CLOUDS : Collaborating Foundation models for Domain Generalized Semantic Segmentation

Yasser Benigmim [1, 2], Subhankar Roy [3], Slim Essid [1], Vicky Kalogeiton [2], Stéphane Lathuilière [1]

[1] LTCI, Télécom-Paris, Institut Polytechnique de Paris
[2] LIX, Ecole Polytechnique, CNRS, Institut Polytechnique de Paris
[3] University of Aberdeen

https://arxiv.org/abs/2312.09788
https://github.com/yasserben/CLOUDS

CVPR 2024

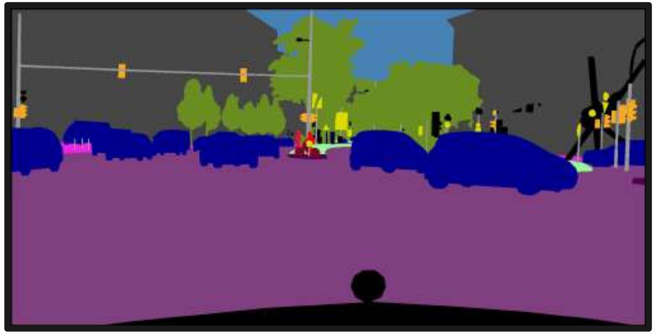
Task : Semantic Segmentation

- The objective of Semantic Segmentation is to assign a class for every pixel in the image

- A real-life HR image (2048x1024) contains ~ **$2x10^6$** pixels

- **It takes around ~90min to manually segment one image**

- Training on huge amounts of **labeled real-life data** for Semantic Segmentation is very expensive



Problem setting

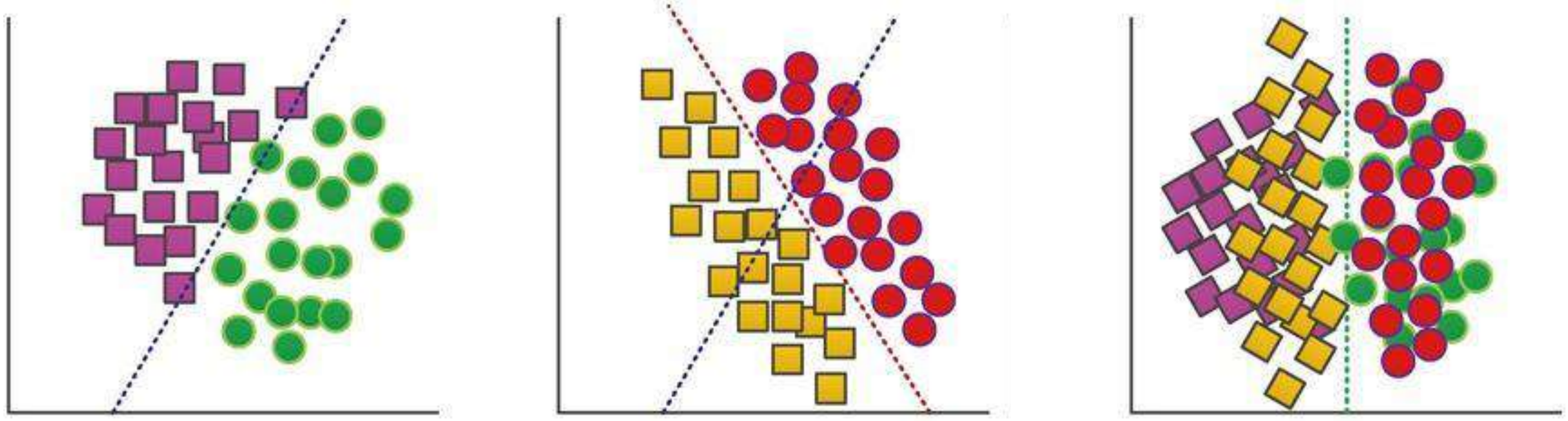
To alleviate the problem of annotations, multiple research axes have been proposed :

- Weakly-Supervised Learning, Semi-Supervised Learning …

- **Domain Adaptation :**
  - Train a model in a supervised way on a **source dataset easy to collect (synthetic dataset)**
  - Use the model at inference **on a target dataset (real-life dataset)**

