Towards Attention-based Approaches for Video Object Segmentation

Quang-Trung TRUONG

Hong Kong University of Science and Technology

June 21, 2023

THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

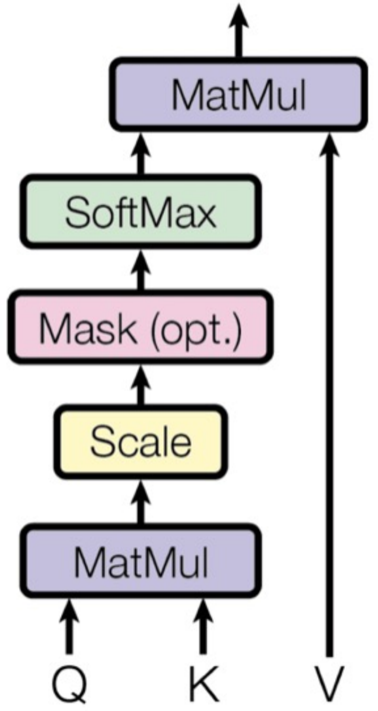
Outline

- **Attention-based methods**
  - **Scaled Dot-Product Attention**
  - **Transformer variants**
- Video object segmentation
  - Introduction
  - A SOTA method – DeAOT [NeurIPS2022]
- A new video dataset “MVK” for retrieval

Scaled Dot-Product Attention

- Scaled dot-product attention
  - Taken from “Attention Is All You Need”
  - Attention(Q, K, V) = softmax($\frac{QK^T}{\sqrt{d_k}}$)V

scaling factor of $\sqrt{d_k}$

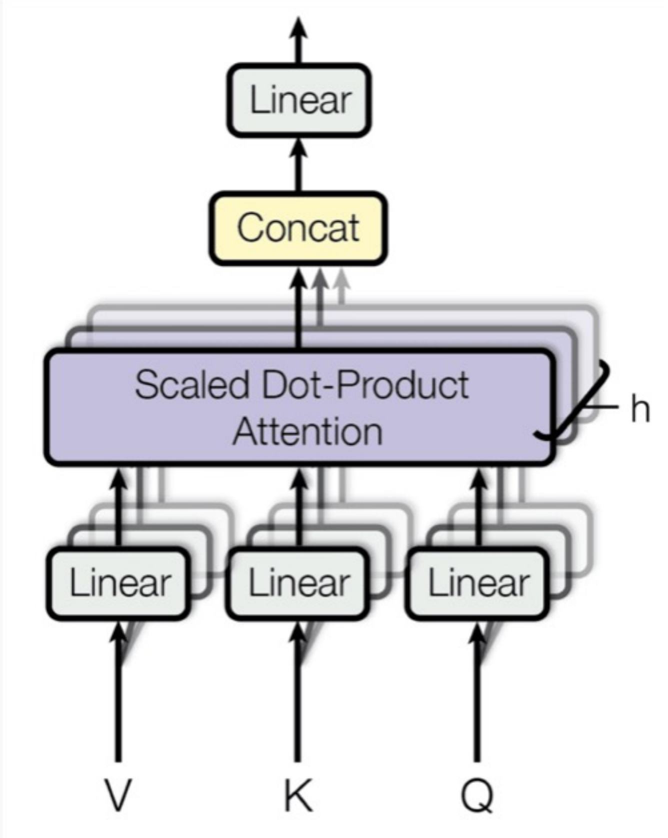


- Have the input: $Q = \{n_q * d_{qk}\}, K = \{n_k * d_{qk}\}, V = \{n_k * d_v\}$
- The output: $\{n_q * d_v\}$

Vaswani, Ashish, et al. "Attention is all you need." *NeurIPS* (2017).

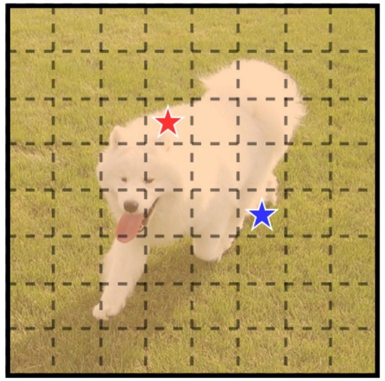
Multi-head Attention

- Linearly projects the queries, keys, and values times
  - Using a different learned projection each time.



=> Extract information from different representation subspaces

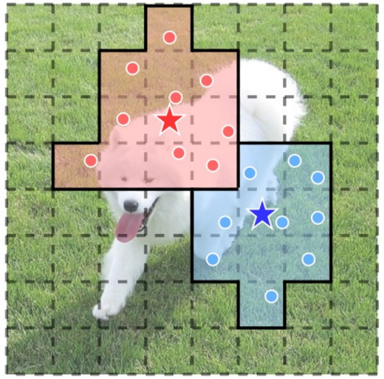
Transformer architectures



(a) ViT

Full global attention

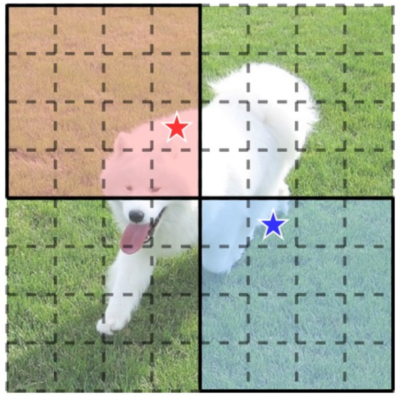
- ✓ Large receptive field
- x High computation cost
- x Slow convergence



(c) DCN

Deformable convolution

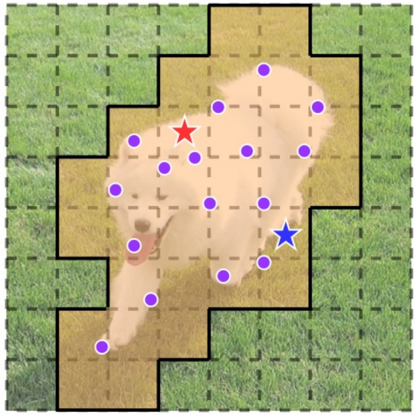
- ✓ Flexible receptive field
- ✓ Different offsets for each query
- x High memory assumption



(b) Swin Transformer

Shift window attention

- ✓ Efficient local relation
- ✓ Data agnostic pattern
- x Receptive field grow slow

