

Exploring Temporally-Aware Features for Point Tracking

Inès Hyeonsu Kim^{1*} Seokju Cho^{1*} Jiahui Huang² Jung Yi¹
 Joon-Young Lee² Seungryong Kim¹

¹KAIST AI ²Adobe Research

Abstract

Point tracking in videos is a fundamental task with applications in robotics, video editing, and more. While many vision tasks benefit from pre-trained feature backbones to improve generalizability, point tracking has primarily relied on simpler backbones trained from scratch on synthetic data, which may limit robustness in real-world scenarios. Additionally, point tracking requires temporal awareness to ensure coherence across frames, but using temporally-aware features is still underexplored. Most current methods often employ a two-stage process: an initial coarse prediction followed by a refinement stage to inject temporal information and correct errors from the coarse stage. These approach, however, is computationally expensive and potentially redundant if the feature backbone itself captures sufficient temporal information.

In this work, we introduce **Chrono**, a feature backbone specifically designed for point tracking with built-in temporal awareness. Leveraging pre-trained representations from self-supervised learner DINOv2 and enhanced with a temporal adapter, Chrono effectively captures long-term temporal context, enabling precise prediction even without the refinement stage. Experimental results demonstrate that Chrono achieves state-of-the-art performance in a refiner-free setting on the TAP-Vid-DAVIS and TAP-Vid-Kinetics datasets, among common feature backbones used in point tracking as well as DINOv2, with exceptional efficiency. Project page: <https://cvlab-kaist.github.io/Chrono/>

1. Introduction

Point tracking aims to track any point within a casual video, which has wide-ranging applications such as robotics [46], video editing [23], and view synthesis [48]. It involves establishing correspondences of specific points across frames to determine their positions over time and whether they are visible or occluded. Accurate point tracking is challenging

*Equal contribution



Figure 1. **Chrono** is a highly precise, temporally-aware feature backbone specifically designed for point tracking. Without relying on extensive iterative refinement [11–13, 19, 25] or test-time optimization [45, 47], our model demonstrates a significant improvement over existing feature backbones such as TSM-ResNet [31].

due to complex motions, occlusions, and deformations [11–14]. Achieving robust tracking requires effective matching of points across frames and a comprehensive understanding of temporal dynamics.

In many computer vision tasks such as segmentation, object detection, and depth estimation, it is often beneficial to use pre-trained feature backbones along with task-specific heads [4, 6–8, 21]. These backbones, trained on large-scale datasets, offer robust representations that enhance generalizability to real-world data. In contrast, point tracking models often rely on simpler backbones such as ResNet [20] and are frequently trained from scratch [11–13, 19, 25] on synthetic dataset [16, 52]. This difference raises an important question: *Could point tracking similarly benefit from adopting pre-trained feature backbones?*

While these backbones are often pre-trained on single-image tasks [15, 32, 36, 45, 51], point tracking inherently demands temporal awareness to maintain coherence

across video frames. Unlike single-image tasks, point tracking requires consistent point matching over time, necessitating a model capable of capturing temporal information [11, 14, 25, 26, 30, 46]. Therefore, features used in point tracking also need to be temporally aware.

Despite its critical role, temporally-aware backbone has been relatively under-explored in this area. Early efforts, such as simply incorporating TSM-ResNet [31] in point tracking models [12, 13], yielded limited improvements. This outcome may be attributed to two factors: First, training on small synthetic datasets [16] may limit generalization to real-world data. Second, TSM-ResNet’s temporal awareness, constrained to adjacent frames, may not provide long-term temporal context for point tracking.

On the other hand, point tracking methods often adopt a two-stage process [11, 13, 14]: an initial stage that predicts coarse tracks directly from the simple feature backbone such as ResNet, followed by a refinement stage that introduces temporal information by analyzing trajectories across frames and iteratively refining the prediction. This pipeline has become standard, with refinement methods aiming to mitigate noise and enhance track smoothness [11, 14, 25, 26, 30, 46]. However, these refinement steps can be computationally demanding and query-dependent [10, 11], often requiring temporal refinement for each query point individually, which impacts efficiency. Moreover, a strong reliance on post-hoc refinement adds a substantial burden in correcting initial prediction errors from the first stage. This error could potentially be mitigated by leveraging features with built-in temporal awareness.

To address these limitations, we propose a feature backbone for point tracking that incorporates temporal awareness, dubbed as **Chrono**. Our key insights are twofold: First, we leverage robust feature representations learned from large-scale, real-world datasets. We employ DINOv2 [36], a pre-trained model known for strong feature representations across various tasks such as segmentation, classification, and localization [29, 36]. However, these representations are not directly compatible with point tracking due to a lack of temporal awareness. Second, we incorporate temporal awareness by designing a temporal adapter that enables the pre-trained backbone to process data with temporal context, without losing its learned knowledge. This enables the use of these strong feature backbones for point tracking tasks. Unlike the adjacent-frame context used in TSM-ResNet [31], our approach incorporates a temporal context that is six times longer, enabling the capture of complex dynamics and enhancing tracking performance.

