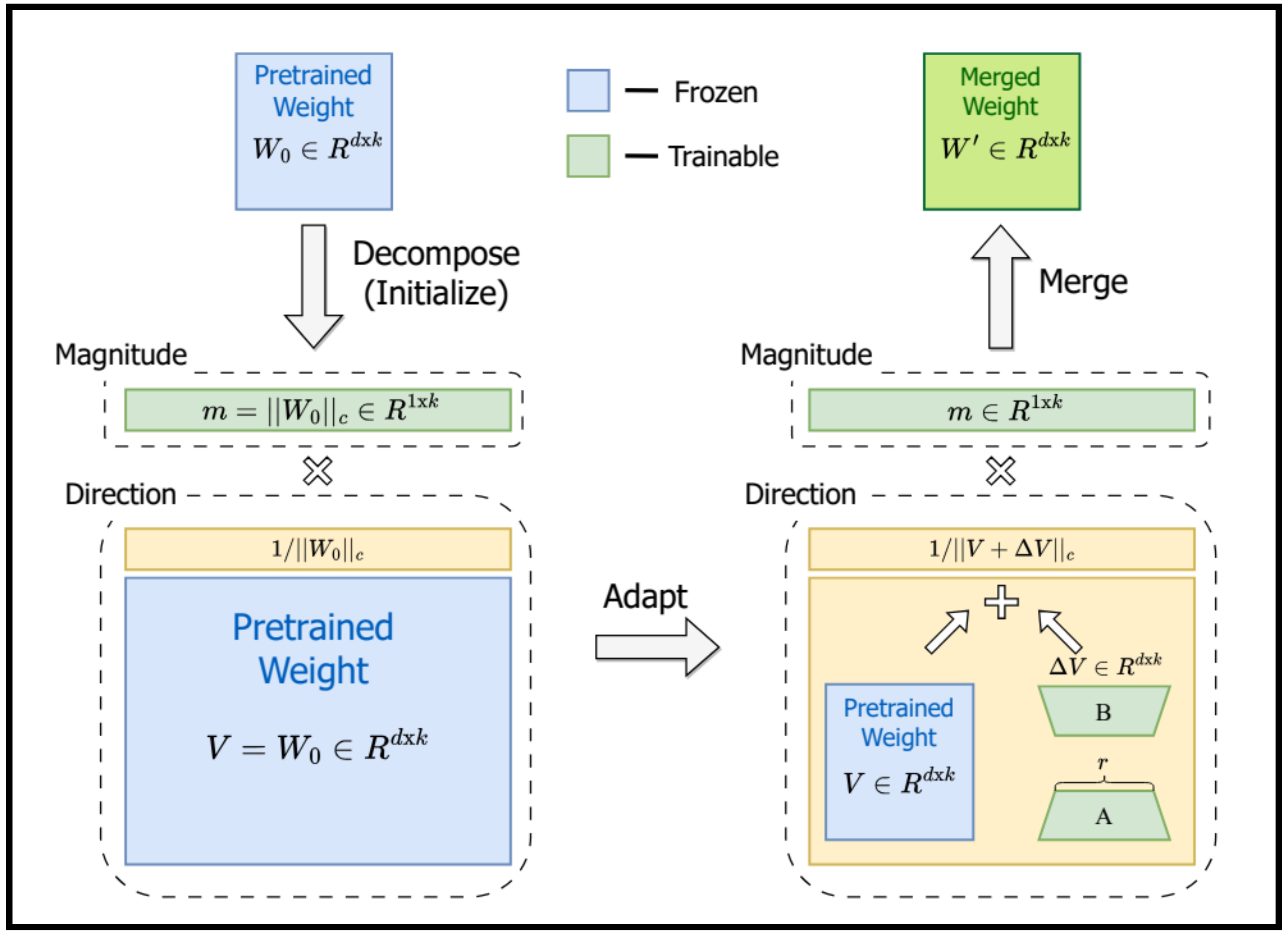


- Adapt the direction, not the magnitude
- See also weight-norm (2016)

Fundamental AI Lab

DoRA: Weight-Decomposed Low-Rank Adaptation. Liu et al. 2024
Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. Salimans & Kingma. NeurIPS 2016

DoRA: Weight-Decomposed Low-Rank Adaptation



- Adapt the direction, not the magnitude
- See also weight-norm (2016)

Table 5. Average scores on MT-Bench assigned by GPT-4 to the answers generated by fine-tuned LLaMA-7B/LLaMA2-7B.

Model	PEFT Method	# Params (%)	Score
LLaMA-7B	LoRA	2.31	5.1
	DoRA (Ours)	2.33	**5.5**
	VeRA	0.02	4.3
	DVoRA (Ours)	0.04	**5.0**
LLaMA2-7B	LoRA	2.31	5.7
	DoRA (Ours)	2.33	**6.0**
	VeRA	0.02	5.5
	DVoRA (Ours)	0.04	**6.0**

- Combinable with VeRA

Fundamental AI Lab

DoRA: Weight-Decomposed Low-Rank Adaptation. Liu et al. 2024
Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. Salimans & Kingma. NeurIPS 2016

Code: https://github.com/huggingface/peft

Code: https://github.com/huggingface/peft

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

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

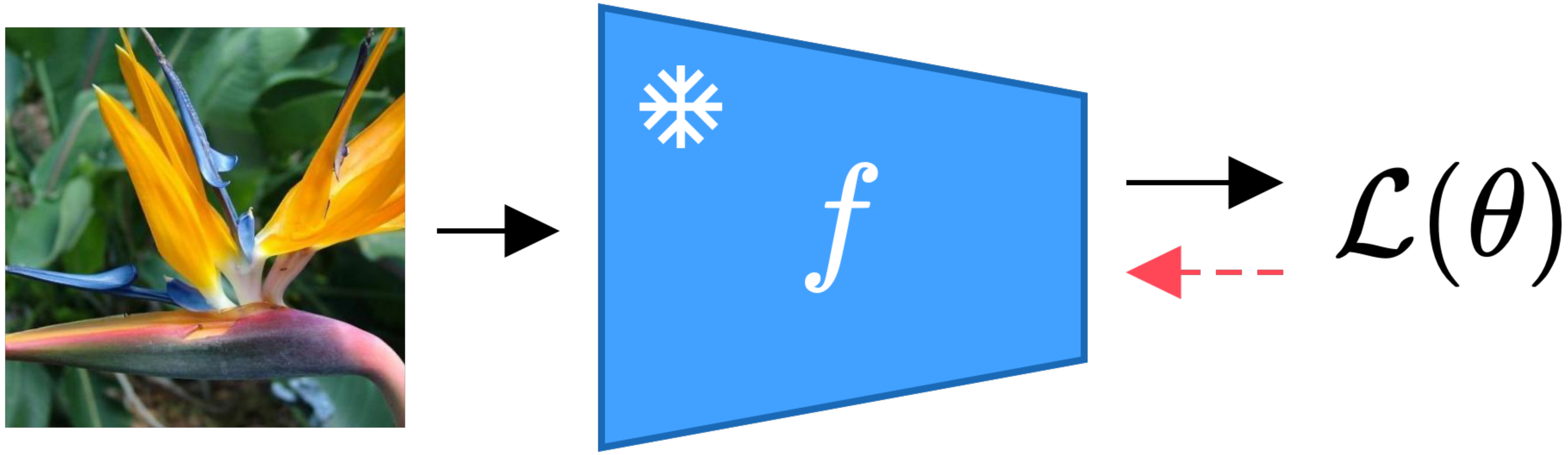
NeurIPS 2024

Fundamental AI Lab

Valeo

Idea

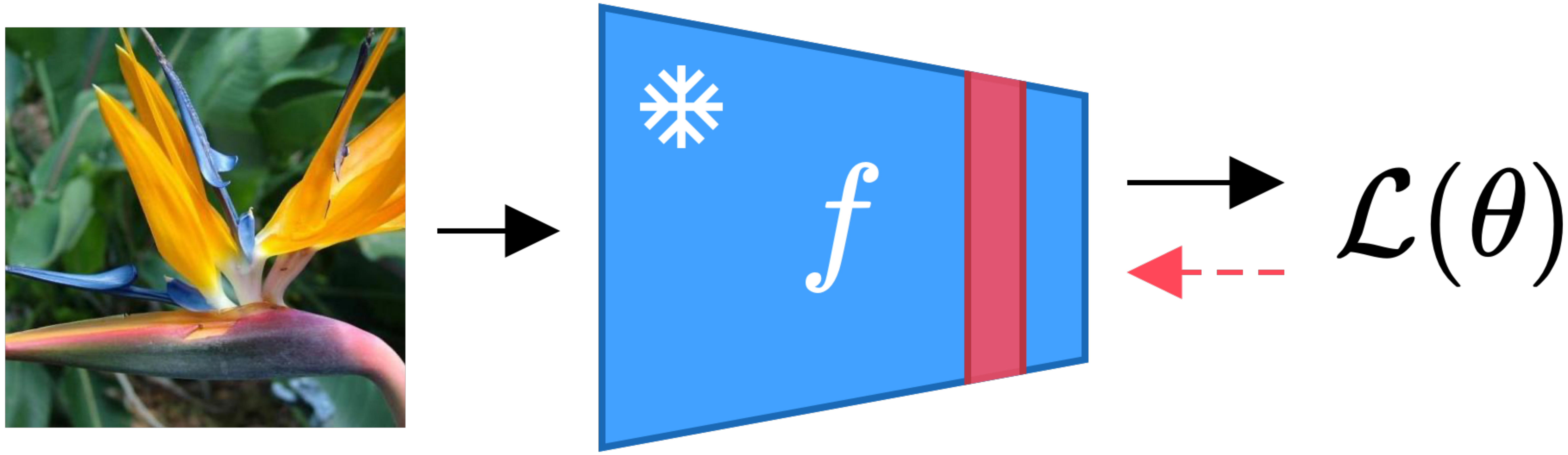
The **loss** indicates how the network output should **change** to solve a task



Fundamental AI Lab

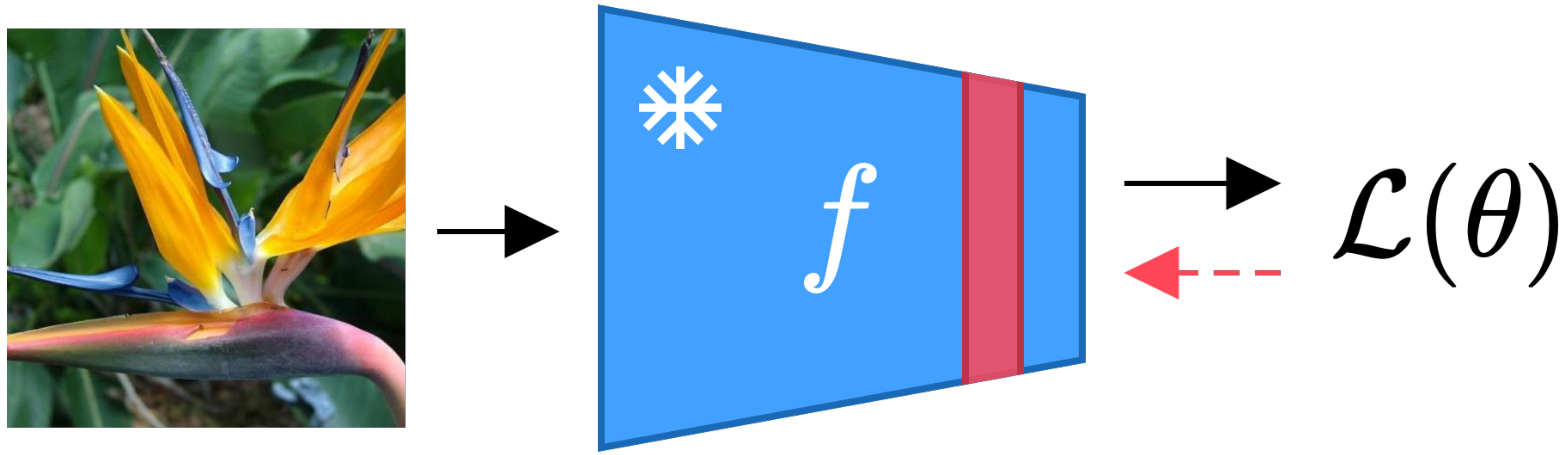
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Idea



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

Idea



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

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

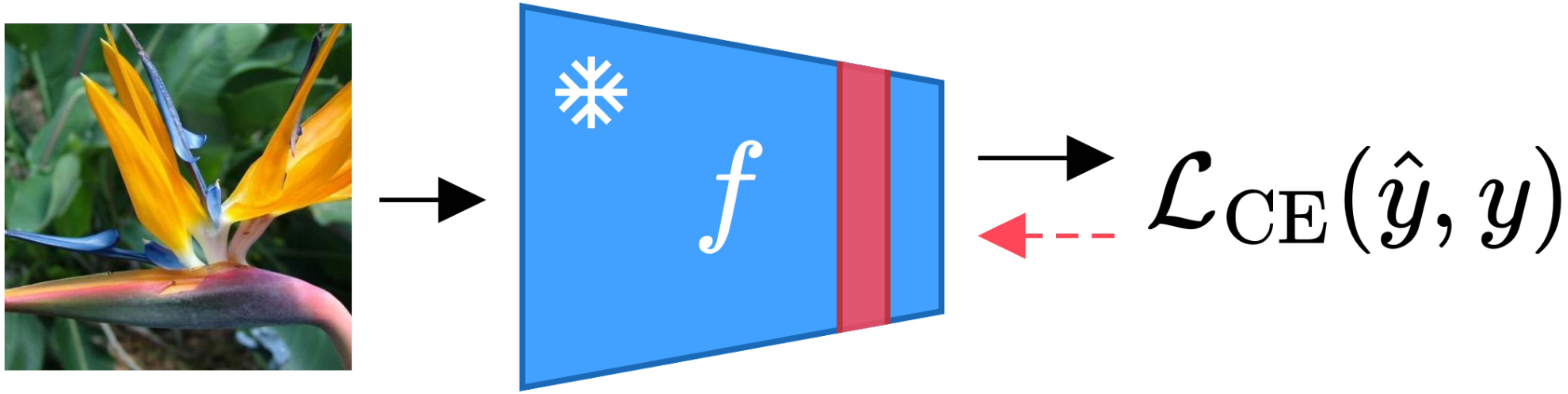
Fundamental AI Lab

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 😢



Fundamental AI Lab

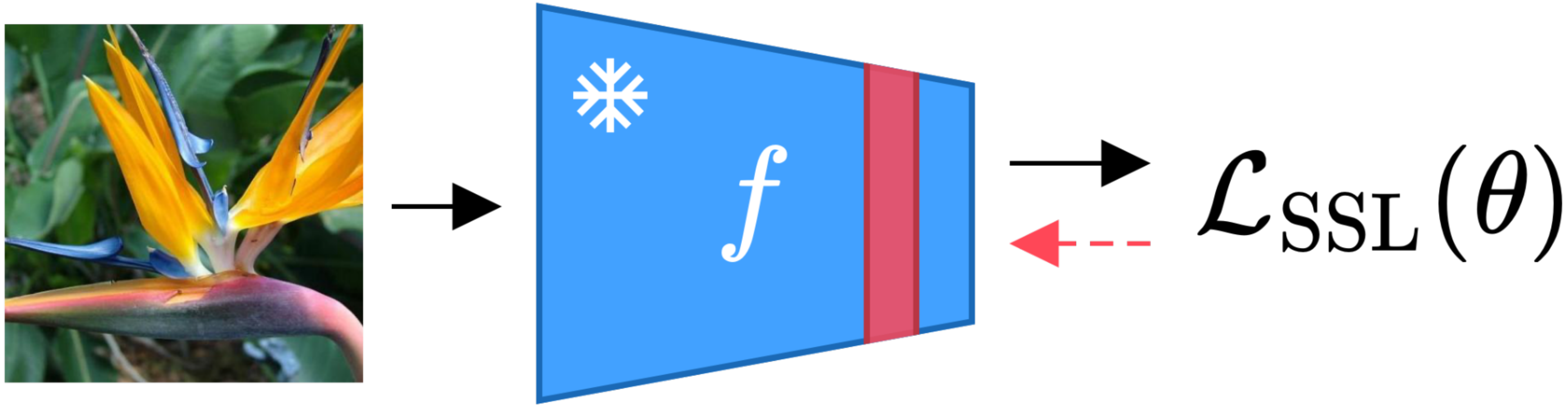
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Idea