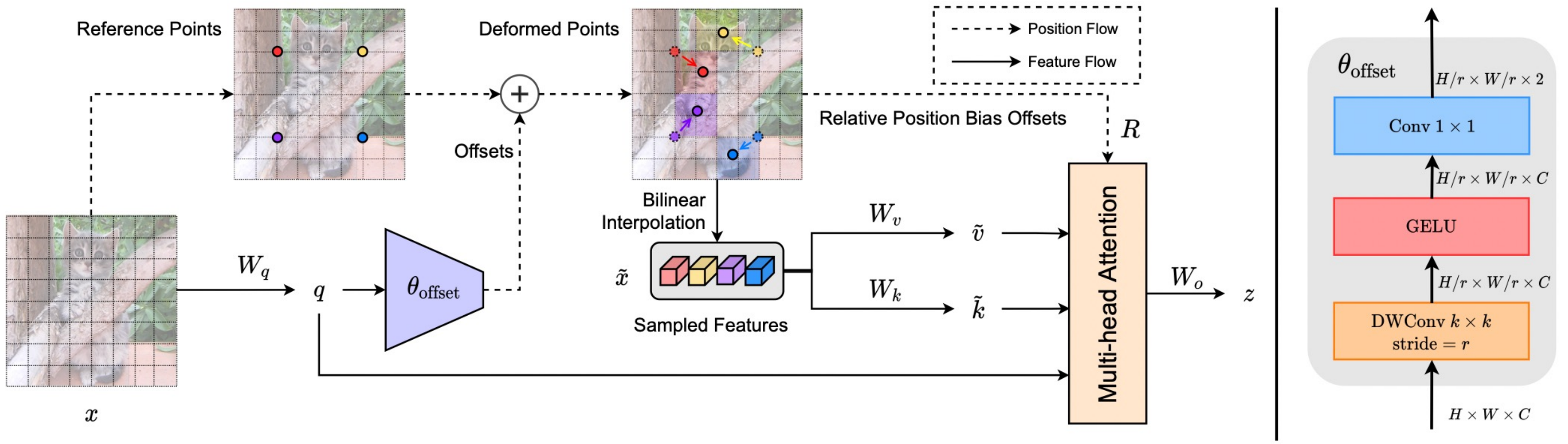


Deformable Attention Transformer

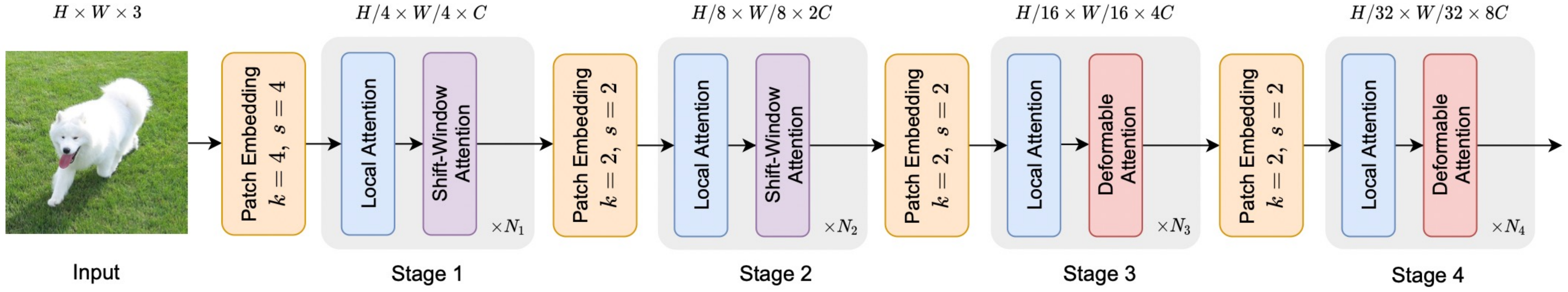
- ✓ Share offset for all queries
- ✓ Shift key to important parts
- ✓ Learned attention pattern
- ✓ Linear space complexity

Xia, Zhuofan, et al. "Vision transformer with deformable attention." *CVPR*. 2022.

Deformable attention

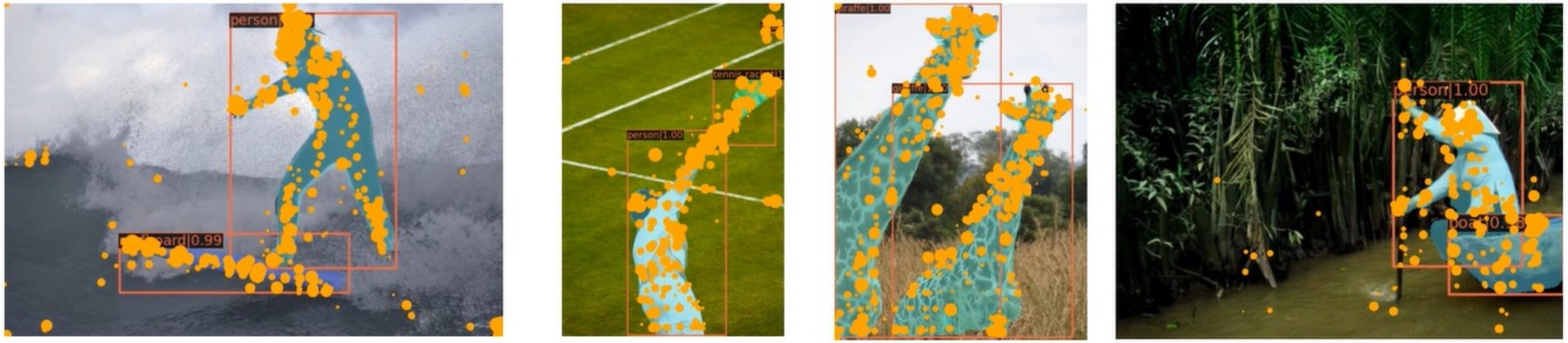


(a) Deformable attention module **(b) Offset network**



Xia, Zhuofan, et al. "Vision transformer with deformable attention." *CVPR*. 2022.

Visualization of DAT results



Visualizations show the most important keys.

Larger circle indicates higher attention scores.

The important keys cover the main parts of the objects.

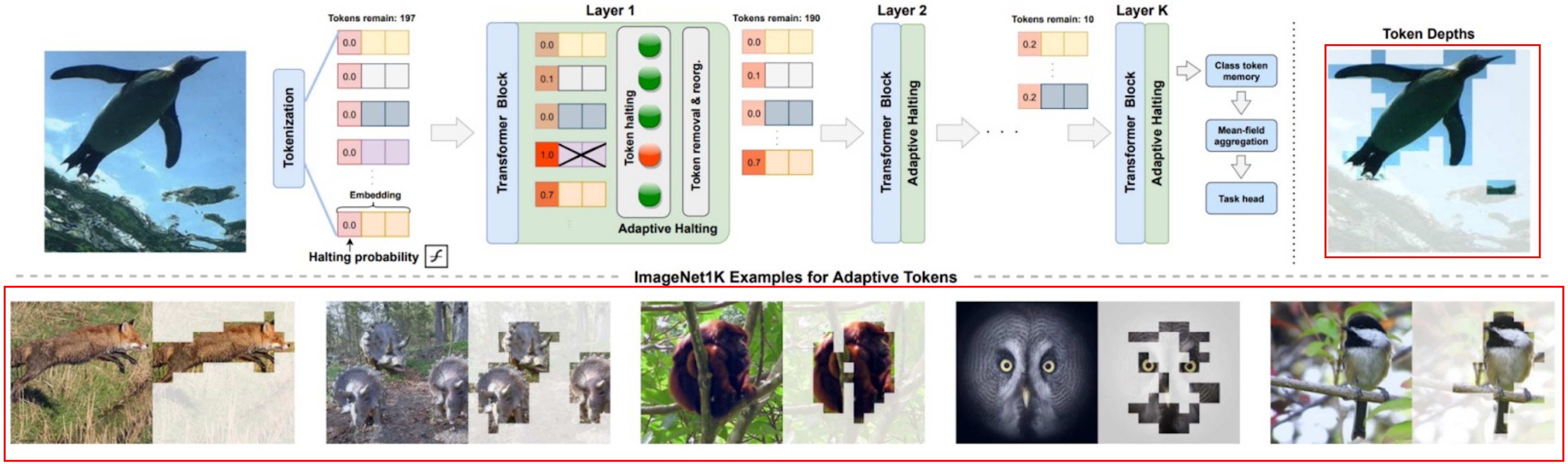
- Attention paradigms show human-like ability in which focuses on the region of interest

- Transformer-based methods (attention paradigm) involve the global receptive field, which is beyond to CNNs.

Xia, Zhuofan, et al. "Vision transformer with deformable attention." *CVPR*. 2022.

A-ViT: Adaptive token

. Not all tokens are equally informative! Let the network decide which ones to halt, adaptively for varying input images.



Enhanced Interpretability

Token depths intuitive
Aligning with varying image semantics

Off-the-shelf Speedup

40-60% throughput improvements of DEIT
No hardware-software modifications

No Auxiliary Nets/Params

Halting based on existing params.
One (existing) embedding re-scaled/biased

Yin, Hongxu, et al. "A-ViT: Adaptive Tokens for Efficient Vision Transformer." *CVPR*. 2022.

Qualitative results

A-ViT

- A benchmark for classification
- We can integrate temporal tokens into patch tokens for video object segmentation

