since 1978
동서울대학교
DONG SEOUL UNIVERSITY

Research on embedded system porting of SemiVL: Semi-Supervised Semantic Segmentation with Vision-Language Guidance

2024. 08. 06

Computer Software

박승민

Written by Park Seung Min

Contents

Last Q&A

Semi-supervised semantic segmentation분야에서 GAN이 더 안 쓰이는 이유

GAN은 Generator와 Discriminator 두 네트워크를 동시에 훈련해야 하기 때문에 Consistency Learning에 비해 훈련이 복잡하며, 종종 mode collapse와 같은 문제가 보입니다.

Unlabeled Image에 대한 Consistency training에서 Encoder와 Decoder의 역할

Image Encoder : 입력 이미지를 받아 feature를 추출합니다. perturbation을 통해 변형된 이미지 x^{fp}를 생성합니다
Decoder : Encoder를 거쳐 나온 feature를 입력 받아 최종 예측 결과를 생성합니다

Dense CLIP Guidance에서 Encoder가 3개 쓰인 것인가?

Encoder는 모두 CLIP으로부터 가져온 것으로, Unlabeled Image에 대한 학습과정에서 생길 수 있는 overfitting을 방지하기 위해 Unlabeled Image는 spatial fine tunning을 적용한 Image encoder 하나와 기존 CLIP에서 freeze한 Image encoder를 통과합니다. 마지막으로 text가 통과하는 Text encoder까지 3개가 쓰였습니다

1. Introduction

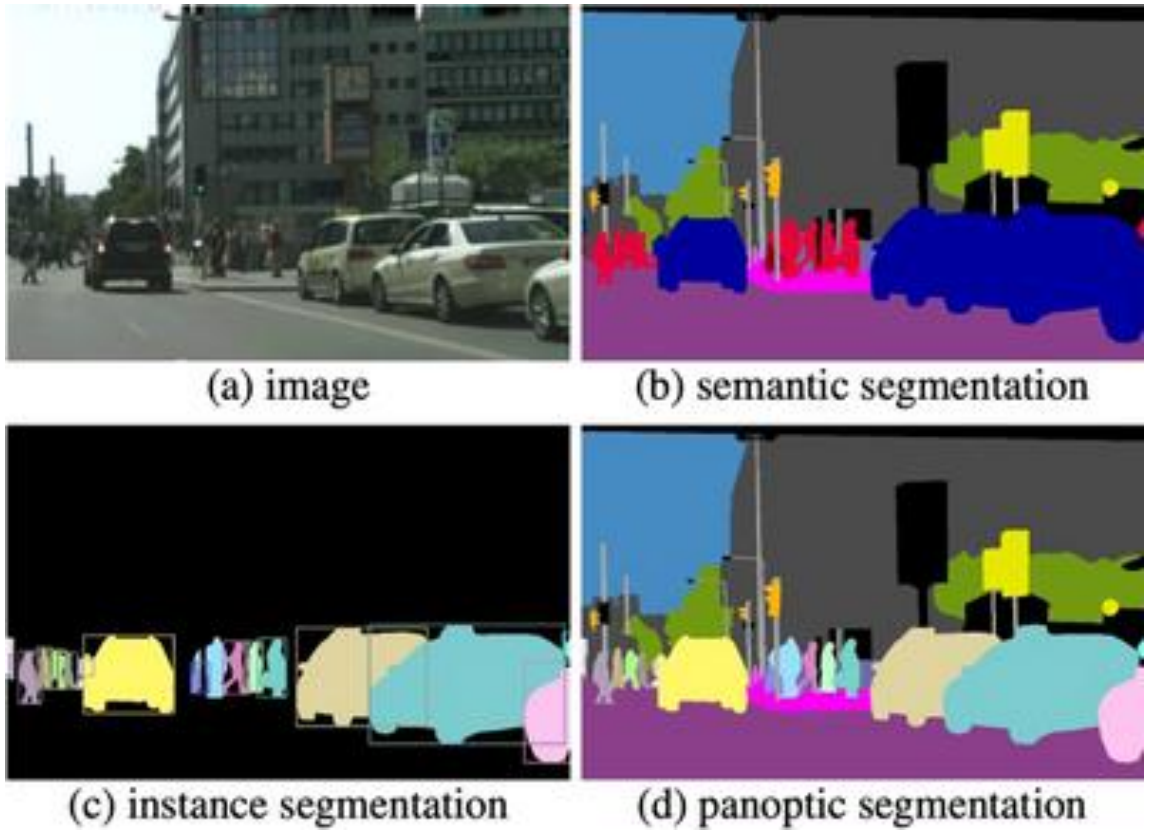
Sementic Segmentation

- computer vision task in which the goal is to categorize each pixel in an image into a class

(b) Segmentation by class (Car, Person,...)

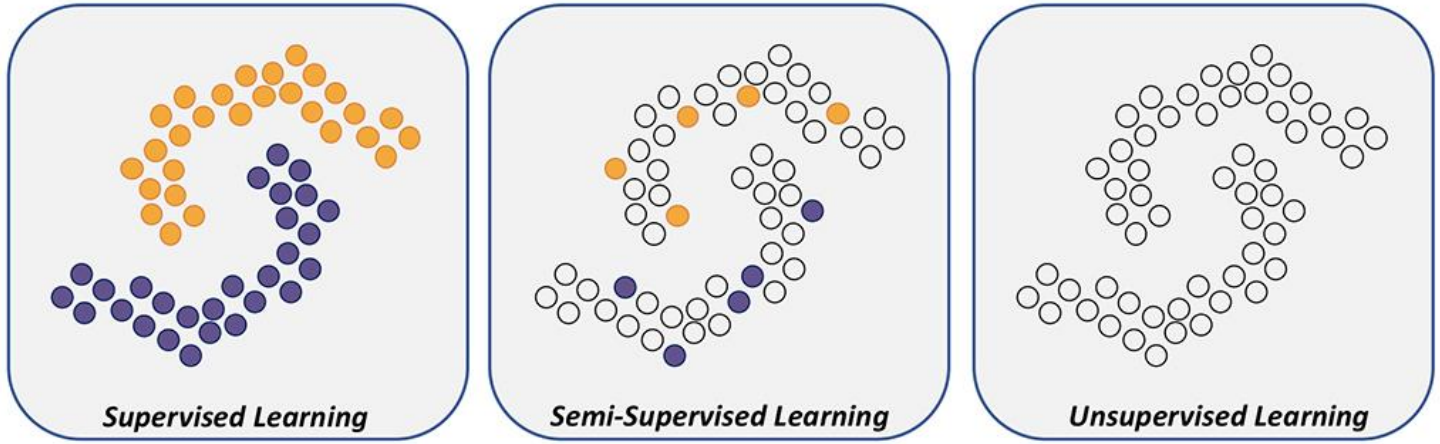
(c) Segmentation by instance (Car1, Car2,..., Person1)

(d) Overall segmentation (Instance + backbone)



Semi-supervised learning

- Semi-supervised learning uses both labeled and unlabeled data
- Semi-supervised learning aims to utilize additional information in one piece of data to improve performance in training on another piece of data.
- Emerged because of the high resource and cost of "data labeling" to collect correct answer data

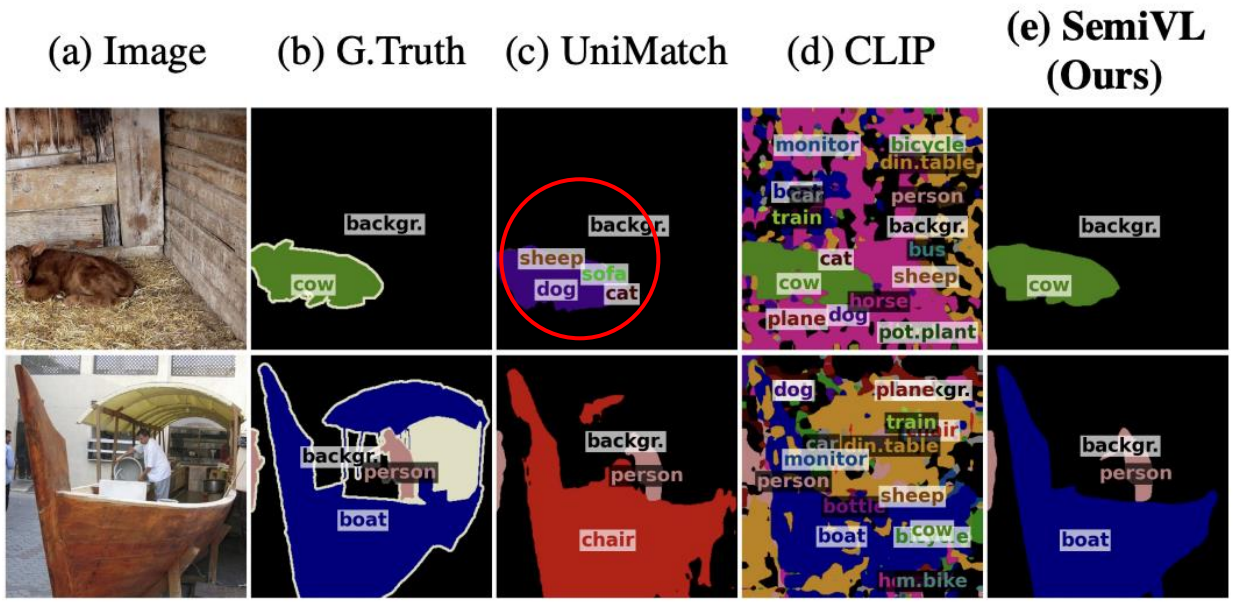


1. Introduction

Semi-supervised Sementic Segmentation

- The current SOTA, UniMatch model, learned segmentation masks well, but segments with similar visual features had difficulty learning accurate semantic decision boundaries.

