point tracking

Adam Harley
Meta

Point tracking

Given any pixel coordinate, track the corresponding world point.

dently on each frame. Additionally, long-range motion estimation can simplify interactive video manipulation,

1

including matting, rotoscoping, labelling, and object removal. Goldman *et al*. [17] de
particle-based video annotation applications that would be difficult to create using standard motion representations.

1.1 Particle Video Representation

Our approach represents video motion using a set of particles that move through time. Each particle denotes an interpolated image point sample, in contrast to a feature patch that represents a neighborhood of pixels [30]. Particle density is adaptive, so that the algorithm can model detailed motion with substantially fewer particles than pixels.

The algorithm optimizes particle trajectories using an objective function that combines point-based appearance matching and inter-particle distortion. The algorithm extends and truncates particle trajectories to model motion near occlusion boundaries.

Our contributions include posing the particle video problem, defining the particle video representation, and presenting an algorithm for particle motion estimation. We provide a new motion optimization scheme that combines variational techniques with an adaptive motion representation. The algorithm uses weighted links between particles to implicitly represent grouping, providing an alternative to discrete layer-based representations.

1.2 Design Goals

The particle video problem can be described as dense feature tracking or long-range optical flow. We want to track the trajectory of each pixel through a given video. Ideally each trajectory would correspond to the motion of a physical real-world surface point.

A primary goal is the ability to model complex motion and occlusion. We want the algorithm to handle general video, which may include close-ups of people talking, hand-held camera motion, multiple independently moving objects, textureless regions, narrow fields of view, and complicated geometry (e.g. trees or office clutter).

A particle approach provides this kind of flexibility. Particles can represent complicated geometry and motion because they are small; a particle's appearance will not change as rapidly as the appearance of a large feature patch, and it is less likely to straddle an occlusion boundary. Particles represent motion in a non-parametric manner; they do not assume that the scene consists of planar or rigid components.

This flexibility in modelling complex motion can also be achieved by optical flow, but the optical flow representation is best suited to successive pairs of frames, not to long sequences. Frame-to-frame flow fields can be concatenated to obtain longer-range correspondences, but the resulting multi-frame flow must be refined at each step to avoid drift.

In contrast, the particle representation allows a form of random-access motion evaluation: given a set of particles, we can easily find correspondences between any pair of frames (assuming the frames have a sufficient number of particles in common). Furthermore, unlike a sequence of motion fields, the particle representation provides discrete motion primitives, which are valuable for subsequent use of the motion information, such as interactive video matting [26] and interactive video annotation [17].

1.3 Overview

Section 2 describes related work in motion estimation. We combine several of these methods to create an optical flow algorithm described in Section 3 (with details in Appendix A). Optical flow is used as an input to our particle-based motion estimation.

2

The particle algorithm is described in Section 4, which explains how particles are added, propagated, linked, optimized, and pruned. These steps are performed as the algorithm sweeps back and forth across a video, constructing a complete particle representation of the video's motion.

Section 5 includes an evaluation of the particle video algorithm on a variety of real-world videos. We quantify the performance of the algorithm and possible alternatives. We provide mechanisms for visualizing the

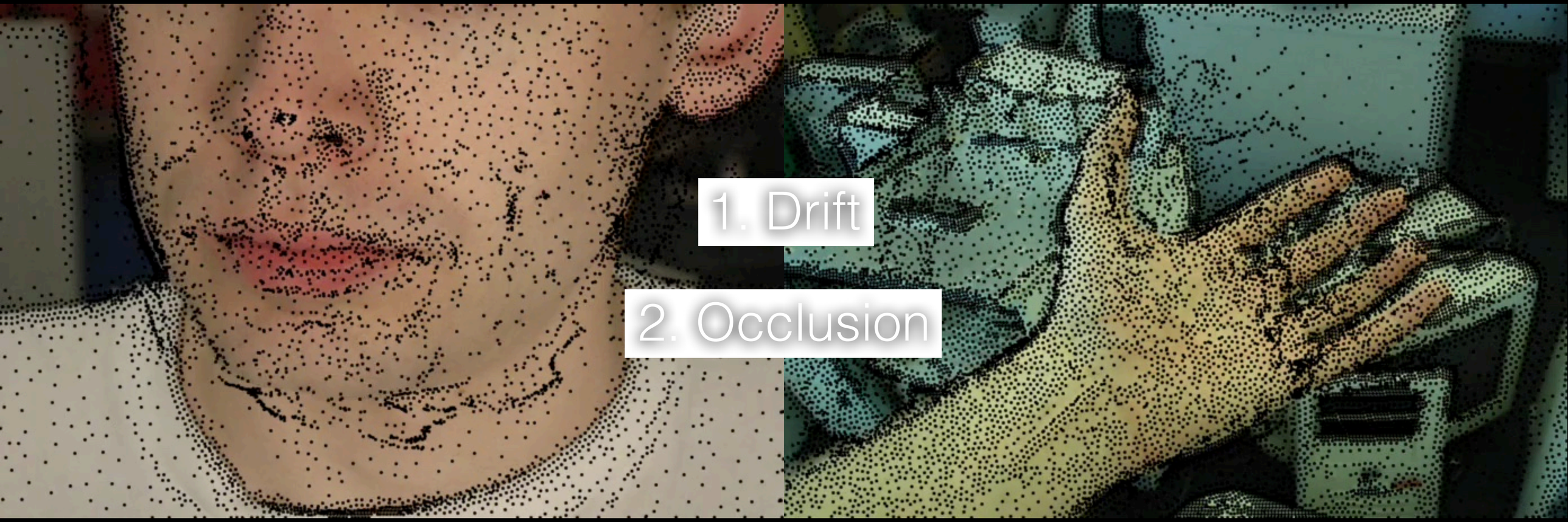
The "particle video" problem

"We want to track the trajectory of each pixel through a given video.

Ideally each trajectory would correspond to the motion of a physical real-world surface point."

Sand & Teller, Particle Video: Long-Range Motion Estimation using Point Trajectories, 2006.

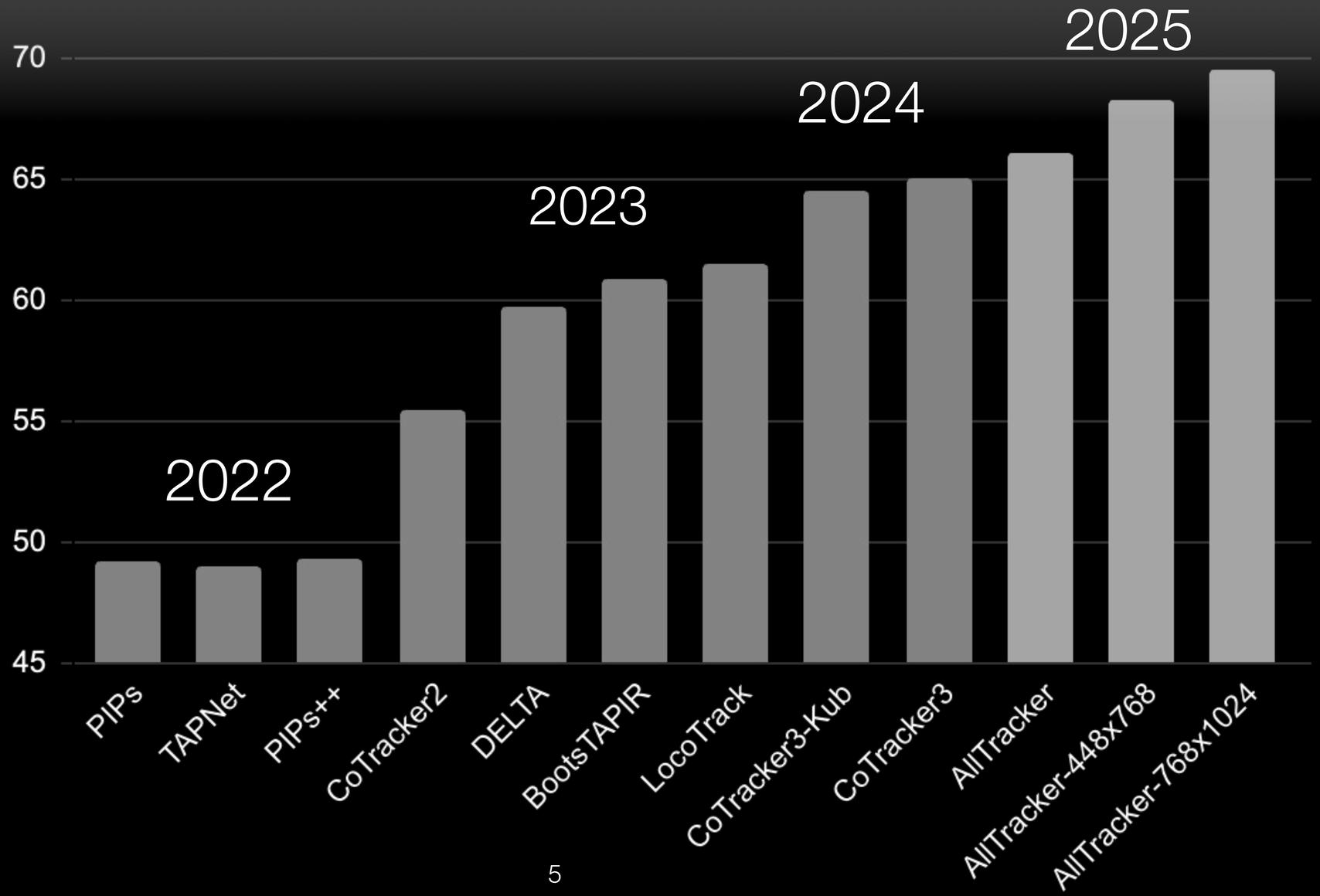
Before: Point tracking through optical flow



Sand & Teller. "Particle video: Long-range motion estimation using point trajectories." IJCV 80.1 (2008): 72.

