POINT PROMPTING: COUNTERFACTUAL TRACKING WITH VIDEO DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in video generation have produced powerful diffusion models capable of generating high-quality, temporally coherent videos. We ask whether space-time tracking capabilities emerge automatically within these generators, as a consequence of the close connection between synthesizing and estimating motion. We propose a simple but effective way to elicit point tracking capabilities in off-the-shelf image-conditioned video diffusion models. We simply place a colored marker in the first frame, then guide the model to propagate the marker across frames, following the underlying video's motion. To ensure the marker remains visible despite the model's natural priors, we use the unedited video's initial frame as a negative prompt. We evaluate our method on the TAP-Vid benchmark using several video diffusion models. We find that it outperforms prior zero-shot methods, often obtaining performance that is competitive with specialized self-supervised models, despite the fact that it does not require any additional training.

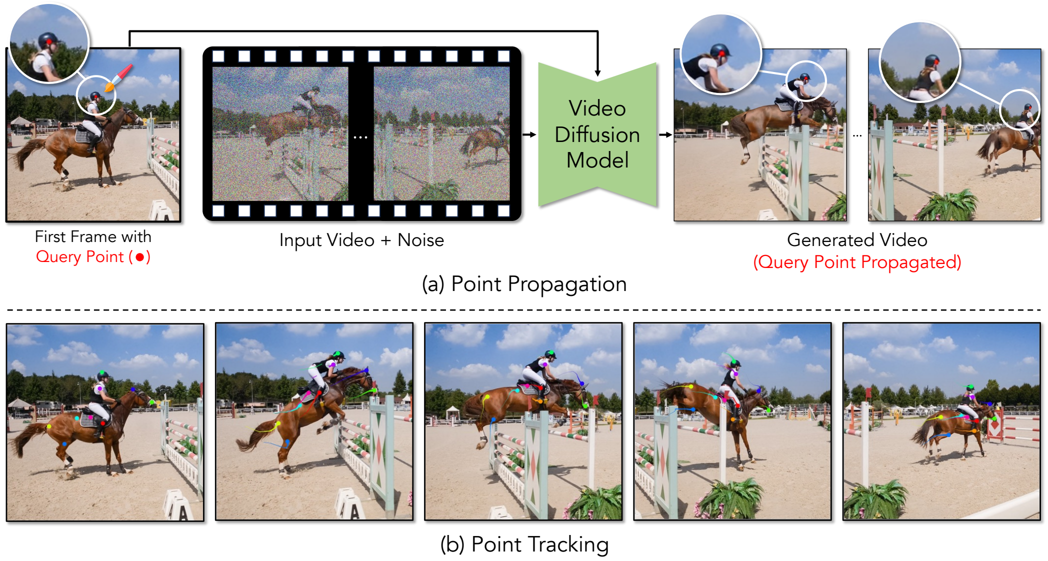
1 INTRODUCTION

Recent generative models have shown the remarkable ability to produce temporally consistent videos. The objects within them persist across frames, through occlusion, and despite variations in camera pose and lighting. These capabilities are closely related to the *visual tracking* problem. While generation deals with producing videos that contain temporally persistent objects, tracking deals with analyzing such videos to estimate motion. A variety of methods have exploited the connections between these two problems, such as by using trackers to supervise or control video generators (Chefer et al., 2025; Burgert et al., 2025; Geng et al., 2025; Hao et al., 2018; Ardino et al., 2021) and to evaluate the temporal consistency of generated videos by measuring how "trackable" they are (Allen et al., 2025; Lai et al., 2018; Ceylan et al., 2023; Geyer et al., 2023).

In this paper, we ask whether tracking capabilities *emerge automatically* in video diffusion models, as a consequence of the close connection between the two problems. Unlike high-level understanding tasks that are naturally described by captions, like object recognition, tracking cannot easily be induced by text prompting. To elicit these capabilities from a video generator, we propose a novel approach to *counterfactual modeling* that allows us to directly obtain high-quality point tracks "zero shot" from pretrained image-conditioned video diffusion models. We simply mark the position of the query point in the initial video frame using a distinctively colored dot (Fig. 1), then propagate it to future video frames by regenerating the video using SDEdit (Meng et al., 2021). After generation, the query point's position can be estimated in each frame by basic image processing.