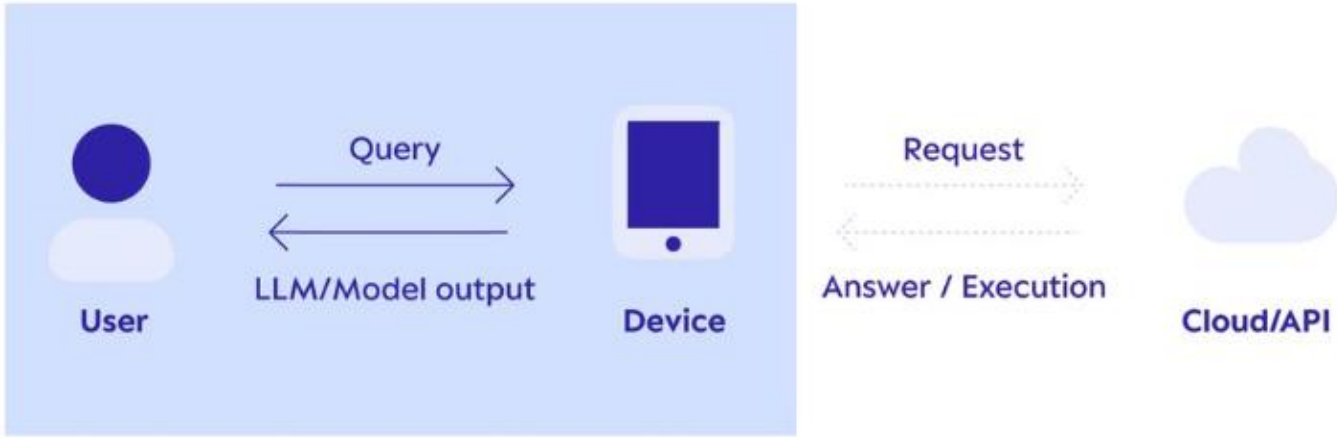
- To capture richer semantics, we propose that SimVL complements Semi-Supervised Semantic Segmentation by utilizing guidance from Vision Language Models (VLMs)



On-device AI

- On-device AI refers to artificial intelligence models and algorithms running directly on a device, such as a smartphone or IoT device, rather than relying on cloud-based servers. This approach is crucial for ensuring low latency, enhanced privacy, and reduced dependency on internet connectivity

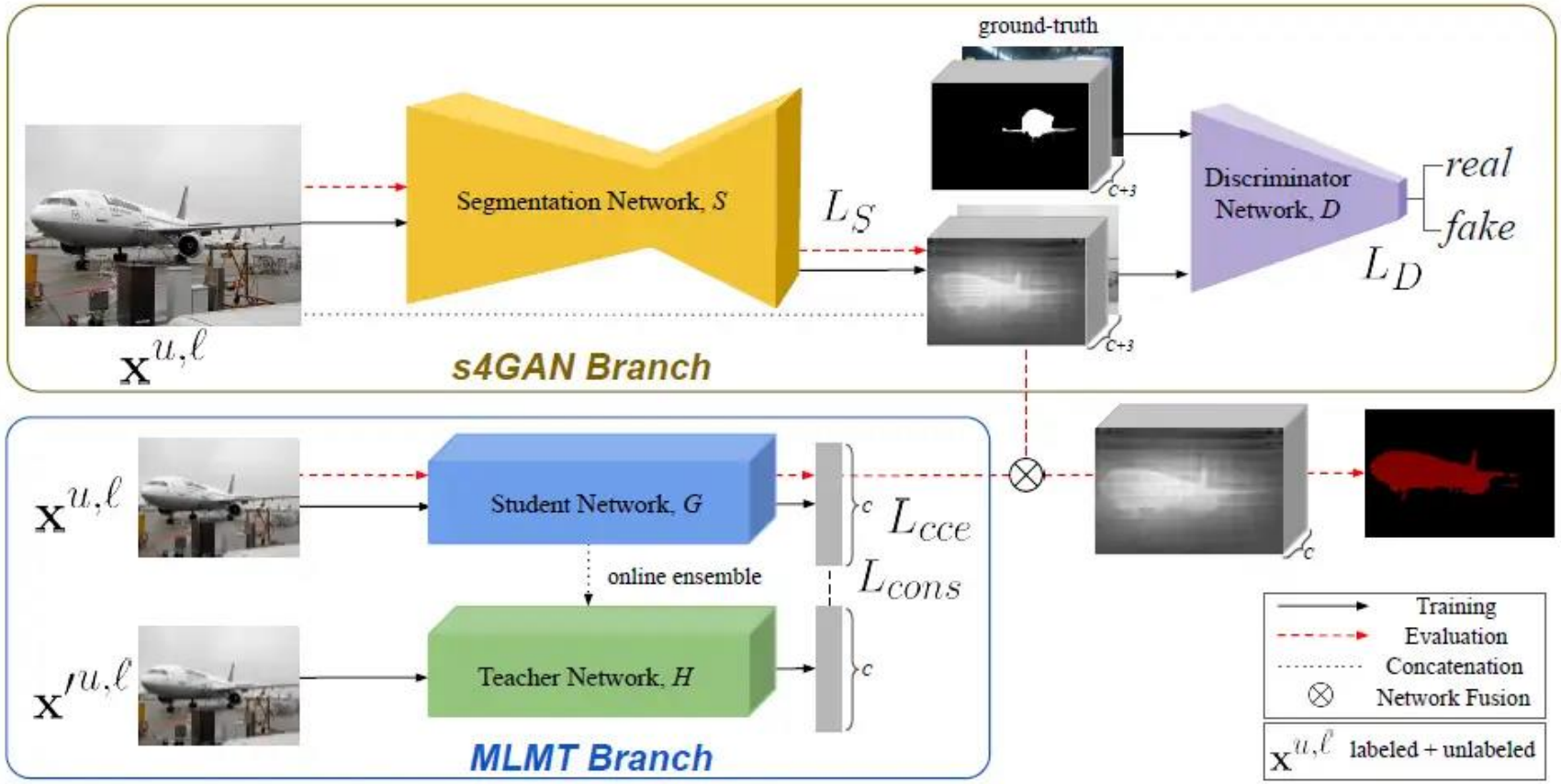
- It enables real-time data processing and decision-making, making it particularly valuable for applications requiring immediate feedback or operation in environments with limited connectivity.