Self Supervised Learning to the rescue! 🎉

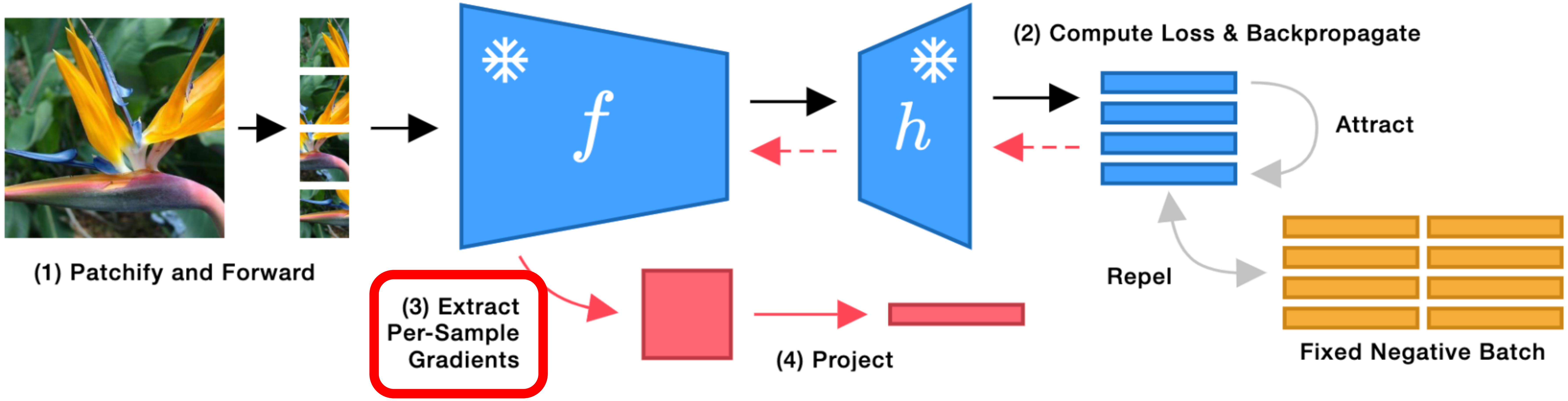
No Labels

Several Proxylosses



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

Method

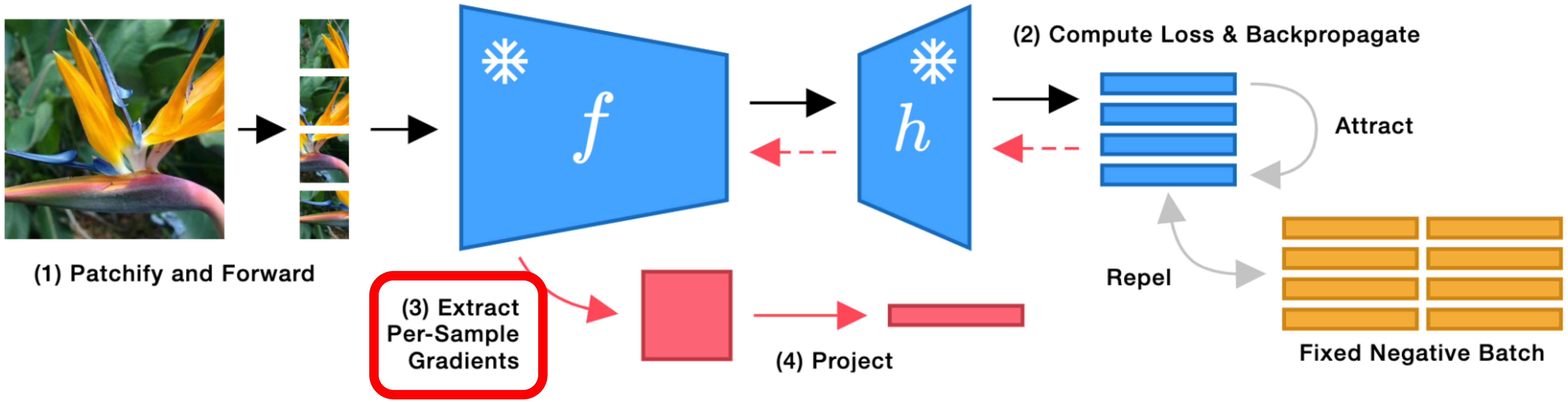


Fundamental AI Lab

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

Method

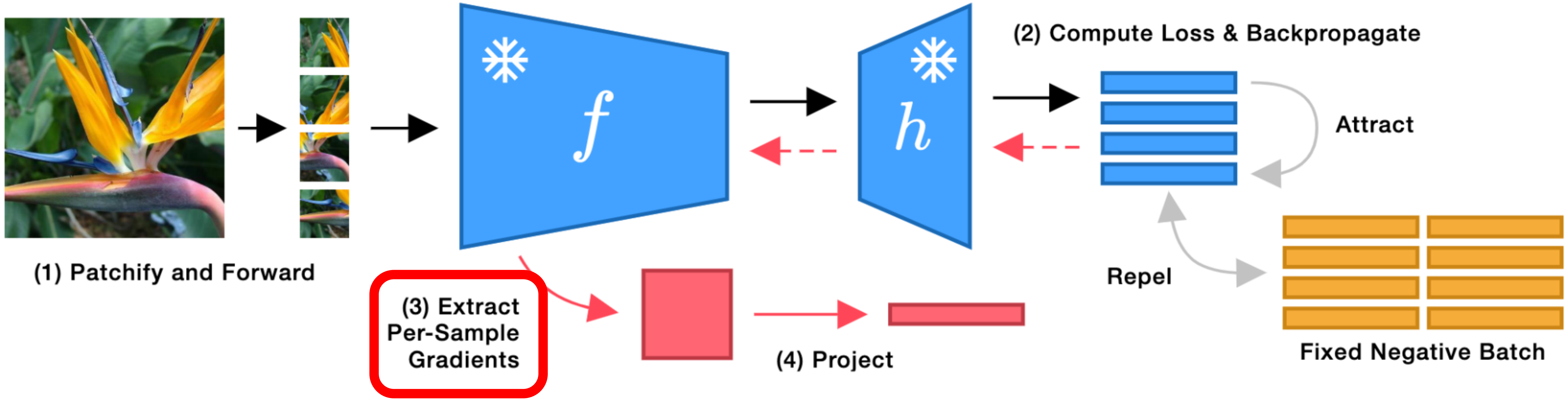
Given a pre-trained vision transformer we



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).



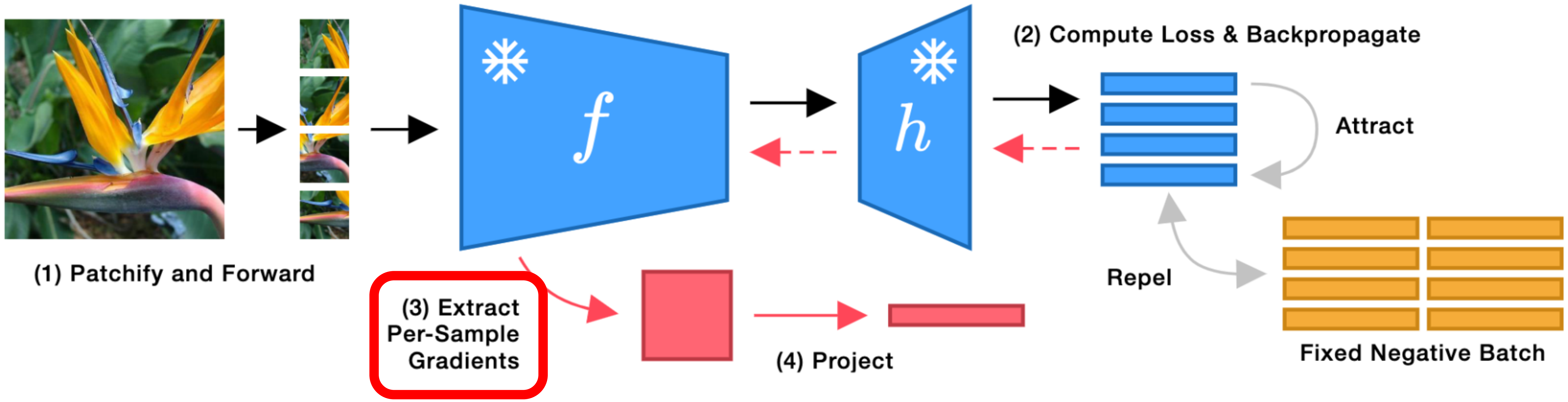
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Method

Given a pre-trained vision transformer we

Forward an image (or multiple views of it).

Compute a self-supervised loss & backpropagate.



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

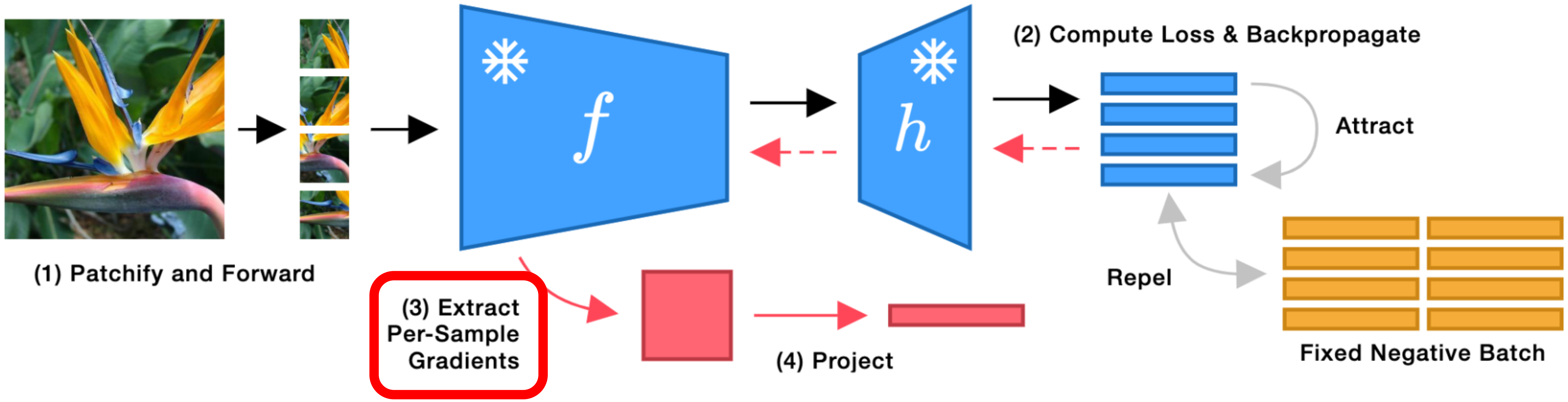
Method

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

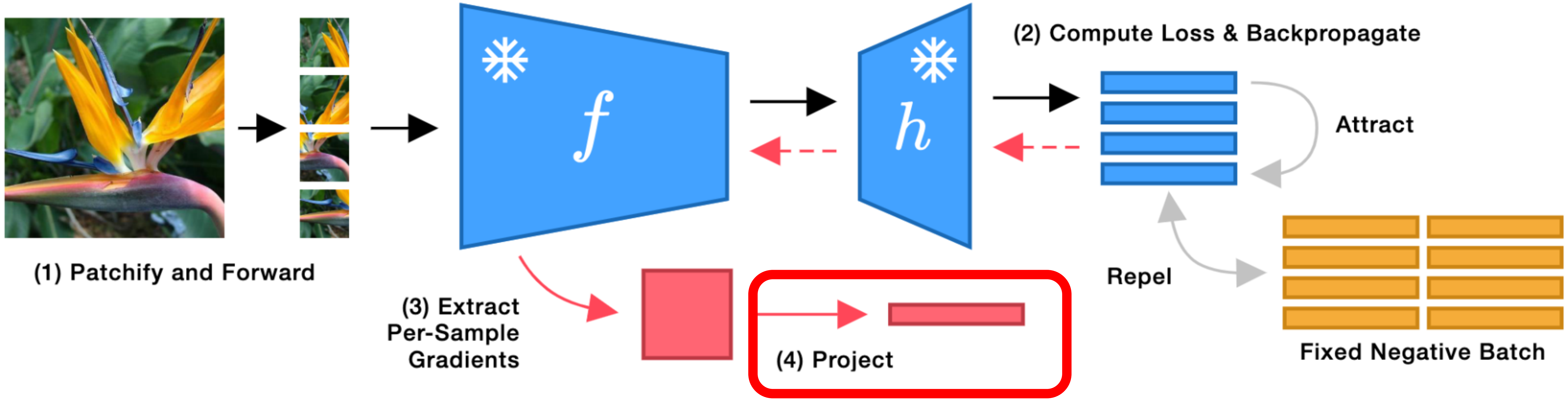
Given a pre-trained vision transformer we

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.

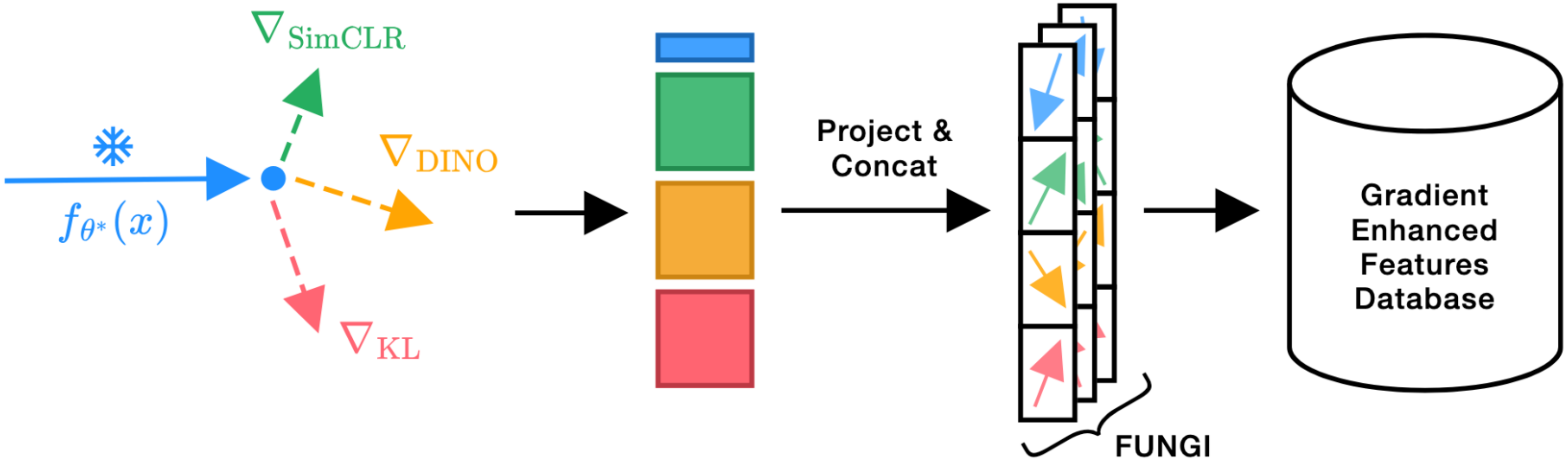
Project gradients and obtain a FUNGI (Feature from UNsupervised GradIents).



Fundamental AI Lab

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

Self-Supervised Objectives

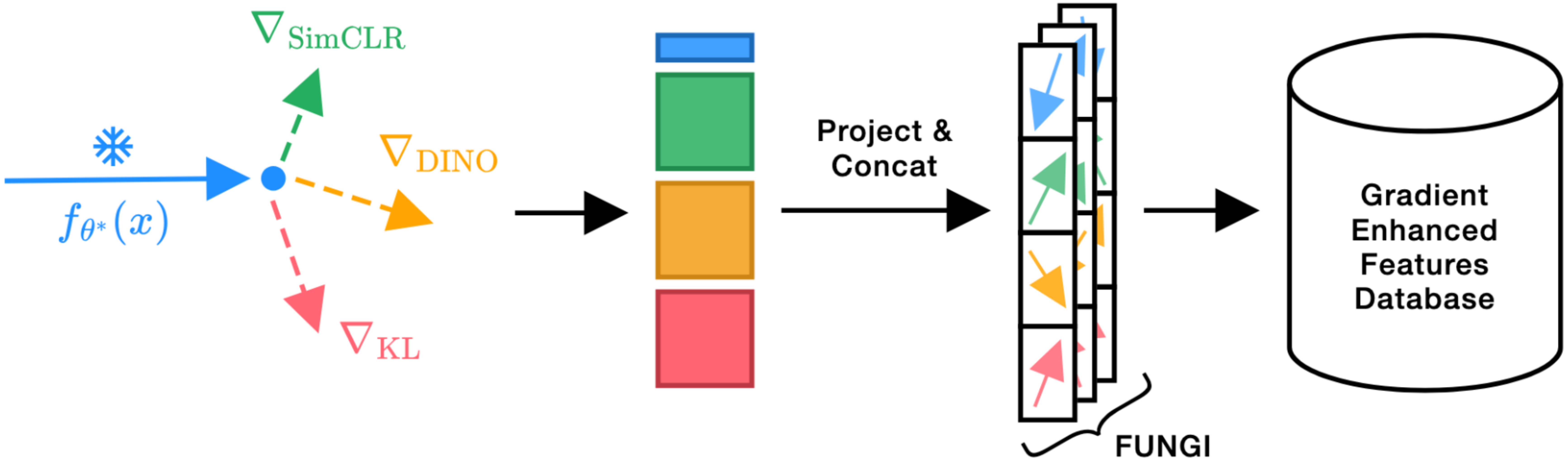


Fundamental
AI Lab

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.

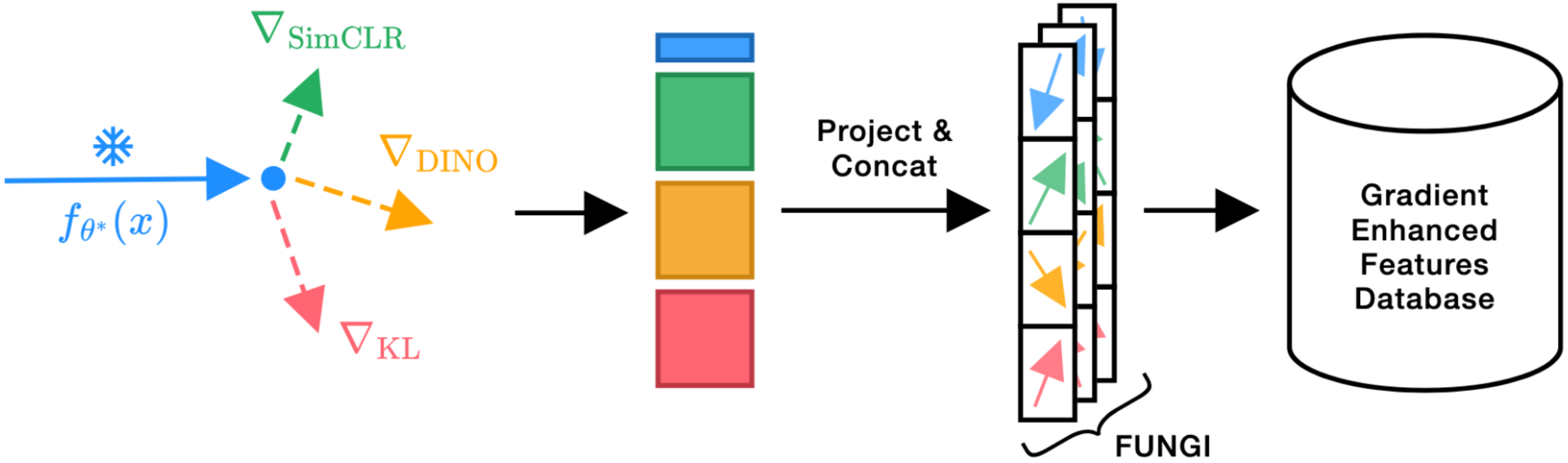


Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Self-Supervised Objectives

Three objectives: DINO, SimCLR and KL.

We concatenate (multiple) gradients and the model embeddings.



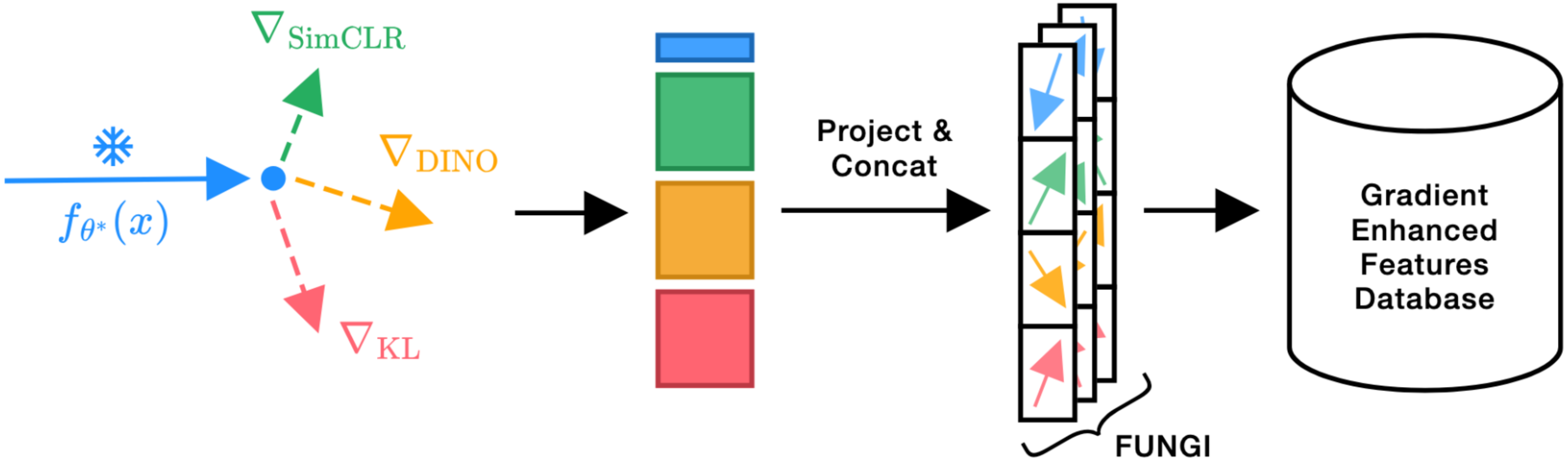
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

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.