In counterfactual modeling (Bear et al., 2023), one carefully perturbs the input variables, then analyzes how the generation changes in response. Yet large generative models have strong priors that sometimes conflict with this goal. The marker in Fig. 1, for example, may be unnatural in some environments, and so samples from a generative model may ignore it. We use a simple but effective method to address this issue: when sampling from the model, we use the unmodified initial input frame as a negative prompt for the diffusion model, thereby guiding the model toward samples that contain the marker.

Our approach is closely related to (and takes inspiration from) a recent line of work that applies counterfactual modeling to self-supervised motion estimation (Bear et al., 2023; Venkatesh et al., 2023). These methods train a future prediction model, then measure how the predicted future changes when a given point is perturbed in the initial frame, indicating its motion. This requires training a special-purpose model (based on masked autoencoders) that is designed specifically with this



(a) Point Propagation

(b) Point Tracking

Figure 1: **Prompting a diffusion model for tracking**. (a) We use an off-the-shelf video diffusion model to perform point tracking. We add a small, distinctive marking—a red dot—to the first frame of an input video, then ask the diffusion model to regenerate the rest of the video using SDEdit (Meng et al., 2021), which propagates the marking to subsequent frames. (b) We then track the motion of this marking over time. This motion corresponds to the trajectory of the underlying physical point. The model successfully tracks through occlusion. We show results for point propagation and tracking at https://iclr-2026-demo.github.io/project-demo .

downstream use case in mind, and requires training auxiliary models to obtain high performance. By contrast, we show that *off-the-shelf* video diffusion models can track points by prompting. In this way, our work is closely related to “zero shot” emergent correspondence methods (Tang et al., 2023; Zhang et al., 2023a). However, previous methods rely on extracting internal *feature* representations from image, are highly architectural specific, and largely deal with semantic correspondence from image diffusion models.

Our results suggest that video diffusion models are capable of tracking points through video via counterfactual modeling, without need for additional training. Through experiments on the TAP-Vid (Doersch et al., 2022) benchmark, we show:

- Pretrained video diffusion models can be directly used as visual trackers.
- The object permanence capabilities of generative models enable tracking through occlusion.
- Points can be reliably propagated through video using a novel diffusion prompting strategy.
- Tracking performance can improve through iterative refinement using inpainting.
- We significantly outperform previous zero-shot tracking methods, such as those that use features from pretrained image diffusion models.

We see this work as being a step toward understanding the capabilities of large, pretrained video diffusion models, and new ways to extract these capabilities from them.

2 Related Work

Self-supervised Motion Estimation. Deep learning has significantly advanced motion estimation. Early dense optical flow methods (Dosovitskiy et al., 2015; Sun et al., 2018; Teed & Deng, 2020) showed strong performance but often struggle with long-range tracking and occlusions. Inspired by Sand and Teller (Sand & Teller, 2008), recent methods instead track individual points over time (Harley et al., 2022; Doersch et al., 2022), with newer architectures (Doersch et al., 2023; Karaev et al., 2024c;a; Neoral et al., 2024; Zheng et al., 2023; Doersch et al., 2024; Zholus et al., 2025) improving long-term accuracy. However, these models often rely on synthetic data, limiting their real-world generalization. To bridge this gap, self-supervised optical flow methods (Jonschkowski et al., 2020; Liu et al., 2019; Huang et al., 2023) have been proposed, but they inherit many limitations of supervised approaches. Other work focuses directly on long-range tracking: Vondrick et al. (Vondrick et al., 2018) train a model to propagate color in grayscale videos, implicitly learning motion. Cycle consistency has also been leveraged (Jabri et al., 2020; Wang et al., 2019), including for point tracking (Shrivastava & Owens, 2024). Models trained for semantic understanding, such as