2. Related Works

Semi-supervised Sementic Segmentation

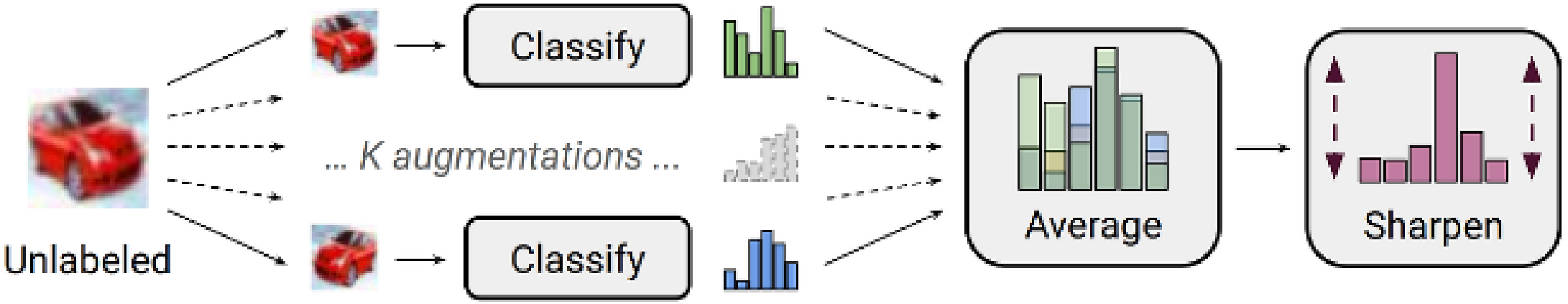
- GAN Methods



2. Related Works

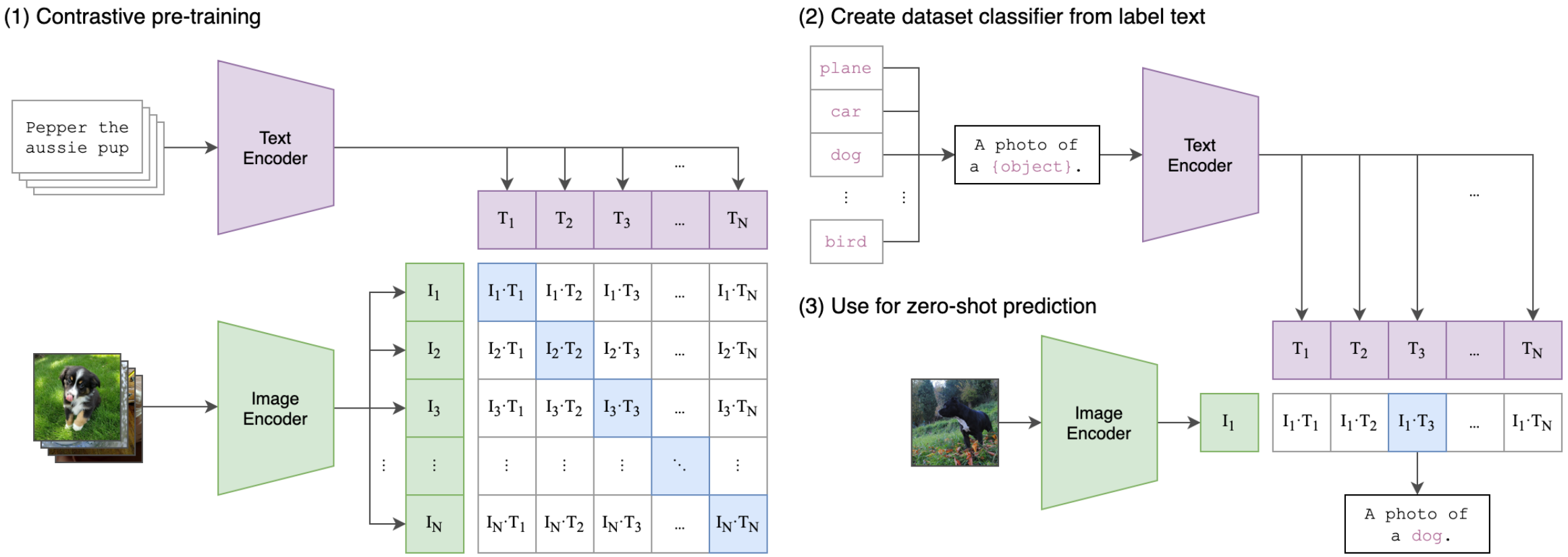
Semi-supervised Sementic Segmentation

- Consistency Regularization



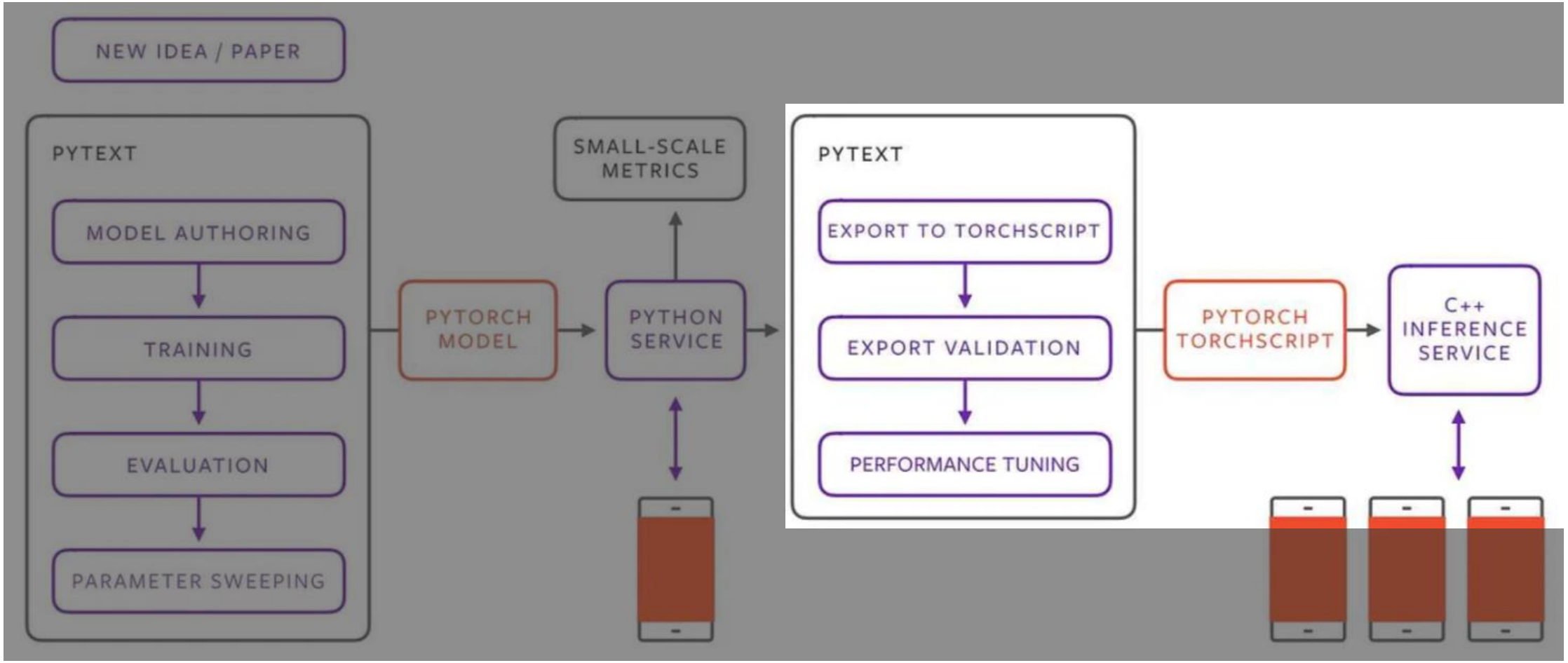
2. Related Works

CLIP (Contrastive Language-Image Pre-Training)



2. Related Works

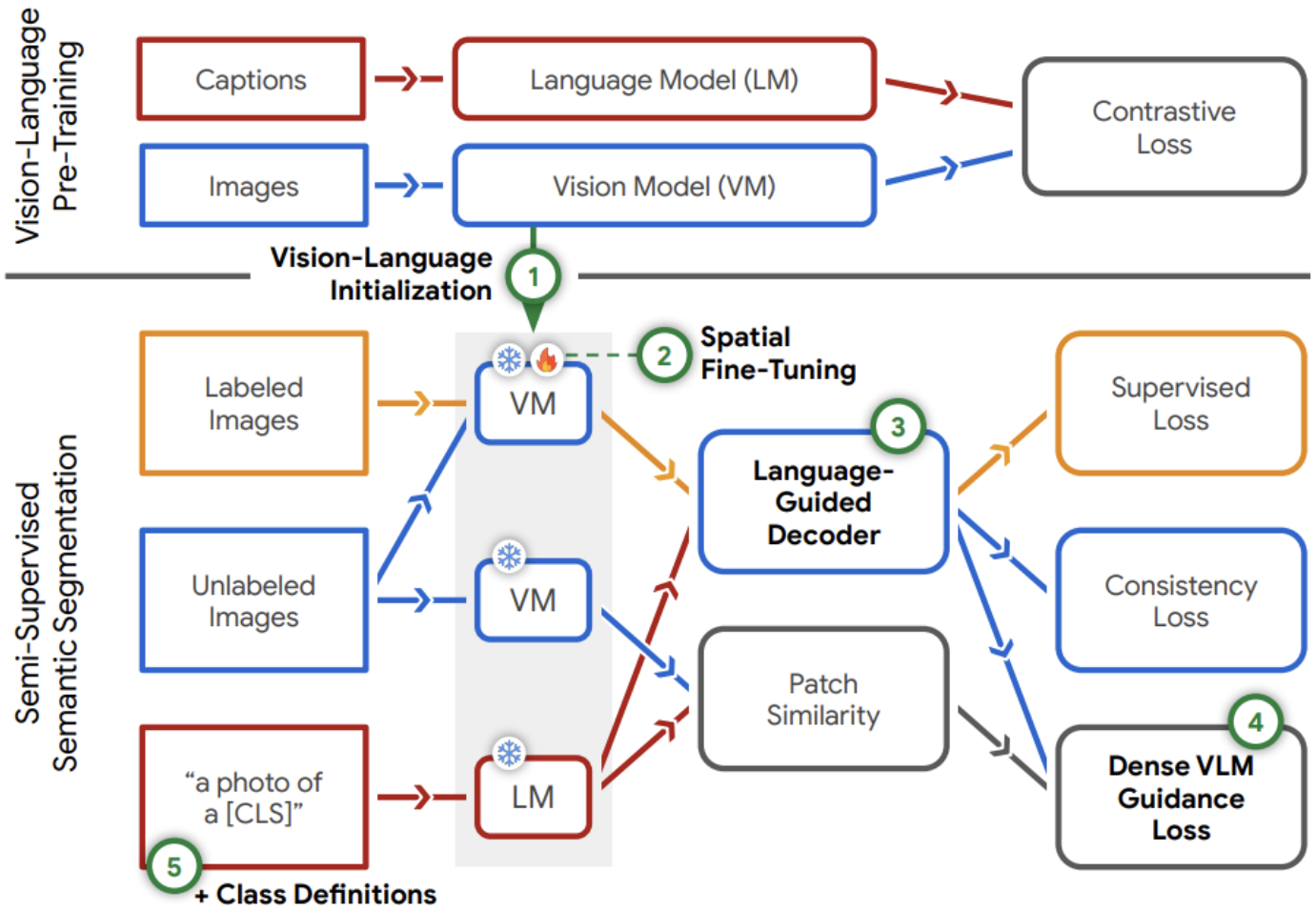
TorchScript



3. Framework

SemiVL

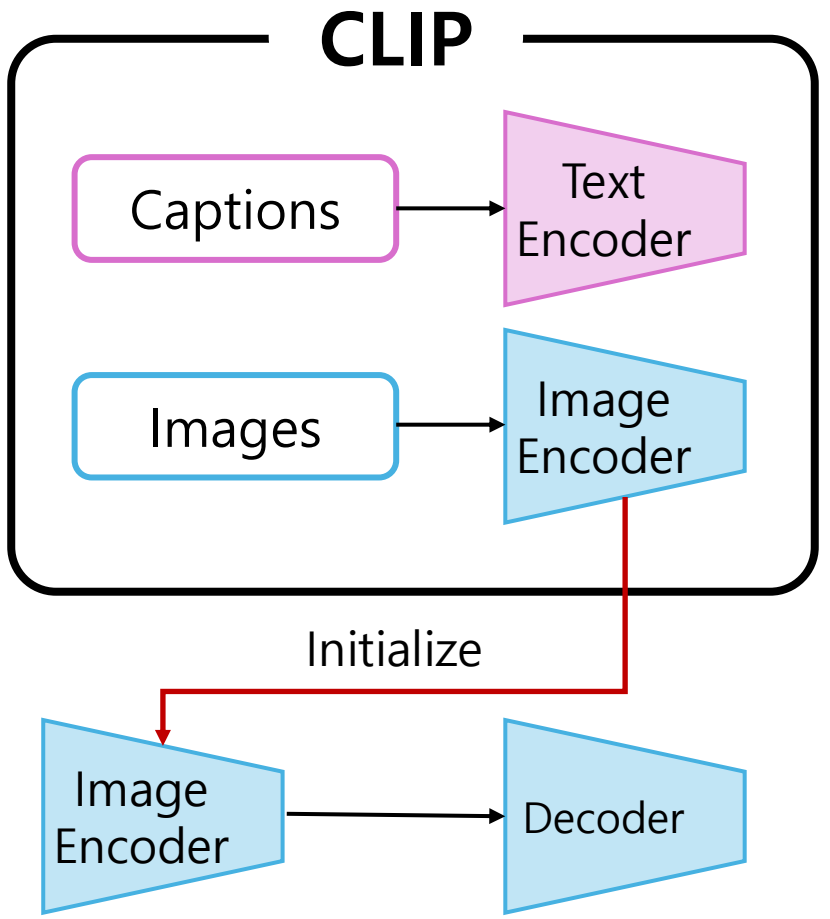
1. Vision-Language Pre-training
2. Spatial Fine-Tuning
3. Language-Guided Decoder
4. Dense CLIP Guidance
5. Class Definitions



4. Methods

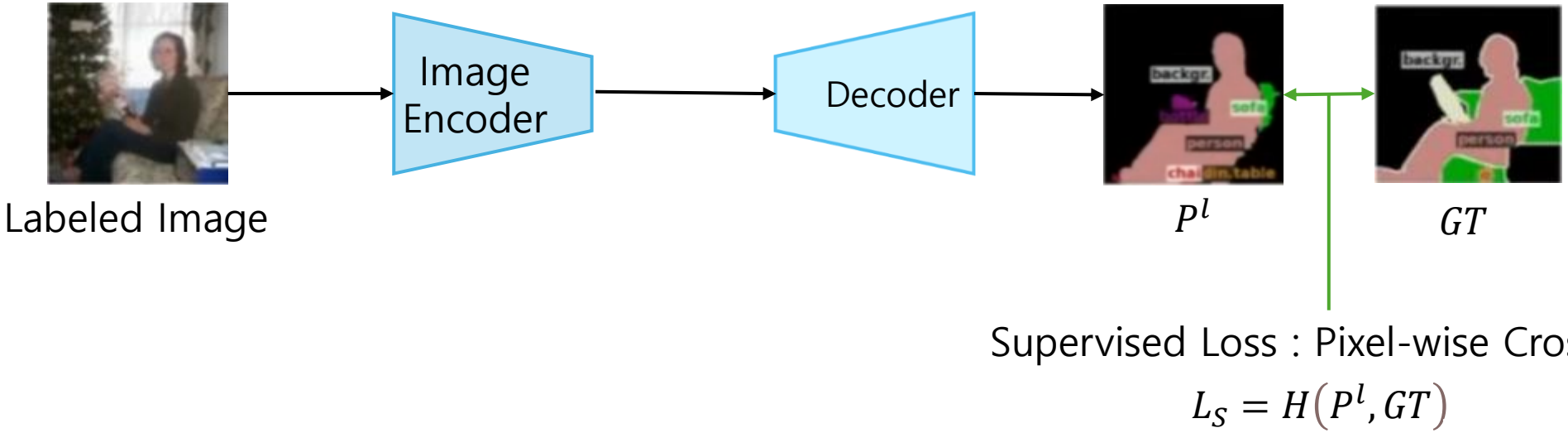
Vision-Language Pre-training

- CLIP is trained on web-scale image-text datasets that cover almost all semantic classes a vision agent can ever come across.
- VLM pre-training does not require a manually annotated dataset but can be trained on web-crawled image-caption pairs.
- Initializing the semantic segmentation encoder with the pre-trained VLM vision encoder to utilize its rich semantic prior.