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

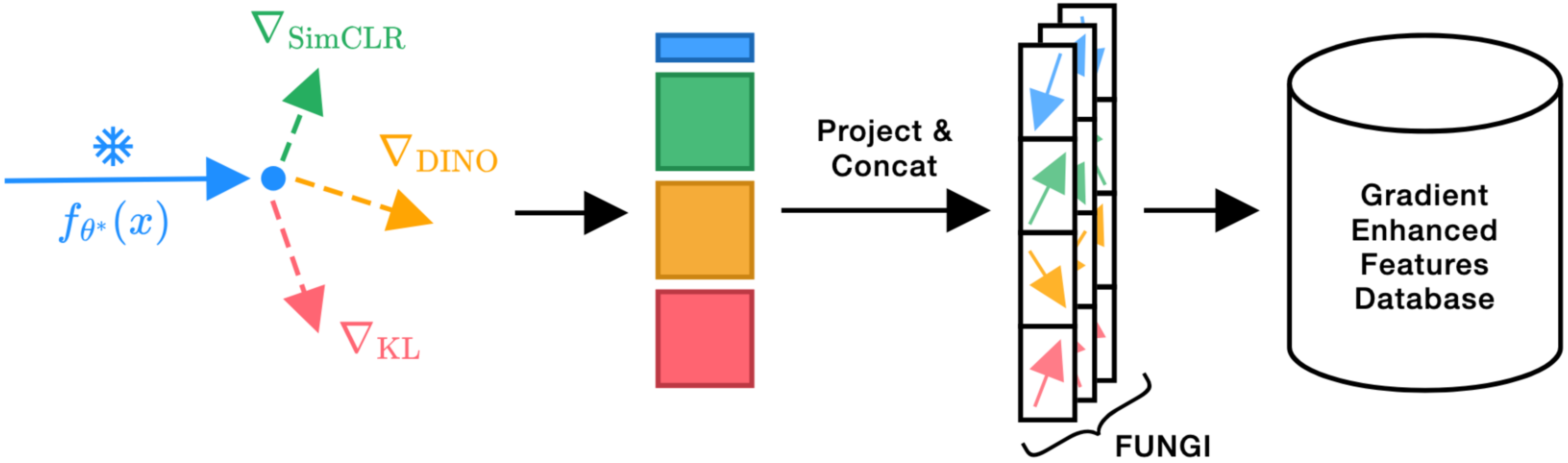
Self-Supervised Objectives

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

Fundamental AI Lab

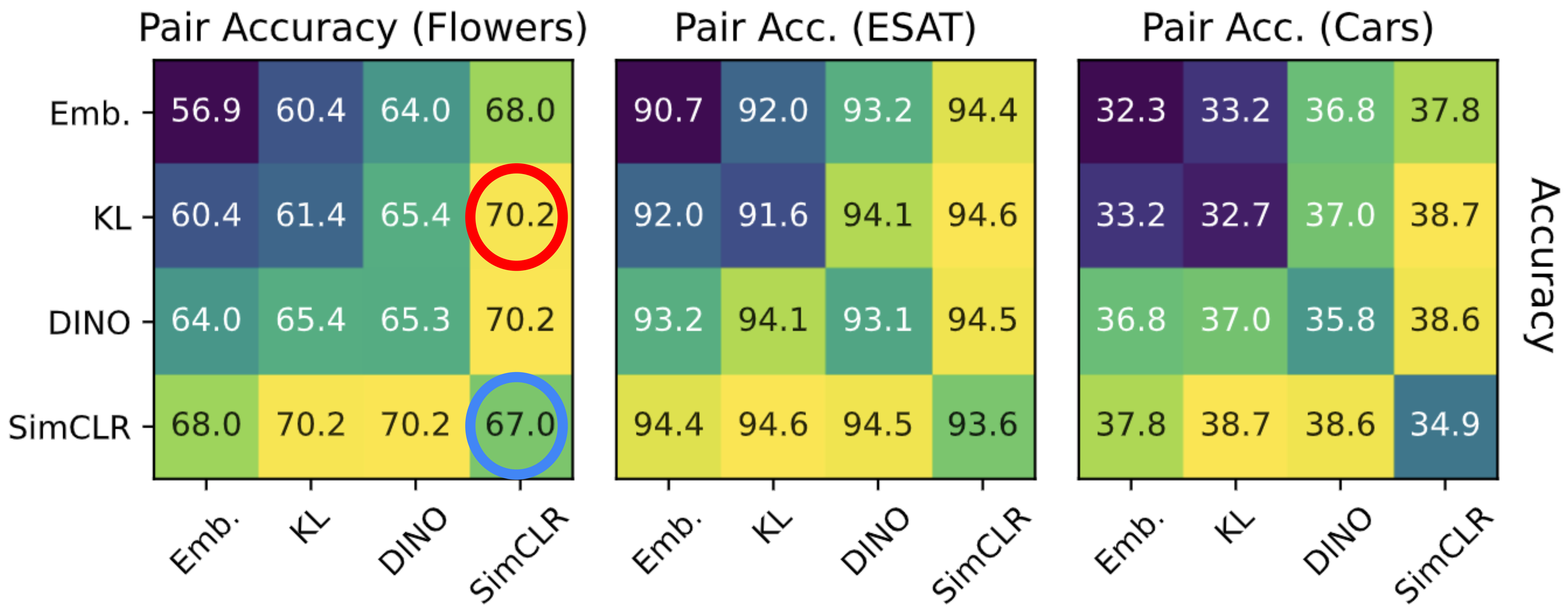
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

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



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

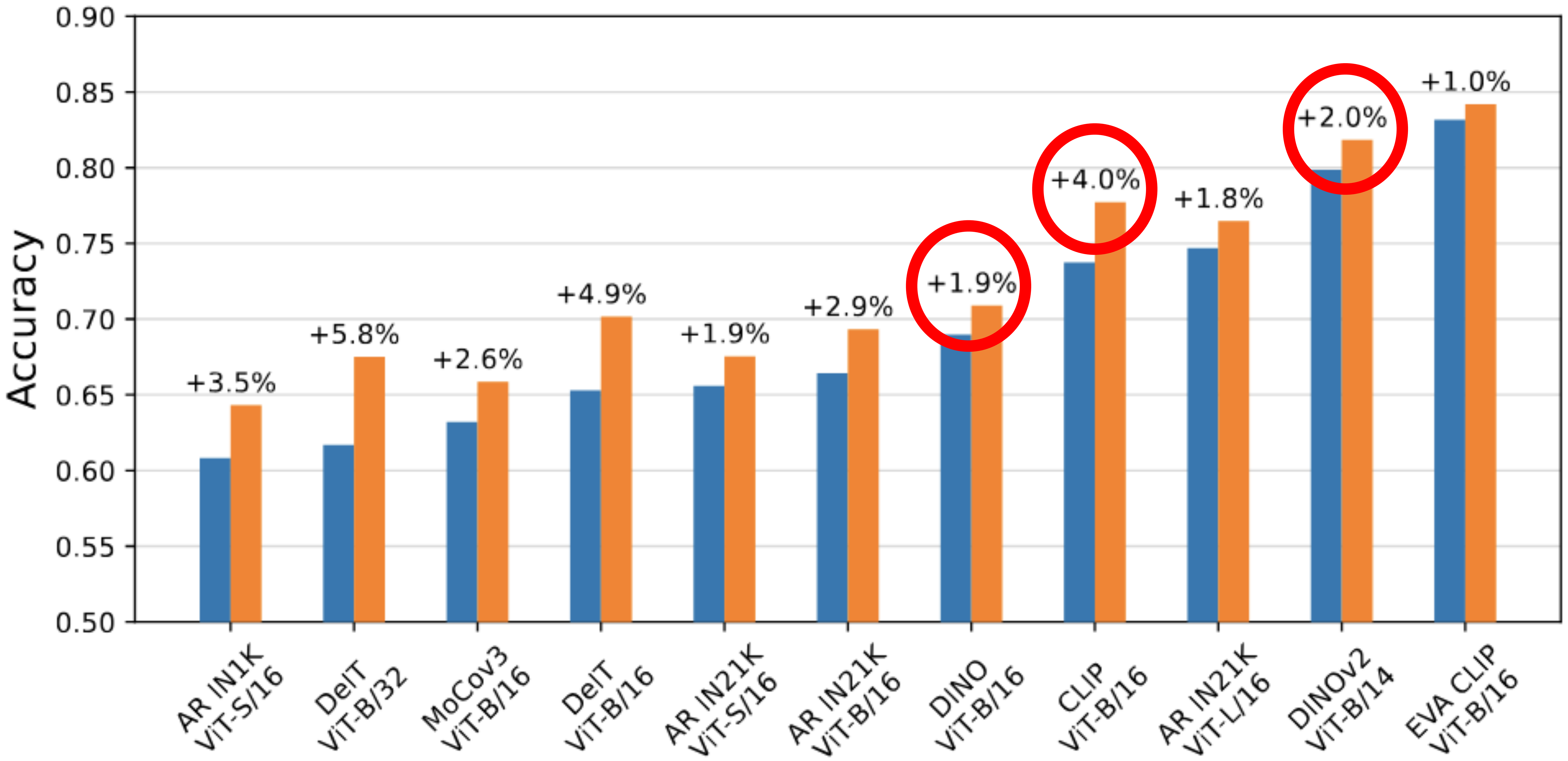
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

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

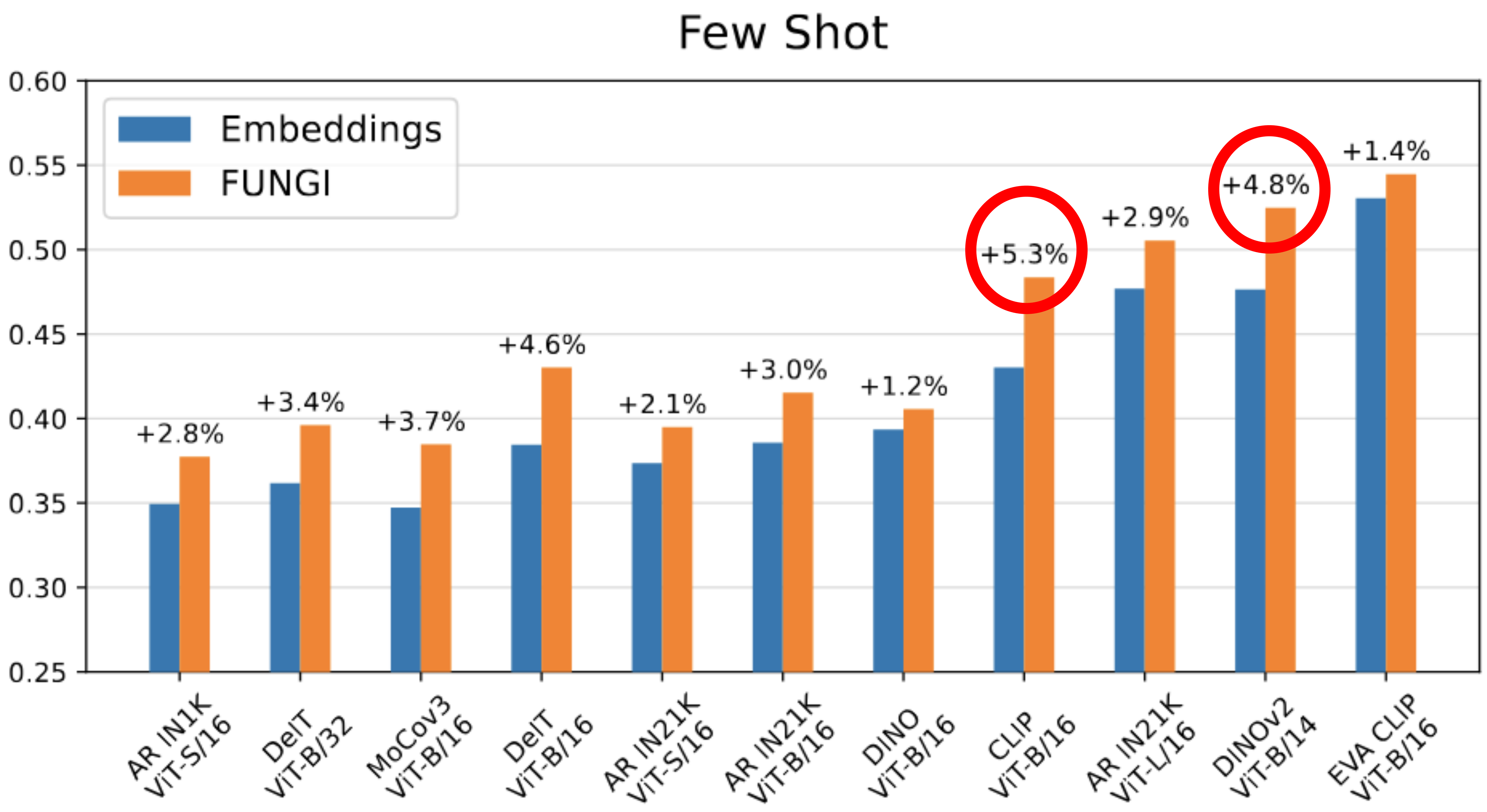


Fundamental AI Lab

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

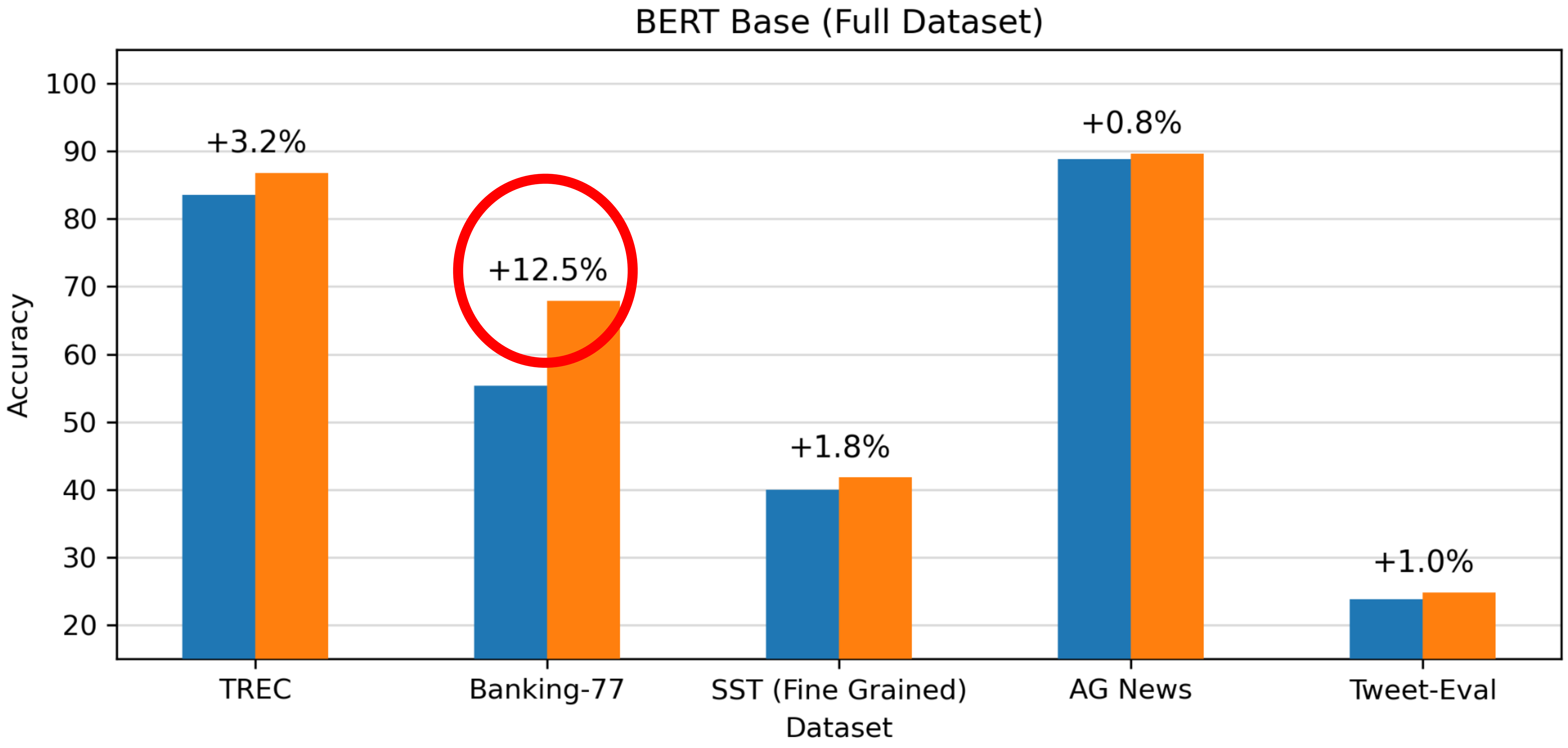


Fundamental AI Lab

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

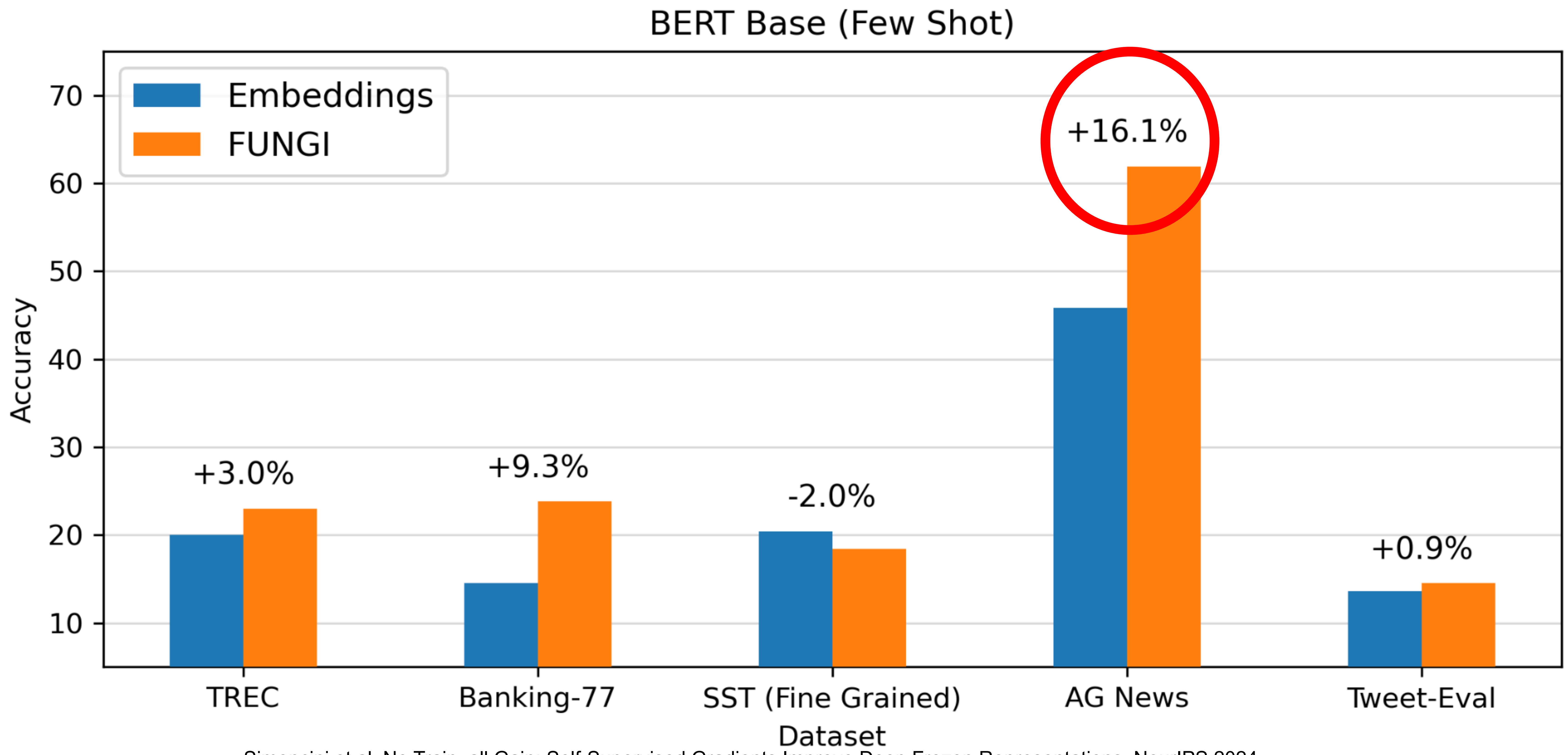


Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Fundamental AI Lab

k-nn classification (language)

