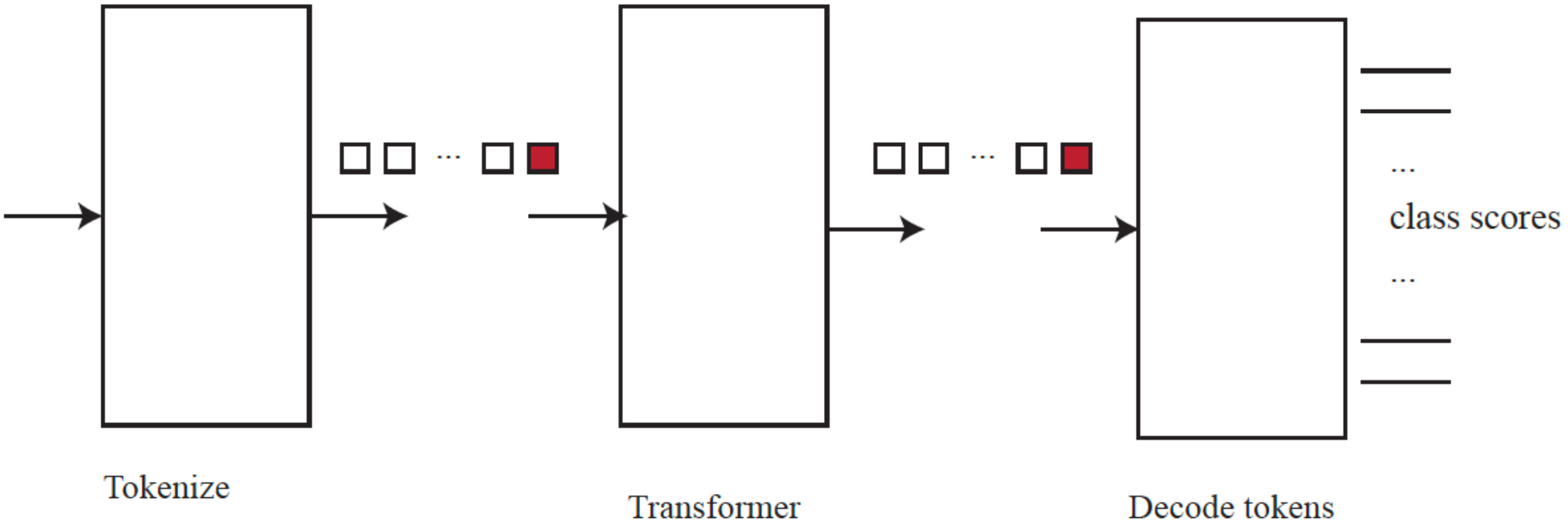
Using transformers: Birds Eye View

D.A. Forsyth,

University of Illinois at Urbana Champaign

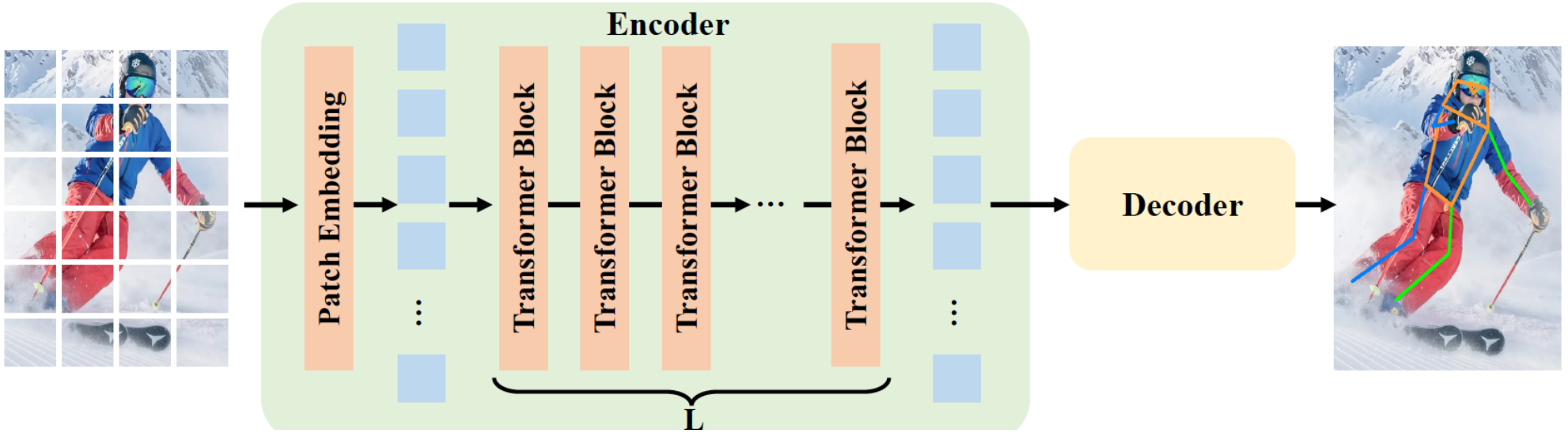
Classification by transformer



- Key questions
  - details of transformer
  - details of decode
    - use class token
    - use all tokens (how?)