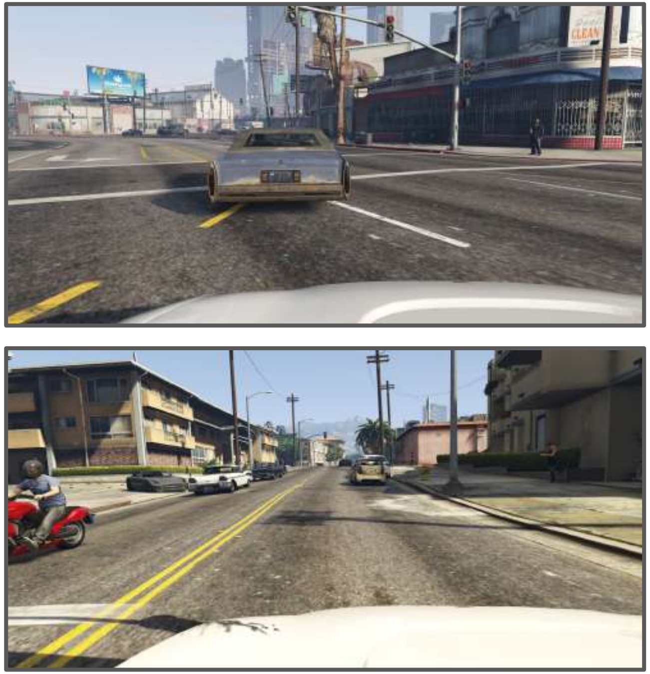
→ **Domain shift !**

Domain Adaptation : Toy example



(a) Source Domain (b) Target Domain (c) Domain Adaptation

Domain Adaptation : Semantic Segmentation



Domain shift

Synthetic data (GTA5)

real-life data (Cityscapes)

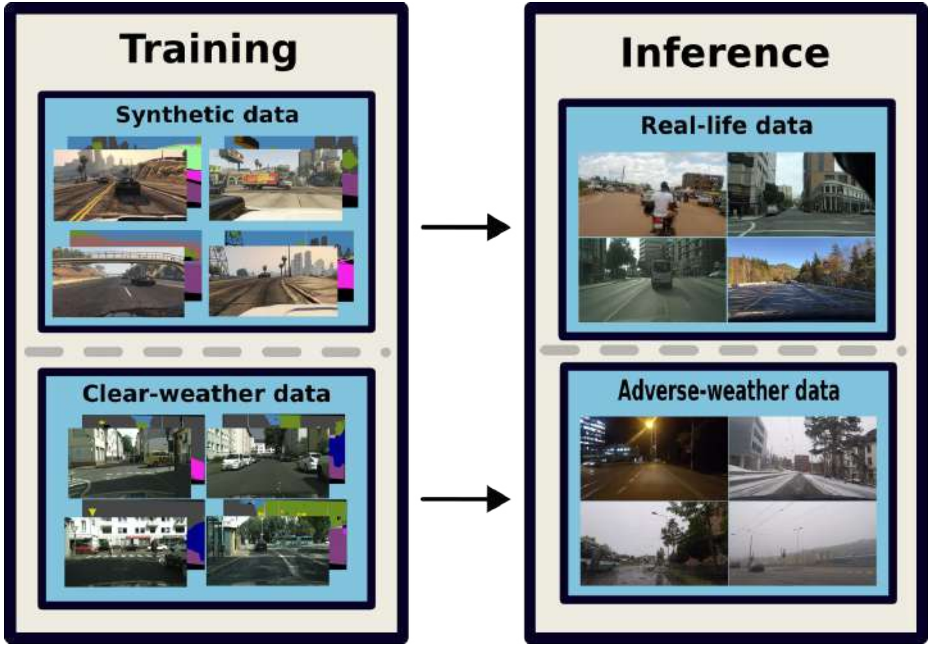
Image credits : S.Richter, et al. "Playing for Data: Ground Truth from Computer Games" ECCV 2016
M. Cordts, et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding” CVPR 2016

Unsupervised Domain Adaptation (UDA)