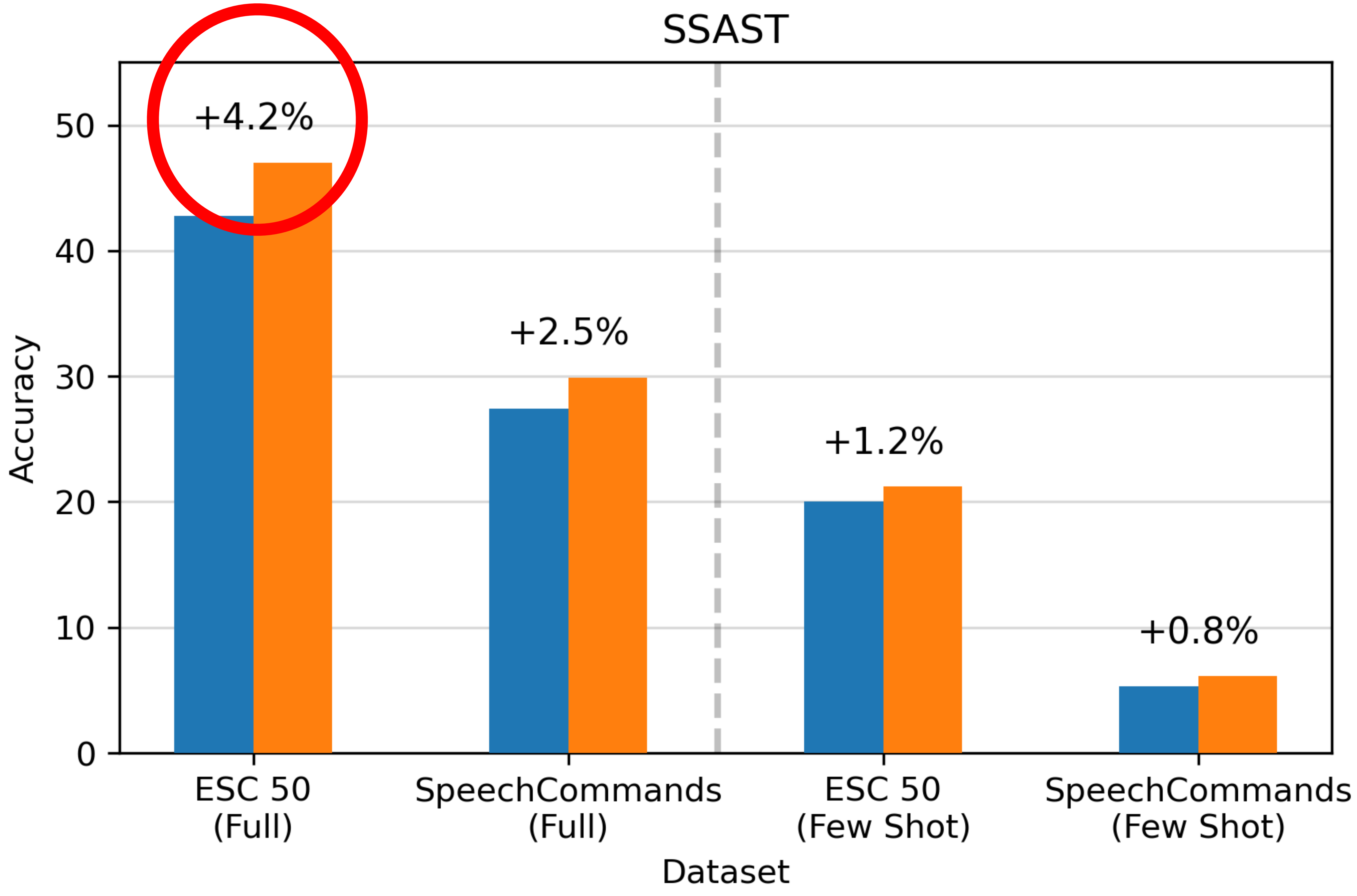
Up to **16%** better in few shot classification using BERT Base



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



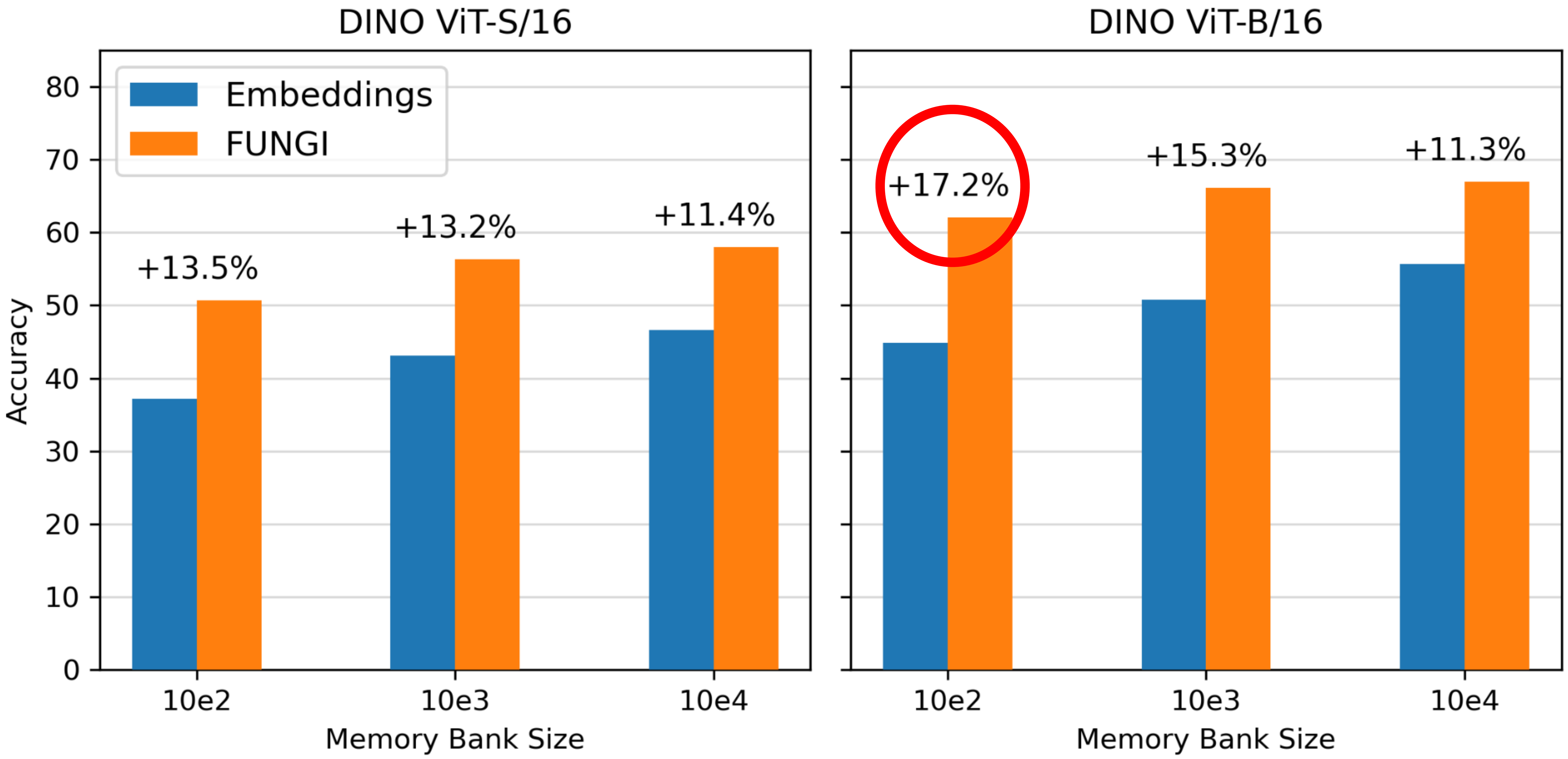
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

Visual In-Context Segmentation

Fundamental
AI Lab

In-Context Semantic Segmentation (Hummingbird) on Pascal VOC

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

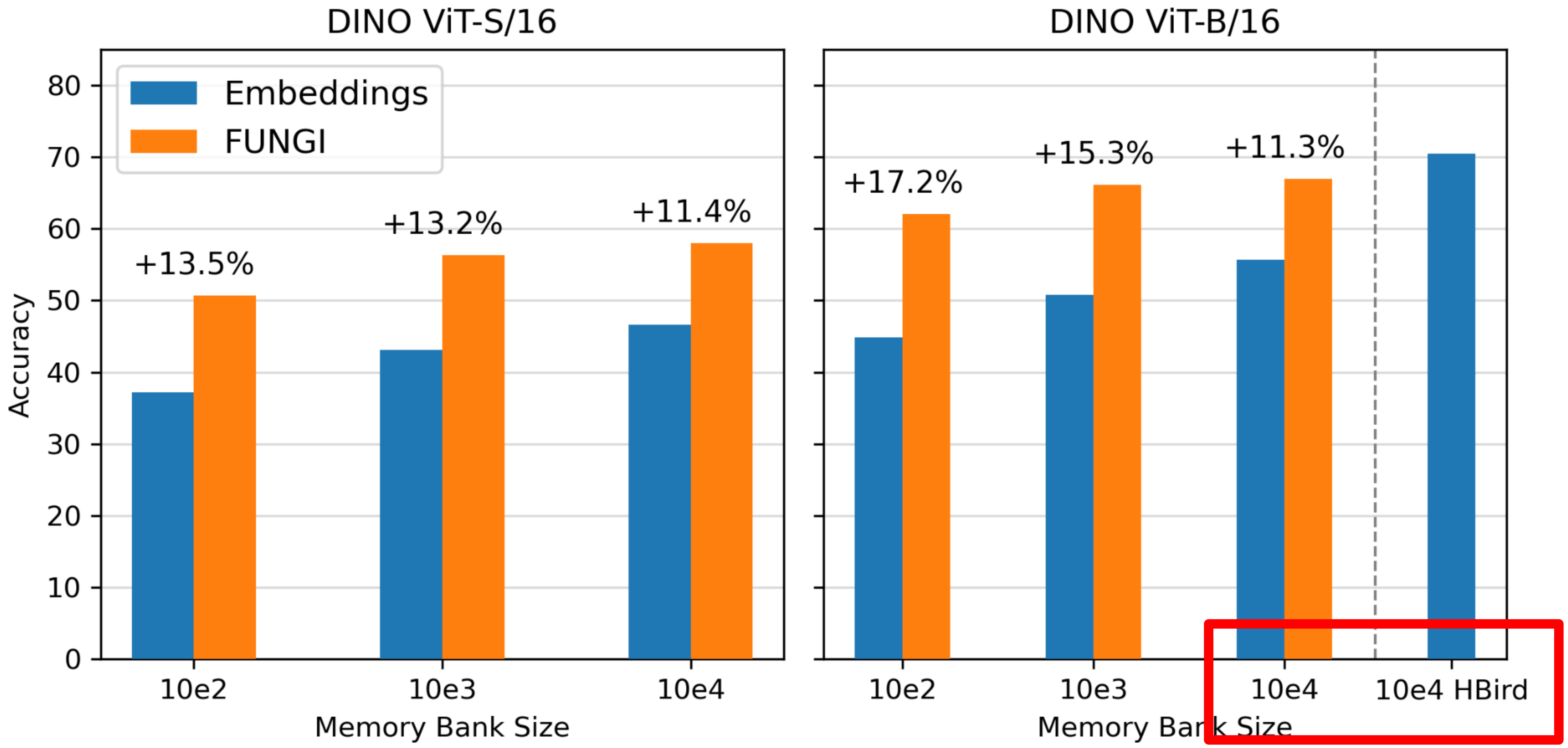


Fundamental AI Lab

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

In-Context Semantic Segmentation on Pascal VOC

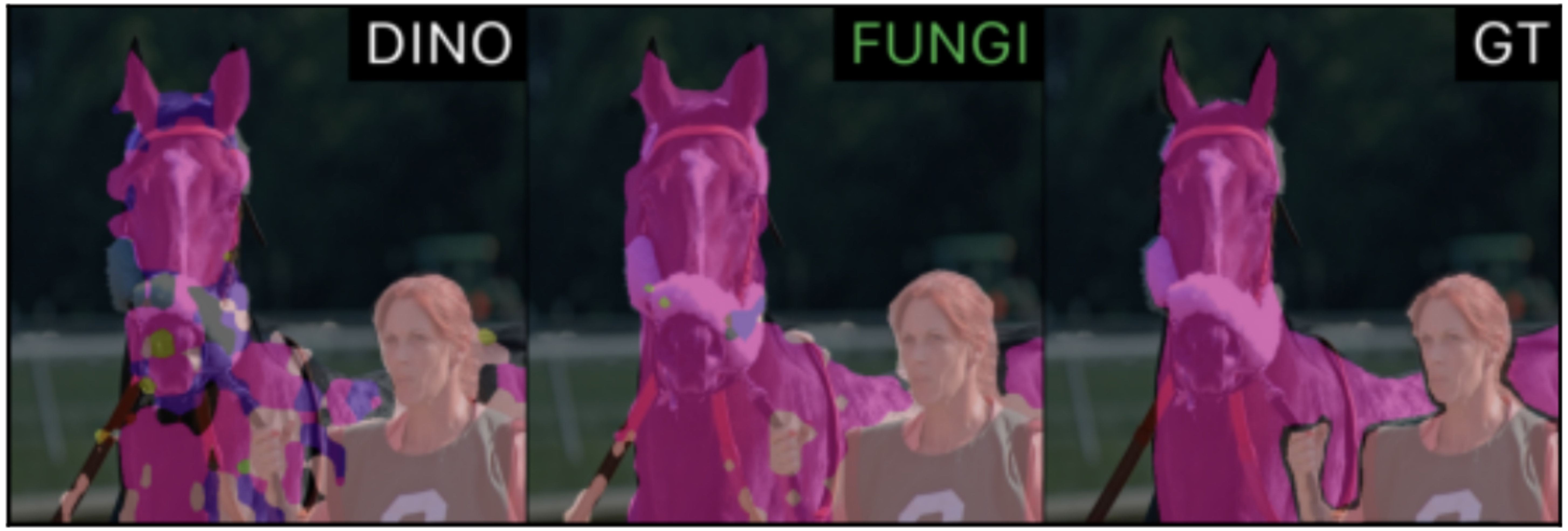
Close to SoTA, **without any training!**



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



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	$\boxed{\textbf{91.2} \uparrow 2.5}$

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

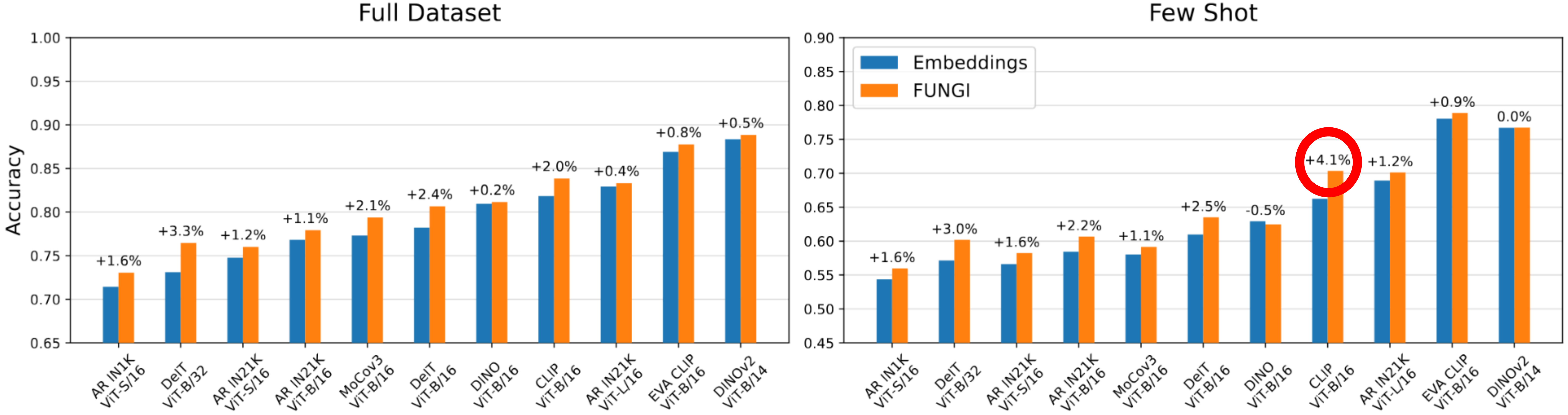


Figure 10: FUNGI works across backbones for linear probing. Accuracy in logistic regression-based image classification of embeddings versus FUNGI features on various ViT backbones, both for full dataset and few shot setups, averaged over 11 datasets. For the FUNGI features, we chose the best performing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

Fundamental AI Lab

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

Summary

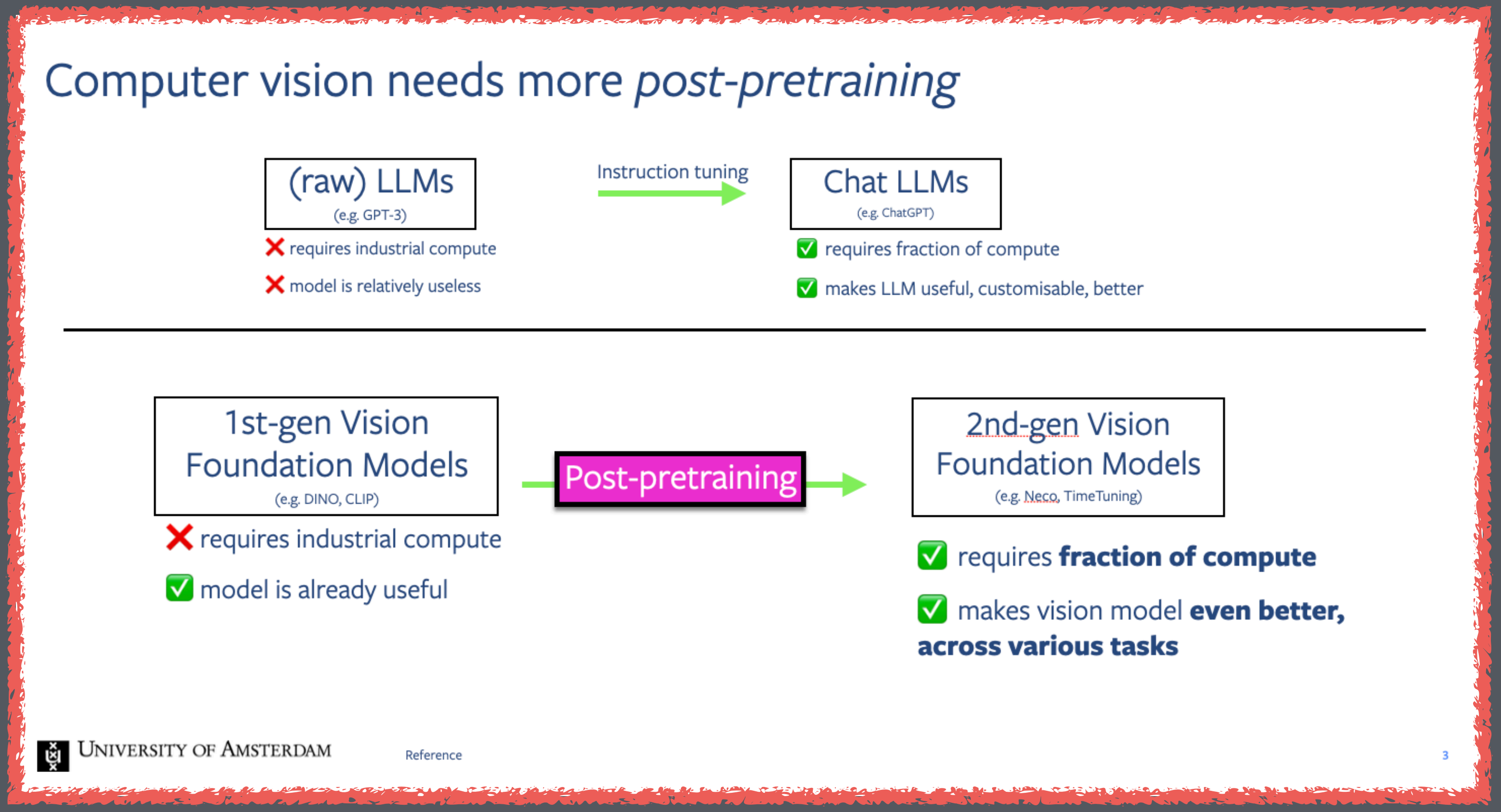
Self-supervised **gradients can be used as features**, and can perform better than the embeddings

