TAPTRv2: Attention-based Position Update Improves Tracking Any Point

Hongyang Li[1,2] **Hao Zhang**[2,3] **Shilong Liu**[2,4] **Zhaoyang Zeng**[2]
Feng Li[2,3] **Tianhe Ren**[2] **Bohan Li**[5] **Lei Zhang**[1,2*]
[1]South China University of Technology.
[2]International Digital Economy Academy (IDEA).
[3]The Hong Kong University of Science and Technology.
[4]Dept. of CST., BNRist Center, Institute for AI, Tsinghua University.
[5]Shanghai Jiao Tong University.

Abstract

In this paper, we present TAPTRv2, a Transformer-based approach built upon TAPTR for solving the Tracking Any Point (TAP) task. TAPTR borrows designs from DEtection TRansformer (DETR) and formulates each tracking point as a point query, making it possible to leverage well-studied operations in DETR-like algorithms. TAPTRv2 improves TAPTR by addressing a critical issue regarding its reliance on cost-volume, which contaminates the point query's content feature and negatively impacts both visibility prediction and cost-volume computation. In TAPTRv2, we propose a novel attention-based position update (APU) operation and use key-aware deformable attention to realize. For each query, this operation uses key-aware attention weights to combine their corresponding deformable sampling positions to predict a new query position. This design is based on the observation that local attention is essentially the same as cost-volume, both of which are computed by dot-production between a query and its surrounding features. By introducing this new operation, TAPTRv2 not only removes the extra burden of cost-volume computation, but also leads to a substantial performance improvement. TAPTRv2 surpasses TAPTR and achieves state-of-the-art performance on many challenging datasets, demonstrating the superiority.

1 Introduction

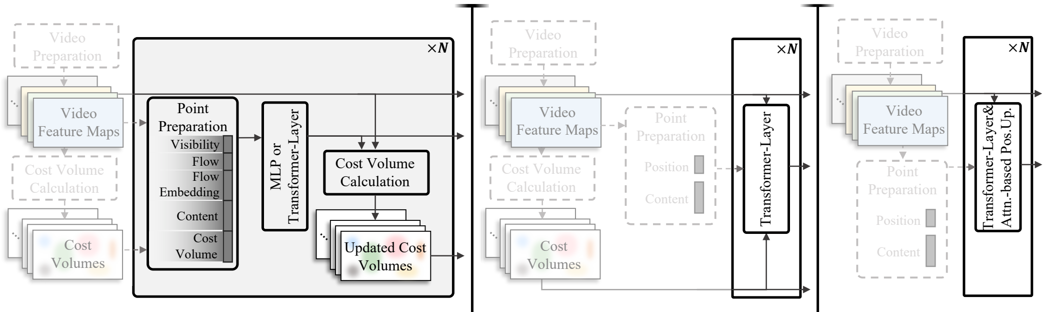
Tracking any point (TAP) in videos is a more fine-grained task compared to tracking objects using bounding boxes [29, 38, 49, 52] or their instance masks [3, 34, 48, 50, 41, 33]. As point correspondence and its visibility prediction in long video sequence is fundamental to many downstream applications, such as augmented reality, 3D reconstruction, and visual imitation [40], TAP has received increasing attention in the past few years [14, 9, 20, 26].

Some works solve TAP from the 3D perspective [28, 11, 51, 13, 22, 45, 43], where they learn an underlying 3D representation of the scene and enable it to transform over time. Although such an approach has obtained impressive results, the learning of the 3D representation is nontrivial and challenging. Thus most methods are not general and have to be fine-tuned for each video.

To develop a more general solution while keeping a good performance, some methods [9, 7, 14, 56, 32, 31] solve the TAP task in 2D space directly. Building upon existing optical flow methods [39, 47,

[*] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Previous Methods (b) TAPTR (c) TAPTRv2

Figure 1: Comparison of the frameworks among previous works, TAPTR, and TAPTRv2. Inspired by DETR-based detection algorithms, TAPTR formulates the point tracking problem as a detection problem and simplifies the overall pipeline to a well-studied DETR-like framework. After introducing the attention-based position update operation into Transformer decoder layers, the overall pipeline is further simplified to be as straightforward as detection methods. The operations within dashed boxes are executed only once.

37, 42, 19, 16, 35, 55], especially RAFT [39], such methods jointly estimate optical flow and point visibility across multiple frames. Supplemented with temporal processing methods such as sliding windows, they achieve remarkable results. However, these methods are largely affected by previous optical flow estimation methods and model each tracking point as a concatenation of multiple features, including point flow vector, point flow embedding, point visibility, point content feature, and local correlation as cost volume [26]. These features normally have clear physical meanings in optical flow, but are simply concatenated and sent as a blackbox vector to MLPs or Transformers and expect MLPs or Transformers to decipher and utilize the features [9, 7, 14, 56, 20]. Such a black box modeling not only makes the model cluttered, but also hinders its optimization and learning efficiency.

To more effectively utilize the features, TAPTR takes inspiration from DEtection TRansformer (DETR) [4, 30, 27] and models each tracking point as a point query as in DETR with a content part and a positional part (point coordinates). Each query is refined layer by layer, with its visibility predicted by its updated content feature. Point queries exchange information through spatial or temporal attention in the same frame or along the temporal dimension. Such a point query formulation not only makes the TAP pipeline conceptually simple, but also lead to a remarkable performance.

However, despite its demonstrated performance improvement, TAPTR still relies on the cost-volume feature and has a questionable design, which concatenates the cost-volume feature of a point query and its content part, followed with an MLP transformation (See Eq. 4 in [26]). As after each Transformer decoder layer, the updated point query needs to predict a relative position to update the query's coordinates, aggregating cost-volume, which is a local correlation information, to the query's content part helps the point query predict a more accurate position. However, aggregating cost-volume also contaminates the query's content part, which has two negative impacts. First, the cross-attention operation in each Transformer decoder layer needs to compute attention maps, which are the similarities between point queries and image feature keys[2]. Yet queries and keys have different formulations. While both queries and keys have their content part and positional part, queries are contaminated by cost-volume whereas keys are not. Such a difference makes the attention computation implausible. Second, a contaminated point query also yields to inaccurate cost-volume as the computation of cost-volume also needs to compare the point query with its local image features. The experiments in TAPTR show that, with such contaminated cost-volumes, the performance will suffer a big drop. Moreover, the incorporation of cost-volume in TAPTR not only results in redundant computations, but also leaves the simplicity one step behind query-based object detection methods [54, 27, 30, 25]. This raises several intriguing questions: Why is cost-volume necessary? Is there any alternative that can be developed without redundant effort? How can the cost-volume or its alternative be better utilized without contaminating a point query?

