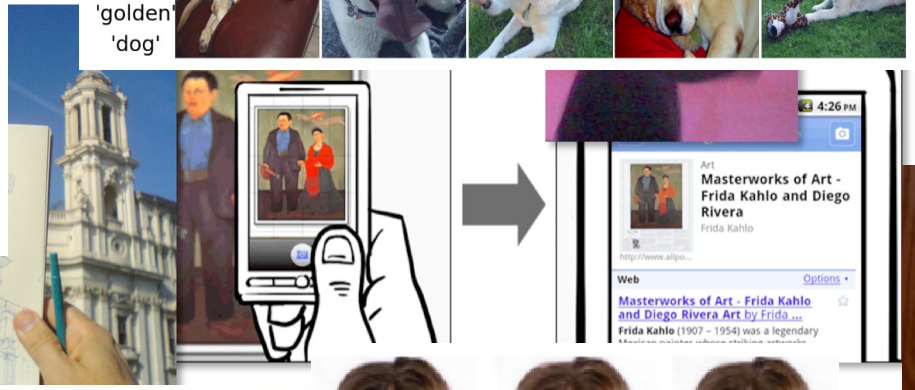
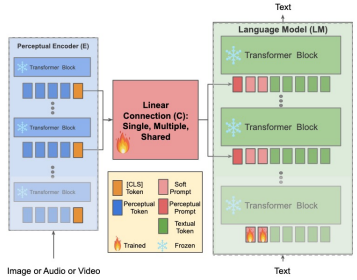
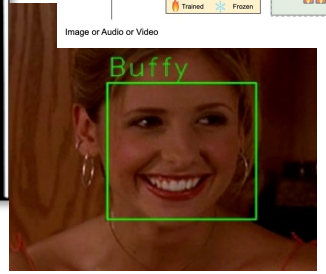
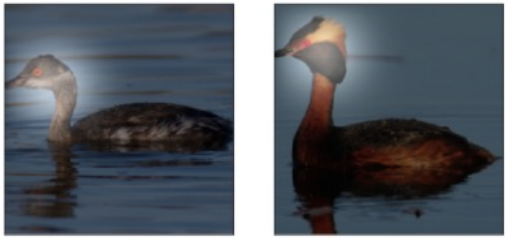
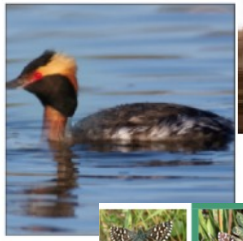
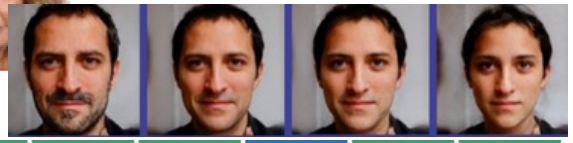












COURS RDFIA deep Image

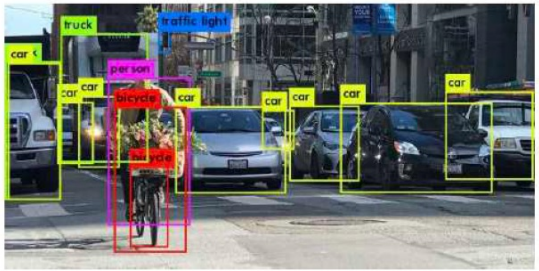
https://cord.isir.upmc.fr/teaching-rdfia/

Self Supervised Learning (SSL)

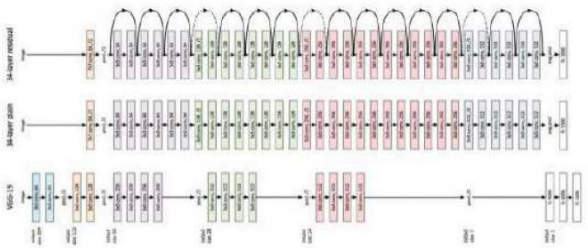
Slide credit to my colleagues A. Bursuc ans S. Gidaris

Supervised Machine Learning for Computer Vision

Deep Learning + Supervised Learning is a really cool and strong combo.



Faster R-CNN, Ren et al. 15



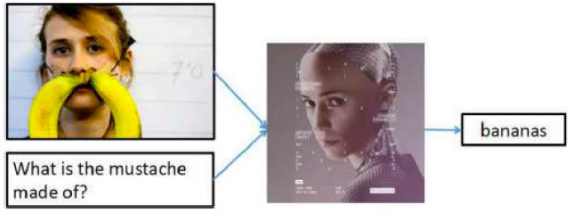
ResNet, He et al. 16

Mask R-CNN, He et al. 17

(a) Mobile phone query (b) Retrieved image of same place

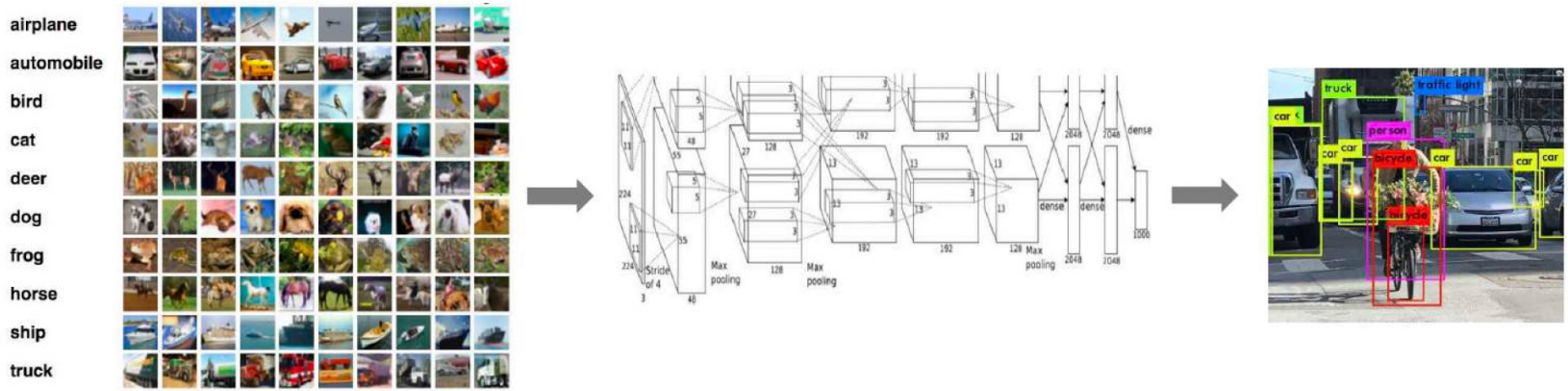
NetVlad: place recognition, Arandjelović et al. 16

Human pose estimation, Newel et al. 17



Visual Question Answering, Antol et al 15

Supervised Machine Learning for Computer Vision

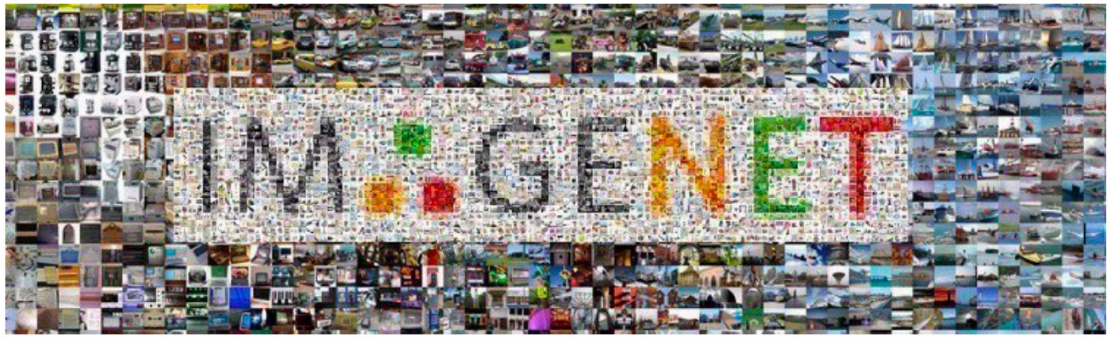


Predefine the set of visual concepts to be learned

Collect diverse and large number of examples for each of them

Train a deep model for several GPU hours or days

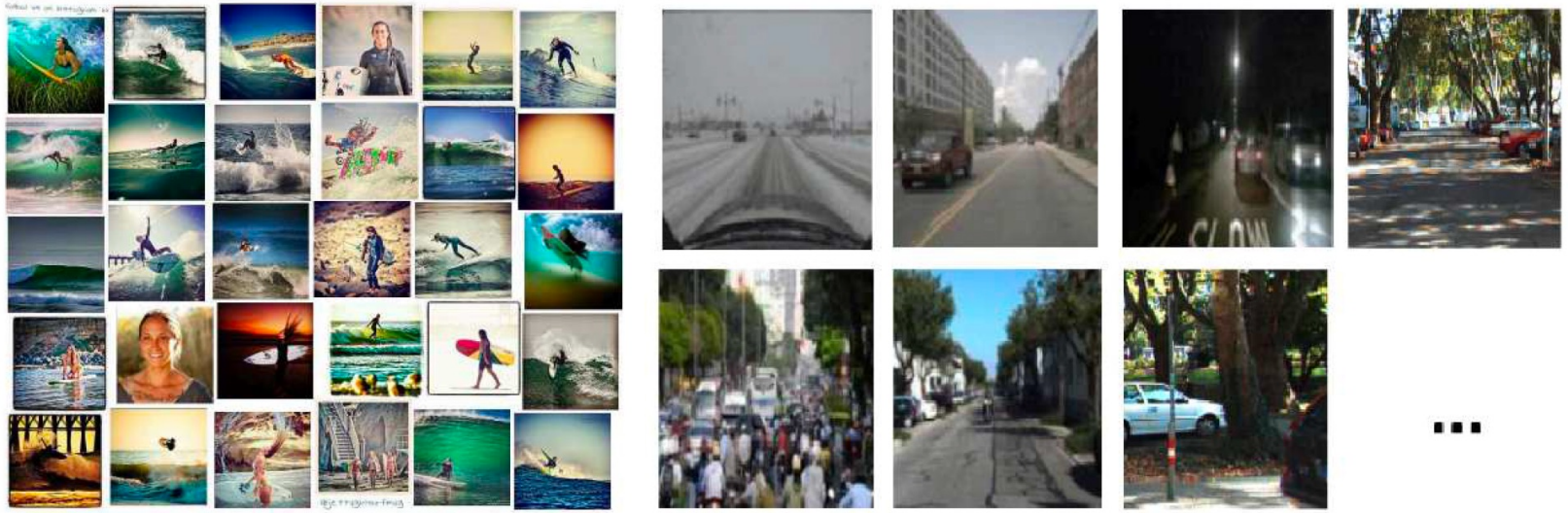
Difficult to acquire and curate large human-annotated datasets



- Requires intense human labor
  - annotating + cleaning raw data
- Time consuming and expensive
- Error prone (human mistakes)

Annotating such image: ~1.5h

Exploiting raw unlabeled data



- Acquiring raw unlabeled data is usually easy
- However, typical supervised methods cannot exploit them

Inspiring success from self-supervision in NLP, e.g., word2vec

Input: The man went to the $[MASK]_1$. He bought a $[MASK]_2$ of milk .
Labels: $[MASK]_1$ = store; $[MASK]_2$ = gallon

Missing word prediction task.

Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence

Next sentence prediction task.