DINOv2 (Oquab et al., 2023), have also been adapted for semantic and temporal correspondence. DIFT (Tang et al., 2023), based on image diffusion models, extracts features suitable for matching, while SD-DINO (Zhang et al., 2023a) combines Stable Diffusion and DINO features to solve a range of semantic and geometric tasks. Some work has been done on counterfactual reasoning for understanding visual scenes. Visual Jenga (Bhattad et al., 2025) progressively removes objects from a single image until only the background remains, revealing geometric relationships among scene elements. Recent research on counterfactual world modeling (Bear et al., 2023; Venkatesh et al., 2023) tackles keypoint prediction and optical flow by training a masked autoencoder for future-frame prediction, then perturbing inputs to estimate motion. In contrast, we exploit properties of diffusion, such as the ability to subtly manipulate videos, to obtain our predictions from an off-the-shelf model; we base our approach on generative video models rather than masked future frame prediction; and we address the long-range point tracking problem rather than optical flow. Stojanov et al. (Stojanov et al., 2025) extended the counterfactual world modeling to point tracking by learning RGB perturbations that can be propagated through a frozen next-frame predictor, optimizing them with a jointly trained sparse optical-flow module. By contrast, our approach relies entirely on prompting a frozen video diffusion model and requires no additional training. A recent concurrent work (Nam et al., 2025) extracts features from a pretrained video model for tracking, using a one-to-one frame-to-latent mapping to avoid temporal compression, but involves a complex, architecture-dependent analysis to identify which layers provide the best features and does not handle occlusion. In comparison, our method does not rely on feature extraction, remains architecture-agnostic and is inherently robust to occlusion.

Pretrained Models. Large pretrained models have become foundational in computer vision, replacing task-specific architectures across classification, detection, and segmentation (Donahue et al., 2014; Chen et al., 2020; He et al., 2020; Zhang et al., 2016; Oquab et al., 2023; Radford et al., 2021; Zhai et al., 2023; Kirillov et al., 2023; Yang et al., 2024a; Liu et al., 2024; Tong et al., 2024; Li et al., 2023). Diffusion models for image generation (Podell et al., 2023; Rombach et al., 2022; Dhariwal & Nichol, 2021; Nichol et al., 2021) introduced generative features that capture semantic correspondences (Tang et al., 2023; Luo et al., 2023; Zhang et al., 2023a), but lack temporal reasoning needed for motion-centric tasks. Video diffusion models (Blattmann et al., 2023a;b; Yu et al., 2023; Wang et al., 2025; Yang et al., 2024b; Polyak et al., 2024; Chefer et al., 2025) address temporal consistency, though many still prioritize appearance over motion. Chefer et al. (Chefer et al., 2025) address this by incorporating optical flow during training. We work in the opposite direction, using generative models to aid motion estimation.

Visual Prompting. Prompting strategies have achieved notable success in natural language processing (Wei et al., 2022; Kojima et al., 2022), motivating analogous techniques in computer vision. One prominent direction frames downstream vision tasks as inpainting problems, using pretrained models to complete images conditioned on visual cues (Bar et al., 2022; Wang et al., 2023; Bai et al., 2024). Another line of work focuses on optimizing prompt representations, showing that both textual and visual prompts can be refined via gradient-based methods to better adapt vision models (Zhou et al., 2022; Bahng et al., 2022). Recent studies also demonstrate that simple visual prompts, such as colored shapes, can elicit useful behaviors from vision-language models (Shtedritski et al., 2023; Yao et al., 2024). We introduce a simple yet effective visual prompt: placing a colored dot at the pixel to be tracked. To our knowledge, this is the first use of image prompting for point tracking in video diffusion models.

Controllable Generation. Controllable generation is a key goal in generative modeling (Hao et al., 2018; Zhuang et al., 2021; Liu et al., 2021; Jo & Park, 2019; Chen et al., 2024; Zhang et al., 2023b; Ruiz et al., 2023; Chen et al., 2023). SDEdit (Meng et al., 2021) introduced a training-free method for guided synthesis using noise perturbation and iterative denoising. More recent work enables fine-grained spatial control in diffusion models (Chen et al., 2024; Lugmayr et al., 2022; Si et al., 2024; Wu et al., 2024; Chefer et al., 2023). RePaint (Lugmayr et al., 2022), for example, inpaints masked regions without affecting the rest of the image. Methods like ControlNet (Zhang et al., 2023b) and DreamBooth (Ruiz et al., 2023) enable control via fine-tuning. These ideas have been extended to video (Zhang et al., 2023c; Feng et al., 2024), providing structured editing through architectural design and hierarchical sampling.