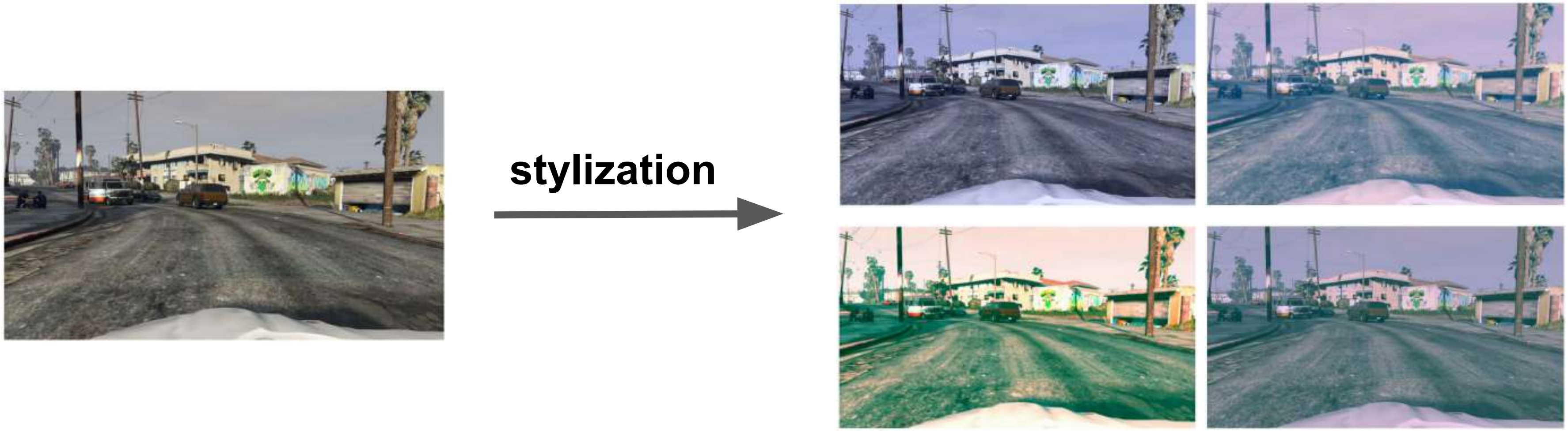
- in UDA, the model leverages **labeled source data** and **unlabeled target data** during training, then evaluates on unseen images from the target domain **target domain**

Domain Generalized Semantic Segmentation (DGSS)

- In DGSS, the model is trained on **labeled source data only** and tested on **unseen domains**



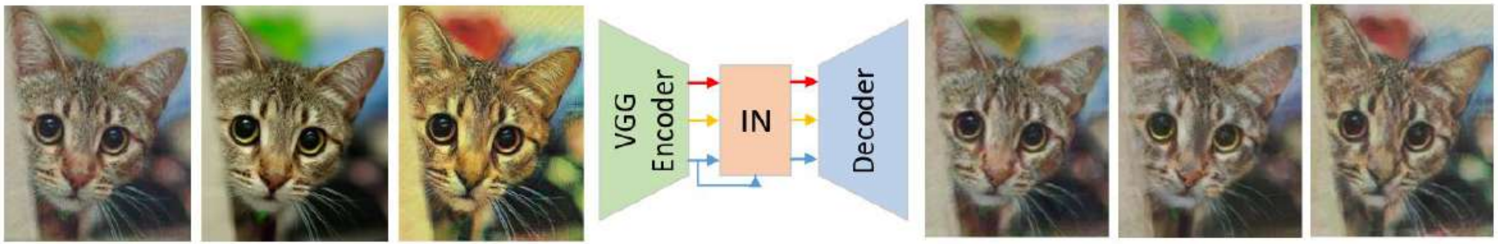
Previous works



Domain Randomization through style diversification only

1 : Z.Zhong, et al. "Adversarial Style Augmentation for Domain Generalized Urban-Scene Segmentation" NeurIPS 2022

Previous works



Tailor-made modules to eliminate **domain specific features**

1 : X.Pan, et al. "Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net" ECCV 2018

