Video Diffusion Models Excel at Tracking Similar-Looking Objects Without Supervision

Chenshuang Zhang[1] **Kang Zhang**[1] **Joon Son Chung**[1]
In So Kweon[1] **Junmo Kim**[1]* **Chengzhi Mao**[2]*
KAIST[1], Rutgers University[2]

Abstract

Distinguishing visually similar objects by their motion remains a critical challenge in computer vision. Although supervised trackers show promise, contemporary self-supervised trackers struggle when visual cues become ambiguous, limiting their scalability and generalization without extensive labeled data. We find that pre-trained video diffusion models inherently learn motion representations suitable for tracking without task-specific training. This ability arises because their denoising process isolates motion in early, high-noise stages, distinct from later appearance refinement. Capitalizing on this discovery, our self-supervised tracker significantly improves performance in distinguishing visually similar objects, an underexplored failure point for existing methods. Our method achieves up to a 6-point improvement over recent self-supervised approaches on established benchmarks and our newly introduced tests focused on tracking visually similar items. Visualizations confirm that these diffusion-derived motion representations enable robust tracking of even identical objects across challenging viewpoint changes and deformations. Project page: `https://chenshuang-zhang.github.io/projects/ted`.

1 Introduction

Imagine tracking one of two similar-looking deer walking in the forest (Figure 1(a)). Humans effortlessly resolve such visual ambiguities by relying on the distinct motion signatures of objects. This ability to perceive coherent objects through their unique temporal dynamics, even when static appearances are confounding, is fundamental. However, imbuing visual representations with this innate understanding of temporal dynamics, especially for tracking similar-looking objects, remains a significant challenge in computer vision [31, 23, 10, 51].

Many self-supervised methods [7, 23, 45], while good at learning intra-frame appearance features, fail when confronted with visually similar targets (see DIFT [45] in Figure 1(c)). Their Achilles' heel is the neglect of inter-frame temporal relationships. Even approaches that incorporate temporal signals through training objectives like cycle-consistency [32, 52, 24] often process frames independently at inference using 2D image encoders. This inherently limits their ability to model the continuous motion crucial for disambiguating similar objects in dynamic scenes (see CRW [24] and Spa-then-Temp [31] in Figure 1(c)).

In this paper, we show that representations for similar-looking object tracking do not need to be learned from scratch with intricate tracking-specific objectives. Instead, they can be repurposed within the internal workings of pre-trained video diffusion models [58, 2]. Unlike methods that view video as a sequence of isolated images [20, 23, 36], video diffusion models, by their very nature of generating coherent and realistic video, must implicitly capture the complex interplay of inter-frame dynamics. We find that the denoising process, particularly as it reconstructs motion from highly noisy states, already encodes a rich, motion-aware representation through its feature activations—a representation ripe for tracking without any explicit tracking supervision.

*Corresponding author. Junmo Kim <junmo.kim@kaist.ac.kr>, Chengzhi Mao <cm1838@cs.rutgers.edu>.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

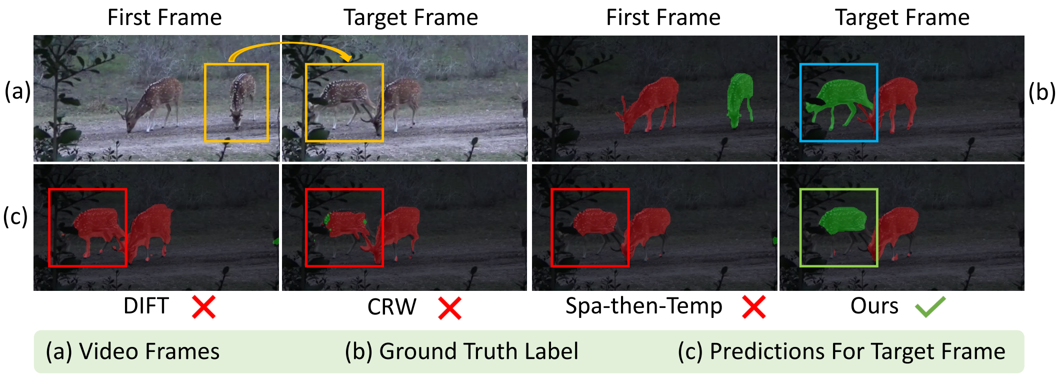


Figure 1: **Video label propagation on similar-looking objects.** State-of-the-art self-supervised trackers, such as DIFT [45], CRW [24] and Spa-then-Temp [31], often struggle when multiple objects look similar in a video. This failure is due to their exclusive reliance on appearance features. By dissecting and repurposing pretrained video diffusion models, we construct a feature that captures intra-frame motions in videos, allowing us to correctly track similar-looking objects, such as the deer highlighted by the green box in (c). In this figure, the green and red masks represent segmentation maps of different objects, while the blue, green, and red boxes highlight the ground truth regions, correctly predicted regions, and incorrectly predicted regions, respectively.

We introduce the **T**emporal **E**nhanced **D**iffusion tracking framework (TED), a simple yet remarkably effective approach that harnesses these latent diffusion features. TED synergizes the motion intelligence distilled by video diffusion models with conventional appearance features, enabling it to conquer the limitations of prior art [6, 23, 45] and robustly track visually indistinct objects (Figure 1(c)).

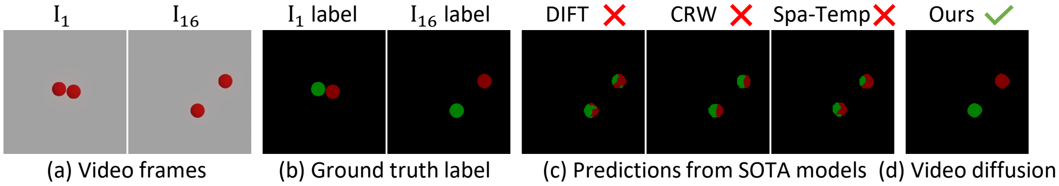
Experimental results show that our TED method outperforms 17 popular self-supervised models, achieving state-of-the-art performance in pixel-level object tracking. On the widely-used DAVIS-2017 benchmark [35], our TED significantly outperforms recent self-supervised methods [23, 45, 36, 31] by up to 6%. When evaluated on videos that include multiple similar-looking objects, our TED method achieves even larger improvement by up to 10%. Visualizations confirm that our representations encode differently for similar looking objects with different motion. Our approach also achieves significant improvement in other challenging scenarios, such as appearance-identical objects, real-world viewpoint changes, and object deformations.

2 Related Work

Learning video representations for temporal correspondence is crucial for visual tracking [46, 55, 31]. Due to limited annotations, recent studies have proposed various pretext tasks to learn representations in a self-supervised manner. We discuss related work below.

Self-supervised representation learning from images. Prior studies learn appearance features in video representations by training models on independent images [20, 6, 23, 45]. Some methods adopt instance discrimination as a pretext task, such as MoCo [20] and SimCLR [6]. SFC [23] improves further by integrating image-level and pixel-level cues for representation learning. DIFT [45] leverages knowledge from image diffusion models [40]. However, these methods only learn intra-frame appearance features, which fail in tracking visually similar objects (Figure 1(c)).

Self-supervised representation learning from videos. Some methods introduce temporal signals to model training, using two pretext tasks: cycle-consistency over time and frame reconstruction. Cycle-consistency task tracks a patch backward and forward in time to align its start and end points [32, 52, 24], while frame reconstruction aims to reconstruct pixels from adjacent frames [47, 28, 27]. Recent studies integrate temporal and spatial cues for training, such as Spa-then-Temp [31] and SMTC [36]. However, during inference, these models process frames independently using 2D image encoders, neglecting temporal context. Therefore, they fail to track similar-looking objects as in Figure 1(c).



