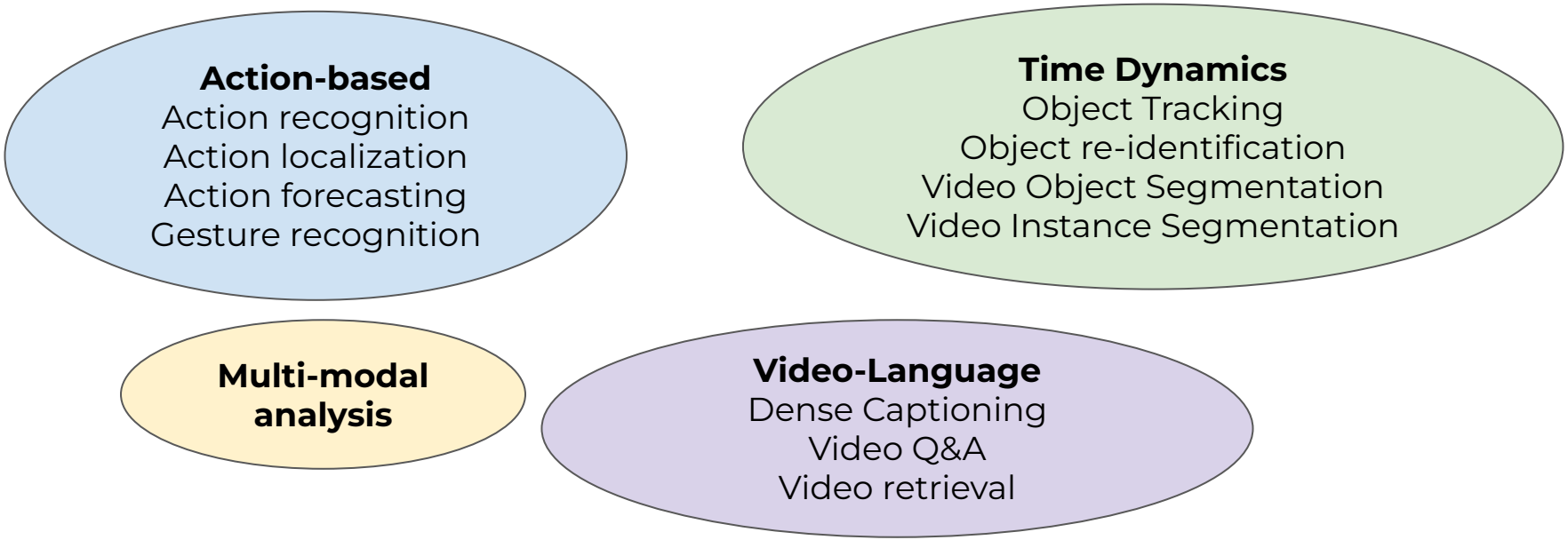
Limited Data Challenges in Video Understanding

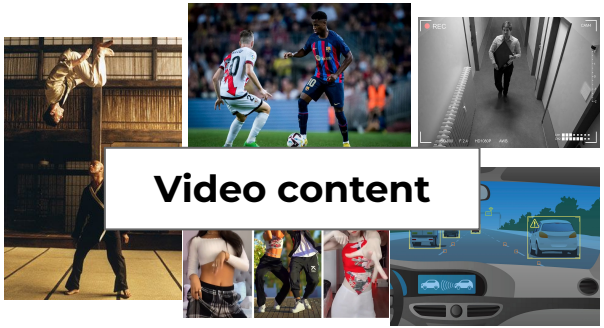
September 13, 2023

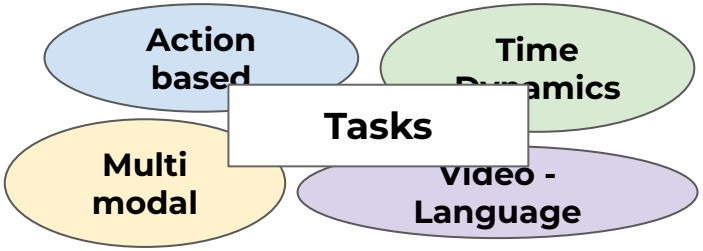
Video content

Video understanding tasks

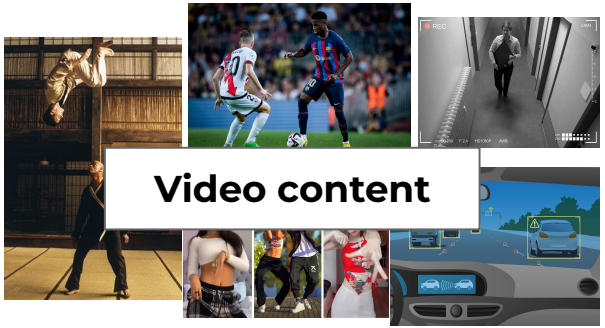


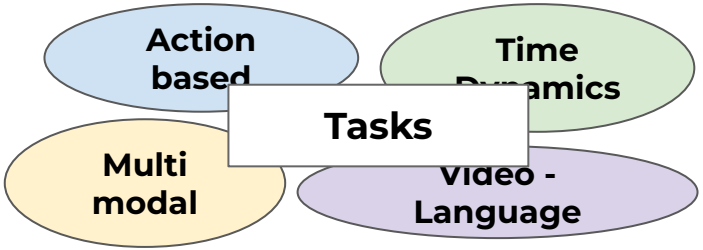
Needs to collect labeled data





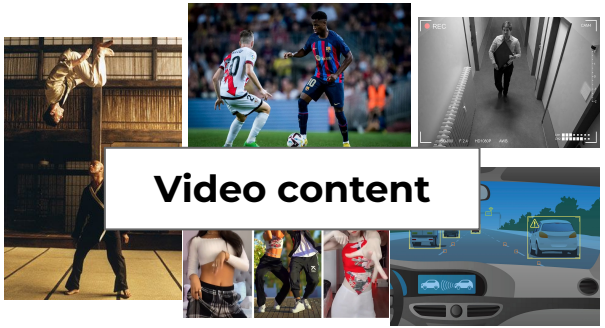
Needs to collect labeled data

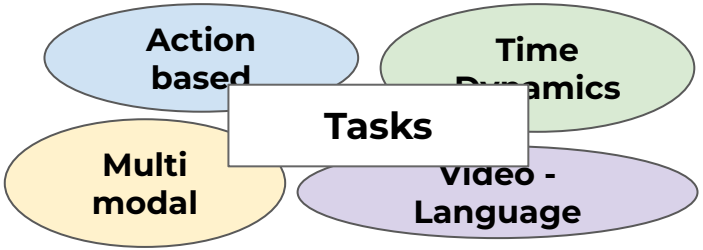




Collect labeled data → **Models**

Needs to collect labeled data

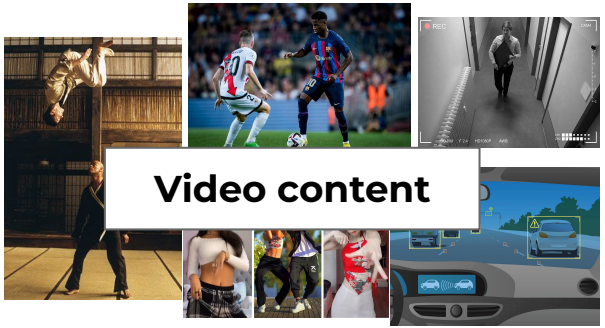


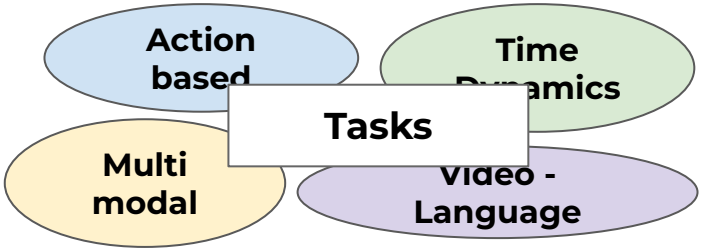


Collect labeled data

Models

Needs to collect labeled data





Limited labeled data

Collect labeled data

Models

Collecting large datasets is time-consuming

Kinetics-400

~850 hours of RGB videos
400 classes
videos: ~440 GB

Epic kitchens-100

~100 hours of RGB videos
97 verbs + 300 nouns
videos + annotations: ~740 GB

Ego4d

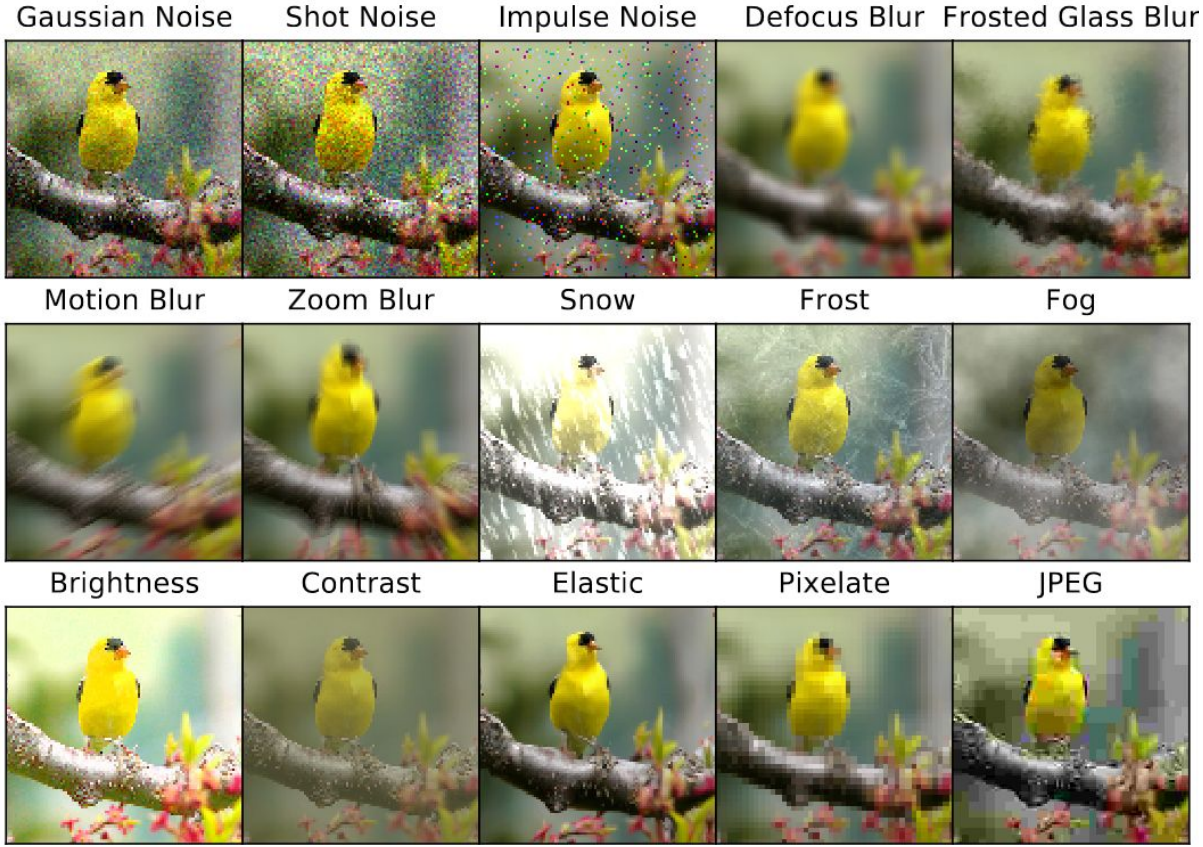
~3670 hours of RGB videos
1772 verbs + 4336 nouns
videos: 7TB / annotations: 2GB

Kinetics-400 (Kay *et al.*, 2017)
Epic-kitchens (Damen *et al.*, 2018)
Ego4d (Grauman *et al.*, 2022)

Collecting annotations at the pixel level

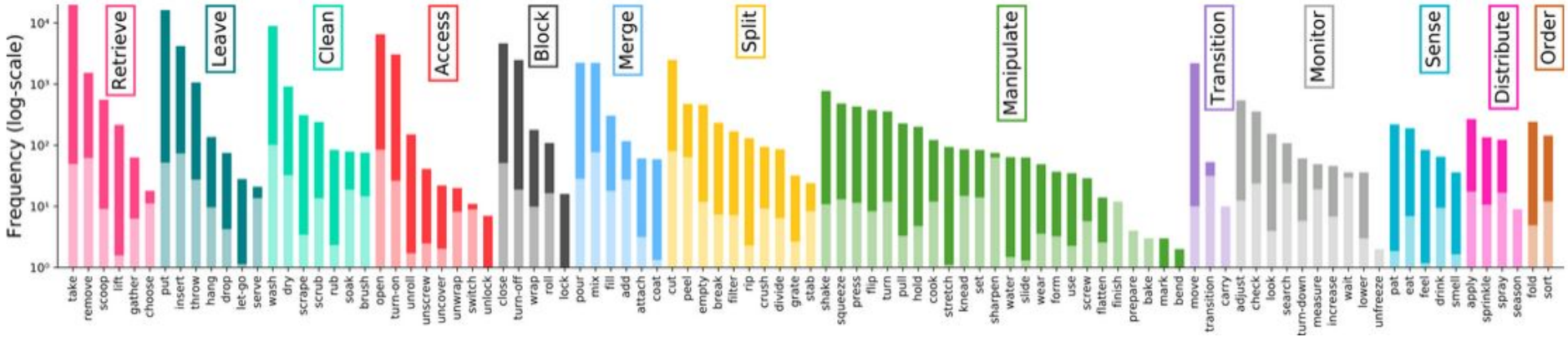
DAVIS (Pont-Tuset *et al.*, 2017)

Distribution shifts observed at test time



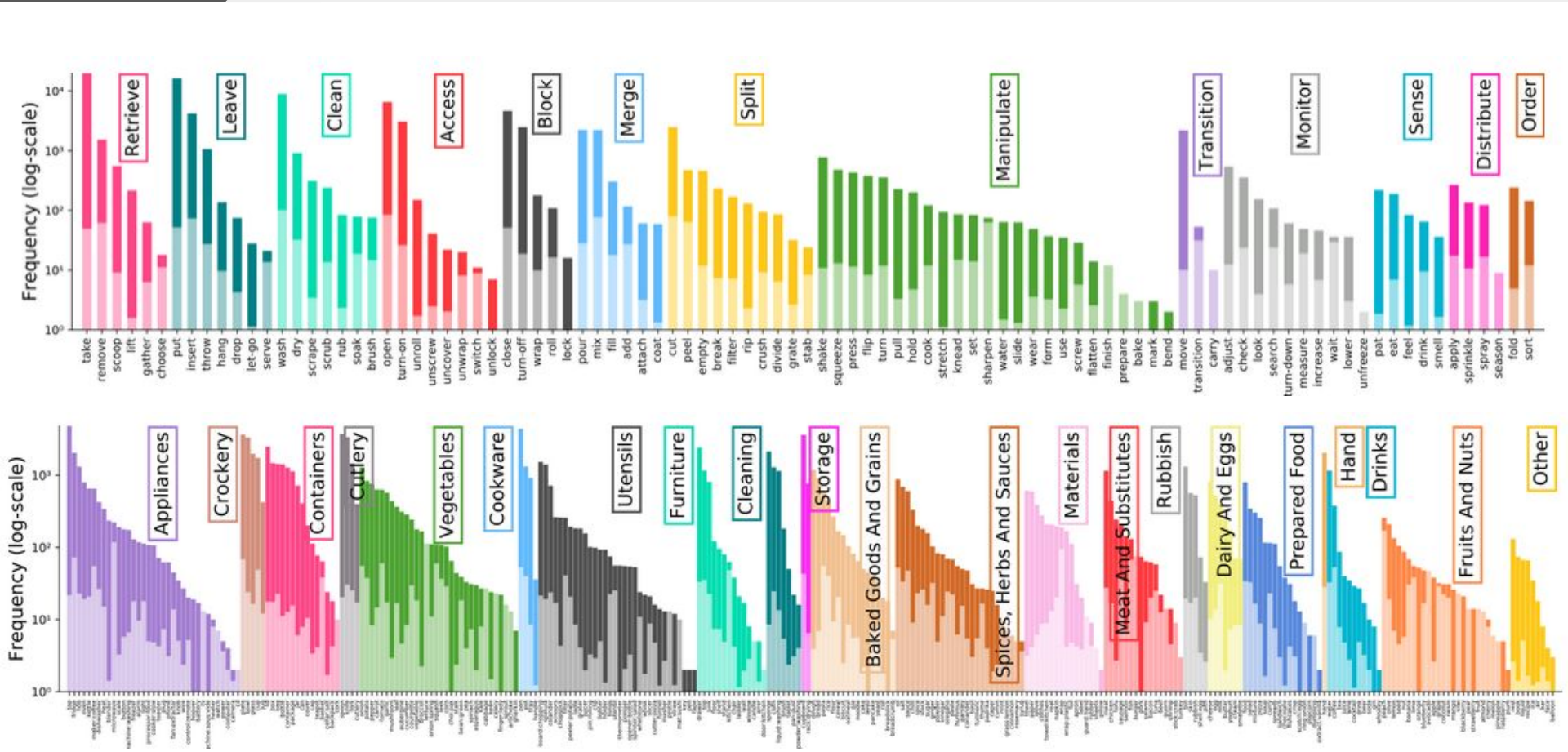
ImageNet-C (Hendrycks *et al.*, 2019)

Long tail problem



Epic-kitchens (Damen *et al.*, 2018)

Long tail problem



Epic-kitchens (Damen *et al.*, 2018)

Which learning paradigm to use?

	Small amount of data	Large amount of data
Mostly unlabeled	Zero-shot learning Manual annotations	Self Supervised Learning Weakly supervised learning Active learning
Mostly labeled	Few-shot learning Transfer learning Test-time training	Classical deep learning (CNN, transformers ...)

Which learning paradigm to use?

	Small amount of data	Large amount of data
Mostly unlabeled	Zero-shot learning Manual annotations	Self Supervised Learning Weakly supervised learning Active learning
Mostly labeled	Few-shot learning Transfer learning Test-time training	Classical deep learning (CNN, transformers ...)

Data Augmentation

Focusing on small amount of labeled data

	Small amount of data	Large amount of data
Mostly unlabeled	Zero-shot learning Manual annotations	Self Supervised Learning Weakly supervised learning Active learning
Mostly labeled	Few-shot learning Transfer learning Test-time training	Classical deep learning (CNN, transformers ...)

Outline

- Test time training for Video Object Segmentation
  - Test-time training for matching-based video object segmentation,
    Juliette Bertrand, *Giorgos Kordopatis Zilos, Yannis Kalantidis, Giorgos Tolias*

- Few shot learning for action recognition
  - Rethinking matching-based few-shot action recognition,
    Juliette Bertrand, *Yannis Kalantidis, Giorgos Tolias*

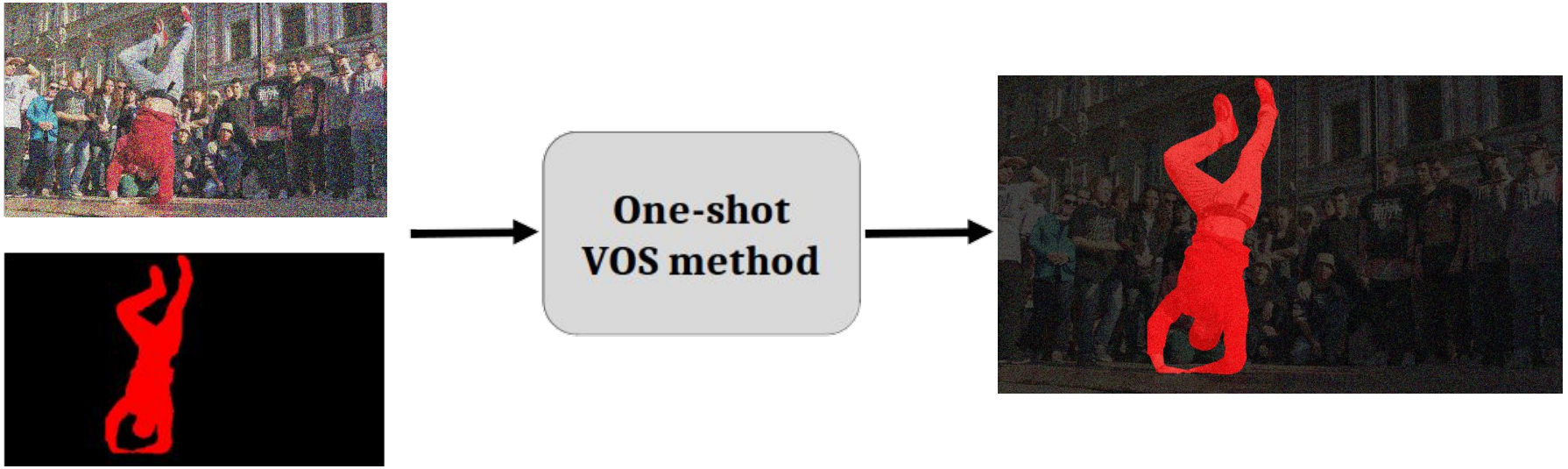
Test-time training for Video Object segmentation

One-shot Video Object Segmentation (VOS)

Given a segmentation mask of one or more target objects in a video

Segment target objects across all video frames in time

Can we keep the performance under distribution shift?