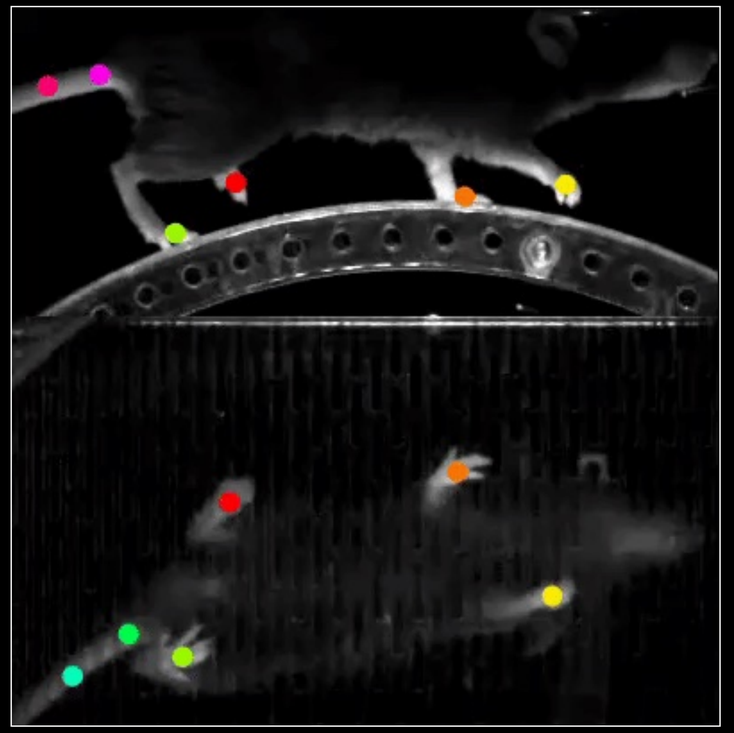
Now: A problem all of its own

Average accuracy across multiple datasets



Point tracks make science easier

Polajnar, Kvinikadze, **Harley**, Malenovsky. “Wing buzzing as a mechanism for generating vibrational signals in psyllids (Hemiptera: Psylloidea)”. Insect Science 2024.



Mathis et al., “DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning." Nature neuroscience 21.9 (2018): 1281-1289

Goode, C. K., and Gregory P. Sutton. "Control of high-speed jumps: the rotation and energetics of the locust (Schistocerca gregaria)." Journal of Comparative Physiology B 193.2 (2023): 145-153.

Point tracks make robotics easier

Flow as the Cross-Domain Manipulation Interface

Mengda Xu [1, 2, 3] Zhenjia Xu [1,2] Yinghao Xu [1] Cheng Chi [1,2]

Any-point Trajectory Modeling for Policy Learning

Chuan Wen
Kai Chen[6] Qi
[1]UC Berkeley [2]III
[4]Shanghai AI Labora

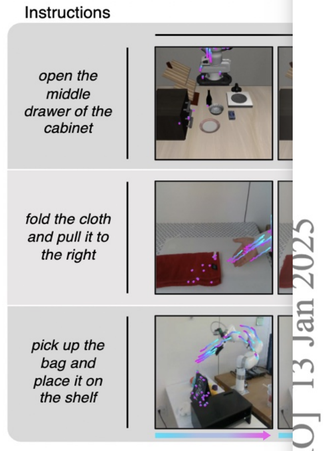
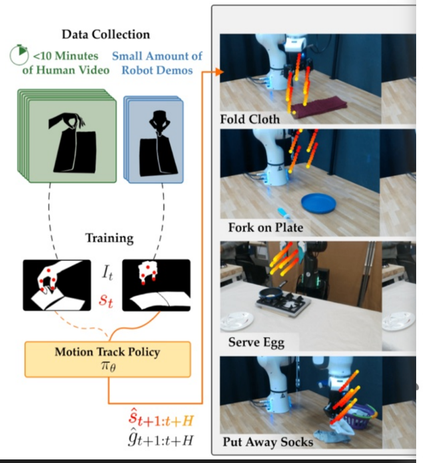


Fig. 1: Given a task instruction and the initial po

MOTION TRACKS: A Unified Representation for Human-Robot Transfer in Few-Shot Imitation Learning

Juntao Ren[1], Priya Sundaresan[2], Dorsa S



Track2Act: Predicting Point Tracks from Internet Videos enables Diverse Zero-shot Robot Manipulation

Homanga Bharadhwaj[1] **Roozbeh Mottaghi**[2] **Abhinav Gupta**[1,*] **Shubham Tulsiani**[1,*]

[1]Carnegie Mellon University and [2]University of Washington, Meta

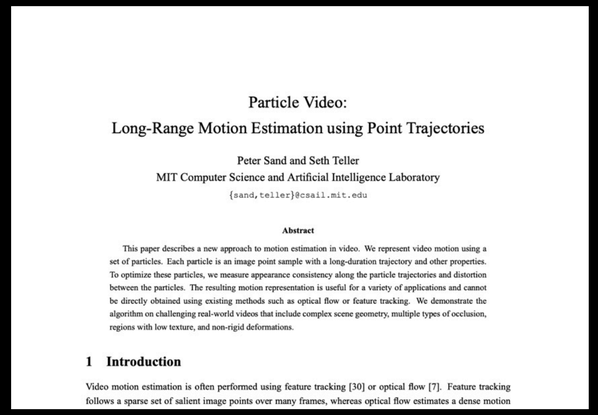
https://homangab.github.io/track2act/

Abstract: We seek to learn a generalizable goal-conditioned policy that enables *zero-shot* robot manipulation — interacting with unseen objects in novel scenes without test-time adaptation. While typical approaches rely on a large amount of demonstration data for such generalization, we propose an approach that leverages web videos to predict plausible interaction plans and learns a task-agnostic transformation to obtain robot actions in the real world. Our framework, *Track2Act* predicts tracks of how points in an image should move in future time-steps based on a goal, and can be trained with diverse videos on the web including those of humans and robots manipulating everyday objects. We use these 2D track predictions to infer a sequence of rigid transforms of the object to be manipulated, and obtain robot end-effector poses that can be executed in an open-loop manner.

The VOTSp2026 Challenge

How do we evaluate point tracks?

Sand & Teller, IJCV 2006.



Particle Video:
Long-Range Motion Estimation using Point Trajectories

Peter Sand and Seth Teller
MIT Computer Science and Artificial Intelligence Laboratory
{sand,teller}@csail.mit.edu

Abstract

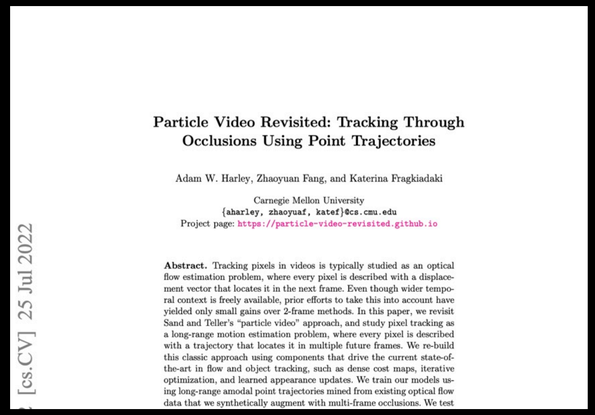
This paper describes a new approach to motion estimation in video. We represent video motion using a set of particles. Each particle is an image point sample with a long-duration trajectory and other properties. To optimize these particles, we measure appearance consistency along the particle trajectories and distortion between the particles. The resulting motion representation is useful for a variety of applications and cannot be directly obtained using existing methods such as optical flow or feature tracking. We demonstrate the algorithm on challenging real-world videos that include complex scene geometry, multiple types of occlusion, regions with low texture, and non-rigid deformations.

1 Introduction

Video motion estimation is often performed using feature tracking [30] or optical flow [7]. Feature tracking follows a sparse set of salient image points over many frames, whereas optical flow estimates a dense motion

Cycle consistency

Harley et al., ECCV 2022.



Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories

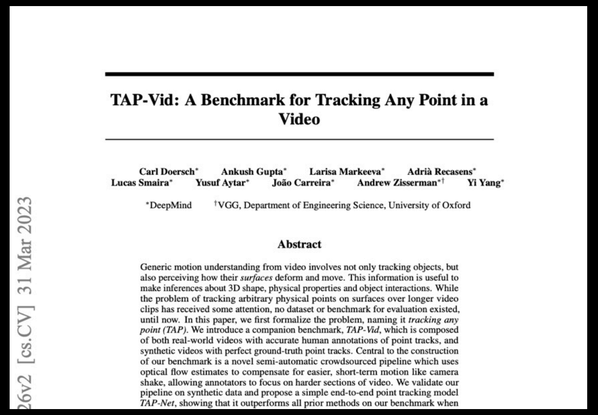
Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki

Carnegie Mellon University
{aharley, zhaoyuaf, katef}@cs.cmu.edu
Project page: https://particle-video-revisited.github.io

Abstract. Tracking pixels in videos is typically studied as an optical flow estimation problem, where every pixel is described with a displacement vector that locates it in the next frame. Even though wider temporal context is freely available, prior efforts to take this into account have yielded only small gains over 2-frame methods. In this paper, we revisit Sand and Teller's "particle video" approach, and study pixel tracking as a long-range motion estimation problem, where every pixel is described with a trajectory that locates it in multiple future frames. We re-build this classic approach using components that drive the current state-of-the-art in flow and object tracking, such as dense cost maps, iterative optimization, and learned appearance updates. We train our models using long-range amodal point trajectories mined from existing optical flow data that we synthetically augment with multi-frame occlusions. We test

Median trajectory error,
Percentage of Correct Keypoints (PCK)

Doersch et al., NeurIPS Datasets & Benchmarks, 2022.



