Self-guided semantic segmentation

Maksymilian Kulicki, Osman Ülger, Martin R. Oswald

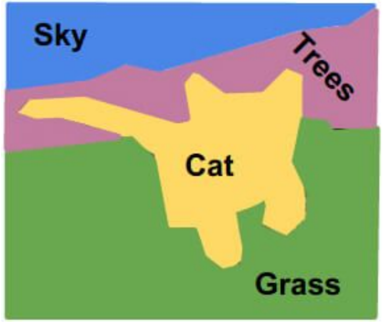
UNIVERSITY OF AMSTERDAM

ATLAS LAB.

Outline

- Background
- Motivation for a new task
- Proposed method
- Evaluation
- Results

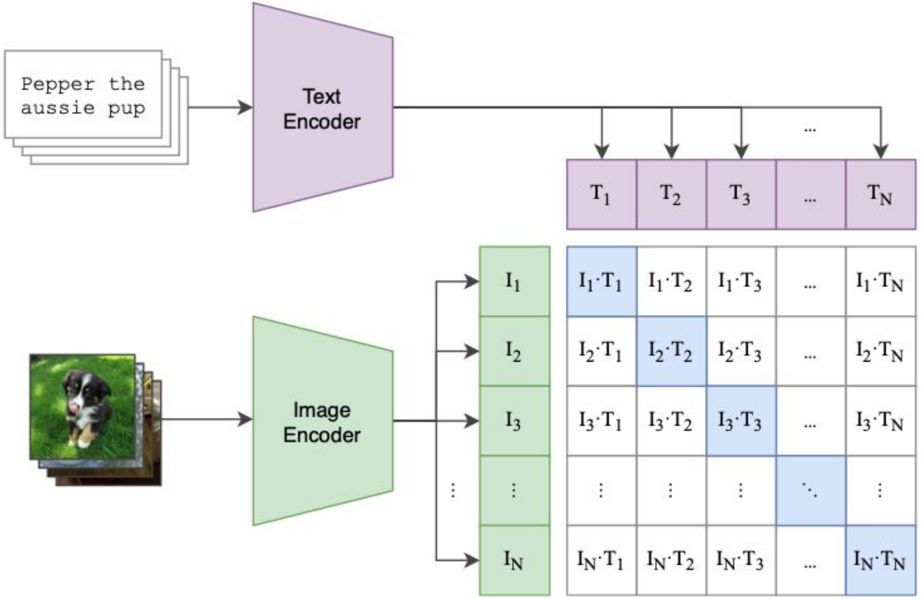
Semantic segmentation



- Assigns each pixel to a specific class (e.g., cat, grass, tree) from a predefined class list
- Trained on labeled datasets with segmentation examples for each class
- Requires laborious manual annotation