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

FUNGI works **across modalities**

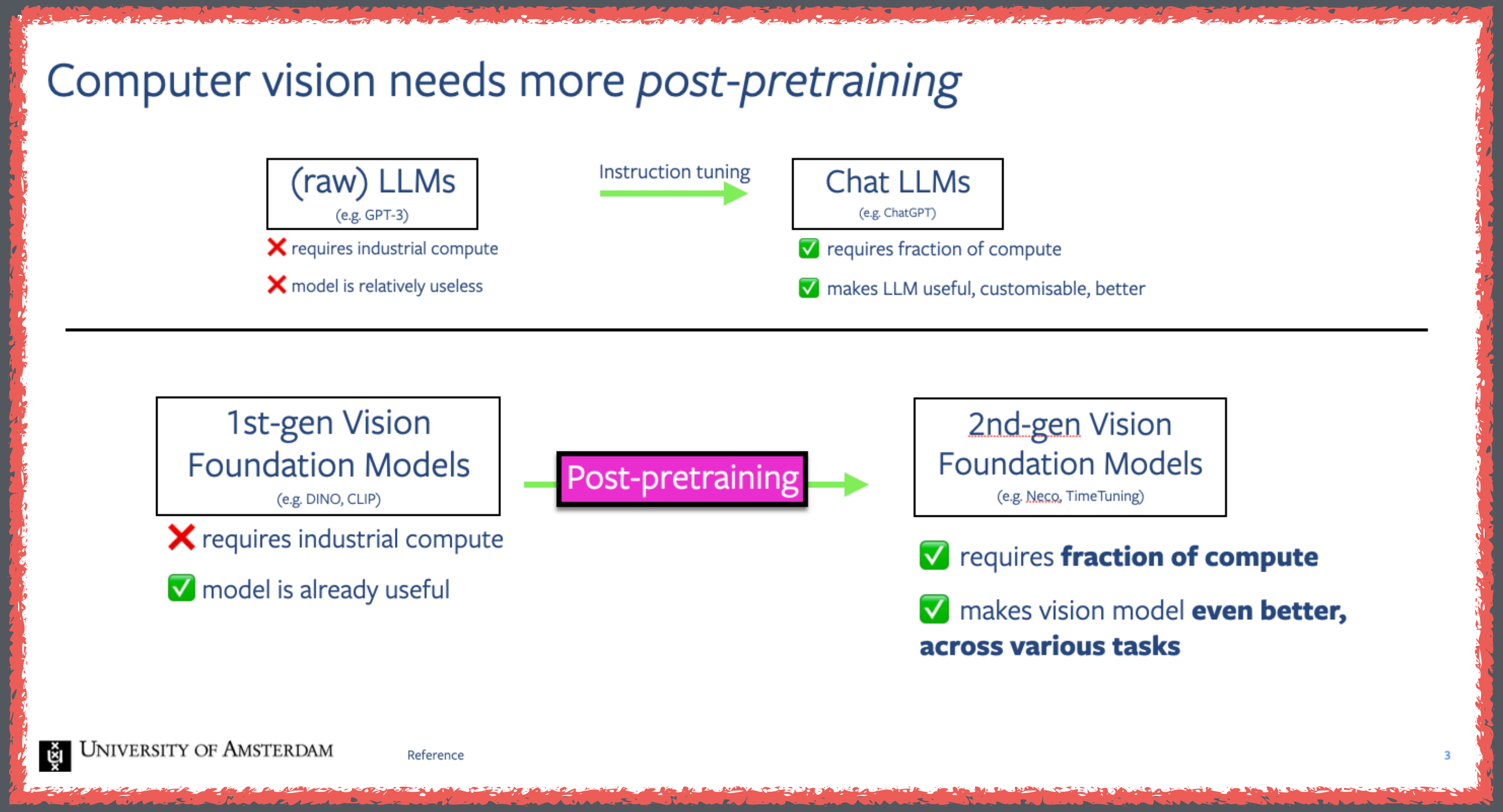
Simoncini et al. No Train, all Gain: Self-Supervised Gradients Improve Deep Frozen Representations, NeurIPS 2024

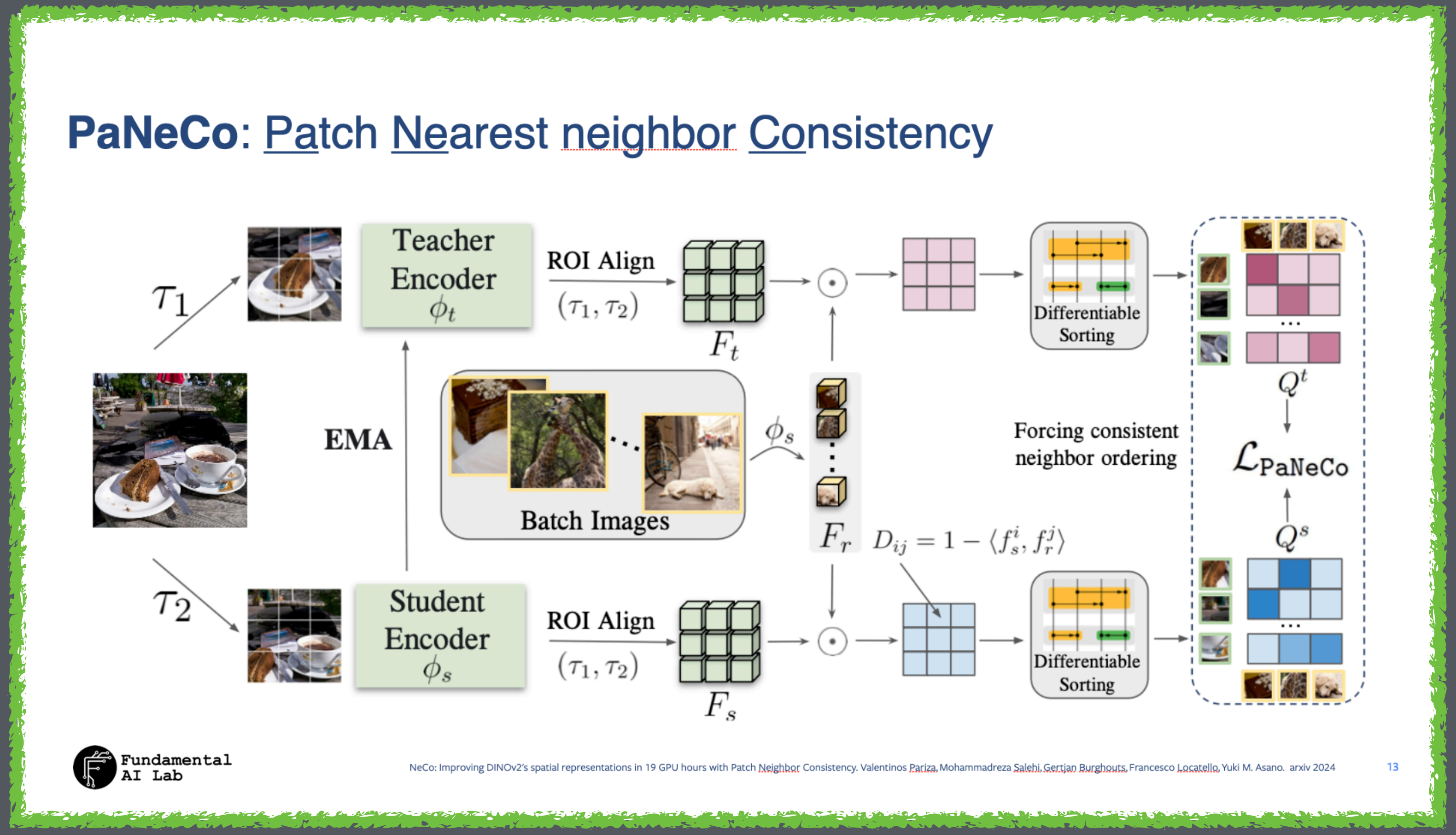
There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.