TAP-Vid: A Benchmark for Tracking Any Point in a Video

Carl Doersch* Ankush Gupta* Larisa Markeeva* Adrià Recasens*
Lucas Smaira* Yusuf Aytar* João Carreira* Andrew Zisserman*† Yi Yang*

*DeepMind †VGG, Department of Engineering Science, University of Oxford

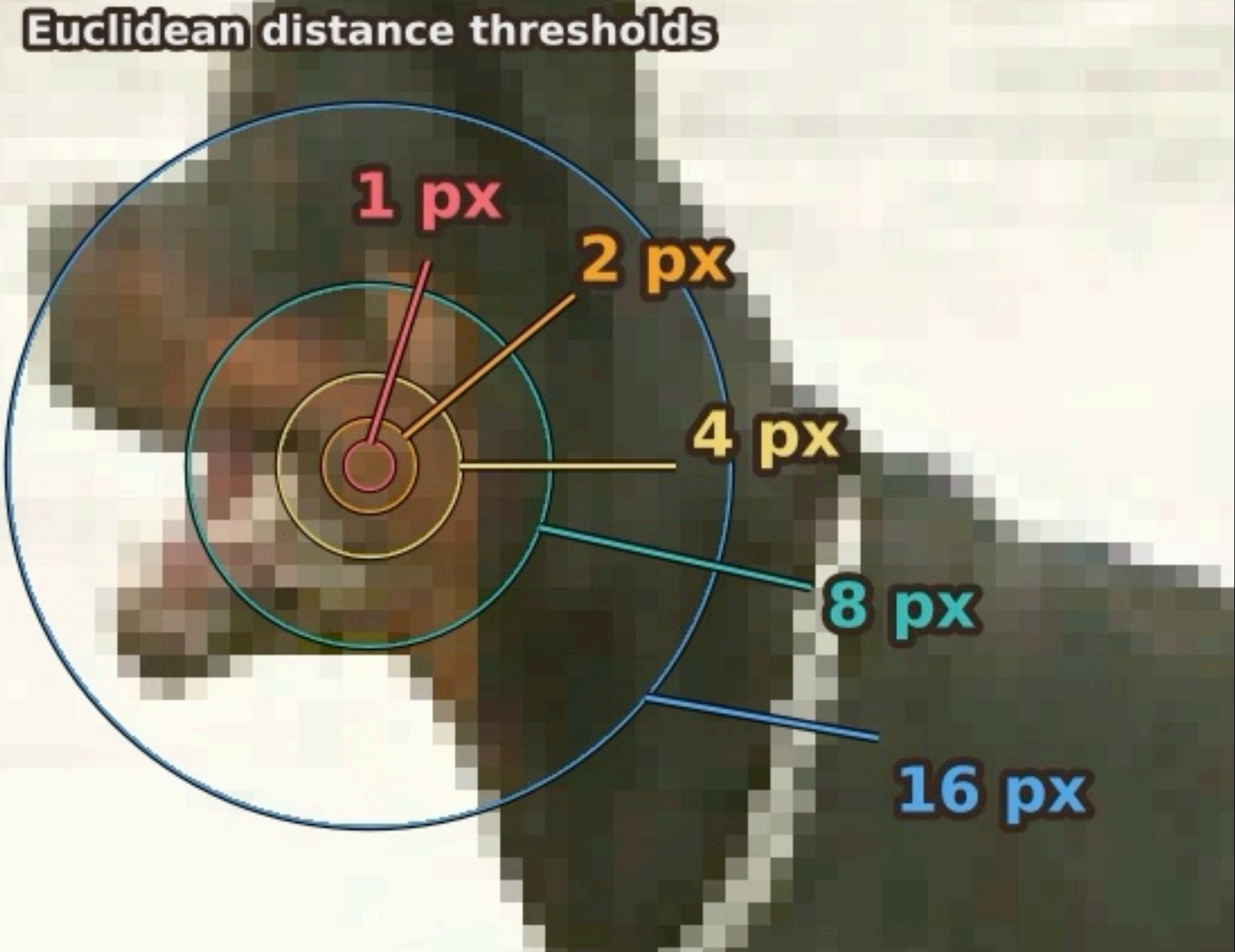
Abstract

Generic motion understanding from video involves not only tracking objects, but also perceiving how their *surfaces* deform and move. This information is useful to make inferences about 3D shape, physical properties and object interactions. While the problem of tracking arbitrary physical points on surfaces over longer video clips has received some attention, no dataset or benchmark for evaluation existed, until now. In this paper, we first formalize the problem, naming it *tracking any point (TAP)*. We introduce a companion benchmark, *TAP-Vid*, which is composed of both real-world videos with accurate human annotations of point tracks, and synthetic videos with perfect ground-truth point tracks. Central to the construction of our benchmark is a novel semi-automatic crowdsourced pipeline which uses optical flow estimates to compensate for easier, short-term motion like camera shake, allowing annotators to focus on harder sections of video. We validate our pipeline on synthetic data and propose a simple end-to-end point tracking model *TAP-Net*, showing that it outperforms all prior methods on our benchmark when

d_avg (PCK),
Occlusion Accuracy,
Average Jaccard

δ_{avg}

δ_{avg}



δ_{avg}

Where do we evaluate point tracks?

Driving: KITTI, Waymo (adapted by PIPs, DriveTrack)

Animals: BADJA, Horse10 (adapted by PIPs, AllTracker)

Robotics: RGB-Stacking, RoboTAP (adapted by TAPVid)

Egocentric: EgoPoints, Aria Digital Twin (adapted by EgoPoints, TAPVid-3D)

Capture studio: Panoptic Studio (adapted by TAPVid-3D)

Youtube: DAVIS, Kinetics (adapted by TAPVid)

Surveillance: CroHD (adapted by PIPs)

Biology: Cell Tracking, SpaceAnimal, Tri-Mouse, Schooling-Fish

Full-Suite Evaluation is Getting Difficult

oTAP) we only track points from the first available query frame, which we find gives similar results to using all possible queries and full video lengths. We argue that using a wide array of benchmarks helps flatten the effects of label noise, and gives a better sense of the overall performance in practice. We encourage future work to follow our example, but we note that our full evaluation is time-consuming (with some baselines requiring multiple days to finish their pass).

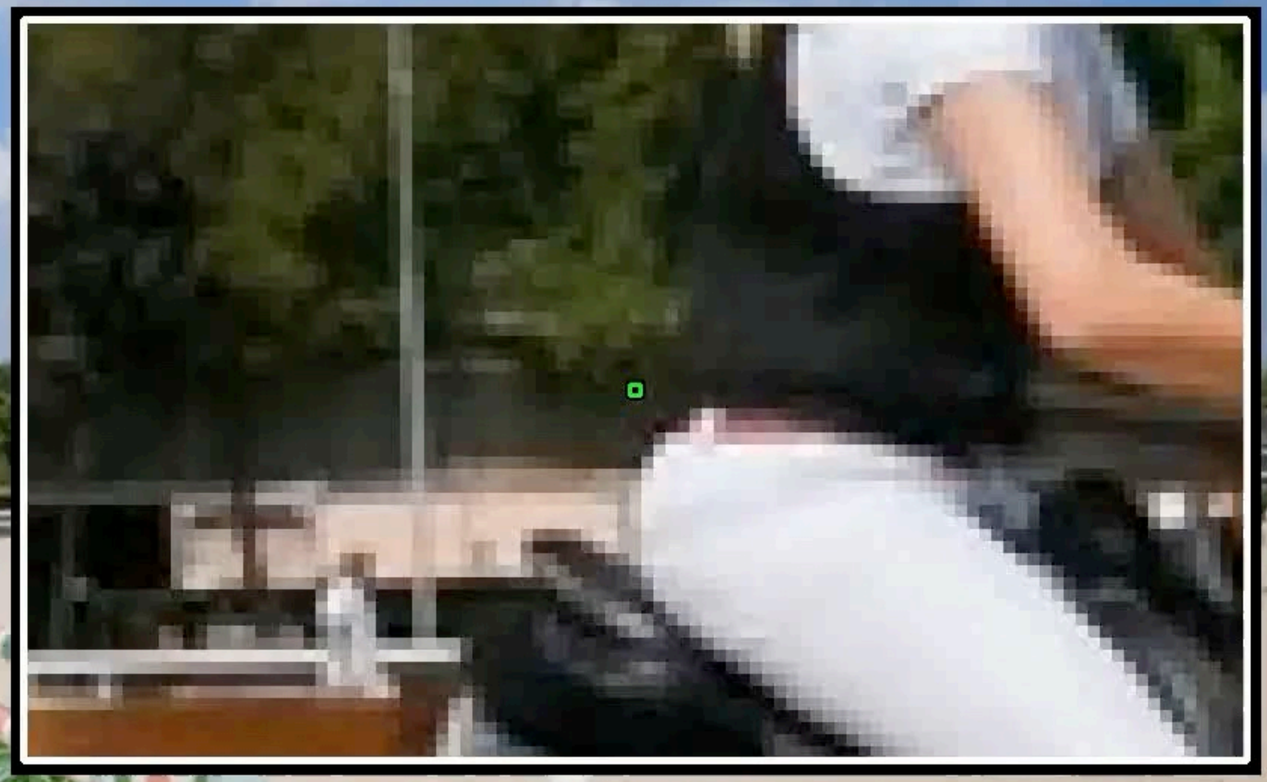
Baselines We benchmark a variety of recent point trackers and optical flow models in our tests, including the optical

Harley et al., AllTracker: Efficient Dense Point Tracking at High Resolution, ICCV 2025.