(a) Video frames (b) Ground truth label (c) Predictions from SOTA models (d) Video diffusion

Figure 2: **Our approach successfully tracks objects with identical appearances.** We conduct a controlled study, that we perform object label propagation on videos featuring two identical-looking and independently moving balls, with frames and their ground truth labels shown in (a) and (b). State-of-the-art methods [24, 31, 45] fail to distinguish these two balls, leading to incorrect predictions (c). In contrast, our approach accurately track both balls despite their identical appearance (d).

Video object segmentation. Supervised methods for video object segmentation [9, 57, 8] achieve impressive results but rely on large-scale annotated datasets for model training. For example, SAM2 [38] is trained on 50.9K videos with 35.5M masks. In contrast, our work addresses *self-supervised* tracking: no segmentation labels are used. Furthermore, models like SAM2 [38] use discriminative training objectives, whereas our work explores the inherent tracking capability of generative models. We find that video diffusion models can effectively track visually similar objects without any tracking-specific supervision, pointing to a promising direction for future trackers.

Video diffusion models. Diffusion models [21] have achieved great success in image generation [37, 43, 34, 42], such as Stable Diffusion [40] and ADM [14]. Video diffusion models further include temporal blocks for frame consistency [3, 48], with pioneering work Sora [4], I2VGen-XL [58], and Stable Video Diffusion [2]. Diffusion models have also been used in tasks like image classification [29, 12], semantic segmentation [1, 60, 53] and pose estimation [22, 16]. In contrast to these studies, we are the first to show that video diffusion models excel at tracking similar-looking objects without any tracking-specific training. Our finding that video diffusion models learn motions at high-noise stages also advances the understanding of video diffusion models.

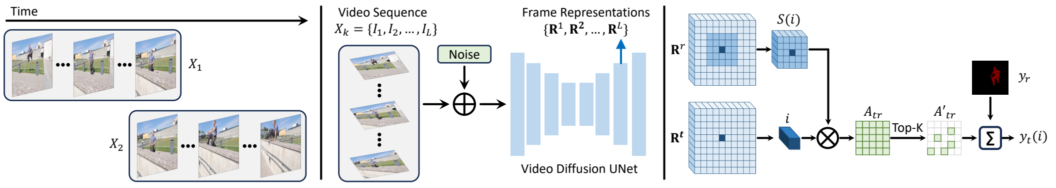
Track by diffusion models. There has been recent interest in applying diffusion models to tracking [25, 59, 50]. Track4Gen [25] tackles point tracking by training video diffusion models on labeled point trajectories, whereas our work explores pretrained video diffusion models for object segmentation without any tracking-specific training. Diff-Tracker [59] is built on image diffusion models with additional motion encoders to learn temporal cues. By contrast, our work directly explores the built-in motion of pretrained video diffusion models without extra modules. VidSeg [50] performs instance-agnostic video semantic segmentation that cannot distinguish different objects in the same category. It also requires maintaining and updating an additional KNN classifier to learn temporal changes during tracking. By contrast, our approach can distinguish even similar-looking objects without extra components.

3 Challenges for Tracking Visually Similar Objects

Task definition. We focus on video label propagation task, which aims to transfer ground truth labels of the first frame (e.g., segmentation map) to subsequent frames [47]. The key is training models to obtain frame representation $\mathbf{R}$, which learns pixel-level correspondence among frames [23, 36, 31]. Due to limited annotations, prior studies train models in a self-supervised manner [24, 31], with pretext tasks like instance discrimination [20, 7, 23]. During inference, for each pixel in the current frame, the label is predicted by aggregating labels of its most similar pixels from previous frames, where the pixel similarity is computed using representation $\mathbf{R}$ (see Section 4.4 for details).

Challenges for tracking similar-looking objects. Label propagation for visually similar objects demands capturing robust motion signals, as appearance cues can become ambiguous and misleading. While prior studies excel at tracking objects using appearance features [23, 36, 31], they often fail when tracking multiple, similar-looking objects. We find this is due to their excessive dependence on appearance—a shortcut effective for distinct objects but a point of failure when objects are visually alike.

To illustrate this, we begin our investigation with a controlled toy experiment featuring two identical-looking, independently moving balls (Figure 2(a)). Given the ground truth segmentation map of each



(a) Handle long videos with sliding windows (b) Motion-aware representation for each window (c) Track via label propagation

Figure 3: **Framework.** Our work tracks objects via video label propagation, which transfers ground truth label of the first frame to subsequent frames. As video diffusion models typically have a maximum input length, we first divide the long video into overlapping video windows (see (a)). For each window, we use video diffusion models to to extract frame representations that capture rich inter-frame motion features(see (b)). Specifically, our method uses the 3D UNet backbone that can process the entire video sequence along the temporal axis. Finally, to predict the label for a query pixel i in the target frame ($\mathbf{R}^t$), we follow prior studies to aggregate the labels of its most similar pixels in previous frames (see (c); details in Section 4.4). We term our method **T**emporal **E**nhanced **D**iffusion tracking framework (**TED**). Experiments demonstrate that our TED improves tracking performance across diverse video scenarios, including those with similar-looking objects.

ball in the first frame (Figure 2(b), left), the task is to predict pixel-level labels of subsequent frames. Since the balls are identical, motion is the only signals that the trackers can rely on to make the right prediction. Figure 2(c) shows that state-of-the-art methods [45, 24, 31] struggle with object identity, leading to poor tracking. Moreover, we experiment on real-world videos with similar-looking objects. As shown in Figure 1(c), state-of-the-art trackers [45, 24, 31] fail to distinguish similar-looking deer when they swap positions. These findings highlight the difficulty of tracking multiple similar-looking objects in video label propagation.

4 Temporal-Enhanced Diffusion for Tracking

In Section 3, we show that state-of-the-art methods fail to track visually similar objects, highlighting the difficulty of self-supervised tracking when visual cues are ambiguous. To address this challenge, we propose a new Temporal-Enhanced Diffusion tracking framework (TED). We show that video representations for similar-objects tracking do not necessarily to be learned from scratch with tracking-specific objectives. Our TED leverages the motion intelligence from a pre-trained video diffusion model, enabling robust tracking of similar-looking objects.

In this section, we first introduce the tracking setup and video diffusion models. We then show how we obtain motion-aware representations without any tracking-specific objectives, and how to complement them with appearance features for further improvement. Finally, we show how these representations yield tracking results by label propagation.

4.1 Preliminaries: Tracking Setup and Video Diffusion Models

We focus on video label propagation task [47] as defined in Section 3. Following prior work [27], we aim to learn a frame representation, $\mathbf{R}^t$, for each frame I_t of a video, such that the similarity between representations reflects the true correspondence of pixels across frames.

