CoTracker3: Simpler and Better Point Tracking by Pseudo-Labelling Real Videos

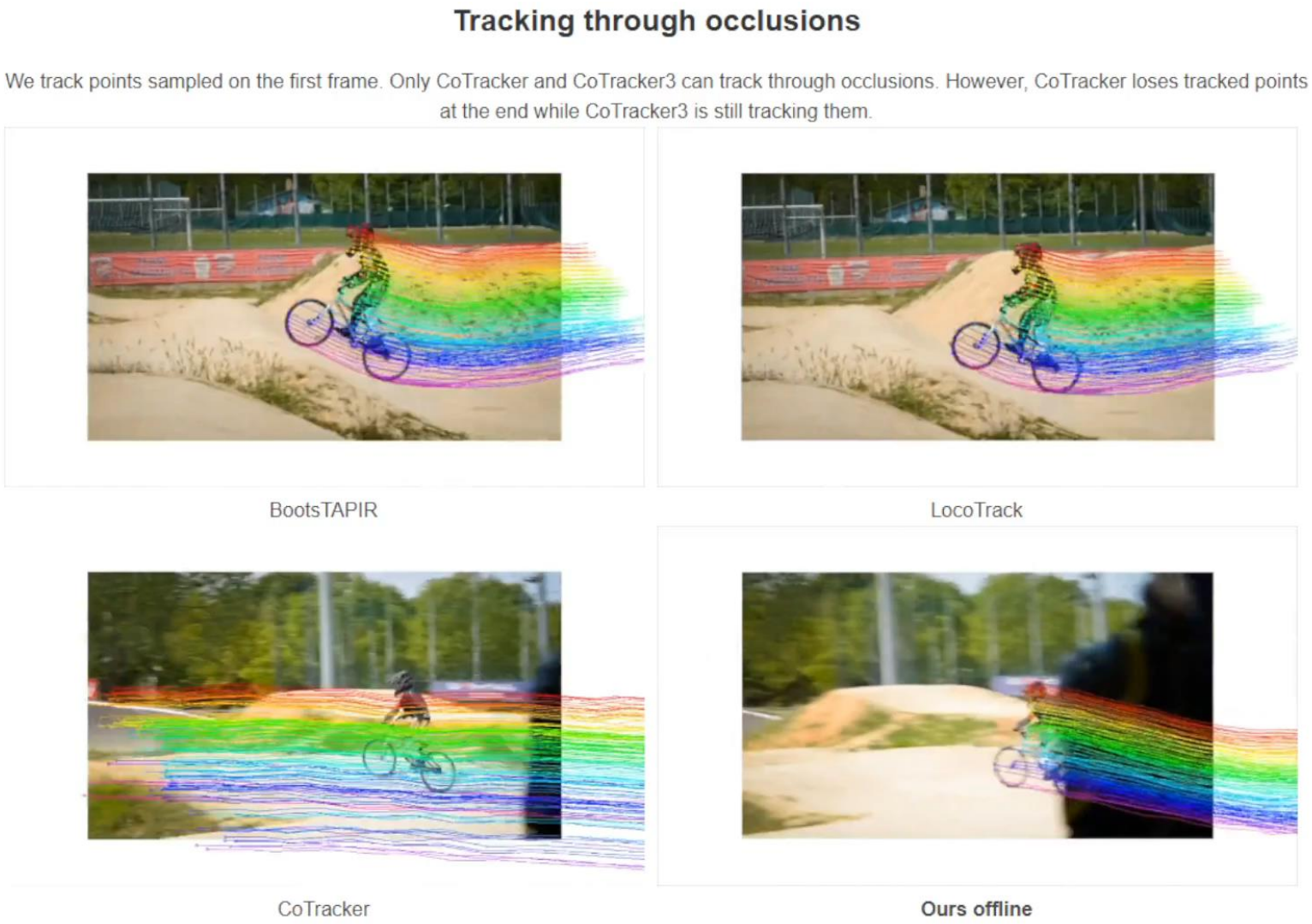
Meta AI
Visual Geometry Group, University of Oxford

(Cotracker: ECCV2025)

Outline

- Introduction
- Related Work
- Method
- Experiments
- Limitation

Abstract


Tracking through occlusions
We track points sampled on the first frame. Only CoTracker and CoTracker3 can track through occlusions. However, CoTracker loses tracked points at the end while CoTracker3 is still tracking them.
BootsTAPIR
LocoTrack
CoTracker
Ours offline

Outline

- **Introduction**
- Related Work
- Method
- Experiments
- Limitation

Introduction

- Point tracking technology is important in 3D reconstruction and video editing.
- Challenges currently faced by point tracking technology.