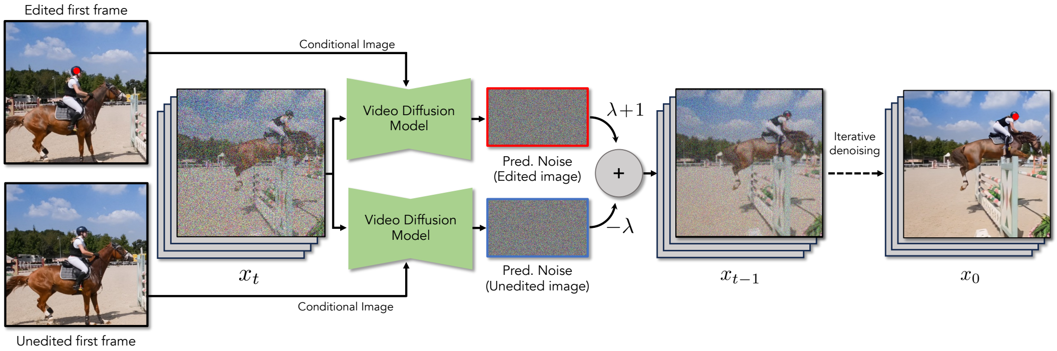


Figure 2: **Enhancing the Counterfactual Signal.** We use negative prompting to ensure that the generated video contains the marker. In each denoising step (Eq. 5), we condition the denoising on two images: (1) *Edited First Frame*: the first frame of the video with a marking added, and (2) *Unedited First Frame*: the original first frame of the video. We then subtract the weighted noise vector of the latter from the former.

3 Method

Our goal is to repurpose a pretrained generative video model to track points in a video. To do this, we exploit several key properties of diffusion models. We review diffusion models, then describe how they can be adapted for point tracking.

3.1 Preliminaries: Video Diffusion Models

Latent video diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Rombach et al., 2022; Blattmann et al., 2023a; Wang et al., 2025) generate a sequence of F RGB frames, $\mathbf{V} \in \mathbb{R}^{F\times H\times W\times 3}$. These models operate on a compact latent representation $\mathbf{x} \in \mathbb{R}^{F'\times H'\times W'\times C}$, where C is the channel dimension, which can be converted into a video via a decoder.

Forward (Noising) Process.[1] Given a clean video latent $\mathbf{x}_0$, we define the noising process using a variance schedule β_t over timesteps $t \in \{1,\dots,T\}$. The corrupted latent is constructed via:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I}), \tag{1}$$

where $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$.

Reverse (Denoising) Process. At each timestep t, the video diffusion model, $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)$, predicts the noise component. These models may be conditioned on additional data c, such as a text prompt or the desired first frame of the video. We denoise the corrupted latent (Sohl-Dickstein et al., 2015; Ho et al., 2020):

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)\right) + \sigma_t\mathbf{z} \tag{2}$$

where σ_t^2 is the variance, and $\mathbf{z} \sim \mathcal{N}(0, I)$.

Video Manipulation. Trained diffusion models can also be used to manipulate existing videos, without additional training. We discuss two such applications: regeneration and inpainting.

Rather than generating a latent vector from scratch, one can regenerate an existing, clean video with modifications using SDEdit (Meng et al., 2021). We add an intermediate level of noise, $1 < t < T$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{3}$$

and then run the reverse diffusion process to denoise it. This results in a video that resembles the coarse structure of the original, but with different fine-grained details (e.g., restyling a real video into a cartoon using a text prompt).

We can also use pretrained video diffusion models for inpainting (Lugmayr et al., 2022). Given a binary spatiotemporal mask $\mathbf{m} \in \mathbb{B}^{F\times H\times W}$ indicating which patches of the input video can (and

[1] Our method is agnostic to the specific diffusion model and therefore follows the widely used standard notation of denoising diffusion models (Ho et al., 2020) with classifier-free guidance (Ho & Salimans, 2022).