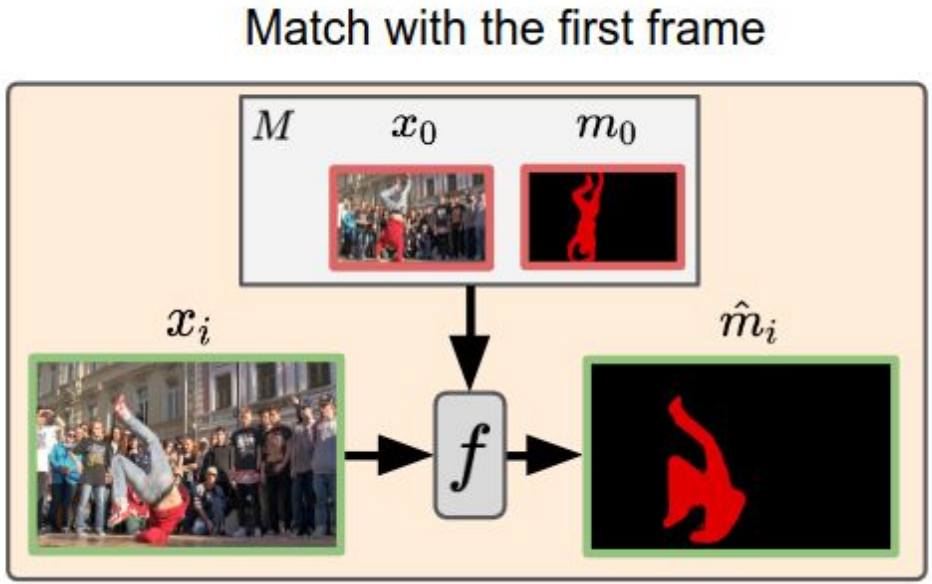


Test-time-training strategy tailored for VOS:

- For each **test video** example, **learn to adapt** using only the first annotated frame

Matching-based methods

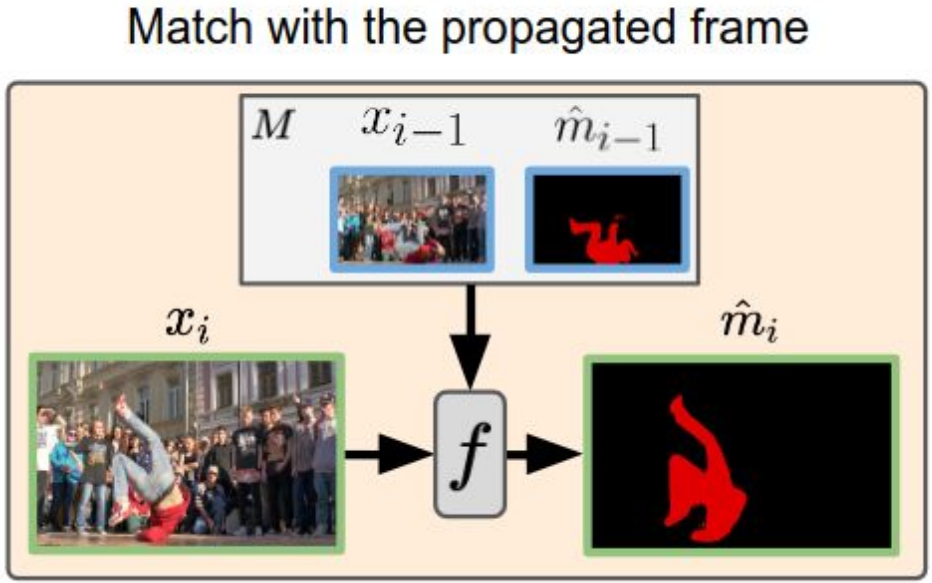
- Match each **current** frame with **past** frame(s)



x_i: arbitrary video frame
$\hat{m}_i$: predicted mask
m_0: ground-truth mask of the first frame
f: matching function
M: memory with frames and masks

Matching-based methods

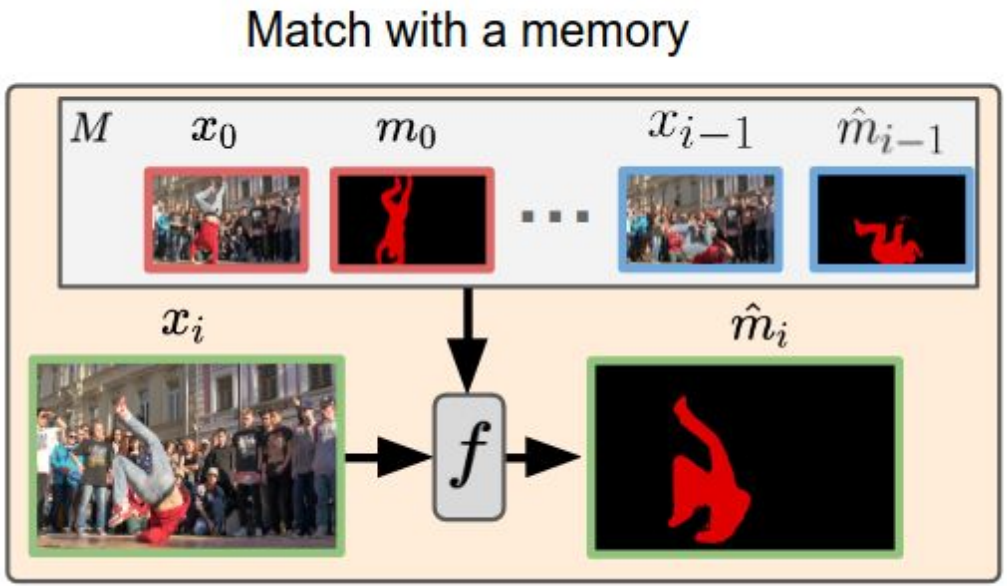
- Match each **current** frame with **past** frame(s)



x_i: arbitrary video frame
$\hat{m}_i$: predicted mask
m_0: ground-truth mask of the first frame
f: matching function
M: memory with frames and masks

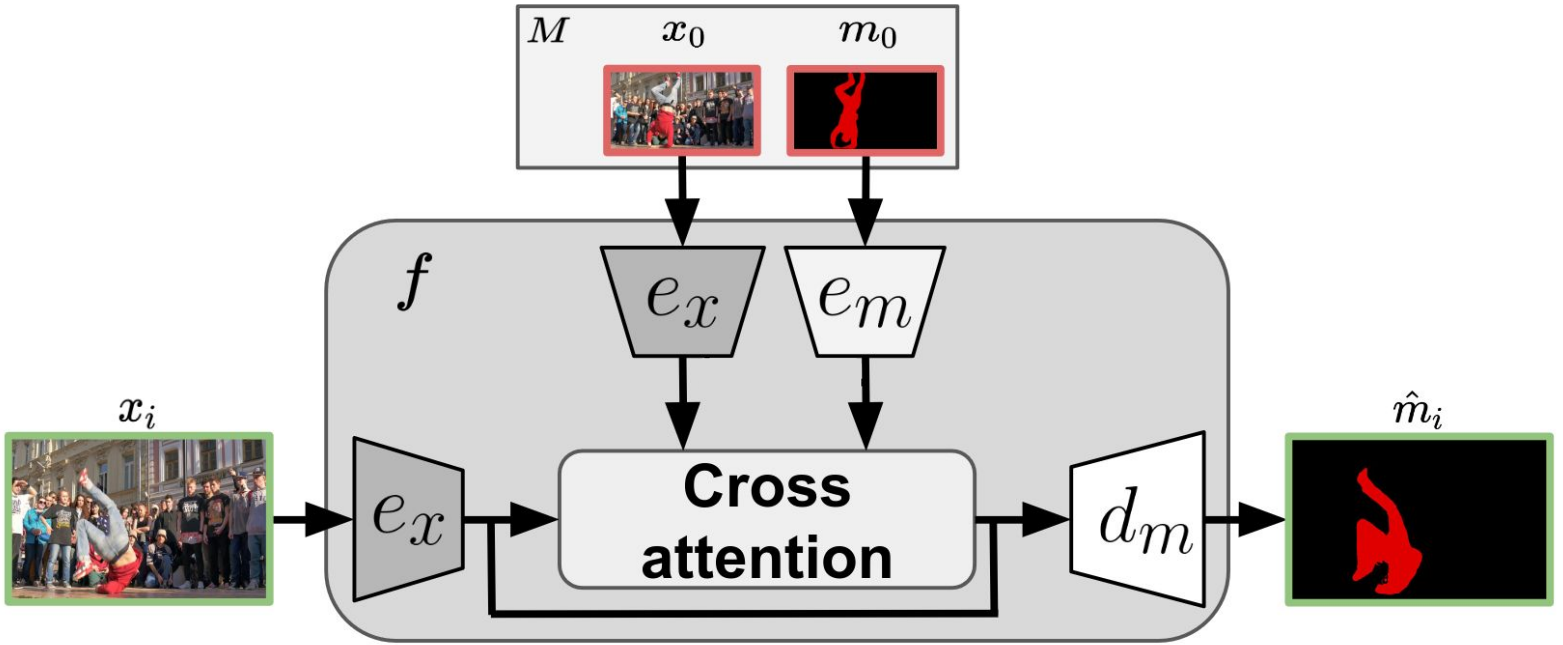
Matching-based methods

- Match each **current** frame with **past** frame(s)
  - Memory of (frame, mask) tuples



x_i: arbitrary video frame
$\hat{m}_i$: predicted mask
m_0: ground-truth mask of the first frame
f: matching function
M: memory with frames and masks

Space-Time Correspondences Networks (STCN / XMem)

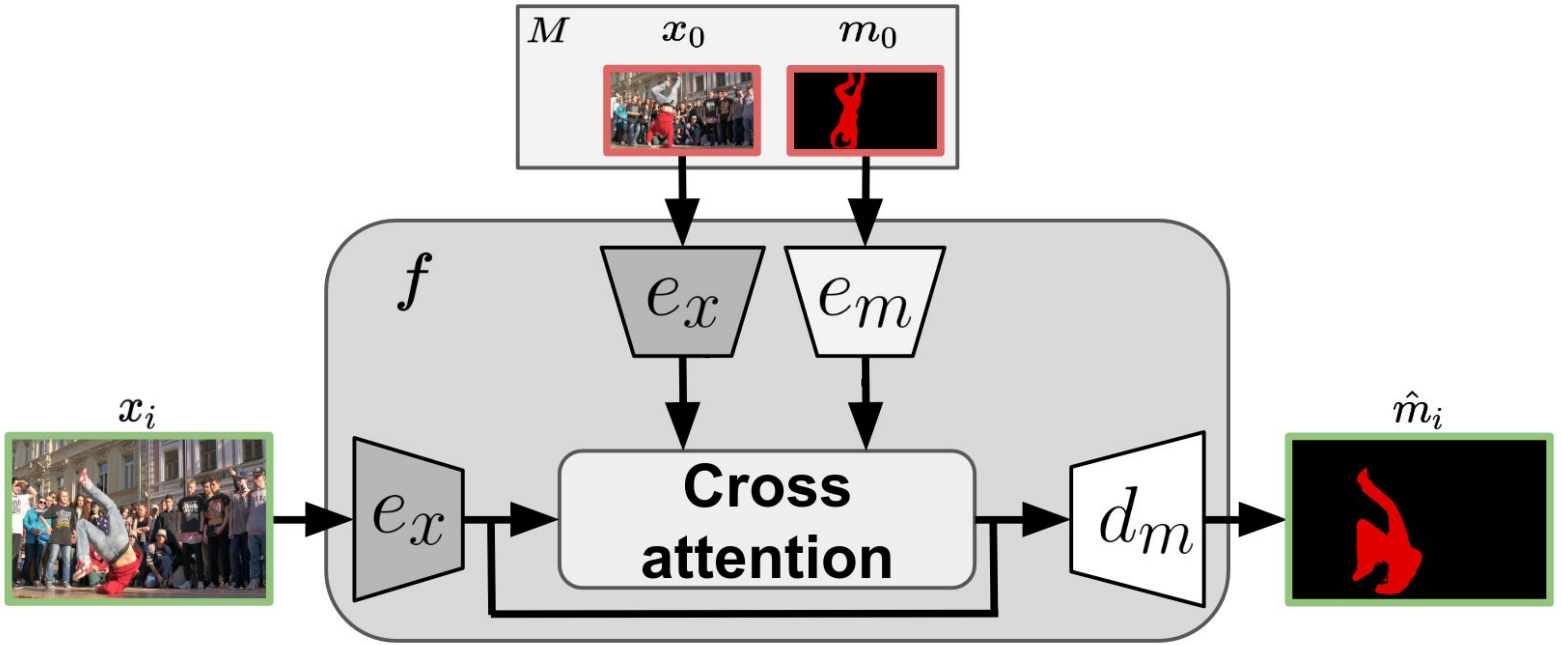


e_x : frame encoder network
e_m: mask encoder network
d_m: decoder network

STCN (Cheng *et al.*, 2021)

XMem (Cheng *et al.*, 2022)

Space-Time Correspondences Networks (STCN / XMem)

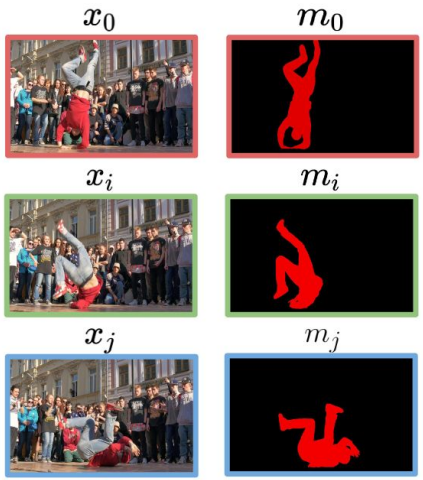


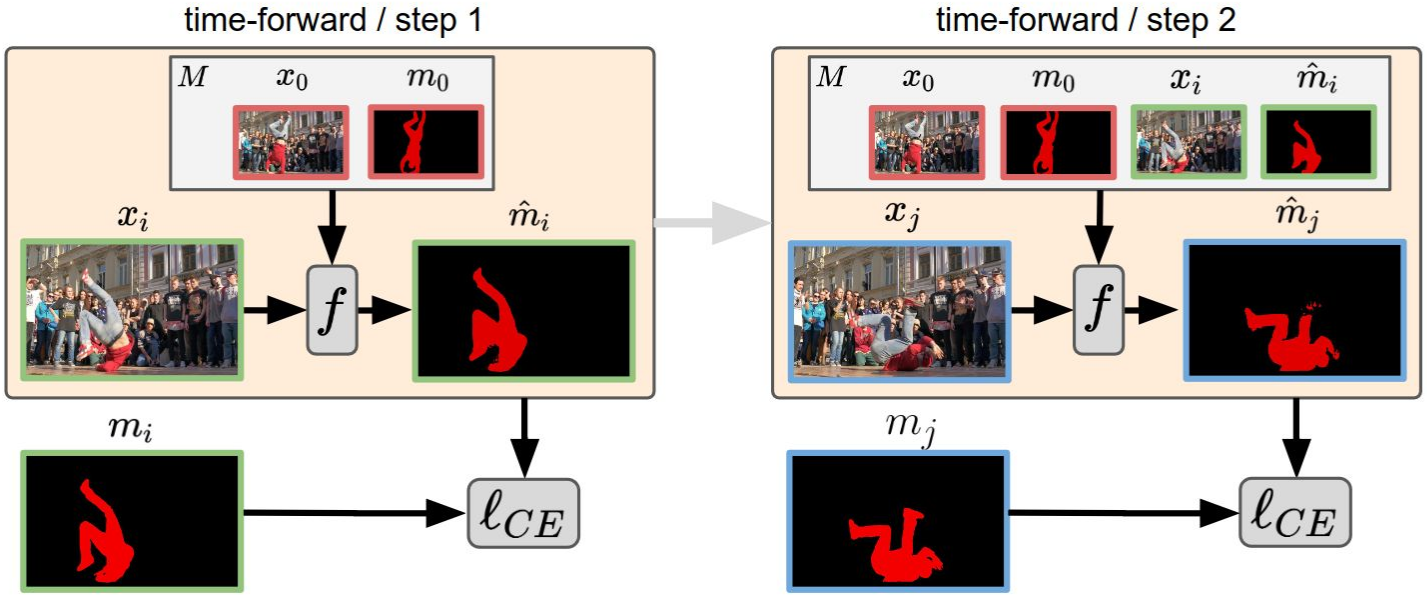
e_x : frame encoder network
e_m: mask encoder network
d_m: decoder network

STCN (Cheng *et al.*, 2021)

XMem (Cheng *et al.*, 2022)

Supervised training





Pretraining on static images

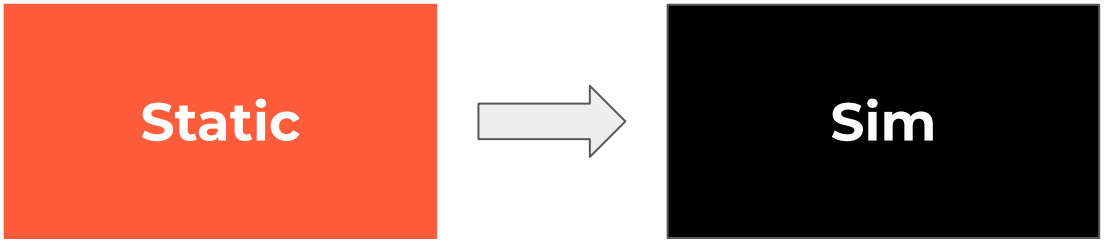
- Synthetic sequences generated by **deforming static images** (DUTS + ECSSD + FSS-1000 + HRSOD + BIG)

Static

DUTS (Wang *et al.*, 2017)
ECSSD (Shi *et al.*, 2015)
FSS-1000 (Li *et al.*, 2020)
HRSOD (Zeng *et al.*, 2019)
BIG (Cheng *et al.*, 2020)

Pretraining on synthetic videos

- Synthetic videos generated by **animating 3D models** from ShapeNet with an open-source **rendering** engine (BL30K)



BL30K (Cheng *et al.*, 2021)

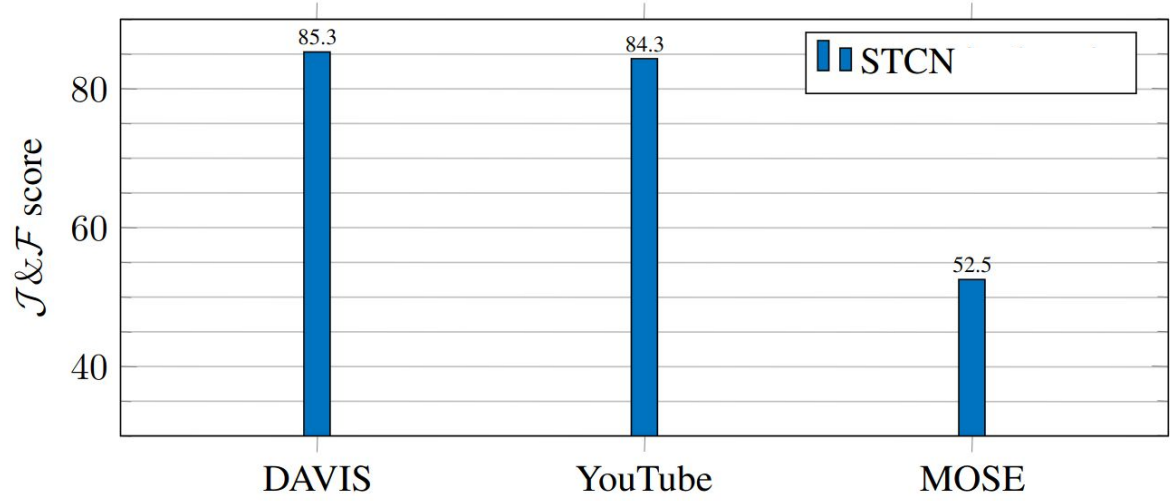
Training on real videos

- **Real** video recordings (DAVIS + YouTube-VOS)

DAVIS (Pont-Tuset *et al.*, 2017)

YouTube-VOS (Xu *et al.*, 2018)

Performance



Metric: $\mathcal{J}\&\mathcal{F}$ score

- $\mathcal{J}$: region similarity
- $\mathcal{F}$: contour accuracy

DAVIS (Pont-Tuset *et al.*, 2017)

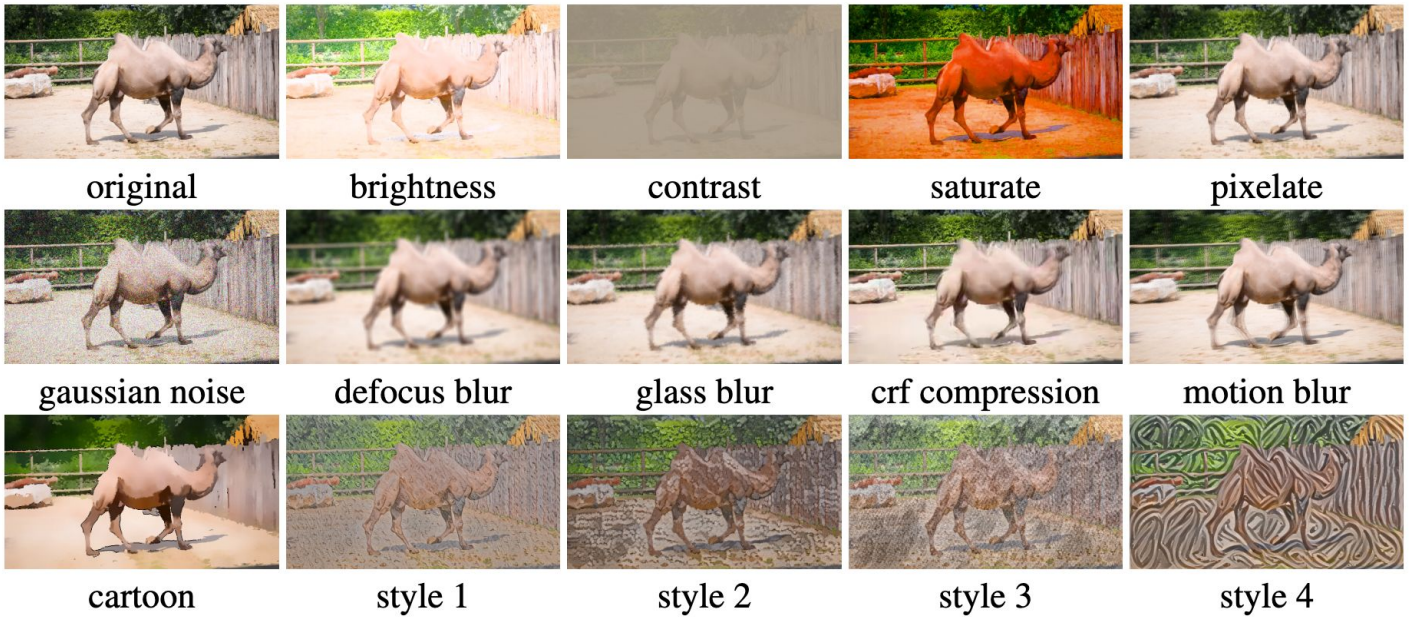
YouTube-VOS (Xu *et al.*, 2018)

MOSE (Ding *et al.*, 2023)

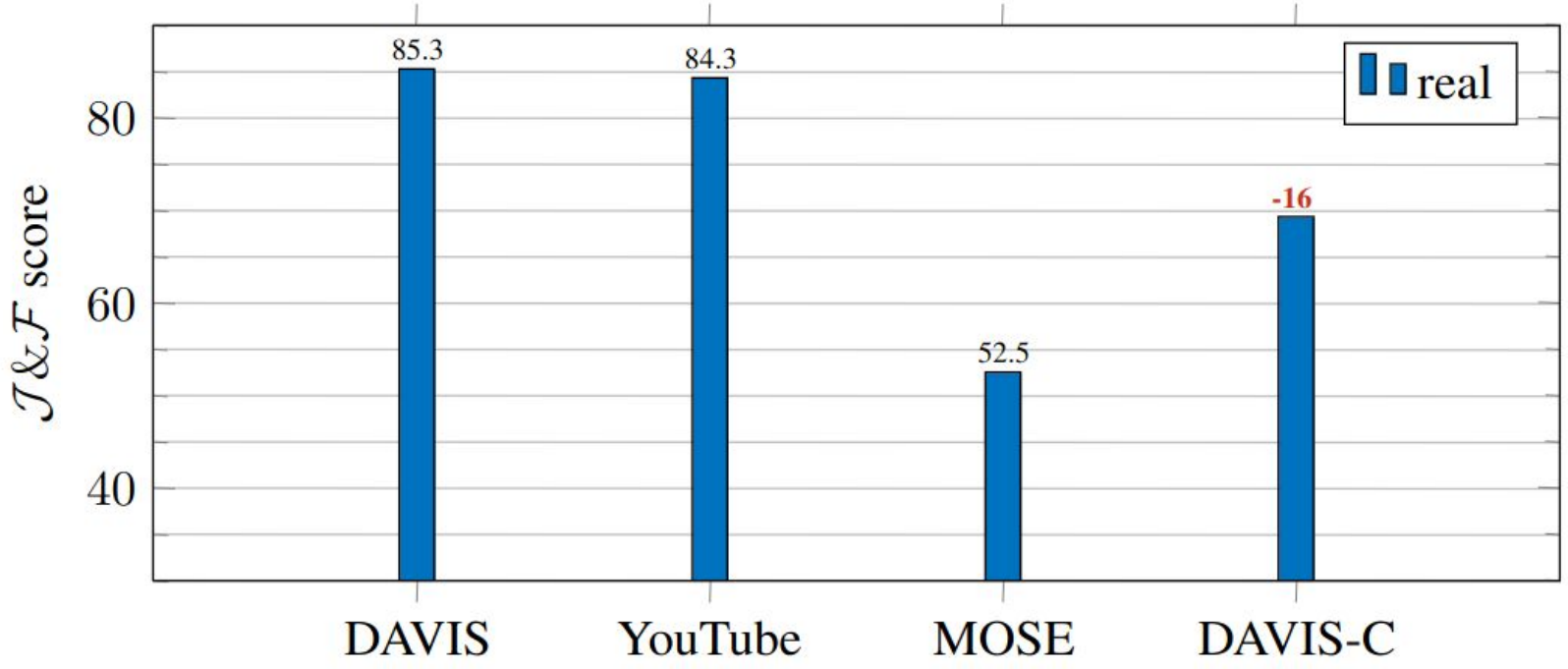
DAVIS-C dataset

14 corruptions and style changes

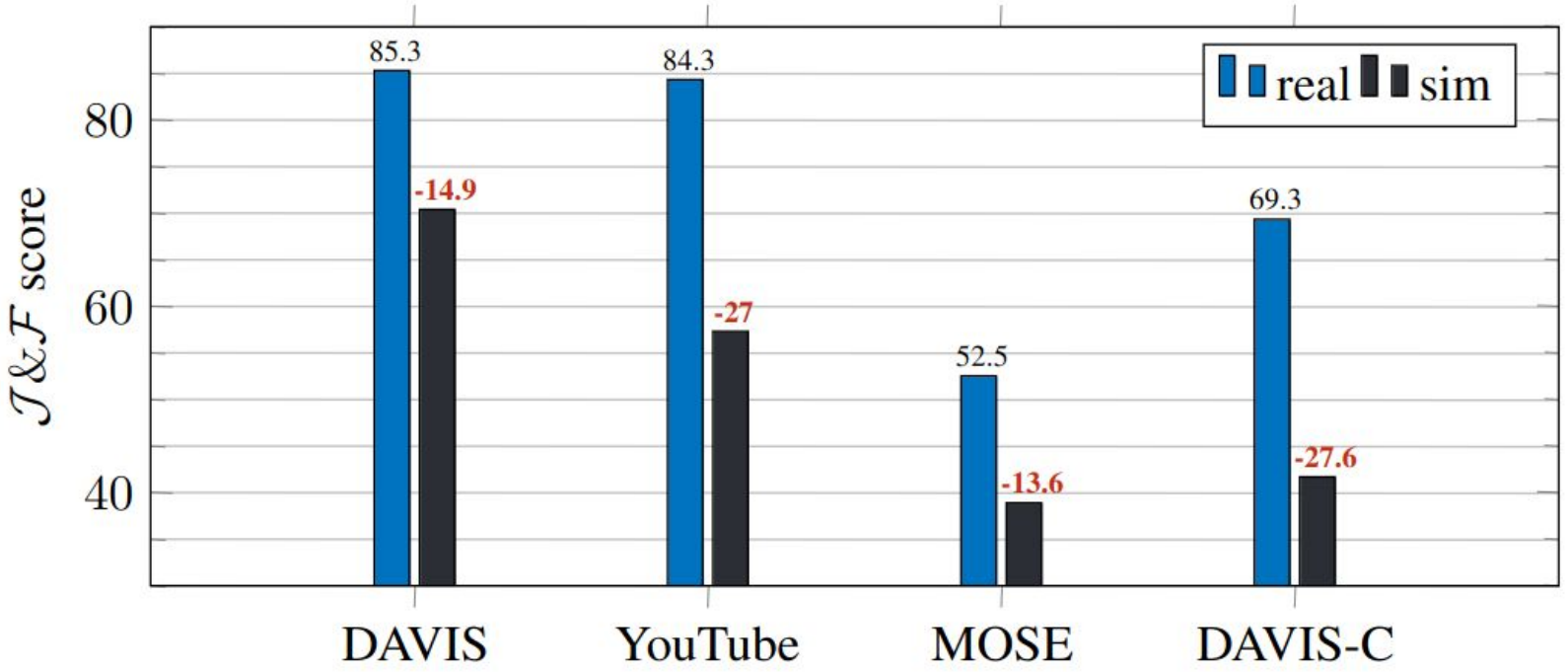
3 strengths (*low*, ***medium***, *high*)