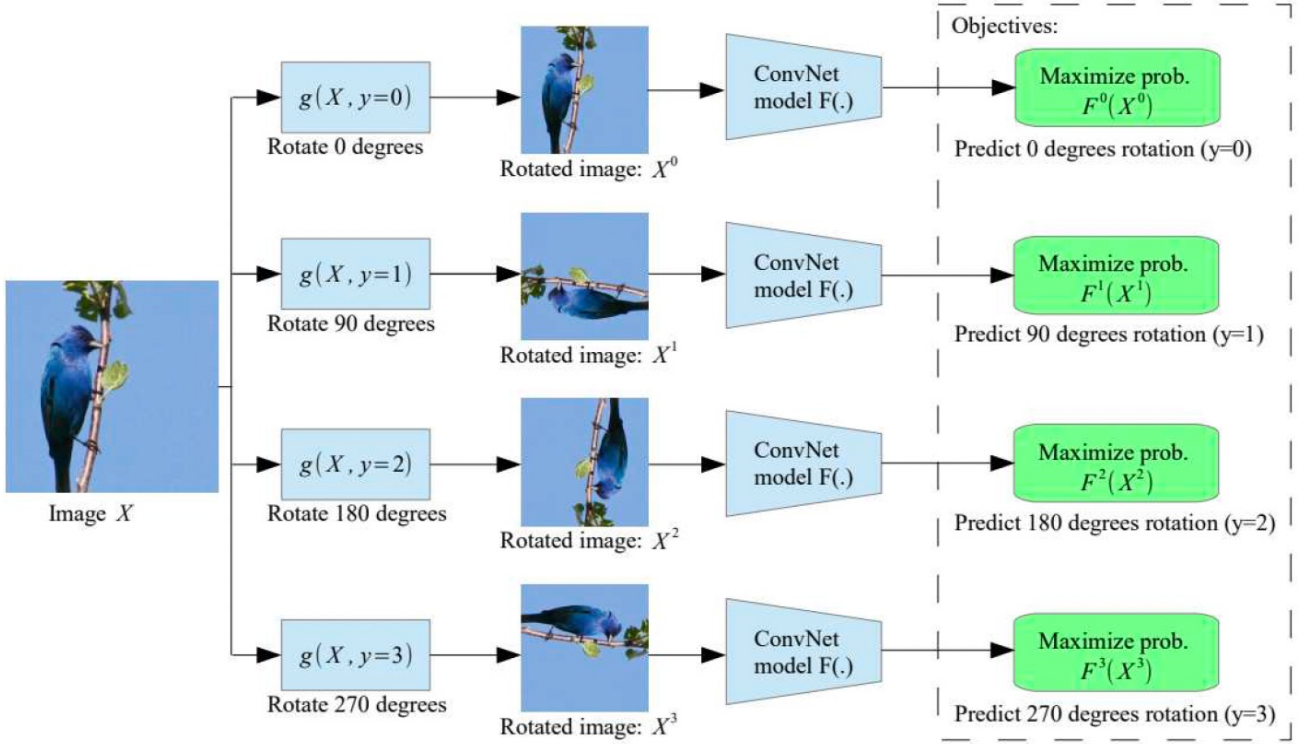
BERT scheme (same time as GPT)

What is self-supervision?

- A form of unsupervised learning where the data (not the human) provides the supervision signal
- Usually, define a pretext task for which the network is forced to learn what we really care about
- For most pretext tasks, a part of the data is withheld and the network has to predict it
- The features/representations learned on the pretext task are subsequently used for a different downstream task, usually where some annotations are available.

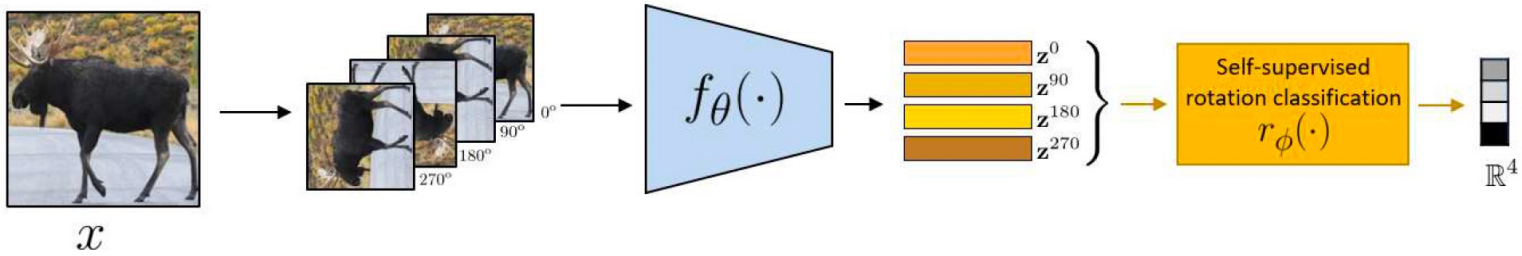
Example: Rotation prediction

Predict the orientation of the image

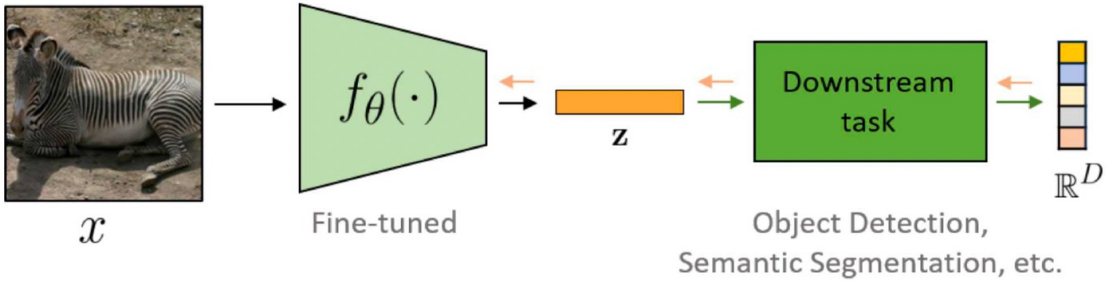


Full processing pipeline

Stage 1: Train network on pretext task (without human labels)

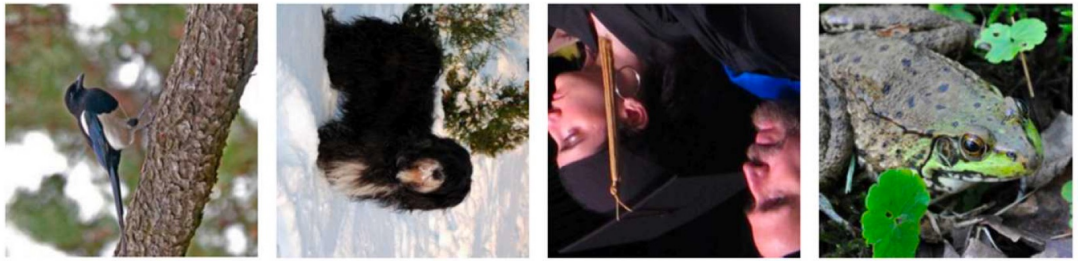


Stage 2: Fine-tune network for new task with fewer labels



Can you guess how much rotated is applied?

Much easier if you recognize the object!



Pros

- Very simple to implement and use, while being quite effective

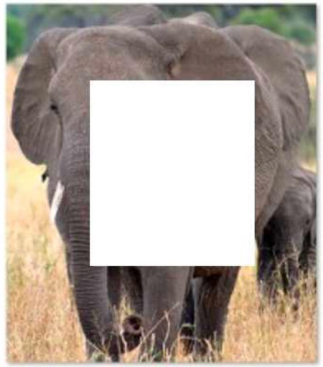
Cons

- Assumes training images are photographed with canonical orientations
- Not fine-grained enough due to no negatives from other images
- Small output space - 4 cases (rotations) to distinguish

Many types of pretext tasks

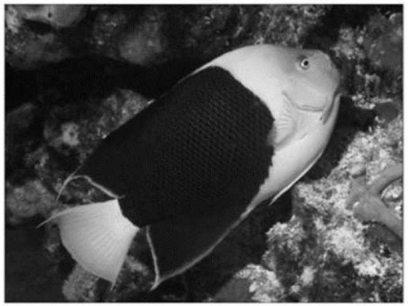
What goes in the middle?

Much easier if you recognize the objects!

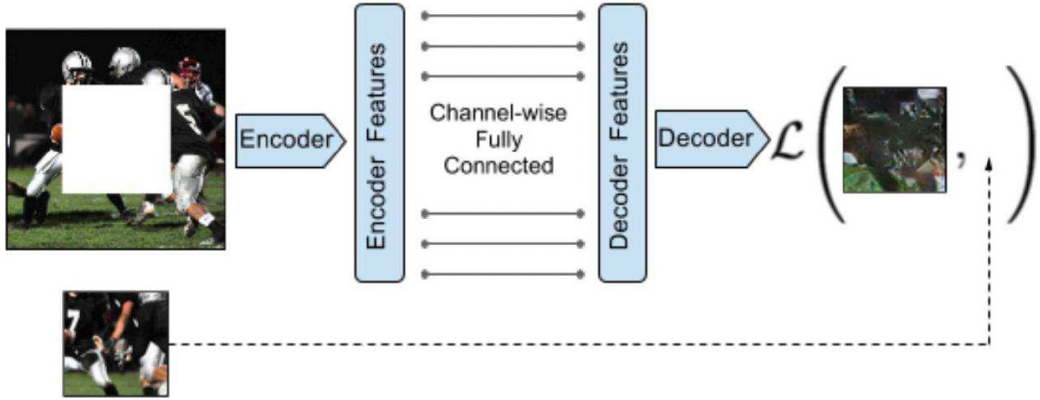


What is the colour of every pixel?

Hard if you don’t recognize the object!



Masking/reconstruction scheme



Pros

- Requires preservation of fine-grained information

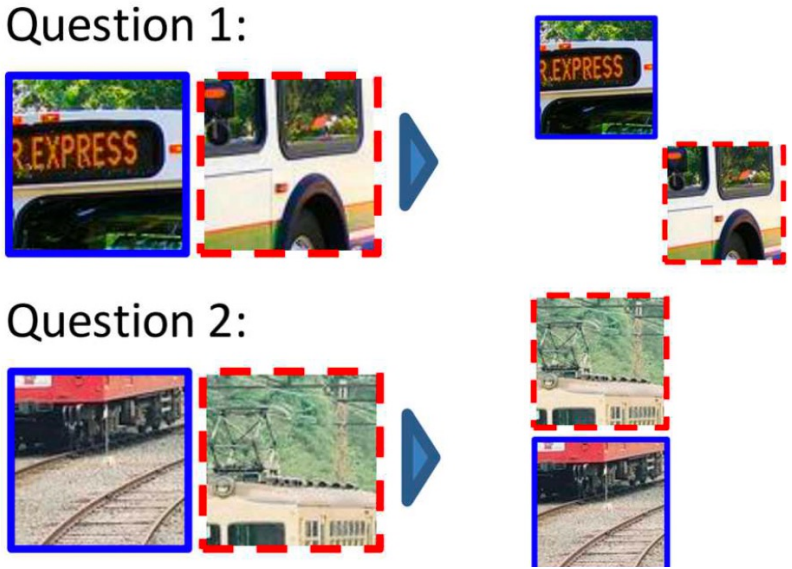
Cons

- Train-eval gap: no masking at eval
- Input reconstruction is too hard and ambiguous
- Lots of effort spent on “useless” details: exact colour, good boundary, etc.

One step further in masking: spatial prediction

Can you guess the spatial configuration for the two pairs of patches?

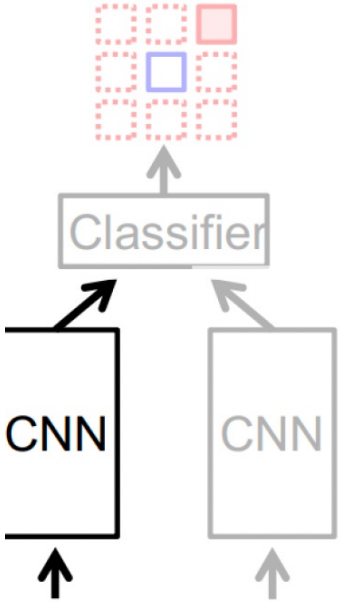
Much easier if you recognize the object!