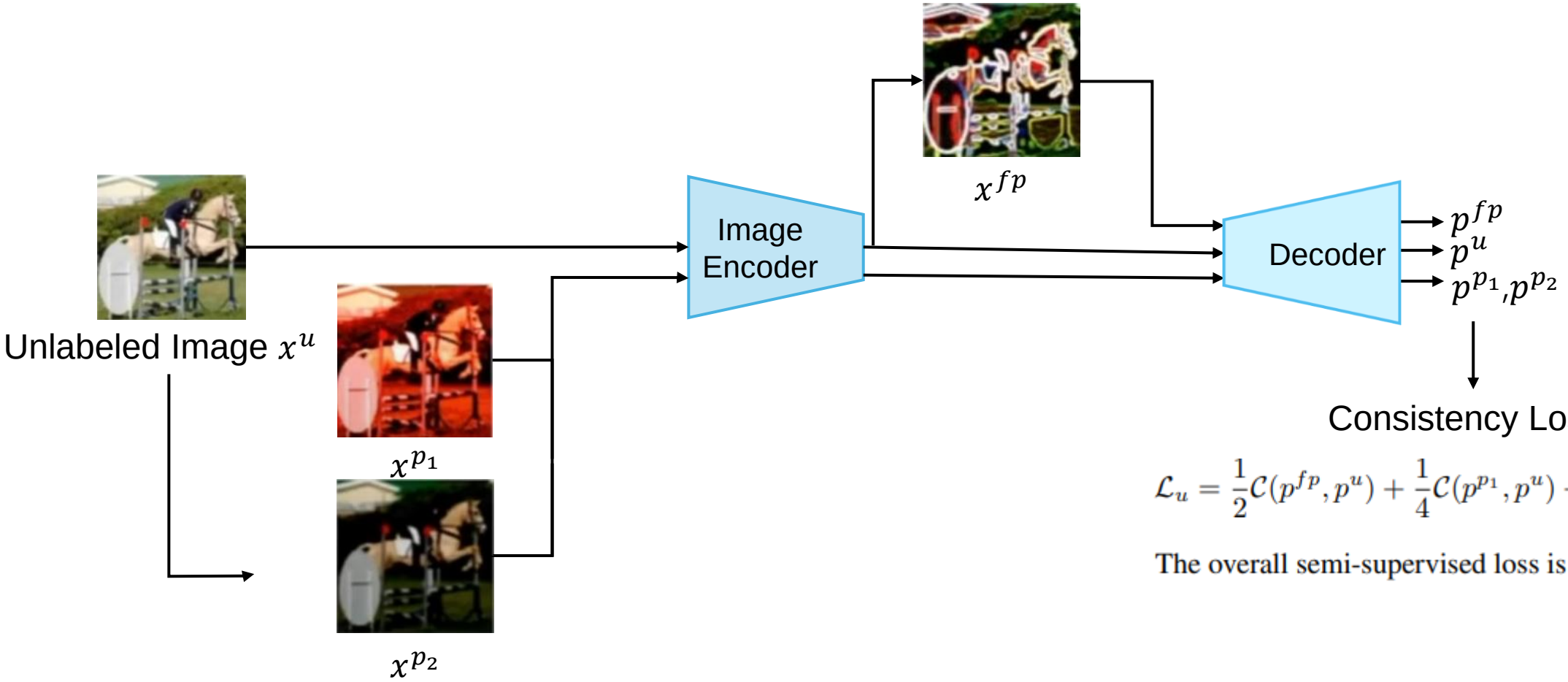
Consistency Training

Semantic segmentation models are usually trained with a supervised loss L_S
such as the pixel-wise cross entropy which is only possible on the labeled images



4. Methods

Consistency Training



$$\mathcal{L}_u = \frac{1}{2}\mathcal{C}(p^{fp}, p^u) + \frac{1}{4}\mathcal{C}(p^{p_1}, p^u) + \frac{1}{4}\mathcal{C}(p^{p_2}, p^u)\,.$$

The overall semi-supervised loss is $\mathcal{L} = \frac{1}{2}(\mathcal{L}_s + \mathcal{L}_u)$.

4. Methods

Consistency Training

Perturbations can achieved by perturbing the features of the model
$p^{fp} = h\left(P\left(g\left(x^{w}\right)\right)\right)$, where g is the encoder and h the decoder of the model f.

```
if is_vlm(self):
    feats = x[0][0]
    x[0][0] = [torch.cat((f, F.dropout2d(f, self.fp_rate))) for f in feats]
    x[0][1] = torch.cat((x[0][1], x[0][1]))
    # perturb features from conv_encoder
    if len(x) == 3 and x[2] is not None:
        x[2] = [torch.cat((f, F.dropout2d(f, self.fp_rate))) for f in x[2]]
    # also provide unperturbed features
    if hasattr(self.decode_head, 'dc_unperturbed') and self.decode_head.dc_unperturbed:
        assert len(x[0]) == 2
        x[0].append([torch.cat((f, f)) for f in feats])
```

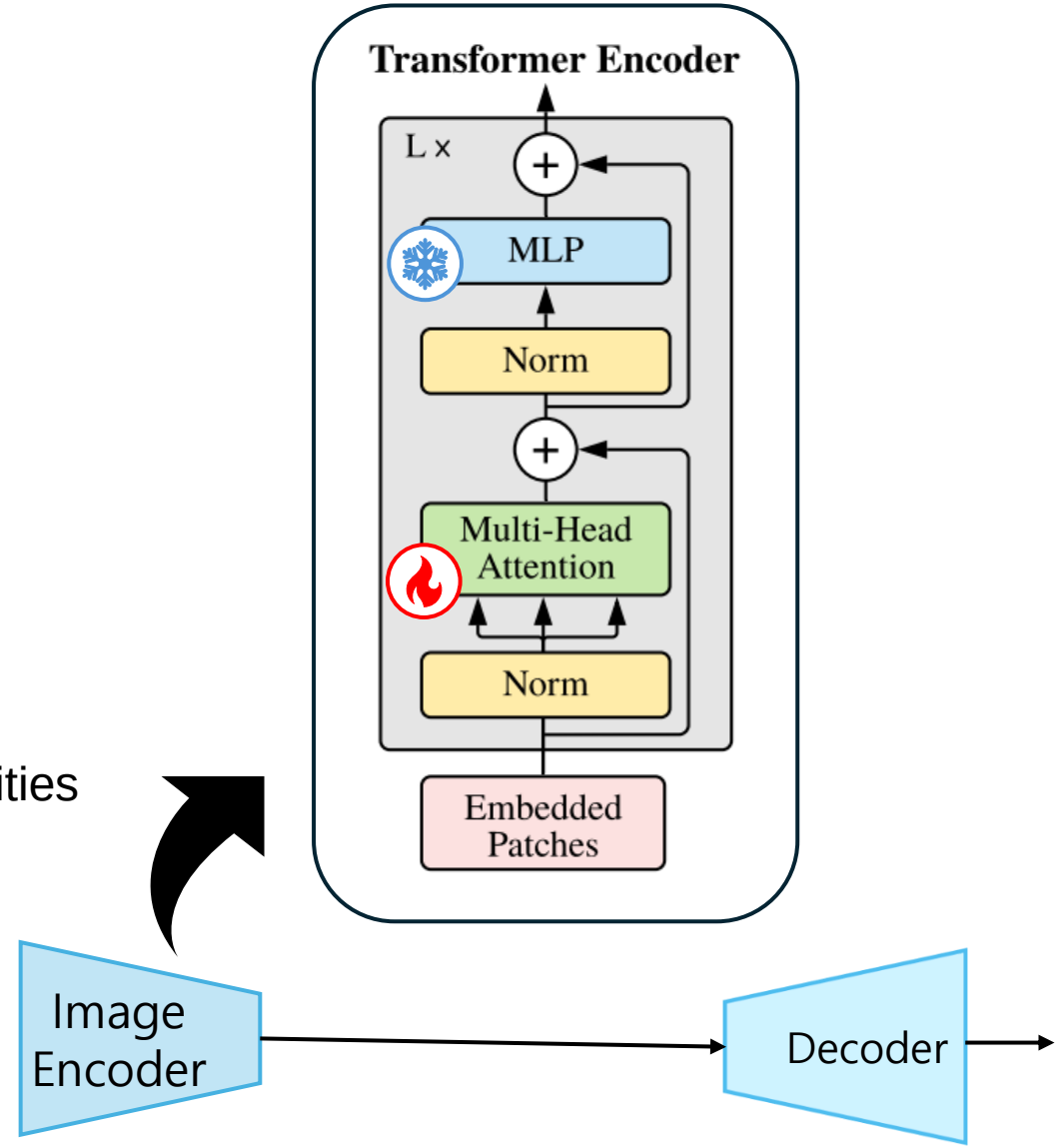
4. Methods

Spatial Fine-Tuning

Spatial fine-tuning only fine-tunes the attention layers, which are responsible for spatial reasoning

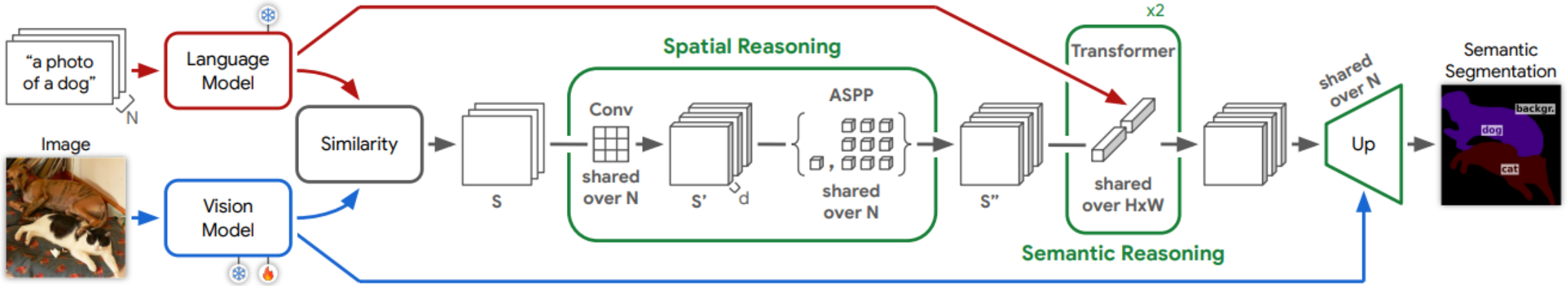
The alignment of semantic features and their corresponding image content can be refined for dense predictions