Slide Attention

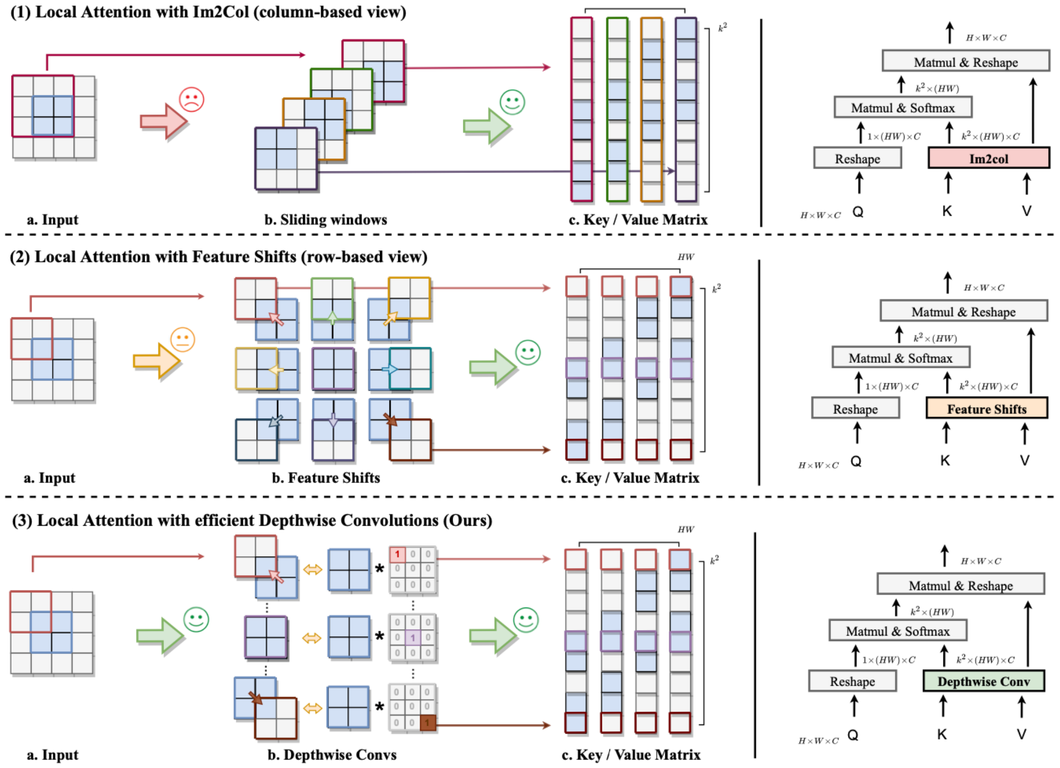


Figure 3. **Different implementation on the local attention module.** We take 3x3 local attention on a 2x2 feature map (in blue) with [1,1] padding (in gray) as an example. **Sub-figure(1)**: Im2Col function is viewed in a *column-based* way, where each column of the key/value matrix corresponds to the local region of a particular query (1.b). The process of sampling windows breaks data locality and leads to inefficiency ✗. **Sub-figure(2)**: we view the key/value matrix in a *row-based* way, where each row is equivalent to the input feature, only after shifting towards certain directions (2.b). Nevertheless, shifting toward different directions is also inefficient when compared with common operators ✗. **Sub-figure(3)**: we take a step forward, and substitute shifting operations with carefully designed depthwise convolutions, which is not only efficient but also friendly to different hardware implementations ✓. Best viewed in color.

- Pros:
  - Local attention
  - Local inductive bias from a query-centric attention pattern
  - Translation-equivariance like traditional convolution
  - Re-interpret the column-based Im2Col function and use Depthwise Convolution
  - Support for devices without CUDA
- Cons: High computation

Pan, Xuran, et al. "Slide-Transformer: Hierarchical Vision Transformer with Local Self-Attention." CVPR 2023.

Slide Attention

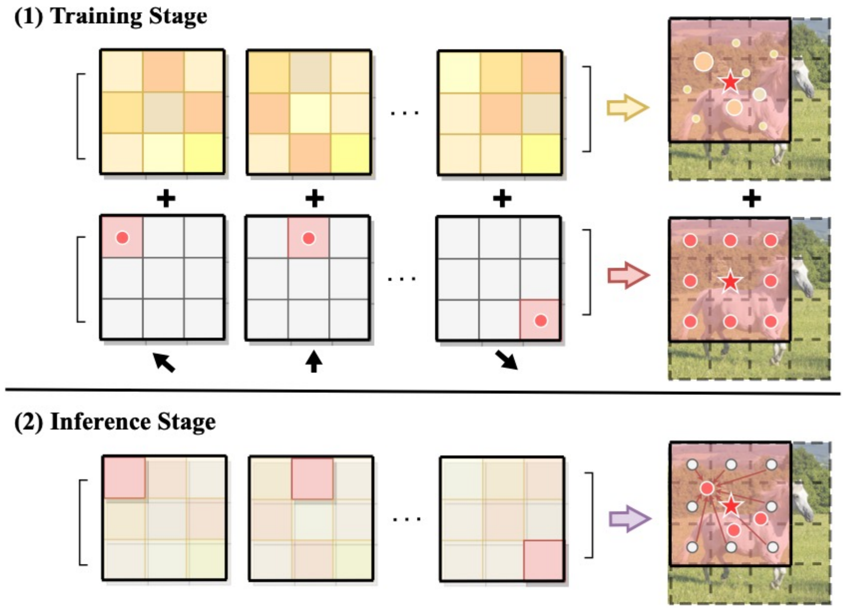
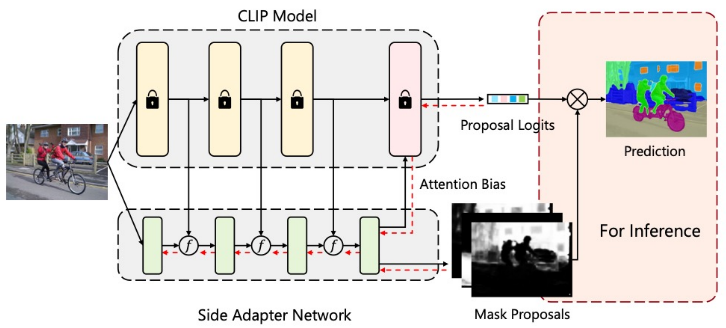


Figure 4. **Deformed shifting module with re-parameterization.** (1) At the training stage, we maintain two paths, one with designed kernel weights to perform shifting towards different directions, and the other with learnable parameters to enable more flexibility. (2) At the inference stage, we merge these two convolution operations into a single path with re-parameterization, which improves the model capacity while maintaining the inference efficiency.

Pan, Xuran, et al. "Slide-Transformer: Hierarchical Vision Transformer with Local Self-Attention." CVPR 2023.

Side adapter network

- Give an intuition of the properties of CLIP
- Propose a gradient flow from the last layers, thought the last layer of CLIP to remaining layer in SIDE network.
- The decoupled design in the architecture is usually superior like DeAOT.
- Regarding engineering, the entire network can be trained end-to-end, allowing the side network to be adapted to the frozen CLIP model, which makes the predicted mask proposals CLIP-aware.



Overview of SAN in training

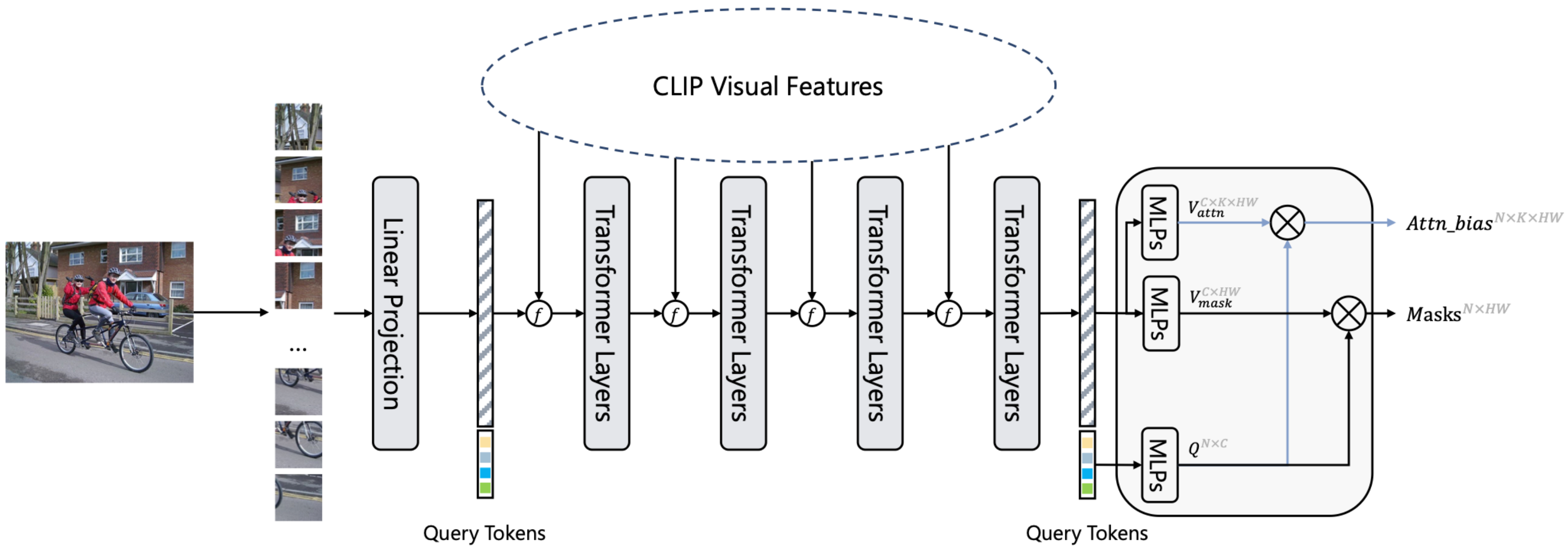


Figure 3. The architecture of the side adapter network. The side adapter network projects the input image to visual tokens and appends query tokens to them at the beginning. Further, it fuses the immediate features of the CLIP model in the middle of transformer layers. The query and visual features are encoded with MLP layers to generate the attention biases and the mask proposals.

Xu, Mengde, et al. "Side adapter network for open-vocabulary semantic segmentation." *CVPR* 2023.

Qualitative results

Figure 1. Segmentation results on ImageNet. For each image, we combine its category with the coco categories as the vocabulary during inference and only visualize mask of the annotated category.