Pose prediction by transformer

- Apply to windows containing person
  - found by, say, detector

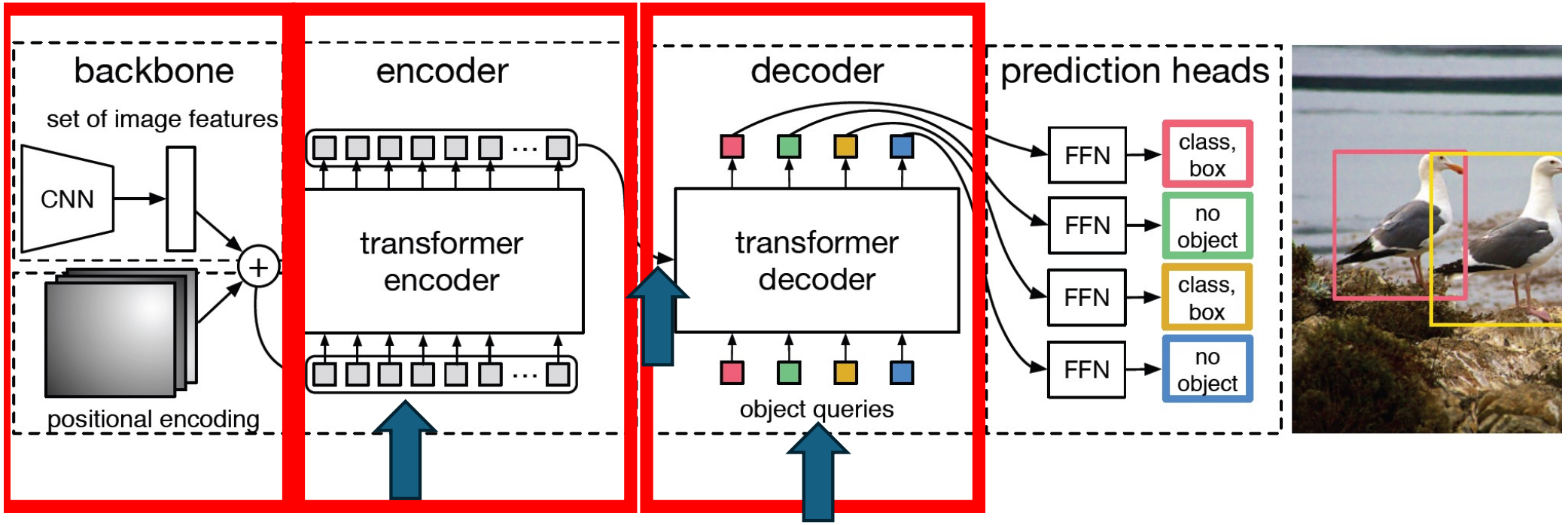


ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation, Xu et al, 2024

Pose prediction by transformer

ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation, Xu et al, 2024

DETR – detection by transformer



Tokenize

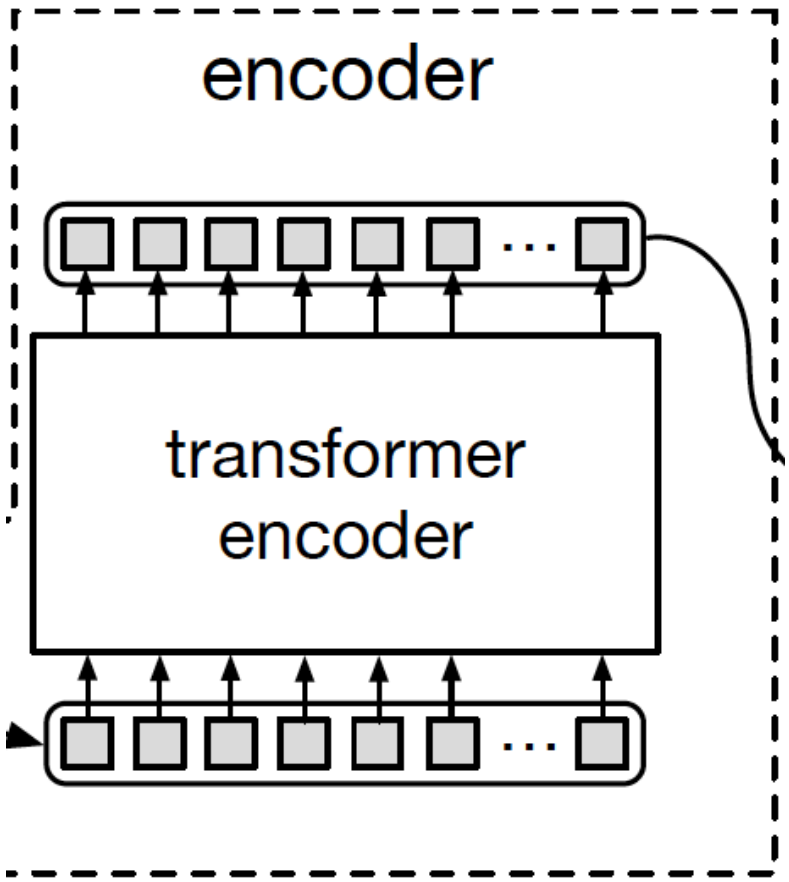
Transformer accept one stream, make one stream

Transformer accept two streams, make one stream

End-to-End Object Detection with Transformers Carion et al, 2020

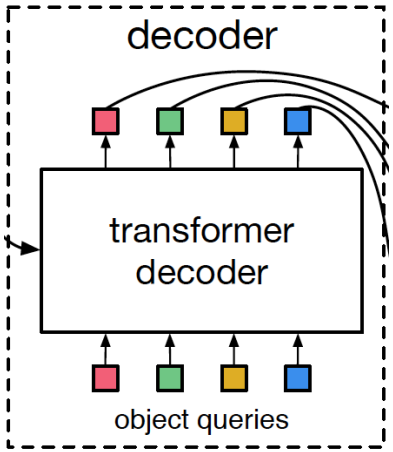
DETR – detection by transformer

- Encoder
  - Accept image tokens, make new
  - using multi-headed self-attention