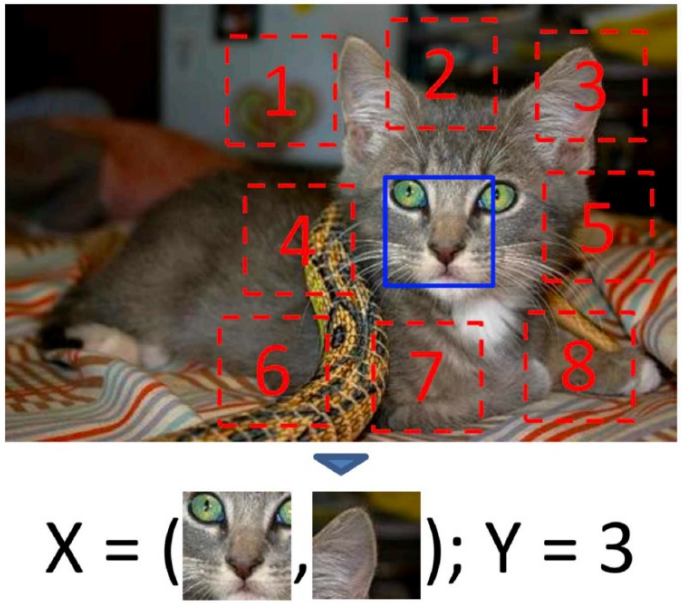


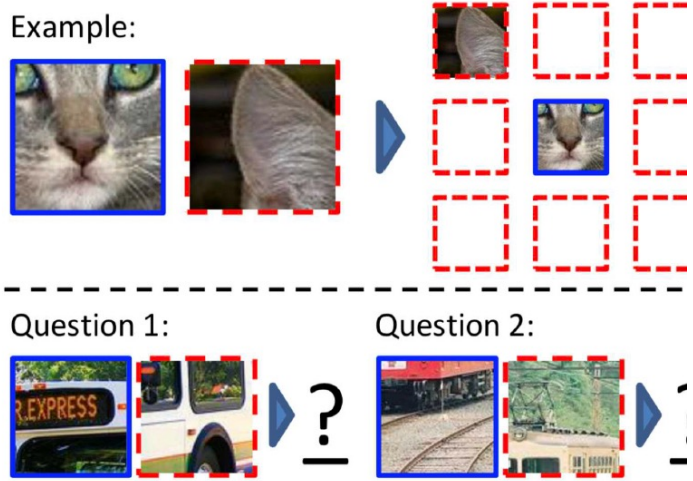
Intuition:

- The network should learn to recognize object parts and their spatial relations

One step further in masking: spatial prediction

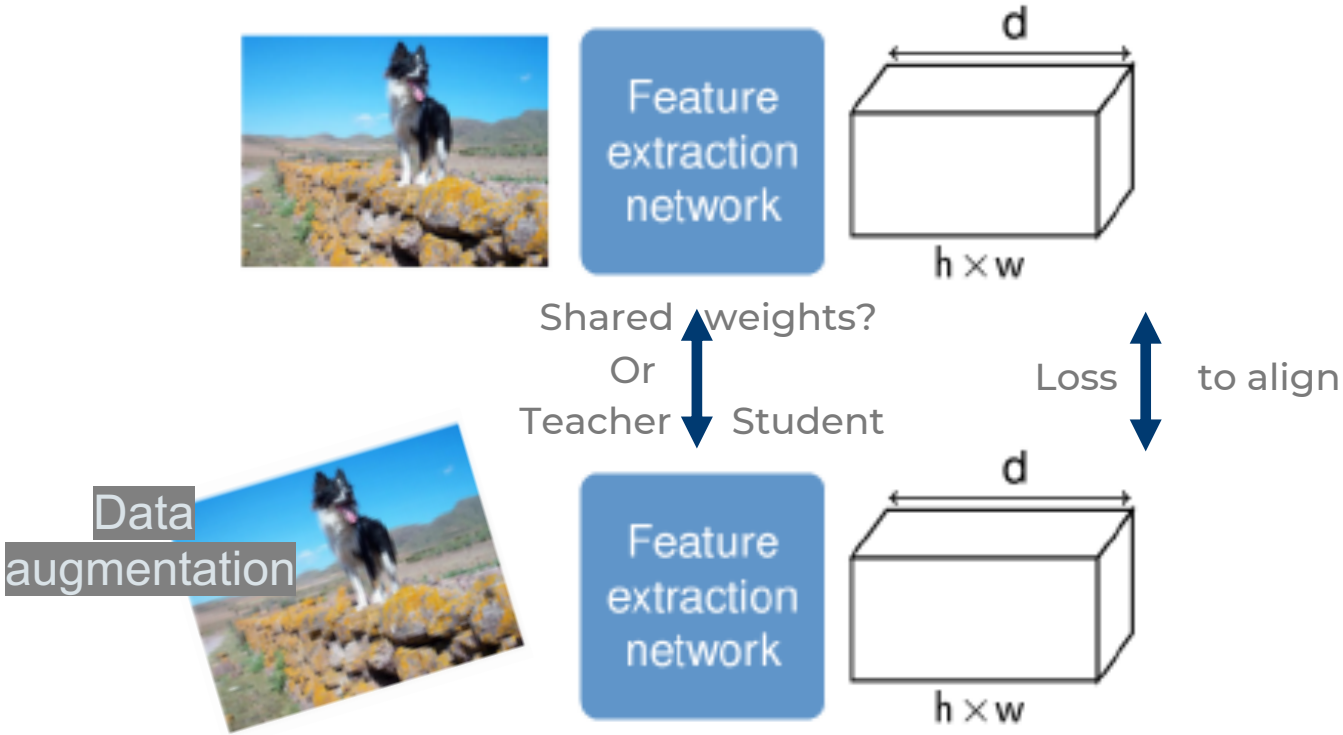






Predict the location of one patch relative to the center patch

Star of SSL: Alignment of representations



Is there a teacher and a student? How to do back-prop?

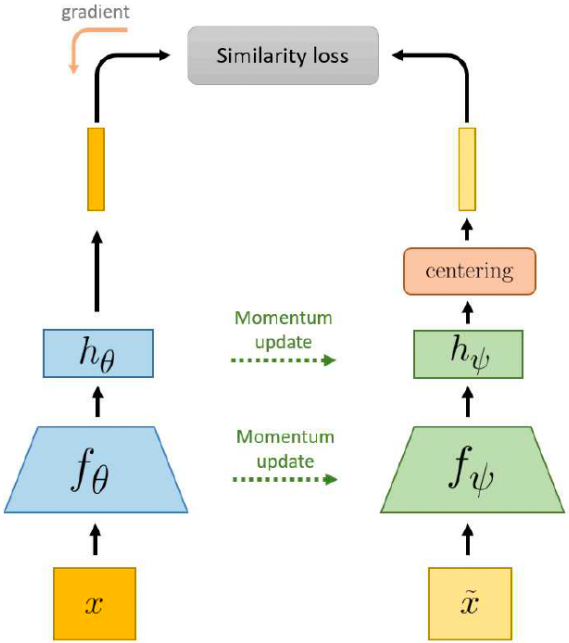
Main pipeline for Alignment: Teacher / Student distillation

Reconstructing the image features at the output of a teacher network could enable representations, as many details are removed via feature abstraction.

Key questions:

- What teacher to use?
- What types of features to use?
- How to outperform the teacher?

The star of SSL methods: DINO APPROACH (V1 V2 V3)



Main idea: No prediction head; post-processing of teacher outputs to avoid feature collapse

- Centering by subtracting the mean feature: prevents collapsing to constant 1-hot targets
- Sharpening by using low softmax temperature: prevents collapsing to a uniform target vector
- Cross-entropy loss

f_θ, f_ψ: encoder (ViT, ResNet-50);

h_θ, h_ψ: projection (MLP).

Training usually using contrastive setting with positive and negative pairs (as for CLIP)
A lot of improvement, in particular with massive curated dataset for training

Remember BERT?

Input: The man went to the $[MASK]_1$. He bought a $[MASK]_2$ of milk .
Labels: $[MASK]_1$ = store; $[MASK]_2$ = gallon

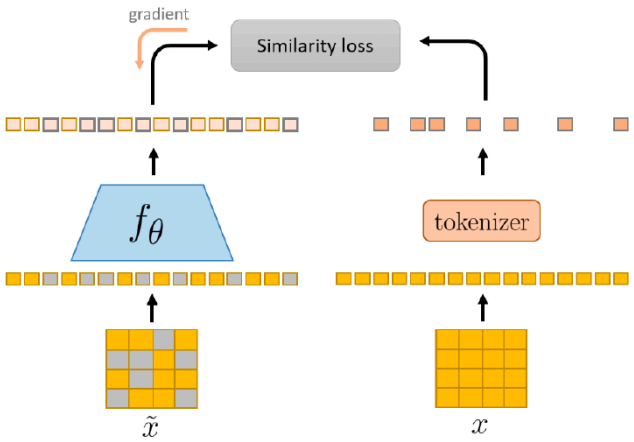
Missing word prediction task.

Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence

Next sentence prediction task.

BEiT (Masked Image Modelling)



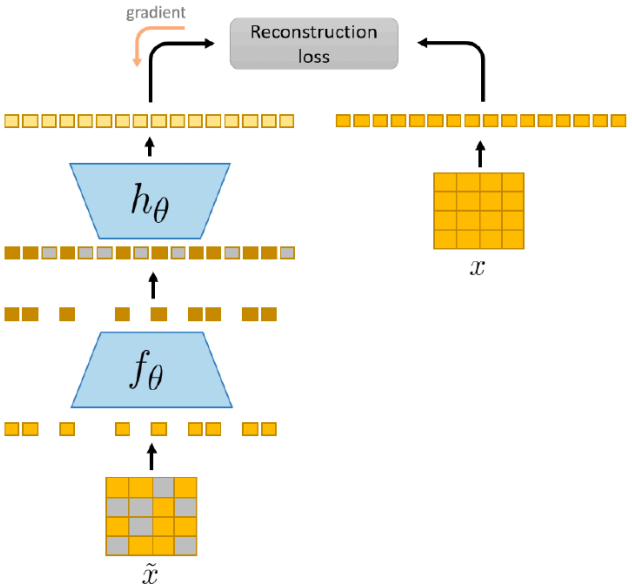
Main idea: pre-train ViTs by learning to predict tokens of masked patches

- Mimicking practices from large language models (BERT)
- Learn to produce discrete visual tokens from masked input images
- Use learnable mask-token for masked patches
- Trained with cross-entropy loss over masked tokens

f_θ: encoder (ViT);

`tokenizer`: pretrained autoencoder (DALL-E).

MAE (Masked Image Modelling)



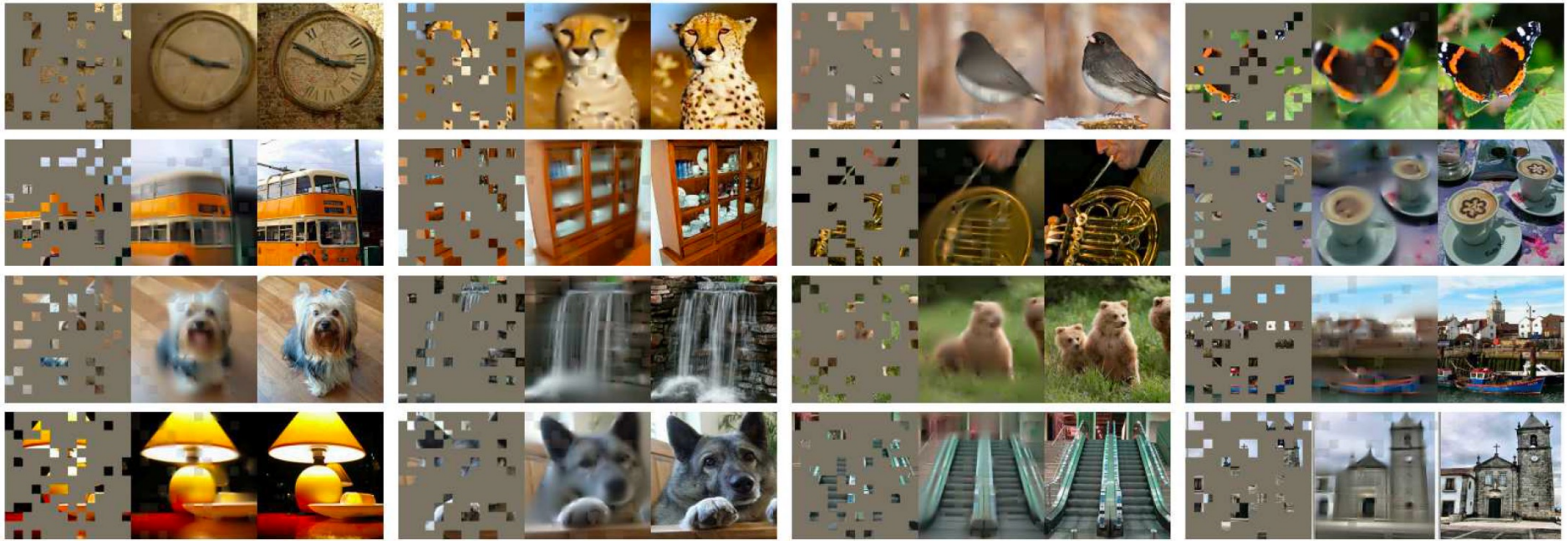
Main idea: learn to reconstruct masked pixels

- Simplified MIM pipeline without pre-trained tokenizer nor data augmentation
- Encoder operates only on visible patches without mask tokens
- Lightweight ViT decoder (removed after pre-training)
- Aggressive masking (up to 75% of patches)
- Shines when fine-tuned on the downstream task

f_θ: encoder (ViT);

h_θ: decoder (ViT).