Xu, Mengde, et al. "Side adapter network for open-vocabulary semantic segmentation." *CVPR* 2023.

Outline

- Attention-based methods
  - Scaled Dot-Product Attention
  - Transformer variants
- **Video object segmentation**
  - **Introduction**
  - **A SOTA method – DeAOT [NeurIPS2022]**
- A new video dataset “MVK” for retrieval

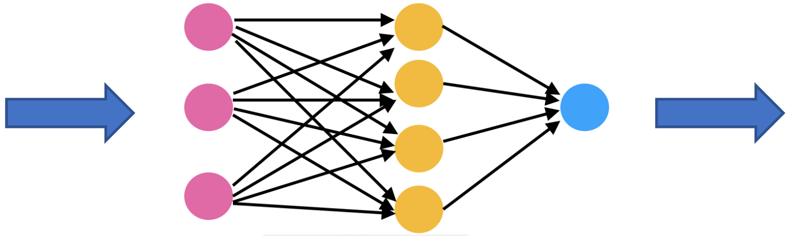
Introduction

- Datasets:
  - DAVIS2016, DAVIS2017, YouTube-VOS

- Input: **video frames** and the **mask** (query objects) at the first frame.
- Output: segment every video frames

mask

Video

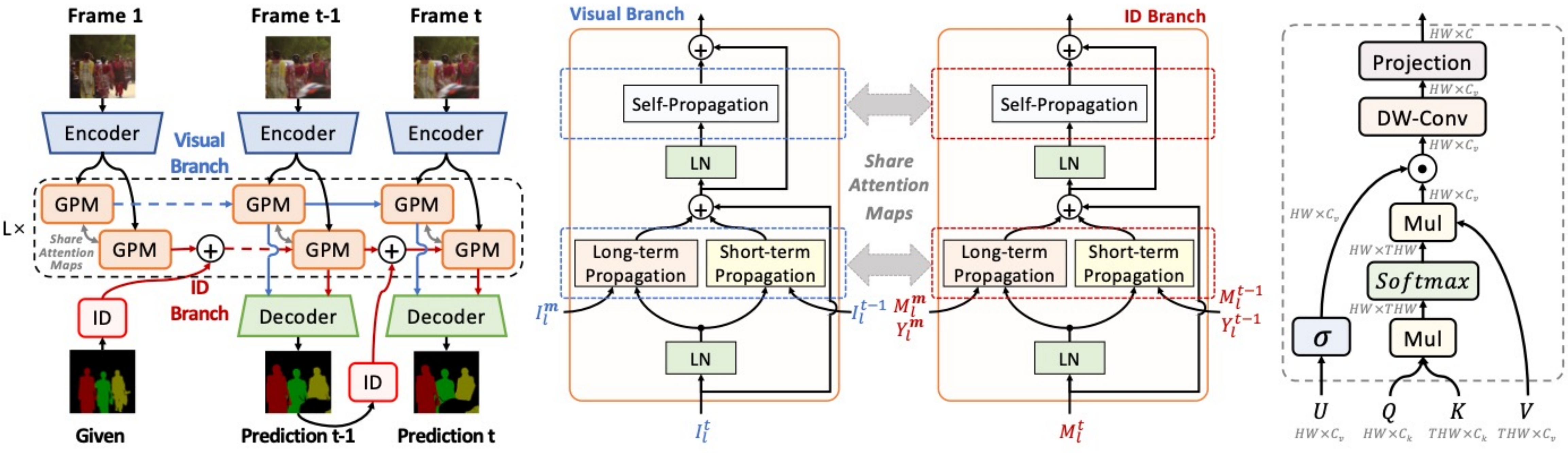


Video Segmentation result

Frame-by-frame technique

Zheng, Sixiao, et al. "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers." *CVPR* 2021.

Memory-based method - DeAOT



(a) Overview (b) Gated Propagation Module (GPM) (c) GP function

Yang, Zongxin, and Yi Yang. "Decoupling Features in Hierarchical Propagation for Video Object Segmentation." *NeurIPS 2022*.

DeAOT: Decoupling Features in Hierarchical Propagation for Video Object Segmentation

- Why paper is good:
  - Decouple the encoder into 2 branches (visual branch and ID branch) in order to the propagation.
  - Replace the convention multi-head attention (many output heads) by Gated Propagation Module (GPM), with only head -> decrease computation.
  - Figure out the limit of number of query objects that AOT fails, this method can maintain the performance.
- Problems:
  - Are separate branches complicated? Because 2 branches are designed for shared weights. One branch is good enough?

DAVIS videos

Quantitative results

Table 1: The quantitative evaluation on multi-object benchmarks, YouTube-VOS [57] and DAVIS 2017 [39]. $\mathcal{J}_S$/$\mathcal{F}_S$/$\mathcal{J}_U$/$\mathcal{F}_U$: $\mathcal{J}$/$\mathcal{F}$ on seen/unseen classes. ‡: timing extrapolated from single-object speed assuming linear scaling in the number of objects. ⋆: recorded on our device.

	YouTube-VOS 2018 Val					YouTube-VOS 2019 Val						DAVIS-17 Val			DAVIS-17 Test			
Method	Avg	$\mathcal{J}_S$	$\mathcal{F}_S$	$\mathcal{J}_U$	$\mathcal{F}_U$	Avg	$\mathcal{J}_S$	$\mathcal{F}_S$	$\mathcal{J}_U$	$\mathcal{F}_U$	fps	Avg	$\mathcal{J}$	$\mathcal{F}$	Avg	$\mathcal{J}$	$\mathcal{F}$	fps
KMN[ECCV20] [43]	81.4	81.4	85.6	75.3	83.3	-	-	-	-	-	-	82.8	80.0	85.6	77.2	74.1	80.3	-
CFBI[ECCV20] [62]	81.4	81.1	85.8	75.3	83.4	81.0	80.6	85.1	75.2	83.0	3.4	81.9	79.3	84.5	76.6	73.0	80.1	2.9
SST[CVPR21] [17]	81.7	81.2	-	76.0	-	81.8	80.9	-	76.6	-	-	82.5	79.9	85.1	-	-	-	-
HMMN[ICCV21] [44]	82.6	82.1	87.0	76.8	84.6	82.5	81.7	86.1	77.3	85.0	-	84.7	81.9	87.5	78.6	74.7	82.5	3.4‡
CFBI+[TPAMI21] [64]	82.8	81.8	86.6	77.1	85.6	82.6	81.7	86.2	77.1	85.2	4.0	82.9	80.1	85.7	78.0	74.4	81.6	3.4
STCN[NeurIPS21] [11]	83.0	81.9	86.5	77.9	85.7	82.7	81.1	85.4	78.2	85.9	8.4⋆	85.4	82.2	88.6	76.1	72.7	79.6	19.5⋆
RPCM[AAAI22] [58]	84.0	83.1	87.7	78.5	86.7	83.9	82.6	86.9	79.1	87.1	-	83.7	81.3	86.0	79.2	75.8	82.6	-
AOT-T [63]	80.2	80.1	84.5	74.0	82.2	79.7	79.6	83.8	73.7	81.8	41.0	79.9	77.4	82.3	72.0	68.3	75.7	51.4
DeAOT-T	**82.0**	**81.6**	**86.3**	**75.8**	**84.2**	**82.0**	**81.2**	**85.6**	**76.4**	**84.7**	**53.4**	**80.5**	**77.7**	**83.3**	**73.7**	**70.0**	**77.3**	**63.5**
AOT-S [63]	82.6	82.0	86.7	76.6	85.0	82.2	81.3	85.9	76.6	84.9	27.1	**81.3**	**78.7**	**83.9**	73.9	70.3	77.5	40.0
DeAOT-S	**84.0**	**83.3**	**88.3**	**77.9**	**86.6**	**83.8**	**82.8**	**87.5**	**78.1**	**86.8**	**38.7**	80.8	77.8	83.8	**75.4**	**71.9**	**79.0**	**49.2**
AOT-B [63]	83.5	82.6	87.5	77.7	86.0	83.3	82.4	87.1	77.8	86.0	20.5	**82.5**	**79.7**	**85.2**	75.5	71.6	79.3	29.6
DeAOT-B	**84.6**	**83.9**	**88.9**	**78.5**	**87.0**	**84.6**	**83.5**	**88.3**	**79.1**	**87.5**	**30.4**	82.2	79.2	85.1	**76.2**	**72.5**	**79.9**	**40.9**
AOT-L [63]	83.8	82.9	87.9	77.7	86.5	83.7	82.8	87.5	78.0	86.7	16.0	83.8	**81.1**	86.4	**78.3**	**74.3**	**82.3**	18.7
DeAOT-L	**84.8**	**84.2**	**89.4**	**78.6**	**87.0**	**84.7**	**83.8**	**88.8**	**79.0**	**87.2**	**24.7**	**84.1**	81.0	**87.1**	77.9	74.1	81.7	**28.5**
R50-AOT-L [63]	84.1	83.7	88.5	78.1	86.1	84.1	83.5	88.1	**78.4**	86.3	14.9	84.9	**82.3**	87.5	79.6	75.9	83.3	18.0
R50-DeAOT-L	**86.0**	**84.9**	**89.9**	**80.4**	**88.7**	**85.9**	**84.6**	**89.4**	**80.8**	**88.9**	**22.4**	**85.2**	82.2	**88.2**	**80.7**	**76.9**	**84.5**	**27.0**
SwinB-AOT-L [63]	84.5	84.3	89.3	77.9	86.4	84.5	84.0	88.8	78.4	86.7	9.3	85.4	82.4	88.4	81.2	77.3	85.1	12.1
SwinB-DeAOT-L	**86.2**	**85.6**	**90.6**	**80.0**	**88.4**	**86.1**	**85.3**	**90.2**	**80.4**	**88.6**	**11.9**	**86.2**	**83.1**	**89.2**	**82.8**	**78.9**	**86.7**	**15.4**