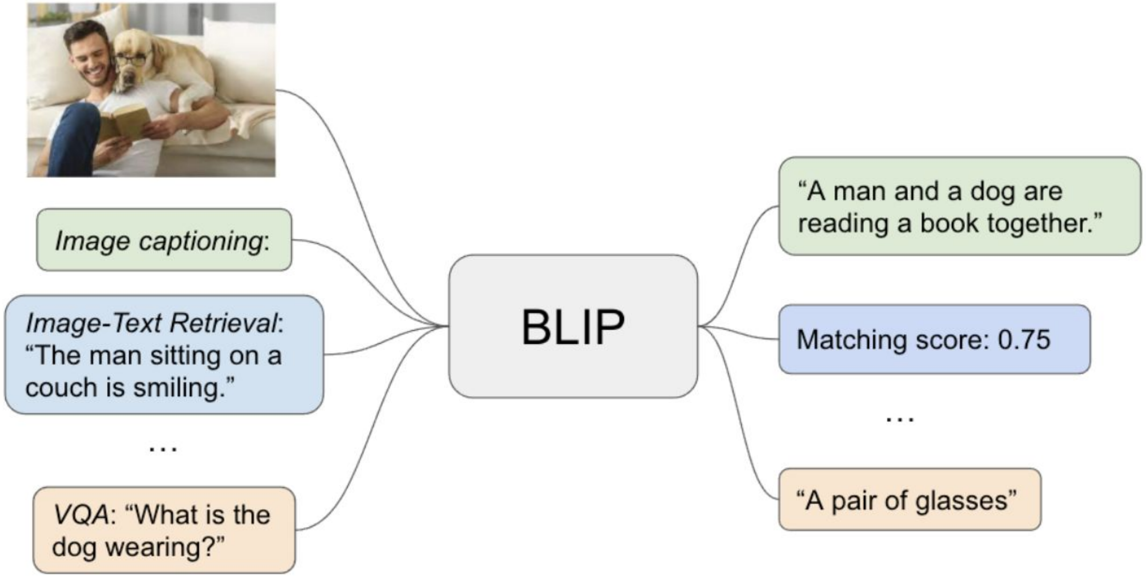
Vision-language pretraining: CLIP

- A model from OpenAI mapping text and images into one embedding space, trained on millions of text-image pairs scraped online
- Enables measuring text-image similarity
- Can be used for image classification, image retrieval, etc.



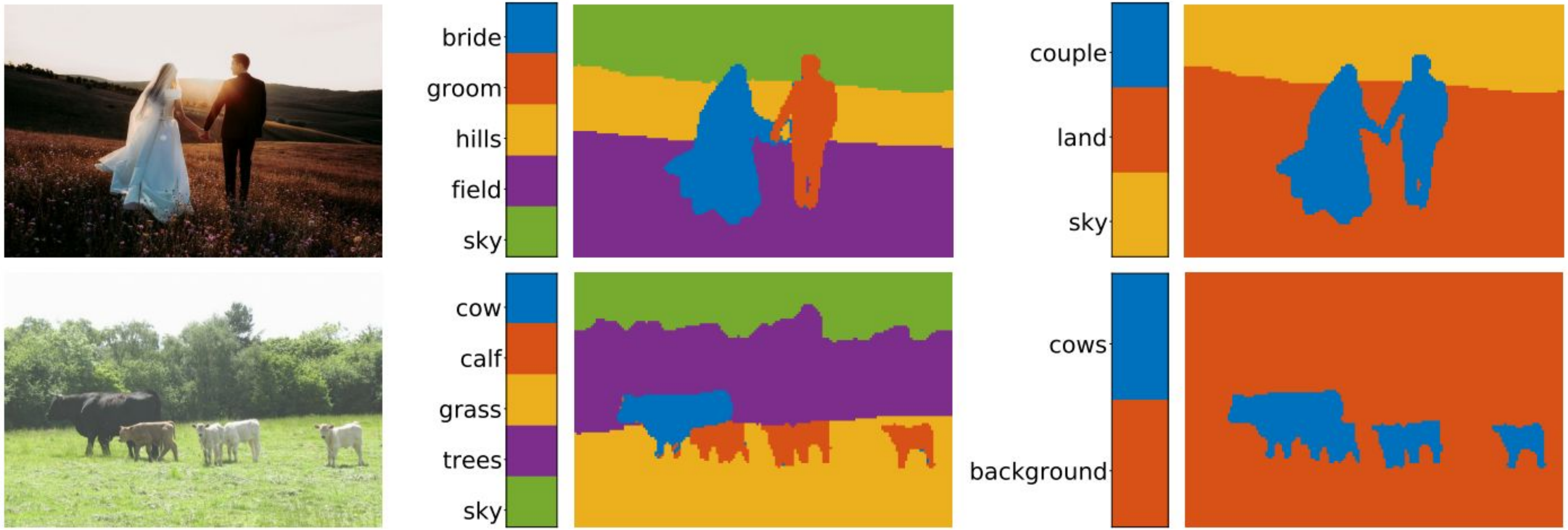
Vision-language pretraining: BLIP

In addition to measuring similarity, BLIP contains a text decoder that can perform image captioning and answer questions



Open-vocabulary segmentation (OVS)