With this motivation, we propose TAPTRv2. Compared to TAPTR, TAPTRv2 does not aggregate cost-volume to queries to avoid contaminating their content features. Meanwhile, with a deeper

[2] Note that in TAPTR, deformable attention is used, which can be considered as a sparse and approximate attention as its attention weights are directly predicted based the feature of a query without comparing the query with image features. Here we use dense attention for discussion for its simple and clear definition.

analysis recognizing the importance of the information captured by cost-volume, we propose a novel Attention-based Position Update (APU) operation, which, for each query, uses its local attention weights to combine its local relative positions to predict a new query position. Such an operation is equivalent to a cross-attention operation from a point query (Q) to image features (K) using local attention, but the values are local relative positions (V) instead of image features. This design is based on the observation that local attention is essentially the same as cost-volume, both of which are computed by dot-production between a query and its surrounding features. By introducing this new operation, the TAP framework is further simplified in TAPTRv2, which not only removes the extra burden of cost-volume computation, but also yields a substantial performance improvement.

In our implementation, we follow TAPTR and adopt deformable attention for its proven efficiency and effectiveness in DETR-based detection algorithms. However, as deformable attention directly predicts attention weights for a query without comparing the query with image features, we use its variant, key-aware deformable attention [24] which computes attention weights by explicitly comparing a query with image features. Our ablation studies show that key-aware deformable attention is indeed more effective as it precisely matches the design of attention-based position update.

As shown in Fig. 1, with the help of our analysis and our simple yet effective designs, TAPTRv2 is much simpler and clearer than previous methods. To further verify the superiority of TAPTRv2 brought by our clear point query design, we conduct experiments on several TAP datasets, TAPTRv2 achieves the best performance on all of the datasets.

2 Related Work

Optical Flow Estimation. Optical flow is a long-standing problem in computer vision, which has attracted a great amount of research [15, 1, 2] over the past few decades. Particularly, in the last decade, deep learning-based methods [10, 17, 47, 37, 42, 19, 46, 53, 16, 35, 55] have demonstrated a strong advantage in this field. DCFlow [47] was the first to verify the feasibility of using cost-volume to address the optical flow problem. The robustness of cost-volume has enabled many subsequent works [37, 42, 39] and dominated this field. However, optical flow estimation methods can only handle flow estimation between two frames, which prevents them from utilizing long-term temporal information to improve accuracy. More importantly, in the presence of occlusions, optical flow methods often suffer from the problem of tracking target change. These issues make it challenging for optical flow estimation methods to process videos directly.

Tracking Any Point. The TAP task is defined to estimate the flow of any point between any two consecutive frames and predict the visibility of the tracked point in every frame in the entire video. Some works [44, 45, 36] aim to address the TAP task by constructing a time-varying 3D field. Due to the difficulty of learning a 4D field, such methods have to retrain their network to fit each video, which is normally too slow and impractical for many applications. Given the similarities between TAP and optical flow, most current methods [14, 56, 7, 9, 20] follow the optical flow methods, especially RAFT [39], but extend to multi-frame scenarios. By contrast, TAPTR [26] takes inspiration from Transformer-based object detection algorithms and models point tracking as a point detection problem, which makes TAP conceptually simple and leads to a remarkable performance improvement.

3 TAPTRv2

3.1 Overview

As shown in Fig. 2, TAPTRv2 shares a similar architecture to DETR-based object detection. More specifically, its point query bears a strong resemblance to queries designed for visual prompt-based object detection [23, 18]. Thus TAPTRv2 mainly consists of three parts, image feature preparation, point query preparation, and target point detection. To process videos of dynamic lengths, we follow previous works [14, 20, 9, 7, 26] and utilize the sliding window strategy, which divides a video into windows of lengths W and processes W frames in parallel once at a time. Since TAPTRv2 is built upon TAPTR, to make this section self-contained, we will first provide a brief overview of the TAPTR framework and then describe how TAPTRv2 improves TAPTR.

Image Feature Preparation. Our method is orthogonal to any vision backbones. In this work, we use ResNet-50 as our backbone as it is the most widely used backbone for fair comparison in

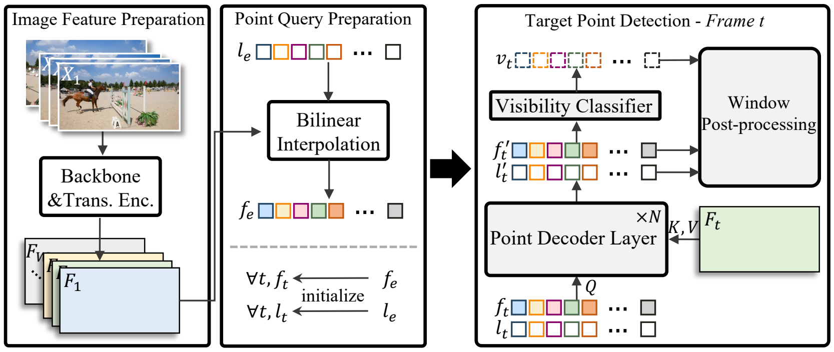


Figure 2: The overview of TAPTRv2. The image feature preparation part and the point query preparation part prepare the image features of each frame of an input video and the point queries for each tracking point in every frame. The target point detection part takes the prepared image features and point queries as input. For every frame, each point query aims to predict the position and visibility of its target point.

DETR-related research works [54, 25, 27, 30]. After obtaining multi-scale image feature maps from the image backbone, we send them into a Transformer-encoder to further enhance the features as well as the receptive fields of image features. After that, each frame X_t is ended up with a set of high-quality multi-scale image feature maps F_t,.

Point Query Preparation. Considering general TAP application scenarios, each tracking point has its unique start frame and initial position. We define their initial locations as $l_e = \{l^i_{e^i}\}^N_{i=1}$, where N is the number of points to be tracked, e^i indicates the start frame ID when the i-th tracking point first emerges or starts to be tracked. Similar to the visual prompt-based detection methods [23, 18], TAPTRv2 needs to prepare a visual feature to describe each target tracking point. Following previous methods [26, 20, 14, 9], without loss of generality, for the i-th target tracking point, its initial feature f^i_e can be obtained by conducting bilinear interpolation on the multi-scale feature maps of its start frame F_{e^i} at its initial position $l^i_{e^i}$. Then the sampled results are transformed using an MLP to fuse multi-scale information. Since the tracking of a target point across a video can be treated as detecting the target point in every frame of the video. Following the formulation of object queries in DETR-based object detection methods, for every video frame, each point query consists of a content part and a positional part, i.e. $Q^i_t = \left(f^i_t, l^i_t\right)$, which are initialized with the prepared initial feature and location of its corresponding target tracking point

$$\forall 1 \leq i \leq N, \forall 1 \leq t \leq T, Q^i_t = (f^i_t, l^i_t) \Leftarrow (f^i_e, l^i_e). \tag{1}$$