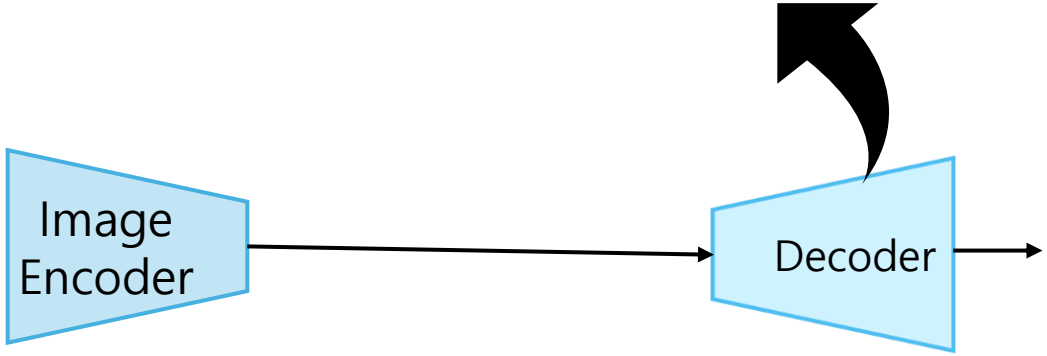
MLP layers are frozen as they do not perform spatial reasoning to preserve the semantic reasoning capabilities



4. Methods

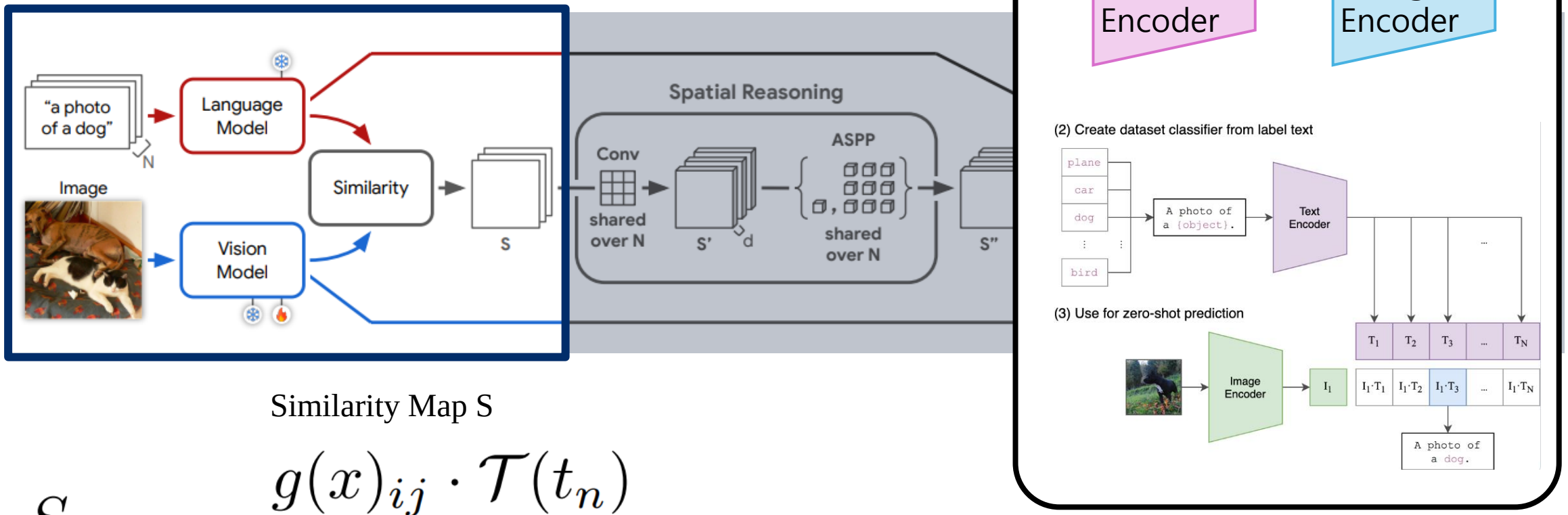
Language-Guided Decoder





4. Methods

Language-Guided Decoder

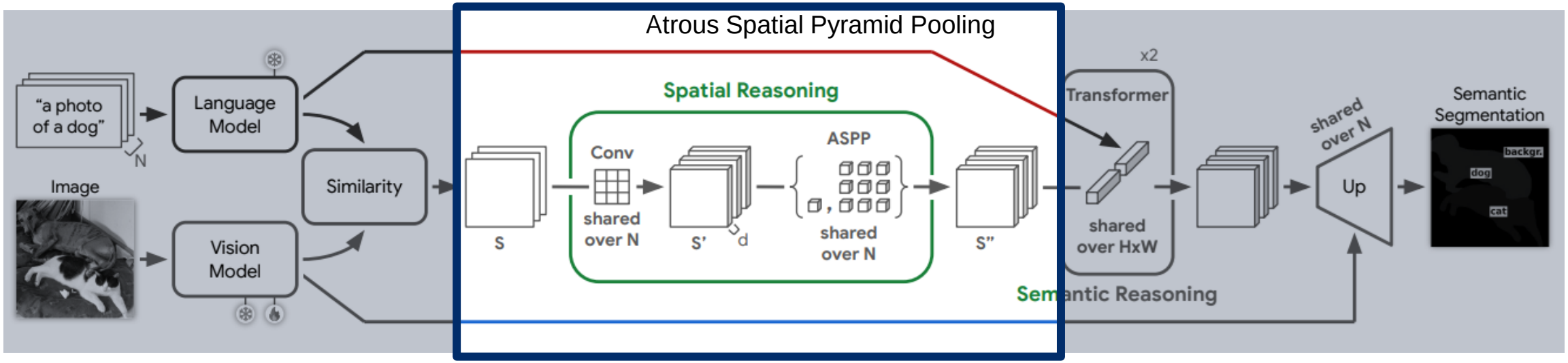


Similarity Map S

$$S_{ijn} = \frac{g(x)_{ij} \cdot \mathcal{T}(t_n)}{\|g(x)_{ij}\| \|\mathcal{T}(t_n)\|} .$$

4. Methods

Language-Guided Decoder



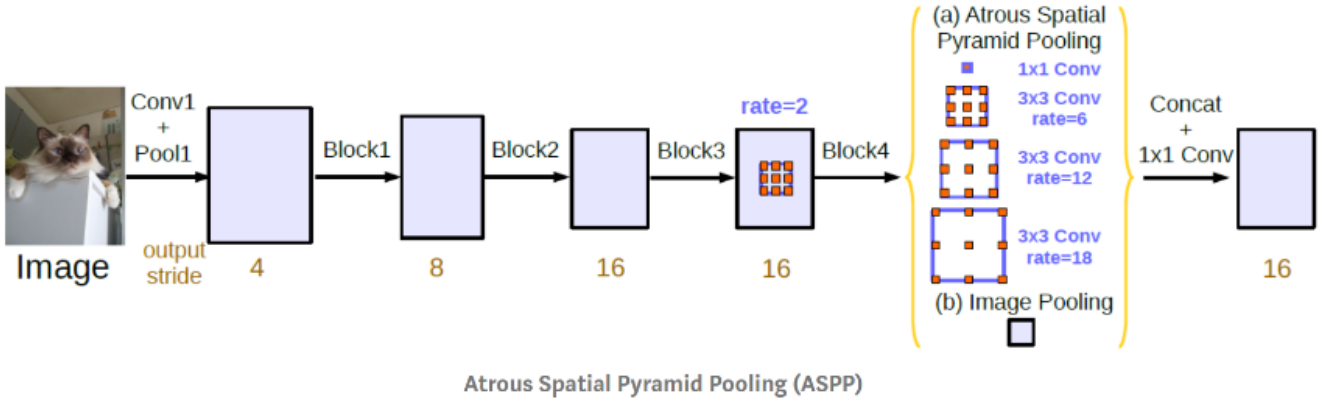
- Spatial reasoning module poerates on each Class similarity map independently and Models no inter-class relations

- Each S_n is processed by a 7x7 convolution to learn local spatial structures and embed them to similarity volumes S'_n of d dimensions

Language-Guided Decoder

ASPP processes the obtained similarity volumes to model long-range context relations, resulting in a combined similarity volume for all classes S″

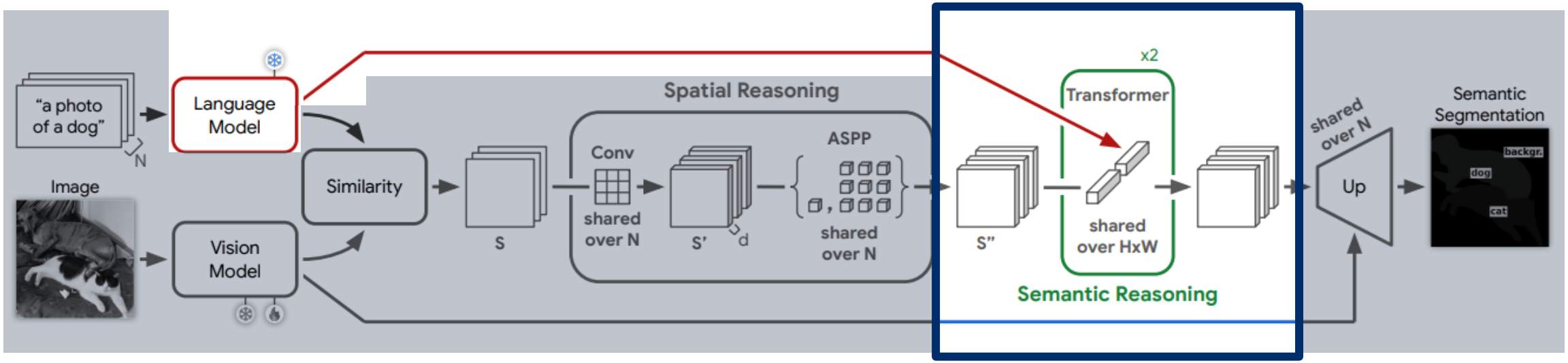
```python
class ASPPModule(nn.Module):
    def __init__(self, in_channels, atrous_rates=(1, 6, 12, 18), out_channels=None):
        super(ASPPModule, self).__init__()
        if out_channels is None:
            out_channels = in_channels

        self.aspp_convs = nn.ModuleList()
        for dilation in atrous_rates:
            ksize = 1 if dilation == 1 else 3
            padding = 0 if dilation == 1 else dilation
            self.aspp_convs.append(
                nn.Sequential(nn.Conv2d(in_channels, out_channels, ksize, padding=padding,
                                        dilation=dilation, bias=False),
                              nn.GroupNorm(out_channels // 16, out_channels),
                              nn.ReLU(True))
            )
        self.aspp_convs.append(ASPPPooling(in_channels, out_channels))

        self.project = nn.Sequential(nn.Conv2d(5 * out_channels, out_channels, 1, bias=False),
                                     nn.GroupNorm(out_channels // 16, out_channels),
                                     nn.ReLU(True))

    def forward(self, x):
        feats = []
        for c in self.aspp_convs:
            feats.append(c(x))
        y = torch.cat(feats, 1)
        y = self.project(y)
        y = x + y
        return y
```