Performance under extreme distribution shifts



Performance with no real-video seen during training

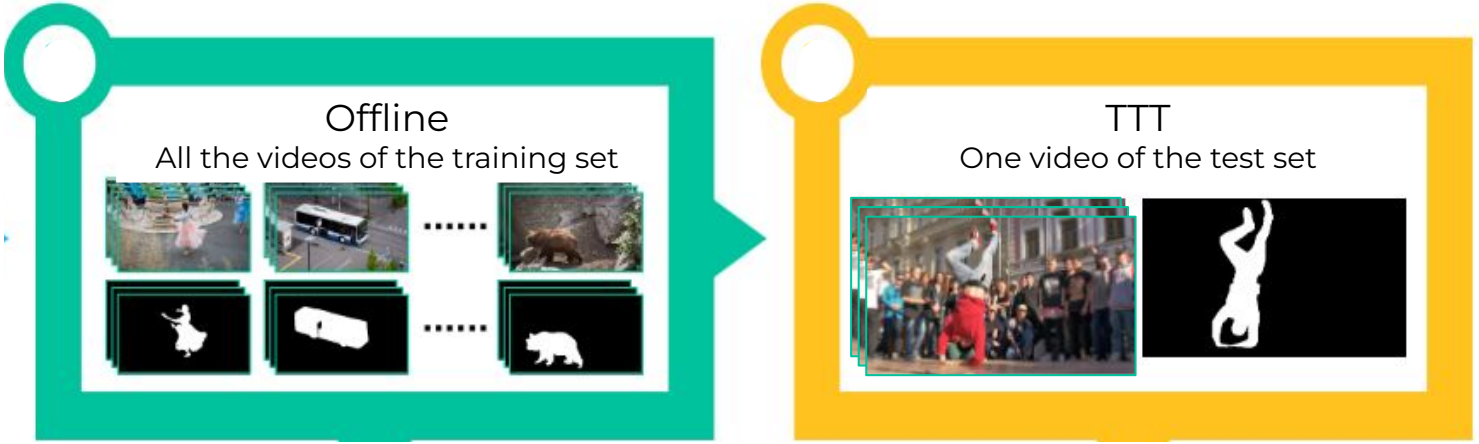


Problem

Can we keep the **performance** under **distribution shift**?

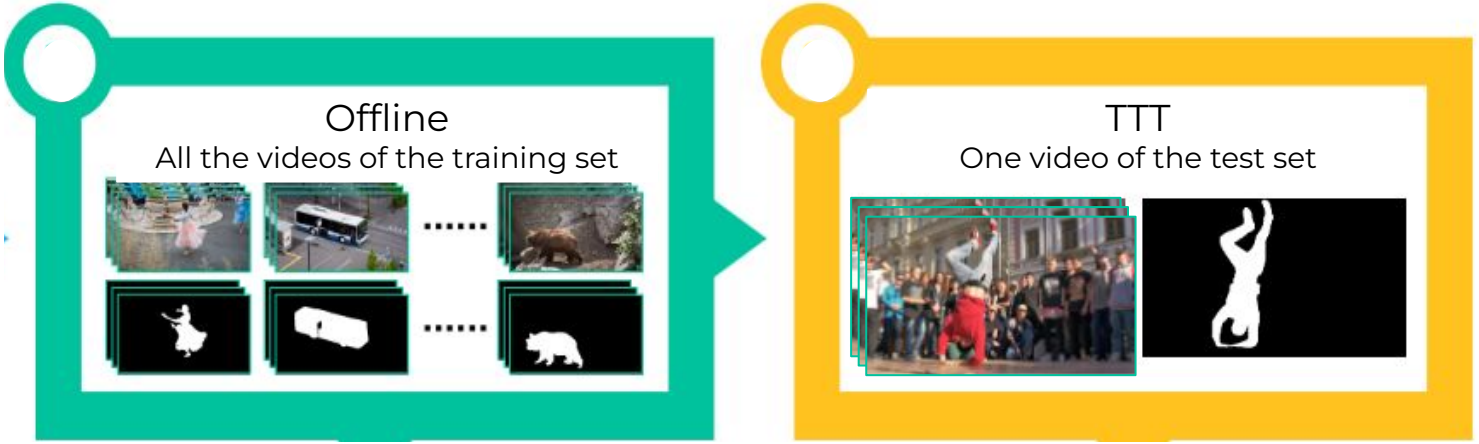
Can we achieve similar **performance** while training only on **synthetic data** ?

Test-time training (TTT)



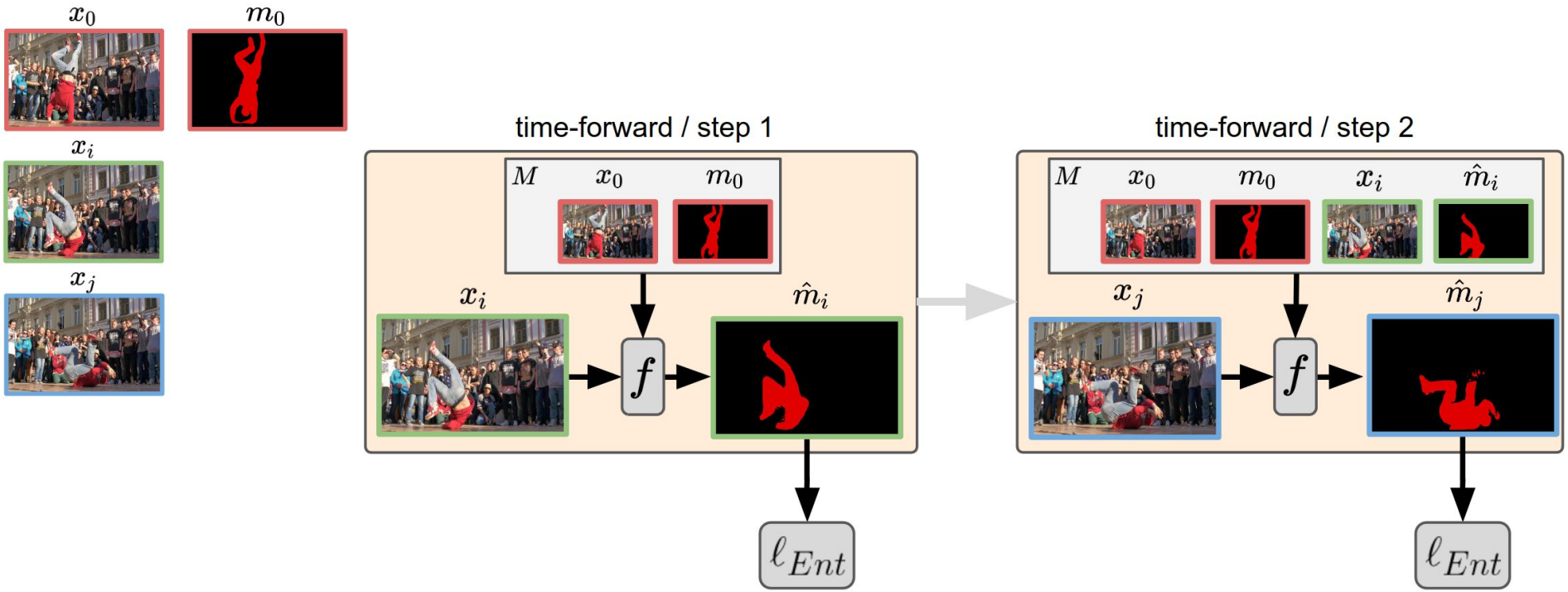
- Build a TTT scheme tailored for **matching-based methods**
  - **Train** using only **one test** example
  - Only the **first frame** is annotated
  - Make use of the **temporal dimension**

Test-time training (TTT)



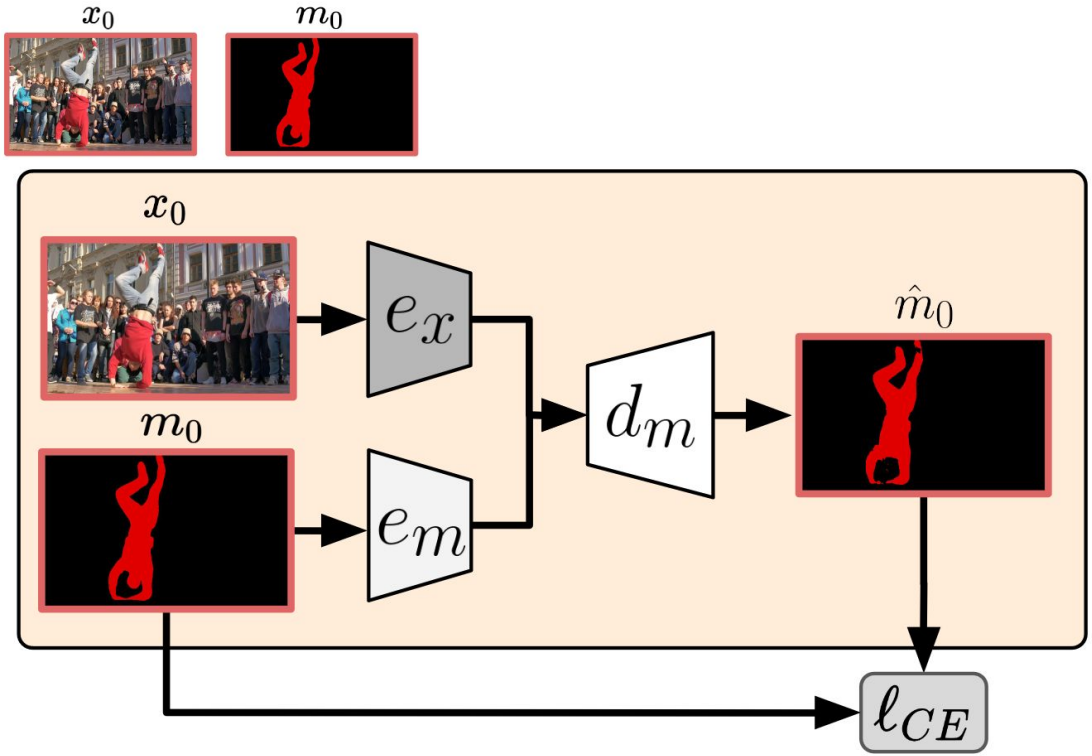
- 3 different methods:
  - tt-Ent: entropy minimization inspired by TTT trends
  - tt-AE: auto-encoder on the first frame (no use of the temporal dimension)
  - **tt-MCC: mask consistency (use of the temporal dimension)**

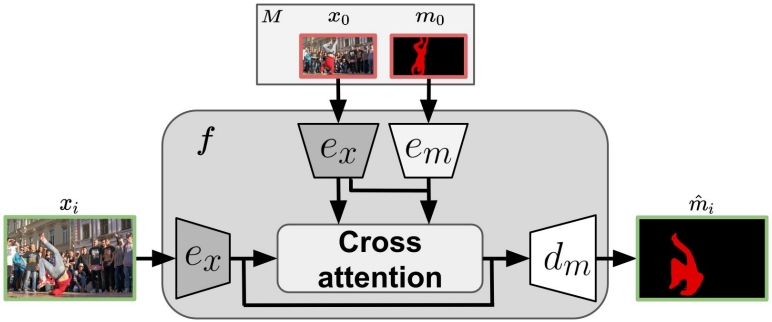
Test Time Entropy (tt-Ent)



- Baseline from image-classification
- Limited improvement

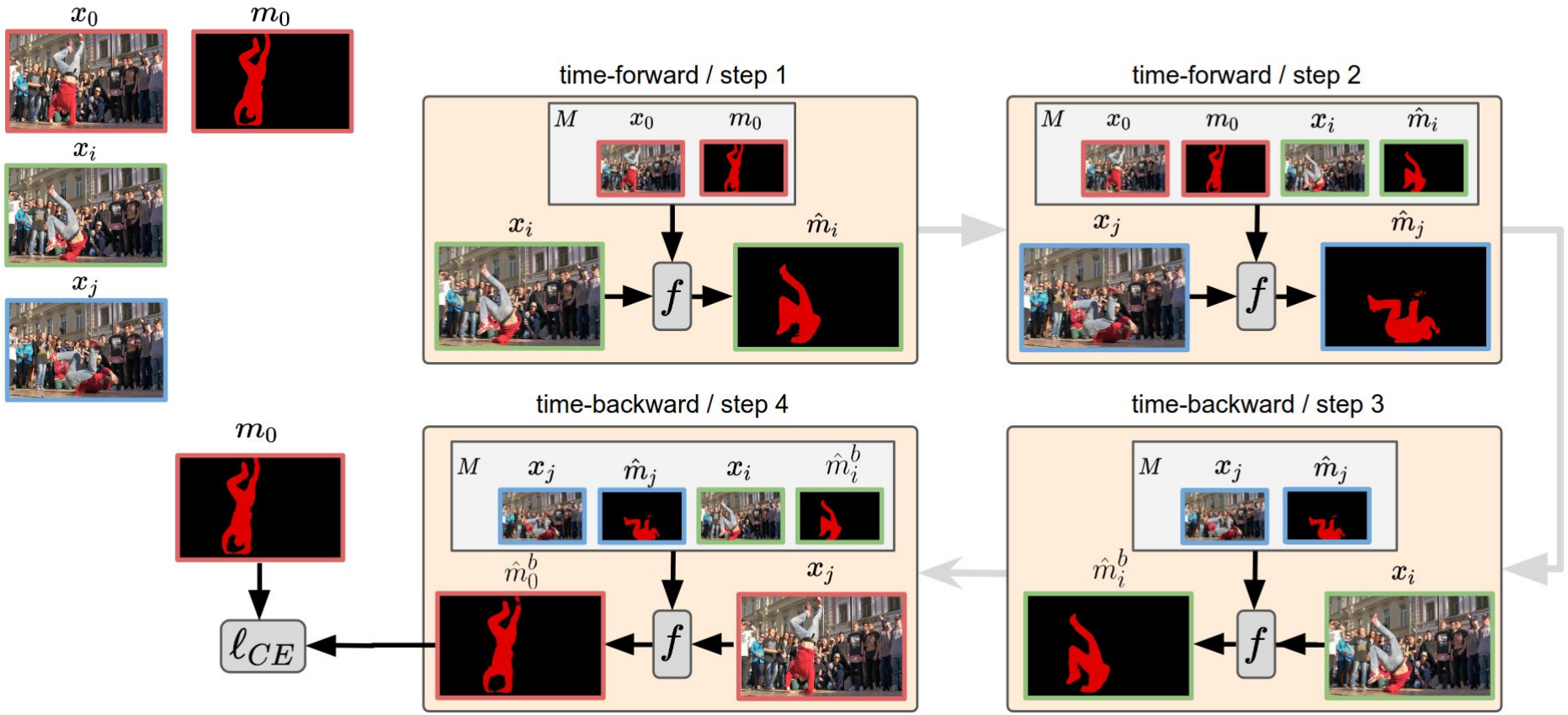
Test Time Auto-Encoder (tt-AE)



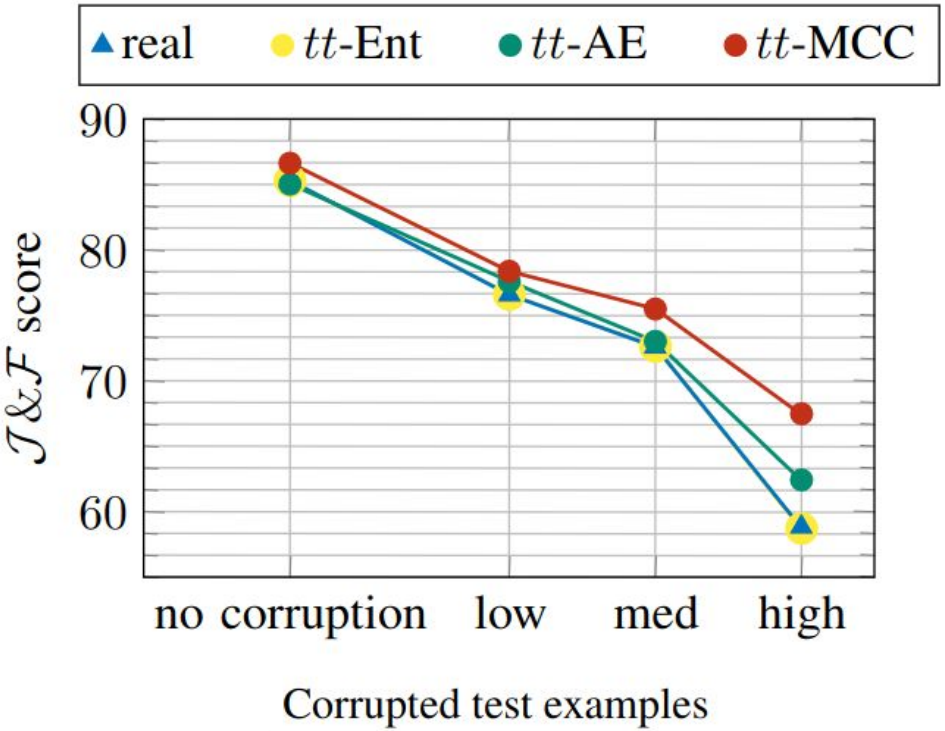


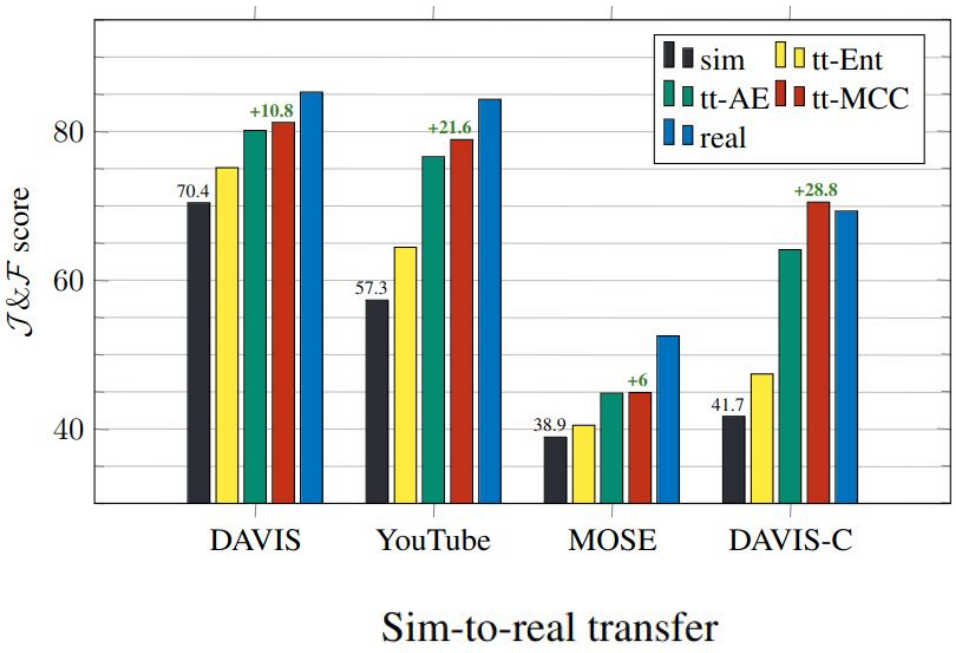
- Noticeable improvement
- Leverage the first frame only

Test Time Mask Cycle Consistency (tt-mCC)



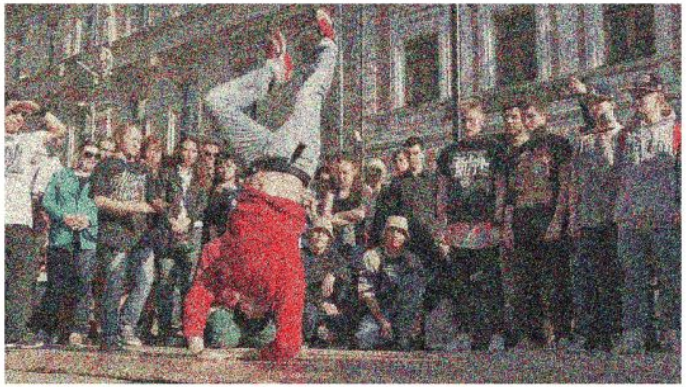
Quantitative performance





Qualitative performance: distribution shift

RGB



GroundTruth

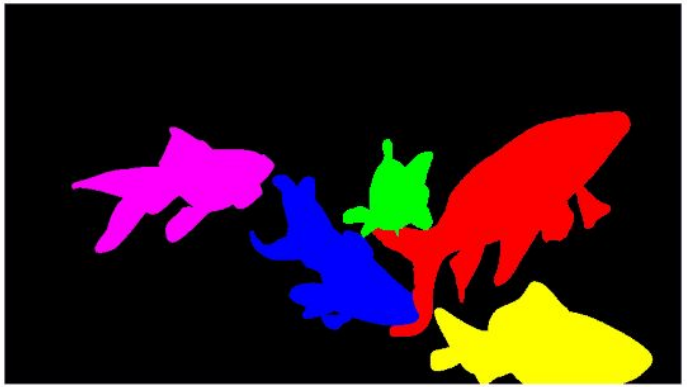
STCN real (J&F=32.25)

tt-MCC (J&F=68.52)

Qualitative performance: sim-to-real

RGB

GroundTruth



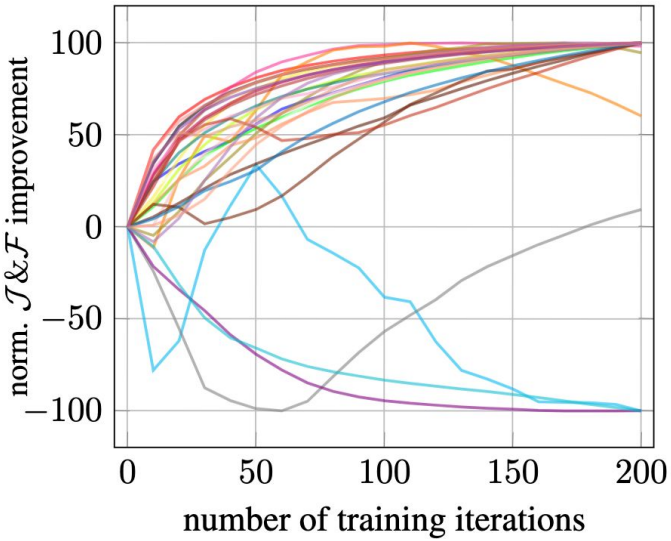
STCN sim (J&F=47.4)

tt-MCC (J&F=90.4)

Take away and future directions

Key take-away

- a new TTT scheme for VOS matching-based method
- tt-MCC can recover the bulk of the performance for the test video while training with no real videos
- tt-MCC improves the performance even under extreme distribution shifts



Future direction

- how to select the number of iterations for training at test time?