Experimental results show that tracks estimated from Chrono features, computed solely using soft argmax without any learnable layers or additional temporal information after feature extraction, significantly outperform traditional feature backbones commonly used in point tracking

as well as DINOv2 [36]. Specifically, Chrono achieves a +20.6% increase in the position accuracy on the TAP-Vid-DAVIS [12] dataset compared to TSM-ResNet-18 [31]. Additionally, our method demonstrates superior efficiency and comparable performance compared to point tracking models equipped with refiners that inject temporal information post-feature extraction. Despite the absence of learnable layers after feature extraction, Chrono achieves high precision, underscoring that directly embedding temporal information within the features is both efficient and powerful. This finding demonstrates that Chrono models temporal dynamics and motion as effectively as more complex refiner-based methods.

In summary, our contributions are as follows:

- We highlight the lack of temporally-aware feature backbones and the reliance on computationally intensive refinement processes in current point tracking methods.
- We propose a feature backbone designed for point tracking that incorporates a long-range temporal adapter, enhancing temporal awareness over extended sequences.
- We demonstrate that our backbone produces accurate initial tracks in simple and effective manner, reducing the need for extensive refinement and achieving both improved performance and efficiency.

2. Related Work

Point tracking. PIPs [19] independently tracks a point by fetching a local correlation around the point estimate and gradually refining the tracking result through iteration. TAP-Net [12] utilizes a shallow TSM-ResNet [31] for point tracking, with lightweight layers added to the backbone. TAPIR [13] integrates TAP-Net with the iterative refinement from PIPs, modifying the architecture to a convolution-based model for temporal processing. Co-Tracker tracks multiple points simultaneously, modeling their dependencies with a Transformer architecture. Locotrack [11] achieves improved correspondence using local 4D correlation, inspired by dense matching literature [9]. TAPTR [30] introduces the DETR-like [6] architecture for point tracking. Another line of work employs test-time optimization with regularization, such as tracking smoothness, geometric constraints, and cycle consistency [45, 47]. While these methods focus on designing better architectures for track refinement, the exploration of more effective feature backbones has been relatively underexplored. Our work focuses on developing an improved feature backbone.

Feature backbone in point tracking. Early works on point tracking [12] explored the performance of self-supervised feature backbones, such as VFS [49], as well as temporally-aware features like TSM-ResNet [31]. However, the exploration of feature backbones has been relatively underdeveloped compared to iterative track refine-

ment [10, 11, 13, 25]. DINO-Tracker [45] explored the use of DINO in point tracking but it requires an hour of optimization for each video. Recently, [2] investigated the tracking capabilities of foundational models, such as Stable Diffusion [40] and DINOv2 [36], though the potential to extend these models for temporal applications remains unexplored.

Adapting large feature backbone. With recent advances in large-scale self-supervised training [22, 36], effectively transferring its vast knowledge has become a key challenge. Specifically, the strong features of DINOv2 have been adapted to various tasks. [42] fine-tunes DINO for video tasks in a self-supervised manner. Additionally, DINO has been applied to various correspondence tasks [1, 17, 29, 34, 43] and has demonstrated robustness in establishing correspondence and semantic segmentation [1, 18, 35], showcasing its rich semantics. We focus on adapting this representation to the point tracking task, enhancing its temporal consistency while preserving its pre-trained knowledge.

3. Method

In this section, we introduce our temporally aware feature backbone. We begin by formally defining the point tracking problem and its associated challenges. We then describe the design of our temporal adapter and its integration with a pre-trained backbone to incorporate temporal awareness. Finally, we present our approach for adapting the pre-trained backbone and training the temporal adapter for point tracking.

3.1. Preliminaries and Motivation

Task definition. Point tracking [10–14, 25, 30] in videos involves establishing correspondences of specific points across frames, determining their positions over time, and identifying whether they are visible or occluded. Formally, given a video sequence $\{\mathcal{I}_t\}_{t=0}^{T-1}$, where $\mathcal{I}_t \in \mathbb{R}^{H \times W \times 3}$ represents the t -th frame of height H and width W , and a query point $q = (x_q, y_q, t_q) \in \mathbb{R}^3$ at frame t_q , the goal is to produce a trajectory $\mathcal{T} = \{\hat{p}_t\}_{t=0}^{T-1}$, where $\hat{p}_t \in \mathbb{R}^2$ denotes the estimated position of the point at time t , and an associated visibility probability $\mathcal{V} = \{v_t\}_{t=0}^{T-1}$, where $v_t \in [0, 1]$ indicates the likelihood of the point being visible at frame t .

Motivation and overview. Accurate point tracking in videos requires robust point matching across frames, a challenging task due to complex motions, occlusions, deformations, and scale variations [11, 25, 47]. Because videos consist of sequences over time, considering both spatial and temporal aspects is essential to understand point dynamics. More specifically, a good feature for point tracking must satisfy two key criteria. Spatially, it should effectively model complex real-world data with powerful feature representations vital for robust matching, the cornerstone of point