There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

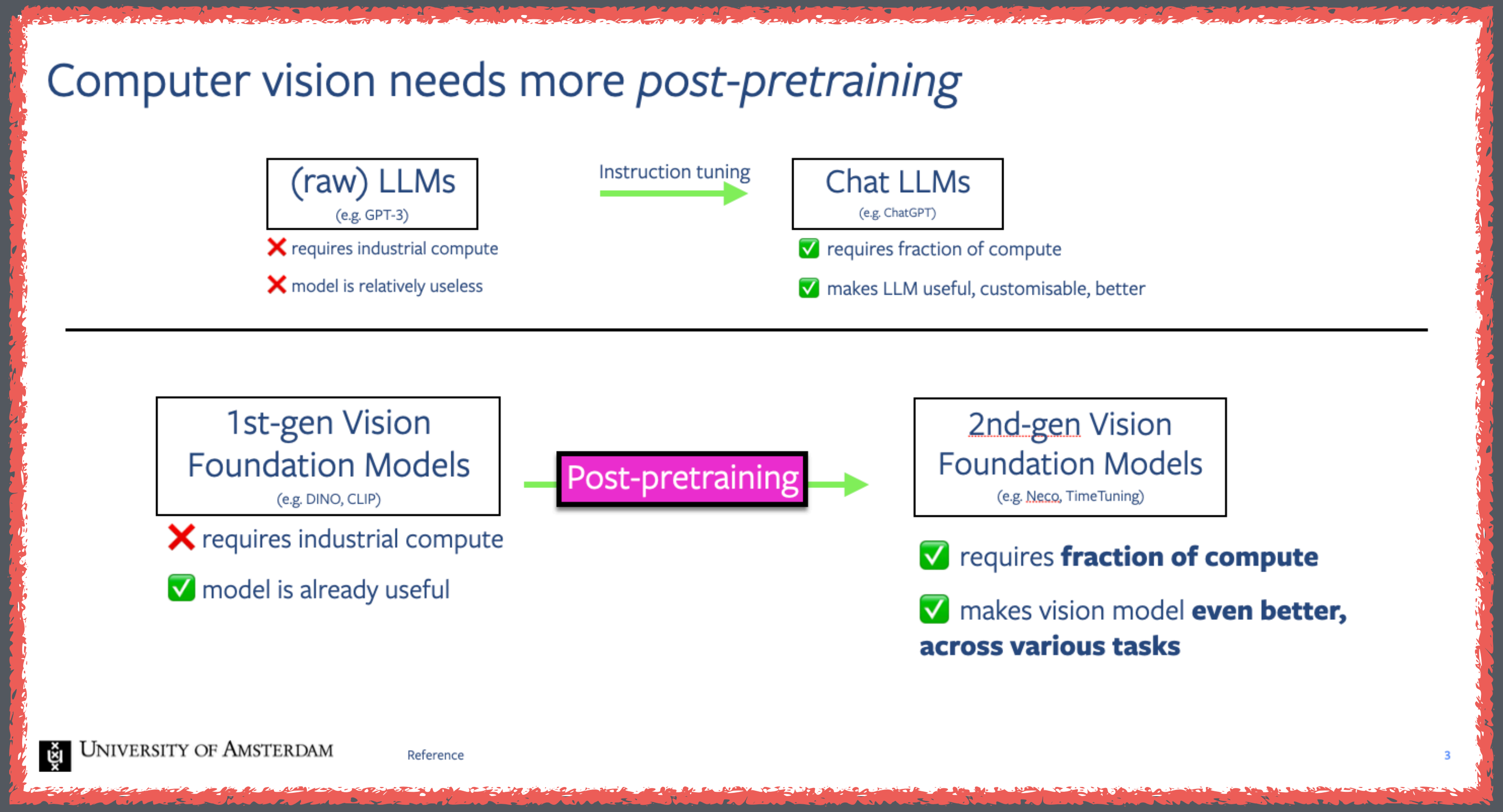
Fundamental AI Lab

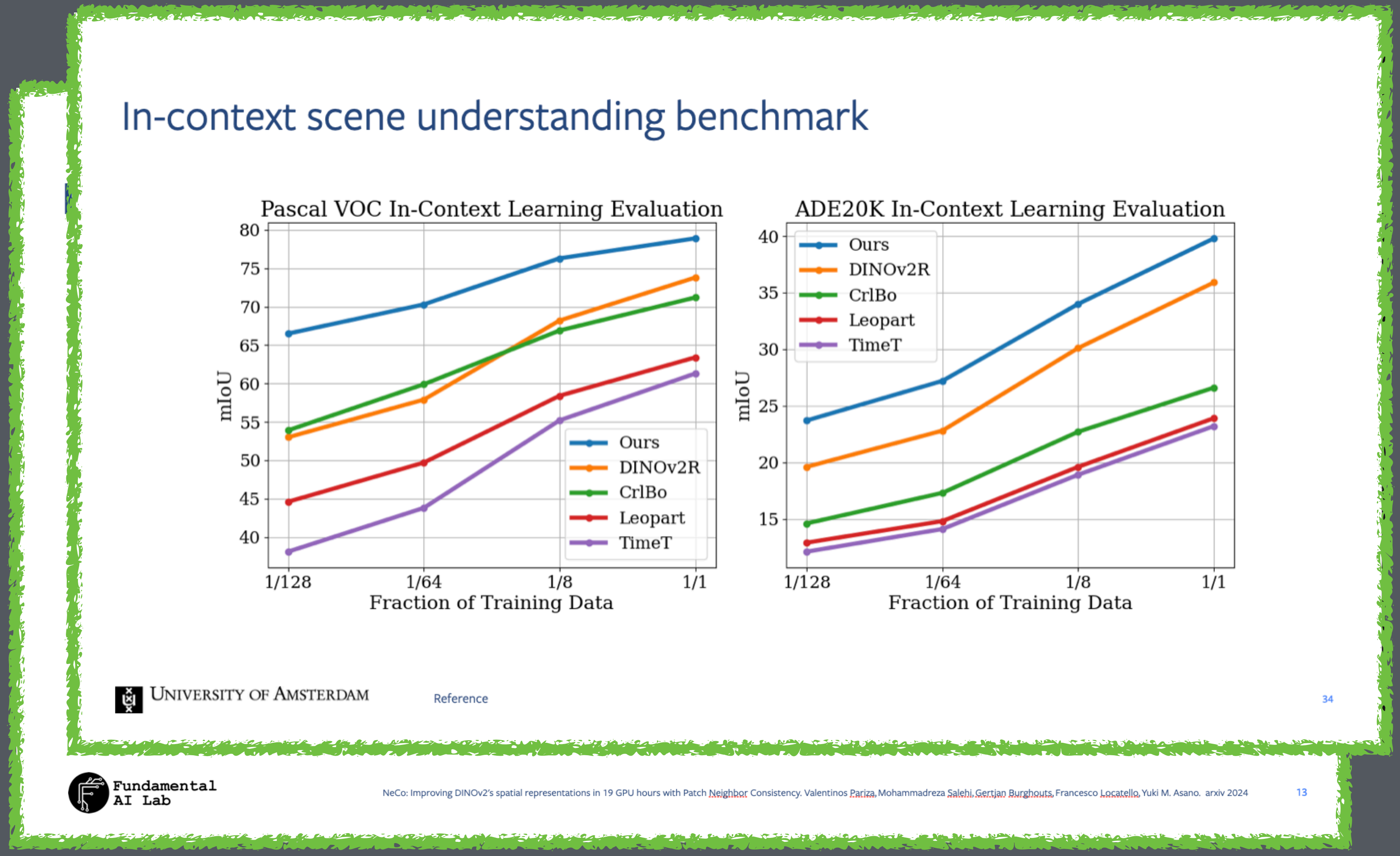




There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

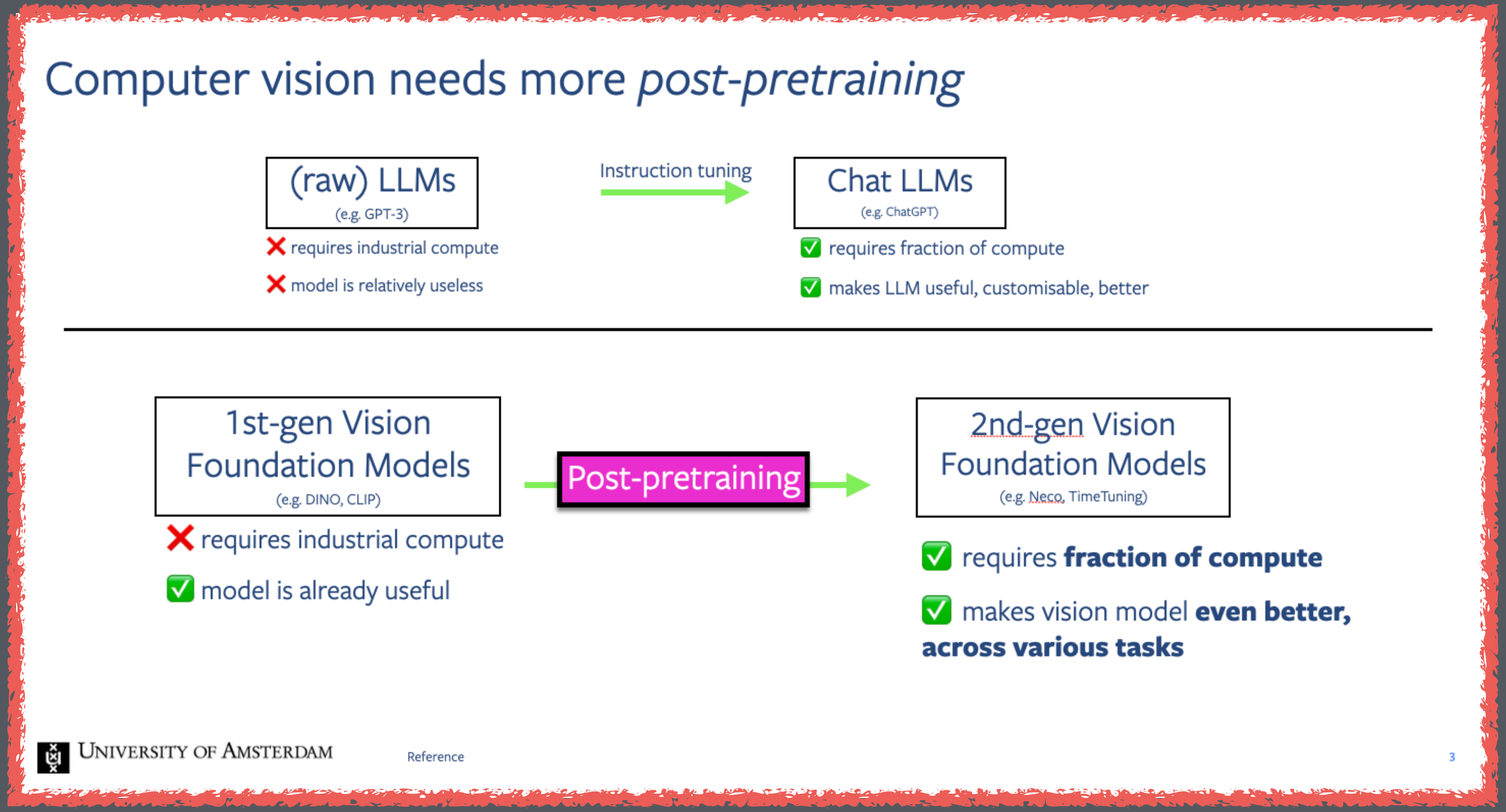
Fundamental AI Lab

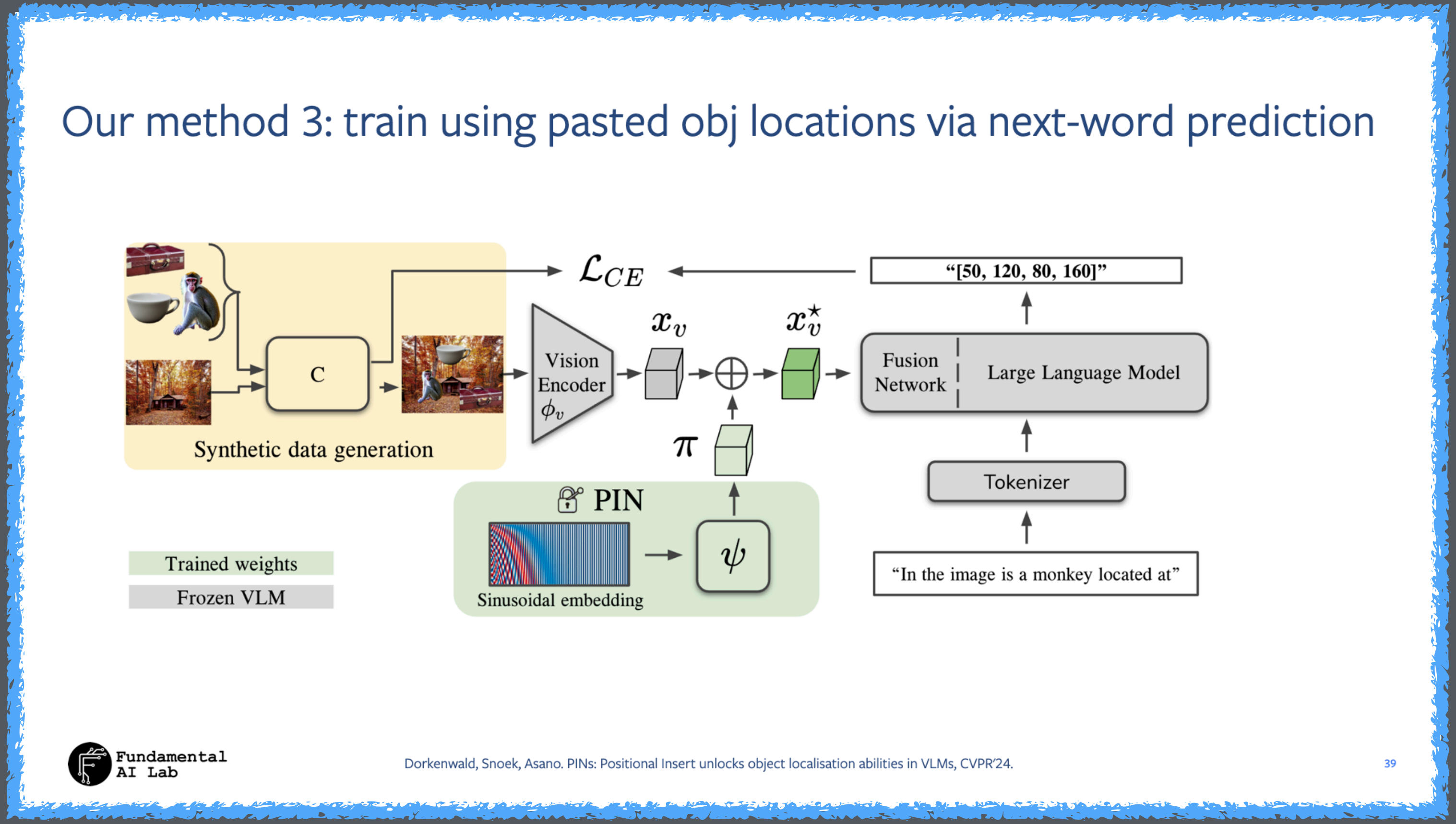


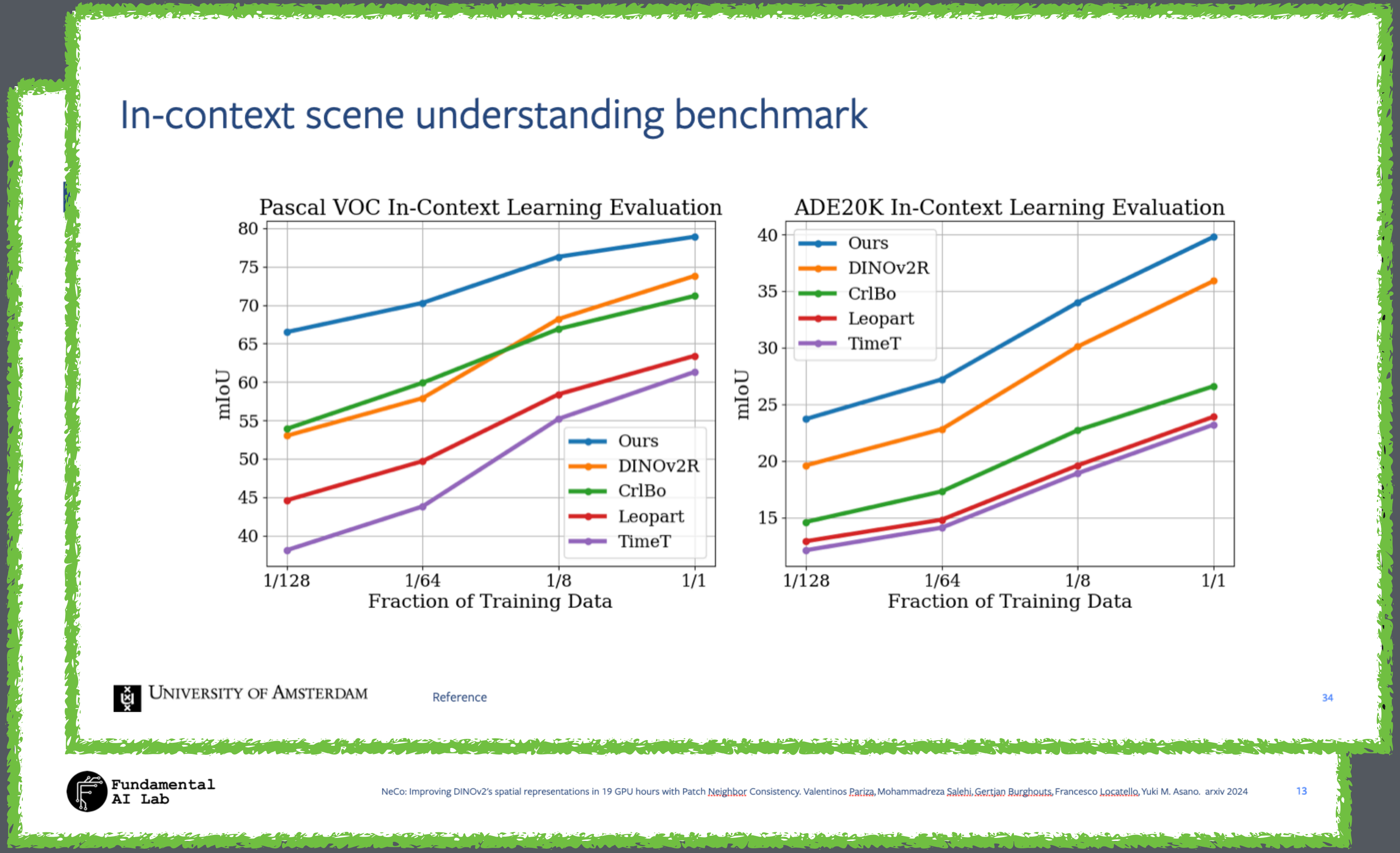


There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

Fundamental AI Lab

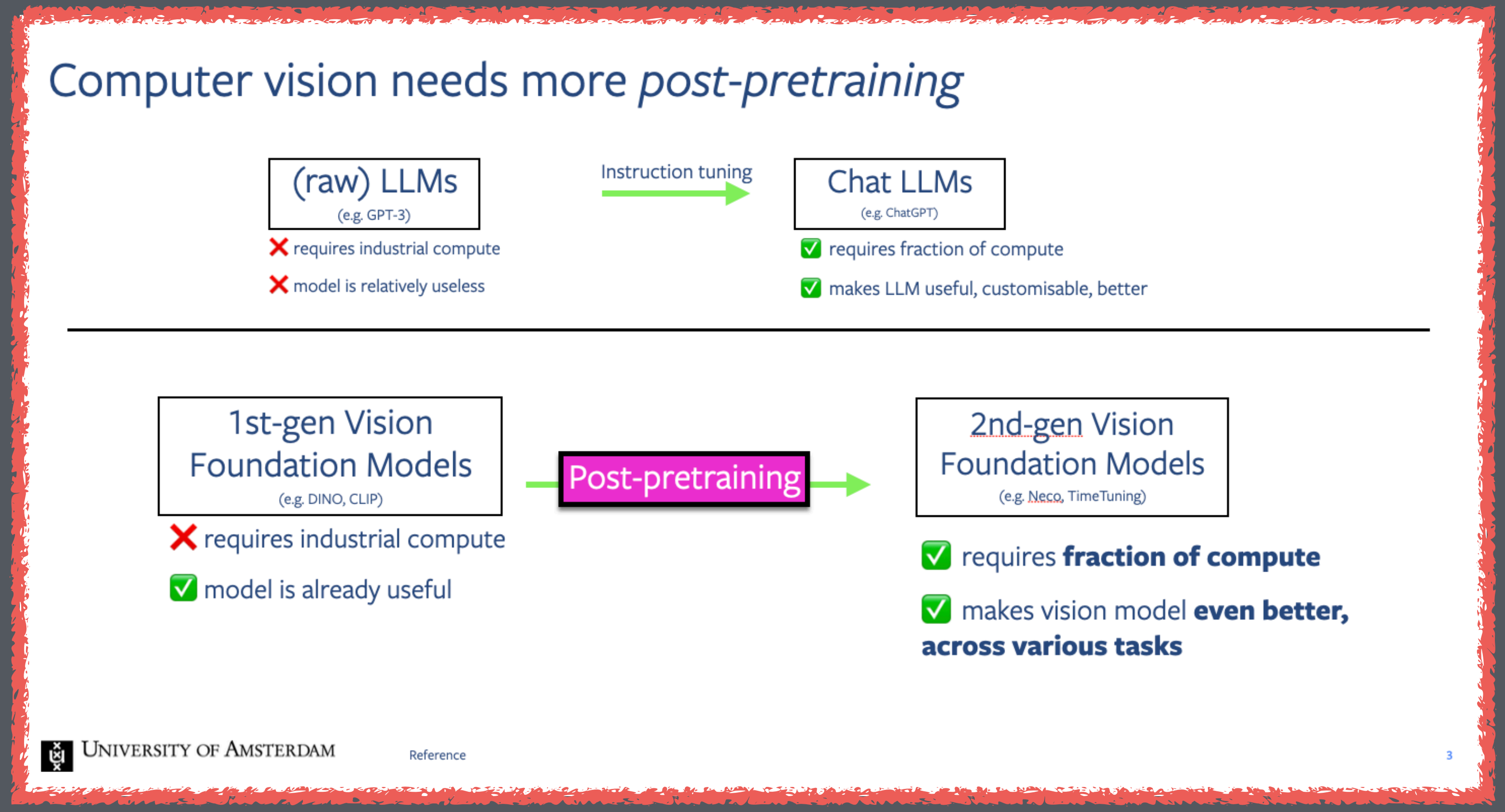


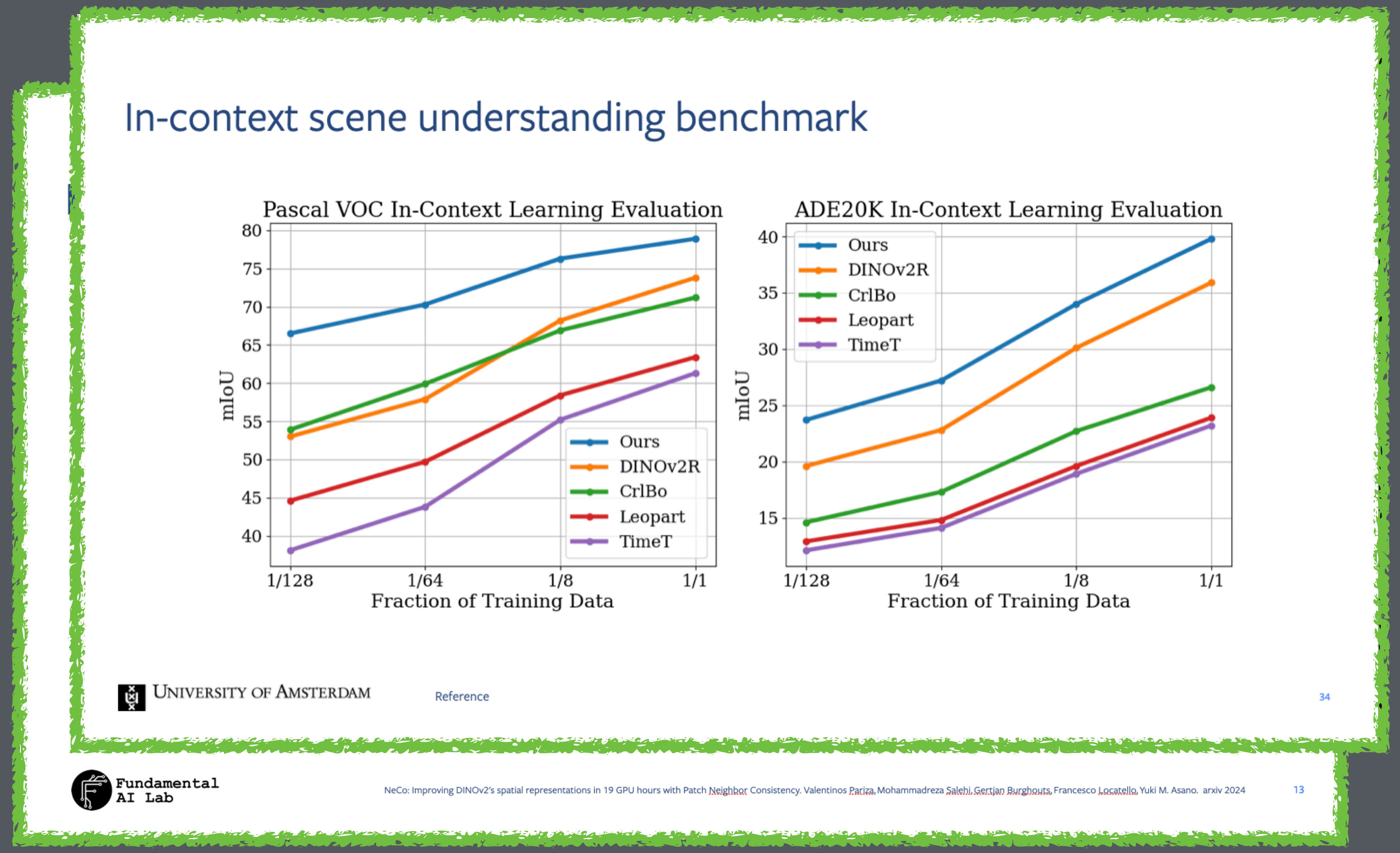