A

Annotation 20 — horsejump-high | frame 000/049 | visible

track xy=(196.47, 176.96) nearest pixel=(196, 177) RGB=(28, 30, 17)



101x61 source pixels at 8x; hollow box = nearest pixel

Annotation 14 — car—roundabout | source frame 000/074 | visible

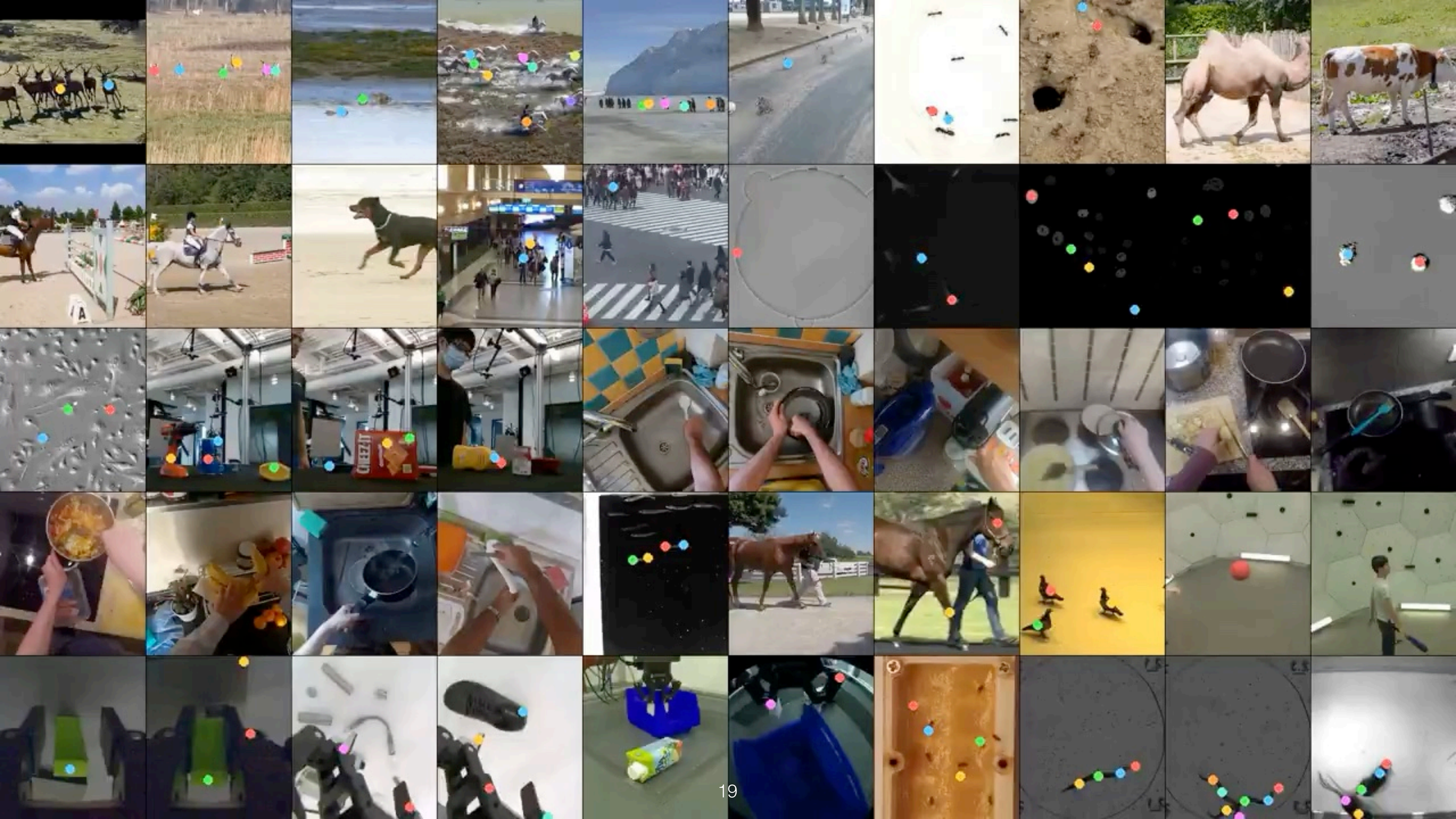
track xy=(465.48, 214.60) nearest pixel=(465, 215) RGB=(31, 33, 32)

VOTSp goals

Selective Track Curation
(Hand-picked)

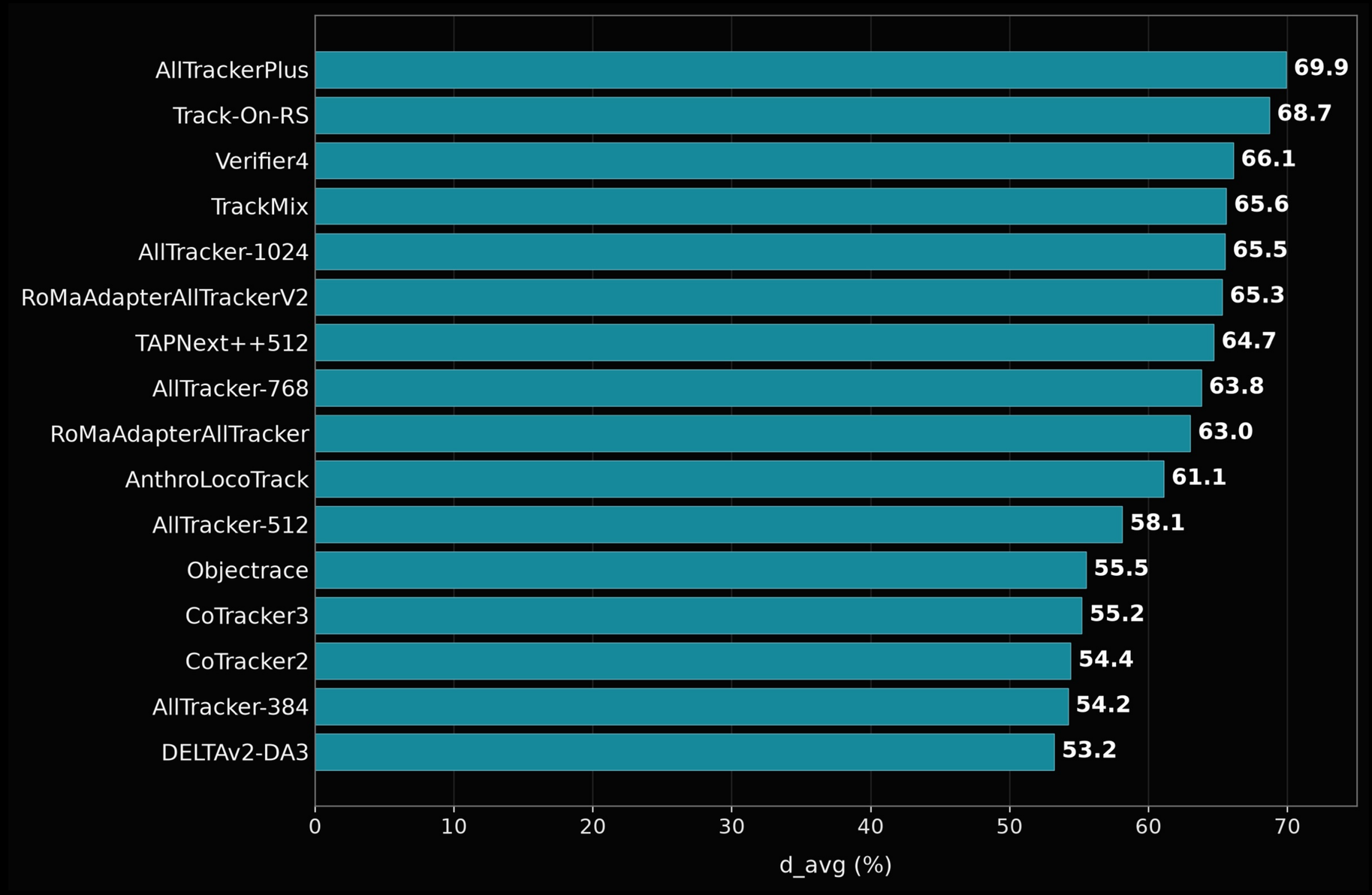
Domain diversity
(15 source datasets)

Lightweight
(50 sequences total)