MAE results



Example results on ImageNet validation images. For each triplet, we show the masked image (left), our MAE reconstruction(middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches

To evaluate representations learned by SSL: several (downstream) tasks

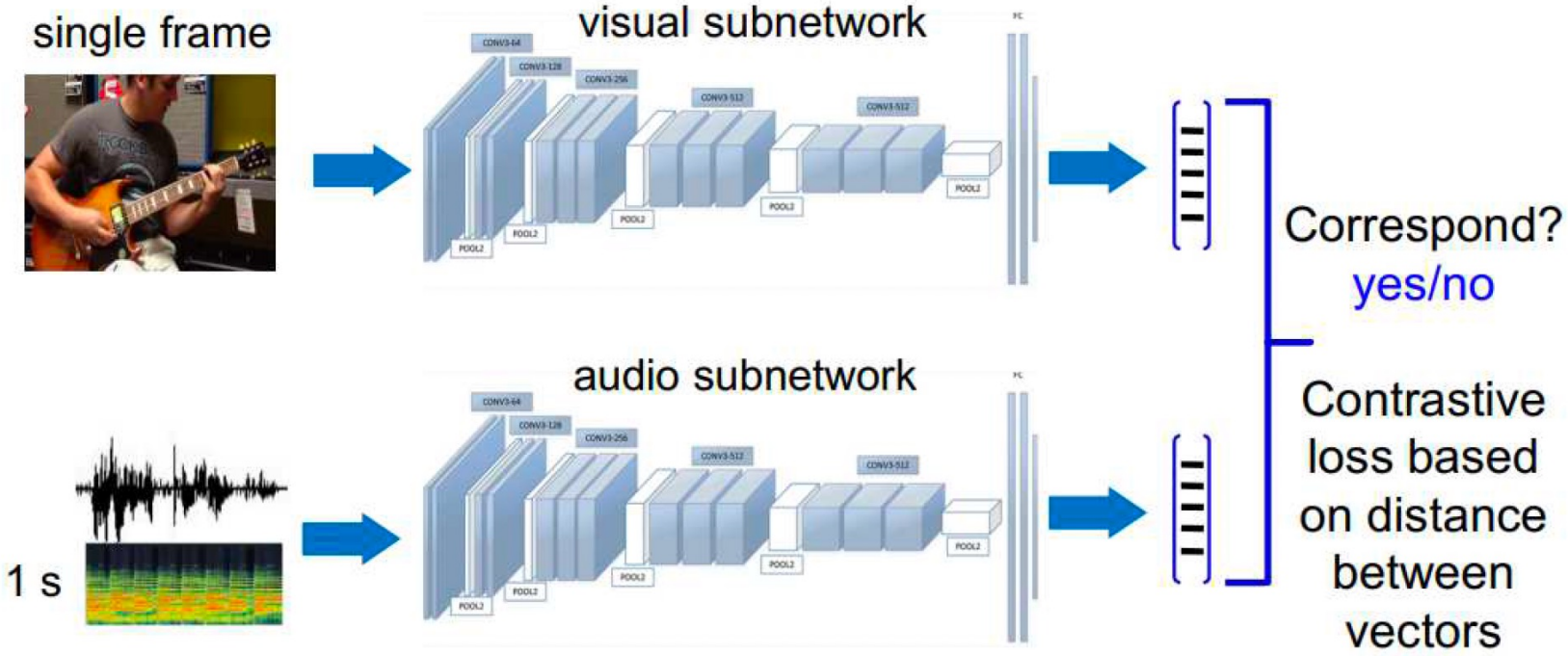
Linear classification

Few-shot learning

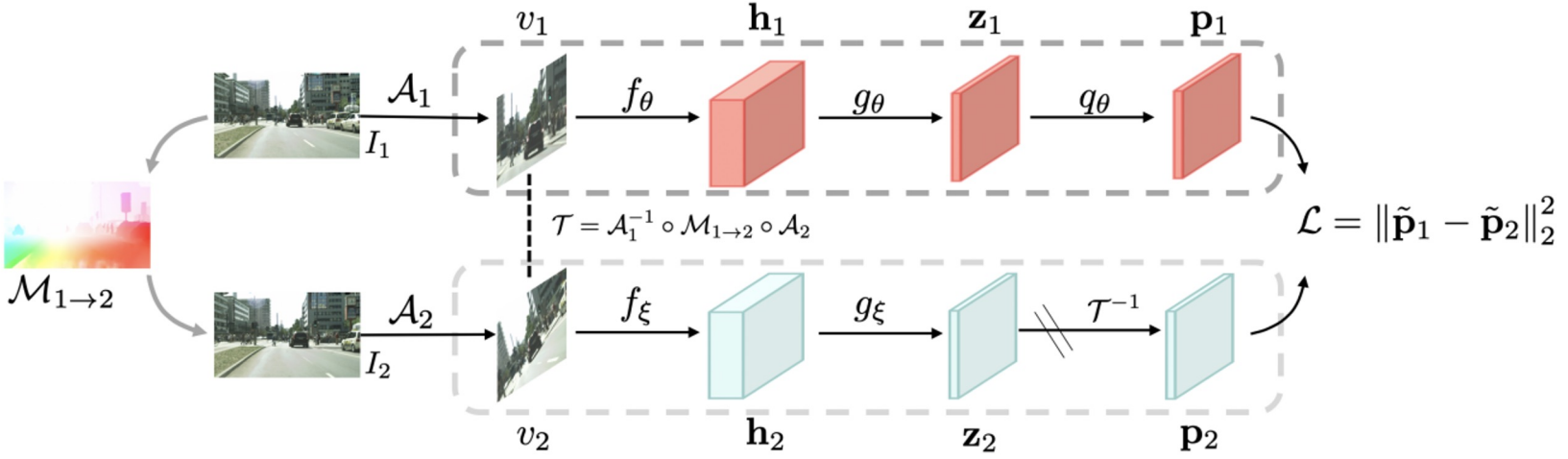
Efficient learning

Transfer learning

Extension of SSL for video, multimodal data ...



Exploiting videos to learn better image representations in SSL

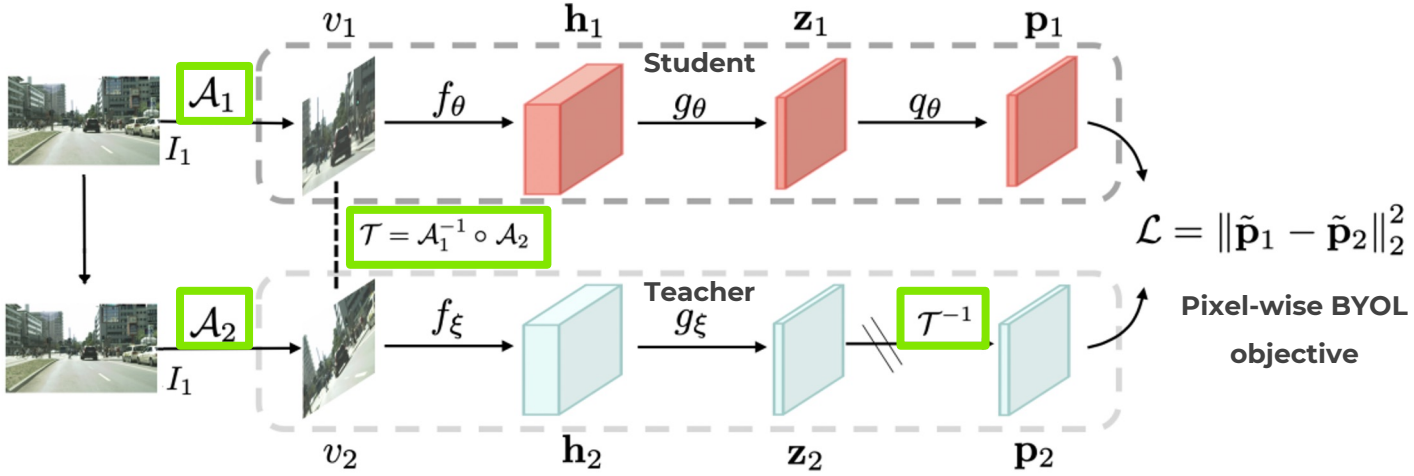


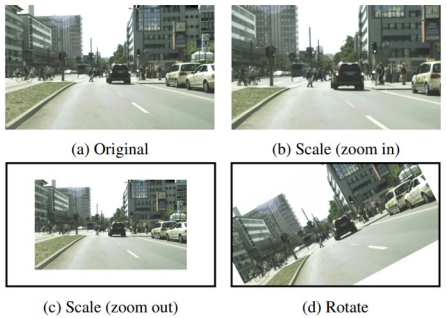
Main idea: use video and its optical flow to learn better pixel-wise image representations

Pixel-wise SSL objective

Optical flow between two frames as an extra transformation ⇒ Flow-equivariant features

Learning image representations from only still images

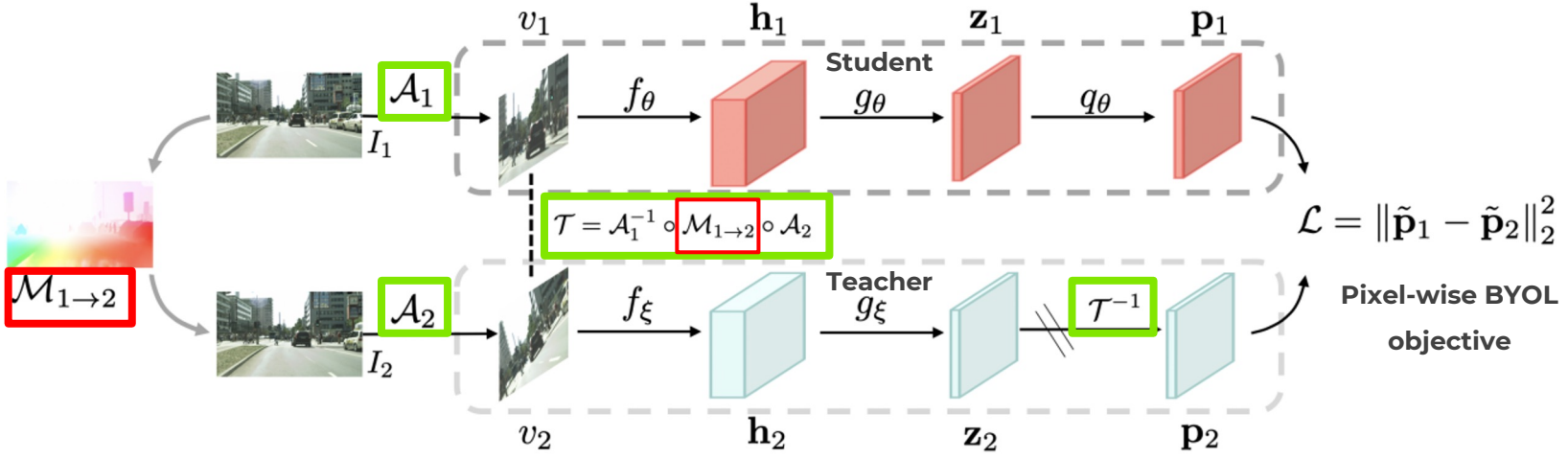


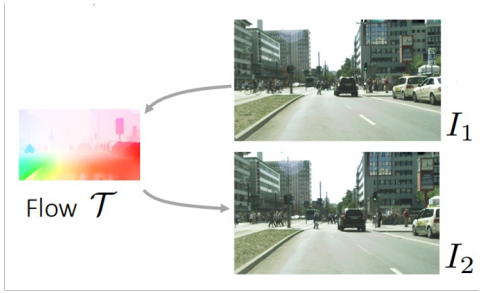


Affine transformations

Input views: affine transforms of a single image
Pixel correspondences: know transform between views