Target Point Detection in Every Frame. After preparing the image features of every frame and every point query in each frame, the TAP task can be clearly formulated as point detection. Taking the t-th frame for example, we treat its image features F_t as keys and values, the point queries (f_t, l_t) as queries, and send them to a sequence of Transformer decoder layers. In every Transformer decoder layer, both the content part and positional part of the point queries will be refined. After the multi-layer refinement, the final positional part $l^{'}_t$ of each point query is treated as the predicted position of its corresponding target tracking point in the t-th frame. Meanwhile, the content part is used to predict the visibility of the tracking point using an MLP-based visibility classifier

$$v_t = \texttt{Vis}(f^{'}_t). \tag{2}$$

Window Post-Processing. After obtaining the detection result of all point queries in a window, each tracking point's trajectory and visibility states in this window can be updated. To proceed with the next window, we use the predicted tracking point positions and their corresponding content features in the last frame of the current window to initialize point queries in the next window. This simple strategy effectively propagates the latest prediction result to the next window.

	Self	Temporal	Cost	DAVIS (Out of Domain)			Kubric (In Domain)		
Row	Attention	Attention	Volume	AJ	$< \delta^x_{avg}$	OA	AJ	$< \delta^x_{avg}$	OA
1	✗	✗	✗	47.4	62.2	82.5	79.7	87.8	94.3
2	✓	✗	✗	50.6 (↑3.2)	64.5	85.7	83.7 (↑4.0)	90.8	95.7
3	✗	✓	✗	54.3 (↑6.9)	68.3	87.0	83.4 (↑2.7)	90.6	96.5
4	✗	✗	✓	52.0 (↑4.6)	66.3	84.7	79.5 (↓0.2)	87.9	94.6

Table 1: We start with a baseline (Row 1) without using self-attention, temporal-attention, and cost-volume, and add each component from TAPTR in turn to show their impact on in-domain and out-of-domain datasets. The addition of self-attention and temporal attention leads to a significant improvement on both the in-domain and out-of-domain datasets. However, the addition of cost-volume only leads to a significant improvement on the out-of-domain dataset but a negative impact on the in-domain dataset, showing that the importance of cost-volume mainly comes from its ability to mitigate the domain gap. Note that the in-domain evaluation set is created by rendering additional 150 videos using the same setting as the training set.

3.2 Analysis of Cost Volume Aggregation in TAPTR Decoder

TAPTR regards cost-volume as indispensable and adds extra cost-volume aggregation blocks before sending point queries to Transformer decoder layers. The extra block for cost-volume not only contaminates the point queries' content feature but also makes the pipeline complex as in Fig 3 (a).

Cost Volume Aggregation. Taking the i-th point query Q_t^i in the t-th frame as an example, TAPTR conducts dot-production between Q_t^i and the image feature maps F_t of the t-th frame to obtain the point query's cost-volume C_t^i. With the help of grid sampling, TAPTR obtains the sampled cost vector c_t^i from C_t^i around the location of the point query l_t^i.

Contaminating Content Feature. After obtaining c_t^i, it is fused into the point query's content feature f_t^i through an `MLP`

$$\tilde{f}_t^i \Leftarrow \texttt{MLP}\left(\texttt{Cat}\left(f_t^i, c_t^i\right)\right), \tag{3}$$

where `Cat` denotes concatenation along the channel dimension, $\tilde{f}_t^i$ indicates the contaminated content feature. Although such a fusion makes use of the cost volume, the point query's content feature, which is expected to describe its target tracking point's visual feature, is contaminated. The contamination will further affect the calculation of cost volume in the next layer, preventing TAPTR from using more accurate cost-volume. The ablation study in TAPTR verifies that, if TAPTR updates the cost volume in every decoder layer, the performance will drop significantly.

Cost-volume Necessity Analysis. Although the use of cost-volume leads to a questionable feature contamination problem, cost-volume still contributes to the performance greatly in TAPTR. To understand the reason why cost-volume is necessary, we conduct an ablation study on TAPTR. As shown in Table 1, we remove the self-attention, temporal-attention, and cost-volume components from TAPTR's decoder, and add them one by one and observe their impact on the performance of in-domain and out-of-domain datasets. The results show that both self-attention and temporal-attention bring significant improvement on both in-domain and out-of-domain datasets. However, while cost-volume also brings a significant improvement on the out-of-domain dataset, it leads to a slightly negative effect (0.2 AJ drop) on the in-domain dataset. This contradictory result indicates that cost-volume is only essential for mitigating the domain gap and enhancing the generalization capability of the model. This is quite reasonable because cost-volume is essentially the information of similarities between features, which is why it is called correlation map in some works [20, 39, 14, 56]. Due to the domain gap, the features learned by a TAP model can hardly be generalized to out-of-domain datasets. In comparison, the correlation information is more robust to domain changes as it captures the similarity information between local features. This motivates us to design a more effective approach to utilizing cost-volume, which we find is equivalent to attention weight in essence.

3.3 Cross Attention with Attention-based Position Update

According to our analysis in Sec. 3.2, the effectiveness of cost-volume comes from its robust deep feature similarity, which is also in essence equivalent to how attention weights are computed. To leverage this insight, we still choose the deformable operation for its computational efficiency in using

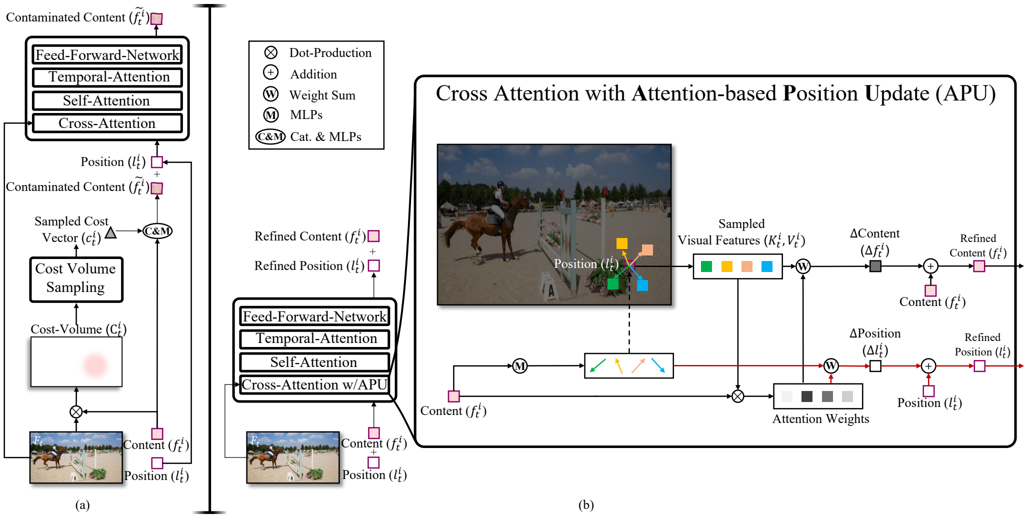


Figure 3: Comparison of the decoder layer in TAPTR (a) and TAPTRv2 (b). In TAPTR (a), cost-volume aggregation will contaminate the content feature, affecting cross-attention and leading to the contaminated cost-volume in the next layer. In TAPTRv2 (b), with the introduction of Attention-based Position Update (APU) in cross attention, not only the attention weights are properly used to update the position of each point query and mitigate the domain gap, but also the content feature of each point query is kept uncontaminated, which is crucial for visibility prediction. We use an RGB image to represent the multi-scale feature maps for better visualization.

multi-scale image features, but replace its attention prediction with key-aware attention prediction, which is called key-aware deformable attention [24].