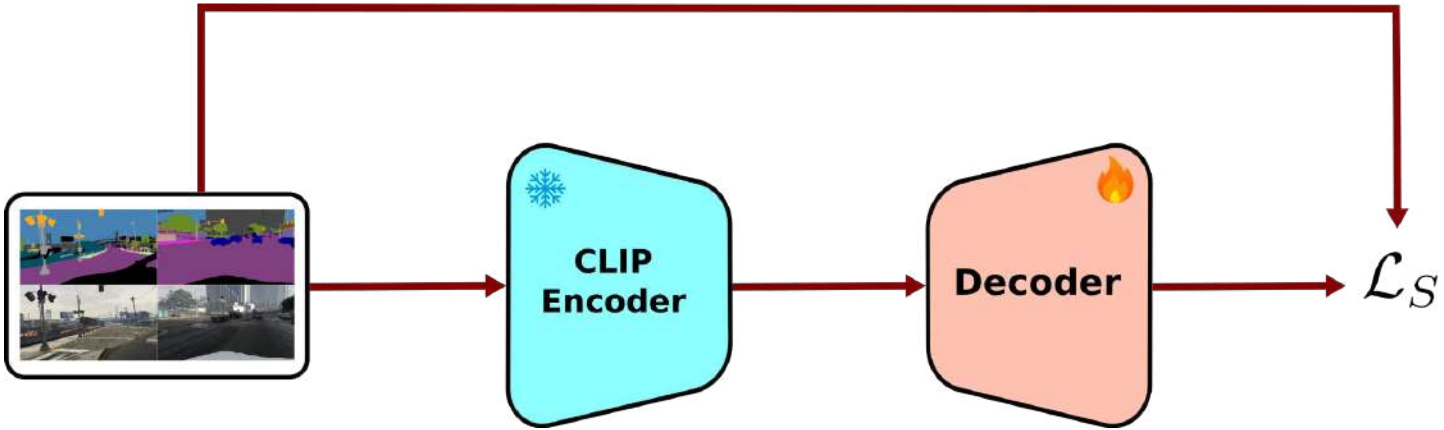
Foundation models

- The rise of large-scale pretrained models, also called Foundation Models (FMs) constitute a new **paradigm shift** in the field

- We believe that bringing the power of FMs would definitely help advance the setting of DGSS :
    - **Obtain robust feature representations to unseen domains** **(CLIP)**
    - **Generate diverse images with varied content and styles for self-training** **(T2I diffusion model + LLM)**
    - **Improve the pseudo labels obtained with self-training** **(SAM)**

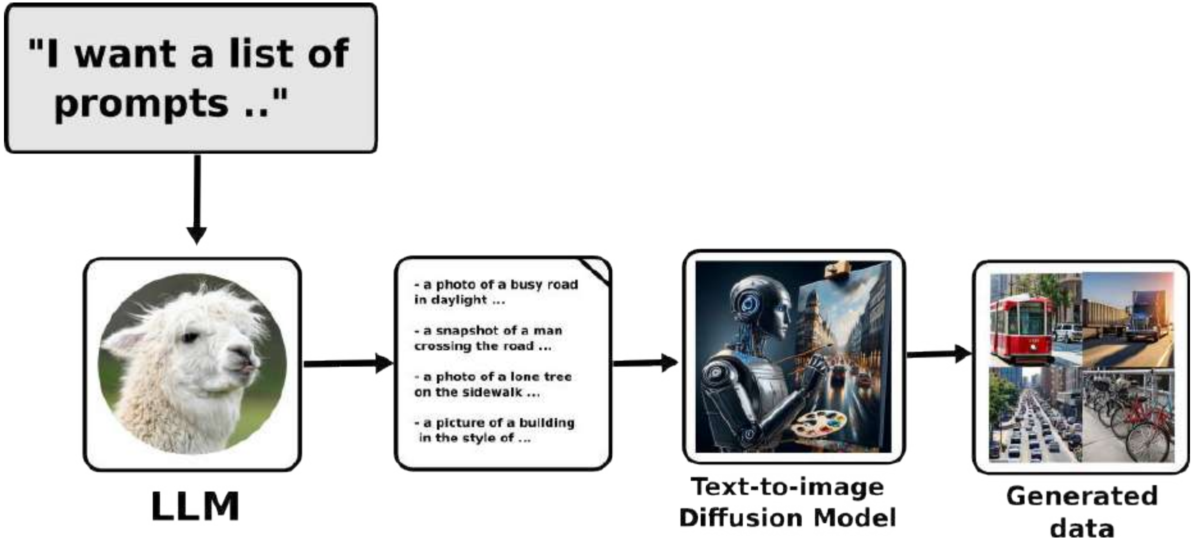
CLOUDS : Segmentation Model

- We use **CLIP** to extract **robust feature representations**
- We freeze the backbone to ensure **preserved generalizability**