Few-shot action recognition

Few-shot action recognition (FSAR)

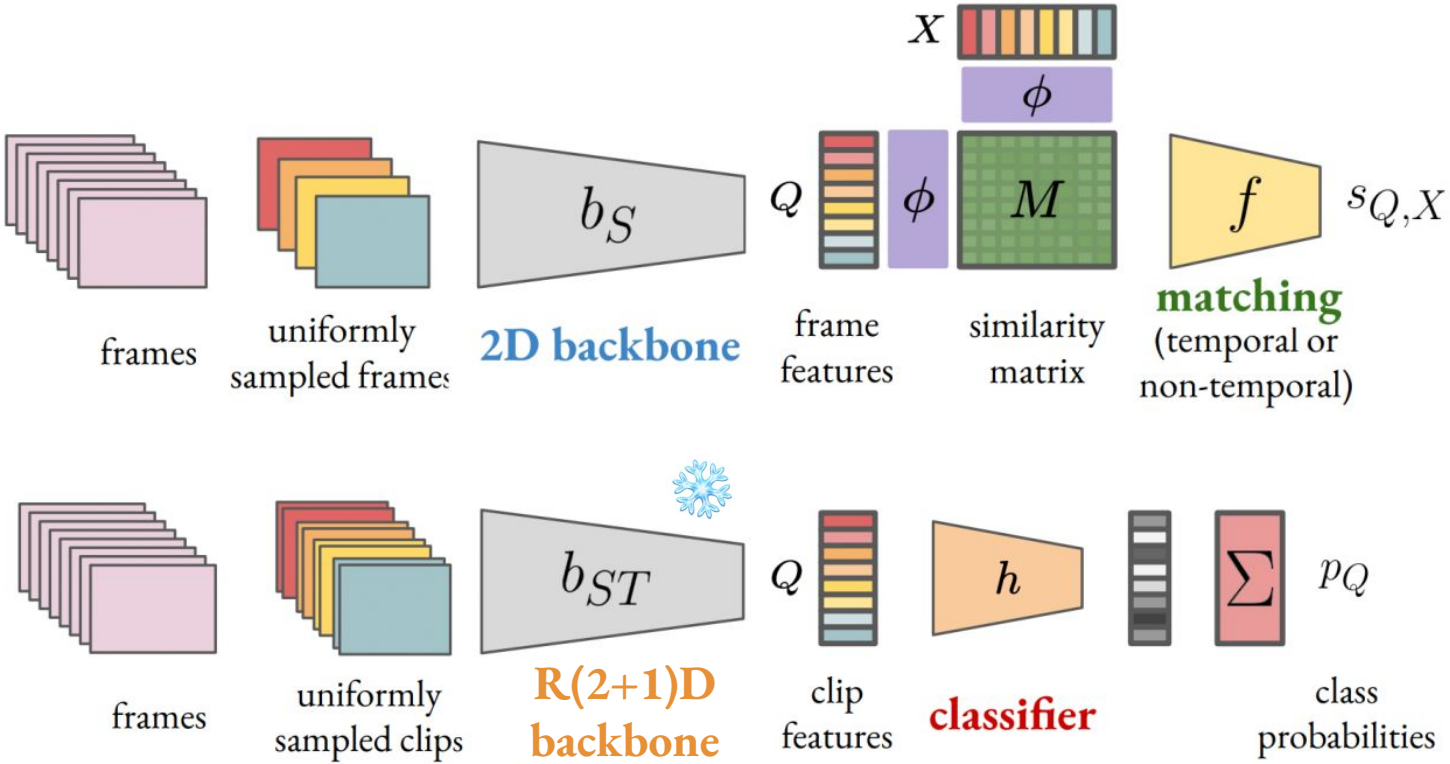
Dropping something **onto** something

Pretending to put something **into** something

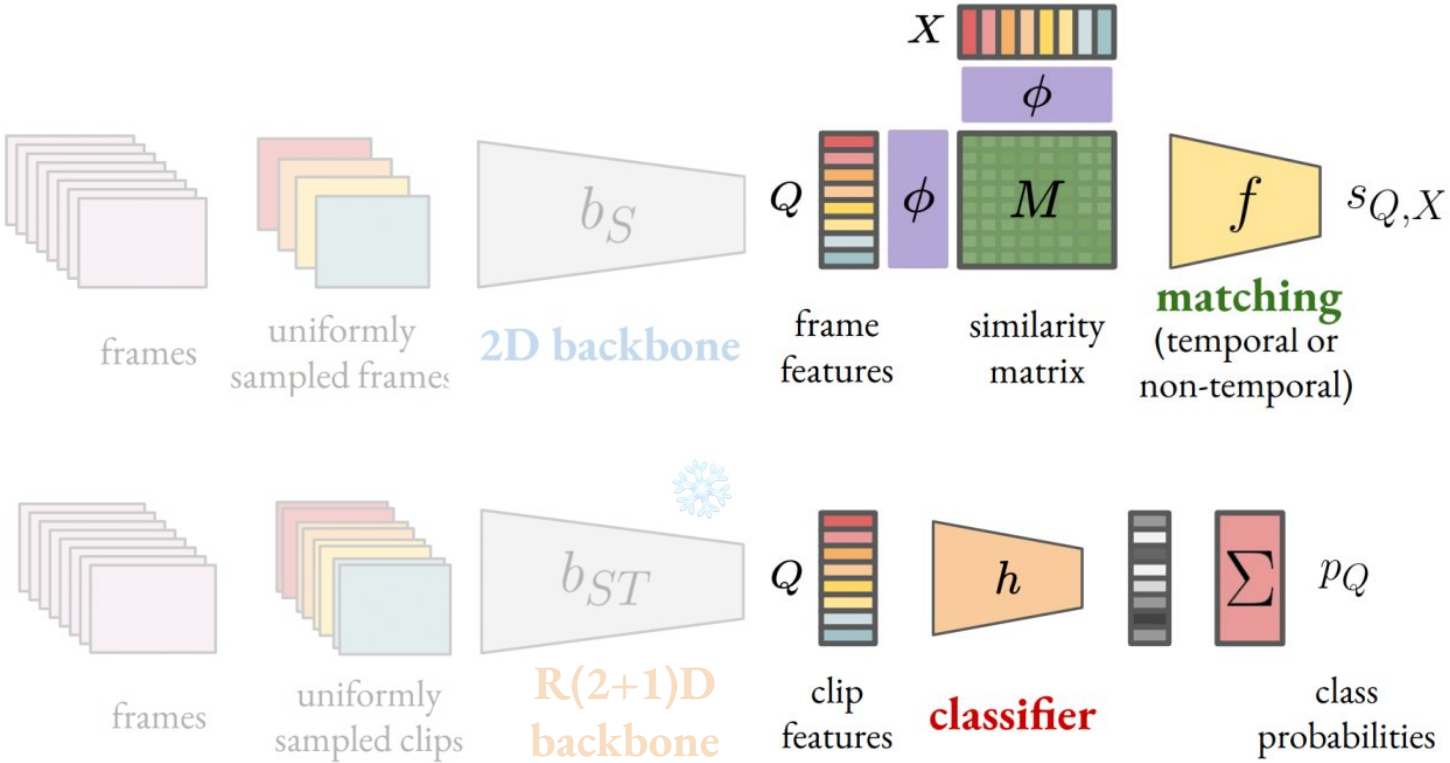
?

Action recognition (or classification) when we have **few annotated** examples per class

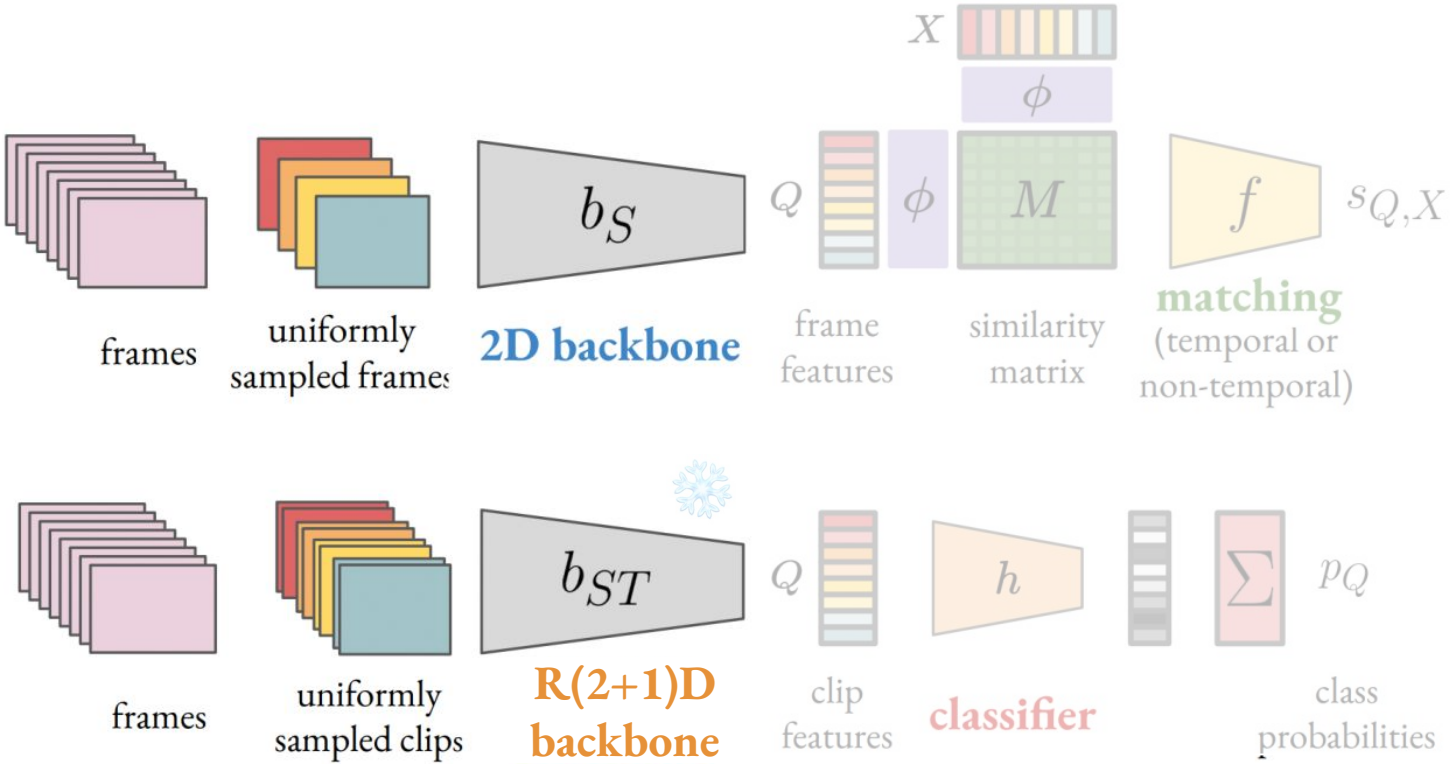
Matching-based vs classifier-based methods



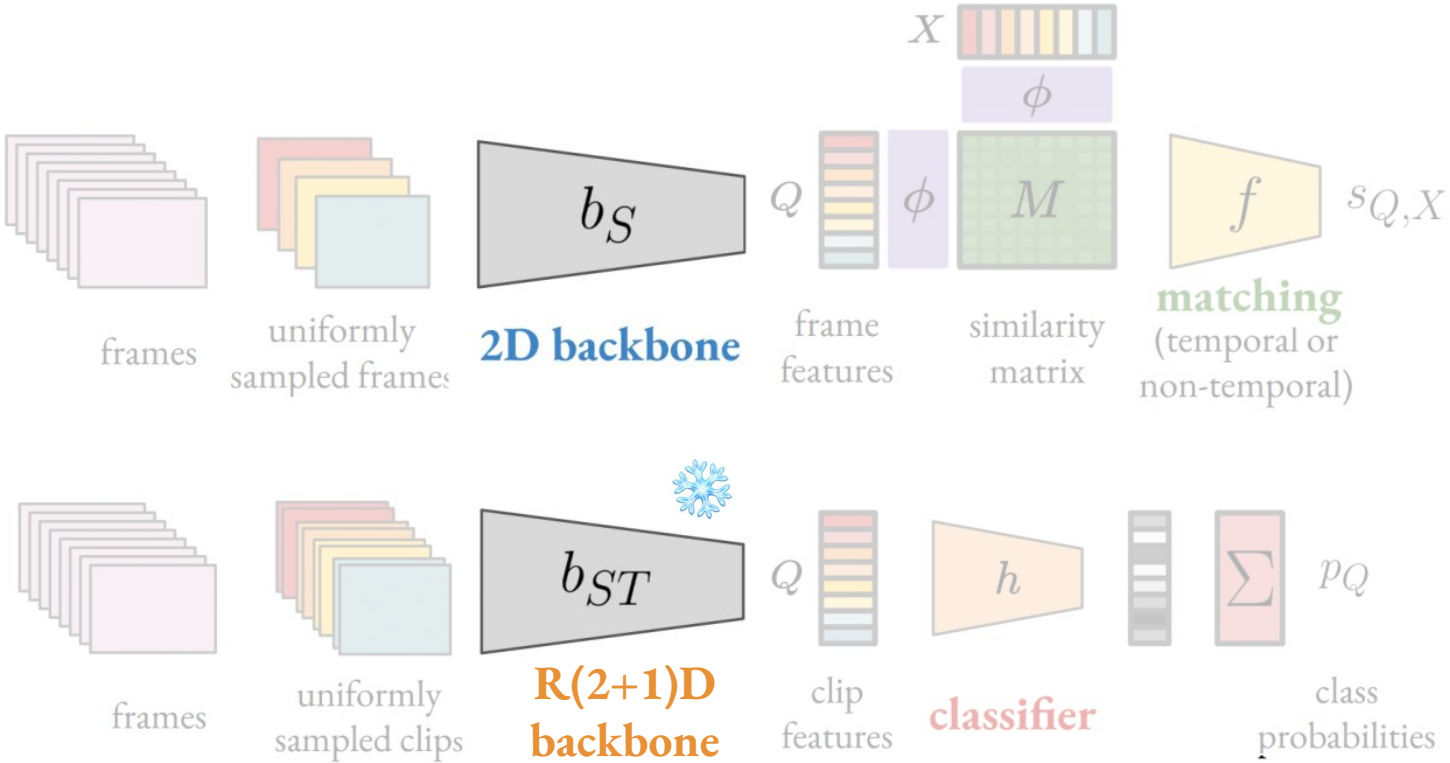
Matching-based vs classifier-based methods



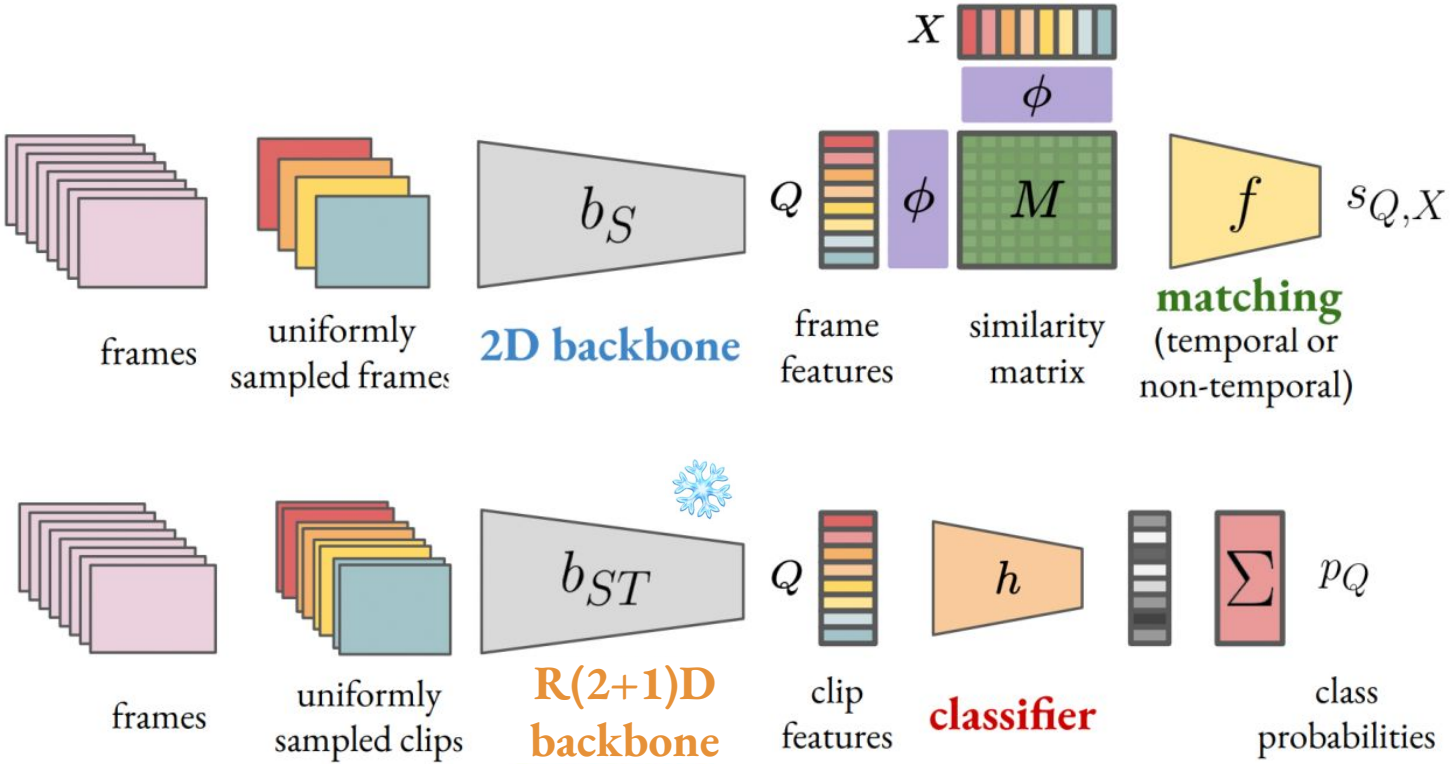
Frame-based vs clip-based features



End-to-end learning vs pretraining the backbone

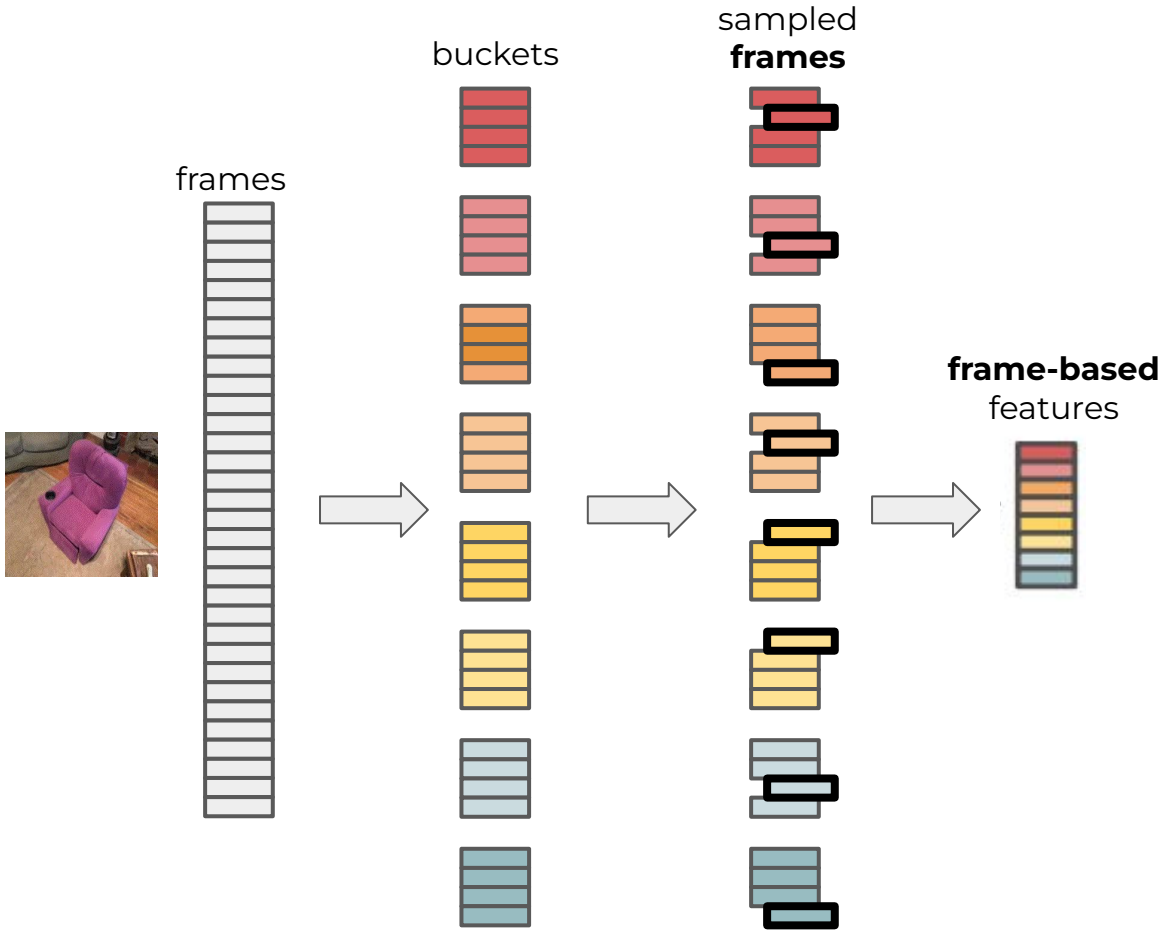


Need for a common setup!