Our approach builds upon video diffusion models, which are often obtained by adding a temporal dimension to image diffusion models, using 3D architectures to capture temporal context. Video diffusion models are trained to generate realistic video sequences by learning to reverse a diffusion process [21, 39, 4]. This process involves adding Gaussian noise to a clean video $\mathbf{X}^0$ at different noise levels, indicated by step τ. The model, ϵ_θ, is trained to predict the added noise at each step τ, minimizing the following loss:

$$L = \mathbb{E}_{\mathbf{X},\boldsymbol{\epsilon}\sim\mathcal{N}(0,1),\tau}\left[\|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{X}^\tau, \tau)\|_2^2\right] \tag{1}$$

where

$$\mathbf{X}^\tau = \sqrt{\alpha_\tau}\mathbf{X}^0 + \sqrt{1-\alpha_\tau}\,\varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1) \tag{2}$$

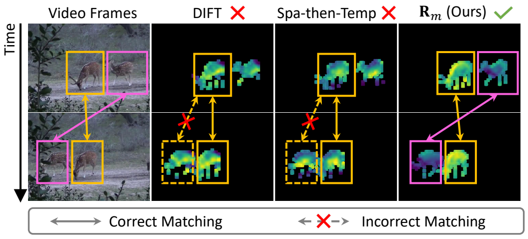


Figure 4: **Representation visualization.** We show a video with two similar looking deer swapping positions in the first column, and PCA results of model representations in column 2-4 (similar pixel colors indicate similar features). While state-of-the-art Spa-then-Temp [31] and DIFT [45] obtain similar features for both deer, our $\mathbf{R}_m$ distinguishes each deer by their different motions.

Here, α_τ is a noise schedule parameter, with larger τ indicating higher noise levels. $\mathcal{N}(0, 1)$ denotes the Gaussian distribution. This training process forces the model to learn not only the visual content of individual frames, but also the coherent motion that connects them. Given a noisy video $\mathbf{X}^\tau$, the model obtains a cleaner $\mathbf{X}^{\tau-1}$ by removing the predicted noises in $\mathbf{X}^\tau$, termed denoising process. In this work, we study video diffusion models using the widely-used 3D UNet as ϵ_θ, which is built by inserting temporal layers, such as temporal attention and 3D convolution, into a 2D UNet [41]. Our work is also model-agnostic, which adapts to any pretrained video diffusion model that may not use a 3D UNet.

4.2 Space-time Learning for Motion-aware Representations

We begin by investigating the internal feature activations of video diffusion models. We find that high noise levels during the denoising process encode a rich representation of motion, while low noise levels primarily capture appearance information (see Figure 6 and Section 5.3). This discovery motivates us to a novel approach for object tracking. We leverage the internal feature activations (ε_θ) at these high noise levels to extract robust motion cues.

Formally, given a video sequence $X = \{I_1, I_2, \dots, I_N\}$, we first add noise to obtain $\mathbf{X}^\tau$ (following Equation 2). Then, we perform a single forward pass through the UNet of video diffusion models (UNet$_v$) to obtain features from n_v layer, with entire video $\mathbf{X}^\tau$ as input (Figure 3(b)):

$$\mathbf{R}^1, \mathbf{R}^2, \dots, \mathbf{R}^N = \text{UNet}_v(\mathbf{X}^\tau, n_v) \tag{3}$$

where τ represents the noise level. Crucially, unlike prior methods that compute $\mathbf{R}^t$ by independently processing each frame ($\mathbf{R}^t = F(I_t)$) using 2D image encoders, our video diffusion model's features $\mathbf{R}^1, \mathbf{R}^2, \dots, \mathbf{R}^N$ incorporate temporal information through temporal attention and 3D convolutions. Consequently, each $\mathbf{R}^t$ represents not only the appearance of frame I_t , but also inter-frame motion dynamics captured within the representation from high-noise inputs, enabling effective tracking.

Handling long videos: sliding window approach. Video diffusion models typically have a maximum input length, L. To handle videos longer than that, inspired by temporal segment networks [49], we adopt a sliding window method, as shown in Figure 3(a). The video $X = \{I_1, I_2, \dots, I_N\}$ is divided into multiple overlapping short video clips $\{X_k\}$ with window size L.

$$\begin{aligned} X_1 &= \{I_1, I_2, \dots, I_L\}, \\ X_2 &= \{I_{1+L-\text{overlap}}, I_{2+L-\text{overlap}}, \dots, I_{L+L-\text{overlap}}\}, \\ &\vdots \end{aligned} \tag{4}$$

where $0 \leq \text{overlap} < L$. By integrating the temporal knowledge from the 3D UNet, for each frame representation $\mathbf{R}^t$ in the clip, it encodes the temporal motions from all frames in the current video clip X_k. Overlapping frames further improve motion consistency among video clips.

Visualization of motion-aware representations. After obtaining representations from video diffusion models ($\mathbf{R}_m$), we study if $\mathbf{R}_m$ *can differentiate similar-looking objects*. We visualize both our representations $\mathbf{R}_m$ and representations from state-of-the-art methods [31, 45] in Figure 4, where two similar-looking deer swapping positions over time. We perform principal component analysis (PCA) [33] on two frames (denoted s and t) for each model (e.g., $\tilde{\mathbf{R}}_m^s, \tilde{\mathbf{R}}_m^t = \text{PCA}(\mathbf{R}_m^s \parallel \mathbf{R}_m^t)$ for our $\mathbf{R}_m$). In Figure 4, similar pixel colors indicate similar representations. Figure 4 shows that prior methods [31, 45] capture similar features for different deer, indicated by similar colors. In contrast, our $\mathbf{R}_m$ learns clearly distinct features for each deer, shown as different color. These results highlight the superiority of our method in capturing object motions, enabling tracking similar-looking objects. Note that PCA is used only for visualization, and the original $\mathbf{R}_m$ is used for tracking.