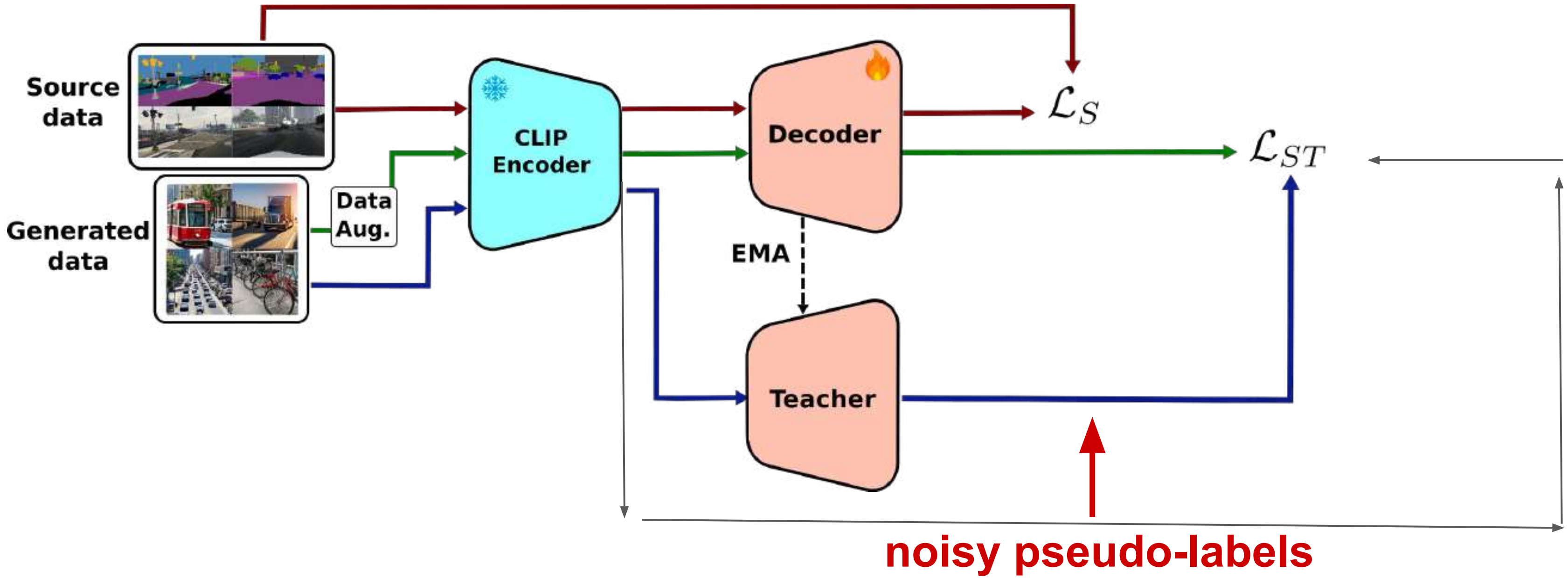
CLOUDS : Data Generation

- We generate synthetic data with a **T2I diffusion model** to simulate unseen domains for self-training
- We use an **LLM** to create descriptive text prompts that condition the T2I diffusion model for generating diverse content and styles