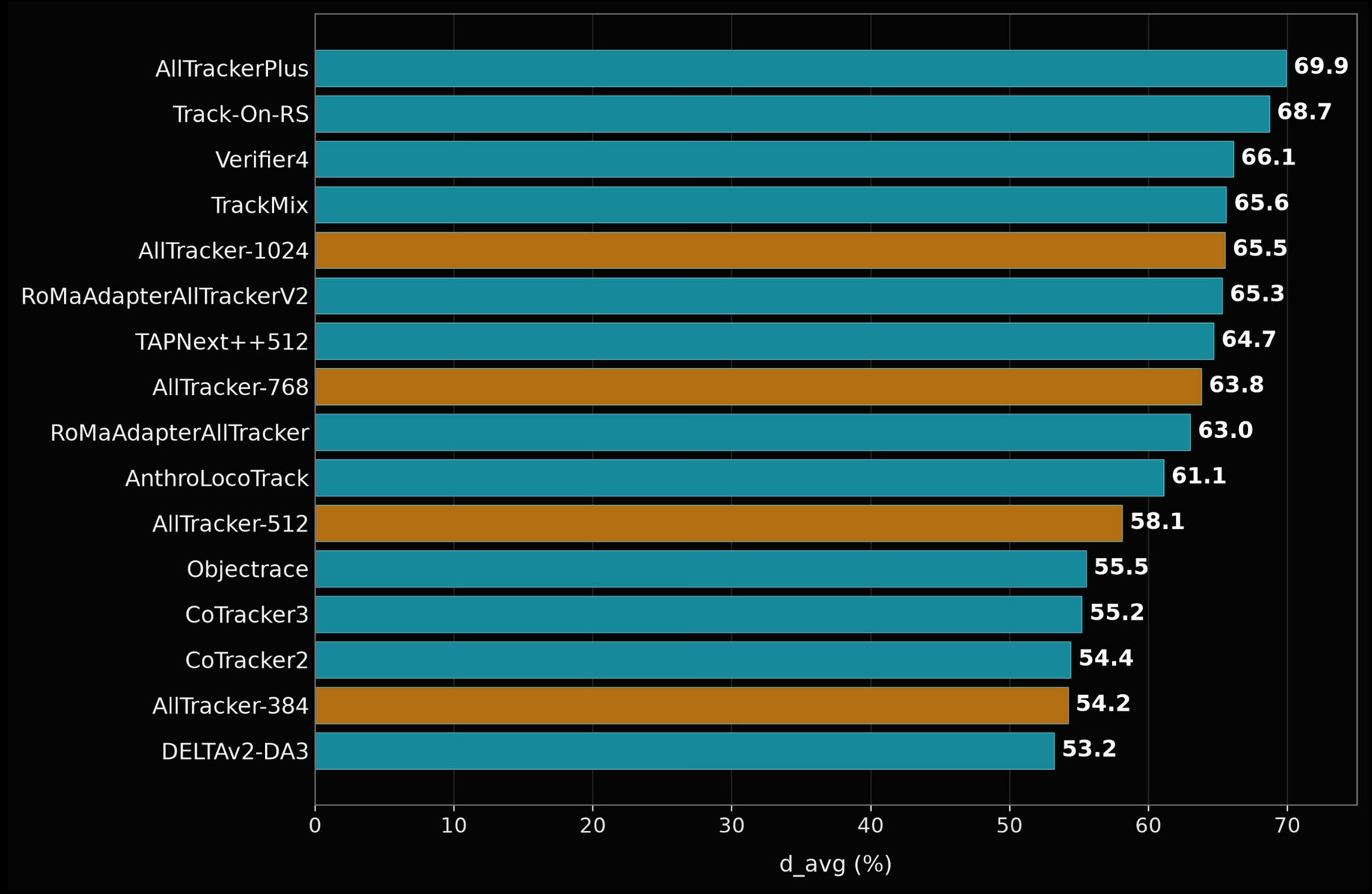


Results & Analysis

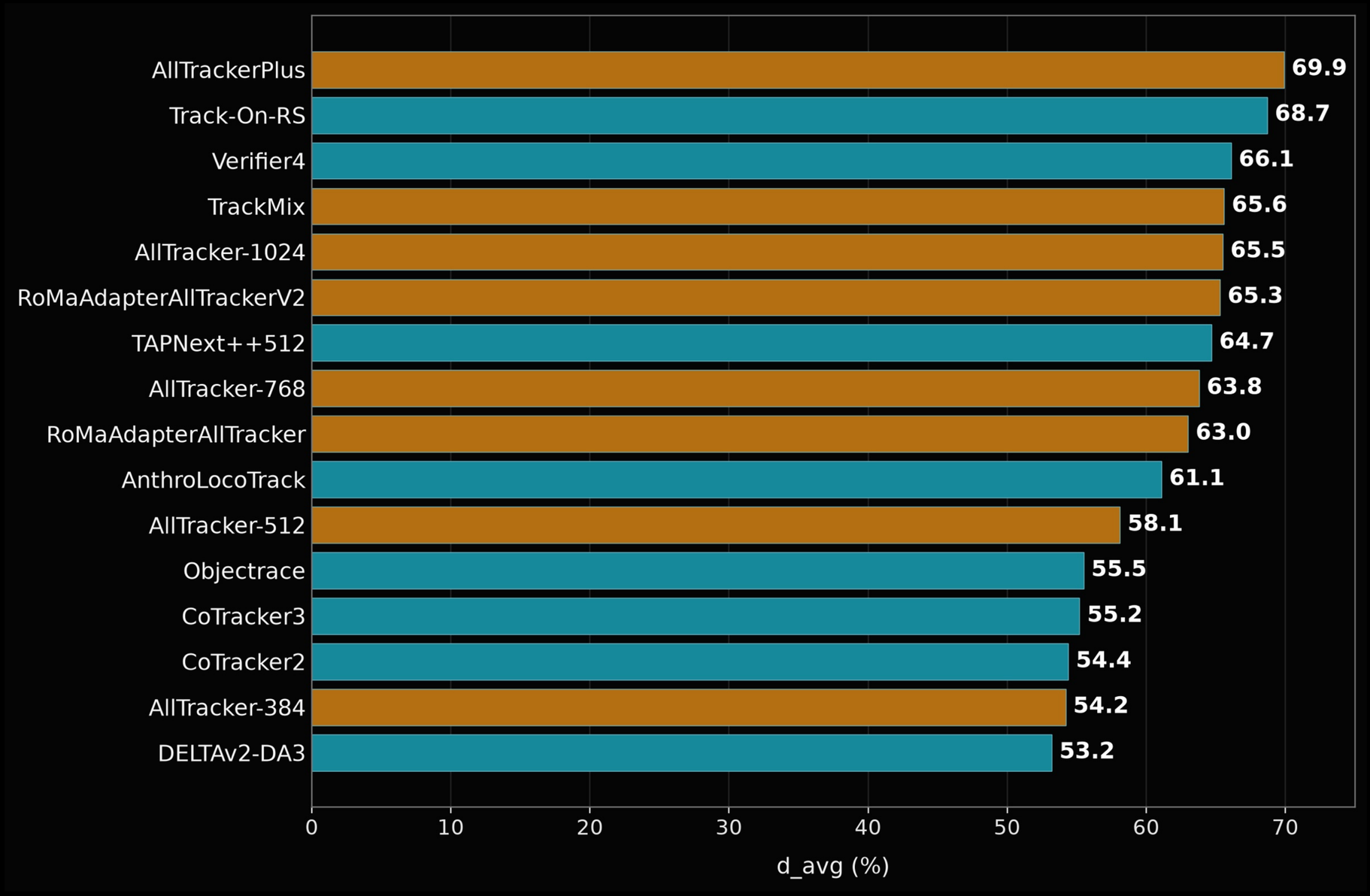
VOTSp Results: 16 submissions



Baseline: AllTracker (at multiple resolutions)