Outline

- Attention-based methods
  - Scaled Dot-Product Attention
  - Transformer variants
- Video object segmentation
  - Introduction
  - A SOTA method – DeAOT [NeurIPS2022]
- **A new video dataset "MVK" for retrieval**

Marine Video Kit: A New Marine Video Dataset for Content-based Analysis and Retrieval

Quang-Trung Truong[1], Tuan-Anh Vu[1], Tan-Sang Ha[1], Jakub Lokoč[2], Yue-Him Wong[3], Ajay Joneja[1], and Sai-Kit Yeung[1]





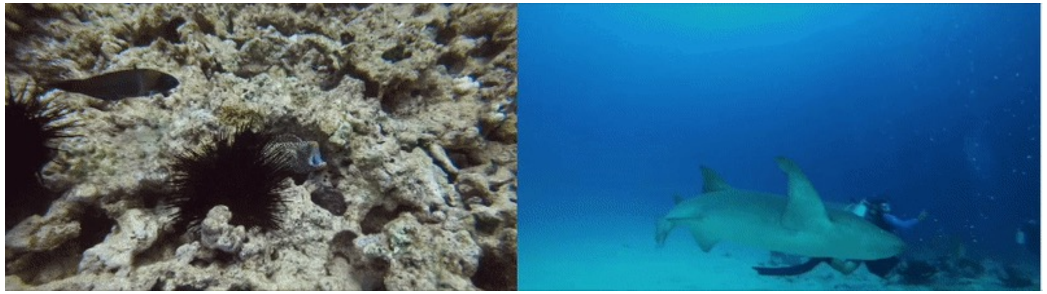


THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY 1

CHARLES UNIVERSITY 2

深圳大学 SHENZHEN UNIVERSITY 3

A new video dataset “MVK”



Marine Video Kit dataset

Existing datasets

Marine-related datasets
Single data, i.e. images or videos

Brackish: object detection
WildFish: fish recognition
OceanDark: image enhancement
Holistic Marine: Object detection, recognition, action recognition

Dataset for content-based retrieval
Provide the text paired with images

V3C dataset

New dataset →

Domain specific dataset for content-based retrieval
Provide the text paired with images

Marine Video Kit dataset

Truong, Quang-Trung, et al. "Marine video kit: a new marine video dataset for content-based analysis and retrieval." *MMM* 2023.

Stats

- 1379 single-shot videos
- 11 dive sites
- Mean duration: 29.9 seconds, median duration : 25.4 seconds
- Videos with a length from 2 seconds to 4.95 minutes
- 43797 selected frames
- Cameras: Canon PowerShot G1 X, Sony NEX-7, OLYM- PUS PEN E-PL, Panasonic Lumix DMC-TS3, GoPro cameras, and consumer cellphones cameras.

Stats

- Naming conventions:
  - format video names as “location_time” pattern to explicitly represent the time and location that they were captured, ex: “Oahu_Jul2022”

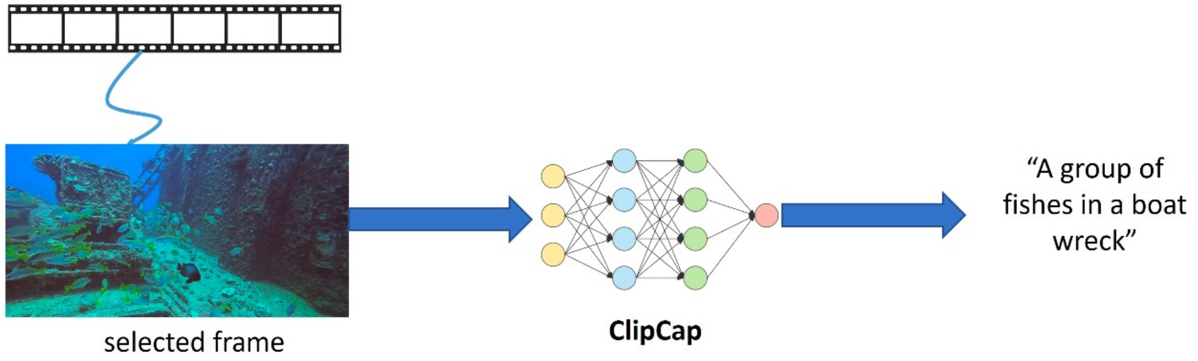
```
Marine Video Kit dataset/
├── videos/
│   └── Oahu_Jul2022/
│       ├── 0001.mp4
│       └── ...
└── information/
    ├── metadata/
    │   └── Oahu_Jul2022/
    │       ├── 0001.json
    │       └── ...
    ├── selected_frames/
    │   └── Oahu_Jul2022/
    │       ├── 0001_00001.jpg
    │       ├── 0001_00002.jpg
    │       └── ...
    └── thumbnails/
        └── Oahu_Jul2022/
            ├── 0001_00001.jpg
            ├── 0001_00002.jpg
            └── ...
```

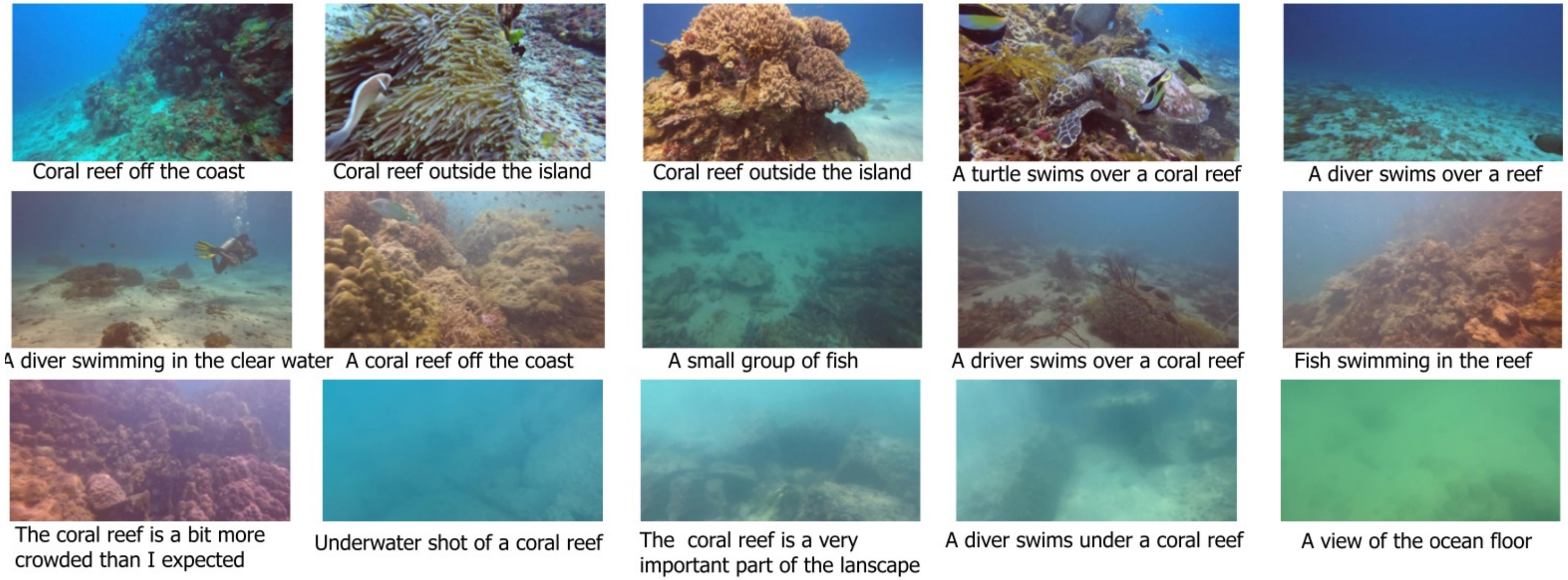
Directory structure


coral reef outside the island.
coral reef outside the island.
coral reef outside the island.
fish swimming around the reef.
aerial view of the coral reef and beautiful fish.
coral reef outside the island.
coral reef outside the island.
a reef of soft corals.
coral reef outside the island.