Key-Aware Deformable Attention Revisiting. Deformable attention directly predicts the attention weights for a query without comparing the query with image features. While this design is proven effective in object detection, it is inappropriate for TAP, as we want to leverage the attention weights as a replacement of cost-volume. Using key-aware deformable attention meets this need. Taking Q_t^i as an example, key-aware deformable attention can be formulated as

$$\begin{gathered} S_t^i = W^S \cdot f_t^i, K_t^i = V_t^i = \texttt{Bili}(F_t, l_t^i + S_t^i), \\ Q_t^i = f_t^i, A_t^i = f_t^i \cdot K_t^i, \Delta f_t^i = \texttt{SoftMax}(A_t^i/\sqrt{d}) \cdot V_t^i \\ f_t^i \Leftarrow f_t^i + \Delta f_t^i, \end{gathered} \tag{4}$$

where S_t^i denotes the sampling offsets, Q_t^i, K_t^i, V_t^i and A_t^i indicate the query, key, value, and attention weights inside the attention mechanism, respectively, W^S is a learnable parameter, d is the number of key channels, `Bili` indicates the bilinear interpolation, Δf_t^i is the update of content feature. Note that, for notation simplicity, we assume there is only one attention head and F_t has only one scale.

Attention-based Position Update. Since the attention weights A_t^i in Eq. 4 reflect the similarity between the point query Q_t^i and the sampled image features (K), the attention weights and their corresponding sampling offsets imply where the target tracking point is in the current frame. Thus we combine the sampling offsets using the computed attention weights to obtain a position update, and the update will be used to update the location of the point query. This is exactly a (sparse) cross-attention operation, in which the sampling offsets are values (V). Note that to update the content part of the point query, there is another cross-attention operation, in which the sampled image features are values (V). These two cross-attention operations can use the same attention weights. However, we empirically find that the sharing of attention weights for content and position update is detrimental to model optimization. We guess the update of content and position may need different distribution of the attention weights (e.g. more spiked or more smooth). Thus, we introduce an `MLP` to work as a `Disentangler` to disentangle the weights required for content and position update. The process can

Method	DAVIS			DAVIS-S			Kinetics		
	AJ	$< \delta^x_{avg}$	OA	AJ	$< \delta^x_{avg}$	OA	AJ	$< \delta^x_{avg}$	OA
PIPs [14]	–	–	–	42.0	59.4	82.1	31.7	53.7	72.9
TAP-Net [7]	36.0	52.9	80.1	38.4	53.1	82.3	38.5	54.4	80.6
MFT [32]	47.3	66.8	77.8	56.1	70.8	86.9	39.6	60.4	72.7
TAPIR [9]	56.2	70.0	86.5	61.3	73.6	88.8	49.6	64.2	85.0
OmniMotion[43]	52.7	67.5	85.3	–	–	–	–	–	–
CoTracker-Single[20]	60.6	75.4	89.3	64.8	79.1	88.7	48.7	64.3	**86.5**
CoTracker2-All[20]	60.7	75.7	88.1	–	–	–	–	–	–
CoTracker2-Single[20]	62.2	75.7	89.3	65.9	79.4	89.9	–	–	–
TAPTR [26]	63.0	**76.1**	91.1	66.3	79.2	91.0	49.0	**64.4**	85.2
LocoTrack [6]	63.0	75.3	87.2	**67.8**	**79.6**	89.9	**52.9**	**66.8**	85.3
BootsTAP† [8]	61.4	74.0	88.4	66.4	78.5	90.7	54.7	68.5	86.3
Ours (TAPTRv2)	**63.5**	75.9	**91.4**	66.4	78.8	**91.3**	49.7	64.2	85.7

Table 2: Comparison of TAPTRv2 with prior methods. Note that, LocoTrack and BootsTAP† are concurrent works, and BootsTAP introduces extra 15M video clips for training.

be formulated as

$$\begin{aligned} \Delta l_t^i &= \texttt{SoftMax}\left(\texttt{Disentangler}\left(A_t^i/\sqrt{d}\right)\right) \cdot S_t^i, \\ l_t^i &\Leftarrow l_t^i + \Delta l_t^i, \end{aligned} \tag{5}$$

where Δl_t^i indicates the position update. Thanks to the separation of cost-volume from the content feature, the content feature can be kept clean, which leads to more accurate point visibility prediction as evidenced in Table 2. Meanwhile, our proposed attention-based position update operation deliberately utilizes attention weights as an equivalent form of cost-volume to perform position update, which effectively helps mitigate the domain gap problem.

4 Experiments

We conduct extensive experiments on multiple challenging evaluation datasets collected from real world to verify the superiority of TAPTRv2. Detailed ablation studies for our main contribution are also provided to show the effectiveness of each design in modeling.

4.1 Datasets and Evaluation Settings

Datasets. Following previous works [26, 20, 14, 9] we train TAPTRv2 on the Kubric dataset, which consists of 11,000 synthetic videos generated by Kubric Engine [12]. In each video of Kurbic, Kubric Engine simulates a set of rigid objects falling down the floor from the air and bouncing. In each video, 2,048 points on the surface of background and moving objects are randomly sampled to generate point trajectories for training. During training, for training efficiency, the resolution of the videos is resized to 512×512, and we randomly select 700-800 trajectories for training from each video. We evaluate our method on the challenging TAP-Vid-DAVIS [34] and TAP-Vid-Kinetics [5] datasets. Both datasets are from TAP-Vid [7] and are collected from real world and annotated by well-trained annotators. TAP-Vid-DAVIS has 30 challenging videos with complex motions and large-scale changes of the objections. TAP-Vid-Kinetics has over 1,000 YouTube videos, and the camera shaking and complex environment make it also a challenging dataset.