CoTracker3’s goal

- To simplify the model structure and enhance point tracking performance using pseudo-labels and a small amount of real videos.

Outline

- Introduction
- **Related Work**
- Method
- Experiments
- Limitation

Related Work

- TAPIR (Tracking Any Point with Initialization and Refinement)
  1. TAPIR combines global matching and local tracking, enhancing the ability to track over long periods.
  2. Its strength lies in handling fast movements and occlusion scenarios, achieving good performance on the TAP-Vid benchmark.
  3. Advantages: High accuracy, capable of handling occlusion and long-term tracking.
  4. Disadvantages: High computational resource requirements, relatively complex model structure.

Related Work

- CoTracker

1. CoTracker uses a Transformer-based architecture to track multiple points simultaneously and improves tracking performance in occlusion scenarios through Cross-Track Attention.
2. Advantages: Excellent at handling occlusion, improves tracking accuracy by leveraging the interrelationship between multiple points.
3. Disadvantages: Large model size, high computational cost.

Related Work

- LocoTrack: Local All-Pair Correspondence for Point Tracking
  1. LocoTrack introduces 4D-related features, simplifying the point tracking process and enhancing computational efficiency, making it suitable for real-time applications.
  2. Advantages: Fast computation, capable of handling large-scale feature correlations in complex scenes.
  3. Disadvantages: Lower tracking accuracy in occlusion scenarios, lacks global matching ability.

Related Work

- BootsTAPIR: Bootstrapped Training for Tracking-Any-Point

1. BootsTAPIR performs large-scale self-training using 15M unlabeled videos, combining augmentation techniques and loss masking to reduce label noise. Additionally, an Exponential Moving Average (EMA) mechanism is employed to improve model stability.
2. Advantages: Achieves state-of-the-art accuracy on the TAP-Vid benchmark, capable of handling point tracking in large amounts of real-world video data.
3. Disadvantages: High training cost, requires substantial computational resources and data volume. The self-training process is complex and relies on techniques.

CoTracker3 Breakthrough

1. Simplified Model and Efficient Pseudo-Label Learning
2. CoTracker3 simplifies the architecture and introduces a pseudo-label learning strategy.
3. Surpassing the performance of BootsTAPIR with just 15k real videos, demonstrating the potential to achieve high accuracy with fewer data.
4. Relative Advantages: Efficient, simple, low data requirements, and can be quickly applied to real-world scenarios.

Outline

- Introduction
- Related Work
- **Method**
- Experiments
- Limitation

- Training Using Unlabelled Videos
  1. Teacher Models
  2. Query Point Sampling
- CoTracker3 Model
  1. Feature Maps
  2. 4D Correlation Features
- Model Training
  1. Training Using Pseudo-label

- **Training Using Unlabelled Videos**
  1. **Teacher Models**
  2. **Query Point Sampling**
- CoTracker3 Model
  1. Feature Maps
  2. 4D Correlation Features
- Model Training
  1. Training Using Pseudo-label

Teacher Models

- Use a variety of existing trackers to label real video datasets as "**teachers**" and train a "**student**" model using pseudo-labels.
- During training, we randomly and uniformly sample a frozen teacher model for every batch, allowing the same video to receive labels from different teachers over multiple epochs.
- This prevents overfitting and enhances generalization. Teacher models remain unchanged throughout training.

Query Point Sampling

- We use SIFT to detect and bias the selection of "good-to-track" points.
- T frames are randomly sampled from each video, and SIFT generates key points for tracking.
- SIFT is chosen for its ability to extract descriptive features while filtering out ambiguous cases, improving training stability.
- If SIFT fails to detect enough points in any frame, the video is skipped to ensure data quality.