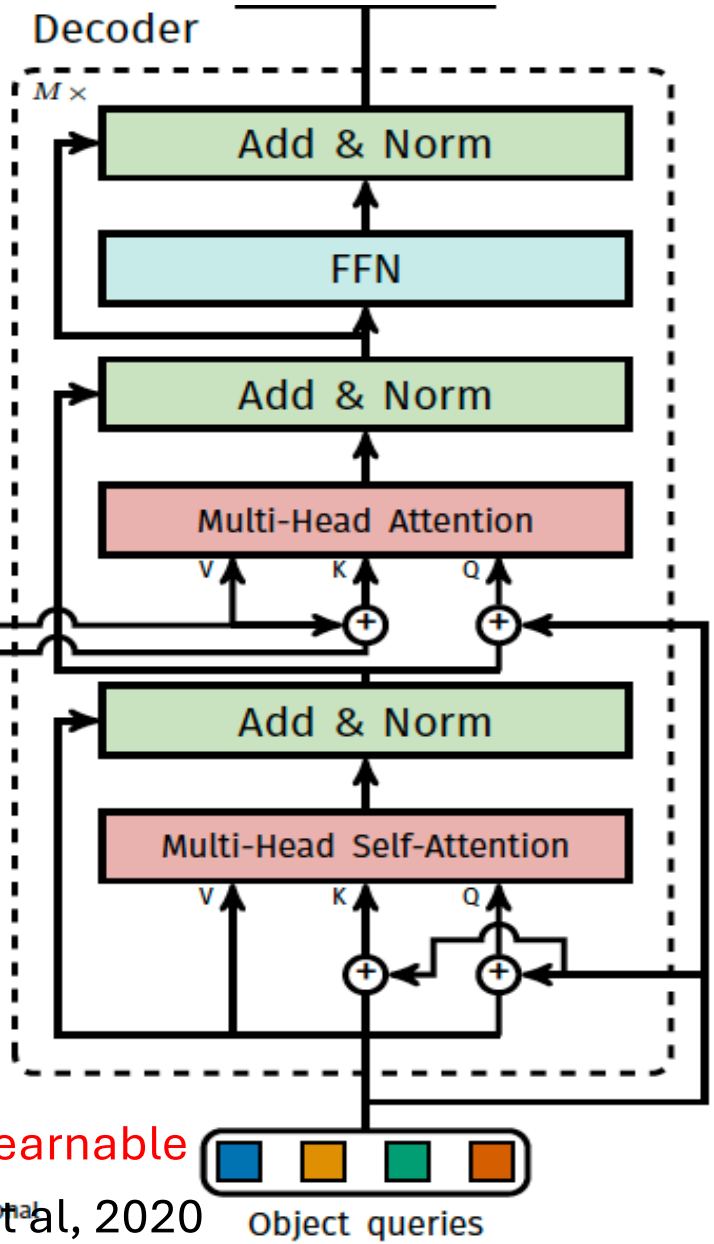
End-to-End Object Detection with Transformers Carion et al, 2020

DETR – detection by transformer



Tokens from decoder

Positional encoding

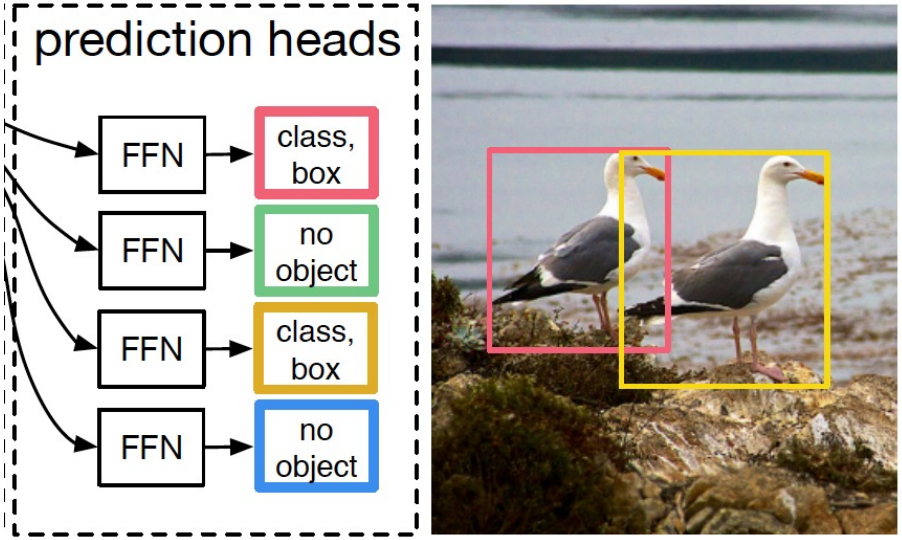


These are learnable

End-to-End Object Detection with Transformers Carion et al, 2020

DETR – detection by transformer

- prediction head
  - accepts output token
  - produces
    - no object OR
    - class scores, box location



- Training (roughly!)
  - Find large number of images with boxes, classes
  - Loss is min over matchings
    - from predicted tokens to image boxes
      - box loss+class loss
  - Use weighted bipartite graph matching to match
  - Descent on this