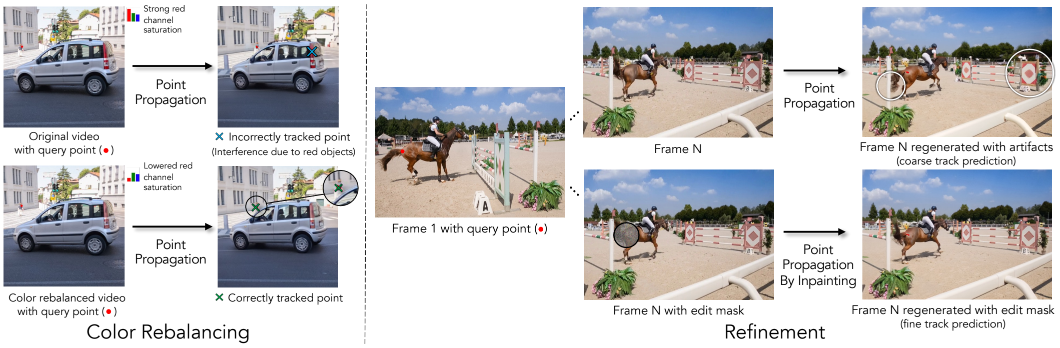

Strong red channel saturation
Point Propagation
Original video with query point (●)
× Incorrectly tracked point (Interference due to red objects)
Lowered red channel saturation
Point Propagation
Color rebalanced video with query point (●)
× Correctly tracked point
Color Rebalancing
Frame 1 with query point (●)
Frame N
Point Propagation
Frame N regenerated with artifacts (coarse track prediction)
Frame N with edit mask
Point Propagation By Inpainting
Frame N regenerated with edit mask (fine track prediction)
Refinement

Figure 3: **Tracking Enhancements.** To improve point tracking in video, we introduce two enhancements: (1) *Color Rebalancing:* remove existing red hues to ensure the red marker remains a unique tracking cue; (2) *Refinement:* obtain initial trajectories with a color-based tracker, then refine them using an inpainting mask to correct temporal artifacts such as object shifts. This two-step procedure first produces coarse tracks and then refines them via mask-constrained reverse diffusion.

cannot) be changed, we run the reverse diffusion process and constrain updates to the masked region. At each denoising step, we constrain the updates such that they occur only in the masked region. In each step of the reverse diffusion process, we compute (Lugmayr et al., 2022):

$$\begin{aligned} \tilde{\mathbf{x}}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\right) + \sigma_t \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x}_{t-1}^{\text{original}} &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \mathbf{x}_{t-1} &= \mathbf{m} \odot \tilde{\mathbf{x}}_{t-1} + (1-\mathbf{m}) \odot \mathbf{x}_{t-1}^{\text{original}}, \end{aligned} \tag{4}$$

where $\boldsymbol{\epsilon}_\theta$ is the estimated noise for the iteration t, and as before $\mathbf{x}_0$ is the latent for the input video.

3.2 Point Prompting for Counterfactual Tracking

We now describe an approach to counterfactual modeling that enables a video diffusion model to perform "zero shot" tracking.

Marking a Point's Trajectory. Given an input video and the pixel location of a query point, our goal is to predict the positions of the point in the subsequent frames. As shown in Fig. 1, we prompt an off-the-shelf video diffusion model to draw a distinctive marker in each frame at the point's position. We then localize the point position using simple low-level image processing.

We insert a distinctive marking on the query point's position in the initial frame. For this, we simply use a circular dot, which can plausibly be interpreted as being part of the object's surface. For simplicity, we color this dot pure red in all of our experiments. We then apply SDEdit (Sec. 3.2) using an intermediate timestep $1 < t < T$ to the video to manipulate the video, while conditioning on the edited initial frame. This propagates the marker to the subsequent frames of the video.

Enhancing the Counterfactual Signal. One of the challenges of applying counterfactual modeling to powerful generative models is that their strong priors lead them to ignore the manipulations that we introduce. For example, when the marker does not naturally fit into a scene, it will often disappear from the generated video within a few frames. We address this problem by using a simple negative prompt that reduces the probability of drawing samples that resemble the original video. We compute the difference between two noise estimates (Fig. 2) that are computed using different types of first-frame conditioning: one where we condition on the original image (i.e., without the marker) and another where we condition on the edited image (i.e., with the marker):

$$\tilde{\boldsymbol{\epsilon}}_\theta\left(\mathbf{x}_t, \mathbf{c}_I\right) = (\lambda + 1) \cdot \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \phi(\mathbf{c}_I)) - \lambda \cdot \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, \mathbf{c}_I\right), \tag{5}$$

where $\tilde{\boldsymbol{\epsilon}}$ is the noise estimate after enhancement, $\mathbf{c}_I$ is the initial-frame conditioning, $\phi(\mathbf{c}_I)$ is the initial frame after applying the counterfactual manipulation (i.e., adding the marker), and $\lambda > 0$ is a weight. Due to the well-known close connection between denoising and score functions, the modified denoiser $\tilde{\boldsymbol{\epsilon}}$ corresponds to the following score function (Ho & Salimans, 2022; Karras et al., 2024):