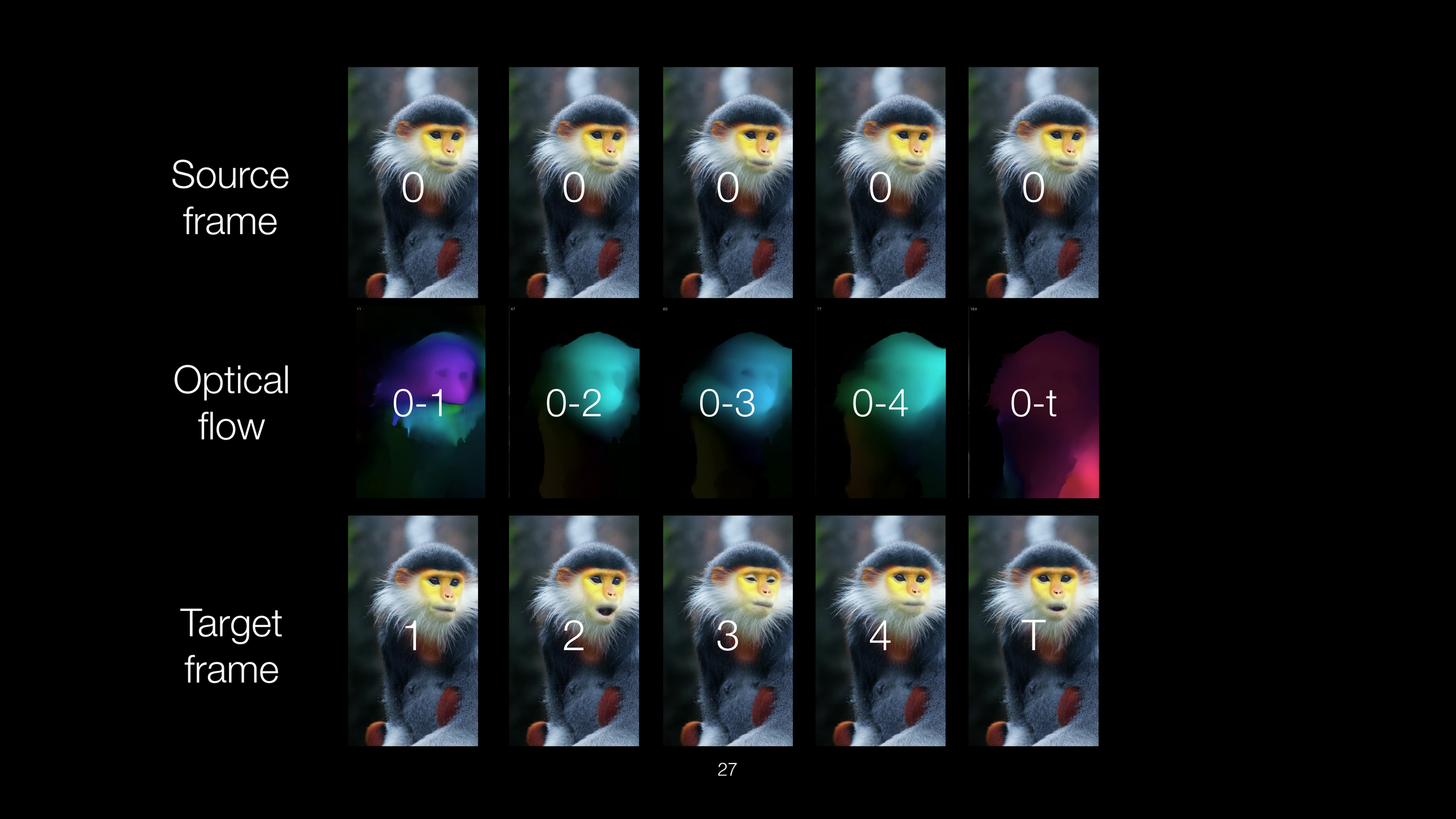
Many submissions based on AllTracker



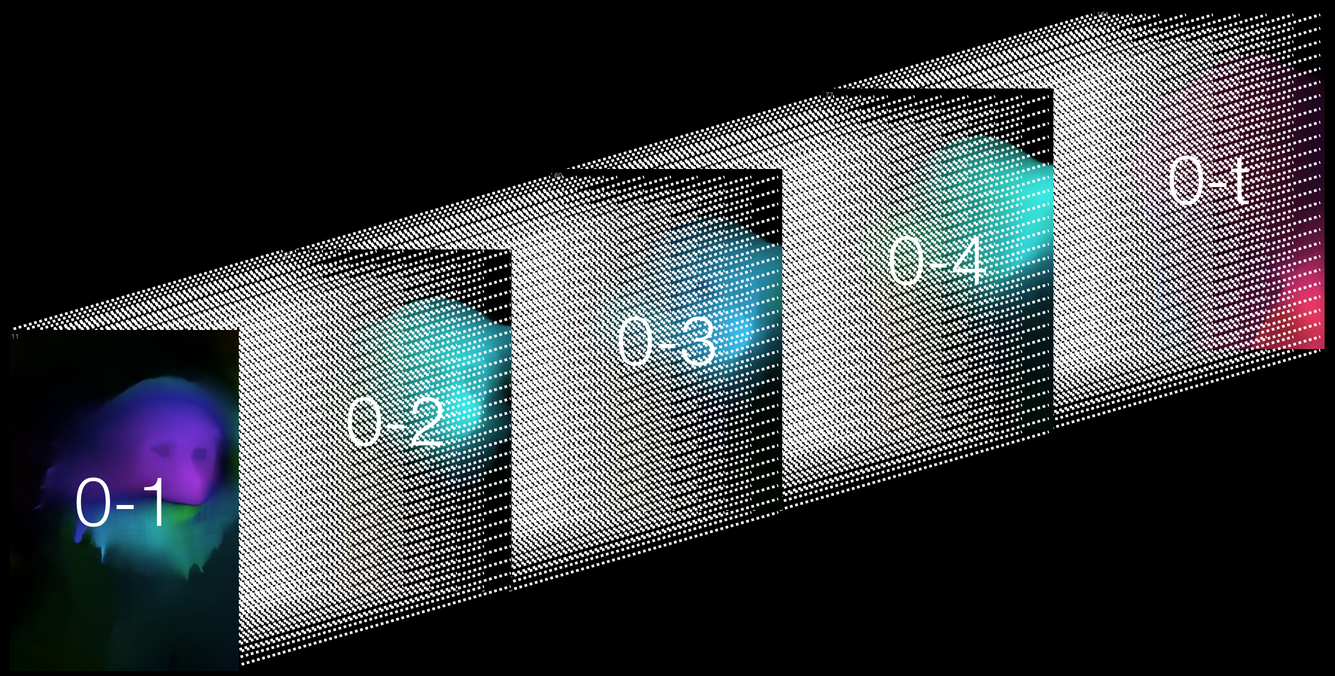
AllTracker Overview


0
1
2
3
4
T


0
0
0
0
0
1
2
3
4
T


Source frame
Optical flow
Target frame
0
0
0
0
0
0-1
0-2
0-3
0-4
0-t
1
2
3
4
T

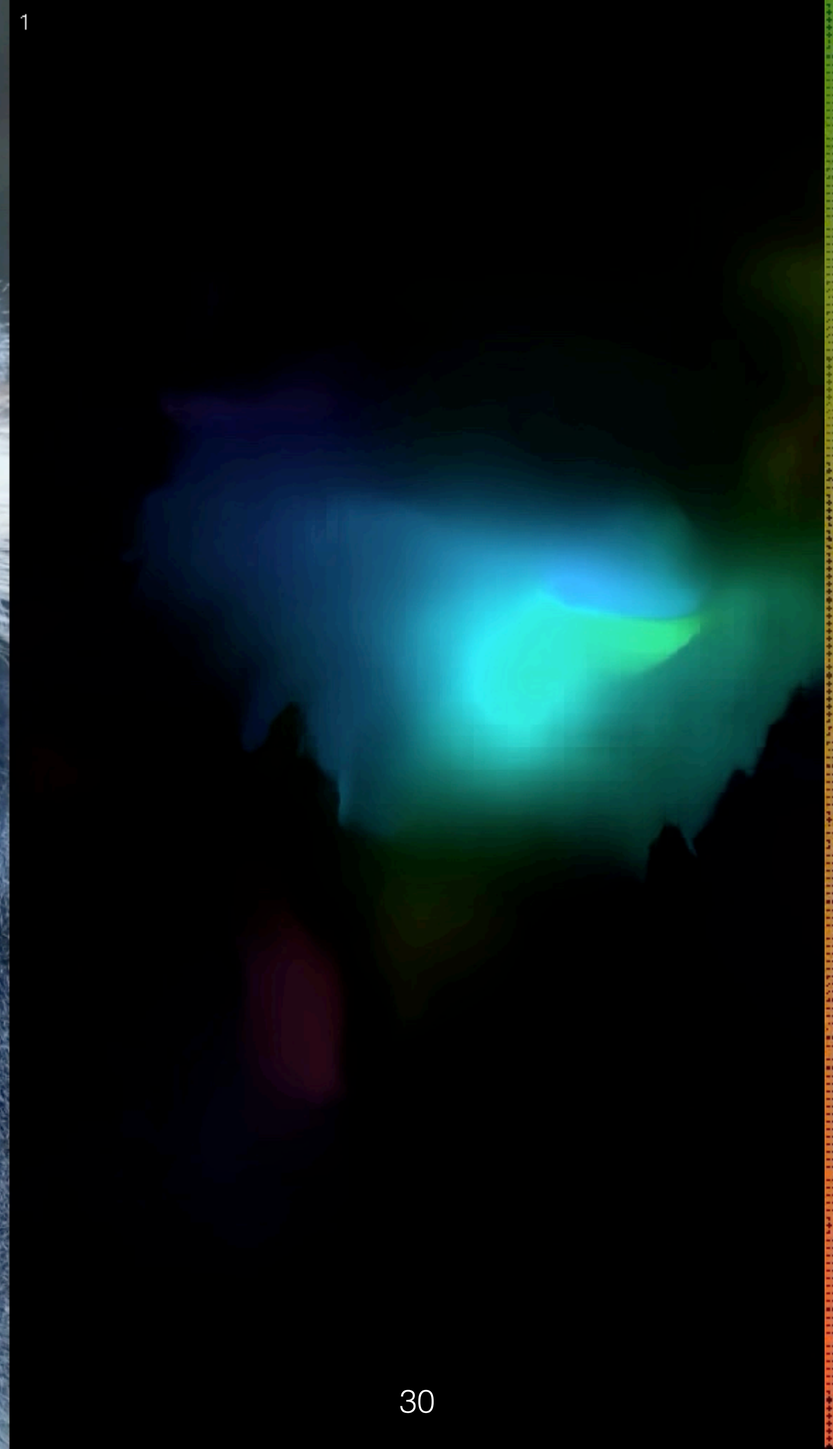
Pixel-aligned temporal attention



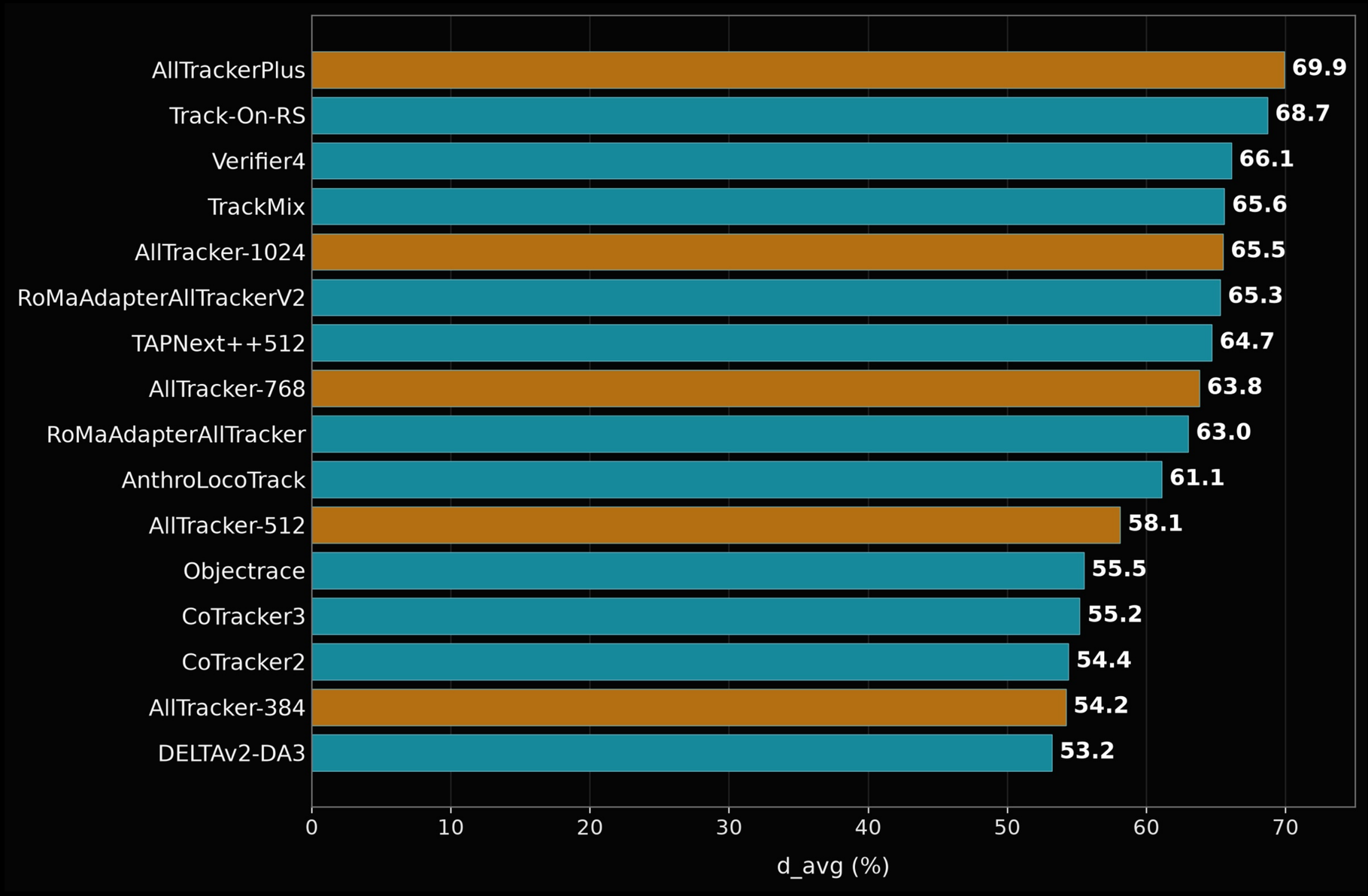
s

t

time-aware
flow

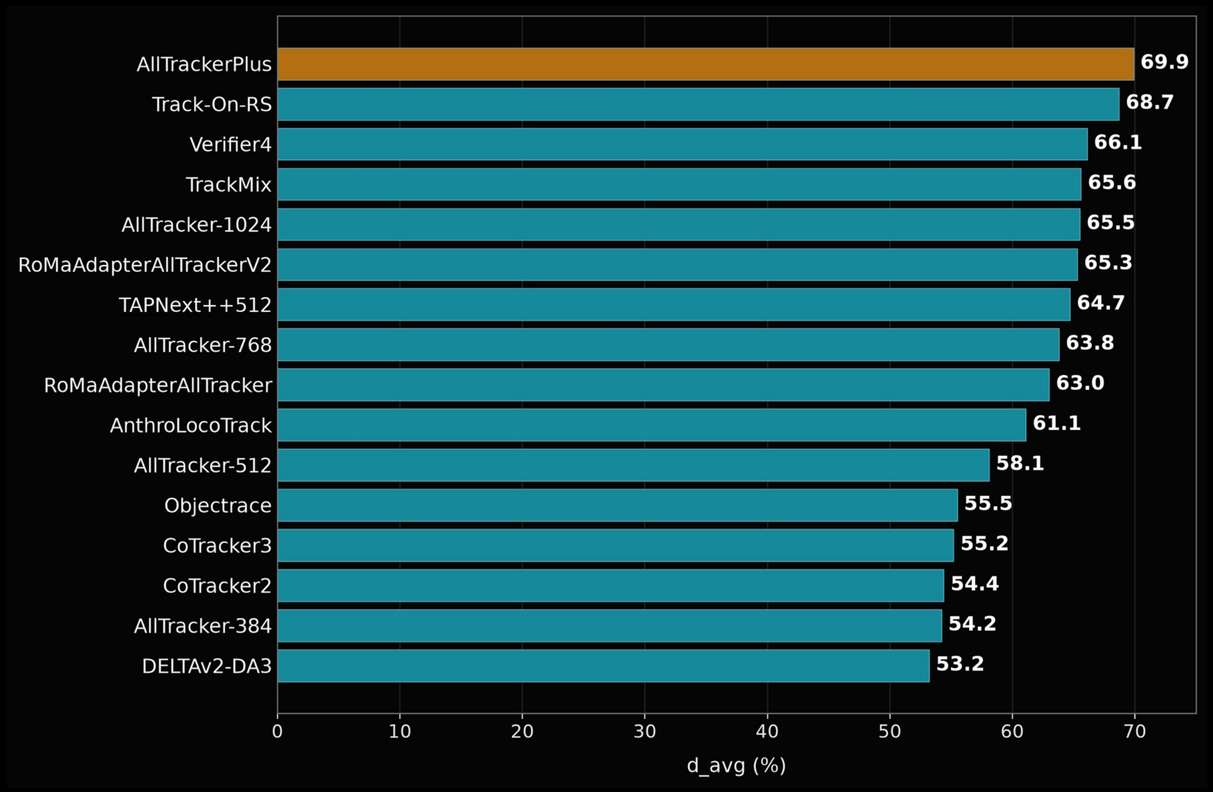


AllTracker scales well with resolution