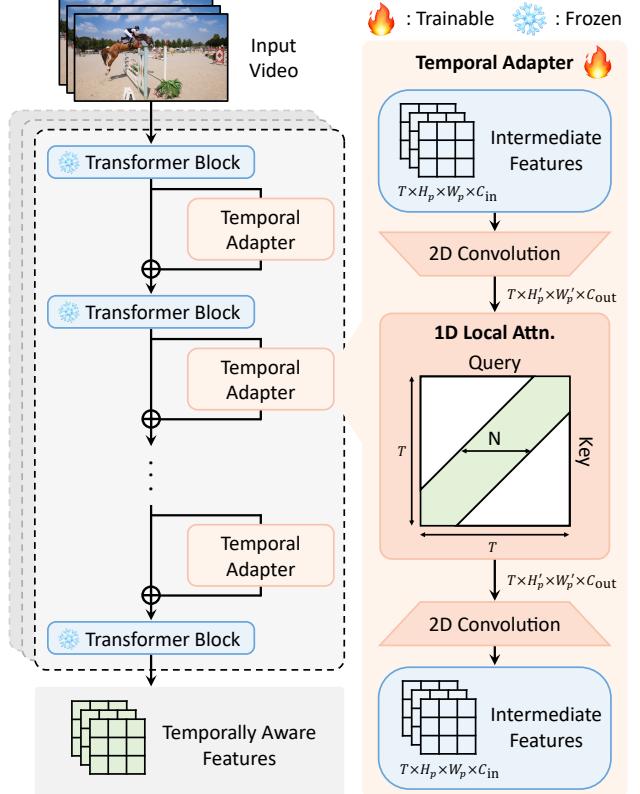


Figure 2. **Overall architecture of Chrono.** Temporal adapters between transformer blocks use 2D convolution and 1D local attention along the temporal axis to output temporally-aware features.

tracking [11]. Temporally, it must capture complex motions and understand the dynamics of points over time to handle their movement across frames.

To fulfill the spatial criterion, we utilize DINOv2 [36], a self-supervised model trained on large-scale real-world data, renowned for its robust feature representations across various tasks [5, 29, 36]. However, DINOv2 lacks inherent temporal awareness, which is essential for point tracking in videos. To address this limitation, we introduce a temporal adapter that supplements DINOv2’s features with temporal information. This adapter integrates temporal understanding directly into the feature extraction process, creating a feature backbone tailored for point tracking that combines strong spatial representations with temporal dynamics.

3.2. Temporal Adapter

Temporal awareness is essential for accurately tracking points across frames, as it enables the model to capture motion patterns and temporal dependencies. To embed this capability within DINOv2, we design a temporal adapter that allows the backbone to incorporate information from adjacent frames, enhancing its temporal sensitivity. Figure 2 illustrates the overall architecture of Chrono.

Design of the temporal adapter. Our temporal adapter is placed between each transformer block of DINOv2 [36] to enhance temporal modeling across multiple feature levels. By computing adjacent features from other frames at once, the adapter connects features from different time steps, enabling it to recognize motion across frames.

Each adapter follows a bottleneck structure inspired by the ResNet [20] architecture. It begins with a 2D convolutional layer $\text{Conv2D}^{\text{down}}$ with stride s that reduces the spatial dimensions from $H_p \times W_p \times C_{\text{in}}$ to a compact $H'_p \times W'_p \times C_{\text{out}}$, where $H'_p = H_p/s$ and $W'_p = W_p/s$. This spatial downsampling step optimizes the representation for efficient processing while expanding the spatial receptive field for subsequent operations. This is formally defined as:

$$\mathbf{f}_t^{\text{down}} = \text{Conv2D}^{\text{down}}(\mathbf{f}_t^{\text{in}}), \quad (1)$$

where $\mathbf{f}_t^{\text{in}} \in \mathbb{R}^{H_p \times W_p \times C_{\text{in}}}$ and $\mathbf{f}_t^{\text{out}} \in \mathbb{R}^{H'_p \times W'_p \times C_{\text{out}}}$ are the input and output feature maps of $\text{Conv2D}^{\text{down}}$ at time t , respectively.

After downsampling, we apply a temporal attention layer that captures dependencies within a local temporal window of size N , focusing on temporal correlations across time [3, 41]. The local window attention is carefully calibrated to balance computational efficiency with effective motion capture, ensuring a broad enough temporal window without excessive resource demands. This layer operates over the neighboring frames within the window $[t-k, t+k]$, where $k = \frac{N-1}{2}$. For each spatial location (x, y) , we compute attention weights based on the similarity between the query vector at time t and key vectors from neighboring times. The aggregated feature at time t and location (x, y) , referred to as $\mathbf{f}_t^{\text{attn}}(x, y)$, is computed as a weighted sum of the value vectors from these neighboring frames:

$$\mathbf{f}_t^{\text{attn}}(x, y) = \sum_{n=-k}^k \alpha^{(t,n)}(x, y) \cdot \mathbf{V}_{t+n}(x, y), \quad (2)$$

where $\alpha^{(t,n)}(x, y)$ represents the attention weights for time offset n , computed as:

$$\alpha^{(t,n)}(x, y) = \frac{\exp(\mathbf{Q}_t(x, y) \cdot \mathbf{K}_{t+n}(x, y))}{\sum_{n'=-k}^k \exp(\mathbf{Q}_t(x, y) \cdot \mathbf{K}_{t+n'}(x, y))}, \quad (3)$$

where \mathbf{Q} , \mathbf{K} , and \mathbf{V} are projections of $\mathbf{f}_t^{\text{down}}$, obtained through the linear projection layers \mathbf{W}_Q , \mathbf{W}_K , and \mathbf{W}_V .