There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

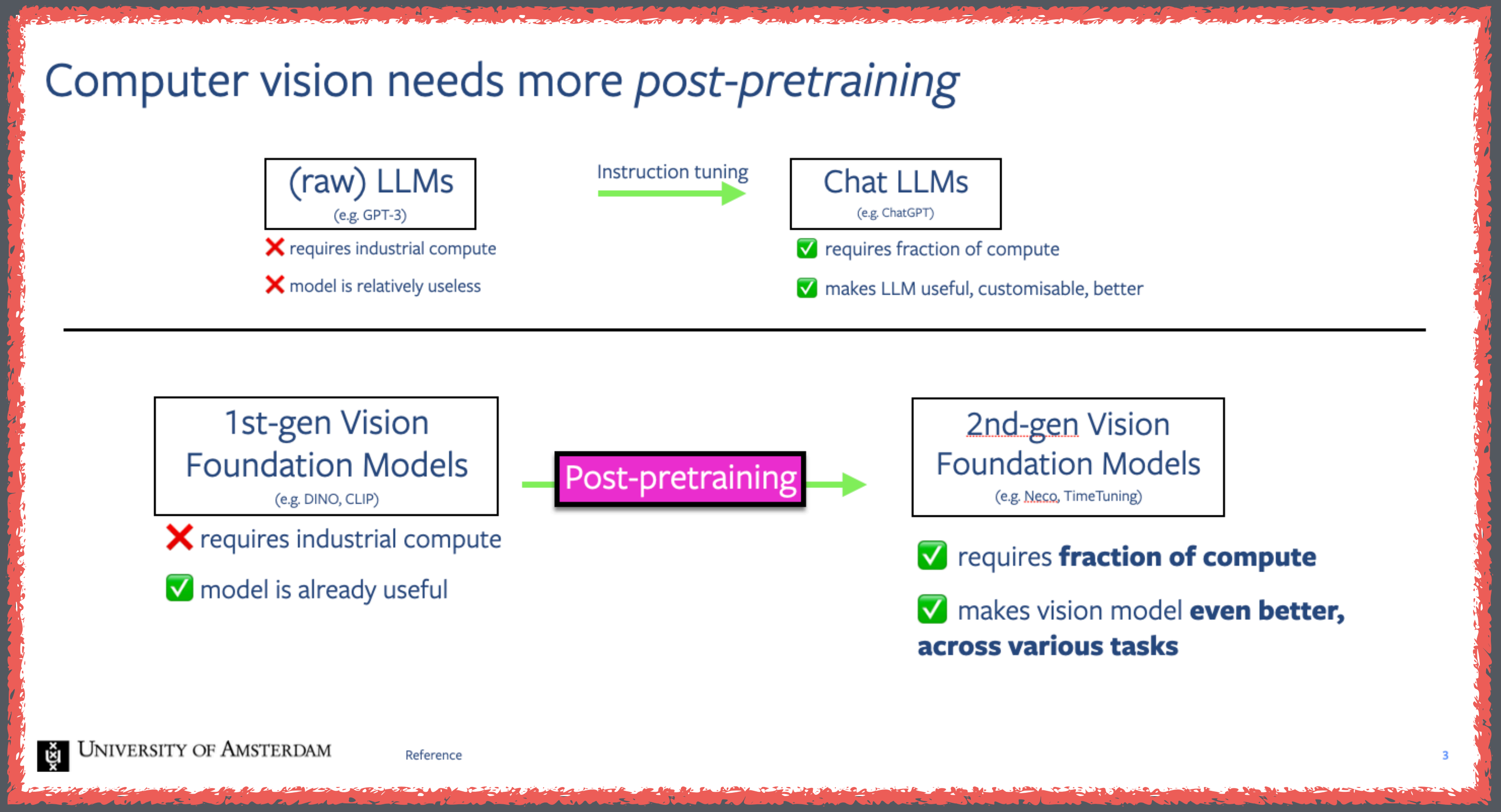
Fundamental AI Lab

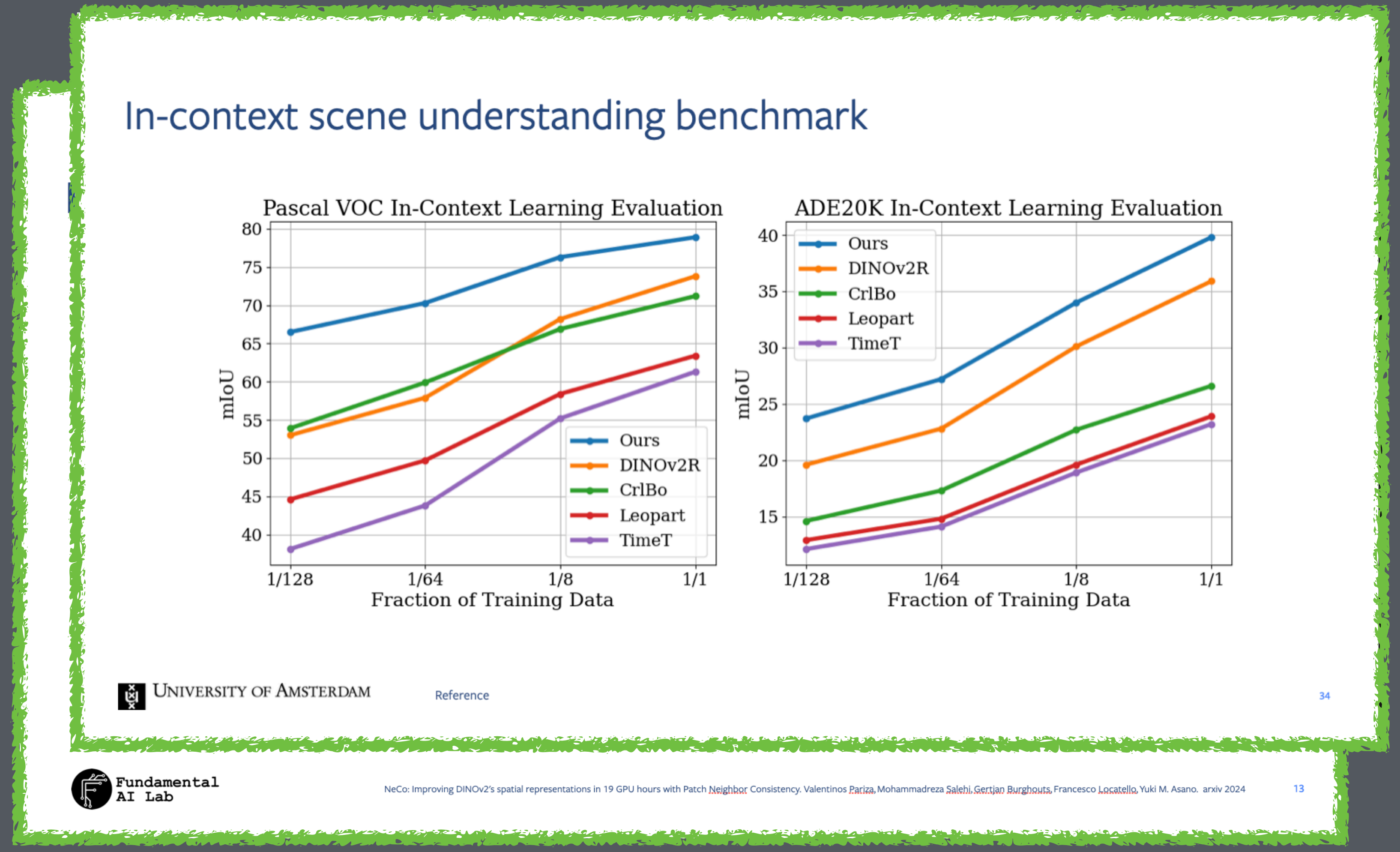


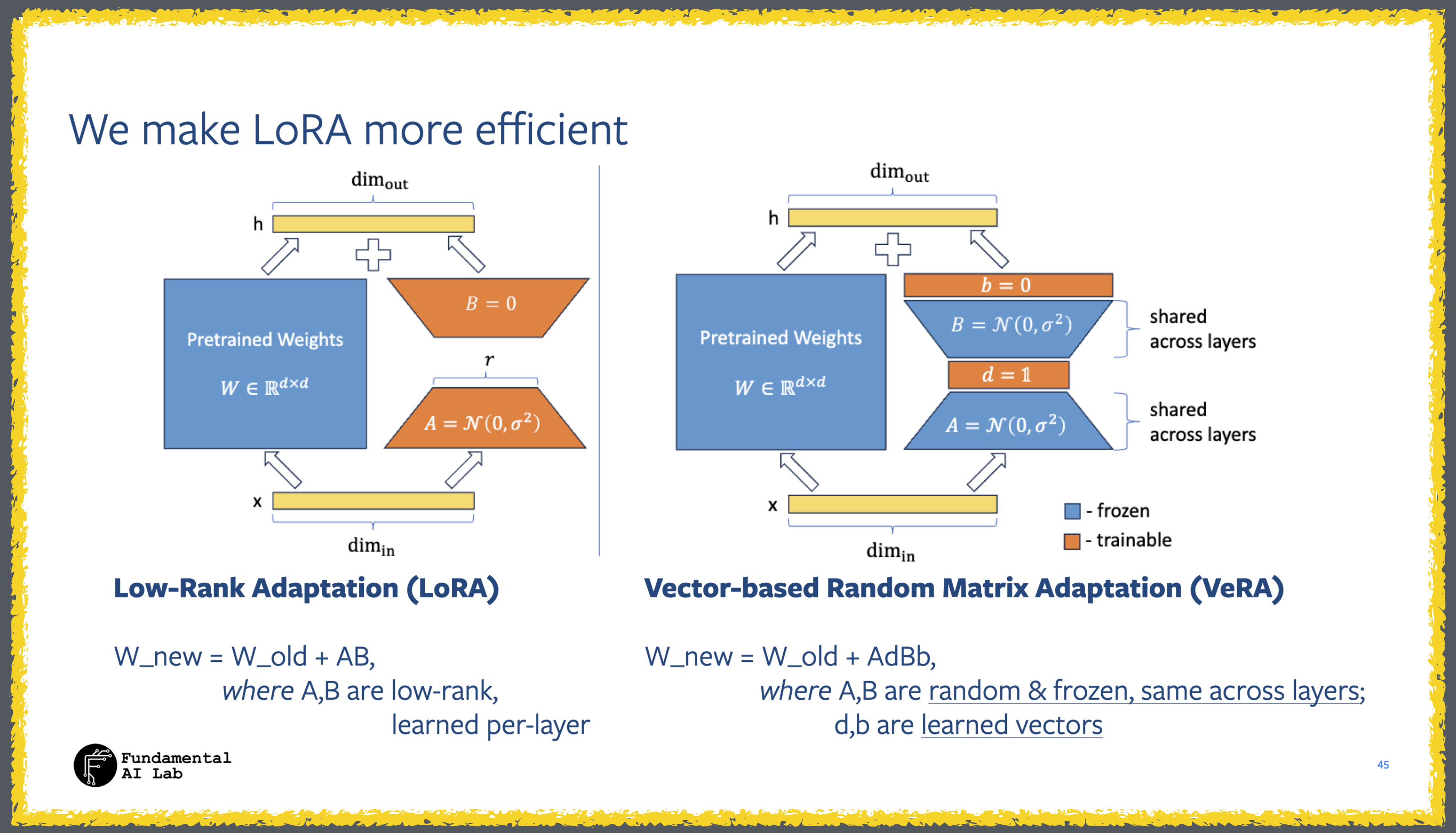


There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

Fundamental AI Lab

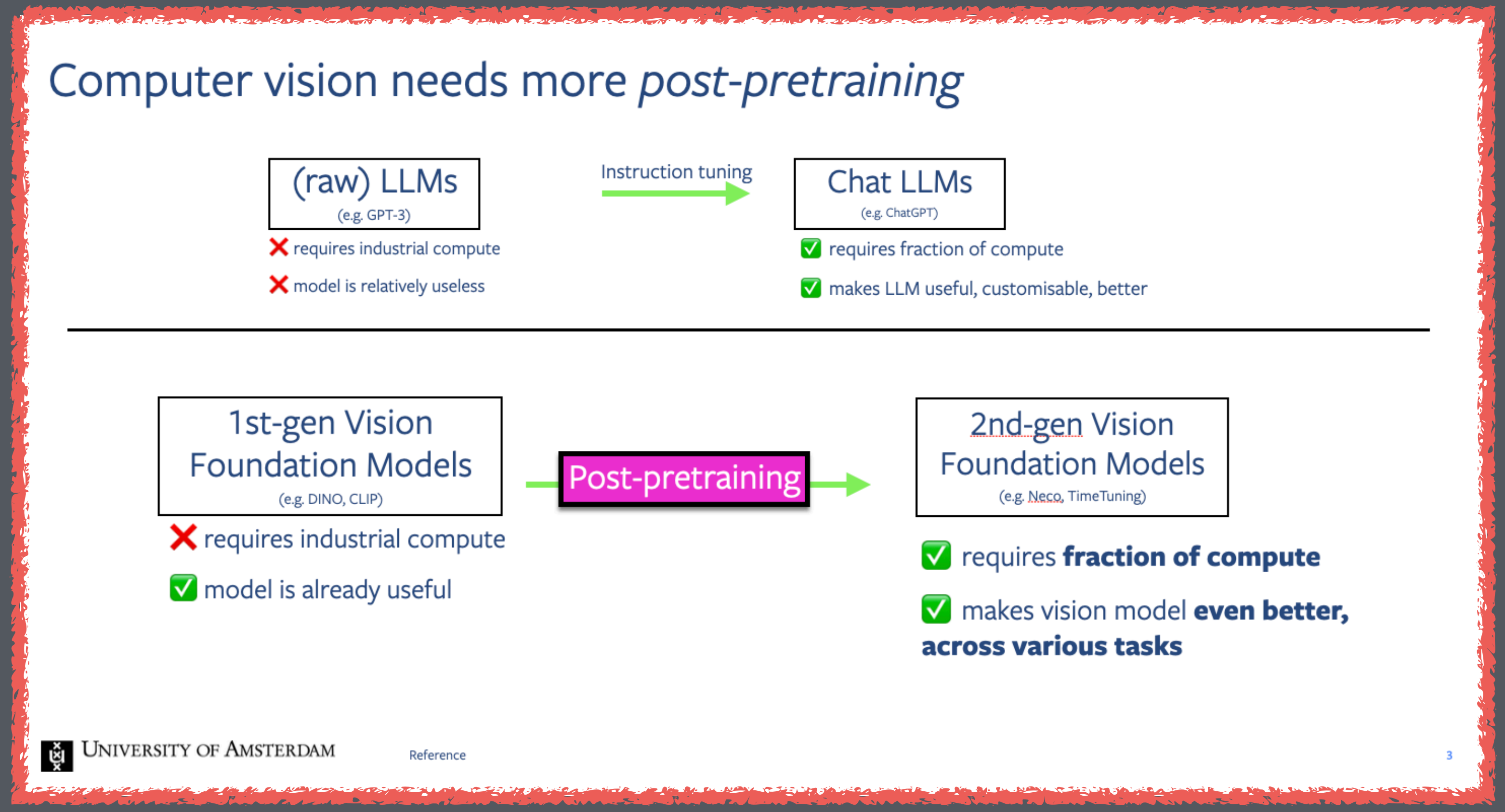


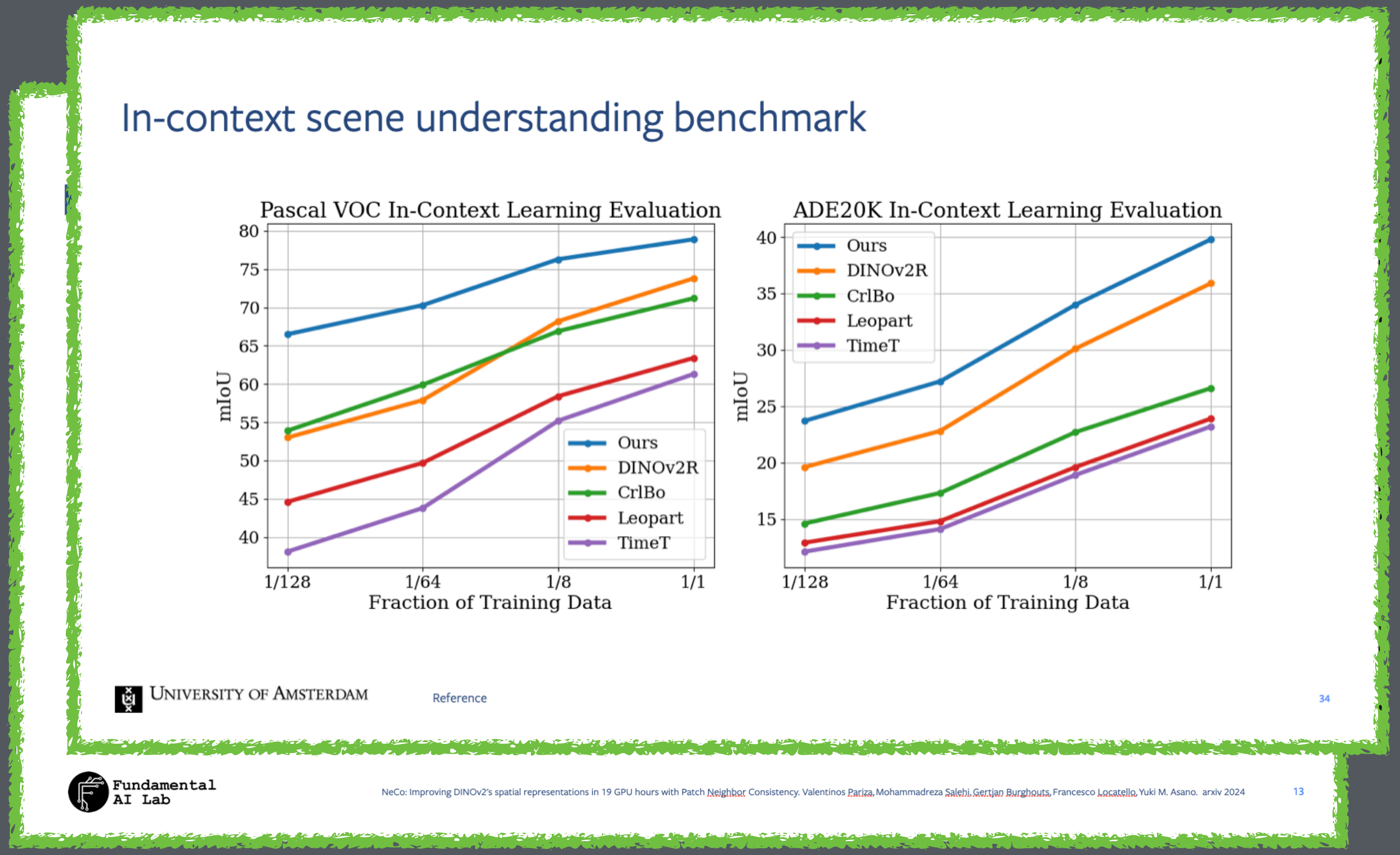


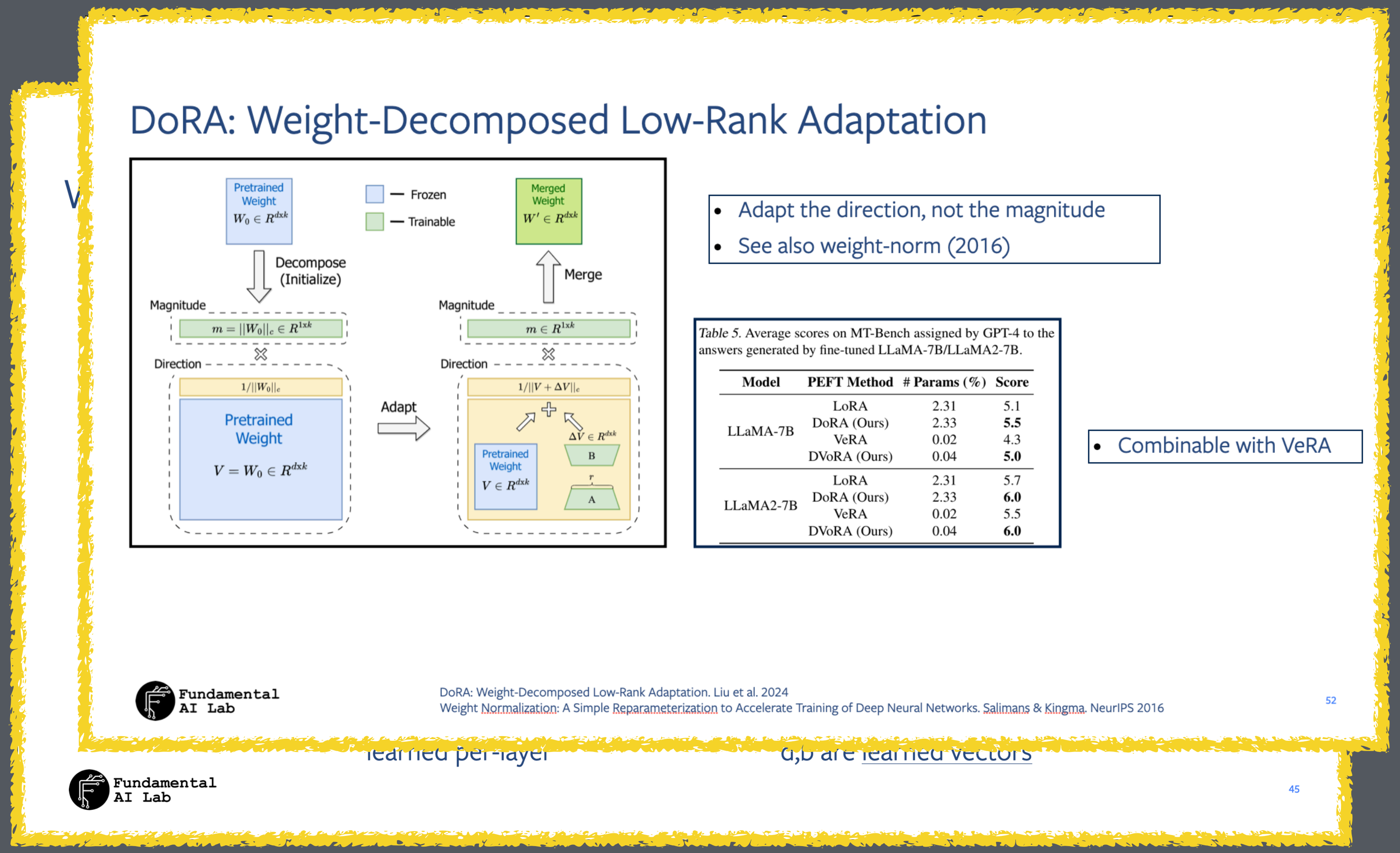


There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

Fundamental AI Lab





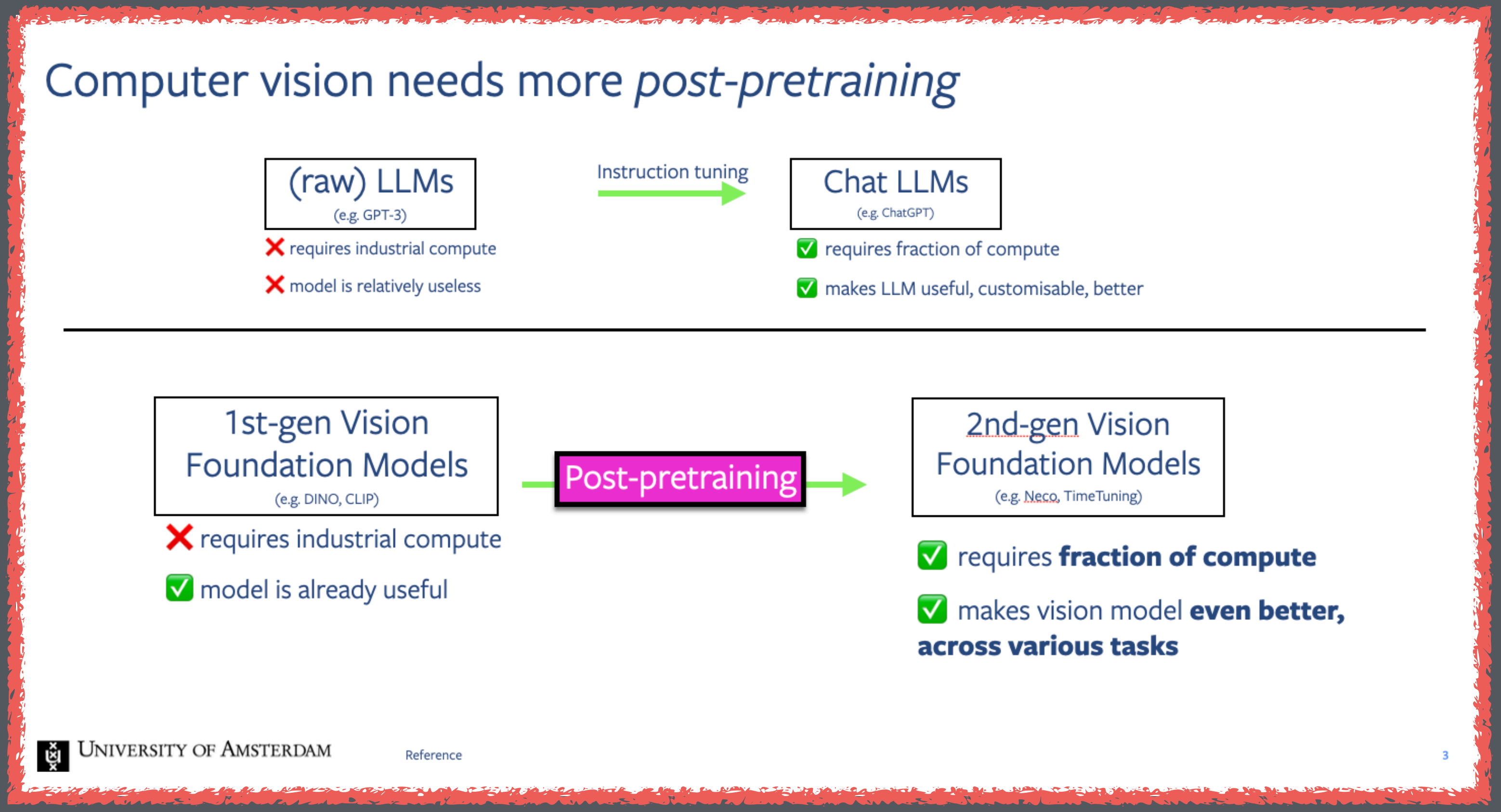