$$\nabla_{\mathbf{x}_t} \log(p_\lambda(\mathbf{x}_t)) = \nabla_{\mathbf{x}_t} \log\left(p(\mathbf{x}_t \mid \phi(\mathbf{c}_I))\left[\frac{p(\phi(\mathbf{c}_I) \mid \mathbf{x}_t)}{p(\mathbf{c}_I \mid \mathbf{x}_t)}\right]^\lambda\right), \tag{6}$$

where $p(\mathbf{x}_t)$ is the probability under the model for the noisy input at time t, and where we have used the well-known fact that $\boldsymbol{\epsilon}(\mathbf{x}_t) \propto -\nabla_{\mathbf{x}_t} \log(p(\mathbf{x}_t))$ and Bayes rule, following the standard formulation of classifier-free guidance (Ho & Salimans, 2022). From this perspective, we see that our sampling procedure generates videos conditioned on the manipulated initial frame, while biasing the score direction away from samples from the unedited conditioning.

We note that this strategy is related to (but distinct from) the approach used in previous work on counterfactual world models (Bear et al., 2023; Stojanov et al., 2025). They generate two possible futures using a masked autoencoder model: one with the marker and one without. They then enhance the signal by directly subtracting the two generated images, which amounts to approximately estimating: $\mathbb{E}_{p(\mathbf{x}|\phi(\mathbf{c}_I))}[\mathbf{x}] - \mathbb{E}_{p(\mathbf{x}|\mathbf{c}_I)}[\mathbf{x}]$. Like our approach, this method enhances their ability to detect the effect of the counterfactual by comparing the generated result to an unedited baseline, but instead of comparing the predicted samples themselves, we include this constraint as guidance in the sampler. In our experiments, we found that objects often subtly change position in different samples of a video diffusion model, leading to this differences between generations to contain significant artifacts, making it challenging to use this approach.

Tracking the Marker. To extract a track from generated videos containing an inserted marker at a query point, we implement a simple tracker that locates the marker in each frame based on color. Given the marker's initial location (u_0, v_0) in the first frame, we track its motion frame by frame. For each subsequent frame k, the tracker searches for red pixels (in HSV colorspace) within a local window of radius r centered at the previous location (u_{k-1}, v_{k-1}), selecting the pixel closest to the previous position. Since the marker appears as a small blob, we refine the estimate by averaging the positions of nearby red pixels to obtain a more stable center, which serves as the predicted track point.

If no red pixels are found within the search region, we treat the marker as occluded and propagate the last known position forward. We expand the search radius r at each step until the marker reappears, after which we reset r to its original value. This adaptive strategy makes the tracker robust to temporary occlusions and large displacements, enabling it to recover from tracking uncertainty.

3.3 Extensions

We can further improve the prediction by coarse-to-fine refinement and by rebalancing the colors in the video to exclude the marker's color (Fig. 3).

Coarse-to-Fine Refinement. Accurate tracking requires that the generated video remain pixel-aligned with the original. However, the generated video may be subtly misaligned with the original video after regeneration, leading to tracking errors. Inspired by coarse-to-fine motion estimation, we improve our tracking predictions after their initial estimates, by exploiting the fact that video diffusion models can be repurposed to perform inpainting. We restrict the model's ability to modify the video during generation, allowing it to generate only regions near the potential tracked point, while preserving the rest of the video content.

After obtaining the initial estimate of marker positions (as described above), we construct a binary spatiotemporal binary mask $\mathbf{m} \in \mathbb{R}^{F \times H \times W}$, where each frame's mask is set to 1 within a small radius r centered on the tracked location, i.e., $\mathbf{m}[u, v]$ is set to 1 if $(u, v) \in B_r(u_k, v_k)$. We then re-run the video generation, while allowing only the image regions indicated by $\mathbf{m}$ to change. and plug it in Eq. 4.

Color Rebalancing. Since our tracker relies on detecting a particular color, we rebalance the video's colors such that the marker's color does not appear within it. We do this by reducing the saturation of the marker's color. For example, when tracking a red marker, we reduce the saturation of red regions, effectively suppressing natural red hues while preserving overall image quality (details provided in Appendix D.1). We find that this reduces mistakes during occlusion, since the marker is not present and thus false detections are more common.

4 Experiments

We evaluate our prompting strategy's ability to accurately track points through a video, using the TAP-Vid benchmarks (Doersch et al., 2022).