Vision-language models have enabled segmentation with arbitrary classes provided by the user

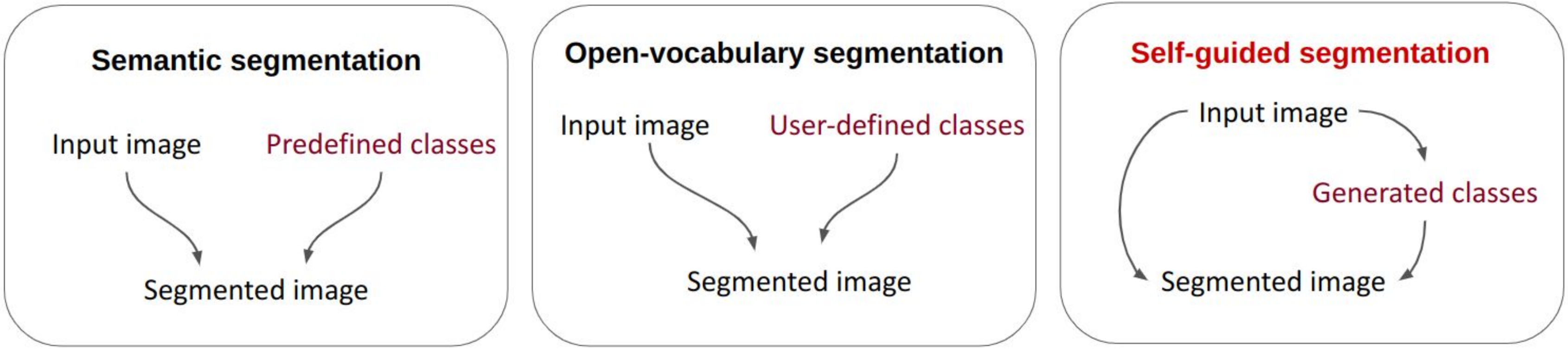


Open-vocabulary segmentation (OVS)

- First method: LSeg (2021)
- Currently there are 21 methods listed on Papers with Code, including OpenSeg (Google Research), OVSeg (Meta AI), X-Decoder (Microsoft)
- Most methods utilize CLIP embeddings as a part of their pipeline

Self-guided segmentation

- Can we remove the need for user-provided labels?
- Our idea: use image captioning to generate labels for OVS for fully automated open segmentation



Problem with image captioning

- Usually only describes the main foreground objects
- Tends to use abstract words

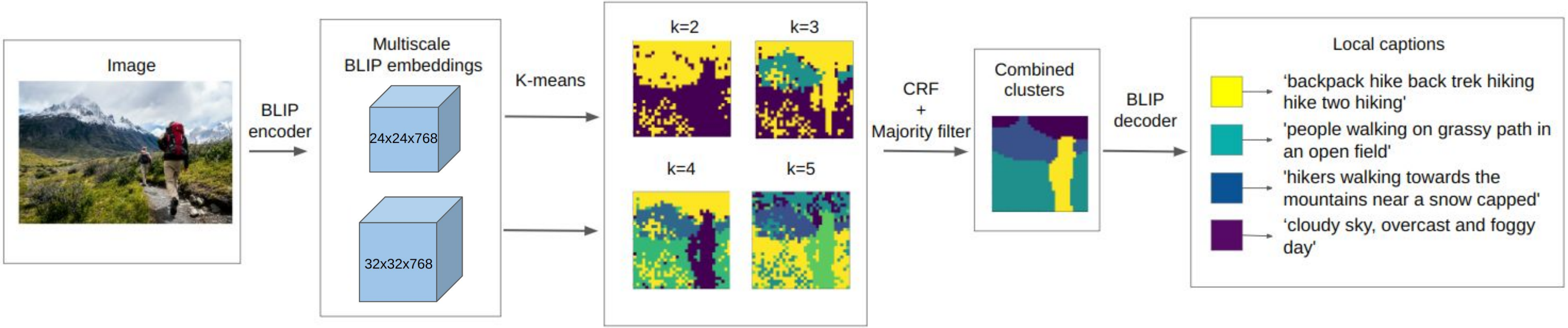
Aerial view of a road in autumn.

a man is riding a motor-bike on a dirt road.