Learning image representations from videos





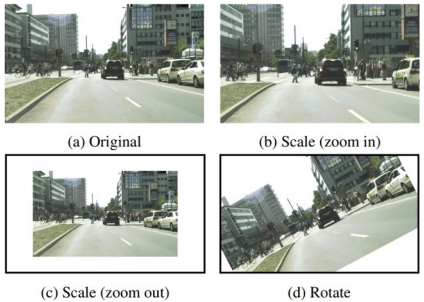
Video frames + optical flow

Input views: from two consecutive video frames
Pixel correspondences: with optical flow between frames

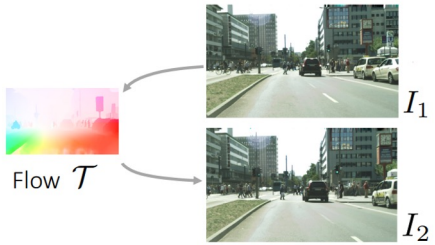
Exploit videos to learn better image representations in SSL

Data augmentations:

Affine transformations



Video frames + optical flow

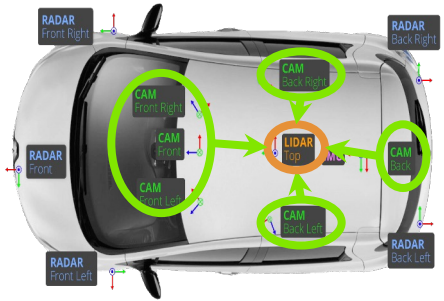


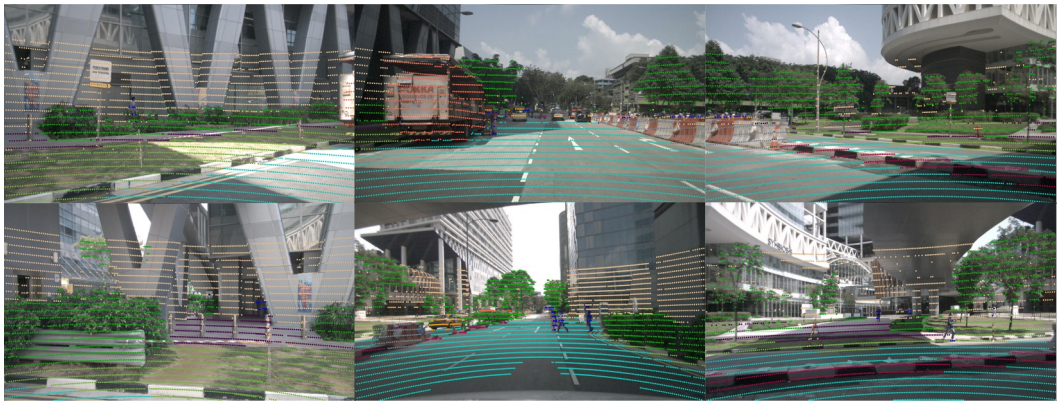
Impact of the affine and optical flow transformations on the downstream tasks of semantic segmentation (mIoU) and instance segmentation (mAP) on UrbanCity

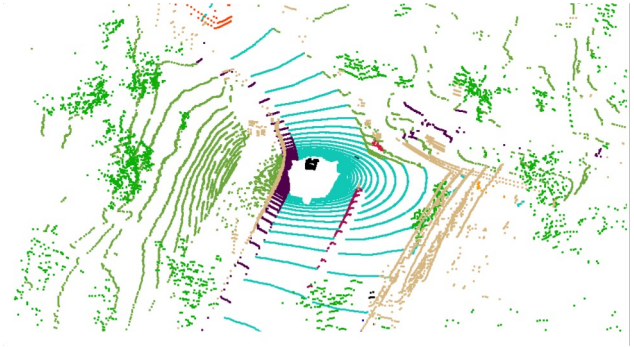
Affine transform	Optical flow	mIoU†	mAP†
		40.1	12.3
✓		45.9	15.1
	✓	51.9	16.2
✓	✓	**53.2**	**16.5**

Extension of SSL for video, multimodal data …

Cameras and LiDAR are synchronized and calibrated





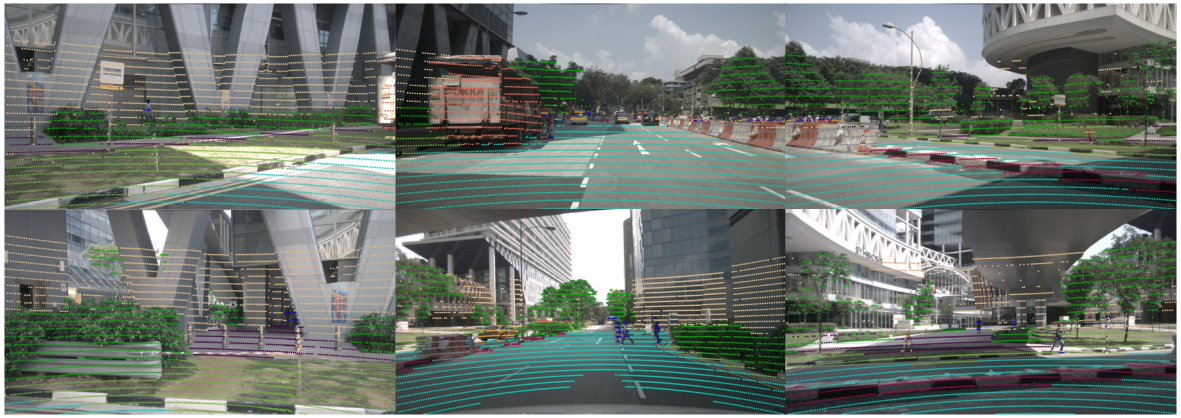


Exploiting videos for self-supervised learning of image representations

Image-to-Lidar Self-Supervised Distillation

Main application: semantic segmentation and object detection of point clouds
Data: **synchronized and calibrated LiDAR and camera sensors**

Use cameras for 3D network pre-training without using any annotations



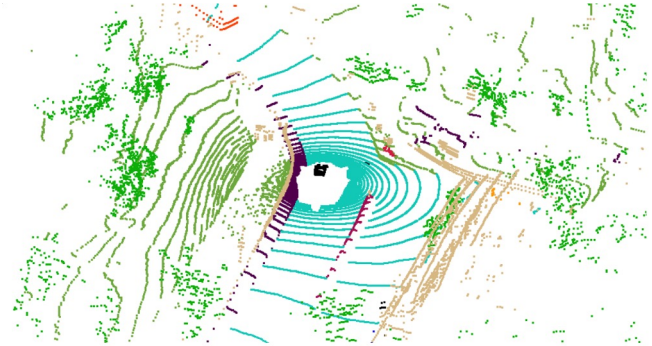
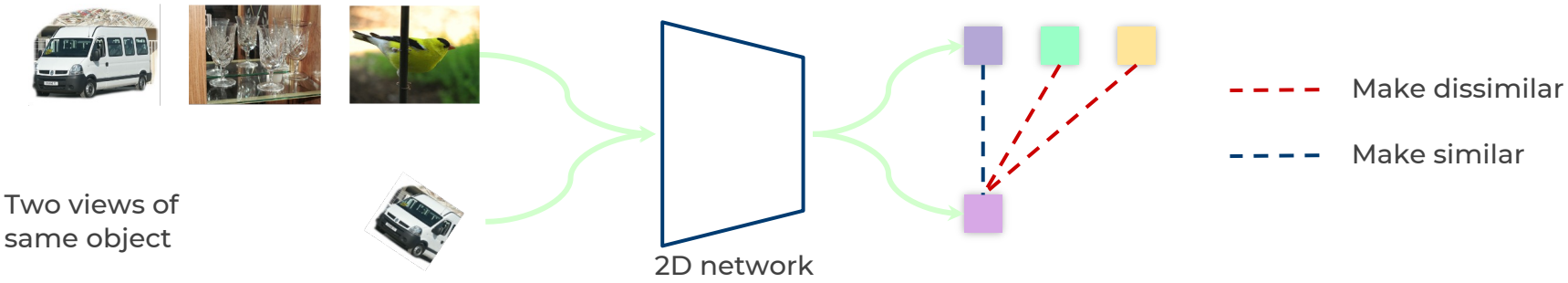


Image-to-Lidar Self-Supervised Distillation

Step 1: 2D networks pre-trained beforehand **without** annotations (DINO, etc.)