Evaluation Metrics and Settings. For evaluation, we follow the metrics proposed in TAP-Vid [7], including Occlusion Accuracy (OA) which describes the accuracy of classifying whether the target tracking points are visible or occluded, $< \delta^x_{avg}$ which reflects the average precision of the predicted tracking points' location at thresholds of 1,2,4,8,16 pixels, and Average Jaccard (AJ) which is a comprehensive metric to measure the overall performance of a point tracker from the perspective of both location and visibility classification. Meanwhile, there are two evaluation modes to accommodate online and offline trackers. The "Strided" mode is for offline trackers. The "First" mode is for online trackers and is much harder. In this paper, without specification, we evaluate our method on the "First" mode, and to facilitate comparisons with offline methods, we follow previous methods [20, 26] to further report our performance on TAP-Vid-DAVIS dataset in the "Stride" mode. Note that, since

Row	Key-Aware	Pos. Update.	Disentangle A. W.	Supervision	AJ	$< \delta^x_{avg}$	OA
1	✗	✗	✗	✗	60.0	73.1	88.6
2	✓	✗	✗	✗	60.7	73.9	89.9
3	✓	✓	✗	✗	61.7	74.8	90.4
4	✓	✓	✓	✗	62.6	75.5	91.0
5	✓	✓	✓	✓	**63.5**	**75.9**	**91.4**

Table 3: Ablation on each key design of the attention-based position updating. "Pos." is short for "Position", and "A. W." for "Attention Weights.

the resolution of the input image has a great influence on the performance, for fair comparison, we follow previous works to limit the resolution of our input image to 256×256.

4.2 Implementation Detail

We follow the previous work [26] and use ResNet-50 as the image backbone for both experimental efficiency and fair comparison. We employ two Transformer encoder layers with deformable attention [57] to enhance feature quality, and five Transformer decoder layers by default to achieve the results that are fully optimized. We use AdamW [58] and EMA [21] for training. We use 8 NVIDIA A100 GPUs, accumulating gradients 4 times to approximate a total batch size of 32, and train TAPTRv2 for approximately 44,000 iterations.

4.3 Comparison with the State of The Arts

We compare TAPTRv2 with previous methods on TAP-Vid-DAVIS and TAP-Vid-Kinetics to show its superiority in online tracking. To broaden our comparison, we also present the performance of TAPTRv2 in the "Strided" mode on DAVIS dataset (DAVIS-S). The results in Table 3 show that TAPTRv2 obtains the best performance in all of the datasets' comprehensive metric AJ. Meanwhile, the consistent improvement of OA on all datasets further verifies the importance of our designs in keeping content feature uncontaminated for more accurate visibility classification. Note that, although the concurrent BootsTAP [8] obtains remarkable performance on Kinetics, it introduces extra 15M real world video clips for training. Moreover, we still outperform BootsTAP by about 2.1 AJ on the DAVIS dataset.

4.4 Ablation Studies and Analysis

We conduct ablation studies for each key design in our main contribution to gain a deeper understanding of what specifically contributes to performance improvement. We also perform ablation on the number of decoder layers.

Ablation On The Introduction of Key-Aware Attention. We take the type of attention mechanism in cross-attention as the only variable. The results in Table 3 show that (Row 2 vs. Row 1), the introduction of the key-aware deformable attention brings 0.7 AJ improvement, which is significant. The improvement indicates that the robust attention weights obtained through dot-production helps cross-attention obtain better query results from image feature maps, thereby improving the quality of point queries' content features.

Ablation On The Position Update. To verify the effectiveness of enabling the key-aware attention weights to function in the positional part of point queries, we conduct ablation studies as shown in Table 3. The results (Row 3 vs. Row 2) show that using the attention weights for updating both the content and positional parts leads to a significant improvement (1.0 AJ). This improvement verifies that the local correlation information helps position estimation greatly, and our proposed attention-based position update is an effective operation to utilize correlation information.

Ablation On The Weight Disentangling. As shown in Table 3, decoupling the attention weights used for updating the content feature and position of a point query through an MLP enhances performance (0.9 AJ) (Row 4 vs. Row 3). This results verify that the attention weights required for the content and position parts may have different distributions, and simply mixing them confuses the network and may lead to sub-optimal results.

Ablation On The Additional Supervision. To guarantee that the attention-based position update in cross attention is always beneficial, it is important to supervise the updated positions in each decoder layer additionally. The results in Table 3 show that this extra supervision leads to a significant improvement (0.9 AJ) (Row 5 vs. Row 4), verifying its importance.

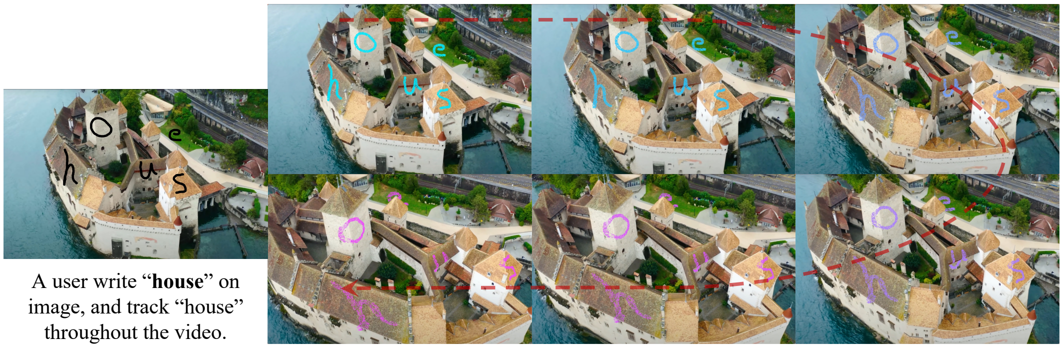
Ablation On The Number of Decoder Layers. Since our improvements over TAPTR mainly focus on the decoder, we conduct ablation studies on the number of decoder layers to verify whether TAPTRv2 still satisfies the conclusion drawn from TAPTR. The results shown in Table 4 indicate that, the performance of TAPTRv2 also improves with increased number of decoder layers, but reaches optimal performance with five decoder layers. This may be because that, with the help of the additional position update, fewer decoder layers are needed for an optimal position update result.

#Decoder Layers	AJ	$< \delta^x_{avg}$	OA
2	56.9	70.7	88.2
3	60.3	74.0	89.8
4	62.3	75.2	90.3
5	**63.5**	**75.9**	**91.4**
6	62.7	75.7	90.7

Table 4: Ablation on the number of decoder layers.

5 Visualization

Stable Tracking Results In The Wild. As shown in Fig. 4, TAPTRv2 shows its stability in point tracking and potential application in 3D reconstruction as well as video editing. More visualizations and corresponding videos please refer to our supplementary materials.



A user write "**house**" on image, and track "house" throughout the video.

Figure 4: Visualization of the tracking results of TAPTRv2 in the wild. A user writes "house" on one frame and requires TAPTRv2 to track the points in the writing area. Best view in electronic version.