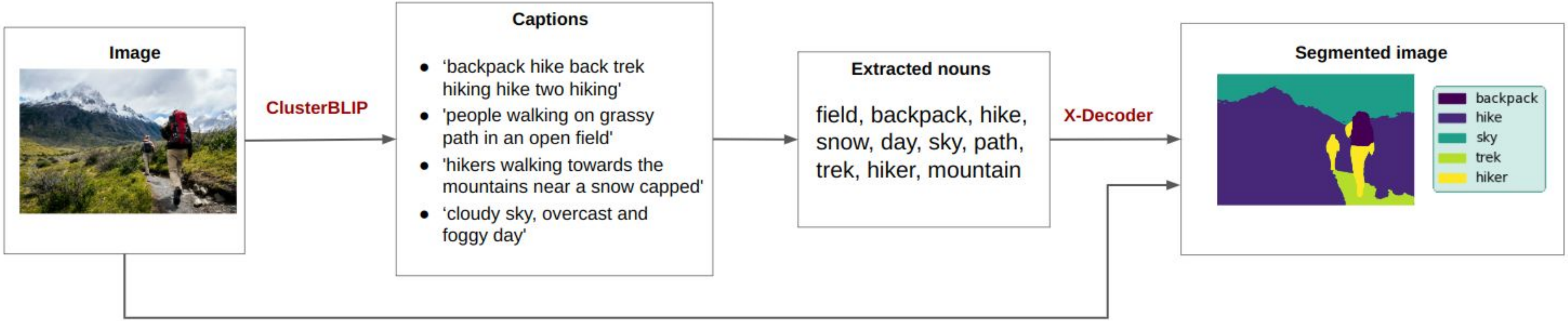
motorcycle that is on dis-play at a show.