- Why is the student model better than any teacher?
  1. It learns from a larger dataset than synthetic data alone.
  2. Real video training reduces distribution shifts between synthetic and real data.
  3. Ensembling/voting reduces noise in pseudo-labels.
  4. The student inherits strengths from various teachers, excelling in different task aspects.

- Training Using Unlabelled Videos
  1. Teacher Models
  2. Query Point Sampling
- **CoTracker3 Model**
  1. **Feature Maps**
  2. **4D Correlation Features**
- Model Training
  1. Training Using Pseudo-label

Feature maps

- Dense d-dimensional feature maps with CNN for each video frame

- Feature maps Φ, k = 4 for efficiency, feature maps at S = 4 different scales

$$\Phi_t^s \in \mathbb{R}^{d \times \frac{H}{k2^{s-1}} \times \frac{W}{k2^{s-1}}}, s = 1, \ldots, S.$$

4D correlation features

Query point Q=(t^q, x^q, y^q) , in frames $t = 1, \ldots, T$

Correlation between feature vectors around the query coordinates (x^q, y^q), and feature vectors around current track estimates $\boldsymbol{P_t} = (x_t,\ y_t)$.

$$\phi_t^s = \left[\Phi_t^s\left(\frac{\mathbf{x}}{ks} + \delta, \frac{\mathbf{y}}{ks} + \delta\right) :\ \delta \in \mathbb{Z},\ \|\delta\|_\infty \leq \Delta\right] \in \mathbb{R}^{d\times(2\Delta+1)^2}, \quad s = 1, \ldots, S,$$

Feature map sampled using bilinear interpolation around $= (x_t,\ y_t)$, contains a grid of $(2\triangle+1)^2$ pointwise d-dimensional features.

4D correlation features

4D correlation

$$< \phi_{tq}^{S}, \phi_{t}^{S} > = stack((\phi_{tq}^{S})^{T} \phi_{t}^{S}) \in \mathbb{R}^{(2\Delta+1)^4}$$

Before passing feature vector to transformer, use MLP to reduce their dimensionality.

Correlations features

$$Corr_t = (MLP(< \phi_{tq}^{1}, \phi_{t}^{1} >), \dots, MLP(< \phi_{tq}^{S}, \phi_{t}^{S} >)) \in \mathbb{R}^{p^S}$$

This MLP architecture is much simpler than ad-hoc module used by Locotrack.