End-to-End Object Detection with Transformers Carion et al, 2020

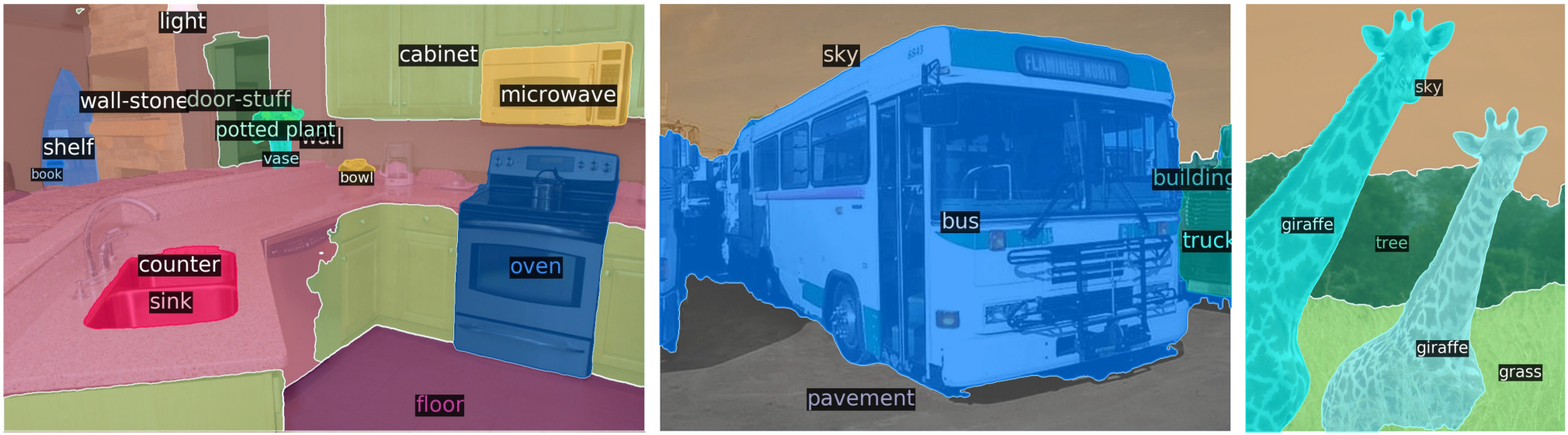
DETR works

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

End-to-End Object Detection with Transformers Carion et al, 2020

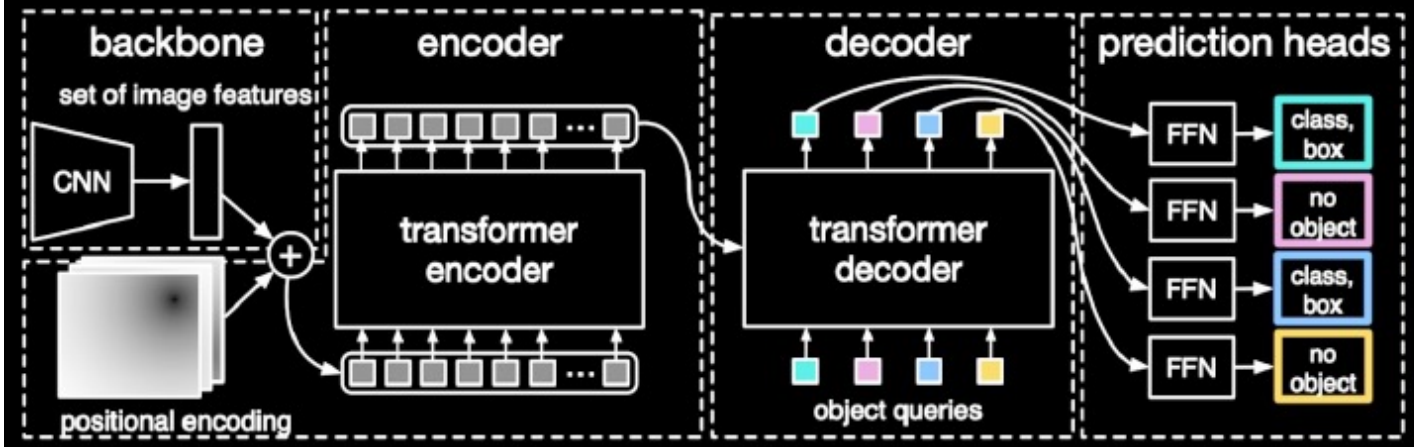
Panoptic segmentation with DETR

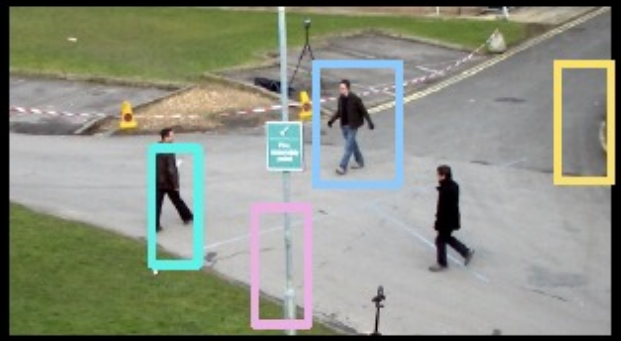
- Adjust decode to get mask



End-to-End Object Detection with Transformers Carion et al, 2020

DETECTION WITH TRANSFORMERS



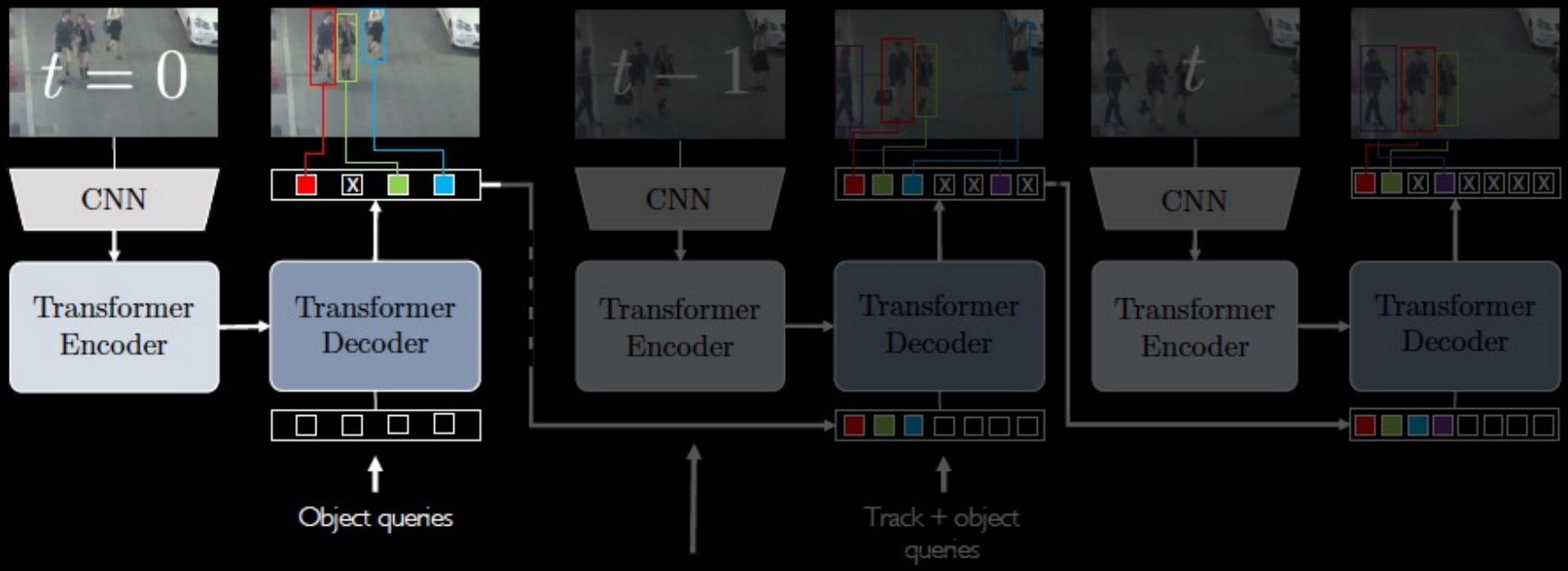


- Object detection a set prediction problem [1, 2]
- Transformer decoder
  - Object query self-attention ({class, box} or no-object)
  - Encoded image feature and object query cross attention