ClipCap Descriptions [1]

- CLIP [2] (Contrastive Language–Image Pre-training) is good for the general scene, including text and image data
- CLIP builds on a large body of work on zero-shot transfer, natural language supervision, and multimodal learning
- Pros:
  - Train on costly datasets, namely 14 million images for 22,000 object categories
  - Exploit computation power for automatic generation of data in high quality.
- Cons:
  - Struggle on more abstract or systematic tasks such as counting the number of objects in an image and on more complex tasks





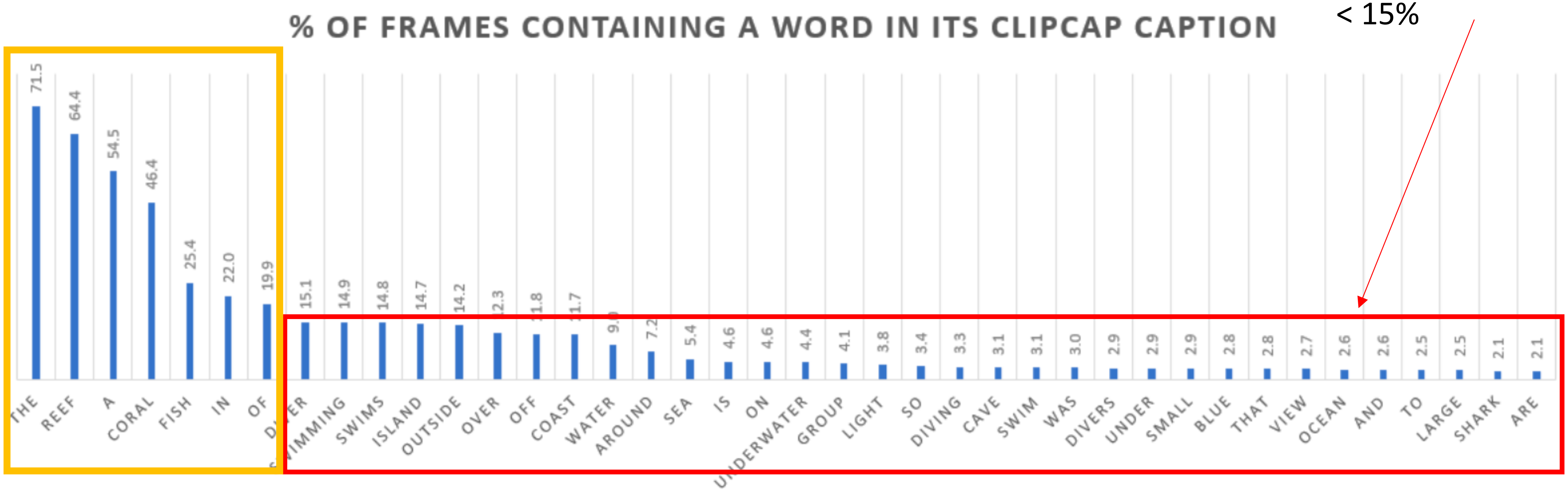
[1]. Mokady, R., Hertz, A., Bermano, A.H.: Clipcap: Clip prefix for image captioning. arXiv preprint
[2]. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning.

ClipCap Descriptions

- Frequencies for individual words in frame captions
- There are 43797 descriptions on the dataset.

Word's frequencies < 15%

Most common words



=> Marine Video Kit dataset is challenging to many vision tasks, especially image captioning

A Benchmark for Known-item Search

- We provide an experiment for video content-based retrieval and analysis
- Three main retrieval contents are presented:
  - Descriptions created by novice users
  - Descriptions created by VBS experts
  - Descriptions generated by ClipCap model
- Motivation for the experiment
  - Made a new domain specific video collections that represents an important practical problem
    - Introduce a benchmark for a respected cross-modal based know-item search approach

Known Item Search

- Given 40K video frames $F = \{f_i\}$, $i \in \{1,2,3.., 40K\}$ from MVK
- KIS task consists of several steps as the following:
  1. Randomly select 5 video frames from F: $F_q = \{f_{11}, f_{15}, f_{23}, f_{25}, f_{40}\}$
  2. Users provide text queries with respect to F_q. Users are given query images from the dataset but they don't know their id in the dataset. They need to find ID "i".

Ex:

frame0-00-28-03.jpg	CLIPCAP	Novice user	Expert
	underwater footage of a coral reef.	blurred underwater footage, maybe a coral reef	blurry view of a sea bottom covered with brown stones

Known Item Search (cont)

- KIS task consists of several steps as the following:
  1. Randomly select 5 video frames from F: $F_q = \{f_{11}, f_{15}, f_{23}, f_{25}, f_{40}\}$
  2. Users provide text queries with respect to F_q
  3. CLIP extractor: Texts -> CLIP embeddings

     Extract CLIP embeddings of F_q: $Q=\{q_i\}=\{q_{11}, q_{15}, q_{23}, q_{25}, q_{40}\}$
  4. CLIP extractor: An image -> CLIP embeddings

     Extract CLIP embeddings of F: $T=\{t_i\}=\{t_1, t_1 \ldots t_{40K}\}$
  5. Use cosine distance D_{cosin} as the similarity metric to find a pair of similar embeddings Q and T. Ex: given query q_{11}, we ranks the cosine distances of the query and video frames from the dataset as the ascending.

     Top1. $d_{cosin}(q_{11}, t_{234}) = 0.001$

     Top2. $d_{cosin}(q_{11}, t_{102}) = 0.003$

     Top3. $d_{cosin}(\mathbf{q_{11}, t_{11}}) = 0.004$

     Top4. $d_{cosin}(q_{11}, t_{34}) = 0.009$

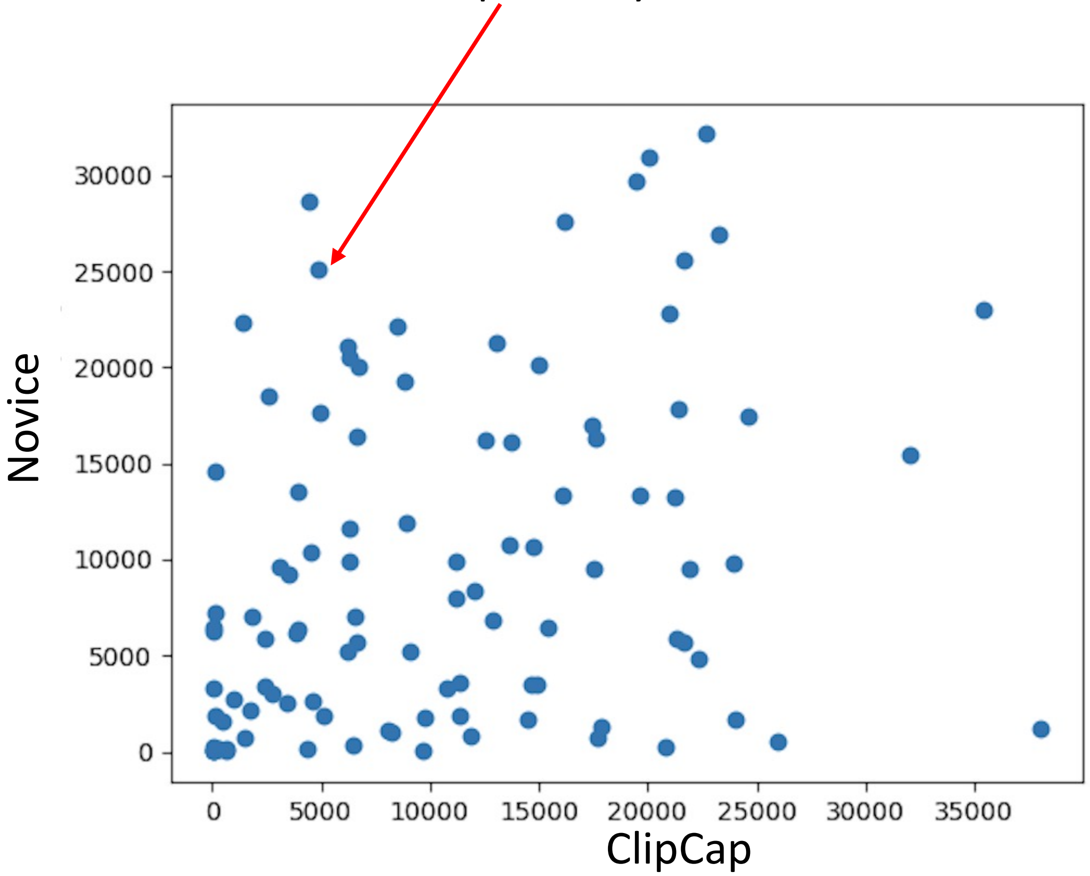
     Top40k. $d_{cosin}(q_{11}, t_i)$ = the highest