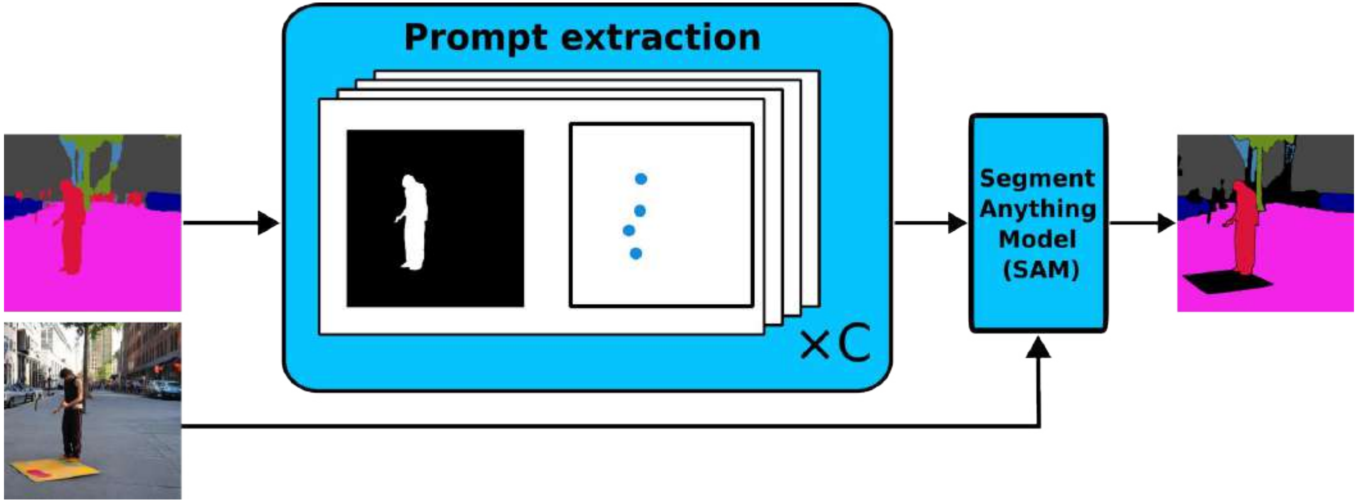
CLOUDS : Self-Training

- We **self-train** the model on the generated data using **pseudo-labels (PLs)**

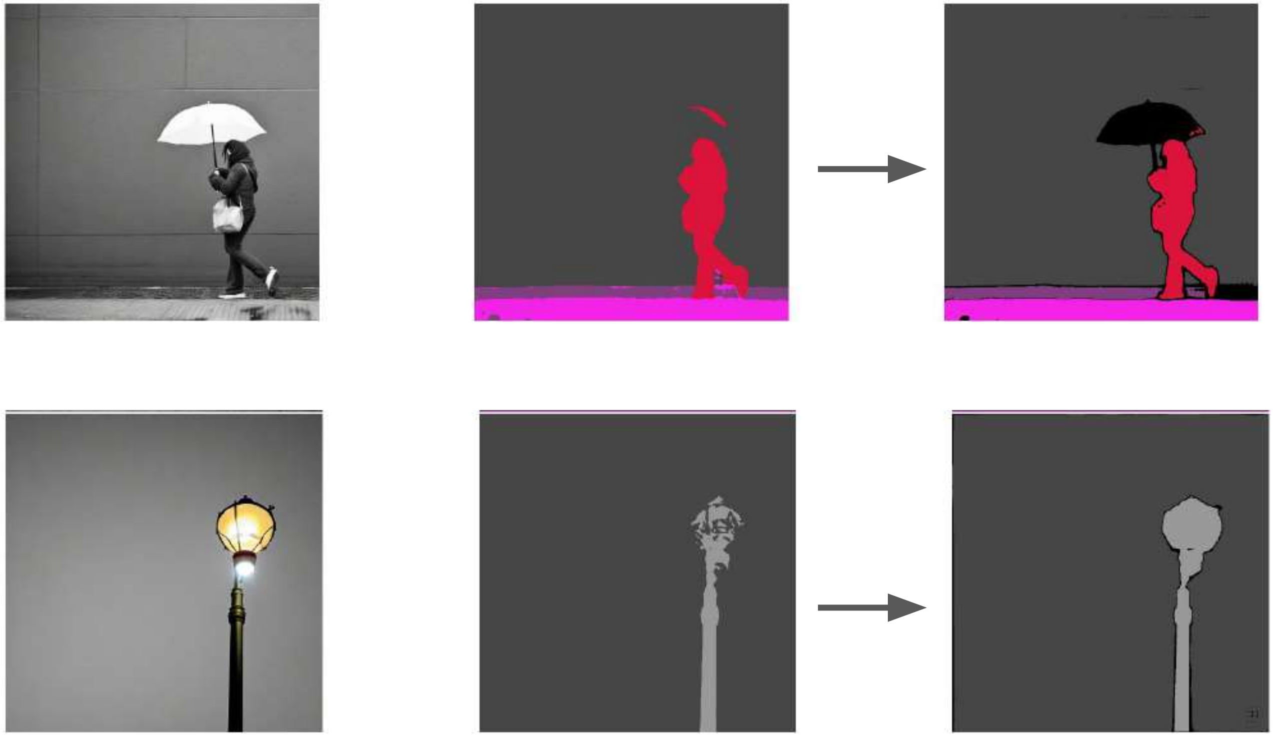


CLOUDS : Pseudo-Label Refinement

- To improve the **noisy pseudo-labels (PLs)**, we use the **Segment Anything Model (SAM)**
- We extract class-wise masks and point prompts for each noisy PL, feeding them to **SAM** to refine masks