[1] Carion et al. *End-to-End Object Detection with Transformers*. ECCV, 2020.
[2] Zhu et al. *Deformable DETR: Deformable transformers for end-to-end object detection*. ICLR 2021.

TRACKFORMER

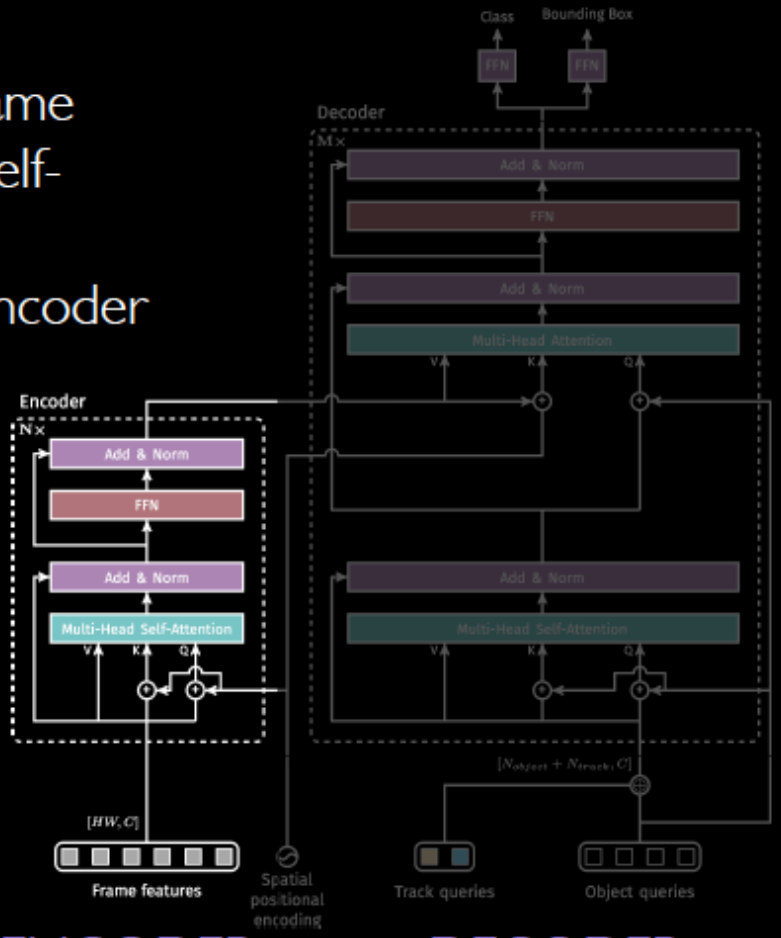
- MOT as a frame-to-frame set prediction problem



Meinhardt et al. *Trackformer: multi-object tracking with transformers.* CVPR 2022

ENCODER-DECODER TRANSFORMERS

Encoding of frame features with self-attention in a Transformer encoder

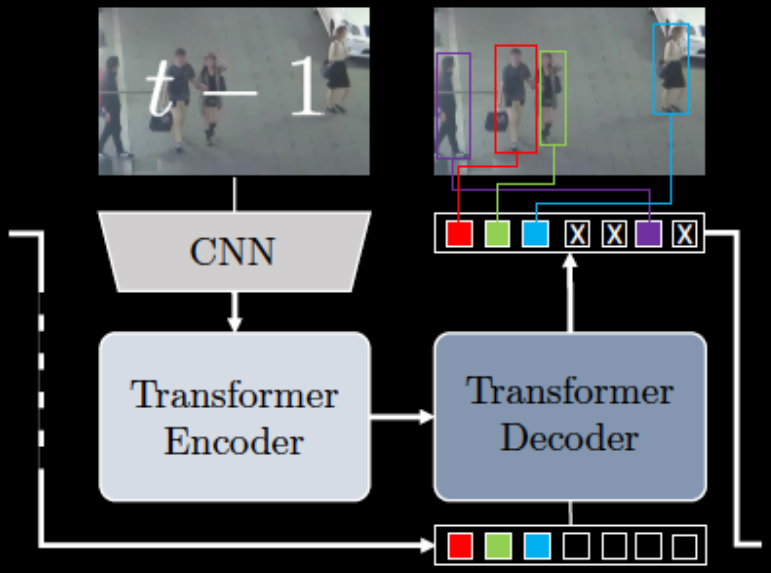


Mapping of queries to box and class predictions using MLPs

Self- and encoder-decoder attention

Concatenation of object and track queries.

Transformer Query Decoding

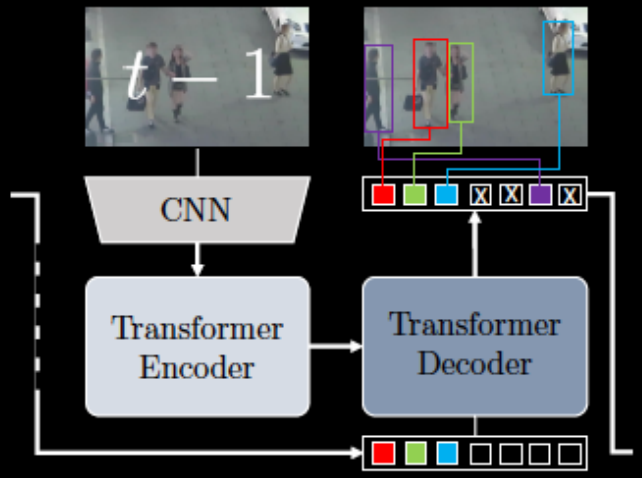


1. Self-attention between queries
   a. Initialize new track (object query)
   b. Terminate occluded track

2. Encoder-decoder attention
   a. Find new object in frame
   b. Adjust to changed position of tracks

Can I Recover From Occlusions?