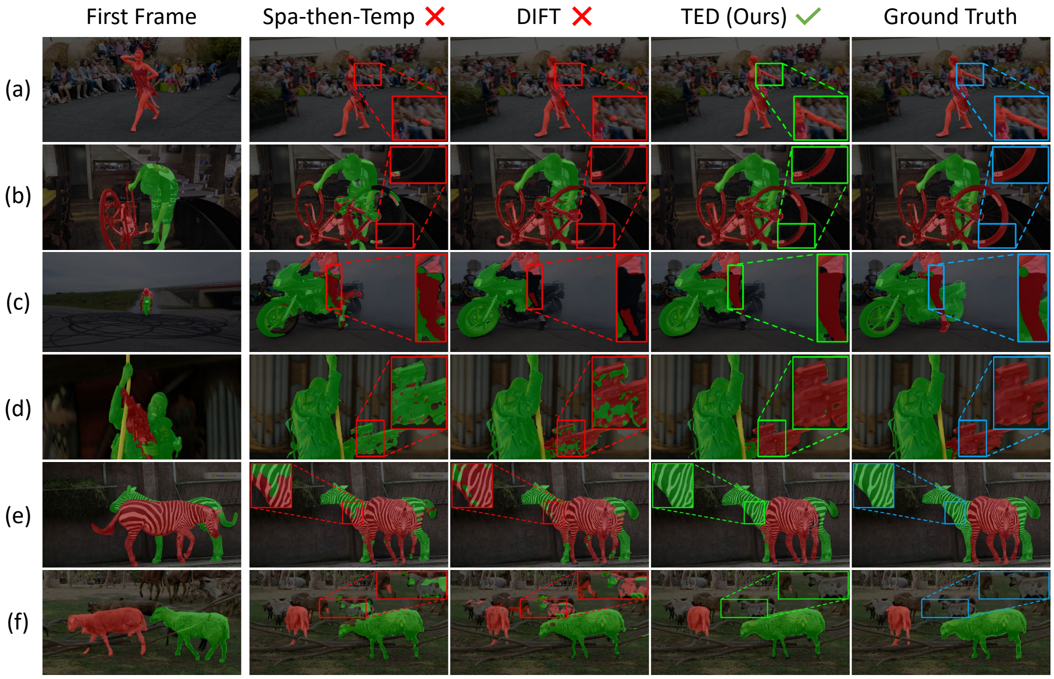


Figure 5: **Predictions for pixel-level object tracking.** We evaluate TED on the video label propagation task, comparing its predicted segmentation maps with those from state-of-the-art methods [31, 45]. Our TED consistently outperforms both methods [31, 45] on DAVIS (Figure a-d) and YouTube-Similar (Figure e-f) datasets, aligning with Table 1. Notably, our TED delivers more accurate predictions in scenarios with complex deformations (a) and viewpoint changes (b), while Spa-then-Temp [31] and DIFT [45] struggle with tracking completeness, e.g., the missing arm in (a). Our TED also achieves superior tracking in multi-object scenarios, such as interacting objects (c-d) and similar-looking objects (e-f). In contrast, Spa-then-Temp [31] and DIFT [45] have mislabeling issues, such as incorrect labels for the gun in (d) and misaligned labels for sheep in the background (f). These results show that our TED significantly improves tracking performance, highlighting the superiority of our motion-aware representations in tracking. (**Best viewed when zoomed in.**)

4.3 Motion Meets Appearance for Robust Tracking

While the motion-aware representations extracted from the video diffusion model ($\mathbf{R}_m$) are powerful, they are not the only source of useful information for tracking. Appearance cues remain important, particularly for distinguishing objects that are not identical. Inspired by the Two-Stream ConvNets [44], we combine the orthogonal motion ($\mathbf{R}_m$) features from video diffusion models and appearance ($\mathbf{R}_a$) features from a pre-trained image diffusion model [45]:

$$\mathbf{R}_f = \text{concat}\left(\lambda \cdot \frac{\mathbf{R}_m}{\|\mathbf{R}_m\|_2}, (1-\lambda) \cdot \frac{\mathbf{R}_a}{\|\mathbf{R}_a\|_2}\right) \tag{5}$$

where $\|\cdot\|$ denotes L2 normalization, and λ is a weighting factor (between 0 and 1) that controls the relative importance of motion and appearance. Different from $\mathbf{R}_m$, which learns inter-frame features as defined in Equation 3, $\mathbf{R}_a$ is obtained by feeding each frame independently into the 2D UNet$_i$ of an image diffusion model. Specifically, for each frame I_t, we compute its appearance feature as $\mathbf{R}_a^t = \text{UNet}_i(I_t^\tau, n_i)$, where I_t^τ is computed by adding noise to I_t, and n_i is the block index within the image diffusion model for feature representation. We refer to this combined approach using frame representation $\mathbf{R}_f$ for tracking as Temporal-Enhanced Diffusion tracking method (TED).

4.4 Tracking via Label Propagation

To perform tracking (i.e., label propagation), we follow the standard protocol used in previous work [52, 24, 23]. Given the ground truth labels in the first frame I_1, we use a recurrent method to propagate the labels to subsequent frames from I_2 to I_N, based on frame representations ($\mathbf{R}_f$).

Figure 3(c) shows how we predict the label for a query pixel i in the target frame ($\mathbf{R}^t$). Define the first frame and the previous m frames as reference frames, we first compute the pairwise similarities between pixel i and pixels in the reference frames. In Figure 3(c), we show the case with one reference frame, with representation termed $\mathbf{R}^r$. Following prior studies [24, 23, 31], we restrict the similarity computation to a spatially local neighborhood $\mathcal{S}(i)$ around i. This yields a similarity matrix A_{tr}, where each element $A_{tr}(i,j)$ is the dot product of the representations of pixel i in $\mathbf{R}^t$ and pixel j in $\mathbf{R}^r$ (with $j \in \mathcal{S}(i)$). To identify the most similar pixels to i, we retain only the top-K values from A_{tr} to form A'_{tr}, setting all other values to zero. Finally, the label for pixel i is predicted by aggregating the labels from its most similar pixels in the reference frames using a weighted sum:

$$y_t(i) = \sum_r \sum_{j \in \mathcal{S}(i)} A'_{tr}(i,j) \cdot y_r(j) \tag{6}$$

The pixel-level labels y_t is computed in the representation space and then interpolated to match the size of video frames following [24, 23, 31]. We show the pseudocode of our TED in Appendix A.

4.5 Implementation Details

Motion-aware representations $\mathbf{R}_m$. Any pretrained video diffusion models can serve as our motion feature backbone. We default to the widely-used I2VGen-XL [58] and also explore Stable Video Diffusion [2]. Since I2VGen-XL supports up to 16 frames, longer videos are split into multiple 16-frame clips. We pass each clip through model's 3D UNet and extract features from the third block as $\mathbf{R}_m$. The diffusion step τ for computing model input $\mathbf{X}^\tau$ is chosen empirically.

Appearance-aware representations $\mathbf{R}_a$. Our TED framework supports any frame representations that learn appearance features. Different from $\mathbf{R}_m$ that takes the entire video sequence as model input (Section 4.2), we obtain $\mathbf{R}_a$ for each frame by inputting the image independently to the image encoder (i.e., $\mathbf{R}^t = F(I_t)$). We default to image-diffusion model ADM [14] as image encoder, extracting features from its eighth block as $\mathbf{R}_a$. We also test Stable Diffusion [39].

Tracking via label propagation. Our TED method uses $\mathbf{R}_f$, a fusion of $\mathbf{R}_m$ and $\mathbf{R}_a$, to obtain tracking results. We follow the setups of prior studies [24, 23, 45] for label propagation, with pseudocode and details in Appendix A and Appendix B.1.

5 Experiments

5.1 Experimental Setups

Baselines. Our TED advances self-supervised tracking without any labeled training data. We evaluate on video label propagation task, and compare against 17 state-of-the-art self-supervised methods.

`Self-supervised representation learning from images.` We evaluate 7 models that learn appearance features by training on independent images. We consider instance-discrimination methods, e.g., MoCo [20]. We also test SFC [23], a strong baseline that integrates image-level and pixel-level cues, and DIFT [45], which leverages knowledge from image diffusion models [40].

`Self-supervised representation learning from videos.` We benchmark 10 models that incorporate temporal cues to training through diverse pretext tasks. We evaluate on strong baselines trained for frame reconstruction (e.g., UVC [32]), cycle consistency (e.g., CRW [24]), and video contrastive learning (e.g., VFS [55]). We also include recent Spa-then-Temp [31] and SMTC [36].

Datasets. Our method uses pretrained diffusion models for tracking, without additional training. We benchmark on following test sets, with video examples in Appendix B.2. We follow prior studies [24, 36, 31] to report region similarity ($\mathcal{J}_m$) and contour accuracy ($\mathcal{F}_m$) for evaluation.

`Standard benchmark.` We follow previous work [32, 27, 52, 24, 36, 31] and evaluate on the widely-used DAVIS-2017 validation set [35], which contains 30 videos (2023 frames, 59 objects).

Table 1: **Results for pixel-level similar-looking object tracking task.** Our TED advances *self-supervised* tracking without any labeled data. We evaluate it on the video label propagation task against 17 state-of-the-art self-supervised methods. 'Temporal Train' indicates whether the method uses temporal signals during training. Colored numbers indicate the **best** results. Our TED achieves significant improvements across all datasets. On the widely-used DAVIS benchmark [35], our TED outperforms recent methods by up to 6%. On Youtube-Similar, featuring real-world similar-looking objects, our TED achieves an even larger gain of 10%. On Kubric-Similar, with two identical-looking, independently moving balls, TED reaches a high $\mathcal{J}_\text{m}$ of 87.2%, while most methods stay near 50%, equivalent to random guessing due to the objects' identical sizes. These results highlight the effectiveness of our TED in tracking similar-looking objects.