Step 2: knowledge distillation using image-lidar data

Image-to-Lidar Self-Supervised Distillation

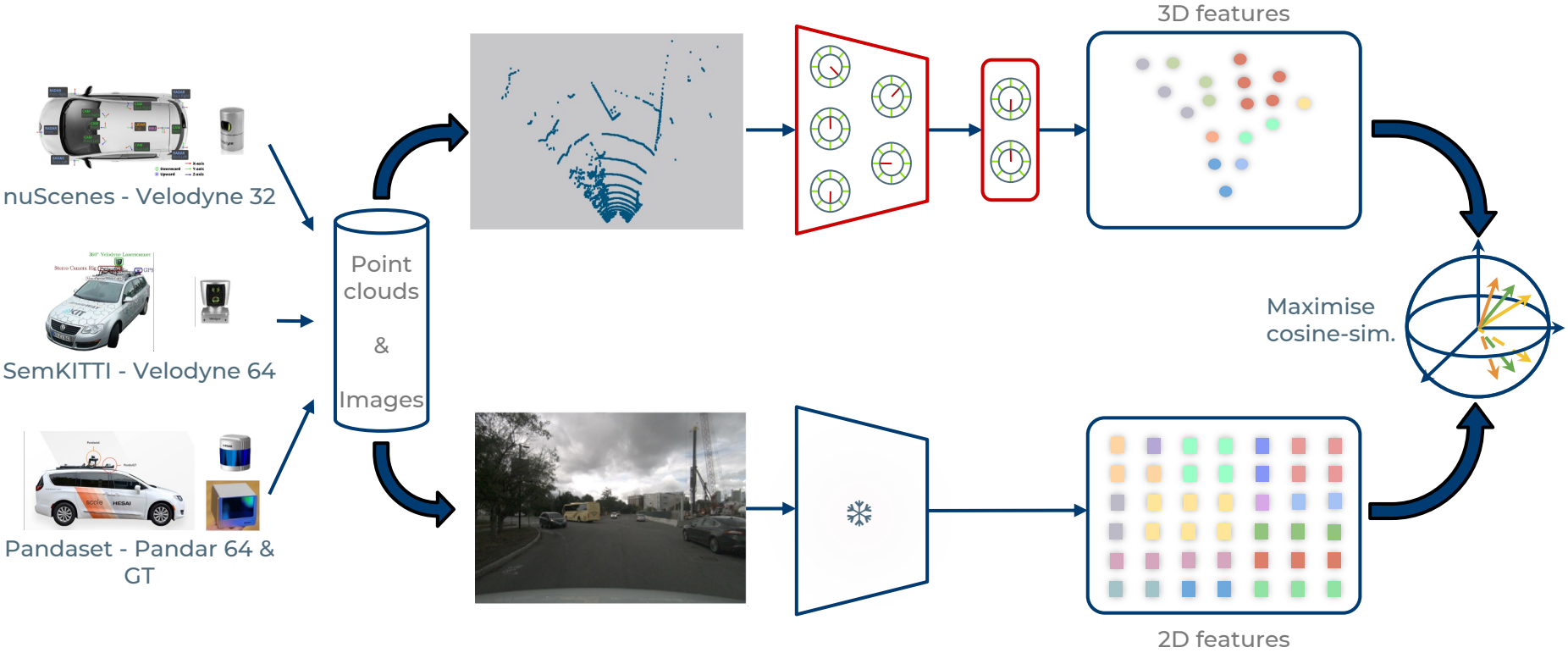
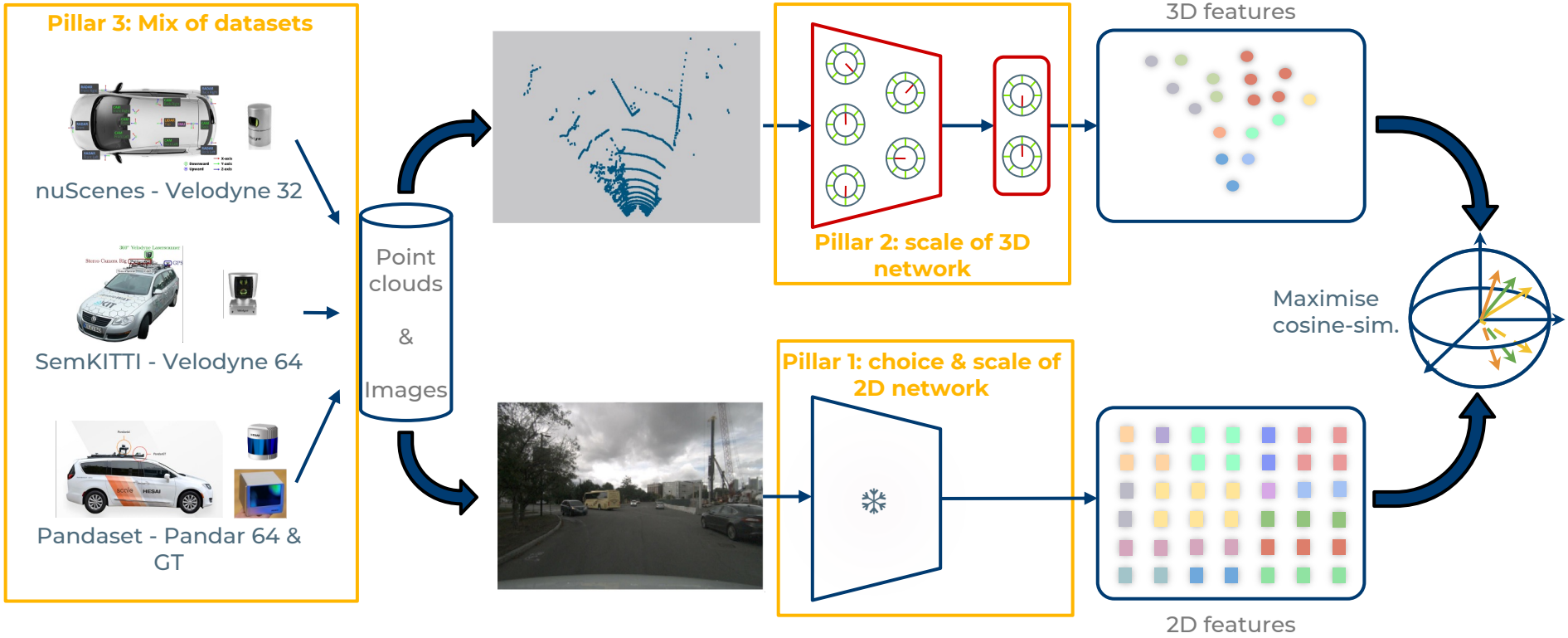
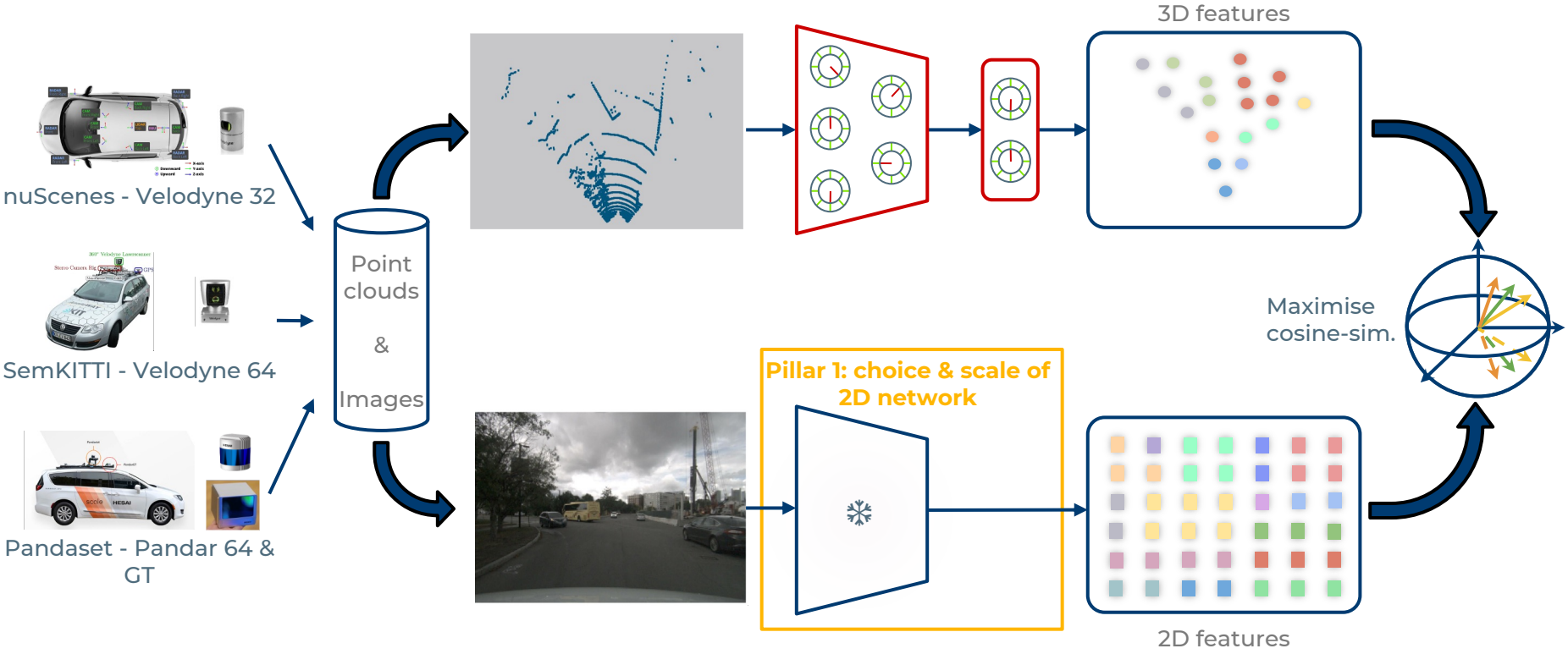


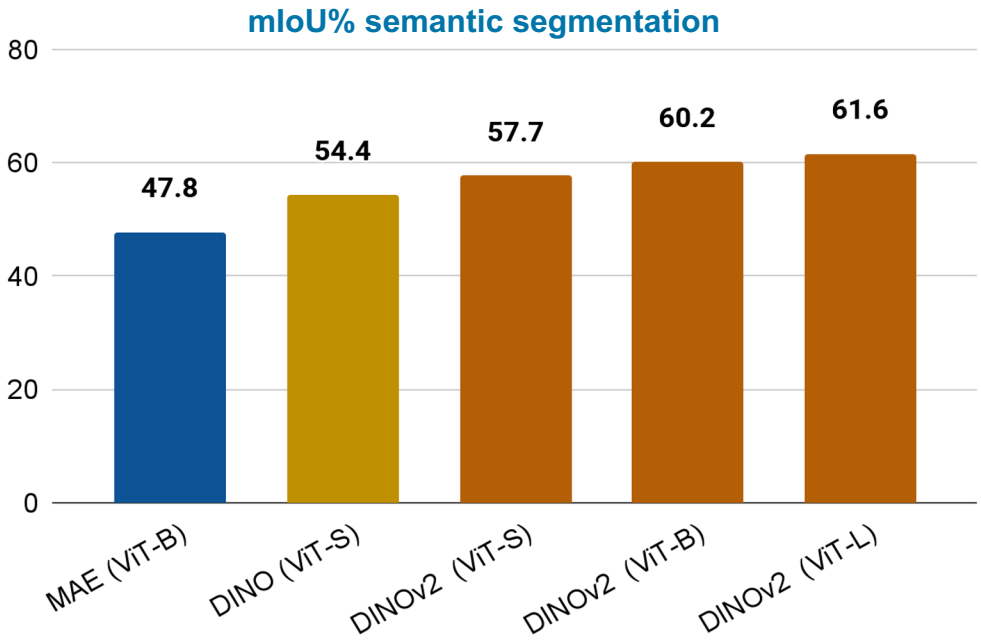
Image-to-Lidar Self-Supervised Distillation



Pillar 1: choice & scale of 2D network

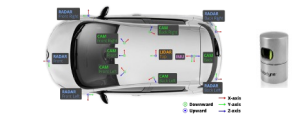


Pillar 1: choice & scale of 2D network



Pretraining & linear probing on nuScenes - Distilled into WaffleIron-256 (features of size 256)

Pillar 2: scale of 3D network

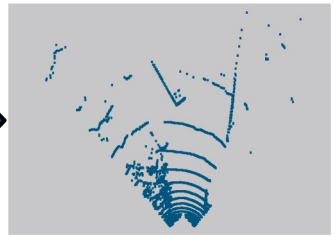


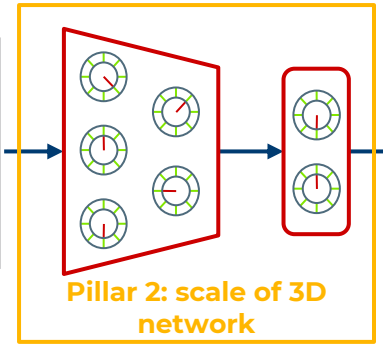
nuScenes - Velodyne 32

SemKITTI - Velodyne 64

Pandaset - Pandar 64 & GT

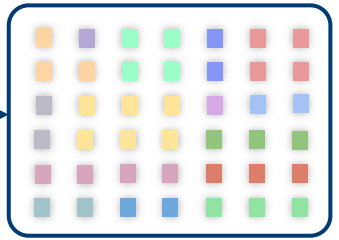
Point clouds & Images





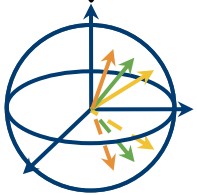
Pillar 2: scale of 3D network

3D features

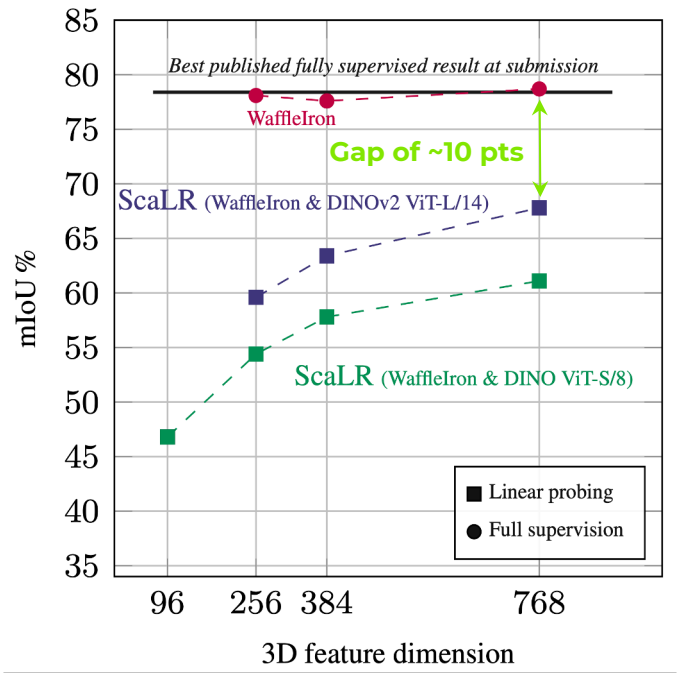


2D features

Maximise cosine-sim.