     => We find exactly query frame f_{11} in top 3 when $d_{cosin}(\mathbf{q_{11}, t_{11}})$ appears in top 3

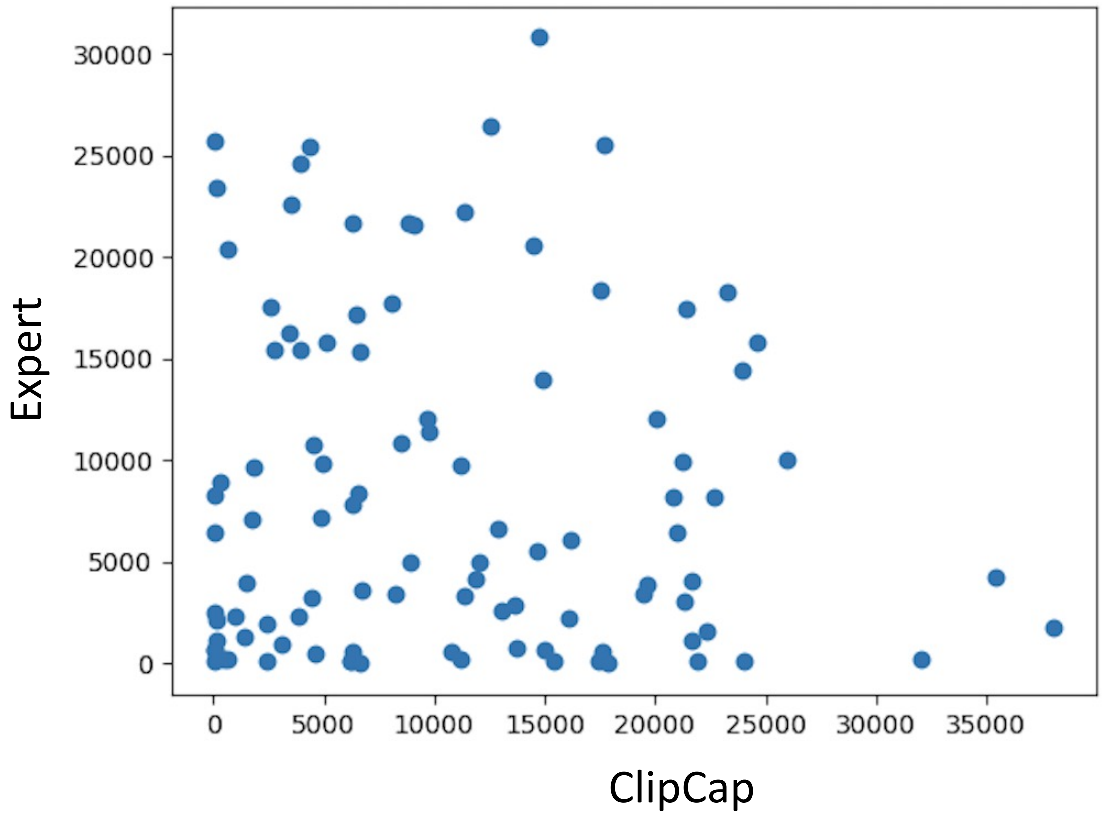
Plot 100 queries

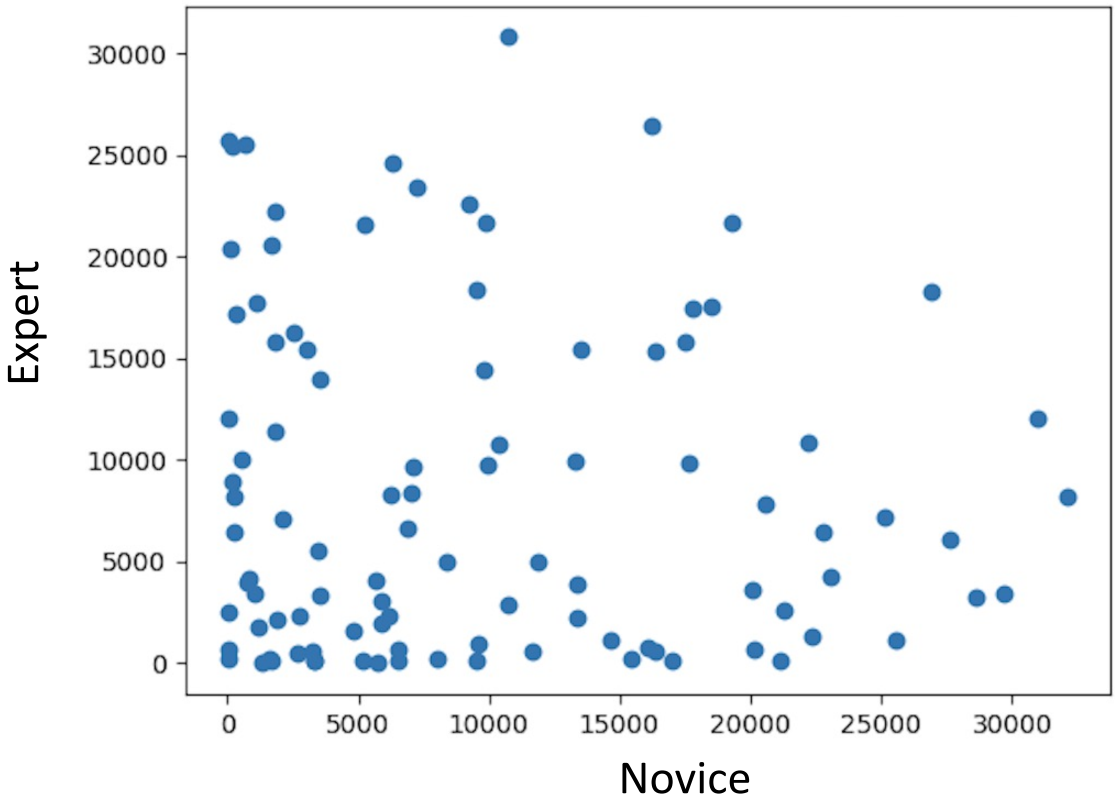
- Given 100 images, we visualize ranks for Novice and ClipCap text queries
- Each point represents a rank for Novice query and ClipCap queries belonging to a video frame $f \in F_q$

This point figures out the text queries of Novice and ClipCap will find the target image when searching top **28000 and 4000** respectively



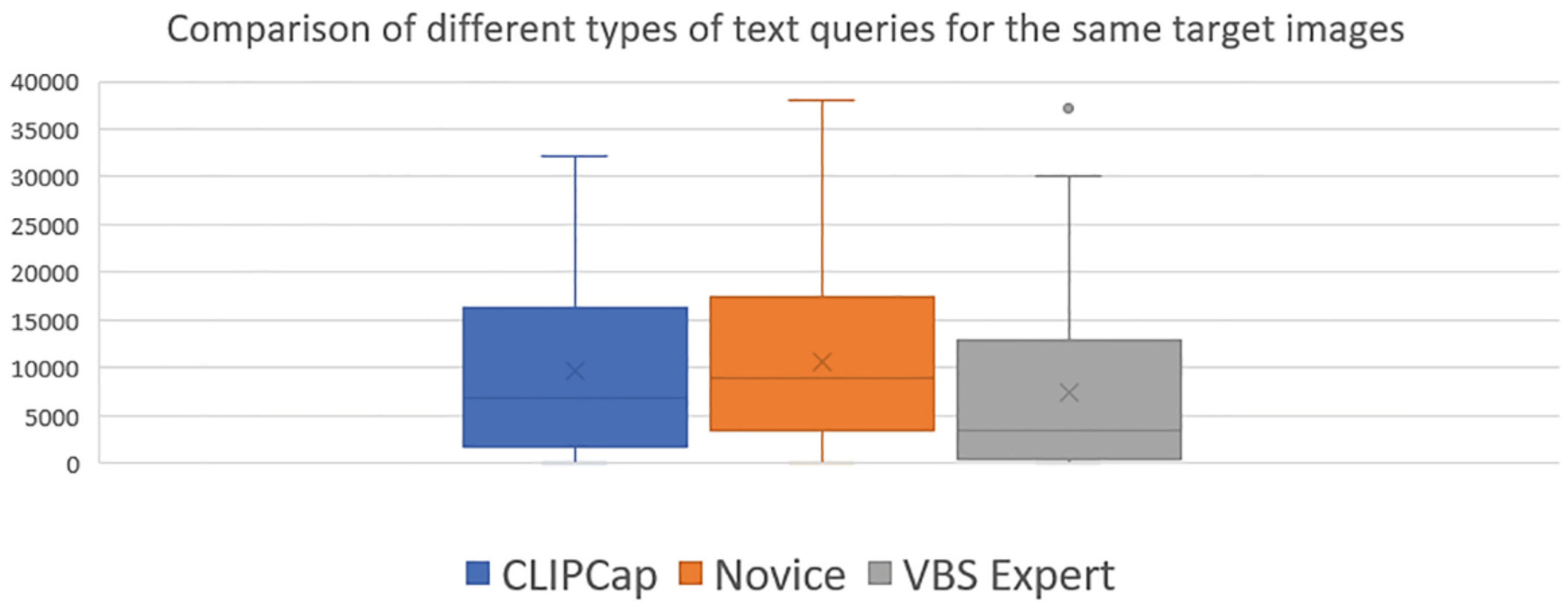
Plot 100 queries





Ranks for ClipCap, Novice, and VBS Expert text queries for 100 target images.

