ReTracker: Exploring Image Matching for Robust *Online* Any Point Tracking

Supplementary Material

	DAVIS			Ego-Retrack		
Training	AJ	$< \delta^x_{avg}$	OA	AJ	$< \delta^x_{avg}$	OA
S(ScanNet)	63.3	77.4	87.8	43.9	55.7	82.0
M(Megadepth)	63.3	77.3	88.1	42.3	54.5	81.9
M + S	63.5	77.6	88.0	44.3	56.6	82.2

Table 1. **Ablation of the Pre-training Dataset**.

1. Implementation Details

1.1. The Details of our model

Backbone. Following recent image matching network [2, 4], We extract 8× and 2× feature map by ResNet18 with FPN, and 16× feature by DINOv2 with Vit-L/14 backbone. We use a trainable autoencoder to encode the feature dimension from 1024 to 384. The dimensions of 8× and 2× feature map are 256 and 128, respectively.

Spatial-Temporal Attention Blocks. Following [4], the temporal block consists of a cross-attention local feature Transformer. Given features in the initial frame F_{init}, features in current frame F_{cur}, initial Frame History Tokens T_{init} and Current Frame History Tokens T_{cur}, the Spatial-Temporal Attention Blocks integrates spatial and temporal dependencies by:

$$\begin{aligned} F_{\text{init}}, T_{\text{init}} &= \text{CA}(F_{\text{init}}, T_{\text{init}}), \\ F_{\text{cur}}, T_{\text{cur}} &= \text{CA}(F_{\text{cur}}, T_{\text{cur}}), \\ F_{\text{init}}, F_{\text{cur}} &= \text{CA}(F_{\text{init}}, F_{\text{cur}}). \end{aligned} \tag{1}$$

Here, CA denotes Cross Attention Blocks. Updated features F_{init} and F_{cur} are then feed into next step. We use the Transformer Decoder introduced in [2] to decode the match distribution map of each query point in the current frame.

4D Correlation and Encoding. The 4D correlation operation is widely used in image matching methods. LocoTrack[1] is the first to introduce the 4D correlation component into the point tracking task. We utilize this technique in our pipeline, too. As described in the main paper, the 4D correlation matrix of query and current patches are encoded from $w \times w \times w \times w$ to $w \times w \times c$. Here, we employ the encode operation introduced in [2] too.

Spatial-Temporal Attention in Multi-Scale Local Refinement Block. The spatial-temporal attention block in multi-scale Local Refinement Block takes mixed 4D tokens with the shape of $B \times T \times N \times C$ as input. We employ the spatial attention in the N channel and temporal attention in the T channel, respectively.

Decoder in Multi-Scale Local Refinement Block The Decoder takes the spatial-temporal enhanced 4D tokens as input, and aggregated the information from temporal channels T and spatial channels N by weighted average operations. The mixed tokens are decoded by a tiny MLP to produce the Δ_x, Δ_y, $logit_{occ}$ and $logit_{exp}$. The outputs are supervised by the objectives described in the main text.

1.2. Image matching Pretraining

We initially pre-train our model using wide-baseline image pairs to learn correspondences between two images with co-visible areas. We use MegaDepth [3] and ScanNet++ [5] to train the matching backbone. These datasets provide estimated depth and pose generated via structure-from-motion and multi-view stereo (MVS) methods. In the image matching task, these datasets are often used to obtain pixel-level correspondence supervision between two images. For each image pair with co-visible area, we sample 1000 points with ground truth correspondence to supervise our model. The optimizer is AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate $2e-4$, and weight decay $1e-5$. We use 8 NVIDIA A100 GPUs for pre-training the matching task with a batch size of 256 for 200,000 steps. We use mixed precision provided by PyTorch-lightning framework during training.

Moreover, we observed that the model pre-trained on the ScanNet++ dataset leads to faster convergence than training on Megadepth and is prone to have better performance on downstream tracking tasks. Training on both datasets promotes the performance further.

1.3. Point Tracking Training

Subsequently, we fine-tune the model with video sequences. We pre-train the tracking backbone without temporal blocks in the Megadepth and ScanNet++ dataset. Due to the restriction of GPU memory, we train the global tracking block and local refinement block separately. For the training of the global tracking block, we initialize the parameters from pre-trained matching task and fine-tune the model on the Kubric dataset. We use 11000 sequences generated by this engine and fine-tune the whole model.

1.4. Re-track Strategy

Owing to the smoothness prior of spatial motion, the position of the query point in the subsequent frame is typically in the vicinity of the preceding frame. Therefore, global re-tracking is employed in the subsequent frame only when the local prediction confidence falls below a threshold in the last frame. For more challenging datasets, a higher re-tracking threshold is utilized, whereas for relatively simpler motion datasets, a lower threshold is selected. Specifically, the correction thresholds are set to 0.05 and 0.2, respectively.

1.5. History feature Patches Selection

During the refinement stage, we incorporate patches from multiple frames into the temporal attention Transformer. To collect representative feature patches, we employ different strategies for training and inference stage. In the training stage, we simply store the features of the nearest N frames in the memory and utilize these features for attention computation. In the inference stage, only patches with high prediction confidence are stored in the memory.