Winner: AllTrackerPlus

Zhenglin Du, Zhengyang Li, Yi Wen (Xidian University)



Keys to victory

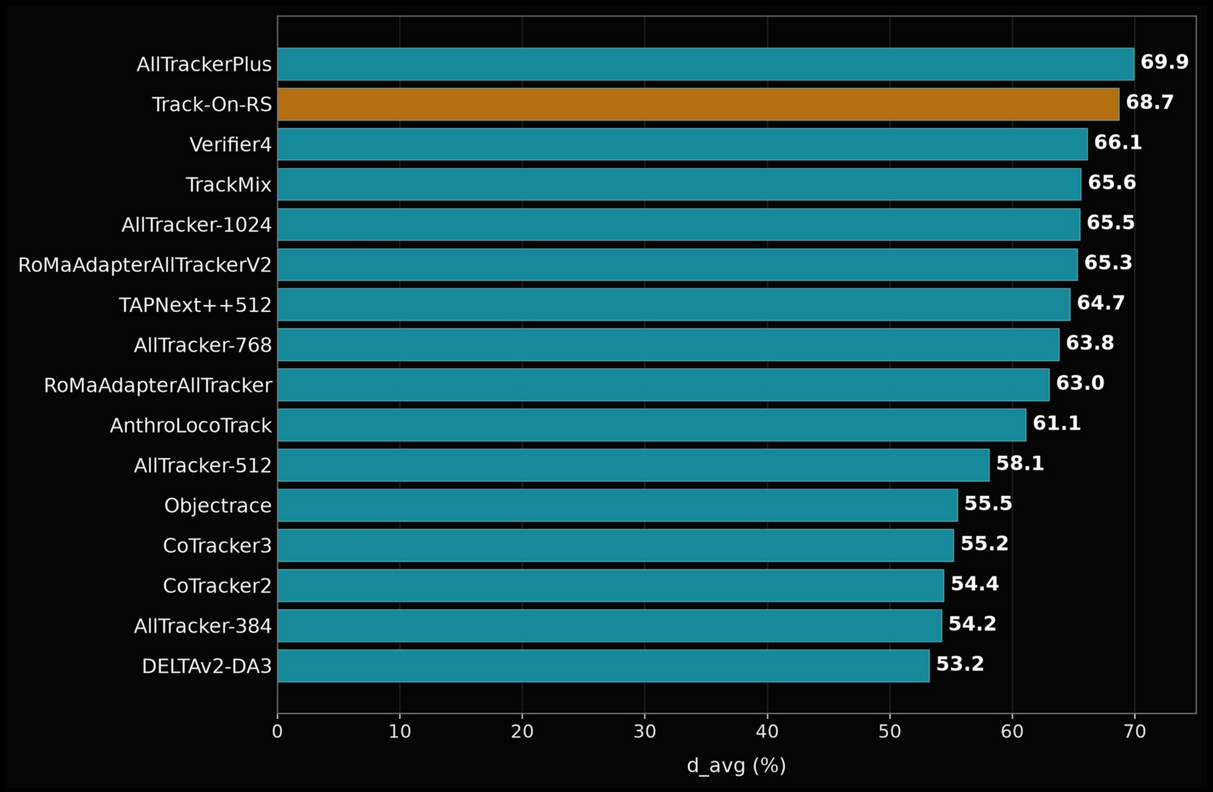
better memory engineering

forced max-resolution inference

base `alltracker.pth` weights

Runner-Up: Track-On-RS

Görkay Aydemir, Yunus Kurt, Volkan Aydıngül, Nasrin Rahimi, Misra Yavuz, Burak Biner,
Ahmet Emirdagi, Suleyman Aslan, M. Akin Yilmaz (Codeway AI Research & fal.ai)



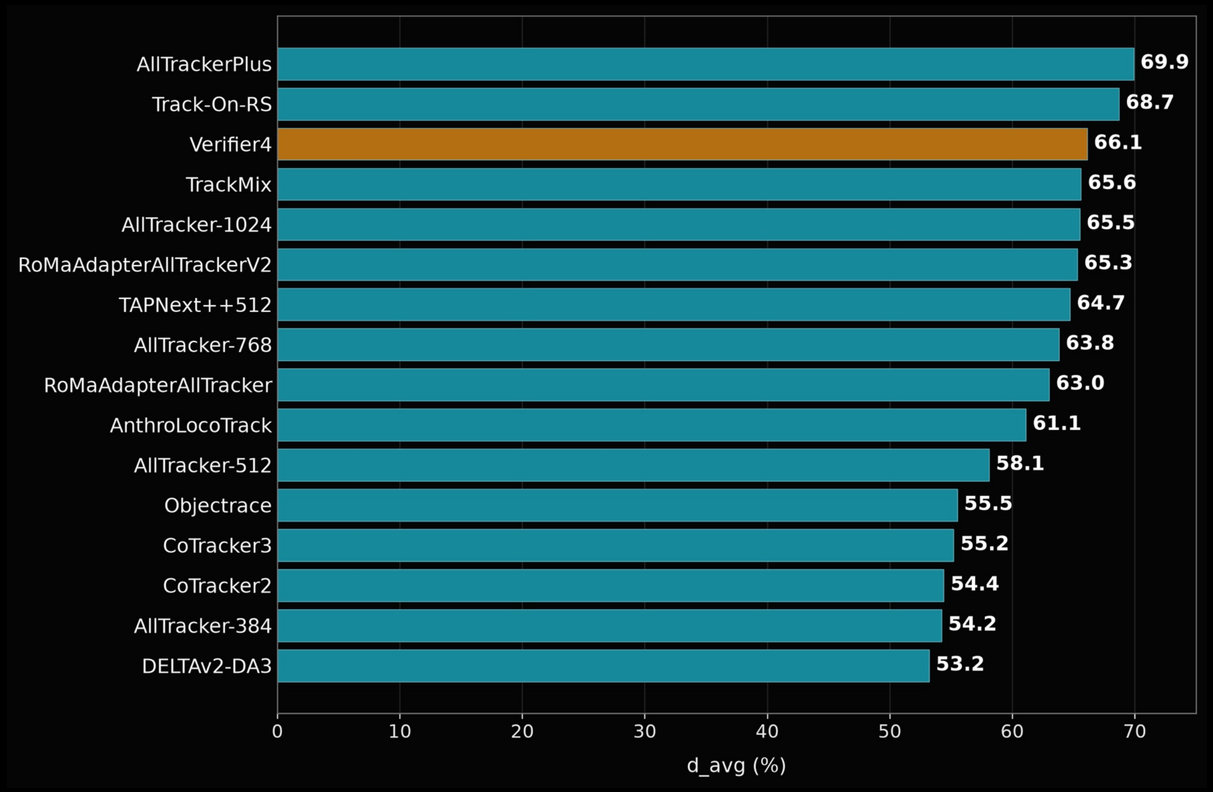
Key Mechanics

causal tracker (!)