Atrous Spatial Pyramid Pooling (ASPP)

4. Methods

Language-Guided Decoder



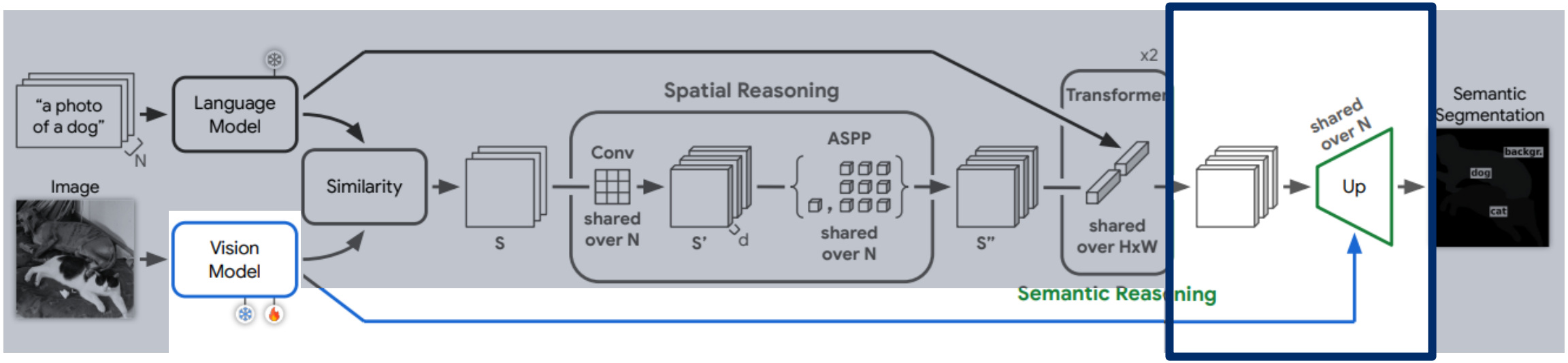
The semantic reasoning module models the relationship between classes

By decoupling spatial and semantic reasoning, the learned weights can be shared over different classes for spatial reasoning and shared over different locations for semantic reasoning

The limited annotations can be utilized more effectively and overfitting is reduced

4. Methods

Language-Guided Decoder



The common vision transformers operate on a 16 times smaller feature resolution than the input, and limit precise segmentation boundaries

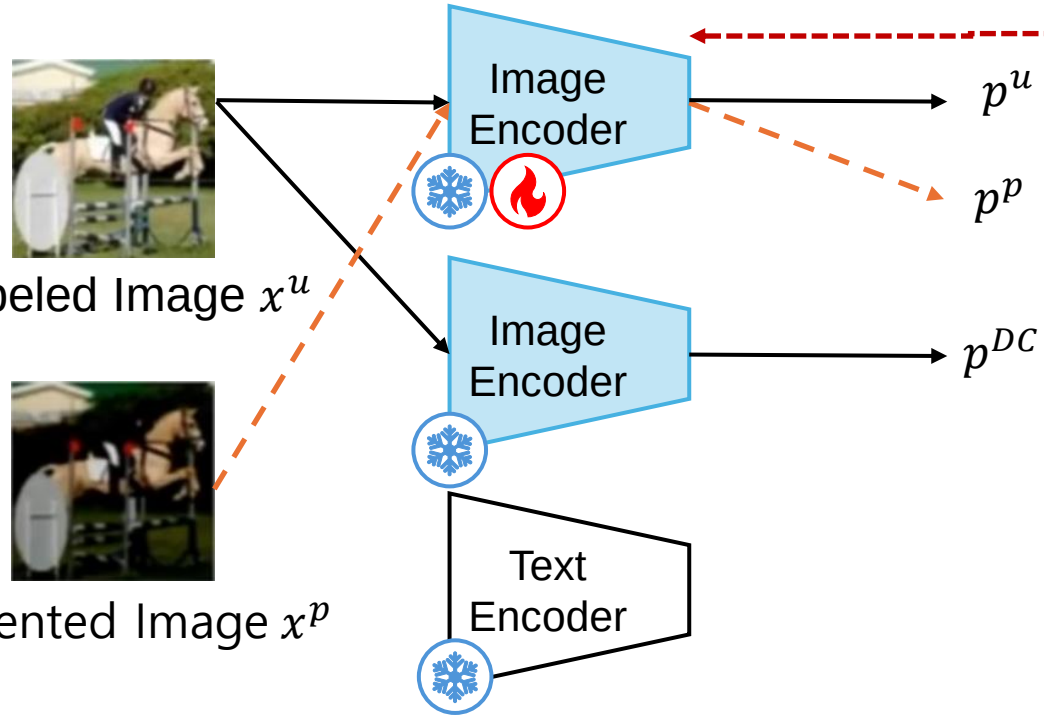
We add 2 upsampling blocks, which learn the upsampling using a transpose convolution.

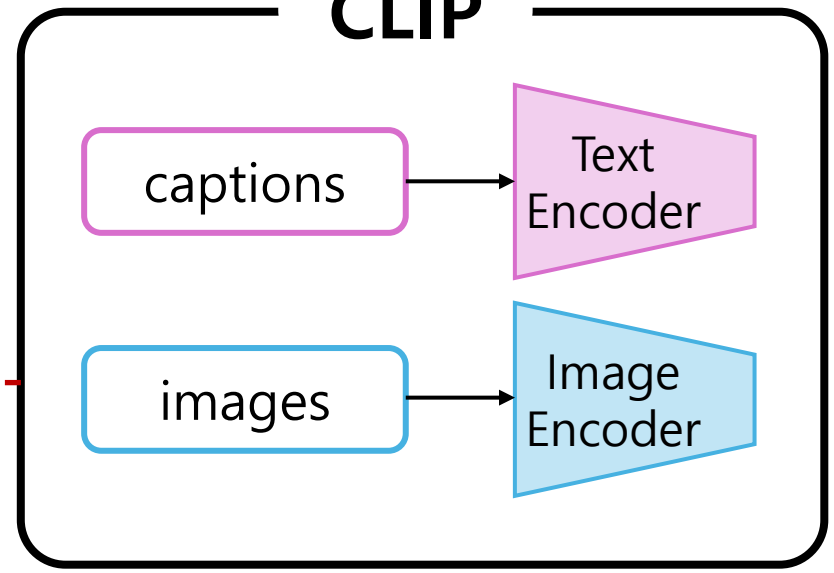
A final convolution maps the d dimensions to one channel to obtain the logits for a class.

4. Methods

Dense CLIP Guidance

Self-training on unlabeled images in vision-language pre-training for semantic segmentation can lead to a drift towards erroneous predictions and self-confirmation bias.



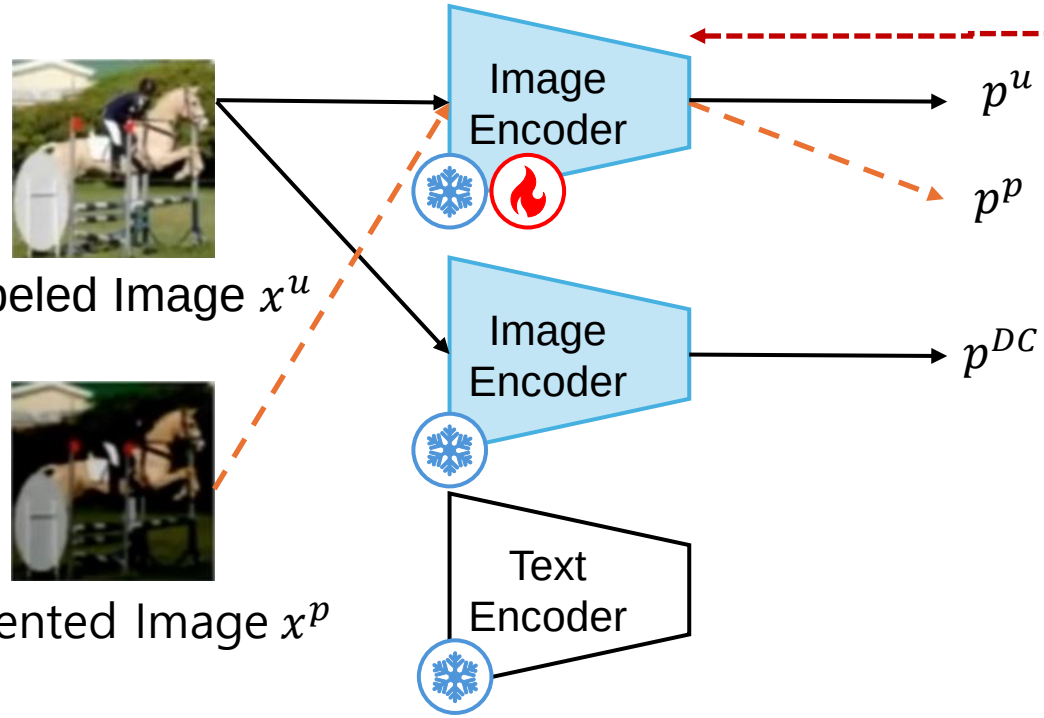


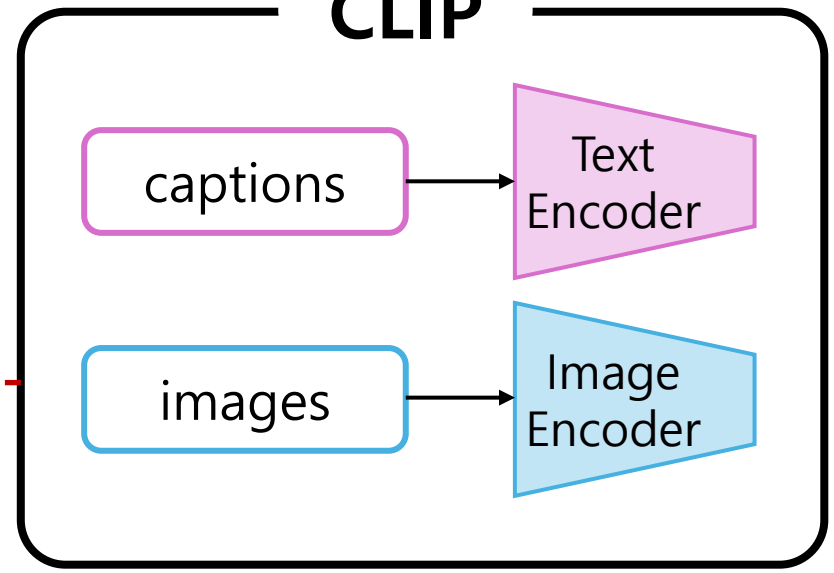
$$\mathcal{C}(p^p, p^u, p^{DC}) = \mathcal{C}(p^p, p^u) + \lambda_{DC} \sum \mathbb{1}[\max(p^{DC}) \geq \zeta] H(p^p, p^{DC})$$