- Just keep track queries active for a time window.
- No need to an extra re-ID head.
- The spatial information embedded into each track query prevents their application for long-term occlusions.

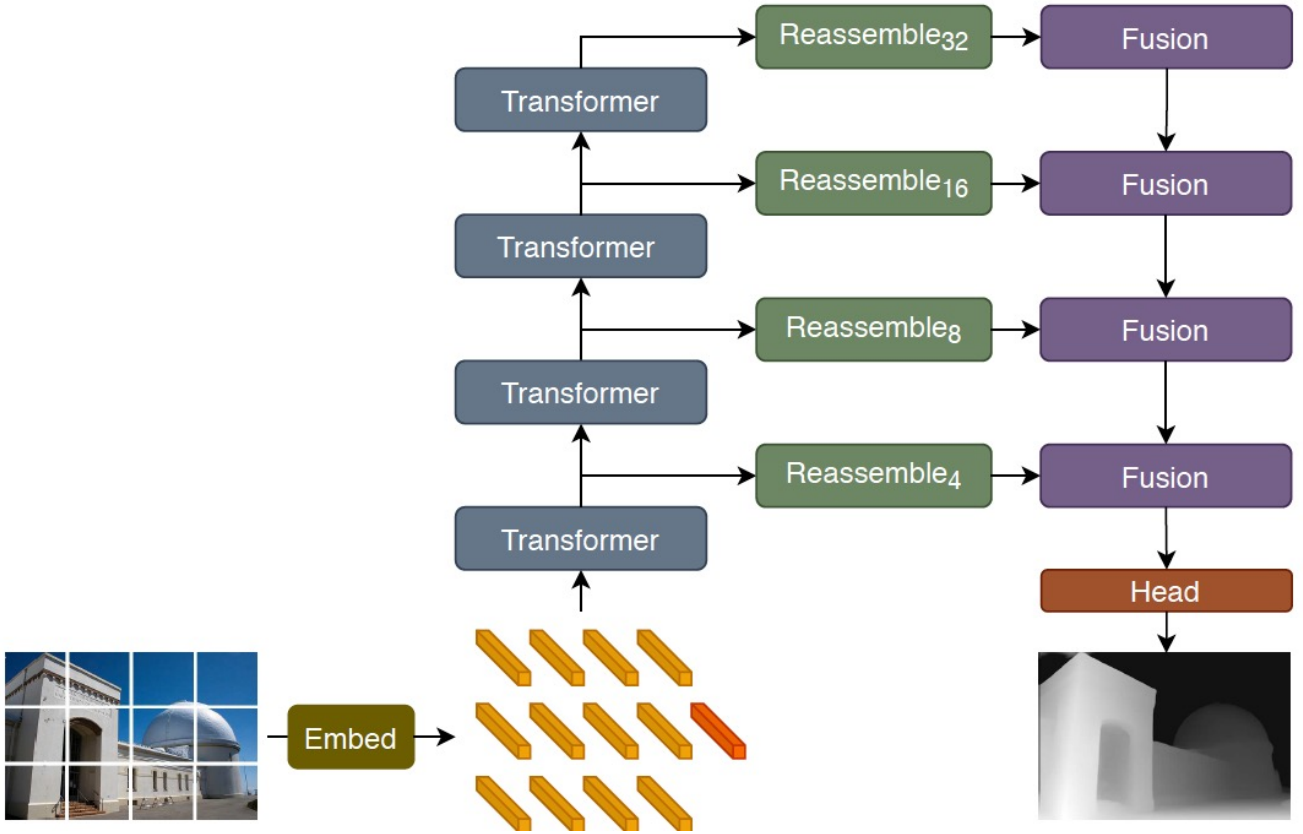


TrackFormer

- Elegant formulation of tracking which naturally merges detection and data association
- Good performance with partial occlusions
- Good performance where detectors are weak
- State-of-the-art results (with some data and some tricks)
- Similar concurrent papers: MeMOT (ECCV22) and MOTR (CVPR22)