Missing: trees | mountain, fence | people, floor, lamps

Solution - local captions using ClusterBLIP

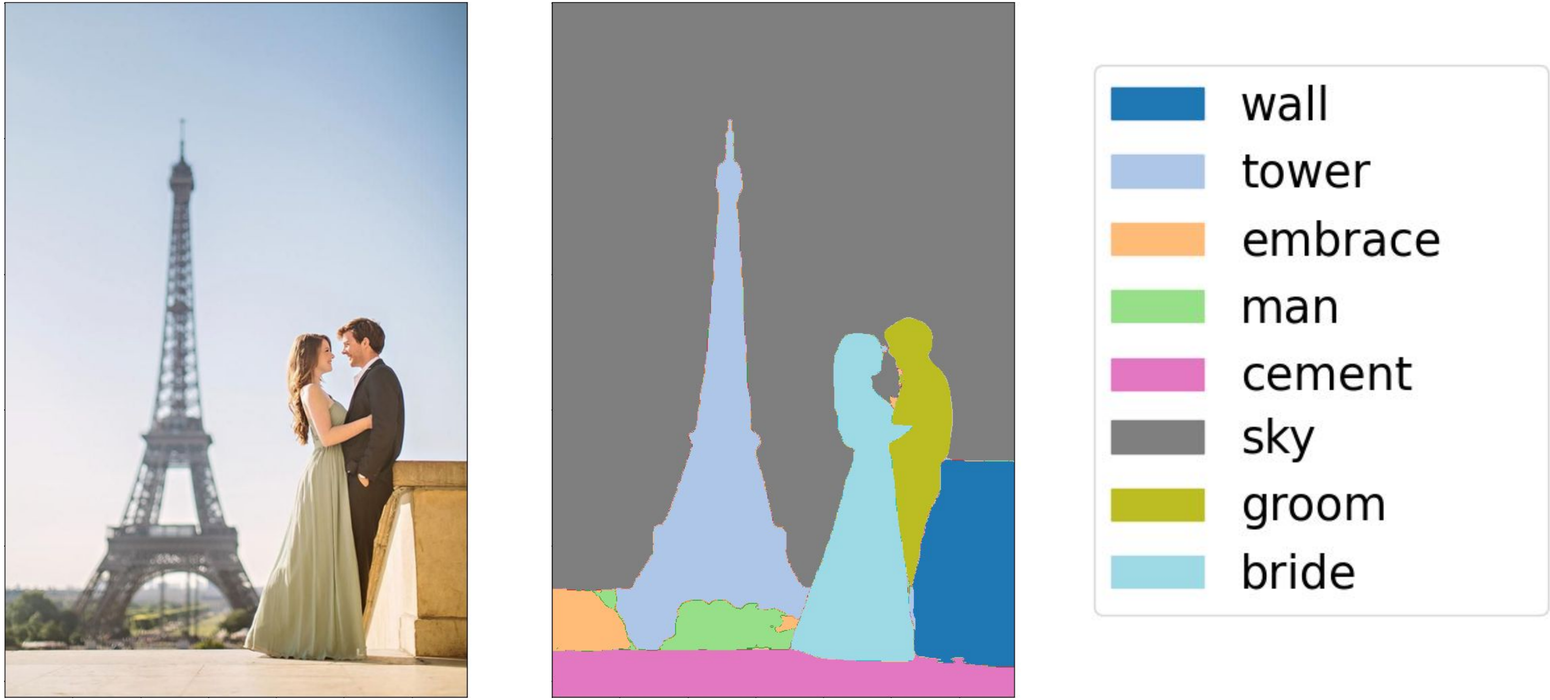


Combining with OVS

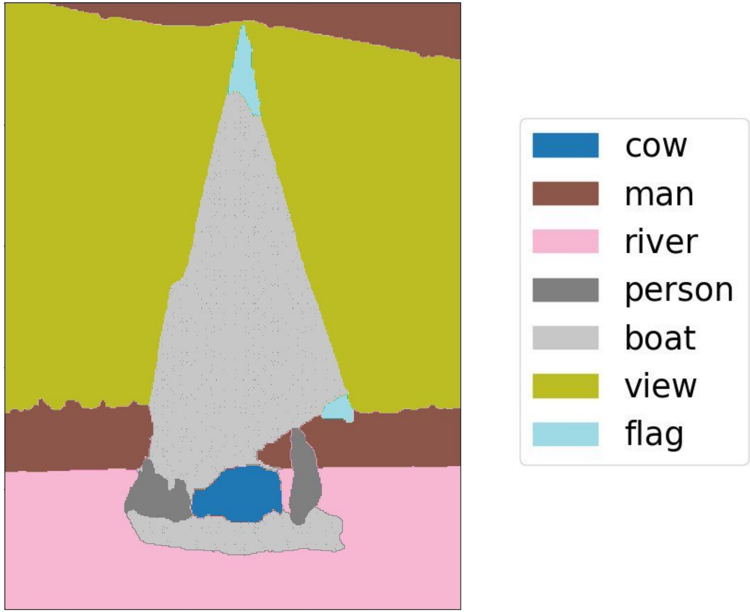


Example output

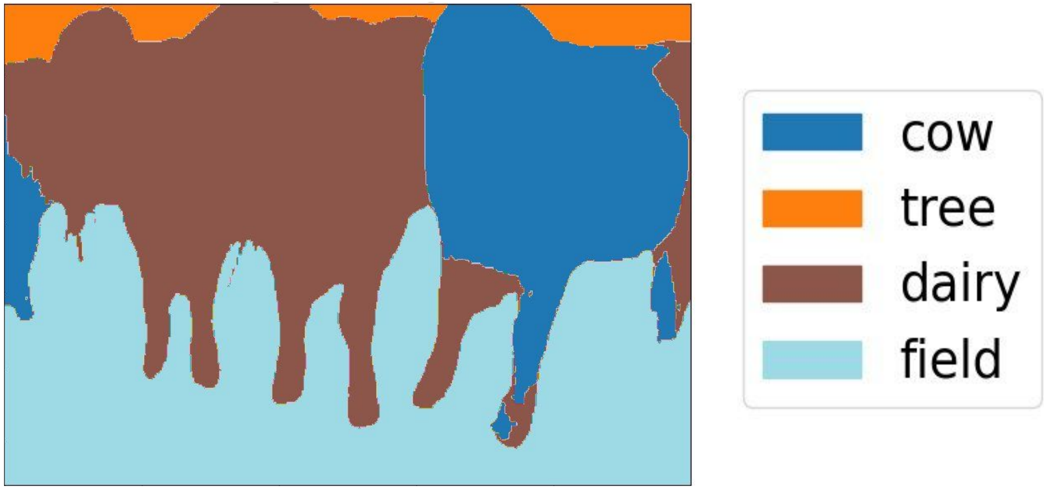
Failure case 1: good masks, wrong labels



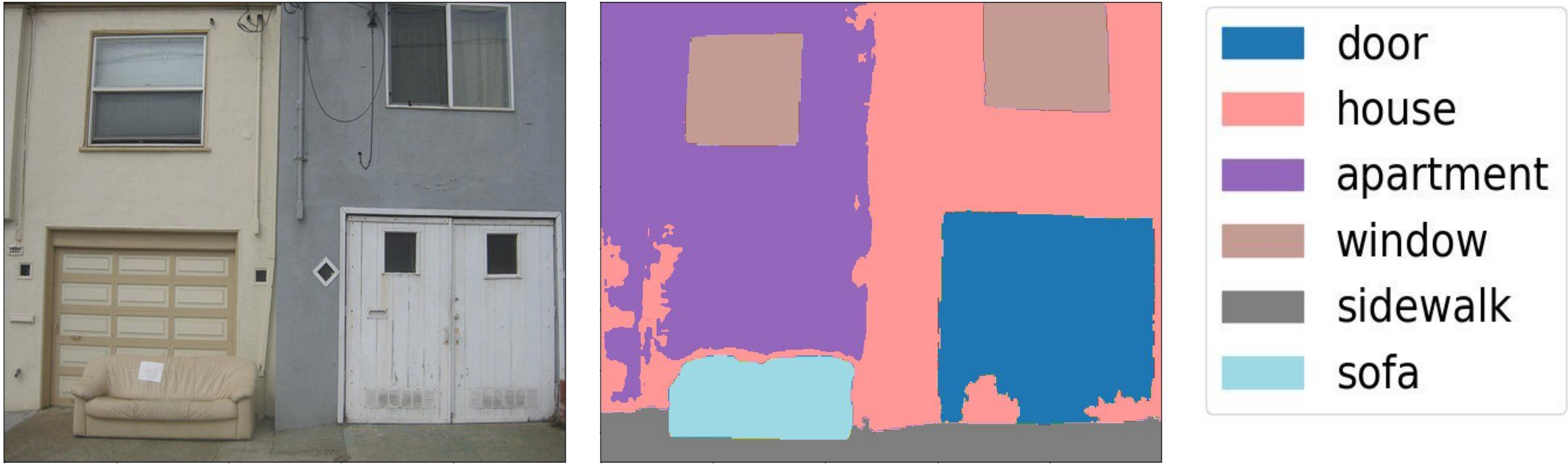
Failure case 1: good masks, wrong labels



Failure case 2: Competition of related labels



Failure case 2: Competition of related labels



Evaluation

- The possible classes are unlimited so there is no clear ground truth
- To evaluate on an established dataset, we map the generated classes to possible ground truth classes
- The mapping is done using SentenceBERT word embeddings
- We measure cosine similarity to find the closest match

- We evaluate on CityScapes (urban driving dataset, 20 classes)

Evaluation example

Baselines

New task, so there are no established baselines. We compare with OVS and with more naive self-guided approaches:

OVS:

- X-Decoder with ground-truth classes present in the image.
- X-Decoder with all possible ground-truth classes from the dataset

Self-guided:

- BLIP + X-Decoder: caption generation with one BLIP embedding per image
- Grid BLIP + X-Decoder: image divided in a 4-part square grid, one BLIP embedding per part

In addition, we try generating multiple captions per embedding. This provides a larger and more diverse set of nouns for X-Decoder.