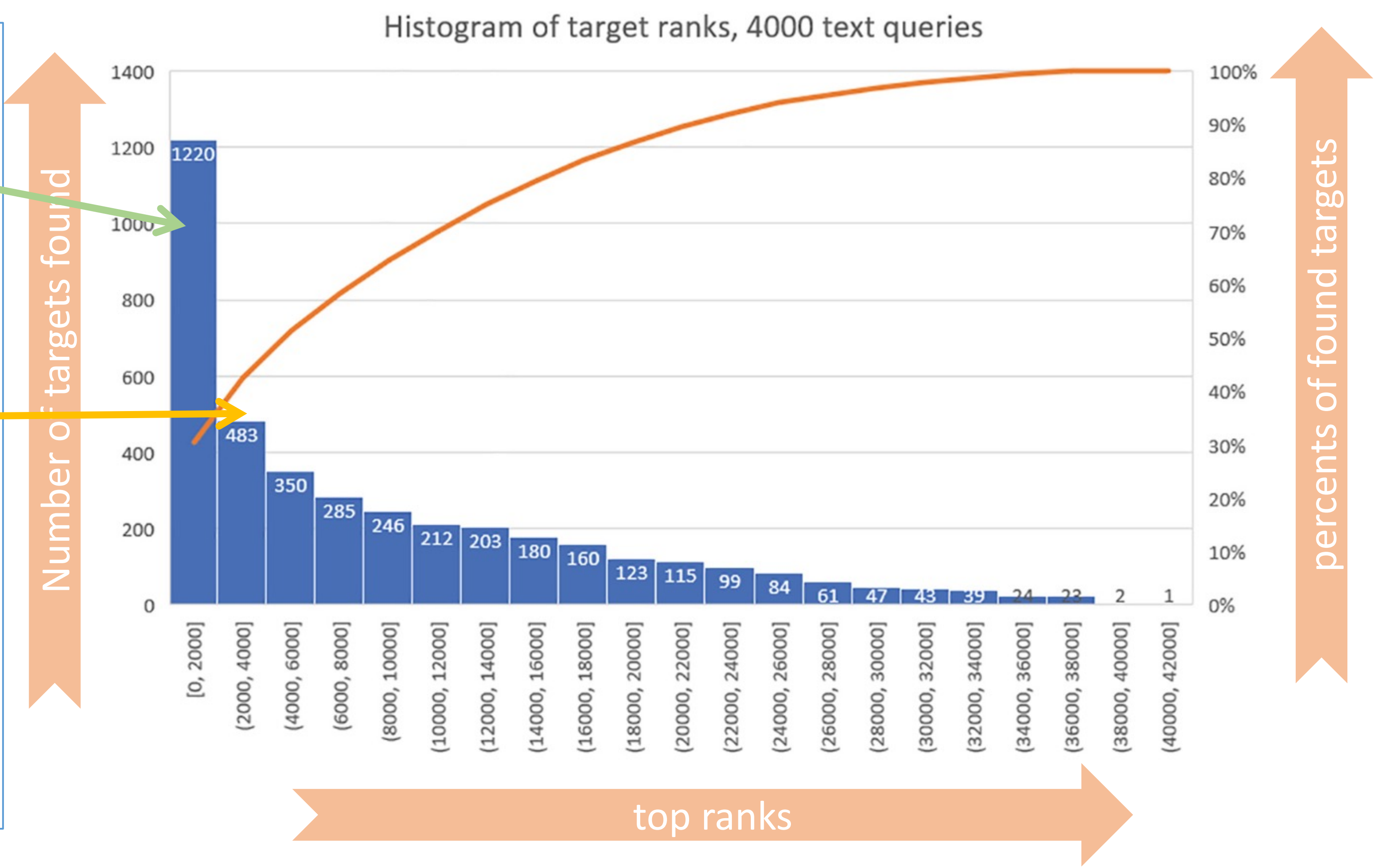
Plot for all types of queries



Ranks for ClipCap, Novice, and VBS Expert text queries for 100 target images.

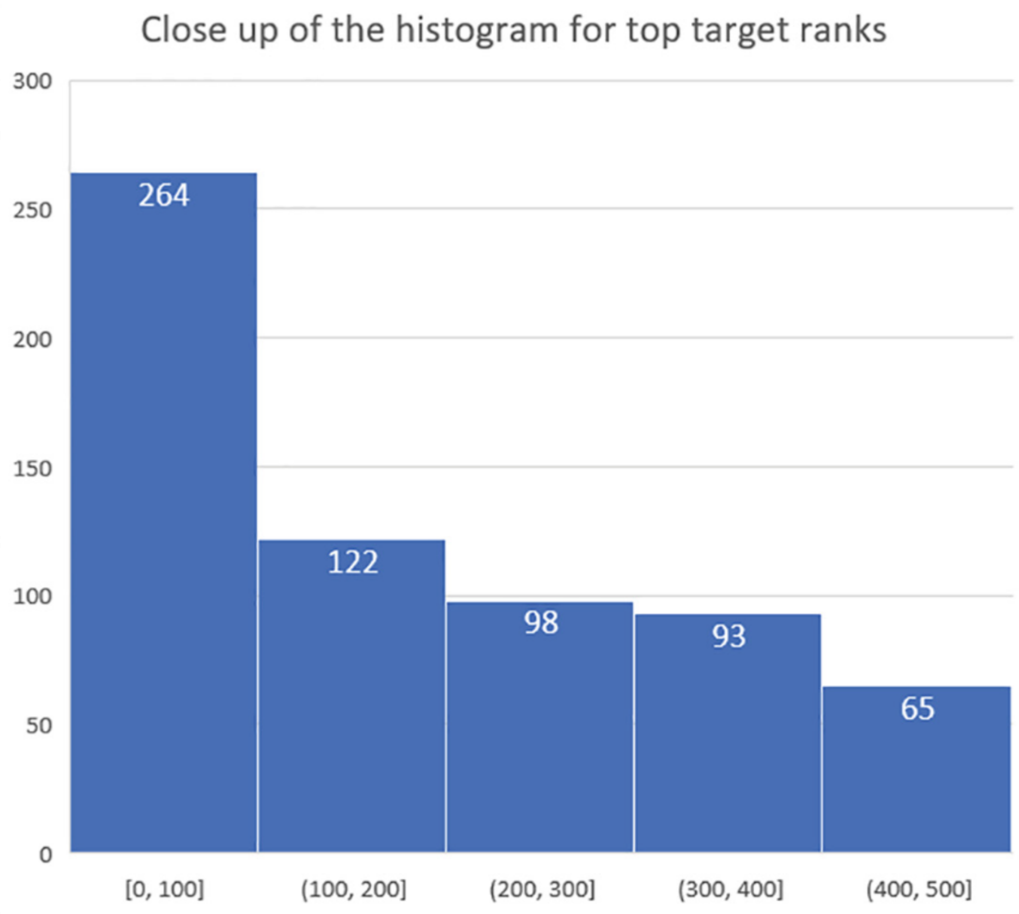
KIS for ClipCap queries

- There are 4000 ClipCap queries
  - In the top_rank [0,2000]: only 1220/4000 = 30% found
  - In the top_rank [2000,4000]: 483/4000 =12 % found.
    - Top_rank 4000 consists of [0,2000] and [2000,4000]: 30%+12% = 42%
  - In the top_rank i+2000 is #top rank i + top_rank[i+2000]



KIS for ClipCap queries

- Look at only small top_rank[0,500]
  - In the top 100, 264/4000=6.6% items found
  - In the top 200, (264+122)/4000=9.65% items found



Thank You
For Your Attention