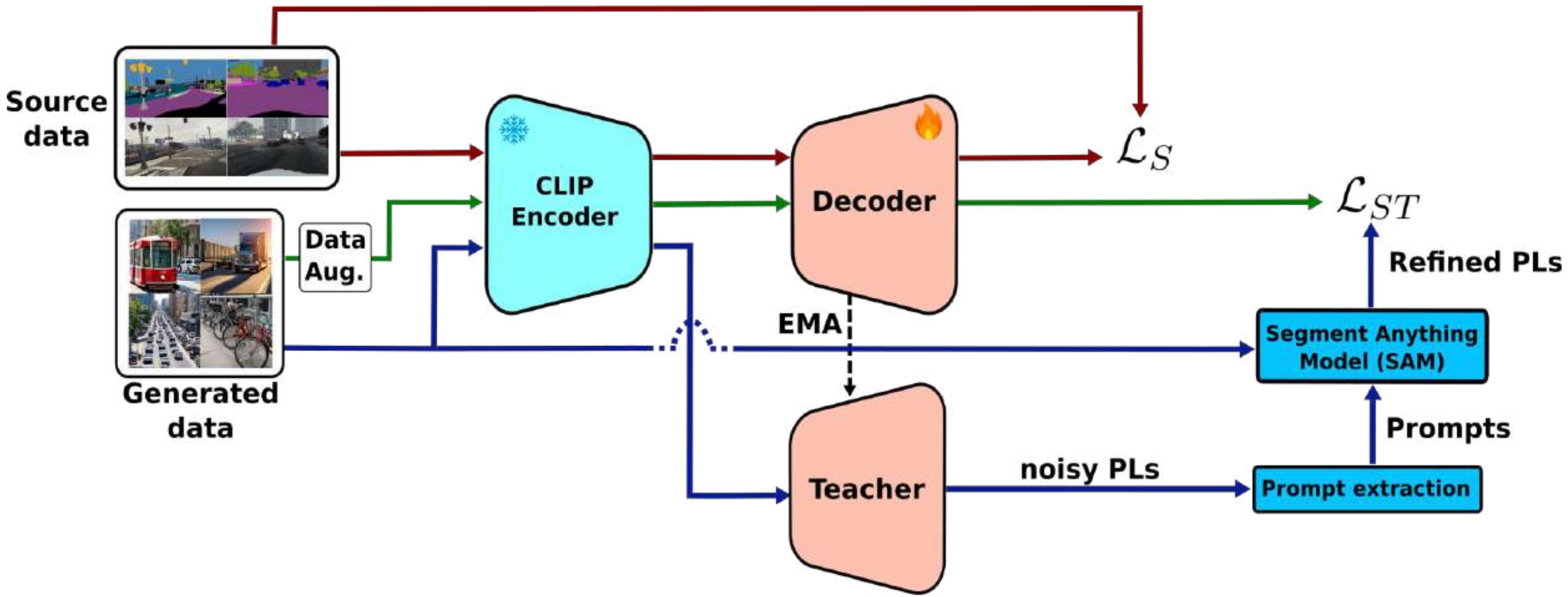
CLOUDS : Pseudo-Label Refinement



Input image Before After

CLOUDS : Full Framework

- We incorportate our PL-refinement module during training for better self-training



Results

- CLOUDS exhibits strong performance on **different backbones** (ResNet-50, 101 and ConvNext-L) using **CLIP pre-training**.

- CLOUDS outperforms SOTA methods pre-trained on **ImageNet** (MiT-B5 backbone)

Method	Encoder	C	B	M	Avg
DRPC [79]	ResNet-50	35.7	31.5	32.7	33.3
SAN-SAW [53]		38.9	35.2	34.5	36.2
MoDify [31]		38.9	33.7	36.2	36.3
TLDR [33]		41.9	34.4	36.8	37.7
CLOUDS (Ours)		**46.1**	**37.6**	**48.1**	**43.9**
DRPC [79]	ResNet-101	37.6	34.4	34.1	35.3
GTR [52]		39.7	35.3	36.4	37.1
FSDR [27]		40.8	37.4	39.6	39.3
SAN-SAW [53]		40.9	36.0	37.3	38.0
TLDR [33]		42.6	35.5	37.5	38.5
HRDA * [26]		34.9	25.0	34.0	31.3
MoDify [31]		43.4	39.5	42.3	41.7
CLOUDS (Ours)		**49.1**	**40.3**	**50.1**	**46.5**
HRDA * [26]	MiT-B5	39.6	32.6	40.0	37.4
CLOUDS (Ours)		**42.2**	**38.3**	**43.6**	**41.4**
FC-CLIP * [78]	ConvNext-L	38.0	29.9	39.0	35.6
CLOUDS (Ours)		**53.4**	**47.0**	**55.8**	**52.1**