After temporal attention, another 2D convolutional layer restores the spatial dimensions back to $H_p \times W_p \times C_{\text{in}}$:

$$\mathbf{f}_t^{\text{up}} = \text{Conv2D}^{\text{up}}(\mathbf{f}_t^{\text{attn}}), \quad (4)$$

where $\mathbf{f}_t^{\text{up}} \in \mathbb{R}^{H_p \times W_p \times C_{\text{in}}}$. Finally, a residual connection adds the original input feature map to the output of the adapter to preserve the feature representation of DINOv2:

$$\mathbf{f}_t^{\text{out}} = \mathbf{f}_t^{\text{up}} + \mathbf{f}_t^{\text{in}}. \quad (5)$$

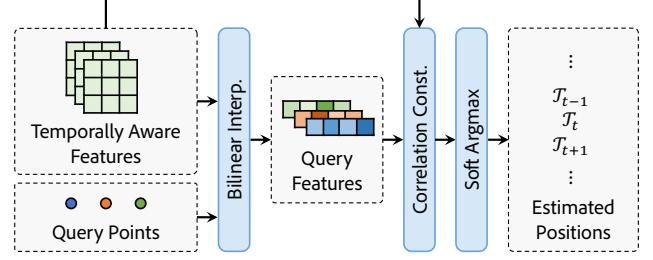


Figure 3. **Point track prediction.** To predict point positions, we simply match the query points with features from other frames, without using any learnable layers.

3.3. Point Prediction with the Feature Backbone

As shown in Figure 3, we track a query point by simply performing feature matching using features from Chrono. This approach avoids the need for learnable modules such as iterative refinement, as used in [10, 11, 13, 25].

Correlation construction. Given the query point $q = (x_q, y_q, t_q)$, we extract the query feature \mathbf{f}_q from the feature map of frame t_q at position (x_q, y_q) using bilinear interpolation. For each frame t , we compute the correlation map C_t by calculating the cosine similarity between \mathbf{f}_q and the feature map \mathbf{f}_t at every spatial location:

$$C_t(x, y) = \frac{\mathbf{f}_q^\top \mathbf{f}_t(x, y)}{\|\mathbf{f}_q\| \|\mathbf{f}_t(x, y)\|}, \quad (6)$$

where $\mathbf{f}_t(x, y)$ is the feature vector at position (x, y) in frame t , $\mathbf{f}_q^\top \mathbf{f}_t(x, y)$ denotes the dot product between \mathbf{f}_q and $\mathbf{f}_t(x, y)$, and $\|\mathbf{f}_q\|$ and $\|\mathbf{f}_t(x, y)\|$ indicate the Euclidean norms of \mathbf{f}_q and $\mathbf{f}_t(x, y)$, respectively. This correlation map C_t quantifies the similarity between the query feature and features across all spatial locations in each frame t .

Point prediction. To estimate the position of points in frame t , we apply the soft-argmax operation to the correlation map C_t . The soft-argmax computes a weighted average of all spatial positions, with weights given by the softmax of the correlation values:

$$\hat{\mathbf{p}}_t = \sum_{(x,y)} \sigma(C_t(x, y)) \cdot (x, y), \quad (7)$$

where $\sigma(C_t(x, y))$ is the softmax over all spatial positions within frame t :

$$\sigma(C_t(x, y)) = \frac{\exp(C_t(x, y) \cdot \tau)}{\sum_{(x',y')} \exp(C_t(x', y') \cdot \tau)}, \quad (8)$$

where τ is a softmax temperature. The soft argmax is non-learnable and provides a differentiable method to estimate the point’s position based on the correlation map. To enhance precision, we mask out positions more than M pixels away from the maximum correlation value, focusing the soft argmax computation on a local neighborhood and reducing the influence of irrelevant regions [28].

Training. To train Chrono, we use the Huber loss [24] to supervise the estimated positions. The Huber loss is chosen for its robustness to outliers compared to the squared error loss. At each time step t , the loss is defined as:

$$\mathcal{L}_{\text{Huber}}(\hat{\mathbf{p}}_t, \mathbf{p}_t) = \begin{cases} \frac{1}{2} \|\hat{\mathbf{p}}_t - \mathbf{p}_t\|^2, & \text{if } \|\hat{\mathbf{p}}_t - \mathbf{p}_t\| \leq \delta, \\ \delta \cdot (\|\hat{\mathbf{p}}_t - \mathbf{p}_t\| - \frac{1}{2}\delta), & \text{otherwise,} \end{cases} \quad (9)$$

where \mathbf{p}_t is the ground truth position, and δ is a threshold parameter. For occluded points indicated by the ground truth visibility status v_t , we exclude the loss computation, effectively masking out those time steps:

$$\mathcal{L}_t = (1 - v_t) \cdot \mathcal{L}_{\text{Huber}}(\hat{\mathbf{p}}_t, \mathbf{p}_t). \quad (10)$$

By minimizing this loss over all time steps and query points, we train the backbone to produce features for predicting accurate tracked positions.

4. Experiments

4.1. Implementation Details

We implement our method using PyTorch [37]. During training, we use AdamW optimizer [33] with a learning rate of 10^{-4} , a weight decay of 10^{-4} , and a batch size of 1 per GPU. All models are trained for 100,000 iterations on 4 A100 GPUs, employing a cosine learning rate scheduler with warmup. For training, we use the Kubric Panning-MOVIE dataset [11, 13] and sample 256 query points per batch. The hyperparameters are set as follows: the softmax temperature is $\tau = 20.0$, the soft argmax pixel threshold is $M = 5$ and the local temporal window size is $N = 13$, and the 2D convolution stride in the temporal adapter is $s = 4$.