SAM3 guidance

Highlight: Verifier4

Jinxia Xie, Liangtao Shi, Daoheng Li, Zixu Li, Jianjun Qian



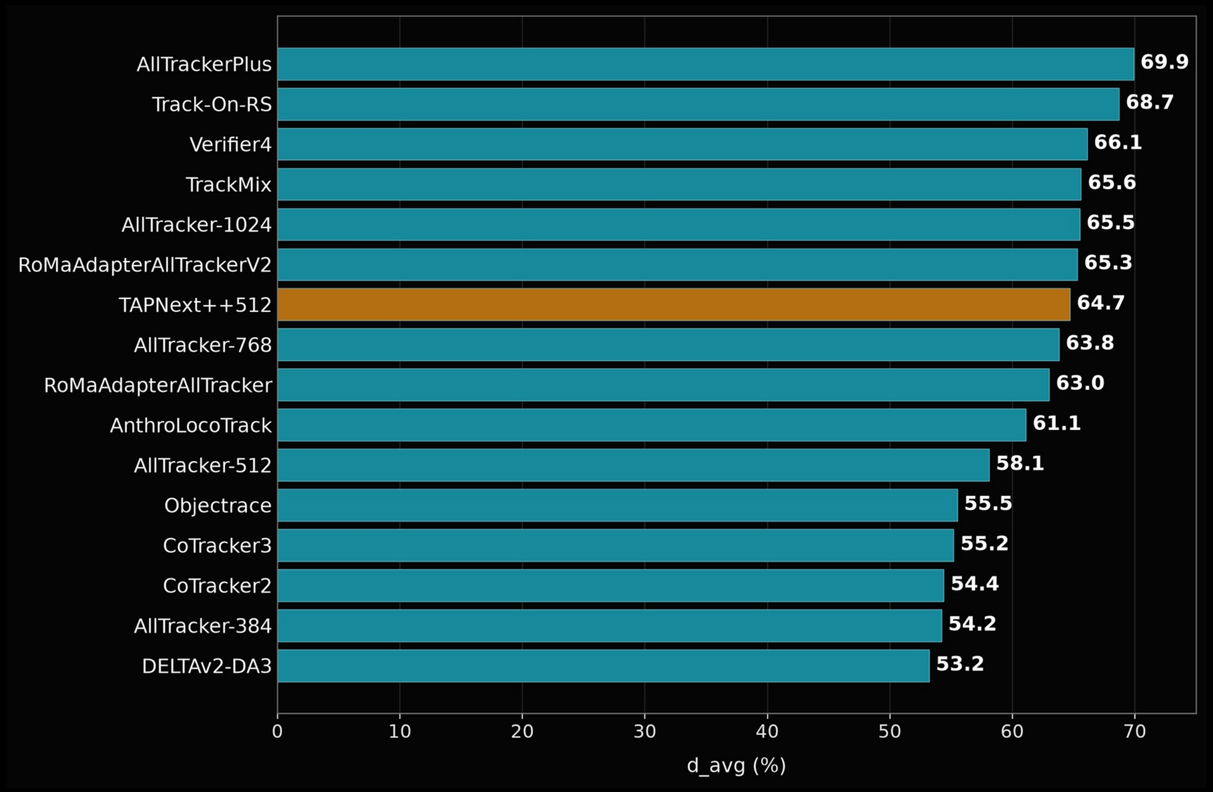
Key Mechanics

4 base trackers in parallel

verifier-guided frame arbitration

Highlight: TAPNext++512

Sebastian Jung, Martin Sundermeyer, Carl Doersch



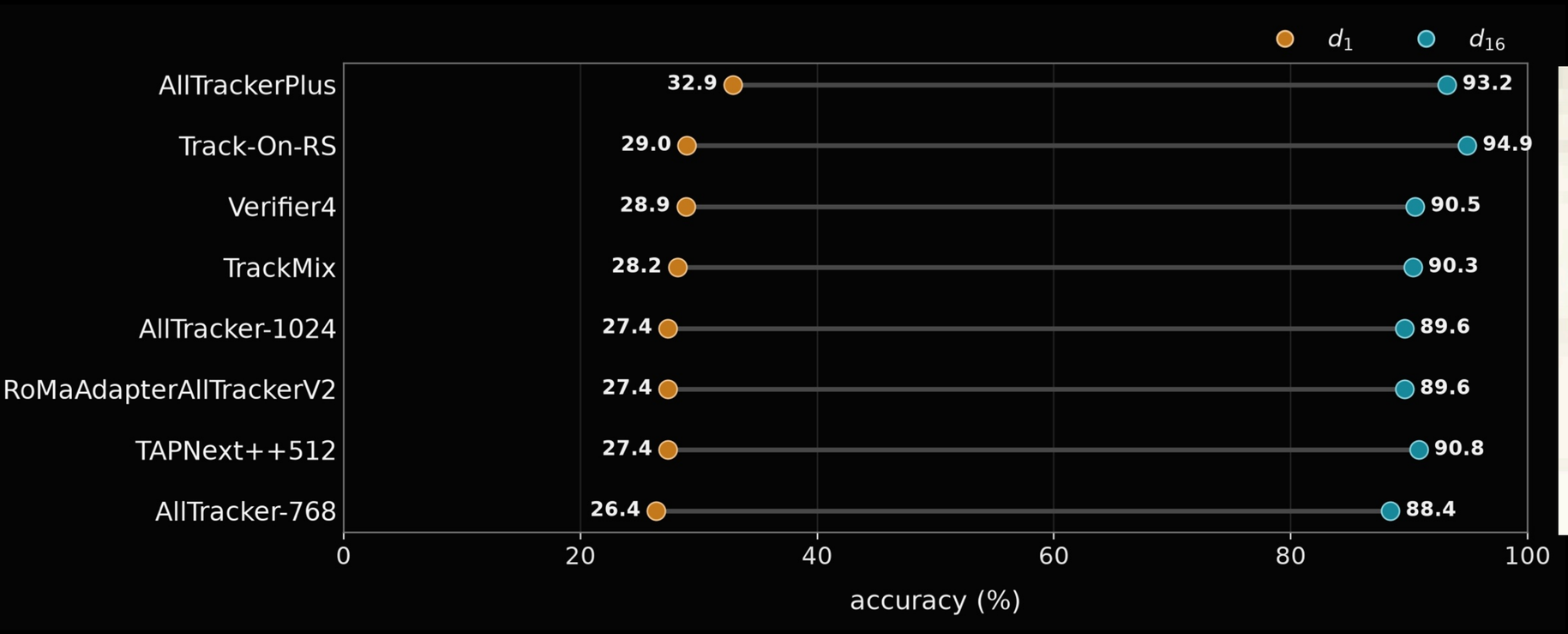
Key Mechanics

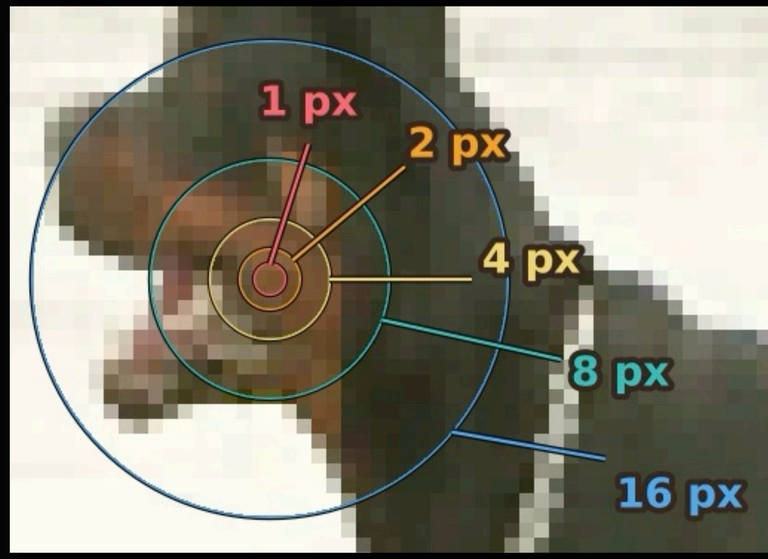
online recurrent state-space model

causal inference (!)

minimal inductive biases

All methods struggle with sub-pixel precision





Conclusion

Takeaways

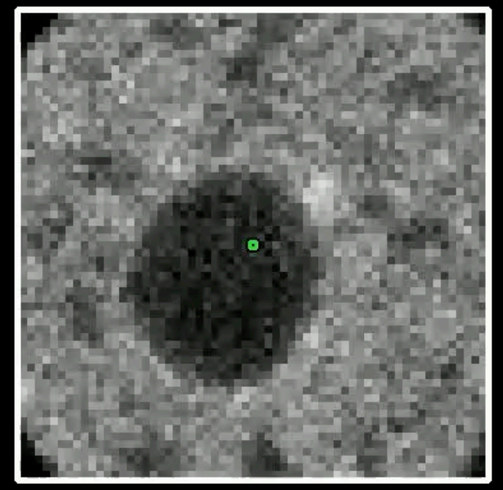
Resolution matters a lot

Causal methods are catching up

“Pure” data-driven methods are catching up

VOTSP point 1 — ctc2d_Fluo-N2DH-GOWT1_01 | frame 001/092

track xy=(78.30, 203.30) nearest pixel=(78, 203) RGB=(32, 32, 32)



61x61 px at 6x; box = nearest pixel

Limitation 1:

not all annotations are pixel-precise

Limitation 2:

position accuracy
was the only metric

point tracking

Adam Harley
Meta