6 Conclusion and Limitation

In this paper, we have presented TAPTRv2, a new approach for solving the TAP task. TAPTRv2 improves TAPTR by developing a novel attention-based position update operation to address the query content feature contamination problem caused by the inappropriate integration of cost-volume in TAPTR. This operation is based on the observation that local attention is essentially the same as cost-volume, both of which are computed by dot-production between a query and its surrounding features. With this new operation, TAPTRv2 not only removes extra burden of cost-volume computation, but also leads to a substantial performance improvement. Compared with TAPTR, TAPTRv2 further simplifies the Transformer-based TAP framework, which we hope will help the TAP community scale up the training process and accelerate the development of more practical TAP algorithms.

Limitation and Future work. For self-attention in our decoder, we currently use vanilla attention, which suffers from a computational cost quadratic to the number of queries. However, there have been many studies to reduce this cost to near linear. We will devote future research to solving it for a larger impact on dense point tracking. Additionally, TAPTRv2 aligns the frameworks of point tracking and object detection, which will facilitate the integration of multiple tasks. This will also be a topic we aim to address in the future.

References

[1] M.J. Black and P. Anandan. A Framework for the Robust Estimation of Optical Flow. In *1993 (4th) International Conference on Computer Vision*, Dec 2002. 3

[2] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *International Journal of Computer Vision,International Journal of Computer Vision*, Feb 2005. 3

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-Shot Video Object Segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017. 1

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[5] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 7

[6] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local All-Pair Correspondence for Point Tracking. In *European conference on computer vision*, 2024. 7

[7] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. TAP-Vid: A Benchmark for Tracking Any Point in a Video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022. 1, 2, 3, 7

[8] Carl Doersch, Yi Yang, Dilara Gokay, Pauline Luc, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ross Goroshin, João Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped Training for Tracking-Any-Point. *arXiv preprint arXiv:2402.00847*, 2024. 7, 8

[9] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking Any Point with per-frame Initialization and temporal Refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 1, 2, 3, 4, 7

[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015. 3

[11] Bardienus P Duisterhof, Zhao Mandi, Yunchao Yao, Jia-Wei Liu, Mike Zheng Shou, Shuran Song, and Jeffrey Ichnowski. MD-Splatting: Learning Metric Deformation from 4D Gaussians in Highly Deformable Scenes. *arXiv preprint arXiv:2312.00583*, 2023. 1

[12] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 7

[13] Yuchao Gu, Yipin Zhou, Bichen Wu, Licheng Yu, Jia-Wei Liu, Rui Zhao, Jay Zhangjie Wu, David Junhao Zhang, Mike Zheng Shou, and Kevin Tang. VideoSwap: Customized Video Subject Swapping with Interactive Semantic Point Correspondence. *arXiv preprint arXiv:2312.02087*, 2023. 1

[14] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle Video Revisited: Tracking Through Occlusions Using Point Trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022. 1, 2, 3, 4, 5, 7

[15] Berthold K.P. Horn and Brian G. Schunck. Determining Optical Flow. *Artificial Intelligence*, page 185–203, Aug 1981. 3

[16] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, KaChun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A Transformer Architecture for Optical Flow. 2, 3

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 3

[18] Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-Rex2: Towards Generic Object Detection via Text-Visual Prompt Synergy. *arXiv preprint arXiv:2403.14610*, 2024. 3, 4

[19] Shihao Jiang, Yao Lu, Hongdong Li, and Richard Hartley. Learning Optical Flow from a Few Matches. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021. 2, 3

[20] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is Better to Track Together. *arXiv preprint arXiv:2307.07635*, 2023. 1, 2, 3, 4, 5, 7

[21] Frank Klinker. Exponential moving average versus moving exponential average. *Mathematische Semesterberichte*, 58:97–107, 2011. 8

[22] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. DynMF: Neural Motion Factorization for Real-time Dynamic View Synthesis with 3D Gaussian Splatting. *arXiv preprint arXiv:2312.00112*, 2023. 1

[23] Feng Li, Qing Jiang, Hao Zhang, Tianhe Ren, Shilong Liu, Xueyan Zou, Huaizhe Xu, Hongyang Li, Chunyuan Li, Jianwei Yang, et al. Visual in-context prompting. *arXiv preprint arXiv:2311.13601*, 2023. 3, 4

[24] Feng Li, Ailing Zeng, Shilong Liu, Hao Zhang, Hongyang Li, Lei Zhang, and Lionel M Ni. Lite DETR: An Interleaved Multi-Scale Encoder for Efficient DETR. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18558–18567, 2023. 3, 6

[25] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. DN-DETR: Accelerate DETR Training by Introducing Query DeNoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 2, 4

[26] Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. TAPTR: Tracking Any Point with Transformers as Detection. *arXiv preprint arXiv:2403.13042*, 2024. 1, 2, 3, 4, 7, 8

[27] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. *arXiv preprint arXiv:2201.12329*, 2022. 2, 4

[28] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 1

[29] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-Object Tracking with Transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8844–8854, 2022. 1

[30] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for Fast Training Convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021. 2, 4

[31] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense Optical Tracking: Connecting the Dots. *arXiv preprint arXiv:2312.00786*, 2023. 1

[32] Michal Neoral, Jonáš Šerỳch, and Jiří Matas. MFT: Long-Term Tracking of Every Pixel. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6837–6847, 2024. 1, 7

[33] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9226–9235, 2019. 1

[34] 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 preprint arXiv:1704.00675*, 2017. 1, 7

[35] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, KaChun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer++: Masked Cost Volume Autoencoding for Pretraining Optical Flow Estimation. Mar 2023. 2, 3

[36] Yunzhou Song, Jiahui Lei, Ziyun Wang, Lingjie Liu, and Kostas Daniilidis. Track Everything Everywhere Fast and Robustly. *arXiv preprint arXiv:2403.17931*, 2024. 3

[37] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 2, 3

[38] Peize Sun, J Cao, Y Jiang, R Zhang, E Xie, Z Yuan, C Wang, and P Luo. TransTrack: Multiple Object Tracking with Transformer. *arXiv preprint arXiv:2012.15460*, 2012. 1

[39] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 1, 2, 3, 5

[40] Mel Vecerik, Carl Doersch, Yi Yang, Todor Davchev, Yusuf Aytar, Guangyao Zhou, Raia Hadsell, Lourdes Agapito, and Jon Scholz. RoboTAP: Tracking Arbitrary Points for Few-Shot Visual Imitation. *arXiv preprint arXiv:2308.15975*, 2023. 1

[41] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. FEELVOS: Fast End-to-End Embedding Learning for Video Object Segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9481–9490, 2019. 1