4. Methods

Dense CLIP Guidance

To anchor the self-training on the unlabeled images and reduce this issue, we guide the consistency training with predictions from a frozen auxiliary CLIP model, which cannot drift.





$$\mathcal{C}(p^p, p^u, p^{DC}) = \mathcal{C}(p^p, p^u) + \lambda_{DC} \sum \mathbb{1}[\max(p^{DC}) \geq \zeta] H(p^p, p^{DC})$$

Class Definition Guidance

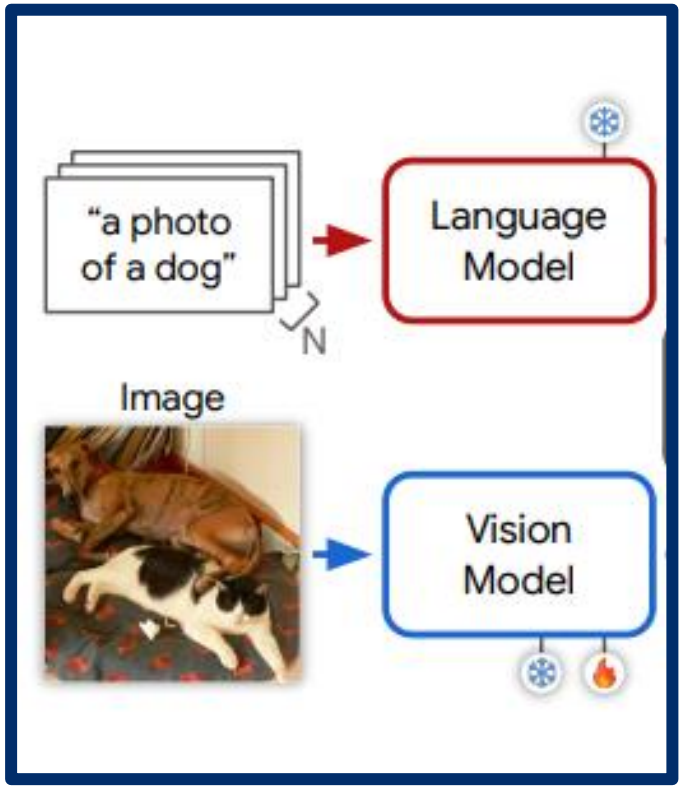
Table S4. Class definitions from **annotation guidelines (raw text)** on Pascal VOC.

Class c	Annotation Guidelines
background	"background"
aeroplane	"aeroplane including gliders but not hang gliders or helicopters"
bicycle	"bicycle including tricycles, unicycles"
bird	"bird"
boat	"boat including ships, rowing boats, pedaloes but not jet skis"
bottle	"bottle including plastic, glass or feeding bottles"
bus	"bus including minibus but not trams"
car	"car including vans, large family cars for 6-8 people, toy cars but not go-carts, tractors, emergency vehicles, lorries, trucks, or the vehicle interior"
cat	"domestic cat"
chair	"chair including armchairs, deckchairs, but not stools, wheelchairs, seats in buses or cars"
cow	"cow"
dining table	"table for eating at but not coffee tables, desks, side tables or picnic benches"
dog	"domestic dog (not wolves etc.)"
horse	"horse including ponies, donkeys, mules etc."
motorbike	"motorbike including mopeds, scooters, sidecars"
person	"person including babies, faces (i.e. truncated people)"
potted plant	"indoor plants or outdoor plants clearly in a pot but not flowers in vases"
sheep	"sheep but not a goat"
sofa	"sofa excluding sofas made up as sofa-beds"
train	"train including train carriages, excluding trams"
tv/monitor	"tv/monitor including standalone screens but not laptops nor advertising displays"

Table S5. Class definitions as **concepts from GPT** on Pascal VOC.

Class c	GPT Concepts
background	"background", "scene", "environment", "setting", "context"
aeroplane	"aeroplane", "aircraft", "plane", "jet", "aviation"
bicycle	"bicycle", "bike", "cycle", "pedal", "two-wheeler"
bird	"bird", "avian", "feathered creature", "fowl", "winged animal"
boat	"boat", "vessel", "watercraft", "ship", "sailboat"
bottle	"bottle", "flask", "container", "jar", "vial"
bus	"bus", "coach", "transit", "shuttle", "public transport"
car	"car", "automobile", "vehicle", "motorcar", "sedan"
cat	"cat", "feline", "kitty", "kitten", "pussycat"
chair	"chair", "seat", "furniture", "stool", "armchair"
cow	"cow", "bovine", "cattle", "ox", "livestock"
diningtable	"diningtable", "table", "dining furniture", "dinner table", "kitchen table"
dog	"dog", "canine", "pooch", "puppy", "man's best friend"
horse	"horse", "equine", "stallion", "pony", "mare"
motorbike	"motorbike", "motorcycle", "bike", "motor", "two-wheeled vehicle"
person	"person", "human", "individual", "human being", "someone"
pottedplant	"pottedplant", "pot plant", "houseplant", "potted flower", "indoor plant"
sheep	"sheep", "lamb", "ewe", "ram", "woolly animal"
sofa	"sofa", "couch", "settee", "divan", "lounge"
train	"train", "locomotive", "railway vehicle", "railroad train", "engine"
tv/monitor	"tv/monitor", "television", "screen", "display", "monitor"

Class c	Oxford Languages Definition
background	"background"
aeroplane	"aeroplane", "a flying vehicle with fixed wings"
bicycle	"bicycle", "a vehicle consisting of two wheels held in a frame one behind the other, propelled by pedals and steered with handlebars attached to the front wheel"
bird	"bird", "a warm-blooded egg-laying vertebrate animal distinguished by the possession of feathers, wings, a beak, and typically by being able to fly"
boat	"boat", "a vessel for travelling over water, propelled by oars, sails, or an engine"
bottle	"bottle", "a glass or plastic container with a narrow neck, used for storing drinks or other liquids"
bus	"bus", "a large motor vehicle carrying passengers by road"
car	"car", "a four-wheeled road vehicle that is powered by an engine and is able to carry a small number of people"
cat	"cat", "a small domesticated carnivorous mammal with soft fur, a short snout, and retractable claws"
chair	"chair", "a separate seat for one person, typically with a back and four legs"
cow	"cow", "a fully grown female animal of a domesticated breed of ox, kept to produce milk or beef"
dining table	"dining table", "a table on which meals are served in a dining room"
dog	"dog", "a domesticated carnivorous mammal that typically has a long snout and non-retractable claws"
horse	"horse", "a large plant-eating domesticated mammal with solid hoofs and a flowing mane and tail, used for riding, racing, and to carry and pull loads"
motorbike	"motorbike", "a two-wheeled vehicle that is powered by a motor and has no pedals"
person	"person", "a human being regarded as an individual"
potted plant	"potted plant", "a plant in a pot"
sheep	"sheep", "a domesticated ruminant mammal with a thick woolly coat"
sofa	"sofa", "a long upholstered seat with a back and arms, for two or more people"
train	"train", "a series of connected railway carriages or wagons moved by a locomotive or by integral motors"
tv/monitor	"tv/monitor", "a device for watching television"

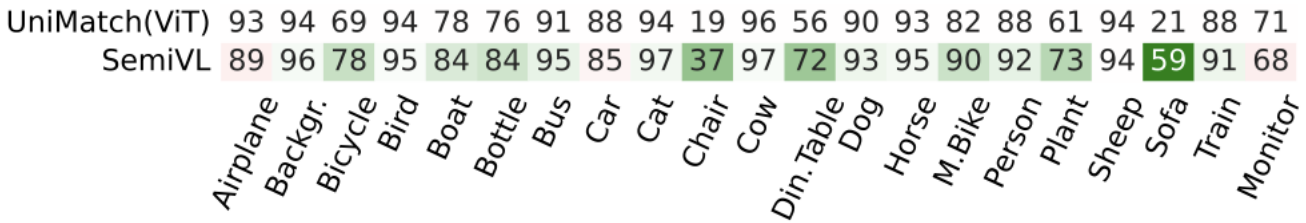


5. Experiments

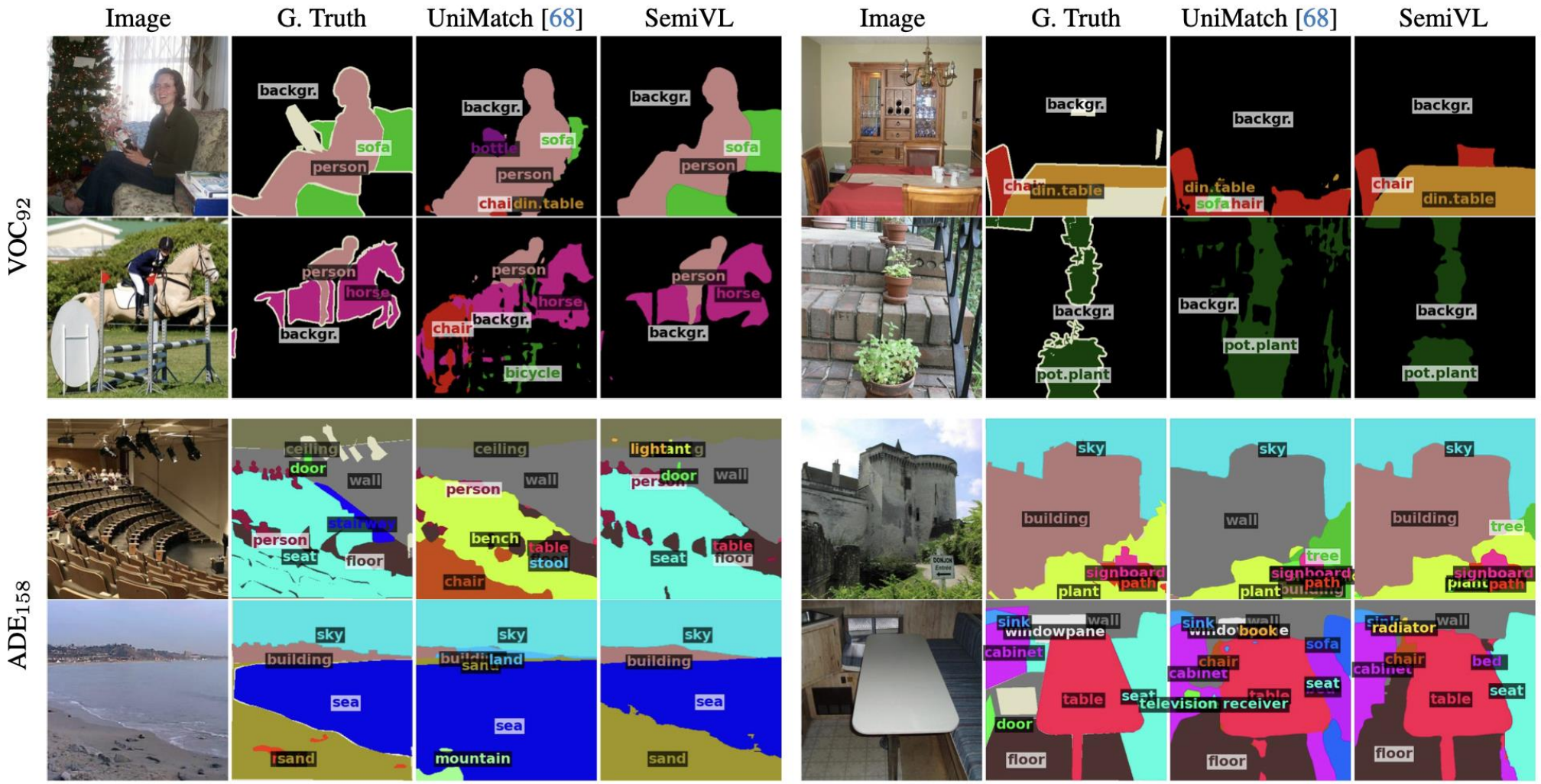
Table 1. State-of-the-art comparison on **Pascal VOC**. The mIoU (%) is compared across different splits for the labeled subset $\mathcal{D}^l$. † denotes re-produced results in the same setting as SemiVL.