4.2. Evaluation Protocol

Evaluation datasets. To assess our approach, we employ the TAP-Vid benchmark [12], which comprises both real and synthetic video sets. The real videos come with accurate annotation tracks, while synthetic videos are paired with perfect ground-truth trajectories. For our analysis, we focus on the real video datasets, specifically TAP-Vid-DAVIS and TAP-Vid-Kinetics. The DAVIS dataset [38] provides 30 videos that include challenges such as substantial scale shifts of objects, while Kinetics [27] consists of 1,189 YouTube videos featuring diverse difficulties, including intense motion blur and abrupt scene transitions.

Evaluation metrics. We assess the accuracy of the predicted tracks using two metrics: position accuracy at various thresholds and average position accuracy ($< \delta_{avg}^x$). Position accuracy is evaluated at five threshold levels: $< \delta^0$, $< \delta^1$, $< \delta^2$, $< \delta^3$, and $< \delta^4$, representing accuracies within pixel distances of 1, 2, 4, 8, and 16, respectively. Each $< \delta^x$ score indicates the percentage of visible ground-truth

points with predictions falling within the specified threshold. $< \delta_{avg}^x$ averages these scores to provide an overall measure of position accuracy across all thresholds.

Following [12], we evaluate the datasets using two modes: strided query mode and first query mode. In strided query mode, query points are sampled along the ground-truth trajectory at intervals of 5 frames. In first query mode, query points are sampled from the first visible frame.

4.3. Main Results

Quantitative comparison. We evaluate our method against the backbones [20, 31] commonly used in point tracking [11–13, 25, 26] and DINOv2 [36] in both the strided query (Table 1) and the first query modes (Table 2). Specifically, we compare with ResNet-18 [20] from TAPIR [13] and TSM-ResNet-18 [31] from TAP-Net [12], both trained for point tracking. We report position accuracy to demonstrate the effectiveness of temporal information in our model.

Our small model, Chrono (ViT-S/14) with DINOv2 (ViT-S/14), achieves state-of-the-art performance in position accuracy across all thresholds, surpassing other backbones. Chrono (ViT-B/14) with DINOv2 (ViT-B/14) delivers even better results. Across both Kinetics and DAVIS datasets, in the strided and first query modes, Chrono (ViT-S/14) and Chrono (ViT-B/14) consistently outperform ResNet-18, TSM-ResNet-18, and other DINOv2 variants in PCK accuracy at all thresholds. The high accuracy at $< \delta^0$ underscores the precision of Chrono features in point tracking, with further gains at higher thresholds. At $< \delta^4$, only Chrono backbones achieve over 90% accuracy on the DAVIS dataset in strided mode. Both Chrono (ViT-S/14) and Chrono (ViT-B/14) reach the highest position accuracy among backbones, demonstrating Chrono’s ability to effectively model spatial and temporal information in videos.

Qualitative comparison. We visualize the estimated tracks from the DAVIS dataset, with a qualitative comparison shown in Figure 4. Unlike other methods, which produce highly jittery and inconsistent tracks over time, Chrono generates temporally smooth and accurate tracks. This jitter in other methods is expected as they lack awareness of neighboring frames. In contrast, Chrono utilizes temporal adapters to maintain frame-to-frame consistency, allowing it to produce smooth tracks even when individual frame estimates are less stable.

4.4. Analysis and Ablation Study

Comparison to point tracking with iterative refiner. As shown in Table 3, Chrono, although solely a feature backbone without a refiner, achieves point estimation precision comparable to full pipelines with refiners, while demonstrating significantly higher efficiency.



Figure 4. **Qualitative comparison of complex real-world video tracking.** We qualitatively compare the results generated by Chrono with those from other commonly used backbones in point tracking and as well as DINOv2. Our model demonstrates better smoothness and precision than other competitors.

Backbone	Kinetics-Strided						DAVIS-Strided					
	$< \delta^0$	$< \delta^1$	$< \delta^2$	$< \delta^3$	$< \delta^4$	$< \delta_{avg}^x$	$< \delta^0$	$< \delta^1$	$< \delta^2$	$< \delta^3$	$< \delta^4$	$< \delta_{avg}^x$
ResNet-18 [13, 20]	10.5	35.3	65.7	81.3	88.6	56.3	9.7	31.1	60.6	78.3	87.0	53.3
TSM-ResNet-18 [12, 31]	9.15	33.2	64.6	79.2	86.5	54.5	8.2	26.8	53.6	73.5	83.9	49.2
DINOv2 (ViT-S/14) [36]	4.9	15.8	41.9	73.8	86.0	44.5	7.3	22.7	52.9	80.0	89.5	50.4
DINOv2 (ViT-B/14) [36]	5.9	18.6	45.9	75.7	86.9	46.6	10.0	28.6	59.5	82.9	87.4	54.4
Chrono (ViT-S/14)	<u>32.2</u>	<u>55.2</u>	<u>73.9</u>	<u>84.0</u>	<u>88.7</u>	<u>66.8</u>	<u>29.7</u>	<u>56.4</u>	<u>76.7</u>	<u>86.8</u>	<u>90.9</u>	<u>68.0</u>
Chrono (ViT-B/14)	<u>33.5</u>	<u>57.2</u>	<u>75.8</u>	<u>85.8</u>	<u>90.2</u>	<u>68.5</u>	<u>31.8</u>	<u>59.2</u>	<u>78.8</u>	<u>88.4</u>	<u>92.2</u>	<u>70.1</u>

Table 1. **Quantitative comparison on the TAP-Vid datasets [12] with the strided query mode.** Compared to DINOv2 and previous feature backbones used in point tracking, our model shows a significant performance boost.