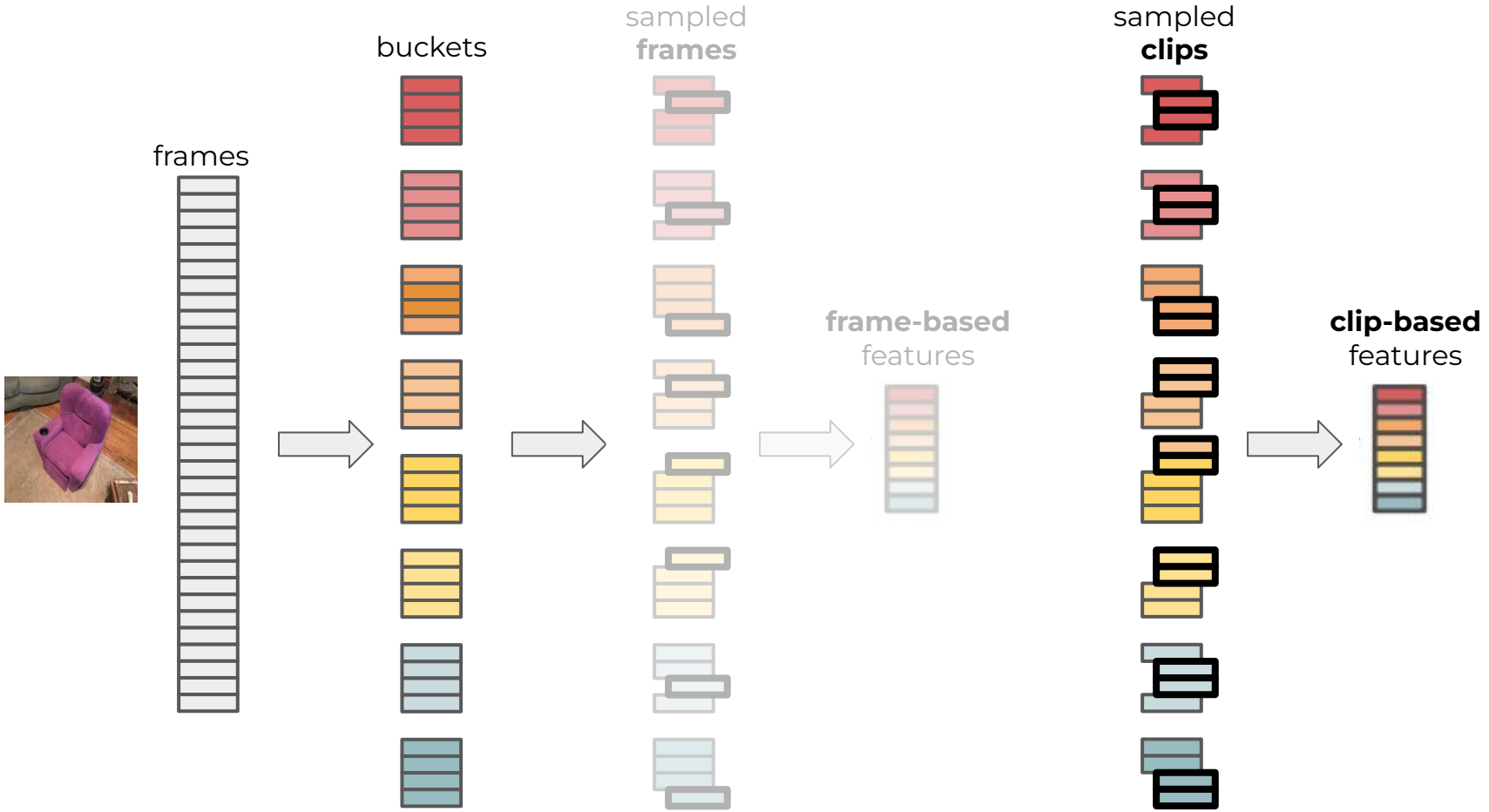


- Need for a common setup to compare fairly classifier-based vs matching-based methods

Frame-based video features ...



… vs clip-based video features



Episode

Novel class data (Few)

support videos

query video

Episode

An **episode** is a classification task where we want to **classify the query video** using support videos

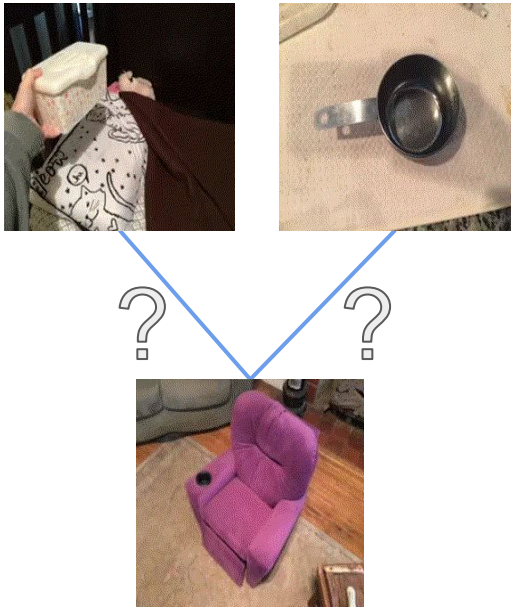
At inference:

- sample multiple episodes
- each episode contains a different set of classes and annotated examples
- accuracy averaged over 10k episodes

Matching-based methods

Dropping something **onto** something

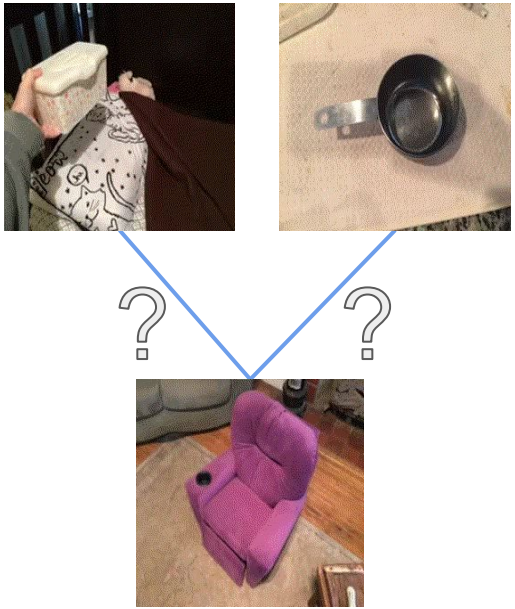
Pretending to put something **into** something

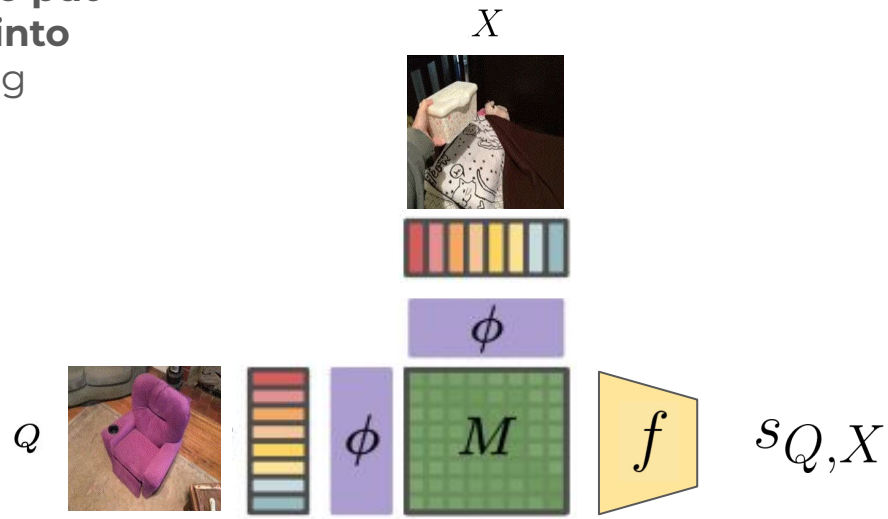


Matching-based methods

Dropping something **onto** something

Pretending to put something **into** something

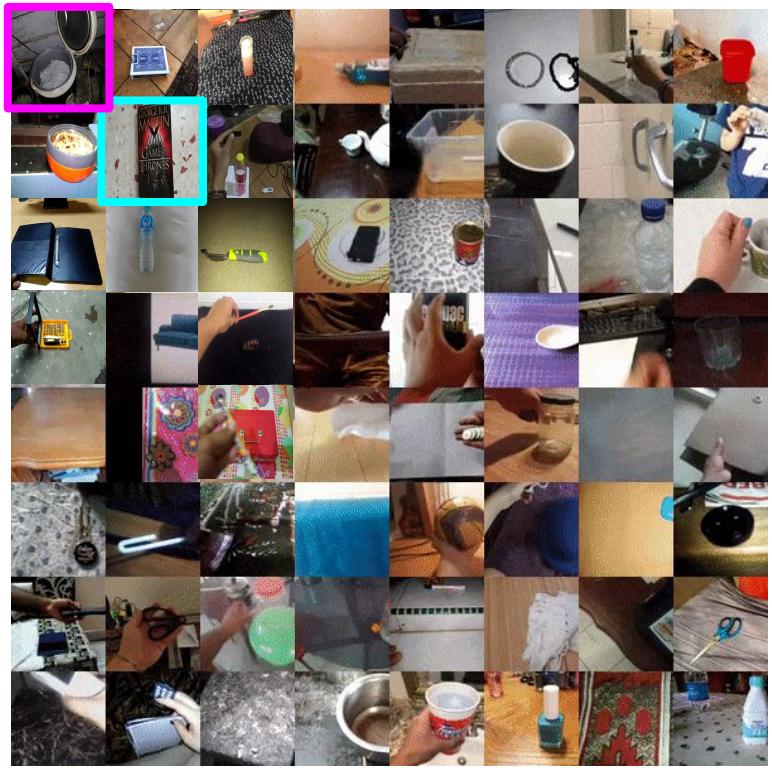




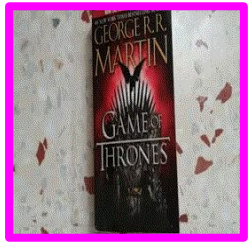
M: temporal similarity matrix
f: matching function
ϕ: optional feature head
$s_{Q,X}$: video-to-video similarity score

Training using episodes

Base class data (Many)



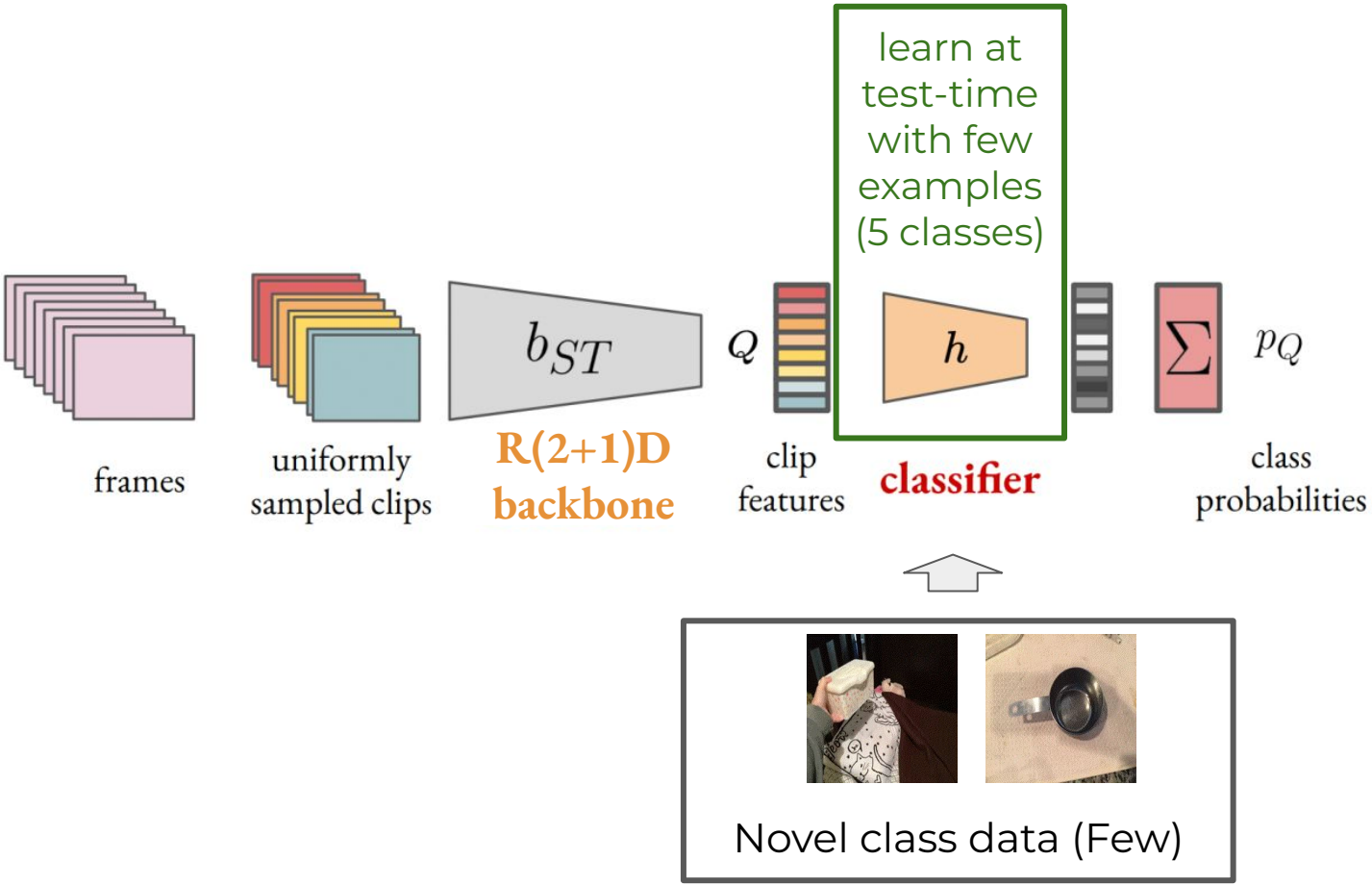
support videos



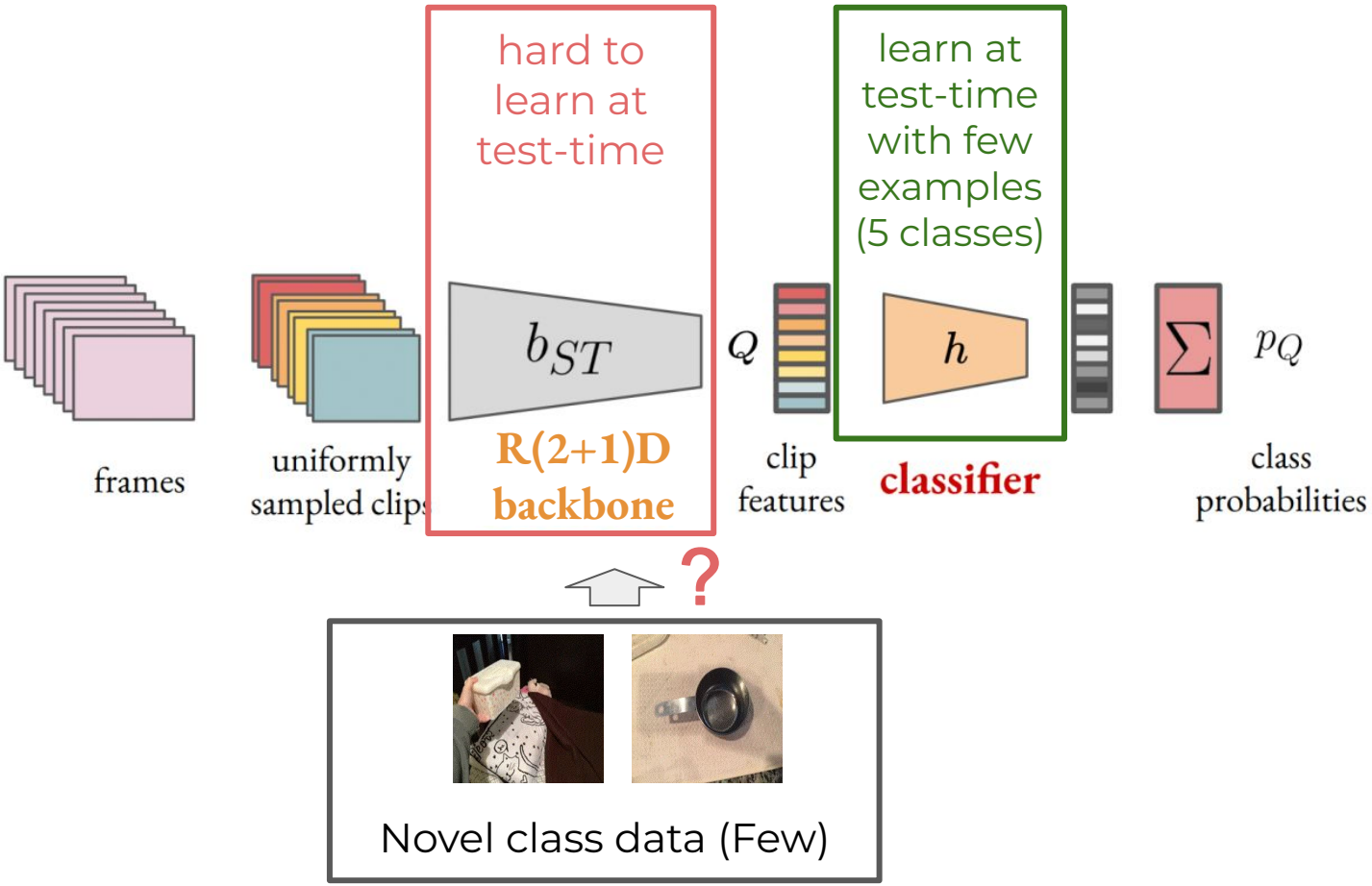
query video

Episode

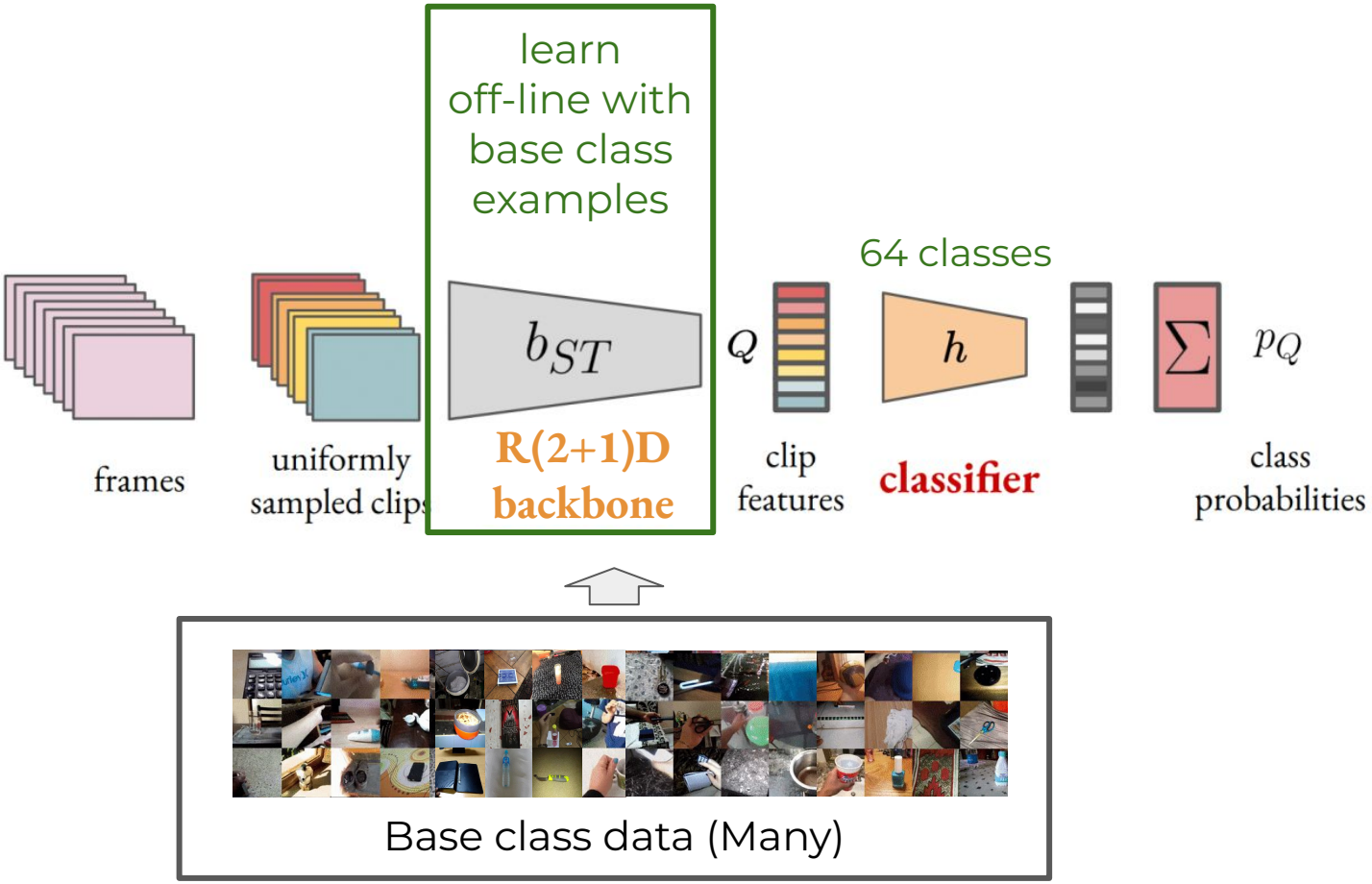
Classifier-based methods



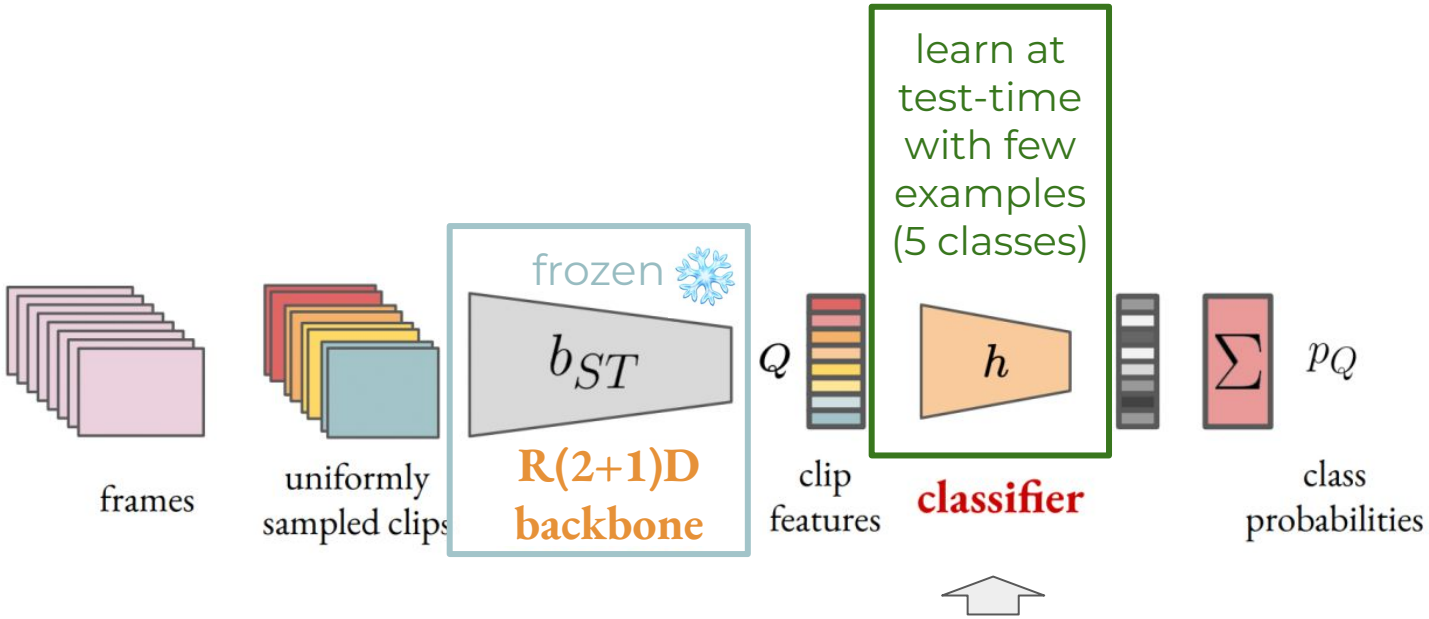
Classifier-based methods



Classifier-based methods



Classifier-based methods



TSL (Xian *et al.* 2021)

Evaluation setup

5 classes,
1 example per class

Metric:

- Top-1 accuracy

Few-shot datasets:

- Kinetics-100: subset of of kinetics-400
- SSv2: subset of the full SSv2 dataset
- UCF-101: subset of the full UCF101 data

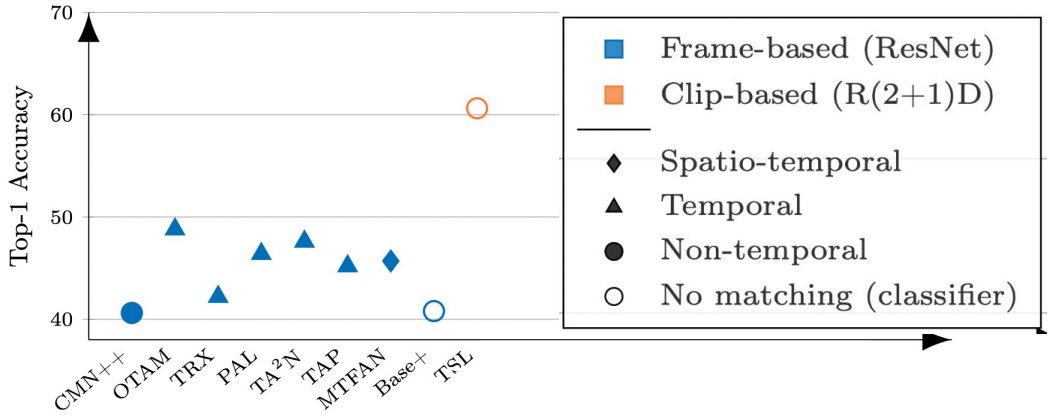
Kinetics (Kay *et al.*, 2017)