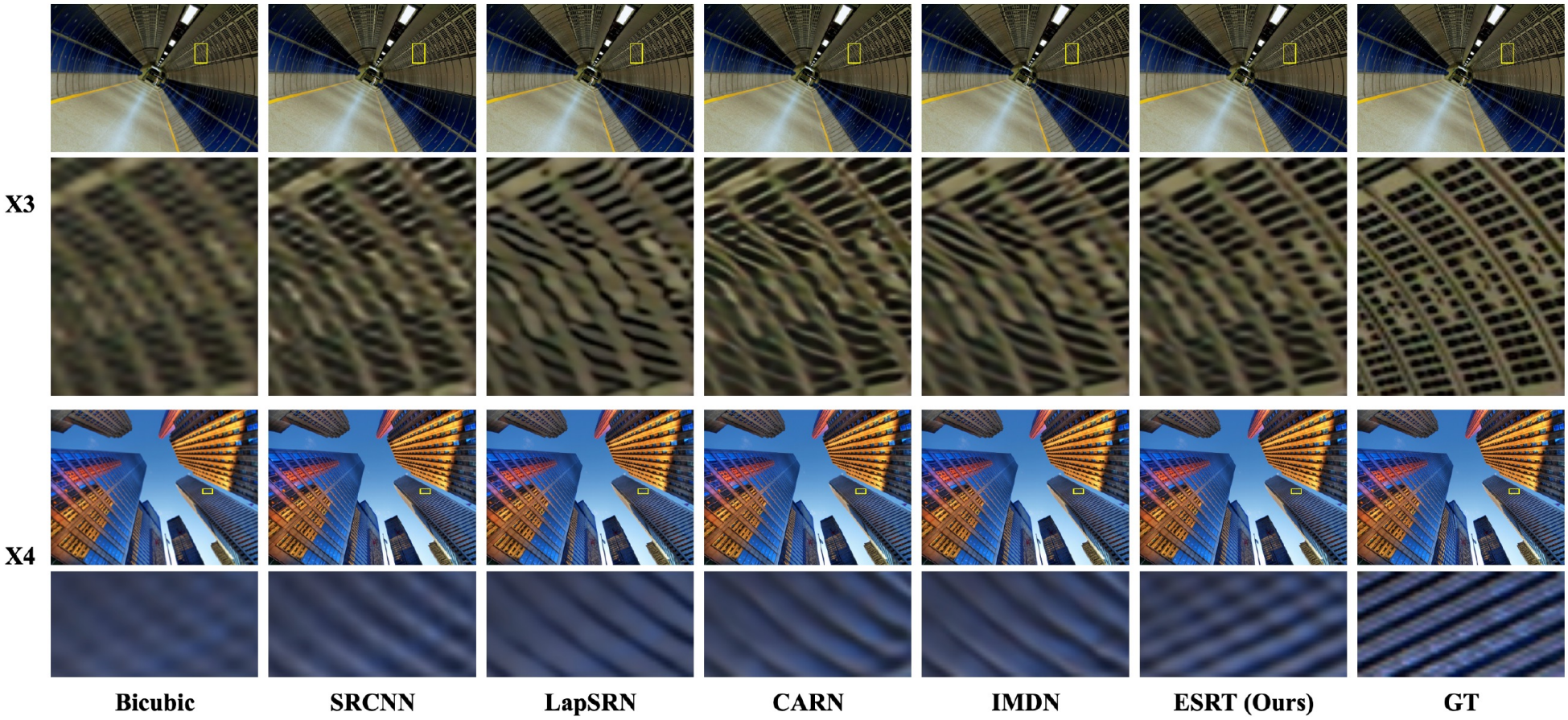
Depth prediction by transformer

- Issue:
  - decode token streams into image-like thing
  - reassemble
    - tokens were each cut from particular location in image



Vision Transformers for Dense Prediction, Ranftl et al, 2021

Superresolution by transformer



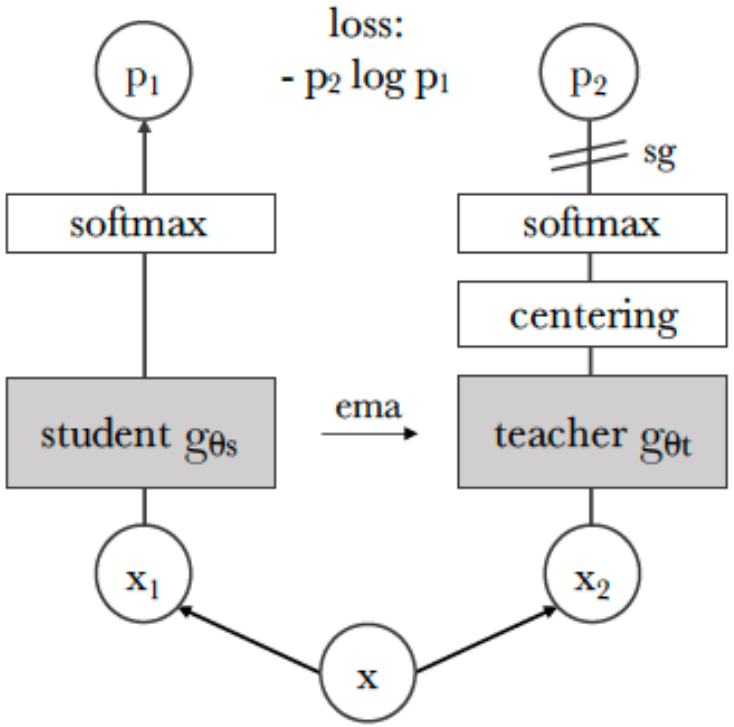
Transformer for Single Image Super-Resolution Lu et al 2022

The general encoder

- key motivating idea
  - Q:can you train an encoder that is good at everything?
    - then all you need to worry about is decoding
  - A:hard to be sure; but you can train encoders that are good at a lot of stuff…
    - all evidence is these need to be transformers
    - training
      - WITHOUT using labels
      - on a huge dataset
- example: DINO V3

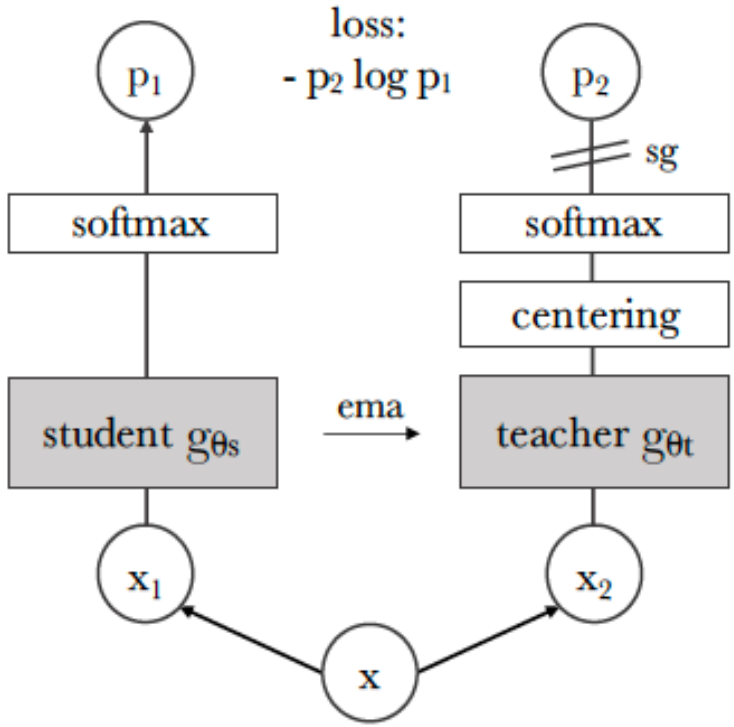
DINO: self-distillation

- x1, x2 are two versions of x
  - eg crops
- Feature is softmax-normalized
  - scaled by temperature
- Teacher network is
  - moving average of student
- Centering and sharpening
  - adjust teacher features to avoid collapse
  - centering:
    - add a bias consisting of moving average across batches
  - sharpening
    - use a different temperature param for teacher