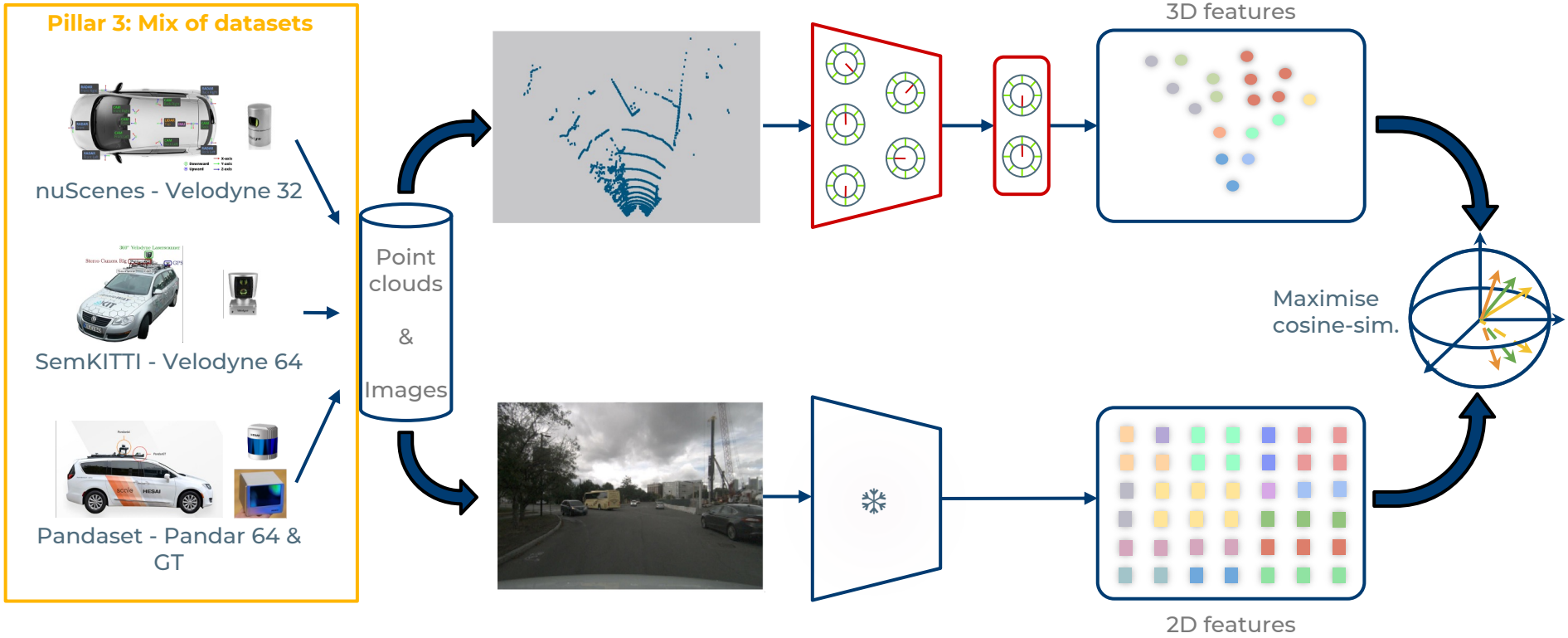


Pillar 2: scale of 3D network



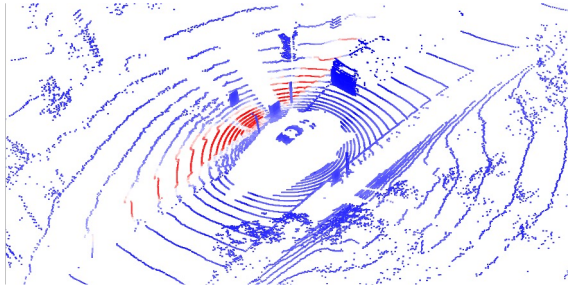
Pretraining & linear probing on nuScenes - Distilled into WaffleIron with varying feature sizes

Pillar 3: Mix of datasets

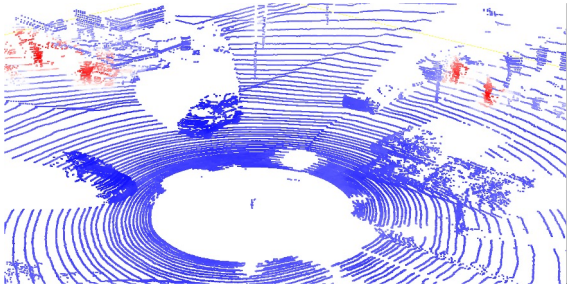


Qualitative results

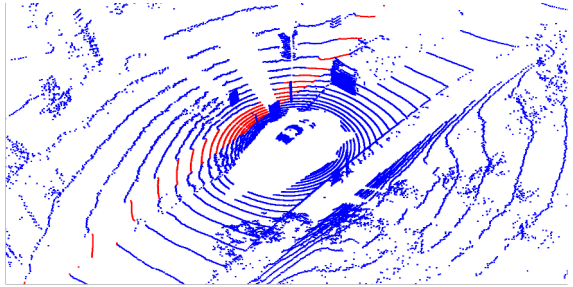
Correlation maps with class prototypes

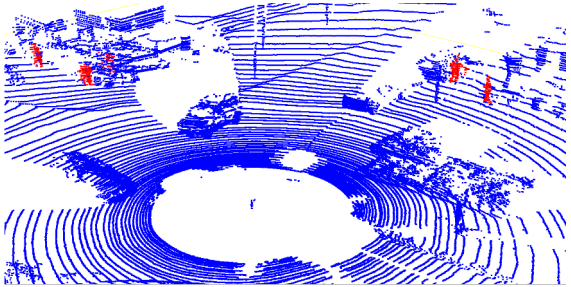


nuScenes - Sidewalk



Sem.KITTI - Pedestrian

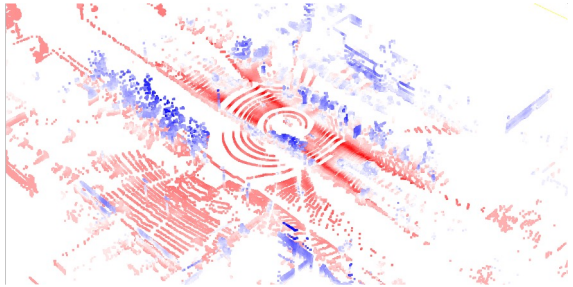




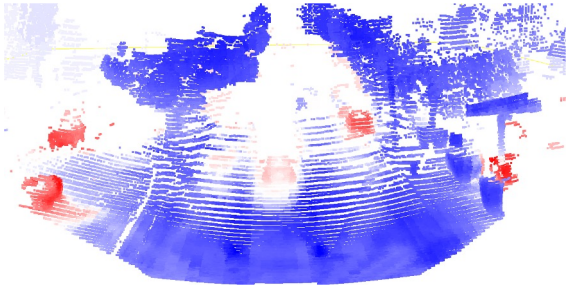
Ground truth

Qualitative results

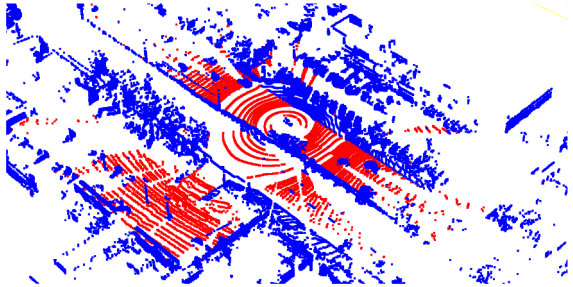
Correlation maps with class prototypes

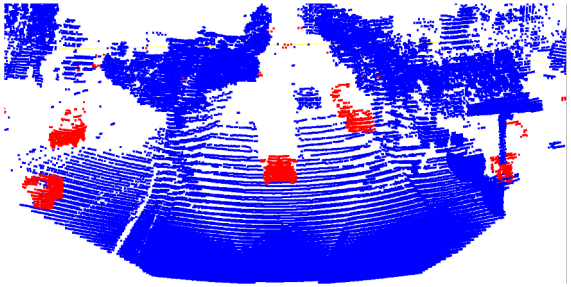


PandaSet 64 - Road



PandaSet GT - Car





Ground truth

Image-Language to Lidar Self-Supervised Distillation

Open-vocabulary 3D semantic occupancy prediction

Training:

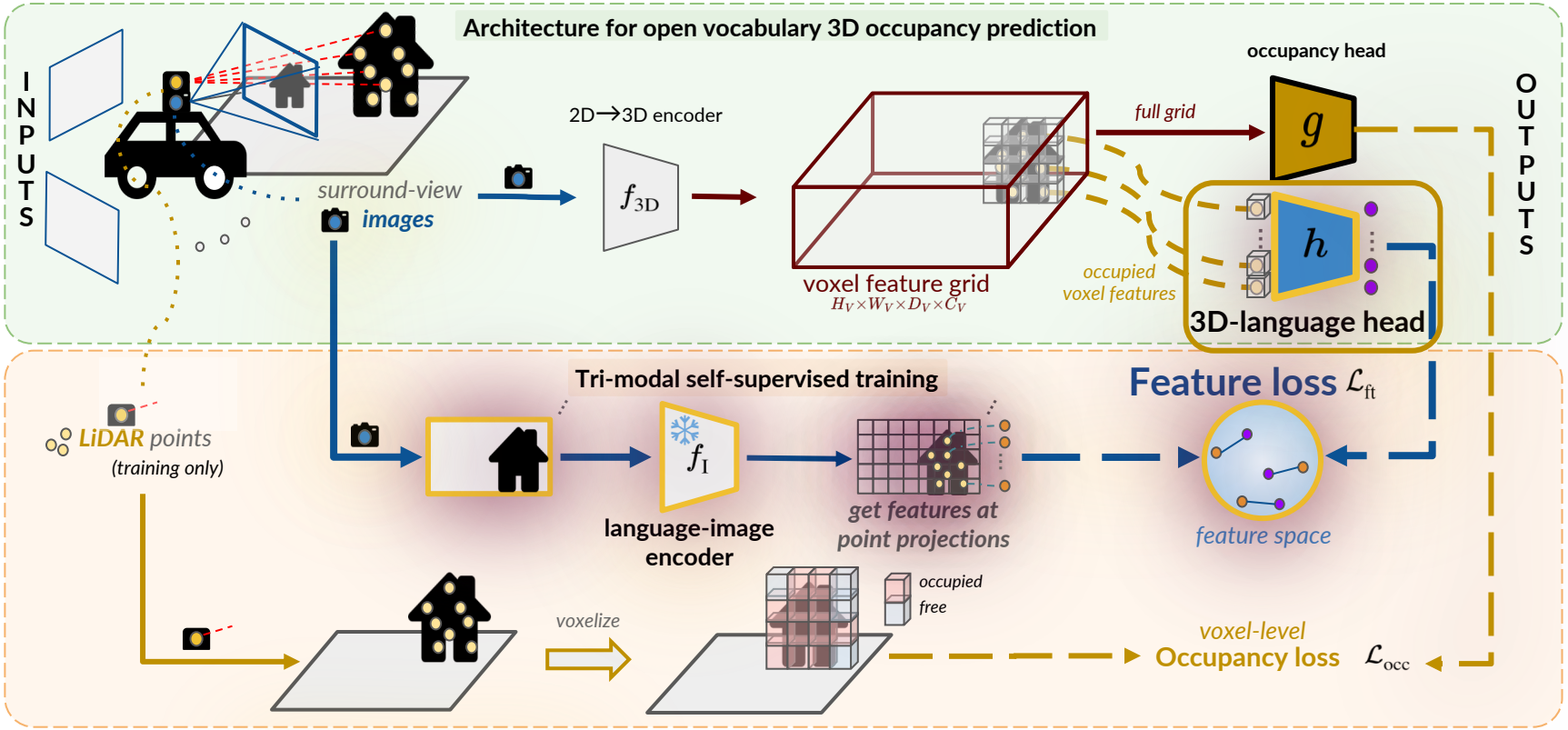
unlabeled image-LiDAR data and a **pre-trained image-language model**

Inference:

only images + text queries

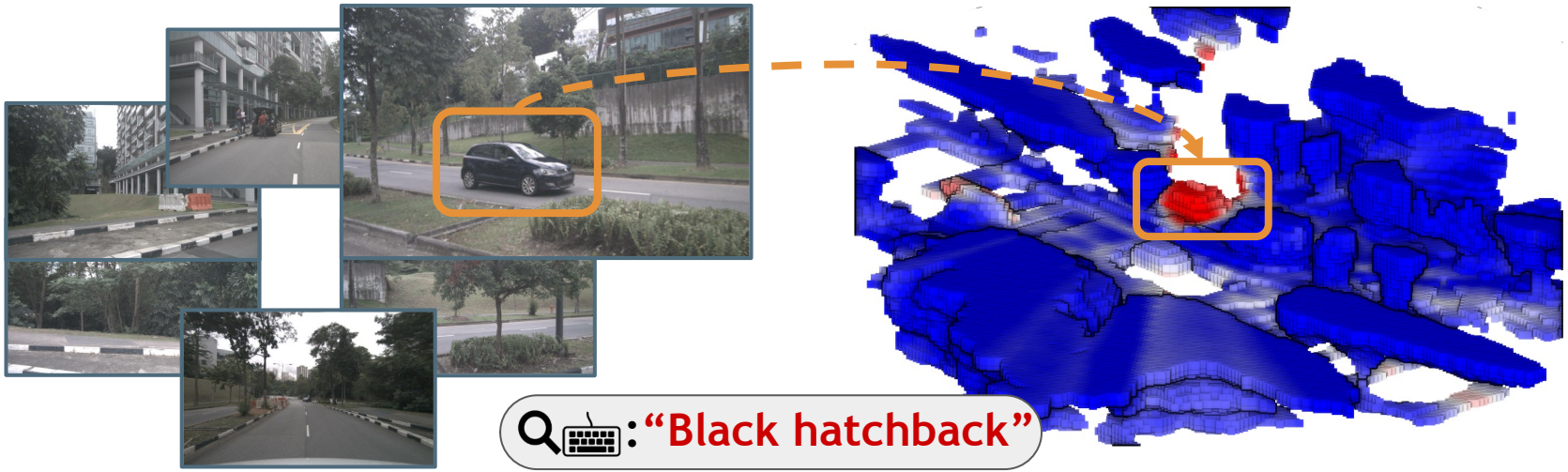
3D-language Losses with Language-image encoder