SSv2 (Goyal *et al.*, 2021)

UCF-101 (Soomro *et al.*, 2017)

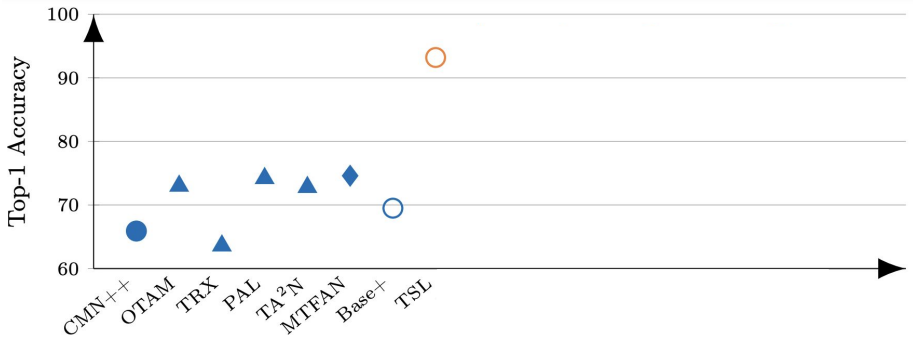
Matching-based vs classifier-based methods

5 classes,
1 example per
class

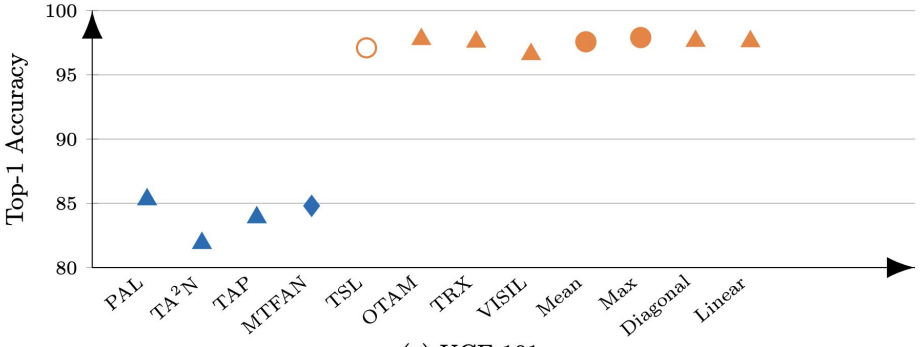


(a) SS-v2

CMN++ (Zhu *et al.*, 2018)
OTAM (Cao *et al.*, 2020)
TRX (Perrett *et al.*, 2021)
PAL (Zhu *et al.*, 2021)
TA^{2}N (Li *et al.* 2021)
TAP (Su *et al.*, 2022)
MTFAN (Wu *et al.* 2022)
Base+ (Zhu *et al.* 2021)
TSL (Xian *et al.* 2021)

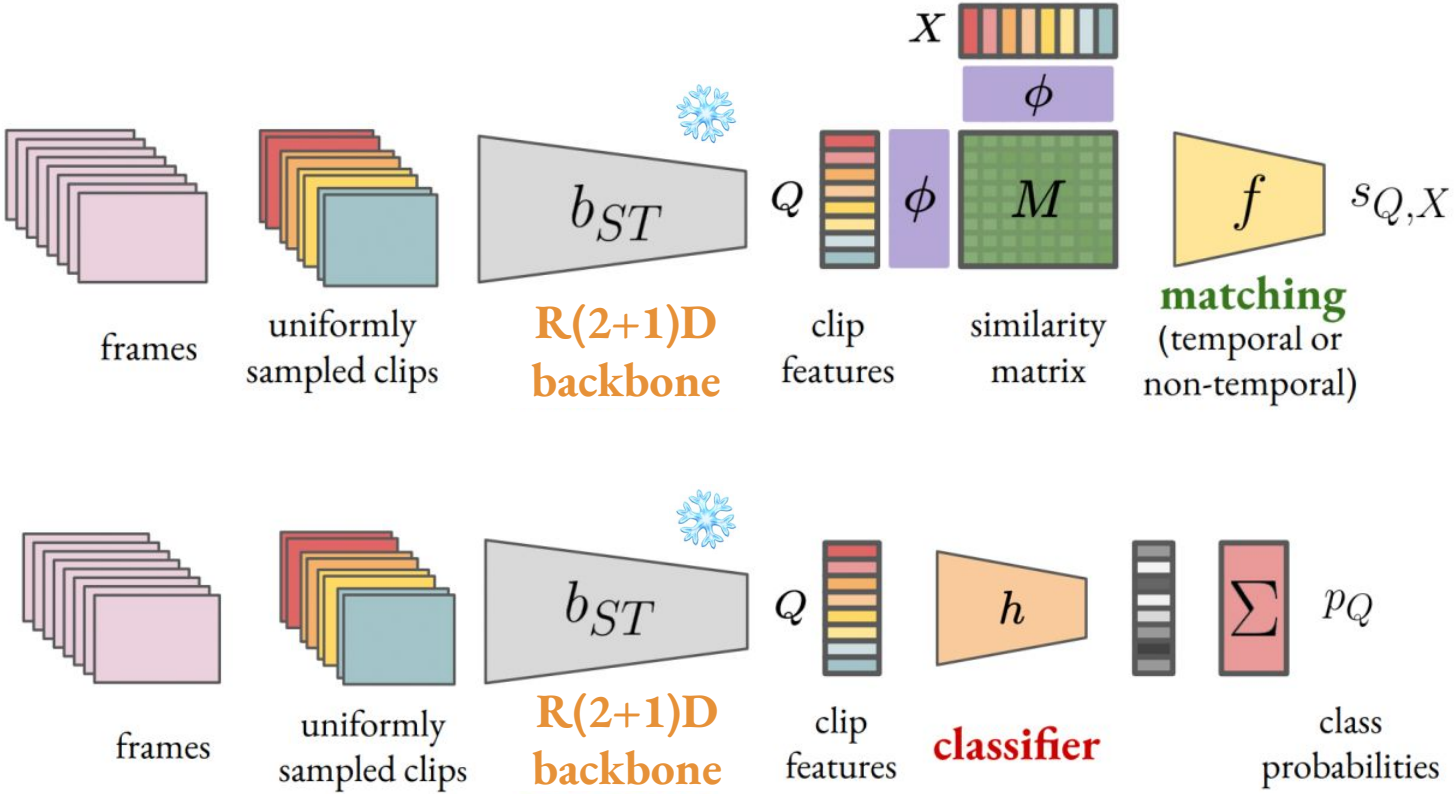


(b) Kinetics-100

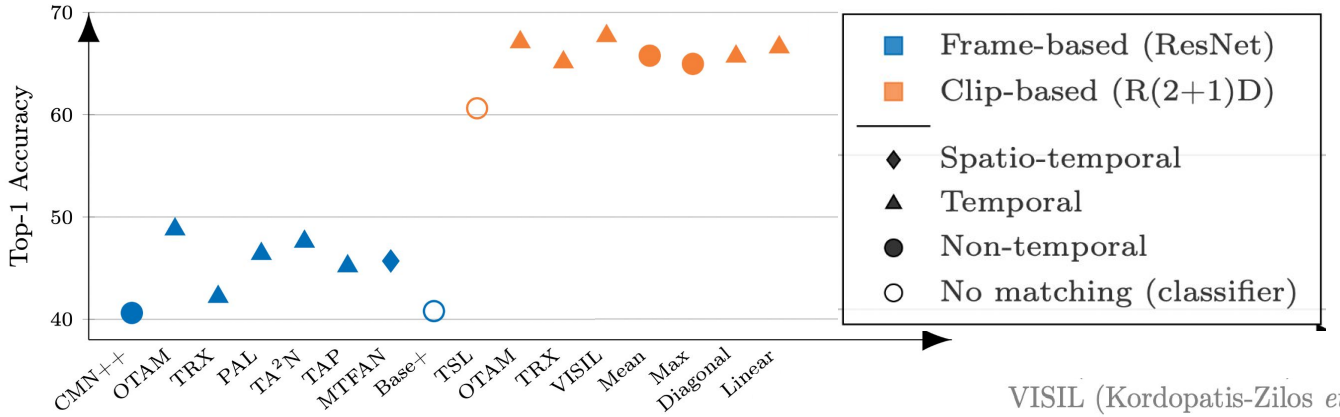


(c) UCF-101

New setup

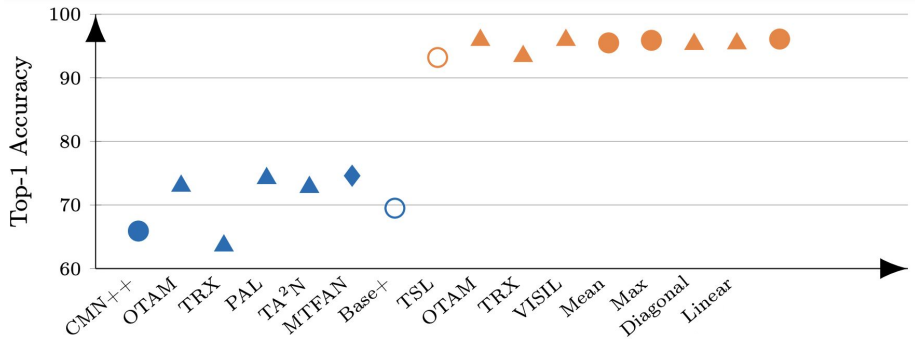

X
ϕ
b_{ST}
Q
ϕ
M
f
$s_{Q,X}$
frames
uniformly sampled clips
R(2+1)D backbone
clip features
similarity matrix
matching (temporal or non-temporal)
b_{ST}
Q
h
$\sum$
p_Q
frames
uniformly sampled clips
R(2+1)D backbone
clip features
classifier
class probabilities

Matching-based vs classifier-based methods

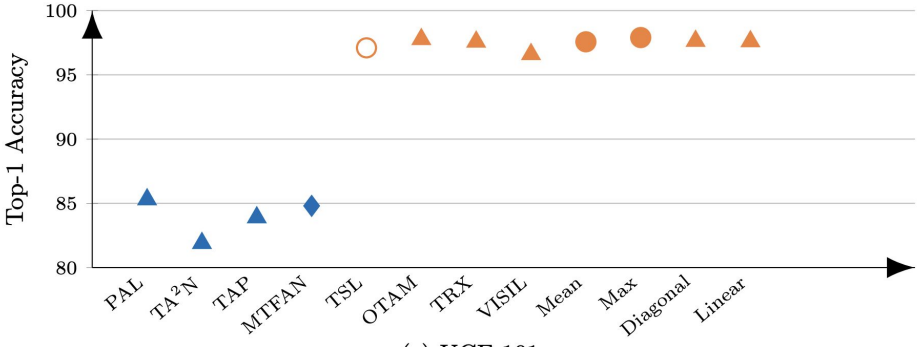


VISIL (Kordopatis-Zilos *et al.* 2019)

(a) SS-v2

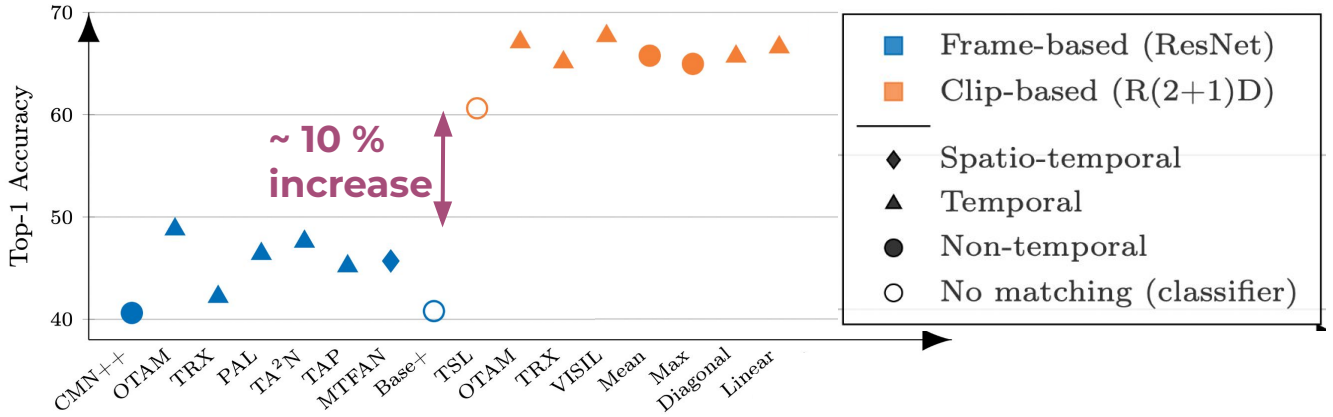


(b) Kinetics-100

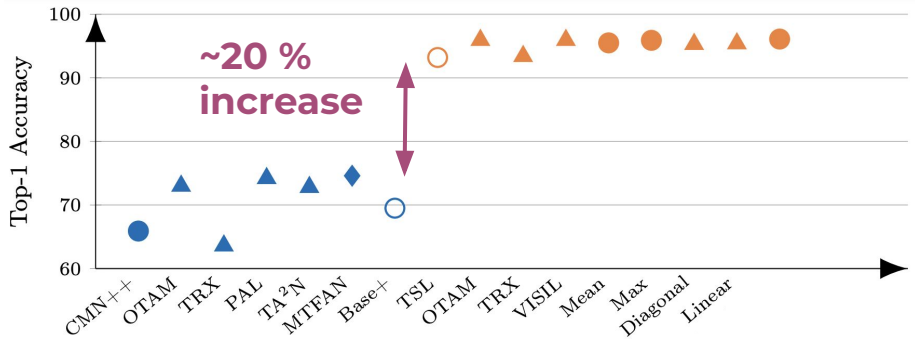


(c) UCF-101

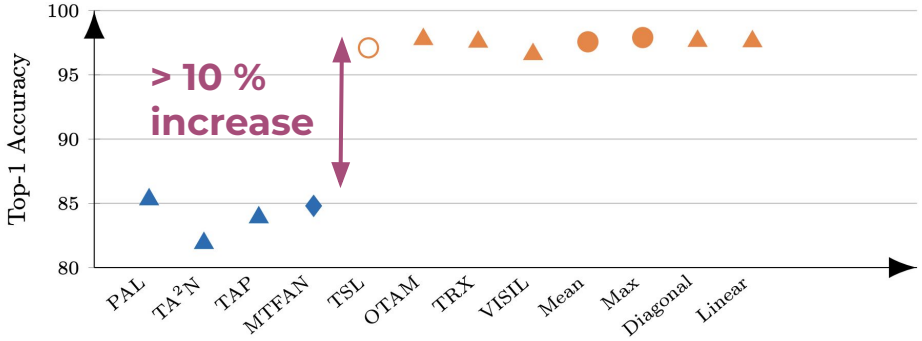
Frame or clip-based features?



(a) SS-v2

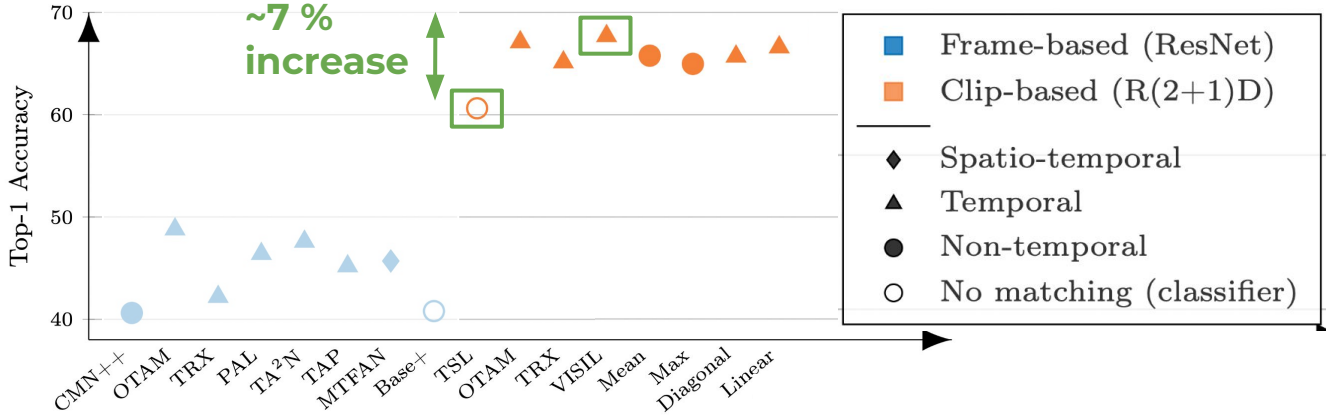


(b) Kinetics-100

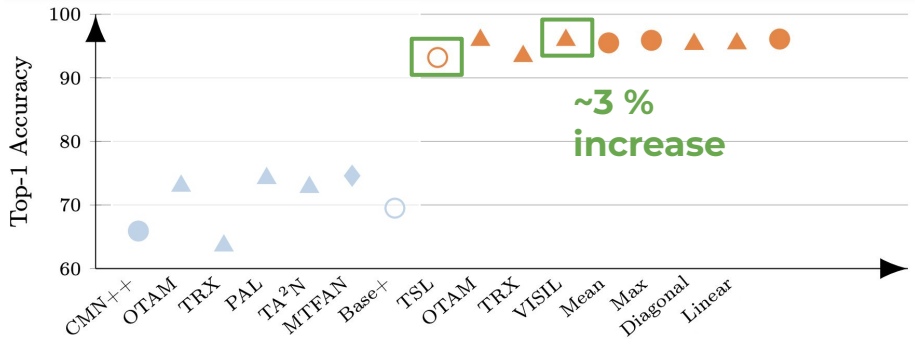


(c) UCF-101

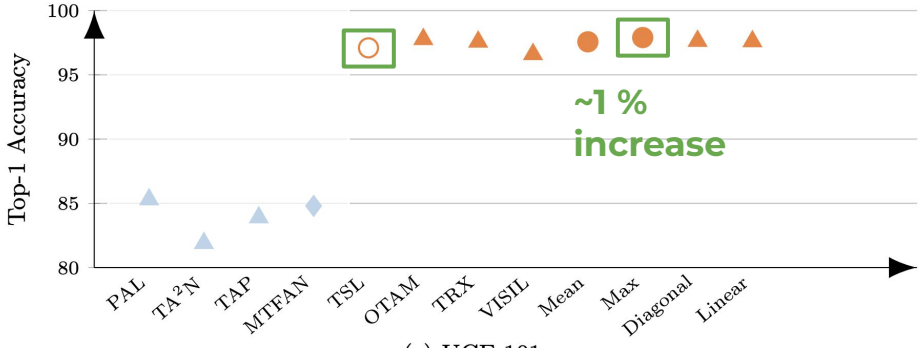
Pairwise matching or classifiers?



(a) SS-v2

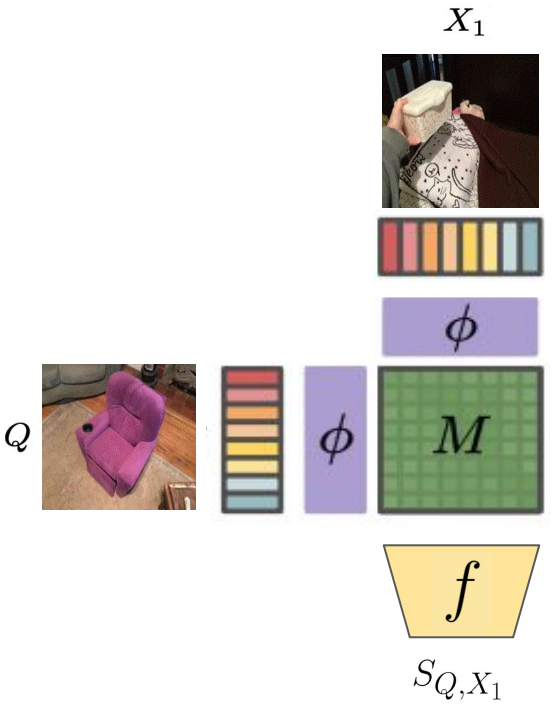


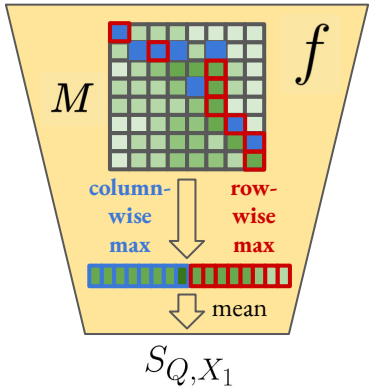
(b) Kinetics-100



(c) UCF-101

Symmetric variant of the chamfer matching function



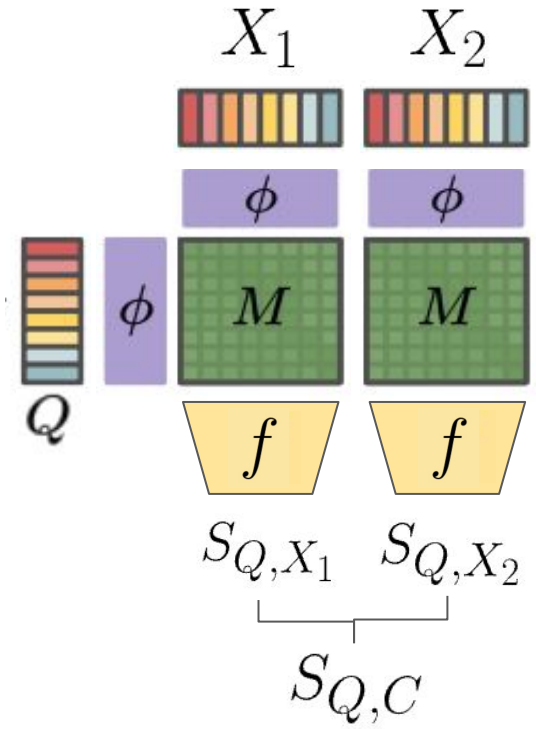


$$f_Q(M) := \frac{1}{n}\sum_i \max_j m_{ij} \quad (1)$$

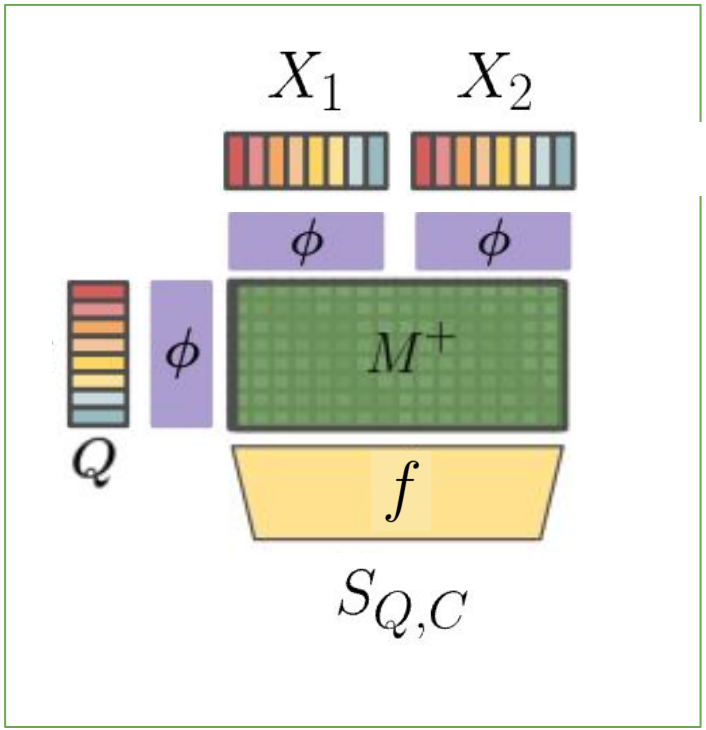
$$f_S(M) := \frac{1}{n}\sum_j \max_i m_{ij} \quad (2)$$