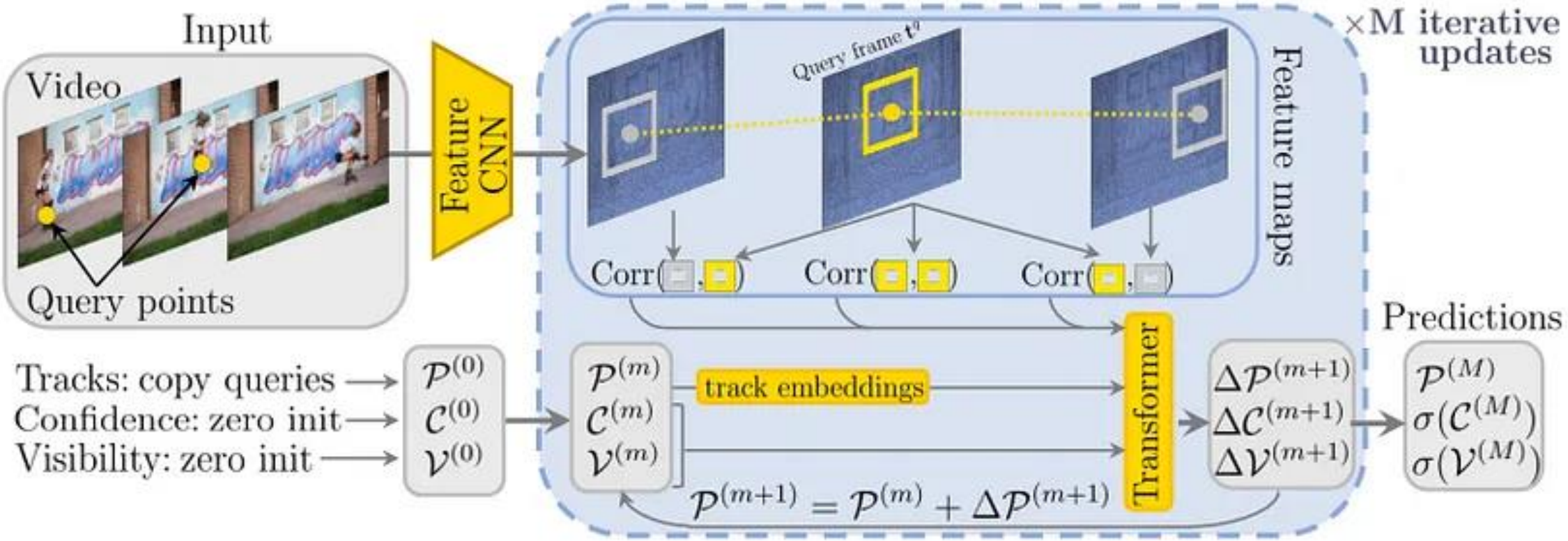


Figure 2: **Architecture.** We compute convolutional features for every frame of the given video, and then the correlations between the feature sampled around the query frame for the query point and all the other frames. We then iteratively update tracks $\mathcal{P}^{(m)} = \mathcal{P}^{(m)} + \Delta\mathcal{P}^{(m+1)}$, confidence $\mathcal{C}^{(m)}$, and visibility $\mathcal{V}^{(m)}$ with a transformer that takes the previous estimates $\mathcal{P}^{(m)}$, $\mathcal{C}^{(m)}$, $\mathcal{V}^{(m)}$ as input.

- Training Using Unlabelled Videos
    1. Teacher Models
    2. Query Point Sampling
- CoTracker3 Model
    1. Feature Maps
    2. 4D Correlation Features
    3. Iterative Update
- **Model Training**
    1. **Training Using Pseudo-label**

Training Using Pseudo-label

- Use Huber loss with a threshold of 6, and assign a smaller weight to the loss term for occluded point

$$\mathcal{L}_{\text{track}}(\mathcal{P}, \mathcal{P}^\star) = \sum_{m=1}^{M} \gamma^{M-m} (\mathbb{1}_{occ}/5 + \mathbb{1}_{vis}) \, \text{Huber}(\mathcal{P}^{(m)}, \mathcal{P}^\star),$$

where $\gamma = 0.8$ is a discount factor. This prioritises tracking well the visible points.

- Confidence and visibility are supervised with Binary Cross Entropy(BCE) loss at every iterative. Checking the predicted track is within 12 pixels for the current update

$$\mathcal{L}_{\text{conf}}(\mathcal{C}, \mathcal{P}, \mathcal{P}^\star) = \sum_{m=1}^{M} \gamma^{M-m} \, \text{CE}\left(\sigma(\mathcal{C}^{(m)}), \mathbb{1}\left[\|\mathcal{P}^{(m)} - \mathcal{P}^\star\|_2 < 12\right]\right),$$

$$\mathcal{L}_{\text{occl}}(\mathcal{V}, \mathcal{V}^\star) = \sum_{m=1}^{M} \gamma^{M-m} \, \text{CE}(\sigma(\mathcal{V}^{(m)}), \mathcal{V}^\star).$$