DINO-V3: self-distillation

- additional losses
  - numerous changes
  - losses change during training
- 1.7e9 images, carefully curated
  - and clustered, to manage what images are used when



Spatial features

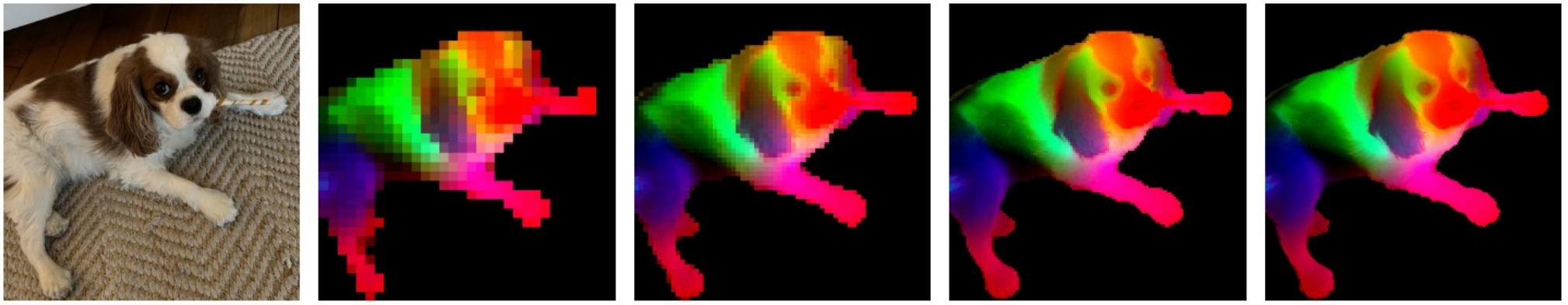


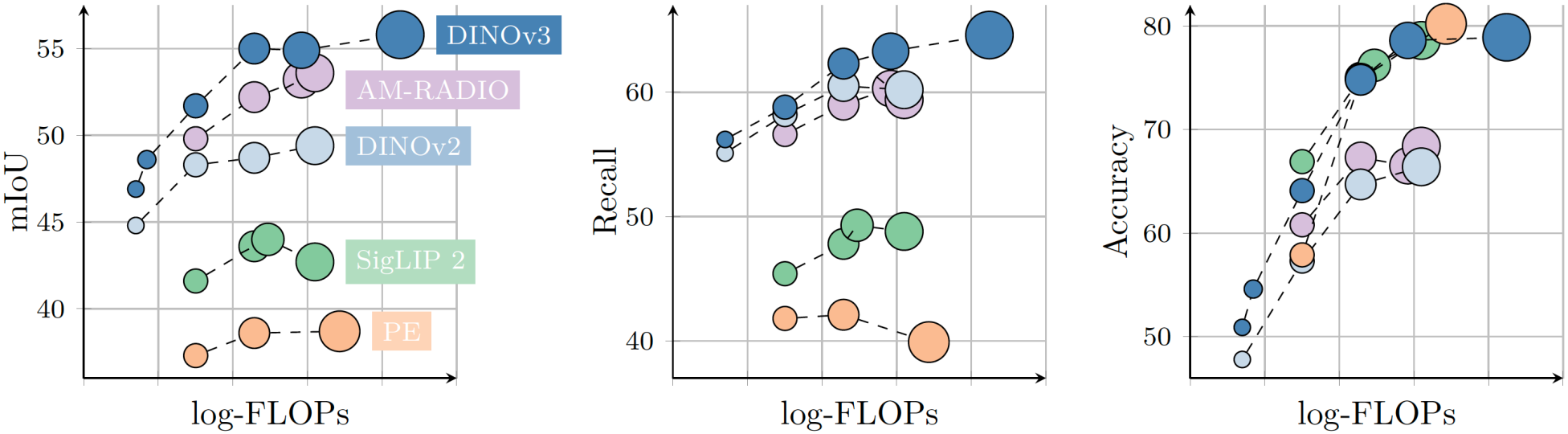
Figure 4: DINOv3 at very high resolution. We visualize dense features of DINOv3 by mapping the first three components of a PCA computed over the feature space to RGB. To focus the PCA on the subject, we mask the feature maps via background subtraction. With increasing resolution, DINOv3 produces crisp features that stay semantically meaningful. We visualize more PCAs in Sec. 6.1.1.

DINOv3, Simeoni et al 25

Cosine similarity of DINO v3 features against point marked in red

DINOv3, Simeoni et al 25

Dense general image features



(a) Semantic segmentation (ADE20k) **(b)** 3D keypoint matching (NAVI) **(c)** OOD classif. (ObjectNet)

Figure 2: Performance of the DINOv3 family of models, compared to other families of self- or weakly-supervised models, on different benchmarks. DINOv3 significantly surpasses others on dense benchmarks, including models that leverage mask annotation priors such as AM-RADIO (Heinrich et al., 2025).

DINOv3, Simeoni et al 25

Resources

- Pretrained models
- Start at:
  - https://huggingface.co/docs/transformers/index
    - there are millions (?)
- Look at:
  - https://github.com/huggingface/transformers?tab=readme-ov-file
    - and scroll to computer vision to get some examples, including
    - Segmentation:
      - https://huggingface.co/facebook/sam-vit-base
    - Depth:
      - https://huggingface.co/apple/DepthPro-hf