$$f_{QS}(M) := f_Q(M) + f_S(M) \quad (3)$$

Jointly match over multiple examples



Matching independently



Matching jointly

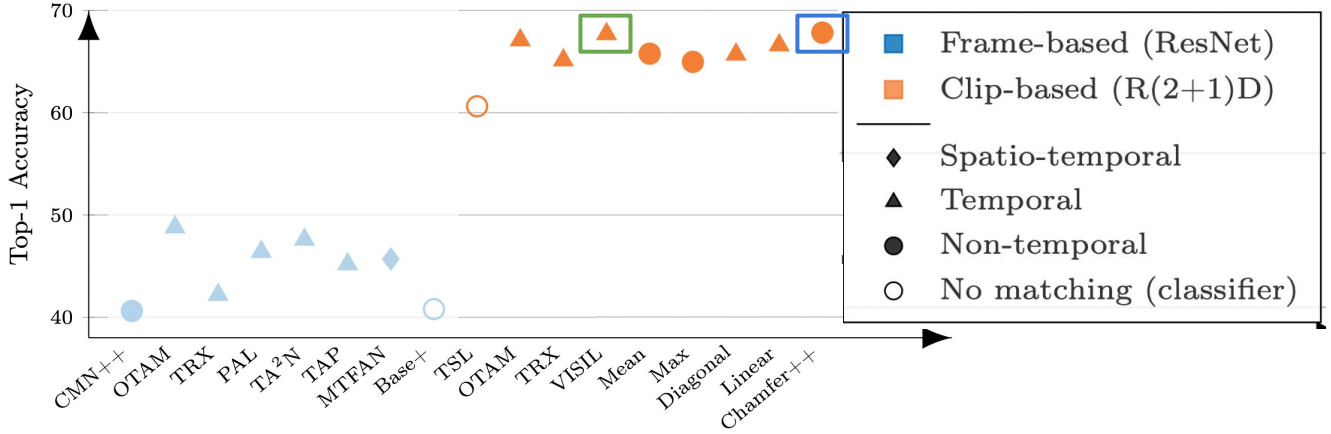
leverage clip-feature pairs

uniformly sampled clips

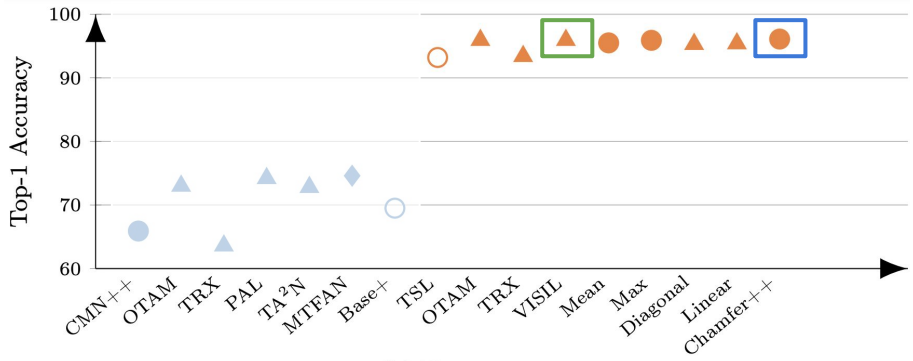
clip features

clip-feature pairs

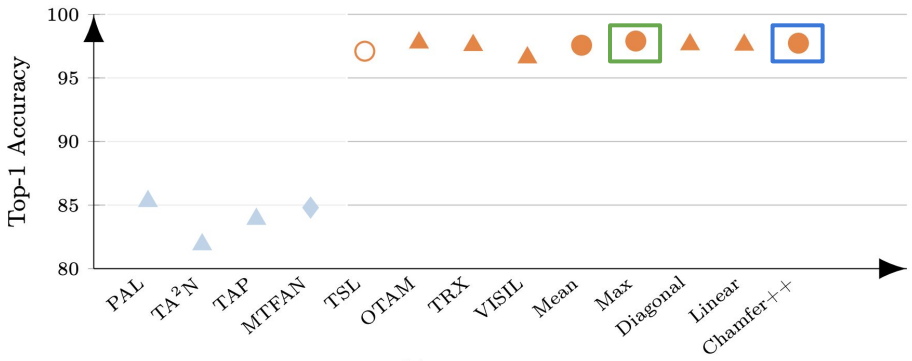
Chamfer++



(a) SS-v2



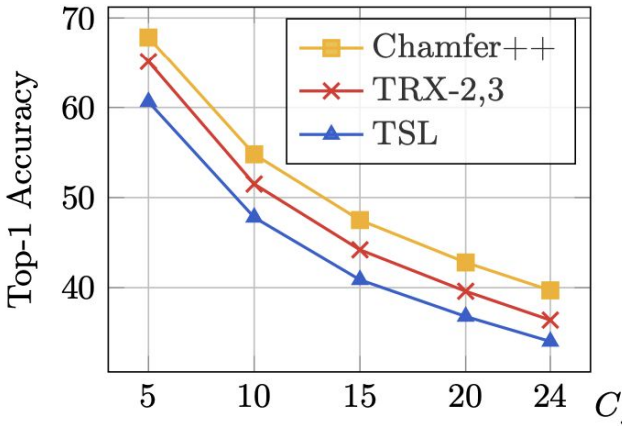
(b) Kinetics-100



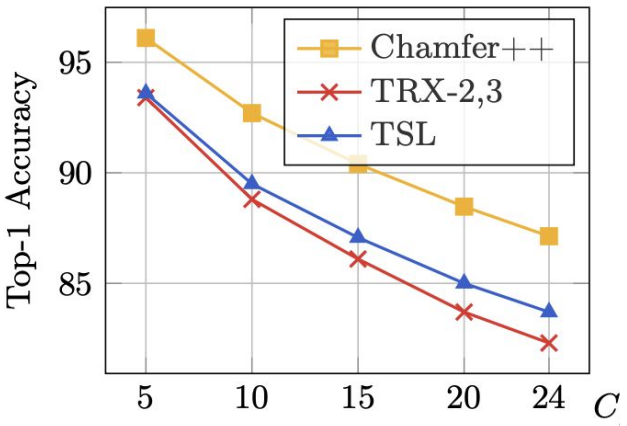
(c) UCF-101

Impact of the number of classes

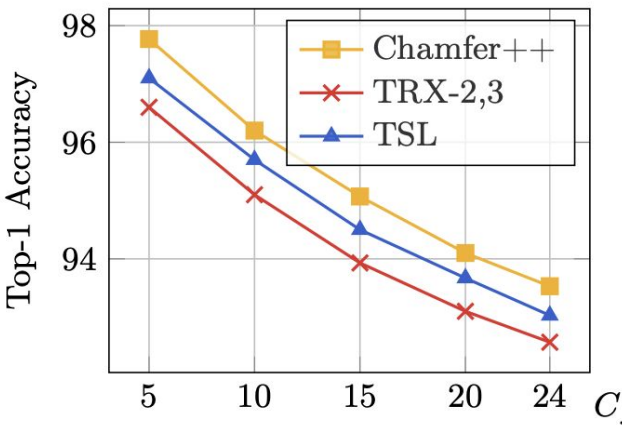
5-24 classes,
1 example per
class



(a) SS-v2



(b) Kinetics-100

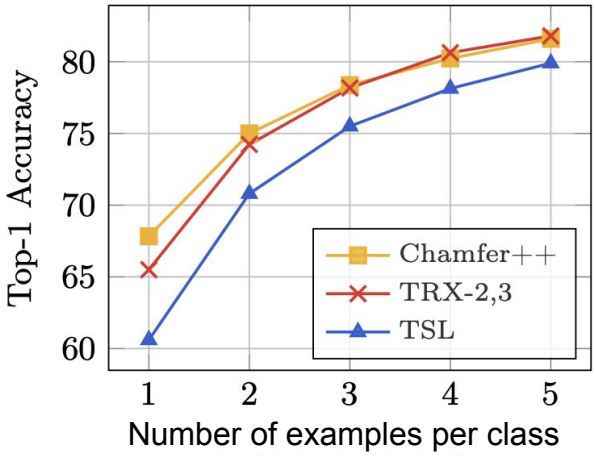


(c) UCF-101

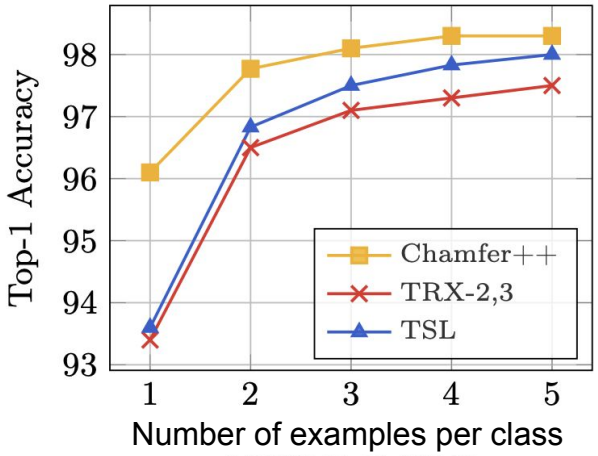
TRX (Perrett *et al.*, 2021)
TSL (Xian *et al.* 2021)

Impact of the number of examples per class

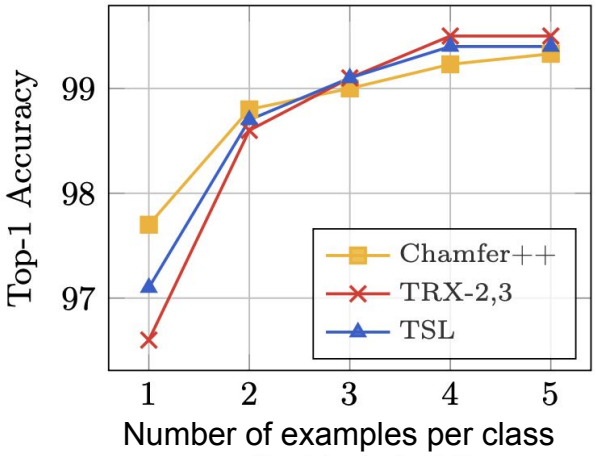
5 classes,
1-5 example per class



(a) SS-v2



(b) Kinetics-100



(c) UCF-101

Take away

Key take-away

- Using temporal information in the video representation is essential
- Matching-based methods outperform classifier-based methods
- Chamfer++, a parameter-free, non-temporal matching function achieves SotA performance

Limited data in video understanding: a summary

- Test Time Training for Video Object Segmentation
  - **tt-MCC**, a strategy which recovers the bulk of the performance under extreme distribution shifts and for sim-to-real transfer

- Few-Shot Action Recognition
  - **Chamfer++**, a parameter-free, non temporal, matching function that achieves SotA performance

Bibliography

[1] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. 2017.

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

[3] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

[4] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv*, 2017.

[5] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[6] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[7] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *NeurIPS*, 2021.

[8] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022.

[9] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *ECCV*, 2018.

[10] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. *arXiv*, 2023.

[11] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017.

[12] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *TPAMI*, 2015.

[13] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. In *CVPR*, 2020.

[14] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *ICCV*, 2019.

[15] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *CVPR*, 2020.

[16] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *CVPR*, 2021.

[17] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

Bibliography

[18] R. Goyal, S. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database for learning and evaluating visual common sense. 2017.

[19] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[20] Y. Xian, B. Korbar, M. Douze, L. Torresani, B. Schiele, and Z. Akata. Generalized few-shot video classification with video retrieval and feature generation. 2021.

[21] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *ECCV*, 2018.

[22] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. 2020.

[23] Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Temporal-relational crosstransformers for few-shot action recognition. 2021.

[24] Xiatian Zhu, Antoine Toisoul, Juan-Manuel Pérez-Rúa, Li Zhang, Brais Martinez, and Tao Xiang. Few-shot action recognition with prototype-centered attentive learning. In *BMVC*, 2021.

[25] Shuyuan Li, Huabin Liu, Rui Qian, Yuxi Li, John See, Mengjuan Fei, Xiaoyuan Yu, and Weiyao Lin. Ta2n: Two-stage action alignment network for few-shot action recognition. 2022.

[26] Bing Su and Ji-Rong Wen. Temporal alignment prediction for supervised representation learning and few-shot sequence classification. In *ICLR*, 2022.

[27] Jiamin Wu, Tianzhu Zhang, Zhe Zhang, Feng Wu, and Yongdong Zhang. Motion-modulated temporal fragment alignment network for few-shot action recognition. In *CVPR*, 2022.

[28] Zhenxi Zhu, Limin Wang, Sheng Guo, and Gangshan Wu. A closer look at few-shot video classification: A new baseline and benchmark. 2021.

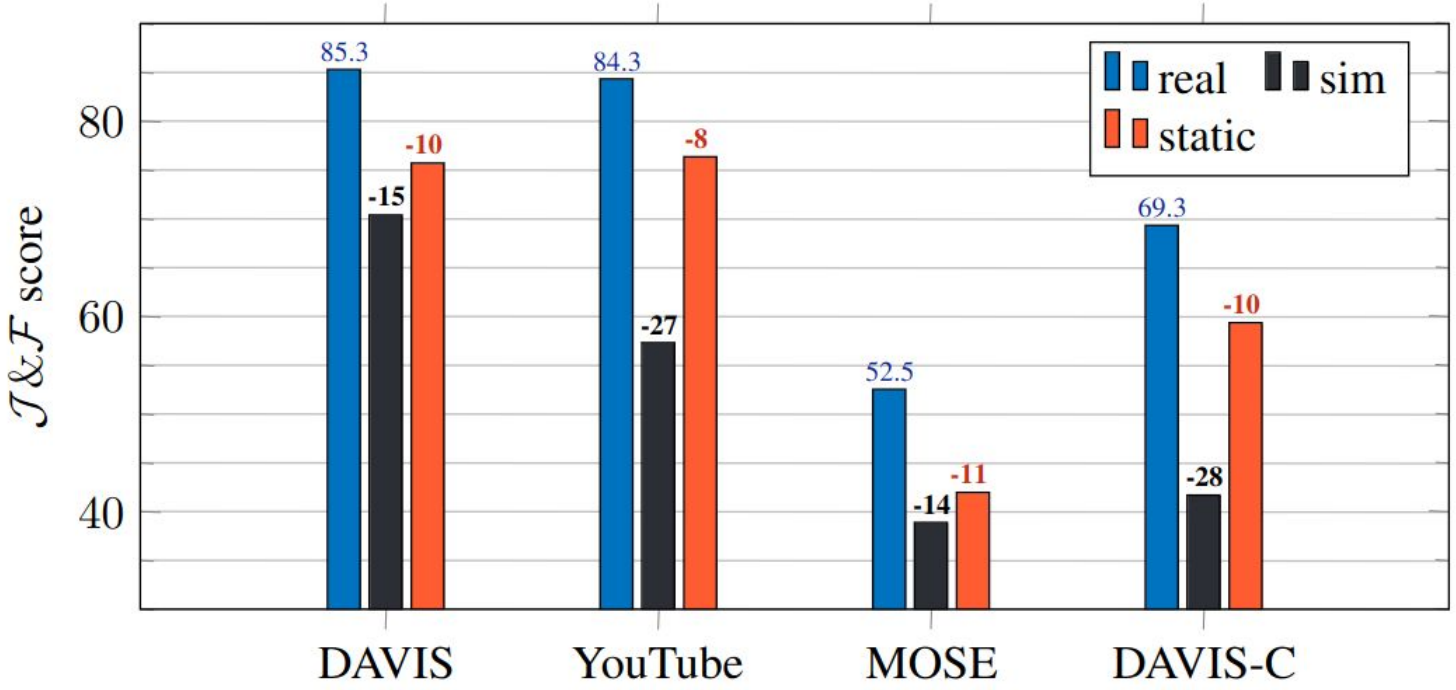
[29] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Visil: Fine-grained spatio-temporal video similarity learning. 2019.

THANK YOU

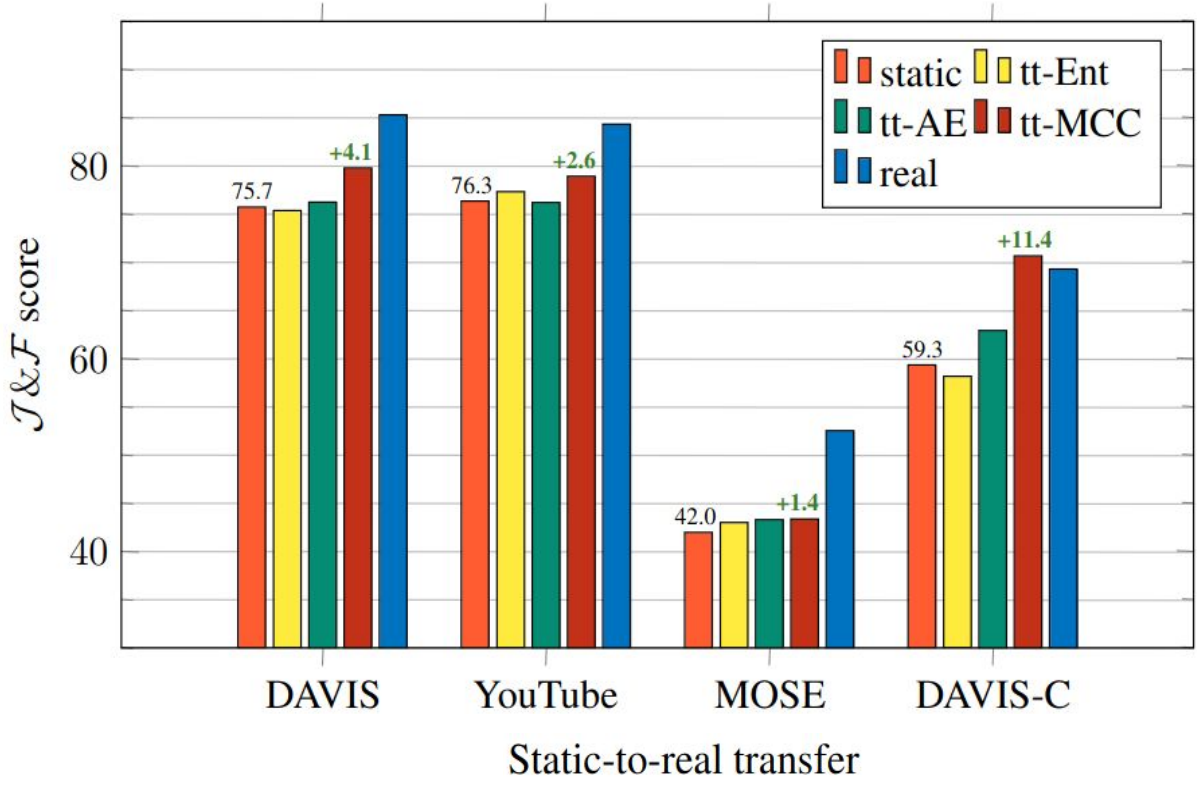
Complements - VOS

static-to-real transfer

Performance with no real-video seen during training (full)



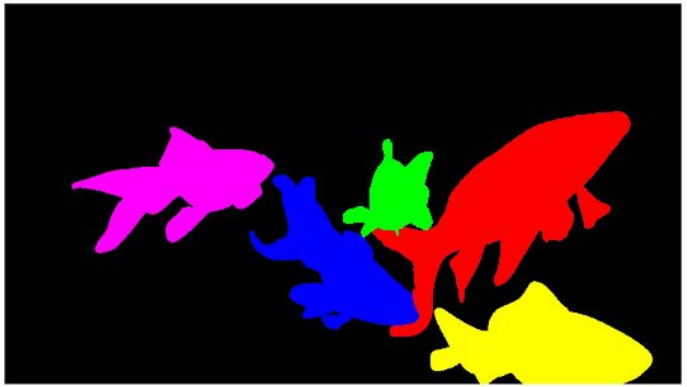
What if we started from the static model?



Qualitative performance: static-to-real

RGB

GroundTruth

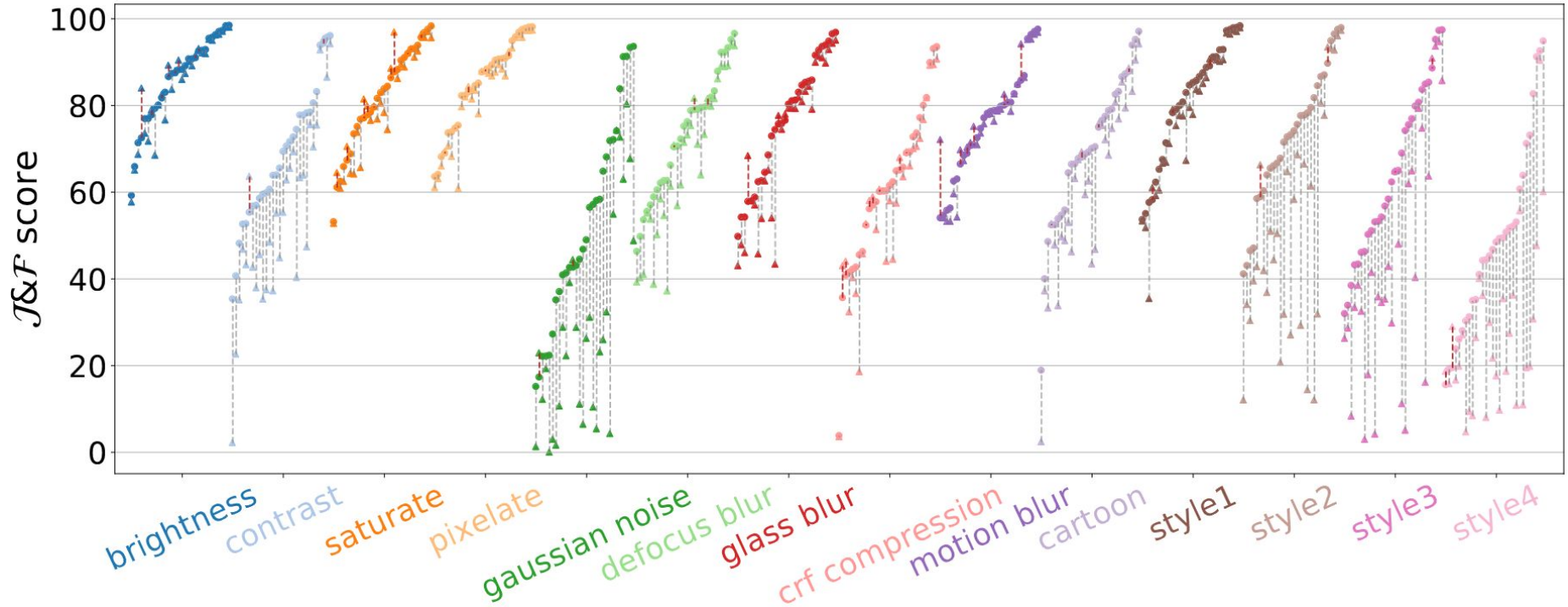


STCN static (J&F=82.67)

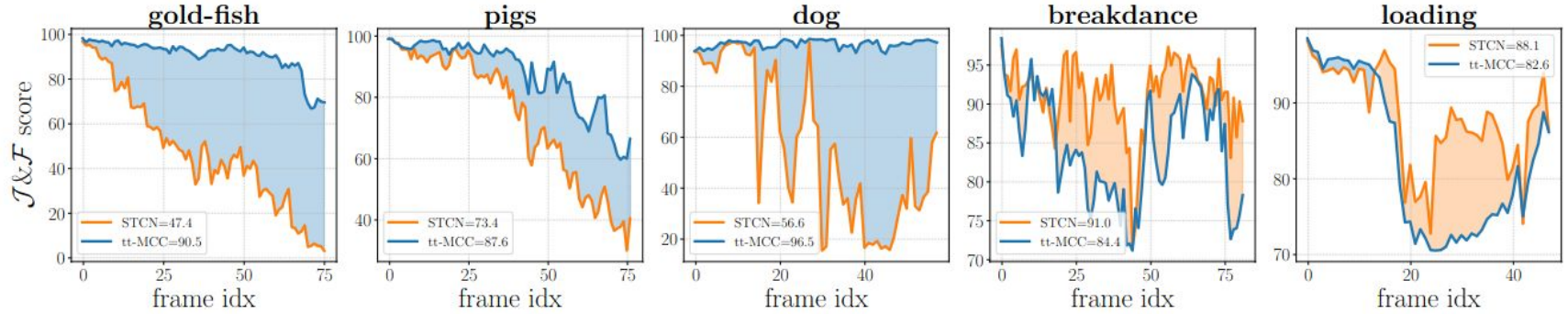
tt-MCC (J&F=84.80)