Temporal Train	Method	DAVIS			Youtube-Similar			Kubric-Similar		
		$\mathcal{J}\&\mathcal{F}_\text{m}(\uparrow)$	$\mathcal{J}_\text{m}(\uparrow)$	$\mathcal{F}_\text{m}(\uparrow)$	$\mathcal{J}\&\mathcal{F}_\text{m}(\uparrow)$	$\mathcal{J}_\text{m}(\uparrow)$	$\mathcal{F}_\text{m}(\uparrow)$	$\mathcal{J}\&\mathcal{F}_\text{m}(\uparrow)$	$\mathcal{J}_\text{m}(\uparrow)$	$\mathcal{F}_\text{m}(\uparrow)$
×	InstDis [54]	66.4	63.9	68.9	-	-	-	-	-	-
	MoCo [20]	65.9	63.4	68.4	48.0	48.5	47.4	56.6	51.6	61.6
	SimCLR [6]	66.9	64.4	69.4	37.5	36.9	38.1	55.6	50.3	60.9
	BYOL [19]	66.5	64.0	69.0	47.1	47.7	46.5	54.8	49.2	60.5
	SimSiam [7]	67.2	64.8	68.8	47.4	47.9	47.0	58.4	52.6	64.1
	SFC [23]	71.2	68.3	74.0	55.5	55.3	55.7	47.7	43.1	52.3
	DIFT [45]	75.7	72.7	78.6	60.7	59.8	61.7	55.1	52.7	57.6
✓	Colorization [47]	34.0	34.6	32.7	-	-	-	-	-	-
	TimeCycle [52]	48.7	46.4	50.0	39.8	41.3	38.2	50.6	44.0	57.2
	CorrFlow [28]	50.3	48.4	52.2	39.6	40.0	39.3	32.6	27.0	38.3
	UVC [32]	60.9	59.3	62.7	49.7	49.8	49.7	56.9	51.3	62.6
	VINCE [17]	65.2	62.5	67.8	44.9	45.4	44.3	54.1	48.5	59.7
	MAST [27]	65.5	63.3	67.6	-	-	-	-	-	-
	CRW [24]	67.6	64.8	70.2	52.0	52.3	51.6	54.9	49.7	60.1
	VFS [55]	68.9	66.5	71.3	57.3	57.1	57.5	44.2	38.5	49.9
	SMTC [36]	73.0	69.4	76.6	57.5	57.2	57.9	68.6	64.7	72.5
	Spa-then-Temp [31]	74.1	71.1	77.1	59.6	59.2	60.1	48.9	44.0	53.8
✓	**TED (Ours)**	**77.6**	**74.4**	**80.8**	**66.0**	**65.1**	**67.0**	**90.2**	**87.2**	**93.1**

`Real-world similar-looking benchmark.` We introduce Youtube-Similar, including 28 videos featuring similar-looking objects from Youtube-VOS [56], totally 839 frames and 69 objects.

`Controlled identical-object benchmark.` In real-world videos, visually similar objects can still differ due to factors like gestures. To eliminate these variations, we introduce Kubric-Similar, including 30 videos (480 frames, 60 objects) in which two identical-looking balls move independently. The dataset is generated by Kubric simulator [18], with random ball colors, sizes, and motions.

5.2 Experimental Results

Quantitative results. We compare our TED method with 17 self-supervised methods in Table 1. Our TED achieves the **state-of-the-art** tracking performance on **all** datasets. On the standard DAVIS dataset, our TED significantly outperforms recent methods by up to 6%, such as SFC [5] by 6.4%, SMTC [36] by 4.6%, Spa-then-Temp [31] by 3.5% and DIFT [45] by 1.9%. By introducing motion-aware features from video diffusion models, our TED achieves an even greater improvement when tracking similar-looking objects on Youtube-Similar, such as Spa-then-Temp [31] by 6.4% and DIFT [45] by 5.3%. On Kubric-Similar that includes identical objects, many methods achieve a J_m around 50%, no better than random guessing due to identical sizes of two balls. By contrast, our TED achieves a high $\mathcal{J}_\text{m}$ of 87.2%. These improvements highlight the effectiveness of our method in object tracking, even for challenging settings with multiple similar-looking objects.

Visualizations. Figure 5 compares our tracking results with state-of-the-art methods on the DAVIS dataset (a-d) and YouTube-Similar (e-f). Our TED approach significantly outperforms prior methods, aligning with Table 1. Our TED effectively handles complex deformations (Figure 5(a)) and viewpoint changes (Figure 5(b)), while Spa-then-Temp [31] and DIFT [45] struggle with elements like the human arm (Figure 5(a)). Our TED also excels in multi-object scenarios, such as interacting objects (Figure 5(c-d)) and similar-looking objects (Figure 5(e-f)). By contrast, Spa-then-Temp [31] and DIFT [45] often confuse different objects, leading to incorrect tracking results. For example, in Figure 5(d), Spa-then-Temp mislabels a gun as a human and DIFT shows significant contour errors. In Figure 5(f), both Spa-then-Temp and DIFT mistakenly assign the target label to a background

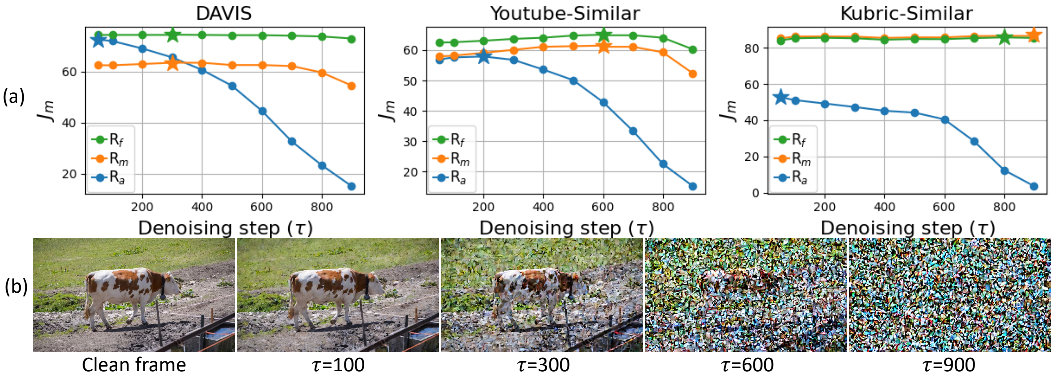


Figure 6: **Tracking results under different denoising steps.** We evaluate tracking performance using model inputs $\mathbf{X}^\tau$ at various denoising steps τ, where larger τ indicates more noise (see (b)). The performance of appearance features $\mathbf{R}_a$ degrade significantly as τ increases, while our motion feature $\mathbf{R}_m$ maintains high tracking accuracy even with a large τ. Notably, $\mathbf{R}_m$ peaks at τ=600 on Youtube-Similar and τ=900 on Kubric-Similar, where appearance cues are almost available. These results reveal that video diffusion models can learn object motions from highly noisy inputs, enabling effective, motion-aware tracking.

sheep. These results demonstrate that our TED significantly outperforms prior methods across various scenarios, highlighting the superiority of motion-aware features in our work.