Method		Net	1/115 (92)	1/58 (183)	1/29 (366)	1/14 (732)	1/7 (1464)
PseudoSeg [77]	[ICLR'21]	R101	57.6	65.5	69.1	72.4	–
CPS [7]	[CVPR'21]	R101	64.1	67.4	71.7	75.9	–
ST++ [67]	[CVPR'22]	R101	65.2	71.0	74.6	77.3	79.1
U^2PL [60]	[CVPR'22]	R101	68.0	69.2	73.7	76.2	79.5
PCR [62]	[NeurIPS'22]	R101	70.1	74.7	77.2	78.5	80.7
ESL [43]	[ICCV'23]	R101	71.0	74.0	78.1	79.5	81.8
LogicDiag [38]	[ICCV'23]	R101	73.3	76.7	77.9	79.4	–
UniMatch [68]	[CVPR'23]	R101	75.2	77.2	78.8	79.9	81.2
3-CPS [37]	[ICCV'23]	R101	75.7	77.7	80.1	80.9	82.0
ZegCLIP† [76]	[CVPR'23]	ViT-B/16	69.3	74.2	78.7	81.0	82.0
ZegCLIP+UniMatch†	—	ViT-B/16	78.0	80.3	80.9	82.8	83.6
UniMatch† [68]	[CVPR'23]	ViT-B/16	77.9	80.1	82.0	83.3	84.0
Ours	—	ViT-B/16	**84.0** (+6.1)	**85.6** (+5.5)	**86.0** (+4.0)	**86.7** (+3.4)	**87.3** (+3.3)

Class-Wise IoU on Pascal VOC with 92 Labels



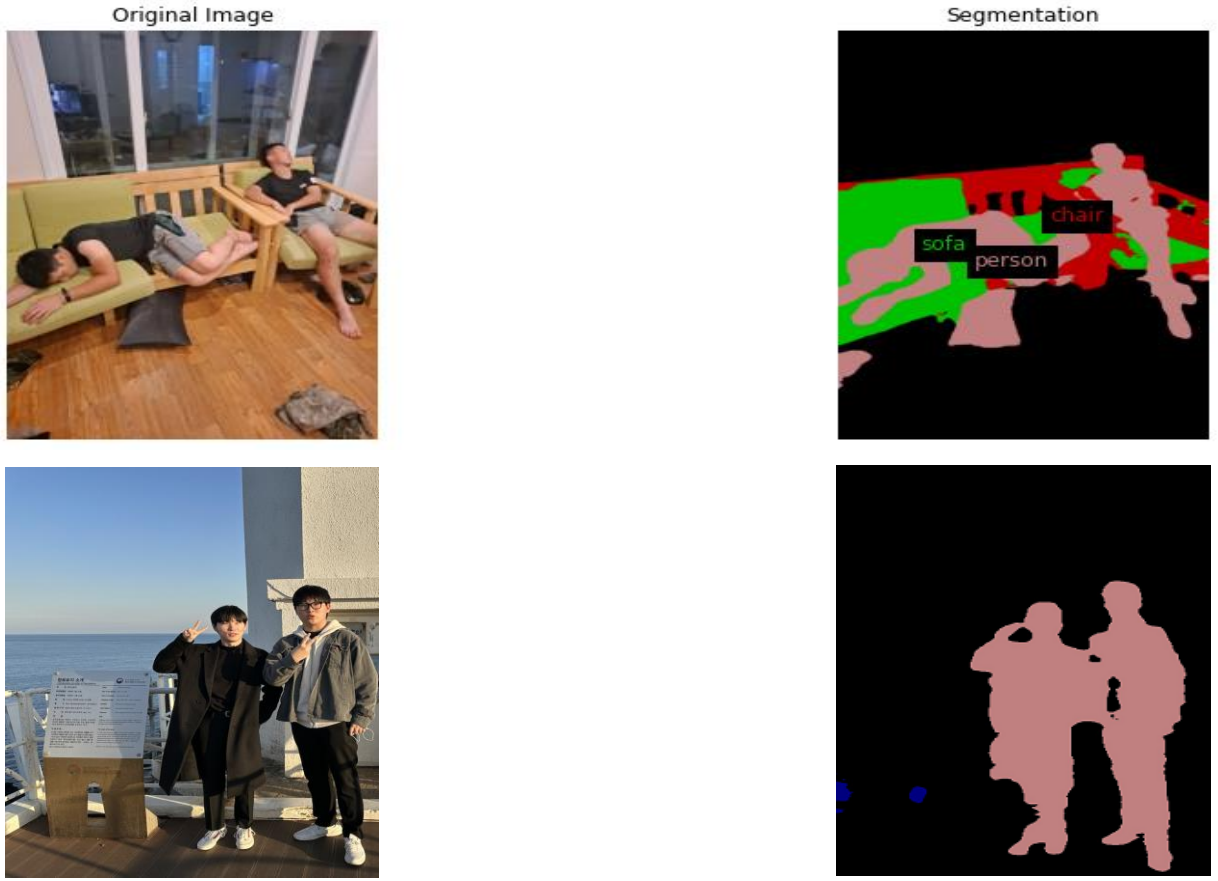
5. Experiments



5. Experiments

Performance Evaluation of TorchScript Model

- After removing the dependency on the GPU, the model converted to TorchScript was used for inference on the VOC dataset in a PC CPU environment.
- The results showed an mIoU of 72.24% with an average inference time of 6438.99 milliseconds. Although this performance is lower than the original model, it still maintains high accuracy, and the increase in inference time is not significant
- Additionally, it demonstrates excellent performance in zero-shot inference on images not used during training.



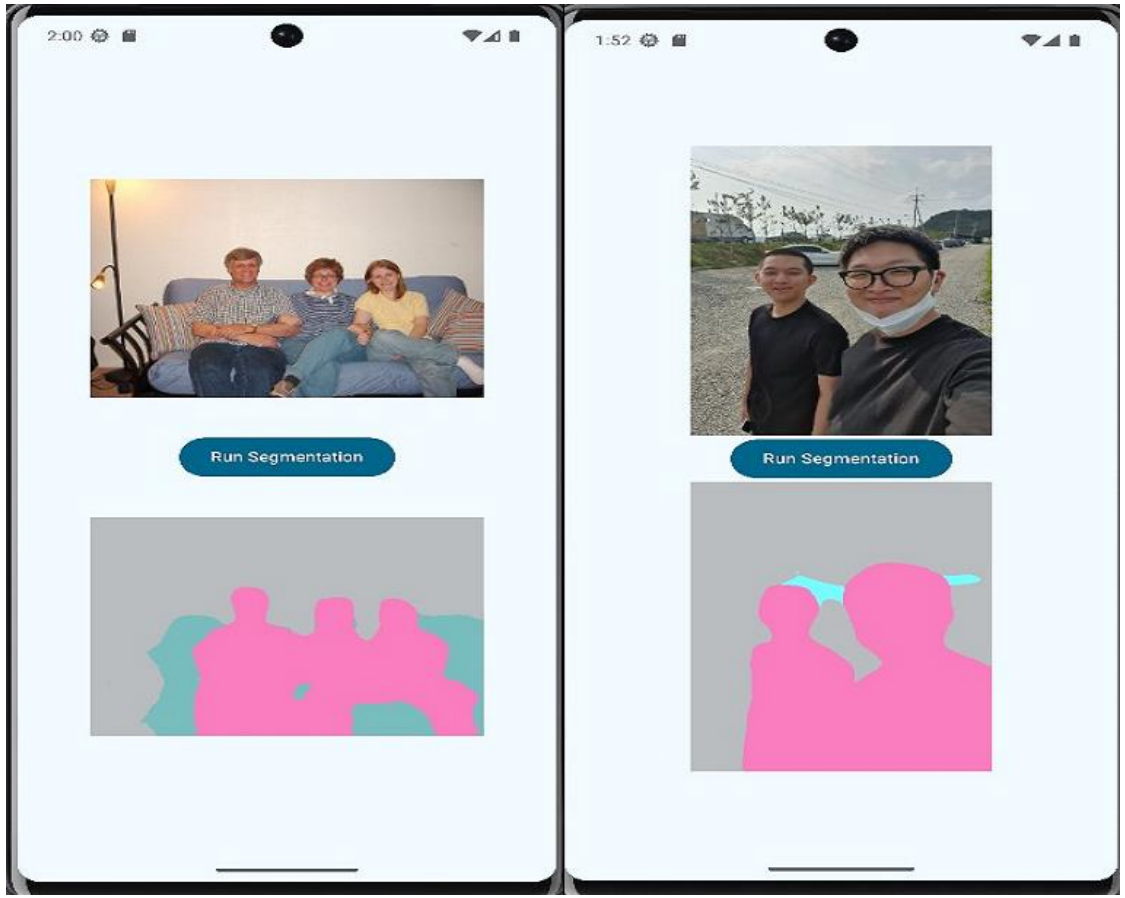
5. Experiments

Performance Evaluation of TorchScript Model

- The Google Pixel 2 shows a 21.2% decrease in performance (mIoU) and an increase of 423.91 milliseconds in inference time compared to a desktop PC (GPU).

- This level of performance degradation is considered acceptable for use as on-device AI in embedded systems.

구분		mIoU(%)	추론시간 (msec)
PyTorch	데스크톱 PC (GPU)	84.0	6234.54
TorchScript	데스크톱 PC (CPU)	77.5	6438.99
	Google Pixel 2	62.8	6658.45



5. Experiments

Performance Evaluation of TorchScript Model (Zero-shot)