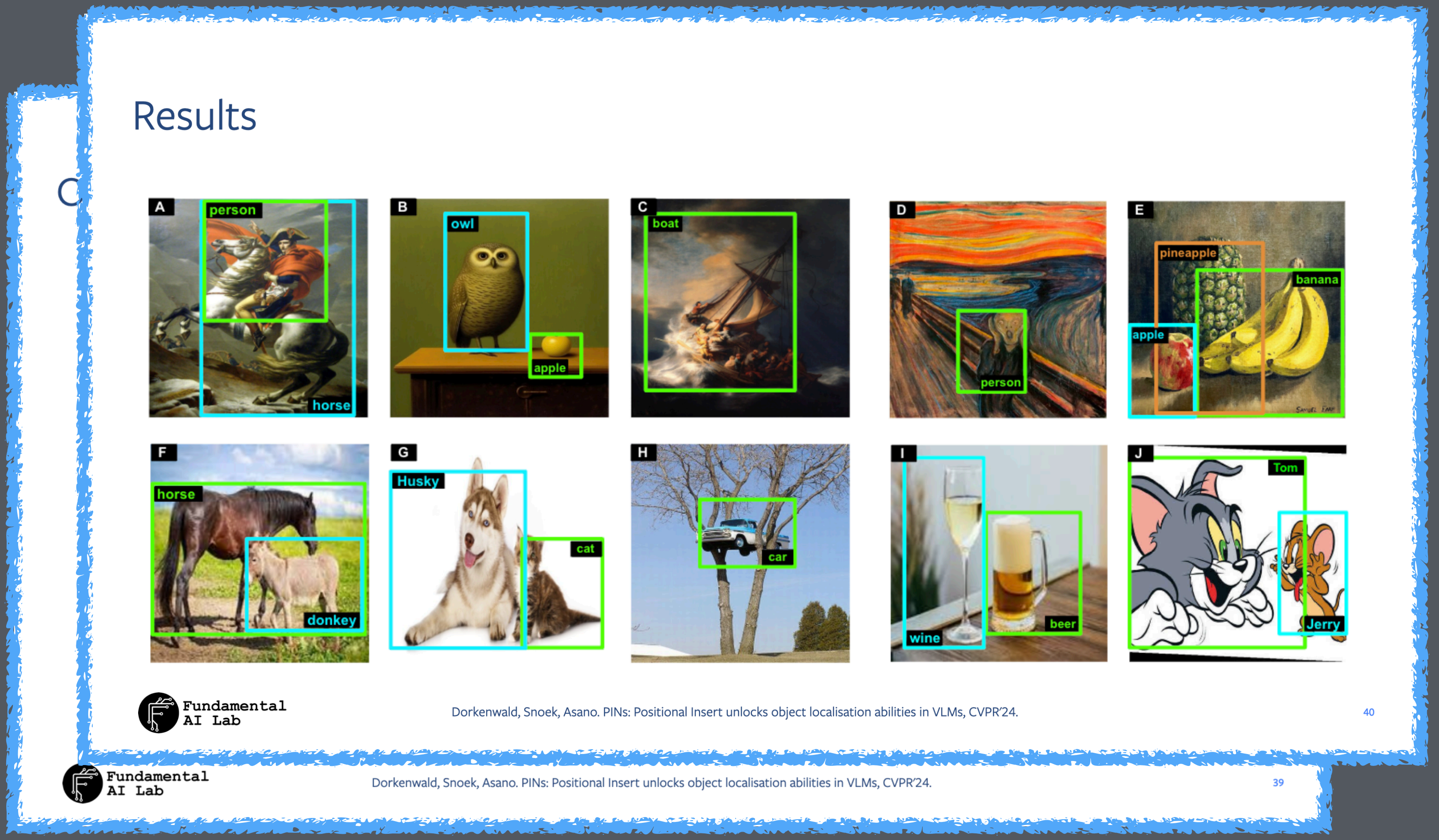
There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

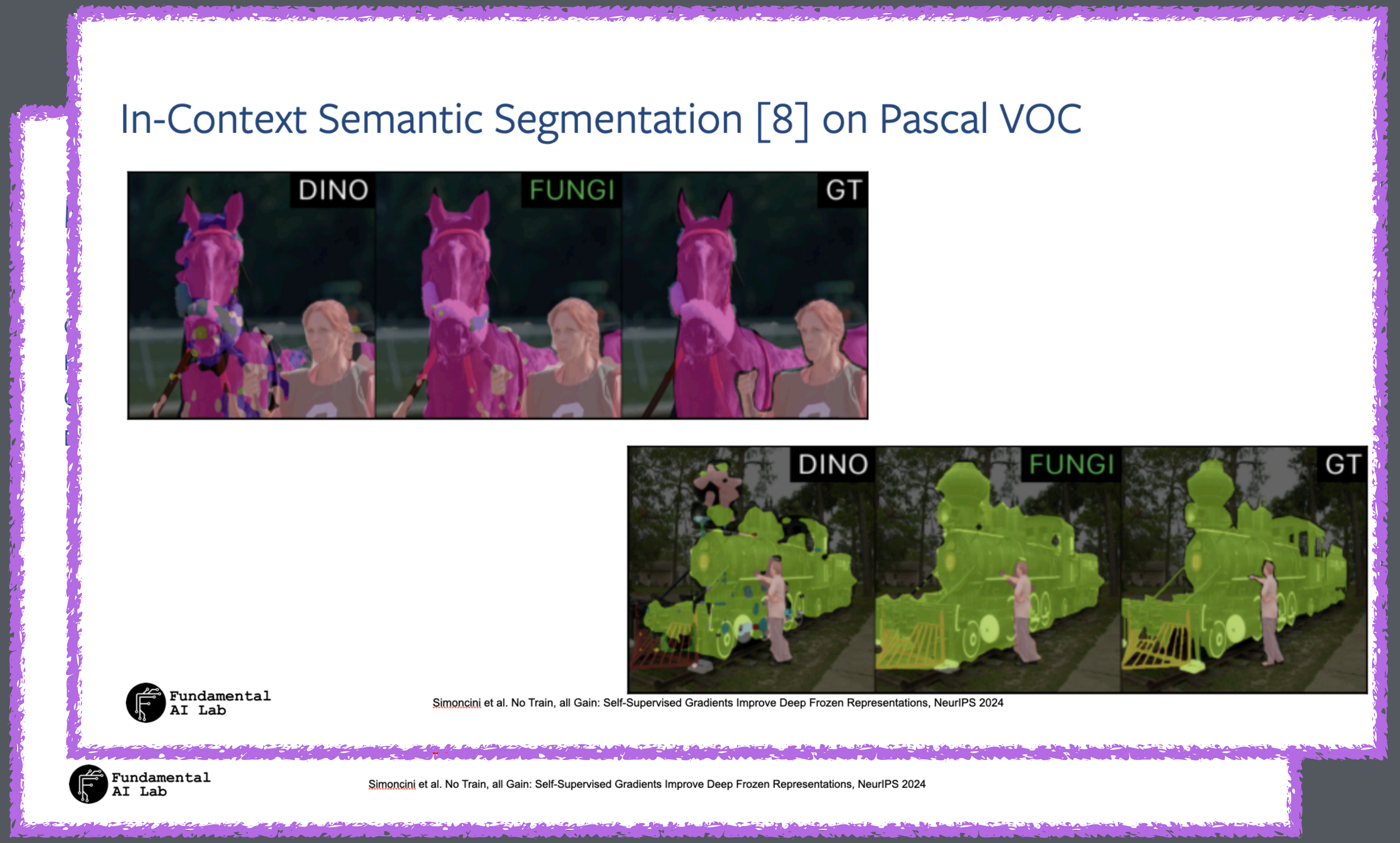
Fundamental AI Lab

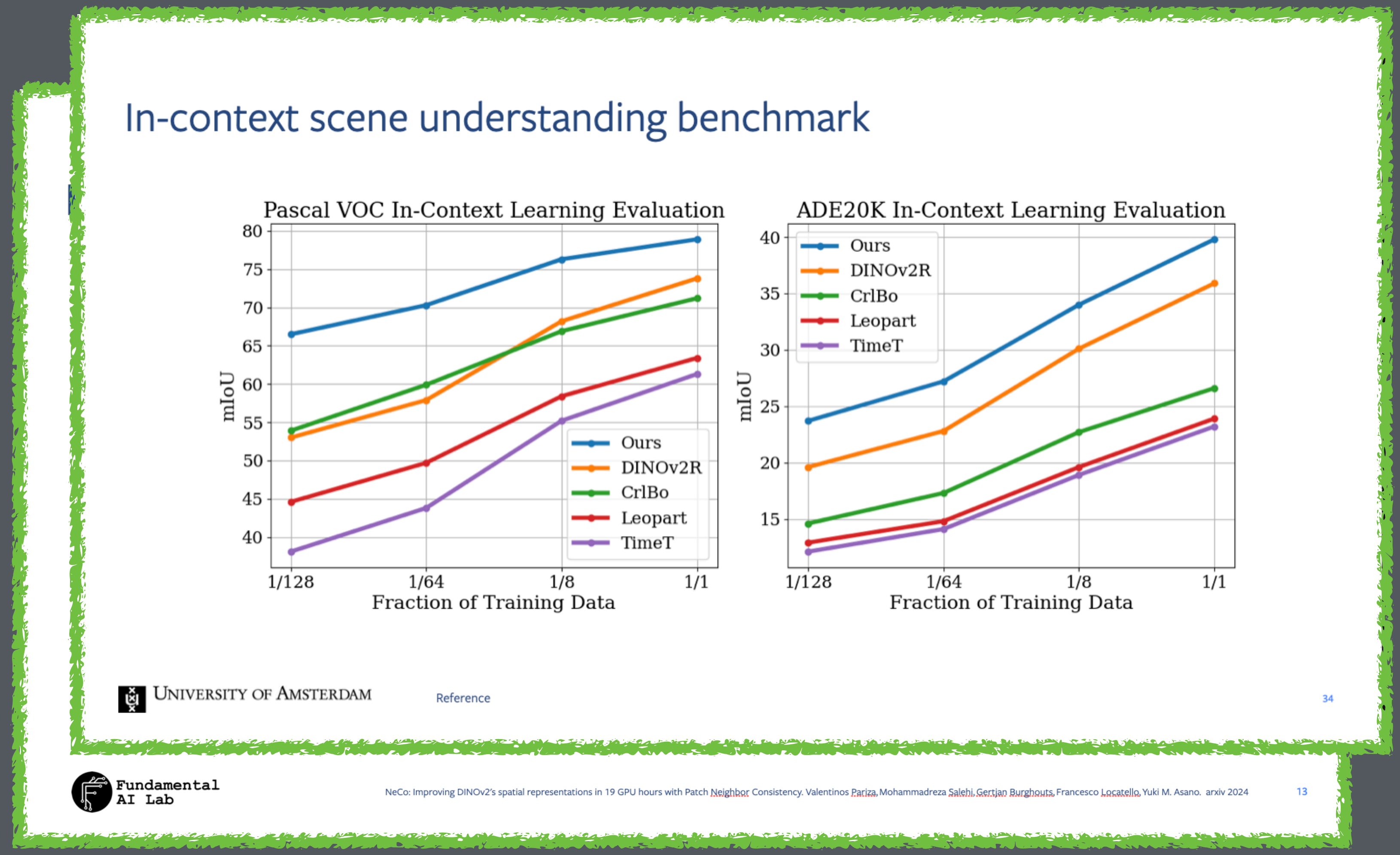
There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

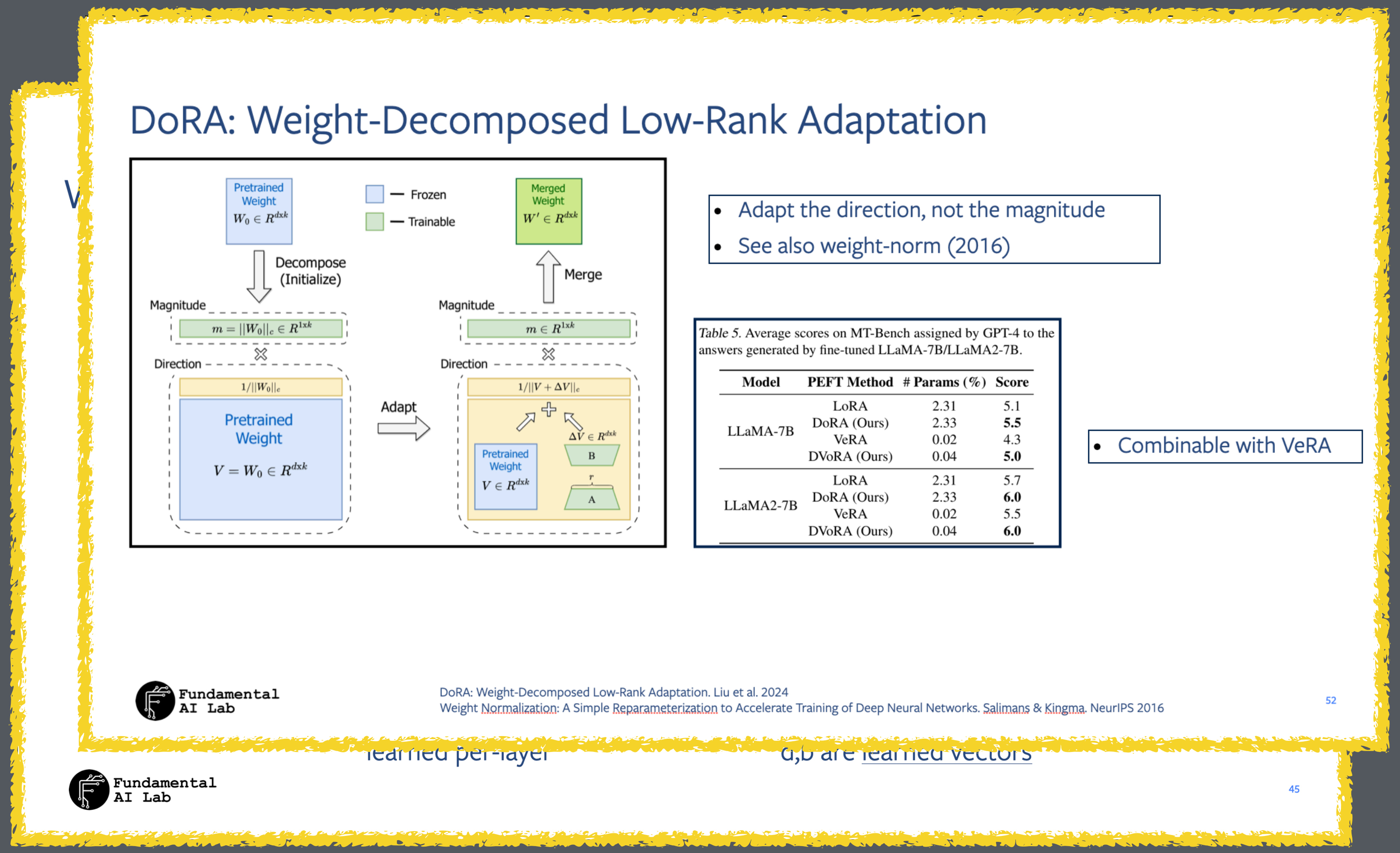
Fundamental
AI Lab











There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

Fundamental AI Lab

There is lots of exciting research to achieve better models (efficient, robust, faster) with post-pretraining.

Fundamental AI Lab