5.3 Analysis and Ablation Studies

The impact of diffusion features from different noisy levels on motion-based tracking. We evaluate tracking performance using model inputs $\mathbf{X}^\tau$ at various denoising steps τ (see Equation 2 and Equation 3), as shown in Figure 6. We will show that video diffusion models capture object motions even at early denoising steps when the input $\mathbf{X}^\tau$ is highly noisy.

Figure 6(a) shows that the tracking performance of appearance feature $\mathbf{R}_a$ drops significantly with larger τ (e.g., $\tau \geq 600$). This is because $\mathbf{X}^\tau$ is heavily corrupted at high noise levels, as shown in Figure 6(b), thus appearance features are almost unavailable. In contrast, our motion feature $\mathbf{R}_m$ achieves high tracking results at high noise levels. Interestingly, $\mathbf{R}_m$ even achieves its best performance (marked by a star in the figure) at τ=600 on Youtube-Similar and τ=900 on Kubric-Similar, when $\mathbf{R}_a$ almost fails. While motion features are crucial for identifying similar-looking objects, appearance features provide fine-grained details for accurate segmentation. This explains why our fused representation $\mathbf{R}_f$ outperforms motion-only features $\mathbf{R}_m$ in real-world data, highlighting the benefit of jointly leveraging motion and appearance cues for tracking.

An interesting question is: how do video diffusion models learn object motions with highly noisy input $\mathbf{X}^\tau$? With loss defined in Equation 1, diffusion models are trained to reconstruct clean input from its noisy counterparts. To achieve this goal, they solve different tasks at different noise levels [11]. When $\mathbf{X}^\tau$ is highly corrupted at high noise levels, video diffusion models are trained to solve the hard task that learns coarse-grained signals in the video, such as motion (e.g., changes of object positions among frames). Therefore, its representation $\mathbf{R}_m$ encodes rich motion information that enables effective tracking of similar-looking objects. When input $\mathbf{X}^\tau$ is less noisy, diffusion model is trained to denoise appearance details, where motion features are also learned but may not be so prioritized, leading to performance decrease at low noise levels. Our analysis and results provide new insights into both tracking and video diffusion models.

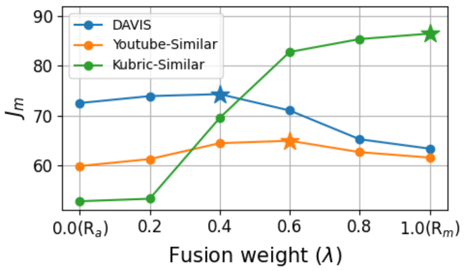


Figure 7: **Fusion weight (λ).** Our method integrates the advantages of motion and appearance features. For dataset where visual clues are ambiguous, such as Kubric, more video diffusion features are important in improving the accuracy.

The effect of coefficients for combining motion and appearance. Figure 7 shows tracking accuracy with varying fusion weight λ (see Equation 5), where λ=1 gives $\mathbf{R}_f$ = $\mathbf{R}_m$ while λ=0 gives $\mathbf{R}_f$ = $\mathbf{R}_a$. On Kubric-Similar with visually identical balls, motion features solely are sufficient for successful tracking. Our results align with this expectation by achieving the best result with λ=1.0. On real-world DAVIS and Youtube-Similar, our $\mathbf{R}_f$ performs best with a moderate λ value around 0.5. These results show that our $\mathbf{R}_f$ effectively integrates the advantages of motion and appearance features in complex scenarios, outperforming the case of using either $\mathbf{R}_m$ or $\mathbf{R}_a$ alone.

The effect of layers for feature representation. We extract representations from internal layers of video and image diffusion models for tracking, with block indices denoted as n_v and n_i as in Section 4. Our framework is agnostic to specific layers. Motion and appearance features can be taken from different layers, and the optimal layers for the two backbones do not need to be the same. Following [45], we use the decoder representations from UNet. We report the tracking results using $\mathbf{R}_m$ alone from different decoder blocks in Table 2. Table 2 shows that the medium block (block 3) yields the best performance among all blocks on the DAVIS dataset.

Table 2: **Block indexes.** Motion representation $\mathbf{R}_m$ achieves the best tracking results when extracted from the third block of pre-trained I2VGen-XL [58].

Block	$\mathcal{J}\&\mathcal{F}_\mathrm{m}(\uparrow)$	$\mathcal{J}_\mathrm{m}(\uparrow)$	$\mathcal{F}_\mathrm{m}(\uparrow)$
1	24.8	28.2	21.4
2	47.6	52.7	42.5
3	**66.3**	**63.4**	**69.1**
4	31.5	27.2	35.8

The impact of overlapping frames on tracking. Motion features are crucial for successful tracking of similar-looking objects. Figure 8 shows the tracking accuracy on Youtube-Similar versus overlapping frames (l) among video clips. Compared to the non-overlapping case (l=0), introducing overlapping frames (l>0) achieves higher tracking accuracy due to improved motion consistency among video clips. At the same time, a small l (e.g., l=2) is sufficient for good performance since tracking accuracy improves marginally with higher values of l. These results highlight the importance of accurate inter-frame motions in frame representations when tracking similar-looking objects.

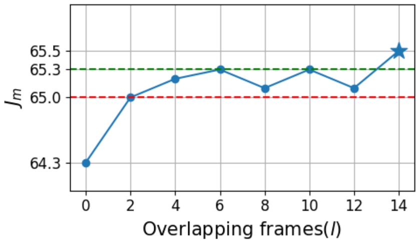


Figure 8: **Overlapping frames among video clips (l).** A small l (e.g., l=2) is sufficient for boosting tracking accuracy, highlighting the importance of using multiple input frames in capturing motion clue for our method.

The effect of diffusion models. We evaluate our TED using representations from different diffusion models on DAVIS dataset, as shown in Table 3. Our TED achieves the best tracking results using motion features from I2VGen-XL [58] and appearance features from ADM [14], which are used by default in this work.

Table 3: **Pretrained diffusion models.**. Our TED achieves the best tracking results using representations from I2VGen-XL [58] and ADM [14].

$\mathbf{R}_m$	$\mathbf{R}_a$	$\mathcal{J}\&\mathcal{F}_\mathrm{m}(\uparrow)$	$\mathcal{J}_\mathrm{m}(\uparrow)$	$\mathcal{F}_\mathrm{m}(\uparrow)$
SVD [2]	SD [39]	71.5	68.9	74.1
SVD [2]	ADM [14]	76.6	73.6	79.7
I2VGen [58]	SD [39]	71.7	69.0	74.5
I2VGen [58]	ADM [14]	**77.6**	**74.4**	**80.8**

Computation cost analysis. Our method has 20% more computation time than generative model method DIFT [45] on DAVIS videos and 89% than the popular self-supervised discriminative model method SFC [23]. For memory, our method uses 64% more than DIFT and 705% more than SFC. See more details in Appendix B.3).

6 Conclusion

We demonstrate that video diffusion models excel at tracking similar-looking objects without any task-specific training. We show that pre-trained video diffusion models possess an inherent, previously unrecognized ability to encode motion information at high noise levels. Rather than designing complex architectures or training objectives for tracking, we simply extract this readily available motion representation, achieving state-of-the-art tracking performance. Our approach achieves significant improvement over prior methods in diverse scenarios, such as challenging viewpoint changes and deformations. Our work opens new avenues for leveraging the latent capabilities of diffusion models beyond generation.