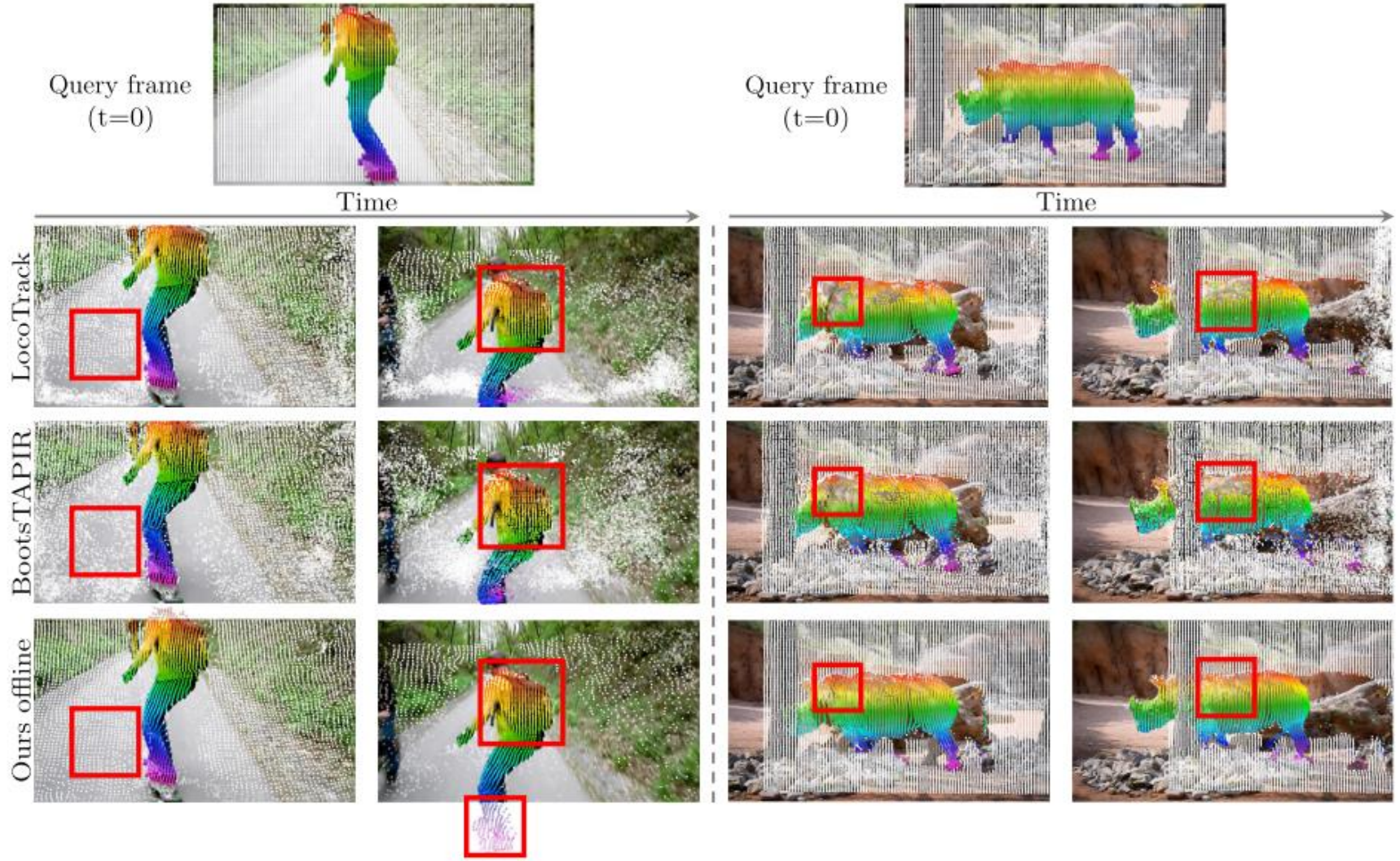
Outline

- Introduction
- Related Work
- Method
- **Experiments**
- Limitation

Experiments

Method	Train	Kinetics			RGB-S			DAVIS			Mean
		AJ ↑	δ^{vis}_{avg} ↑	OA ↑	AJ ↑	δ^{vis}_{avg} ↑	OA ↑	AJ ↑	δ^{vis}_{avg} ↑	OA↑	δ^{vis}_{avg} ↑
PIPs++ (Zheng et al., 2023)	PO	—	63.5	—	—	58.5	—	—	73.7	—	65.2
TAPIR (Doersch et al., 2023)	Kub	49.6	64.2	85.0	55.5	69.7	88.0	56.2	70.0	86.5	68.0
CoTracker (Karaev et al., 2024)	Kub	49.6	64.3	83.3	67.4	78.9	85.2	61.8	76.1	88.3	73.1
TAPTR (Li et al., 2024)	Kub	49.0	64.4	85.2	60.8	76.2	87.0	63.0	76.1	91.1	72.2
LocoTrack (Cho et al., 2024)	Kub	52.9	66.8	85.3	69.7	83.2	89.5	62.9	75.3	87.2	75.1
CoTracker3 (Ours, online)	Kub	54.1	66.6	87.1	71.1	81.9	90.3	**64.5**	76.7	89.7	75.1
CoTracker3 (Ours, offline)	Kub	53.5	66.5	86.4	74.0	84.9	90.5	63.3	76.2	88.0	75.9
BootsTAPIR (Doersch et al., 2024)	Kub+15**M**	54.6	68.4	86.5	70.8	83.0	89.9	61.4	73.6	88.7	75.0
CoTracker3 (Ours, online)	Kub+15k	**55.8**	**68.5**	**88.3**	71.7	83.6	91.1	63.8	76.3	90.2	76.1
CoTracker3 (Ours, offline)	Kub+15k	54.7	67.8	87.4	**74.3**	**85.2**	**92.4**	64.4	**76.9**	**91.2**	**76.6**