Unlike other models, which require refiners to inject temporal information for each query point, Chrono embeds temporal awareness directly in the feature backbone. This approach significantly increases throughput, with Chrono (ViT-B/14) achieving 12.5 times the speed of TAPIR while maintaining only a 3.5%p difference in precision, and Chrono (ViT-S/14) delivering 16.4 times the throughput. These results demonstrate that Chrono provides a simple yet highly effective solution, offering performance comparable to refiner-based models without requiring iterative processing. We define the number of learnable parameters after feature extraction as the number of refiner parameters.

Analysis on Chrono as a backbone in existing point tracking pipeline. In Table 4, we present an analysis of

the impact of adding a refiner after our model to investigate its ability as a feature backbone within a point tracking pipeline that uses iterative refinement [11, 13, 19, 25]. To assess the ability to predict precise occlusion status, we use occlusion accuracy (OA) and the Average Jaccard (AJ) metric [12], which measures both position and occlusion.

For our experiment, we incorporate the iterative refiner from LocoTrack [11]. Since LocoTrack utilizes three levels of hierarchical features from ResNet, we apply a simple convolutional upsampler to our single-resolution feature map. Specifically, using transposed convolution [50], we generate a $2 \times$ feature map with a $4 \times$ channel reduction and then a $4 \times$ feature map with a further $2 \times$ channel reduction. We freeze our model and train the convolutional upsampler and the LocoTrack refiner on the Kubric [16] panning

Backbone	Kinetics-First						DAVIS-First					
	$< \delta^0$	$< \delta^1$	$< \delta^2$	$< \delta^3$	$< \delta^4$	$< \delta_{avg}^x$	$< \delta^0$	$< \delta^1$	$< \delta^2$	$< \delta^3$	$< \delta^4$	$< \delta_{avg}^x$
ResNet-18 [13, 20]	8.6	28.8	56.5	74.2	83.3	50.3	9.0	27.3	54.9	73.7	84.1	49.8
TSM-ResNet-18 [12, 31]	7.8	28.1	55.2	71.4	80.2	48.6	7.3	23.1	46.7	66.6	79.2	44.6
DINOv2 (ViT-S/14) [36]	4.2	13.6	36.3	65.9	79.2	39.8	6.0	19.9	47.5	73.7	84.5	46.3
DINOv2 (ViT-B/14) [36]	5.1	16.0	40.0	67.9	80.4	41.9	8.9	24.7	53.8	77.0	87.1	50.3
Chrono (ViT-S/14)	24.8	46.2	65.8	77.5	83.1	59.5	24.0	49.2	71.2	82.8	87.9	63.0
Chrono (ViT-B/14)	26.0	48.4	68.2	79.8	85.3	61.6	26.1	52.6	74.5	84.9	90.0	65.6

Table 2. Quantitative comparison on the TAP-Vid datasets [12] with the query first mode. Compared to other feature backbones widely used in point tracking, our model shows exceptional performance in position accuracy.

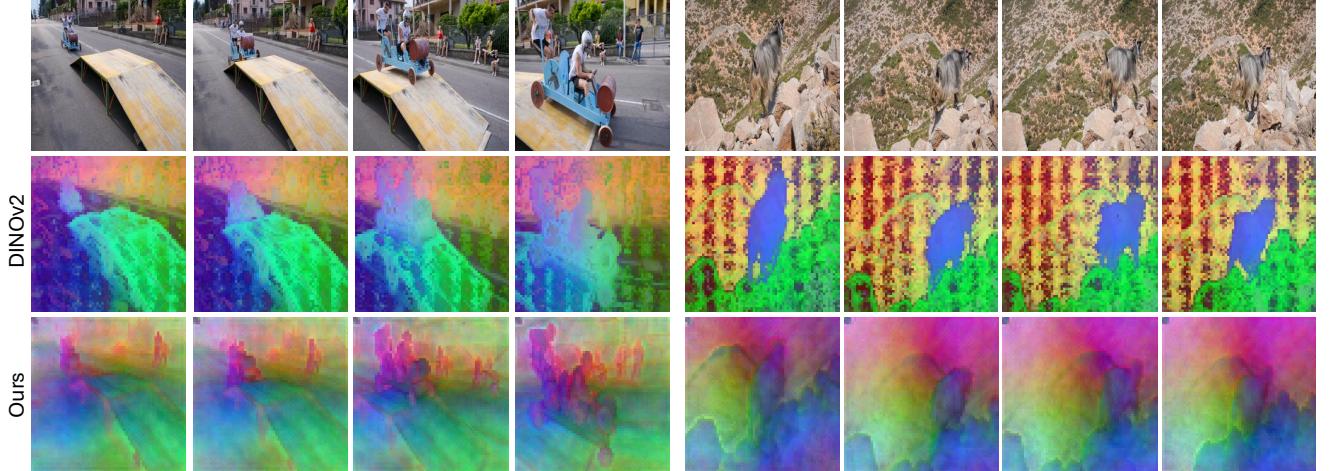


Figure 5. **Visualization of features.** We visualize the features from Chrono and those from DINOv2 with PCA. Our model demonstrates better temporal smoothness than DINOv2.