Acknowledgments and Disclosure of Funding

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics), and by the ITRC (Information Technology Research Center) grant funded by the Korean government (IITP-2025-RS-2023-00259991).

References

[1] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021.

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[4] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[8] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3151–3161, 2024.

[9] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022.

[10] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021.

[11] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022.

[12] Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero shot classifiers. *Advances in Neural Information Processing Systems*, 36:58921–58937, 2023.

[13] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[15] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 20224–20234, 2023.

[16] Junqiao Fan, Jianfei Yang, Yuecong Xu, and Lihua Xie. Diffusion model is a good pose estimator from 3d rf-vision. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024.

[17] Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. Watching the world go by: Representation learning from unlabeled videos. *arXiv preprint arXiv:2003.07990*, 2020.

[18] 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.

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[22] Karl Holmquist and Bastian Wandt. Diffpose: Multi-hypothesis human pose estimation using diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15977–15987, 2023.

[23] Yingdong Hu, Renhao Wang, Kaifeng Zhang, and Yang Gao. Semantic-aware fine-grained correspondence. In *European Conference on Computer Vision*, pages 97–115. Springer, 2022.

[24] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems*, 33:19545–19560, 2020.

[25] Hyeonho Jeong, Chun-Hao P Huang, Jong Chul Ye, Niloy J Mitra, and Duygu Ceylan. Track4gen: Teaching video diffusion models to track points improves video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7276–7287, 2025.

[26] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013.

[27] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2020.

[28] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019.

[29] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2206–2217, 2023.

[30] Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdqunat: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024.

[31] Rui Li and Dong Liu. Spatial-then-temporal self-supervised learning for video correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2279–2288, 2023.

[32] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. *Advances in Neural Information Processing Systems*, 32, 2019.

[33] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993.

[34] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[35] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[36] Rui Qian, Shuangrui Ding, Xian Liu, and Dahua Lin. Semantics meets temporal correspondence: Self-supervised object-centric learning in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16675–16687, 2023.

[37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[38] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[39] 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.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[42] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[44] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[45] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023.

[46] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1420–1429, 2016.

[47] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 391–408, 2018.

[48] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

[49] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[50] Qian Wang, Abdelrahman Eldesokey, Mohit Mendiratta, Fangneng Zhan, Adam Kortylewski, Christian Theobalt, and Peter Wonka. Vidseg: Training-free video semantic segmentation based on diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22985–22994, 2025.

[51] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019.

[52] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2566–2576, 2019.

[53] Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025.

[54] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[55] 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.

[56] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[57] 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.

[58] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qing, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. 2023.

[59] Zhengbo Zhang, Li Xu, Duo Peng, Hossein Rahmani, and Jun Liu. Diff-tracker: text-to-image diffusion models are unsupervised trackers. In *European Conference on Computer Vision*, pages 319–337. Springer, 2024.

[60] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5729–5739, 2023.

[61] Qixian Zhou, Xiaodan Liang, Ke Gong, and Liang Lin. Adaptive temporal encoding network for video instance-level human parsing. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1527–1535, 2018.

A Pseudocode of Our Temporal Enhanced Diffusion Tracking Method (TED)

We provide the pseudocode of our Temporal Enhanced Diffusion tracking method (TED) in Algorithm 1. For clarity, we denote the process of obtaining noisy input $\mathbf{X}^{\tau}$ for video diffusion model in Equation 2 of Section 4.1 as the function AddNoiseVD. We also term the similar process for image diffusion models that adds noise to image inputs as AddNoiseID.

B Experimental Setups and Results

B.1 Tracking Setups for Label Propagation

We follow the experimental setups of prior studies [23, 45, 31] for video label propagation, as summarized in Table 4.

Table 4: Experimental setups of TED for video label propagation.

Dataset	Video diffusion			Image diffusion			Fusion	Softmax	Propagation	k for
	Model	Timestep	Block	Model	Timestep	Block	weight	temp	radius	top-k
DAVIS	I2VGen-XL	300	3	ADM	51	8	0.4	0.2	15	10
Youtube-Similar	I2VGen-XL	600	3	ADM	51	8	0.6	0.1	15	10
Kubric-Similar	I2VGen-XL	900	3	ADM	51	8	1.0	0.1	15	10

B.2 Datasets

Our method applies pretrained diffusion models for tracking, without additional training. We introduce the test sets in Section 5.1 and show video examples from each dataset in Figure 9.

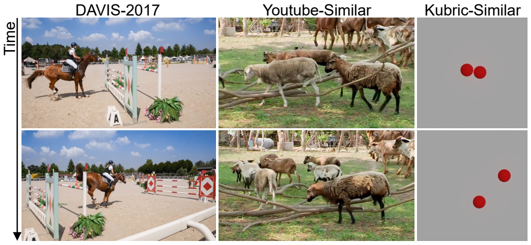


Figure 9: **Video examples from test sets.** Following prior studies [24, 36, 31, 45], we evaluate on the standard DAVIS-2017 benchmark [35] (first column). To evaluate tracking on visually similar objects, we introduce Youtube-Similar (second column), a real-world test set with similar-looking objects, and Kubric-Similar (third column), a controlled set with identical objects.

To ensure the quality of Youtube-Similar, three graduate students were hired to manually select videos from Youtube-VOS [56] according to the following rubrics. Only the videos that all annotators find qualified are included in the final YouTube-Similar dataset. The first rubric is object similarity. To create a dataset with similar-looking objects, we first select videos that contain at least two objects belonging to the same category from YouTube-VOS. The second rubric is video filtering. We exclude static videos from the pool obtained in the first stage. In such videos, spatial appearance features can serve as a shortcut for tracking, which may influence our evaluation of the motion features learned in model representations.

B.3 Computational Cost Analysis

We compare computation cost with prior methods in Table 5, tested on a single A100 GPU using DAVIS videos. Our full model achieves the highest accuracy (77.6%) using a time of 682 ms per frame, compared to DIFT (75.7%, 566 ms) and SFC (71.2%, 360 ms).

For each video, a technique to boost accuracy in our full method is averaging representations computed from multiple noisy inputs. We also provide an efficient variant by removing this averaging step, which runs at 521 ms and achieves 77.2% accuracy, still significantly outperforming prior methods. Our efficient version achieves a tradeoff between tracking accuracy and efficiency.

Our work is the first to show that video diffusion models can track similar-looking objects without tracking-specific training. Our finding that motion features are learned at high-noise stages also provides new insight into video diffusion models. Additionally, our method is compatible with