Table 2. Comparison with state-of-the-art methods for DGSS on Synthia $\rightarrow$ {Cityscapes (C), BDD (B), Mapillary (M)}. * denotes experiment obtained using the official code

Ablation study

- **CLIP alone** exhibit strong performance (better than previous SOTA)
- **Adding SAM helps** to improve the effectiveness of **self-training**

Backbone	CLIP	{LLM, Diffusion}	SAM	Avg
ResNet-50	✓			50.0
	✓	✓		50.7
	✓	✓	✓	**53.3**
ResNet-101	✓			51.9
	✓	✓		53.3
	✓	✓	✓	**54.7**
ConvNext-L	✓			58.5
	✓	✓		58.6
	✓	✓	✓	**61.5**

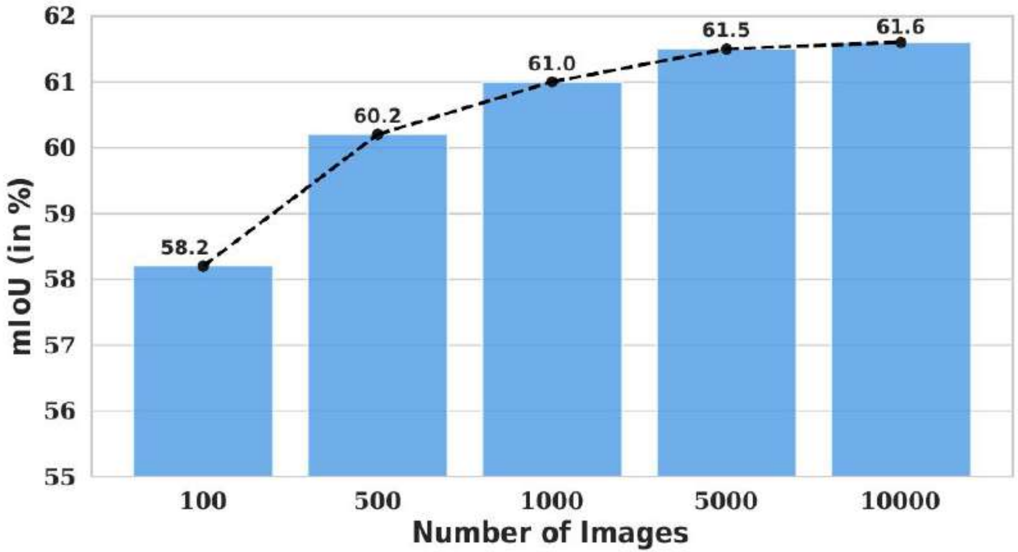
Ablation study

- Freezing the CLIP backbone helps to perform well in unseen domains

Backbone	Cityscapes	BDD100K	Mapillary	Avg.
Trainable	58.6	53.0	62.8	58.1
Frozen	**60.2**	**57.4**	**67.0**	**61.5**

Ablation study

- Increasing the size of the generated data helps to improve results until reaching a plateau



Conclusion

- Our method is one of the first to use **FMs in DGSS**, bridging a gap with the latest advancements in computer vision.

- CLOUDS integrates **CLIP**, a **diffusion model**, an **LLM**, and **SAM** to enhance feature robustness, content and style diversity, and label refinement.

Main Takeaways

- **Use a Foundation Model (CLIP, DINOv2, Eva-clip, etc …) for improved feature representations** :
  - Do not touch the pre-trained weights
  - Apply PEFT if needed

- **If you are in a few-shot scenario, you can use a generative model to generate more data and self-train your model on it.**
  - Use generative models for data augmentation
  - Employ LLMs to increase text diversity (text-to-image diffusion model)
  - Fine-tune diffusion models for domain-specific generation

Future work

- How can we better leverage **the temporal information** to improve adaptation ?
- How can we better integrate **other modalities** to improve visual adaptation ?
- Is it possible to do domain **“un-adaptation”** ?
    - Given a source and target domain closely related, how can we ensure to have a model that performs poorly on a target while keeping a good performance on the source ?