Table 1: **TAP-Vid benchmarks** CoTracker3 trained on synthetic Kubric shows strong performance compared to other models, while the online version fine-tuned on 15k additional real videos (Kub+15k) outperforms all the other methods, even BootsTAPIR trained on 1,000× more real videos. Training data: (Kub) Kubric (Greff et al., 2022), (PO) Point Odyssey (Zheng et al., 2023).



- LocoTrack and CoTracker3 are more consistent than BootsTAPIR, but neither LocoTrack nor BootsTAPIR can track through occlusions and also lose more background (1st column) and object points (3rd and 4th columns).

Cross-track attention	Dynamic Replica	
	δ^{vis}_{avg} ↑	δ^{occ}_{avg} ↑
✗	71.3	35.9
✓	**72.9**	**41.0**

Table 3: **Impact of cross-track attention on occluded tracking.** Cross-track attention improves the tracking of occluded points substantially. It also improves visible points, but the effect is smaller.

Self-training	Mean on TAP-Vid		
	AJ↑	δ_{avg} ↑	OA↑
✗	62.2	74.5	88.2
✓	**63.5**	**75.7**	**89.5**

Table 4: **Self-training.** Training CoTracker3 online on its own predictions improves the model. We use 10k real videos and train to convergence.

RT onl.	RT offl.	TAPIR	CoTr.	Mean on TAP-Vid		
				AJ↑	δ_{avg} ↑	OA↑
✗	✗	✗	✗	62.2	74.5	88.2
✓	✗	✗	✗	63.5	75.7	89.5
✓	✓	✗	✗	**64.5**	76.4	89.9
✓	✗	✓	✗	63.6	76.2	89.7
✓	✗	✗	✓	64.2	76.5	90.1
✓	✓	✓	✗	64.0	76.6	89.9
✓	✗	✓	✓	64.2	76.6	90.1
✓	✓	✗	✓	64.0	76.6	90.0
✓	✓	✓	✓	64.0	**76.8**	**90.2**

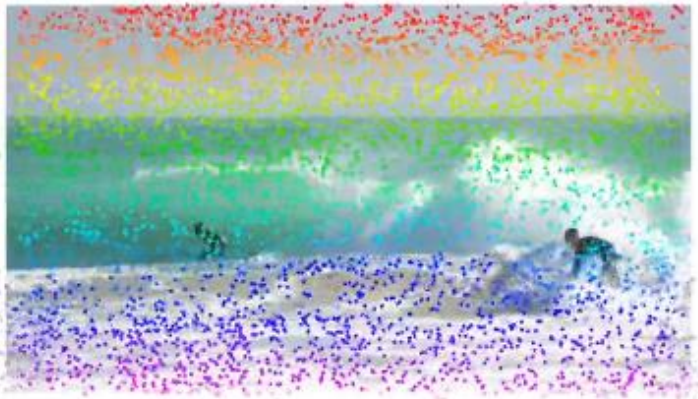
Table 5: **Models used as teachers.** We use CoTracker3 online as a student model and ablate different combinations of teacher models. The first row corresponds to the model trained only on synthetic data. The second row corresponds to self-training. Generally, the more diverse teachers we have, the better is the tracking accuracy (δ_{avg}).

Outline

- Introduction
- Related Work
- Method
- Experiments
- **Limitation**

Limitation

- Featureless surfaces is a common mode of failure:
- the model cannot track points sampled in the sky or on the surface of water.



Thank you for listening