Algorithm 1: **T**emporal **E**nhanced **D**iffusion Tracking (TED)

```
Input: Video frames $I_1, I_2, \dots, I_N$; Ground-truth label $Y_1$ for $I_1$; Video diffusion model UNet$_v$;
       Image diffusion model UNet$_i$; Denoising steps: $\tau_v$ (video diffusion) and $\tau_i$ (image
       diffusion); Block index for features: $n_v$ (video diffusion) and $n_i$ (image diffusion);
       Fusion weight $\lambda$.
Output: Label predictions $Y_2, Y_3, \dots, Y_N$ for frames $I_2, \dots, I_N$.
Initialize a queue $Q \leftarrow \emptyset$ for storing representations and labels of reference frames;
Let $L$ be the maximum input length for UNet$_v$ and $l$ be the number of overlapping frames
  between video clips;
Divide the entire long video to $ClipNumber = \lfloor (N-L)/(L-l) \rfloor + 1$ video clips using
  sliding window approach.
for $k = 0$ to ClipNumber $- 1$ do
    Define current video clip $X_k = \{I_{1+k(L-l)}, \dots, I_{L+k(L-l)}\}$;
    Step 1: Compute Frame Representations
    (a) Motion-aware $\mathbf{R}_m$ : Compute $\mathbf{R}_m$ in one forward pass of UNet$_v$:
            $\mathbf{R}_m^{1+k(L-l)}, \dots, \mathbf{R}_m^{L+k(L-l)}$ = UNet$_v$(AddNoiseVD$(X_k, \tau_v), n_v)$;
    (b) Appearance-aware $\mathbf{R}_a$: Compute $\mathbf{R}_a$ in multiple forward pass of UNet$_i$:
            For each frame $I_t \in X_k$, compute $\mathbf{R}_a^t$ = UNet$_i$(AddNoiseID$(I_t, \tau_i), n_i)$;
    (c) Fused $\mathbf{R}_f$: For each frame $I_t \in X_k$, compute fused representation:
    $\mathbf{R}_f^t = \text{concat}\Big(\lambda \frac{\mathbf{R}_m^t}{\|\mathbf{R}_m^t\|_2}, (1-\lambda) \frac{\mathbf{R}_a^t}{\|\mathbf{R}_a^t\|_2}\Big)$
    Step 2: Predict Tracking Labels
    if $k = 0$ then
        Resize label $Y_1$ to match the size of $\mathbf{R}_f$, termed as as $y_1$. Add $(\mathbf{R}_f^1, y_1)$ to $Q$;
    for each frame $I_t \in X_k$ do
        if $(\mathbf{R}_f^t, y_t)$ is already in $Q$ then
            continue;
        for each query pixel $i$ in $\mathbf{R}_f^t$ do
            for each pixel $j$ from each reference frame $\mathbf{R}_r \in Q$ do
                if $j$ locates in the spatial neighborhood of pixel $i$ ($\mathcal{S}(i)$) then
                    Compute similarity score $A_{tr}(i,j)$ = DotProduct$(\mathbf{R}_f^t(i), \mathbf{R}_f^r(j))$ ;
            Identify the most similar pixels to $i$ by retaining the top-$K$ values in $A_{tr}$ and setting
              others as zero, obtaining $A'_{tr}$;
            Predict the label of pixel $i$ by: $y_t(i) = \sum_{r \in Q} \sum_{j \in \mathcal{S}(i)} A'_{tr}(i,j)\, y_r(j)$
        Add $(\mathbf{R}_f^t, y_t)$ to $Q$;
        if Size(Q) equals the maximum allowed reference frames then
            remove the oldest entry from $Q$;
        Interpolate $y_t$ to the original frame size to obtain $Y_t$;
return $Y_2, Y_3, \dots, Y_N$;
```

Table 5: **Computation cost analysis.** We report the tracking accuracy ($\mathcal{J}\&\mathcal{F}_\mathrm{m}$) and computation cost per frame on DAVIS. Our full method achieves the highest accuracy, while our efficient version achieves a tradeoff between tracking accuracy and efficiency.

Method	Accuracy	Time (ms)	Memory (GB)
SFC [23]	71.2	360	1.9
DIFT [45]	75.7	566	9.3
Ours (Efficient)	77.2	521	11.8
Ours (Full)	77.6	682	15.3

acceleration techniques such as FlashAttention [13] and quantization [30] for further speedup. We leave further efficiency optimization for future work.

B.4 Results on More Tasks and Datasets

We conduct experiments for human pose tracking on JHMDB dataset [26] and human part tracking on VIP dataset [61], as shown in Table 6. We also evaluate our method on YouTube-VOS [56] and MOSE [15] for video object segmentation, as shown in Table 7. Experimental results show that our method consistently improves tracking accuracy across diverse datasets and tracking tasks.

Table 6: Results on JHMDB for human pose tracking and VIP dataset for human part tracking.

Dataset	**JHMDB**		**VIP**
	PCK@0.1(↑)	PCK@0.2(↑)	mIoU (↑)
SFC [23]	61.9	83.0	38.4
Spa-then-temp [31]	66.4	84.4	41.0
DIFT [45]	63.4	84.3	43.7
TED (Ours)	**68.3**	**85.8**	**44.2**

Table 7: Results on YouTube-VOS and MOSE datasets for video object segmentation.

	YouTube-VOS			**MOSE**		
	$\mathcal{J}\&\mathcal{F}_\mathrm{m}(\uparrow)$	$\mathcal{J}_\mathrm{m}(\uparrow)$	$\mathcal{F}_\mathrm{m}(\uparrow)$	$\mathcal{J}\&\mathcal{F}_\mathrm{m}(\uparrow)$	$\mathcal{J}_\mathrm{m}(\uparrow)$	$\mathcal{F}_\mathrm{m}(\uparrow)$
DIFT [45]	70.5	68.2	72.7	34.5	28.9	40.1
TED (Ours)	**71.1**	**68.9**	**73.4**	**35.6**	**30.1**	**41.1**

C Discussions

Limitations and future work. Although our approach achieves significant tracking improvement across various scenarios. it comes with certain limitations. As discussed in Section 5.3 and Appendix B.3, using video and image diffusion models for tracking increases computational cost compared to previous methods. However, our work aim to show that video diffusion models can effectively track similar objects without tracking-specific training, suggesting a new direction for future trackers. Our finding that motions are learned at high-noise stages also advances the understanding of video diffusion models. One promising research direction is distilling motion intelligence from video diffusion models into smaller models for more efficient tracking. We leave further efficiency improvements for future work.

Broader impacts. Our work leverages the motion intelligence from video diffusion models and achieves the state-of-the-art tracking performance without task-specific supervision, reducing reliance on costly labeled data. This greatly benefit various applications, such as robotics and autonomous driving. Since our method leverages pretrained diffusion models for tracking, its performance may reflect biases present in those models, such as the underrepresentation of certain video types. We believe that mitigating bias in diffusion models is a promising research direction. We hope our work encourages further studies in this field, which benefit not only generative tasks but also perception tasks such as tracking.

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: Our work finds that video diffusion models excel at tracking similar-looking objects without task-specific training. Building on this insight, we propose a self-supervised tracking method, Temporal Enhanced Diffusion (TED), which significantly improves tracking performance across diverse scenarios. Our abstract and introduction reflect the paper's contributions and scope.

   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 discuss limitations in Appendix C.

   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: This work does 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: Our work is reproducible, and we introduce all the information needed to reproduce the main experimental results in Section 4, Section 5.1, Appendix A, Appendix B.1, and Appendix B.2.

   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: We will release the code and data.

   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 introduce experimental setups and details in Section 5.1, Appendix A, Appendix B.1, and Appendix B.2.

   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: [No]

   Justification: Our work uses pre-trained models without additional training. During tracking, our work averages video representations computed by a batch of noisy inputs for each video, achieving stable results.

   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 introduce the computation cost of proposed method in Section 5.3 and Appendix B.3.

   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: Our work conforms with the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: We discuss the broader impacts in Appendix C.

    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: Our work does not include data or models that appear to have a high risk for misuse.

    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 use public pretrained models and datasets. We also properly cite and introduce them Section 4.5 and Section 5.1.

    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: We introduce new datasets, Kubric-Similar and Youtube-Similar, in Section 5.1 and Appendix B.2.

    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: Our work does not include 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: This work 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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This work does not involve LLMs for the core method development.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.