Method	$< \delta_{avg}^x$	Throughput (points/sec)	# of Refiner Params
RAFT [44]	46.3	23,405.71	4.2M
TAP-Net [12]	53.1	29,535.98	5.5K
TAIR [13]	73.6	2,097.32	25.9M
PIPs [19]	59.4	46.43	26.0M
Chrono (ViT-S/14)	68.0	34,396.30	None
Chrono (ViT-B/14)	70.1	26,139.86	None

Table 3. Comparison on point tracking pipelines using refiners. Although Chrono does not use a heavy iterative refiner, it achieves comparable performance with significantly higher throughput. All metrics are measured on a 24-frame video using a single NVIDIA RTX 3090 GPU, and the TAP-Vid-DAVIS dataset [12].

MOV-E dataset [13].

Our model combined with LocoTrack surpasses the original LocoTrack and also surpasses the performance of recent state-of-the-art point trackers [10, 25]. Our model shows a boost on all datasets, and notably, it shows a huge boost in the RGB-Stacking dataset, achieving a +6.1 improvement in the AJ score. This result demonstrates the effectiveness of our approach when adopted into an existing point tracking pipeline.

Ablation on temporal aggregation design choice. To model temporal information in our temporal adapter, we tested two approaches for temporal aggregation: convolution and attention. In Table 5, we present an ablation study where only the temporal aggregation layer within Chrono is modified, comparing 1D convolution, 3D convolution, and 1D attention, with all other components kept constant. While 1D convolution and 1D attention aggregate along the temporal axis only, 3D convolution considers both spatial and temporal dimensions. Due to computational constraints, this study was conducted on Chrono (ViT-S/14).

The results suggest that 1D attention is effective for temporal aggregation, as it allows the model to capture motion dynamics by adapting to correlations between frames. This adaptability may contribute to improved tracking accuracy by enabling the model to weigh information from different frames and handle varied motion patterns and occlusions. In contrast, temporal convolutions, with their fixed weights, may be less effective in capturing these complex temporal relationships.

Ablation on the number and position of temporal adapters. In Table 6, we conduct an ablation study on the placement of temporal adapters in Chrono, which consists of 12 transformer blocks. We evaluate three configurations:

Method	RGB-Stacking			Kinetics			DAVIS		
	AJ	$< \delta_{avg}^x$	OA	AJ	$< \delta_{avg}^x$	OA	AJ	$< \delta_{avg}^x$	OA
Kubric-VFS-Like [16]	57.9	72.6	91.9	40.5	59.0	80.0	33.1	48.5	79.4
TAP-Net [12]	59.9	72.8	90.4	46.6	60.9	85.0	38.4	53.1	82.3
PIPs [19]	37.3	51.0	91.6	35.3	54.8	77.4	42.0	59.4	82.1
RAFT [44]	44.0	58.6	90.4	34.5	52.5	79.7	30.0	46.3	79.6
TAPIR [13]	62.7	74.6	91.6	57.2	70.1	87.8	61.3	73.6	88.8
FlowTrack [10]	-	-	-	-	-	-	66.0	79.8	87.2
CoTracker [25]	-	-	-	-	-	-	65.9	79.4	89.9
LocoTrack [11]	77.1	86.9	93.2	59.5	73.0	88.5	67.8	79.6	89.9
Chrono + LocoTrack	83.2	91.0	94.2	60.7	73.7	89.5	68.2	80.2	91.2

Table 4. Quantitative comparison of Chrono, adapted to LocoTrack [11], on the TAP-Vid dataset with the strided query mode. Our model shows a performance boost over LocoTrack on all datasets, with a particularly significant improvement on RGB-Stacking.

Method	DAVIS					
	$< \delta^0$	$< \delta^1$	$< \delta^2$	$< \delta^3$	$< \delta^4$	$< \delta_{avg}^x$
1D Conv.	26.5	53.1	74.5	85.4	90.1	65.9
3D Conv.	27.5	54.3	74.9	85.5	90.0	66.4
1D Attn. (Ours)	29.7	56.4	76.7	86.8	90.9	68.0

Table 5. Ablation on temporal aggregation design choice. 1D attention demonstrates the best performance compared to both 1D and 3D convolution layers.

Configuration	# of Adapters	DAVIS					
		$< \delta^0$	$< \delta^1$	$< \delta^2$	$< \delta^3$	$< \delta^4$	$< \delta_{avg}^x$
Early Blocks	6	23.1	47.8	69.8	81.4	86.6	61.7
Later Blocks	6	26.6	52.9	74.3	85.3	90.1	65.8
Alternating Blocks	6	26.9	53.4	74.3	85.3	90.0	65.9
All Blocks	11	29.7	56.4	76.7	86.8	90.9	68.0

Table 6. Ablation on the number and position of temporal adapter. Rather than placing the temporal adapter into early or later blocks, or alternating blocks in DINOv2, applying it to all layers significantly boosts performance.

placing six temporal adapters within the early six blocks, within the later six blocks, and in alternating blocks.