[42] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Kaihao Zhang, Pan Ji, and Hongdong Li. Displacement-Invariant Matching Cost Learning for Accurate Optical Flow Estimation. *Cornell University - arXiv,Cornell University - arXiv*, Oct 2020. 2, 3

[43] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking Everything Everywhere All at Once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19795–19806, 2023. 1, 7

[44] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking Everything Everywhere All at Once. *ICCV*, 2023. 3

[45] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. *arXiv preprint arXiv:2404.04319*, 2024. 1, 3

[46] Haofei Xu, Jiaolong Yang, Jianfei Cai, Jie Zhang, and Xin Tong. High-Resolution Optical Flow from 1D Attention and Correlation. *Cornell University - arXiv,Cornell University - arXiv*, Apr 2021. 3

[47] Jia Xu, Rene Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 1, 3

[48] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. A Large-Scale Video Object Segmentation Benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1

[49] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. TransCenter: Transformers with Dense Representations for Multiple-Object Tracking. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):7820–7835, 2022. 1

[50] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating Objects with Transformers for Video Object Segmentation. *Advances in Neural Information Processing Systems*, 34:2491–2502, 2021. 1

[51] Yuyang Yin, Dejia Xu, Zhangyang Wang, Yao Zhao, and Yunchao Wei. 4DGen: Grounded 4D Content Generation with Spatial-temporal Consistency. *arXiv preprint arXiv:2312.17225*, 2023. 1

[52] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-End Multiple-Object Tracking with Transformer. In *European Conference on Computer Vision*, pages 659–675. Springer, 2022. 1

[53] Feihu Zhang, Oliver J. Woodford, Victor Prisacariu, and Philip H. S. Torr. Separable Flow: Learning Motion Cost Volumes for Optical Flow Estimation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. 3

[54] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *arXiv preprint arXiv:2203.03605*, 2022. 2, 4

[55] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global Matching with Overlapping Attention for Optical Flow Estimation. 2, 3

[56] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. PointOdyssey: A Large-Scale Synthetic Dataset for Long-Term Point Tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 1, 2, 3, 5

[57] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv preprint arXiv:2010.04159*, 2020. 8

[58] Zhenxun Zhuang, Mingrui Liu, Ashok Cutkosky, and Francesco Orabona. Understanding AdamW through Proximal Methods and Scale-Freeness. *arXiv preprint arXiv:2202.00089*, 2022. 8

A More discussions.

A.1 Different attention weight distribution requirements.

We measured the distributions of the attention weights for content and position update in our cross attention, as visualized in Fig. 5, the distributions of these two groups of attention weights show a significant difference, indicating that the attention weights required by content and position update are different. This can also verify the importance of our weight disentangle design in APU.

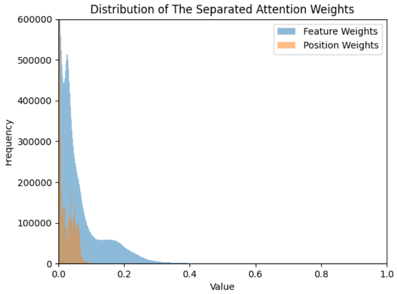


Figure 5: Attention weight distributions for feature and position updating in our cross attention.

A.2 Removing of cost-volume makes framework lightweight.

As shown in Table. 5, TAPTRv2 exhibits a faster speed and lower resource requirements compared to TAPTR. More importantly, it's a common case that in the downstream tasks, we need to track all pixels in a region (e.g. tracking a text written on the back of a horse) rather than just a few scattered points. In this case, the number of points to be tracked will reach tens of thousands. However, since the computation of the cost-volume and also the cost-volume aggregation operation in TAPTR increases sharply with the number of tracking points, with the number of tracking points increased, the advantage of TAPTRv2 will become more and more pronounced. As shown in the right table, when the number of tracking points reaches 5000 (which is only 1.9% of the pixels in a 512x512 image), the advantage of TAPTRv2 in speed and resource consumption becomes much more significant (about 24% faster and 20% fewer computational resource requirements).

800 Points	FPS	GFLOPS	#Param
TAPTR	65.9	147.2	39.2M
TAPTRv2	69.1	143.4	38.2M

5000 Points	FPS	GFLOPS	#Param
TAPTR	11.8	426.8	39.2M
TAPTRv2	14.6	354.2	38.2M

Table 5: Comparison of resource requirements between TAPTR and TAPTRv2. We evaluate TAPTR and TAPTRv2 on A100 GPU (80G), and the computational cost (GFLOPS) is calculated following detectron2.

B More Visualizations

B.1 Application of TAPTRv2 in Video Editing

Here we show the results of the video editing using TAPTRv2. After the users plot on one frame to specify the region to be edited, we sample points in the editing area and track these points across the whole video. For more details please refer to the videos in our supplementary material.

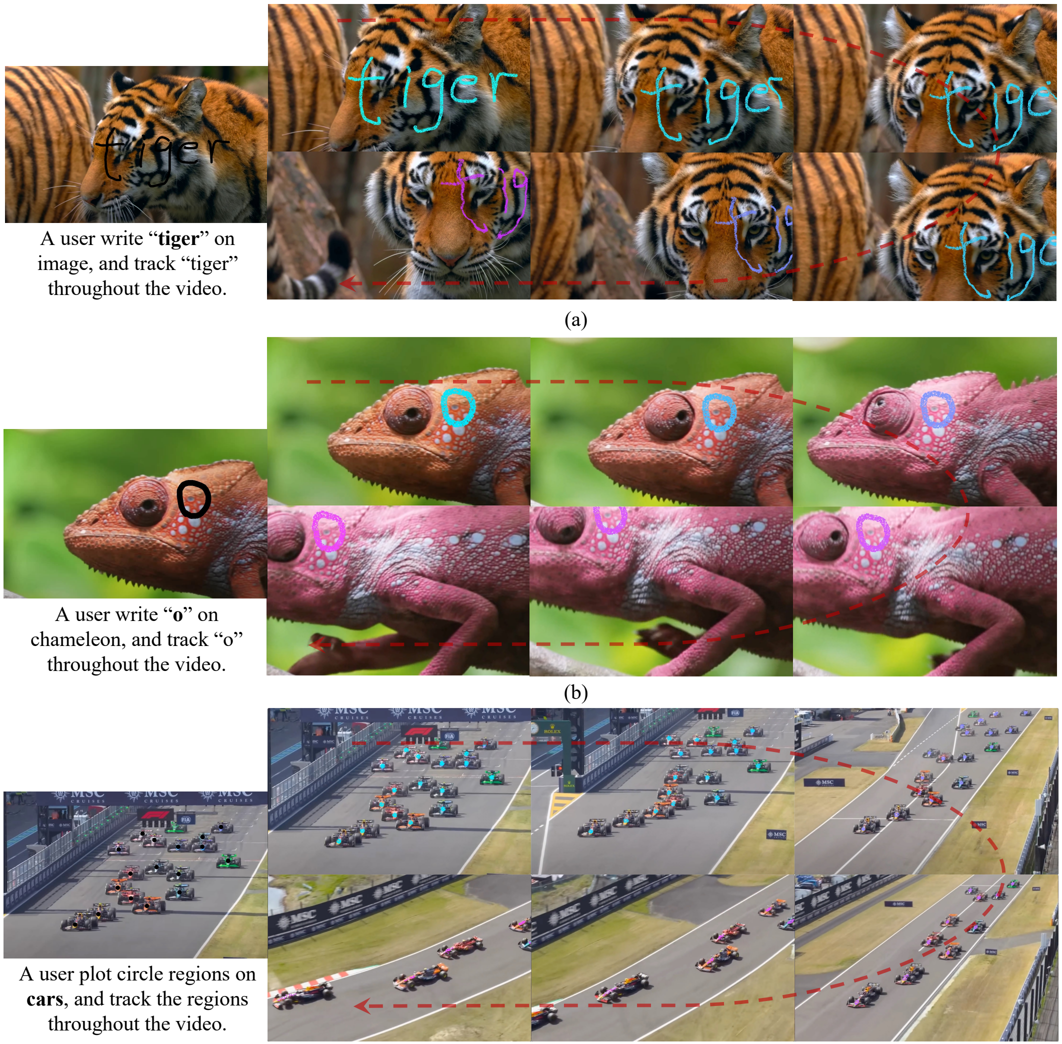
Fig. 6 (a) not only shows the ability of video editing but also the potential of TAPTRv2 in applying in 3D reconstruction.