1.6. Clearification about Matching Advancements

"Matching Advancements" refers to our adaptation of the **core model components** from image matching task to point tracking. Specifically, our model adapts the strong Transformer Encoder-Decoder structure from two-view matching. This structure helps to effectively handle large viewpoint changes caused by long-term occlusion and robust re-track points.

1.7. How patches are compressed into $w \times w \times c$

We construct pairs of feature patches between the new frame and each history frame (n pairs in total). *Each* pair is processed by 4D correlation, yielding a correlation map of size $w \times w \times w \times w$. Then, we employ an encoder on this correlation map, which compress it to $w \times w \times c$.

1.8. Memory capacity for long-term tracking

Our method enables re-tracking across long gaps. The key lies in *explicitly attending new frame with the initial frame* during temporal attention processes, providing consistent context about the query points. The memory mechanism further helps bridge the feature gap among these frames.

1.9. Recent frames selection

We employ a logarithmic sampling strategy to select m frames from history frames, where frames closer to the current time are sampled more frequently.

2. Evaluation Metrics

Metrics: We evaluate tracking performance using metrics from the TAP-Vid benchmark, including Occlusion Accuracy (OA), which evaluates visibility prediction accuracy; δ_{avg}, the mean proportion of correctly tracked visible points within specified pixel thresholds (1, 2, 4, 8, and 16 pixels); and Average Jaccard (AJ), which measures both visibility and localization precision together. We adhere to TAP-Vid's standard evaluation procedure, which involves downsampling videos to 256×256 pixels.

3. Limitation and Future Work

Firstly, the computational overhead during inference is relatively high (12 fps on a single NVIDIA RTX 4090 GPU using mixed-precision), which hinders real-time performance. We will optimize our method to enable faster querying of a larger number of points, thereby improving efficiency. Additionally, our approach currently exhibits instability in regions with weak textures. We plan to enhance the robustness of our method to ensure stable performance even in such challenging scenarios.

In future work, we aim to enhance the matching capabilities of the global matching module to improve its robustness and accuracy for point tracking across various scenes. This will involve refining the module's architecture to better handle complex and dynamic environments. Additionally, we plan to optimize memory management strategies to enable the model to more efficiently utilize historical information. This will be achieved by developing adaptive memory mechanisms that dynamically select and retain the most relevant features from previous frames, thereby improving computational efficiency and tracking performance. Furthermore, we intend to integrate priors from online video object segmentation into the tracking framework. This integration will allow the model to leverage semantic information from video data, thereby enhancing its ability to understand and predict object motion in a semantically informed manner.

4. Ego-Retrack Dataset

Our dataset comprises 30 egocentric videos, with primary sources including open-access YouTube repositories and Ego4D collections (collectively accounting for 83% of samples), supplemented by custom first-person recordings of scene-specific interactions in controlled environments. The substantial majority of sequences contain 300-600 frames, providing sufficient temporal context for tracking analysis while reflecting real-world interaction durations. These sequences emphasize object reappearance challenges under significant viewpoint variations, particularly during entry/exit events where abrupt perspective shifts degrade tracking performance. Following the TAP-Vid-Kinetics annotation pipeline, expert annotators labeled 5-10 semantically consistent tracking points per video across multiple objects, preserving spatial details at the original 768×480 resolution. All data were standardized to 256×256 resolution for evaluation, aligning with the resolution conventions of mainstream tracking benchmarks.

References

[1] Seong Hun Cho, Seokju Hong, and Seungryong Kim. Local all-pair correspondence for point tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–383, Cham, 2024. Springer. 1

[2] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

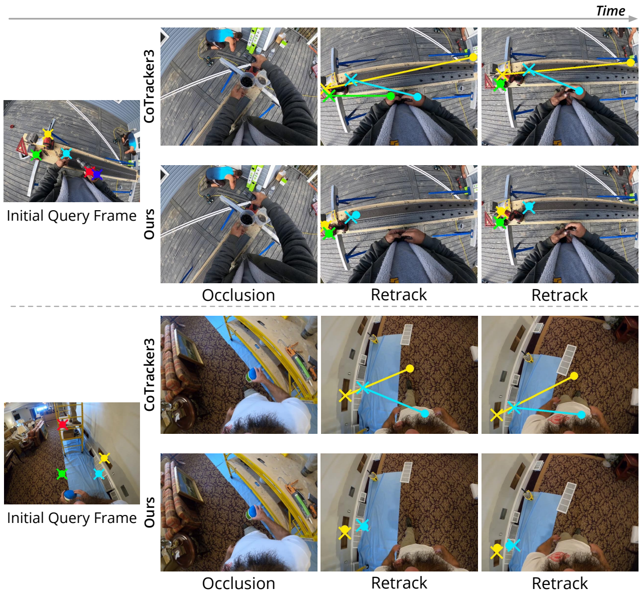

Figure 1. **More Qualtative Results.** Our method is qualtatively compared with CoTracker3 (online) on the Ego-ReTrack dataset. ✕ and ● represent ground-truth and predicted tracking locations, respectively.

[3] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2050, 2018. 1

[4] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 1

[5] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12–22, 2023. 1