Early transformer blocks often capture local details, while later blocks tend to focus on broader, global patterns and complex relationships [15, 39]. Therefore, it’s plausible that adapters in the initial blocks may learn local motion cues, while those in later blocks might capture overall motion patterns. Placing adapters only in the early or later blocks might emphasize either local or global motion, respectively. While the alternating configuration shows some improvement, it may not be optimal. Integrating temporal adapters between each block could potentially allow for a more balanced capture of both local and global motion patterns, making features sensitive to multilevel motions and possibly leading to better performance.

Inference time comparison. Table 7 compares our inference time with that of DINOv2 [36], measuring the time taken to extract features for 24 frames. While our method

Method	$< \delta_{avg}^x$	Inference Time (sec)	
		DINOv2 (ViT-S/14) [36]	DINOv2 (ViT-B/14) [36]
Chrono (ViT-S/14)	39.8	0.194	
Chrono (ViT-B/14)	41.9		0.443

Table 7. Inference Time Comparison with DINOv2. We measure inference time for processing 24 frames without point prediction.

exhibits slower inference due to temporal injection into the feature, Chrono achieves a significantly higher score than vanilla DINOv2. This higher accuracy compensates for the increased time required for feature extraction.

Visualization of feature with PCA. We visually compare the feature of Chrono and DINOv2 [36] with PCA (Principal Component Analysis) in Figure 5. We reduce the feature dimensions to three, represented as RGB in the figure.

While DINOv2’s visualizations demonstrate a noisy background and tend to show a uniform representation within a single object, our model produces consistent and smooth representations over time and also shows fine granularity within the same semantic object. We believe the temporally smooth representation and fine-grained feature detail lead to better point tracking.

5. Conclusion

In this paper, we presented **Chrono**, a temporally-aware feature backbone for point tracking that integrates pre-trained DINOv2 representations with a temporal adapter, enabling long-term temporal context capture in feature space. Through extensive experiments, we demonstrate that embedding temporal information directly into the feature backbone, Chrono reduces reliance on costly refiners while achieving both accuracy and efficiency. We anticipate Chrono will inspire advancements in efficient, temporally-aware backbones for point tracking.

References

- [1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 3
- [2] Görkay Aydemir, Weidi Xie, and Fatma Güney. Can visual foundation models achieve long-term point tracking? *arXiv preprint arXiv:2408.13575*, 2024. 3
- [3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 4
- [4] Shariq Farooq Bhat, Reiner Birk, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 1
- [5] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 3
- [6] 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. 1, 2
- [7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.
- [8] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1299, 2022. 1
- [9] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. *Advances in Neural Information Processing Systems*, 34:9011–9023, 2021. 2
- [10] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. Flowtrack: Revisiting optical flow for long-range dense tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19268–19277, 2024. 2, 3, 4, 7, 8
- [11] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. *arXiv preprint arXiv:2407.15420*, 2024. 1, 2, 3, 4, 5, 6, 8
- [12] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao 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. 2, 5, 6, 7, 8
- [13] 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, 5, 6, 7, 8
- [14] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, et al. Bootstrap: Bootstrapped training for tracking-any-point. *arXiv preprint arXiv:2402.00847*, 2024. 1, 2, 3
- [15] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 8
- [16] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022. 1, 2, 6, 8
- [17] Kamal Gupta, Varun Jampani, Carlos Esteves, Abhinav Shrivastava, Ameesh Makadia, Noah Snavely, and Abhishek Kar. Asic: Aligning sparse in-the-wild image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4134–4145, 2023. 3
- [18] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022. 3
- [19] Adam W. Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *ECCV*, 2022. 1, 2, 6, 7, 8
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 4, 5, 6, 7
- [21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1
- [22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 3
- [23] Jiahui Huang, Leonid Sigal, Kwang Moo Yi, Oliver Wang, and Joon-Young Lee. Inve: Interactive neural video editing. *arXiv preprint arXiv:2307.07663*, 2023. 1
- [24] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992. 5
- [25] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Co-tracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 1, 2, 3, 4, 5, 6, 7, 8
- [26] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Co-tracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 2, 5

- [27] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5
- [28] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2278–2287, 2019. 4
- [29] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 2, 3
- [30] 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. 2, 3
- [31] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019. 1, 2, 5, 6, 7
- [32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1
- [33] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5
- [34] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. Improving semantic correspondence with viewpoint-guided spherical maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19521–19530, 2024. 3
- [35] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022. 3
- [36] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3, 4, 5, 6, 7, 8
- [37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5
- [38] 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. 5
- [39] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021. 8
- [40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3
- [41] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. 4
- [42] Mohammadreza Salehi, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. Time does tell: Self-supervised time-tuning of dense image representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16536–16547, 2023. 3
- [43] Aleksandar Shtedritski, Andrea Vedaldi, and Christian Rupprecht. Learning universal semantic correspondences with no supervision and automatic data curation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 933–943, 2023. 3
- [44] 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. 7, 8
- [45] Narek Tumanyan, Assaf Singer, Shai Bagon, and Tali Dekel. Dino-tracker: Taming dino for self-supervised point tracking in a single video. In *European Conference on Computer Vision*, pages 367–385. Springer, 2025. 1, 2, 3
- [46] 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, 2
- [47] 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, 2, 3
- [48] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 1
- [49] Jiarui Xu and Xiaolong Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10075–10085, 2021. 2
- [50] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, pages 2528–2535. IEEE, 2010. 6
- [51] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022. 1

- [52] 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