Breakdown

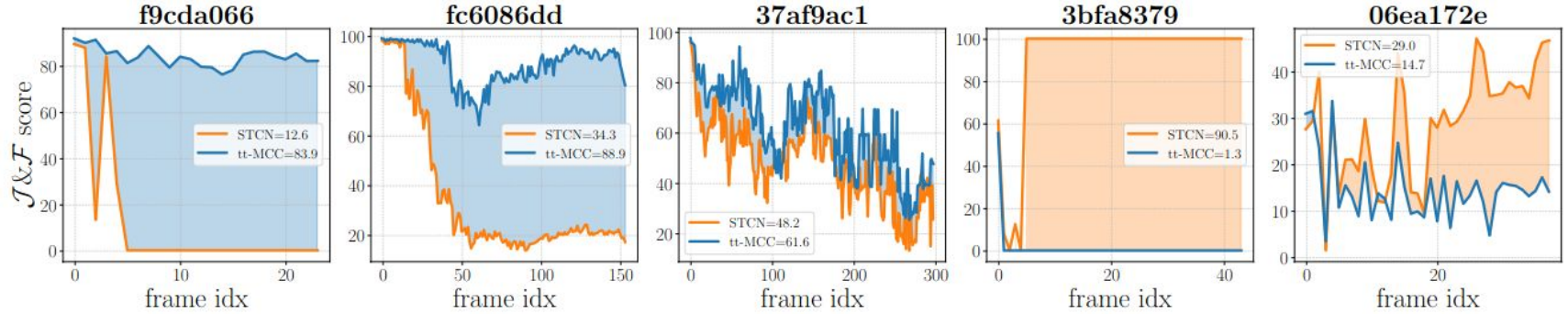
Per-video performance on Davis-C (medium)



Per frame performance



(a) DAVIS



(b) MOSE-train

Breakdown for sim-to-real on MOSE-train

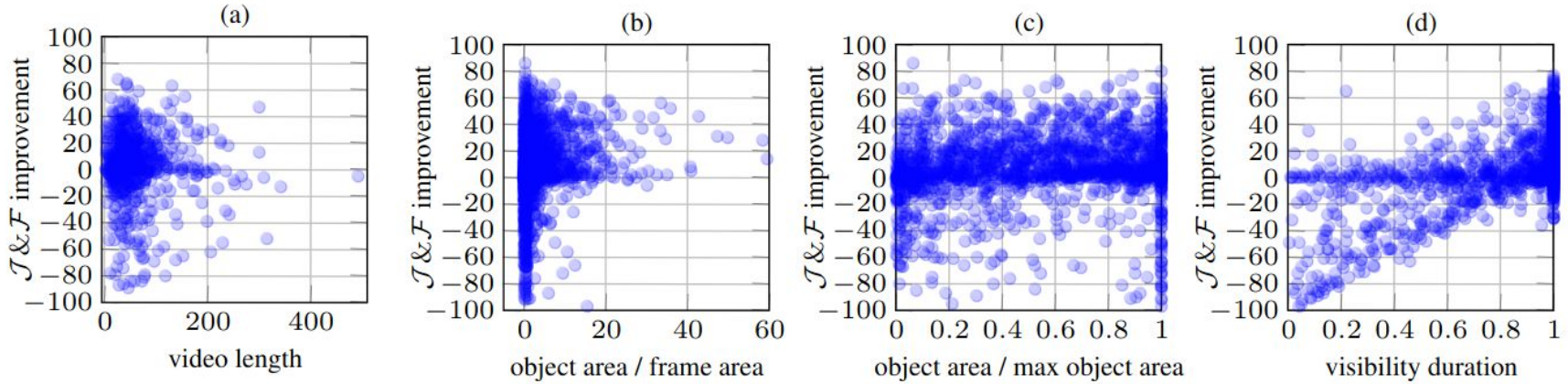
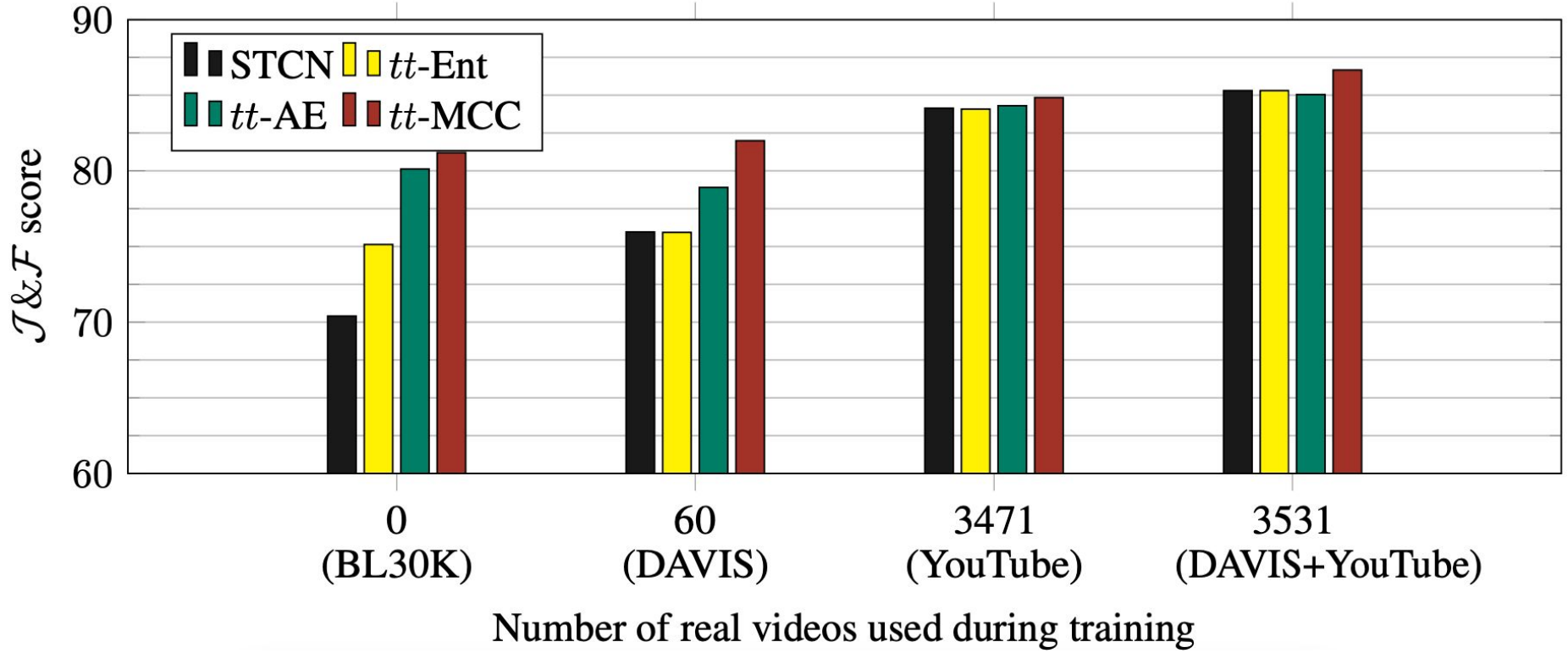


Figure A: **Performance gain of TTT per object for the sim-to-real case on MOSE-train.** We plot performance gain vs video length in number of frames, object area in the 1st frame normalized by the frame area, object area in the first frame normalized by the maximum object area over all frames, and the percentage of the video length where the object is visible.

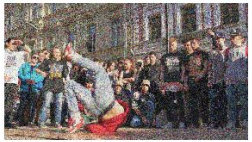
Impact of offline training size



Method

Unravelling the tt-MCC


x_0


x_i


x_j

(a) frame triplet


m_0

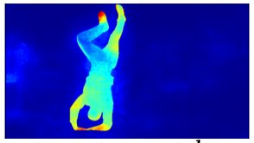
(b) provided mask


$\hat{m}_i$ (step 1)


$\hat{m}_j$ (step 2)


$\hat{m}_i^b$ (step 3)


$\hat{m}_0^b$ (step 4)


$\ell_{CE}(m_0, \hat{m}_0^b)$

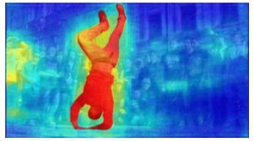
(c) Mask predictions **before** test-time training

(d) loss before TTT


$|\partial L_{MCC}/\partial \hat{m}_i|$


$|\partial L_{MCC}/\partial \hat{m}_j|$


$|\partial L_{MCC}/\partial \hat{m}_i^b|$


$|\partial L_{MCC}/\partial \hat{m}_0^b|$

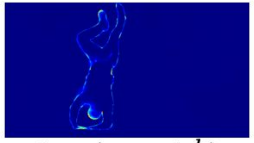
(e) Gradients of mask predictions **before** test-time training


$\hat{m}_i$ (step 1)


$\hat{m}_j$ (step 2)


$\hat{m}_i^b$ (step 3)


$\hat{m}_0^b$ (step 4)


$\ell_{CE}(m_0, \hat{m}_0^b)$

(f) Mask predictions **after** test-time training

(g) loss after TTT

Hyper-parameters

- the number of frames in the cycle:
  - going from 3 to 5 increases the perf by 8% for DAVIS but at the cost of an increase in time of 66%
- triplet sampling:
  - default setup: jump step of 10
  - random sampling / early sampling / late sampling
- the number of iterations

Complements - FSAR

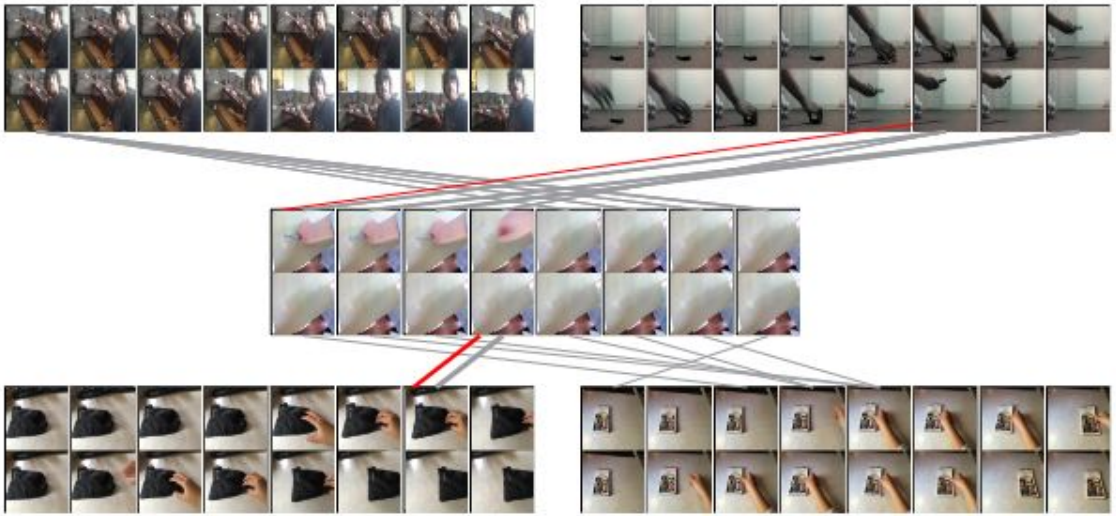
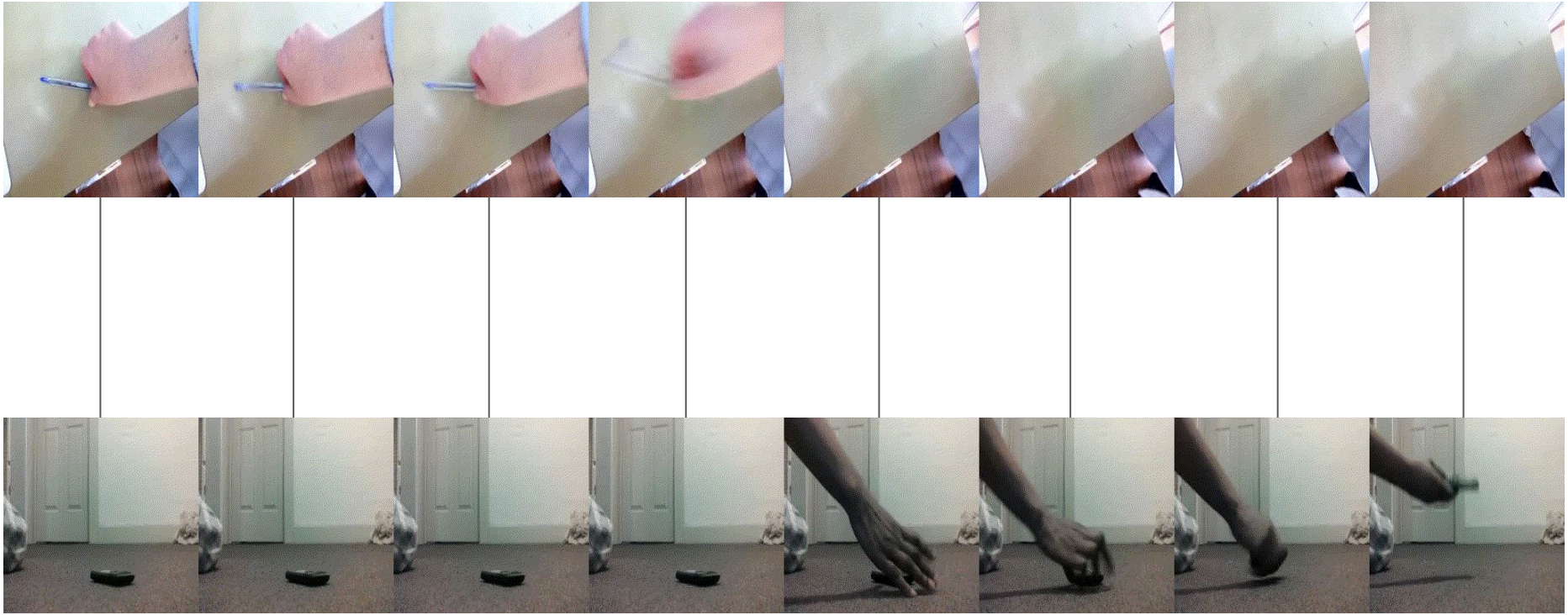


Fig. 5: **Matching a query video (middle) from "Picking something up" with two support videos of "Picking something up" (top) and two support videos of "Pulling something from left to right" (bottom)**. Each video consists of 8 clips with 16 frames each. We only show the first and last frames on top of each other in the figure (see project page for animated versions). In grey, we draw the correspondences between the query and the support videos selected by Chamfer (query-based). In red, we draw the correspondence selected by max (single strongest correspondence). Line thickness corresponds to the pairwise similarity.

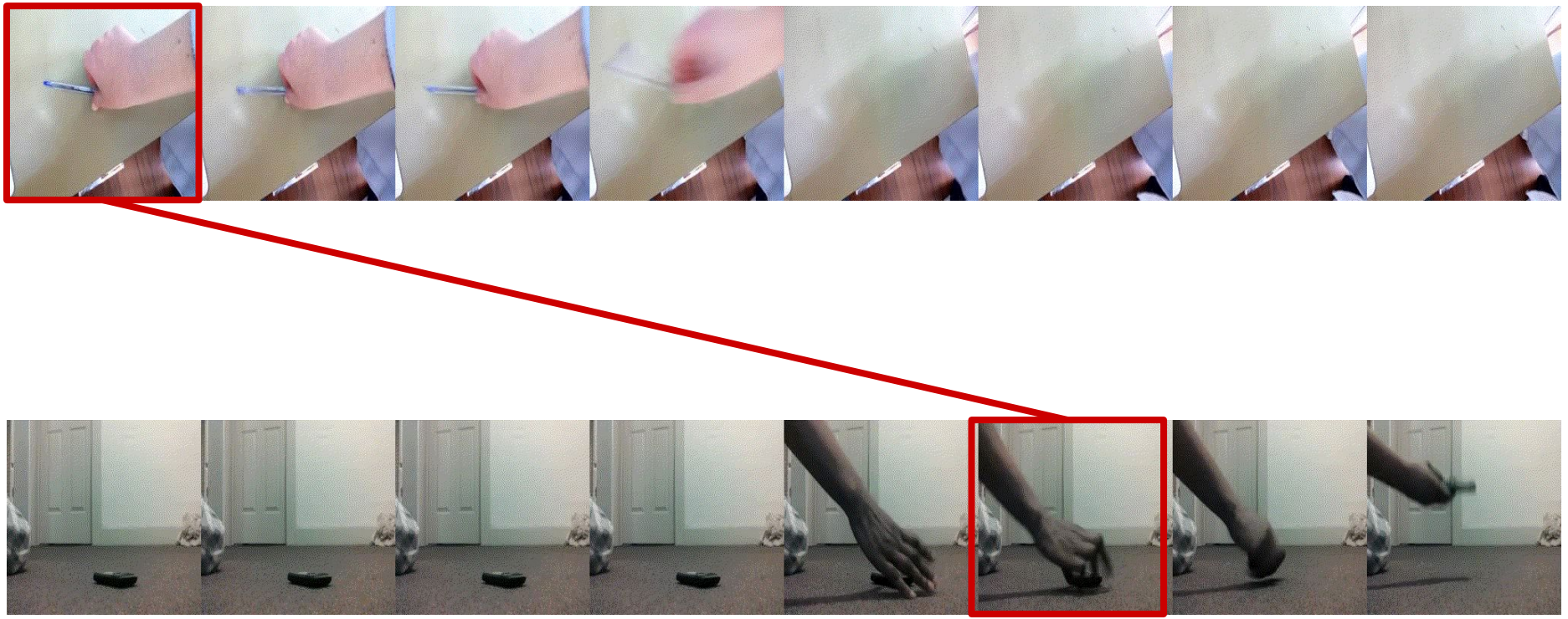
Comparing with the positive example

Comparing with the negative example

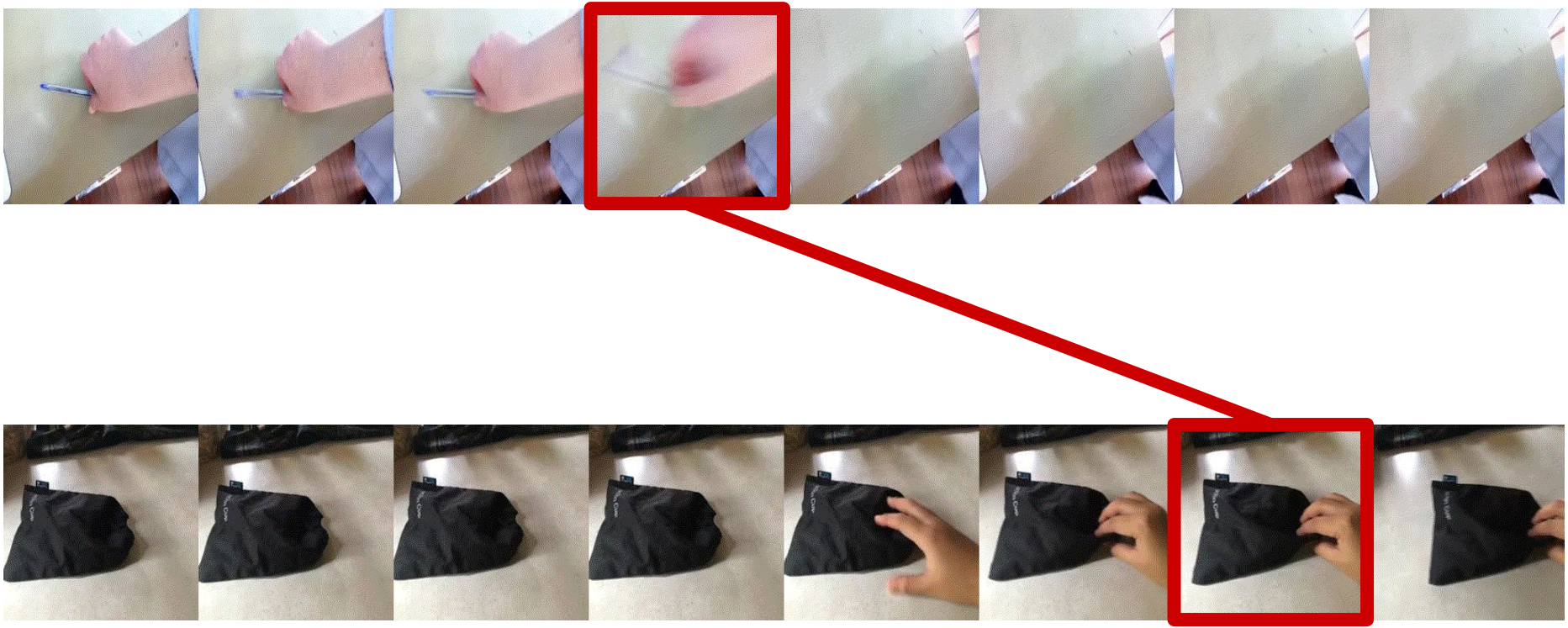
Diagonal matching function



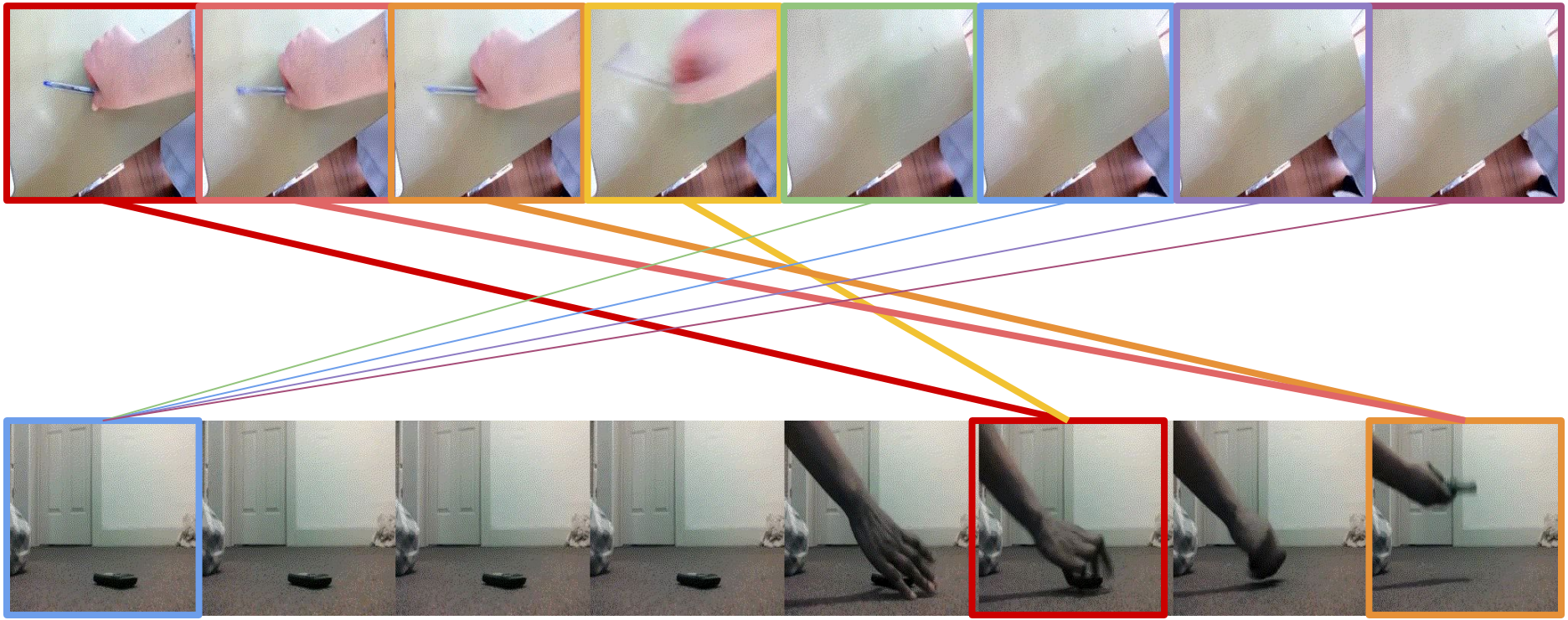
Max matching function



Max matches more with the negative example



Chamfer matching function



Chamfer matching function