Results

	Self-guided	Nr of captions	mIoU
X-Decoder (classes from the image)	✗	-	58.6
X-Decoder (all CityScapes classes)	✗	-	50.2
SegSeg (ClusterBLIP + X-Decoder)	✓	1	11.0
		5	23.4
		15	36.5
		25	40.1
		35	39.0
BLIP + X-Decoder	✓	1	11.1
		5	17.3
		15	22.9
		25	12.6
		35	17.7
Grid BLIP + X-Decoder	✓	1	18.4
		5	22.5
		15	32.7
		25	19.3
		35	32.1

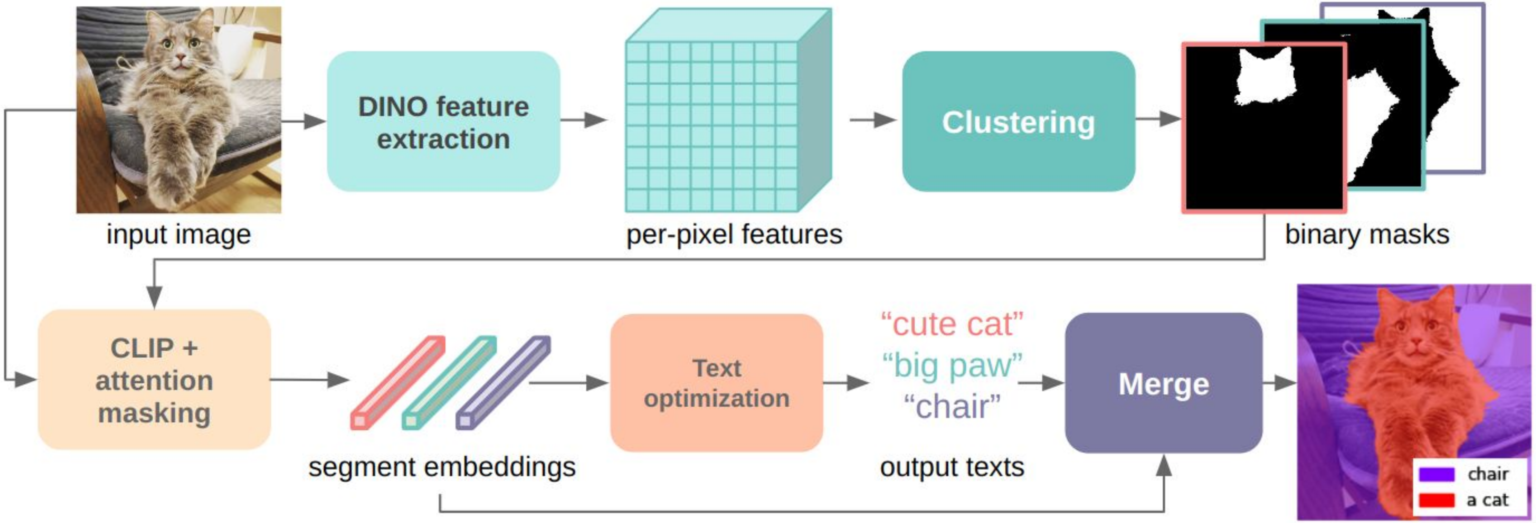
Our method significantly beats the naive self-guided baselines

More captions improve performance, the effect saturates around 15-25 captions.

Our method reaches up to 68.4 percent performance compared to OVS with ground-truth classes provided

Concurrent work

- Rewatbowornwong et al. "Zero-guidance Segmentation Using Zero Segment Labels", ICCV 2023 (2-6 October)
- They propose the same new task, calling it "Zero-guidance Segmentation"
- Their method is different and involved first finding the segments, and then individually labelling them



Thank you for your attention!