Fig. 6 (b) shows that TAPTRv2 can handle the color change during the tracking. More importantly, although the editing area is cluttered in the middle of the video TAPTRv2 can robustly continue tracking the editing area when it reappears again.

Fig. 6 (c) shows that TAPTRv2 has the ability to handle the changing of scale.

B.2 Application of TAPTRv2 in Trajectory Estimation

In Fig. 7 we show the results of the trajectory estimation using TAPTRv2. After the users click on one frame to specify the points to be tracked, TAPTRv2 will keep tracking these points across the whole video to construct their trajectory.



A user write “**tiger**” on image, and track “tiger” throughout the video.

(a)

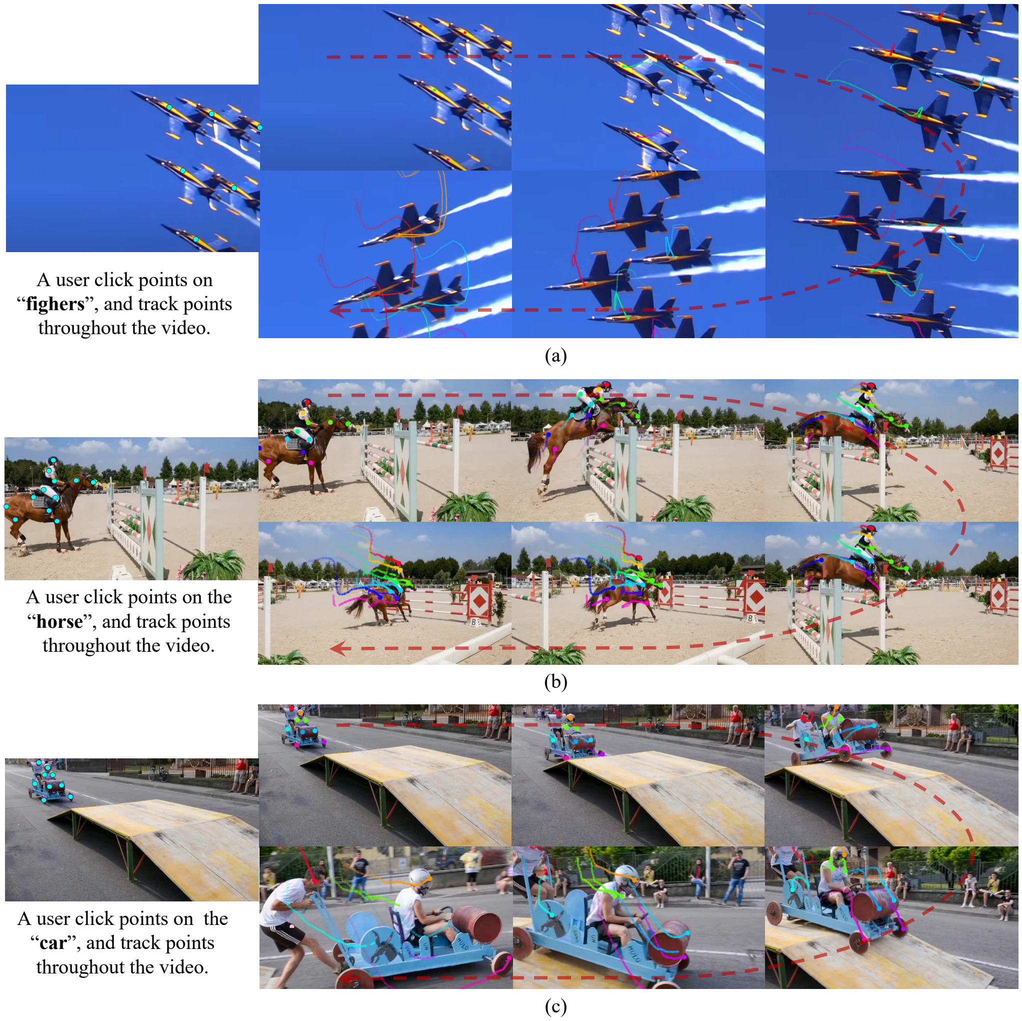
A user write “**o**” on chameleon, and track “o” throughout the video.

(b)

A user plot circle regions on **cars**, and track the regions throughout the video.

(c)

Figure 6: Apply TAPTRv2 in Video Editing. The color of the editing area changes over time.



A user click points on "**fighers**", and track points throughout the video.

(a)

A user click points on the "**horse**", and track points throughout the video.

(b)

A user click points on the "**car**", and track points throughout the video.

(c)

Figure 7: Apply TAPTRv2 in Trajectory Estimation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We show our main contributions in both abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have the limitation part in Section 6

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We do not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide our experiment details in Sec 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the data are open to be derived from the paper we cite in Sec. 4. And our code will be available after the double-blind review process.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all the detailed information in Sec 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We do multiple time experiment and calculate the average of our results. We report this in Sec 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the details in Sec 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: All the data are open to be derived from the paper we cite in Sec. 4. And our code will be available after the double-blind review process.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Tracking any point is a popular topic.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all the assets utilized in our paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All the data are open to be derived from the paper we cite in Sec. 4. And our code will be available after the double-blind review process.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.