One step further: pretrained Langugage-image encoder (like CLIP) distilled to 3D



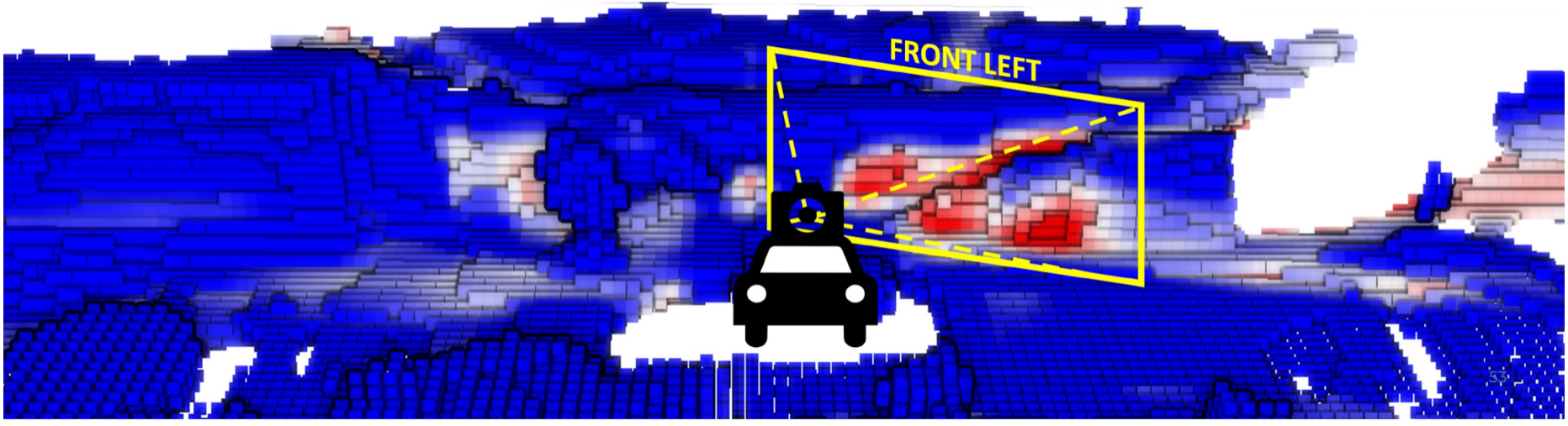
Feature loss: MSE loss **Occupancy loss**: CE losses

Qualitative results: retrieval

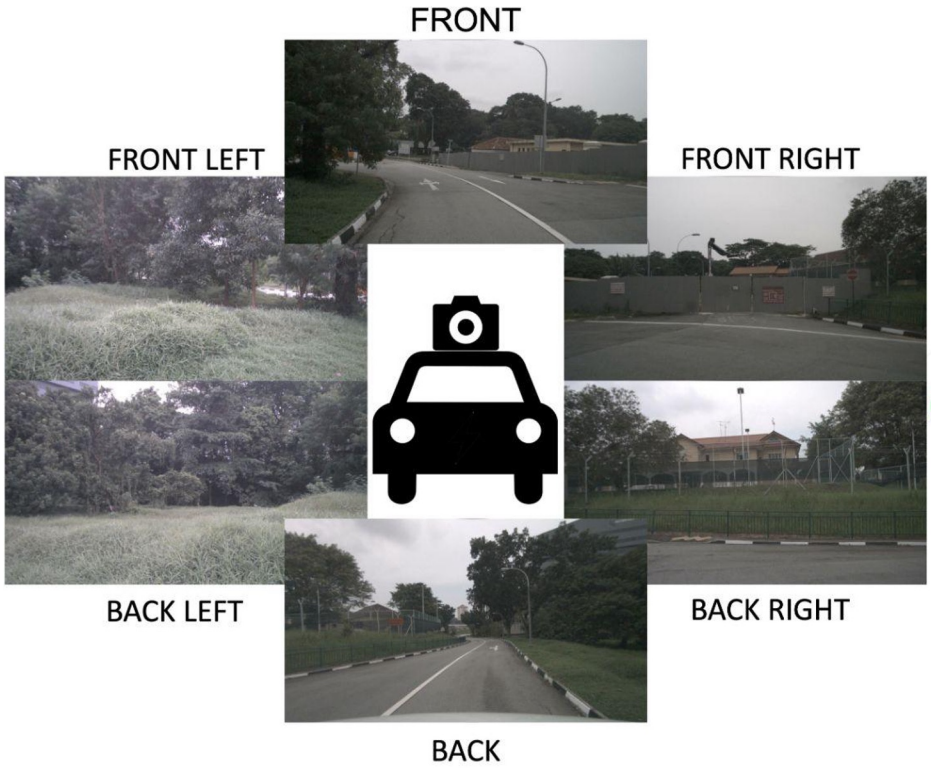


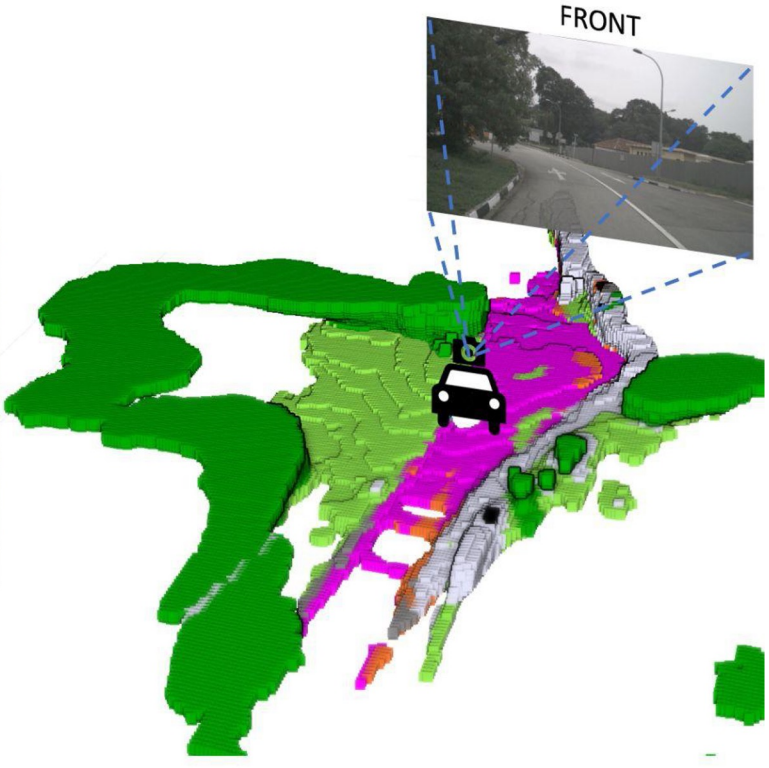
Qualitative results: retrieval

: “stairs”



Qualitative results: zero-shot semantic segmentation

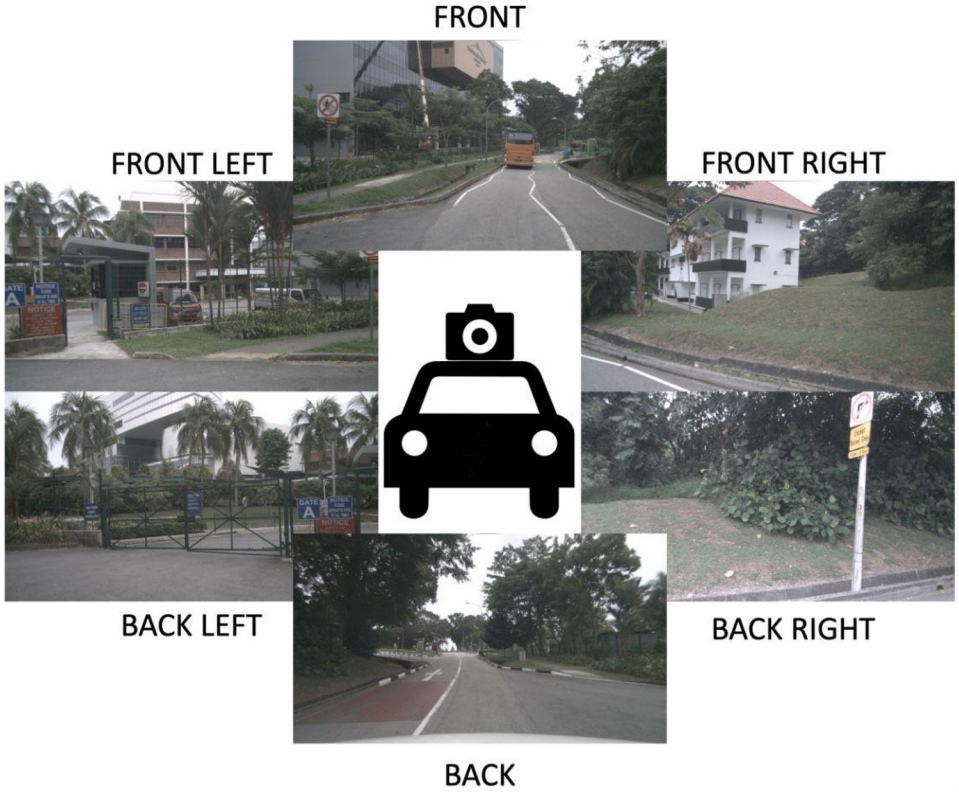


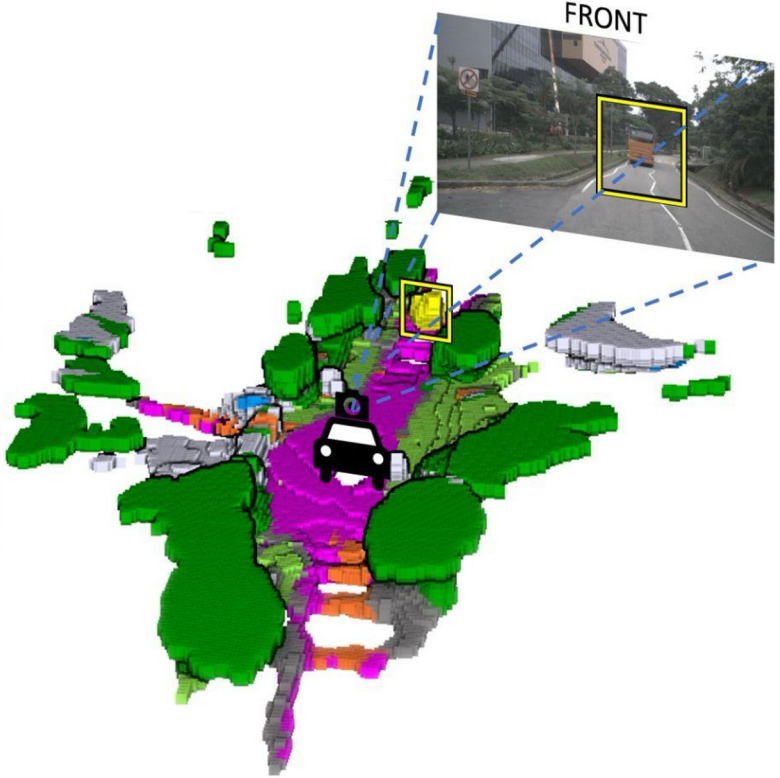


driveable surface | car | bus | truck | terrain | vegetation | sidewalk | other flat | pedestrian | bicycle

manmade | motorcycle | barrier | construction vehicle | trailer | traffic cone

Qualitative results: zero-shot semantic segmentation





driveable surface | car | bus | truck | terrain | vegetation | sidewalk | other flat | pedestrian | bicycle

manmade | motorcycle | barrier | construction vehicle | trailer | traffic cone