4.1 Video Models

We consider recent image-conditioned video diffusion models:

Wan2.1 (Wang et al., 2025) combines a 3D causal VAE with a diffusion transformer (DiT) conditioned on text and an input image and trained using flow-matching (Lipman et al., 2022). The VAE encodes video into latents $x \in \mathbb{R}^{(1+F/4)\times H/8\times W/8}$, keeping the first frame at full temporal resolution and downsampling the rest by $4\times$. Outputs are 480×832. We test 1.3B- and 14B-parameter variants, reporting results with the 14B model unless noted.

Wan2.2 (Wang et al., 2025) extends Wan2.1 with a Mixture-of-Experts (MoE) architecture. By distributing denoising across timesteps among specialized experts, it increases model capacity without extra computation and is trained on a much larger dataset.

CogVideoX (Yang et al., 2024b) is another I2V diffusion model that also combines a 3D causal VAE with a diffusion transformer. It generates 768×1360 videos from a text prompt and reference image. The VAE compresssion is the same as Wan, while the transformer conditions on the image and T5 text embeddings (Raffel et al., 2020).

For all models we use 50 denoising steps with noise strength 0.5 and an empty text prompt. Experiments run on A40 or L40S GPUs (one GPU per video). Generating a 50-frame video for a single query point takes about 7 min for Wan2.1-1.3B, 30 min for Wan2.1-14B, and 20 min for CogVideoX. These runtimes are acceptable given our focus on evaluating the tracking capabilities of video diffusion models, and our method could be distilled into a more efficient model, similar to Opt-CWM (Stojanov et al., 2025).

4.2 TAP-Vid Benchmark

We evaluate on two TAP-Vid benchmark splits: DAVIS (30 videos, 34–104 frames) and Kinetics (30 sampled videos, 250 frames, following (Stojanov et al., 2025)) for efficiency. These natural videos match the training distribution of our video diffusion models (rather than computer generated video). Using the first sampling strategy, we pick one query point per video, overlay a red dot at its position in the first frame, and run our model to propagate the point throughout the video. The resulting trajectory is then extracted using our tracker.

Evaluation Metrics. We report: (1) *Positional Accuracy* (δ^x_{avg}), fraction of visible points within distance thresholds; (2) *Occlusion Accuracy* (OA), visibility prediction accuracy; and (3) *Average Jaccard* (AJ), average overlap between predicted and ground-truth visible points across thresholds (Doersch et al., 2022).

5 Results

Unless otherwise noted, we use Wan2.1-14B (Wang et al., 2025) as the video diffusion model for all experiments.

Method	Supervision	TAP-Vid DAVIS			TAP-Vid Kinetics		
		AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑
RAFT (Teed & Deng, 2020)	Supervised	34.48	53.55	74.90	30.15	46.44	75.44
TAP-Net (Doersch et al., 2022)		32.05	48.42	77.35	34.59	48.42	80.88
TAPIR (Doersch et al., 2023)		58.47	70.56	87.27	47.46	59.56	85.76
CoTracker3 (Karaev et al., 2024b)		64.45	77.13	90.90	**54.35**	**65.99**	**89.43**
TAPNext (Zholus et al., 2025)		**66.56**	**79.48**	**92.21**	52.97	64.46	89.30
GMRW (Shrivastava & Owens, 2024)	Self-Sup.	36.47	54.59	76.36	25.70	41.63	71.33
Opt-CWM (Stojanov et al., 2025)		**47.53**	**64.83**	**80.87**	**44.85**	**57.74**	**84.12**
DINOv2+NN (Oquab et al., 2023)	Zero-Shot	15.19	31.19	61.81	12.69	24.22	62.45
DIFT (Tang et al., 2023)		21.51	39.55	69.71	15.10	25.56	63.17
SD-DINO (Zhang et al., 2023a)		29.68	50.45	69.71	16.47	28.37	62.79
Ours		**42.21**	**57.29**	**82.90**	**27.36**	**41.51**	**71.39**

Table 1: **TAP-Vid Benchmark Results.** We report results on the TAP-Vid First benchmark. Our zero-shot method outperforms all other zero-shot baselines and is competitive with self-supervised and supervised trackers. On TAP-Vid DAVIS-First, it matches self-supervised methods in AJ and exceeds them in occlusion accuracy, highlighting strong object permanence from generative modeling.

Quantitative Results. Table 1 compares our method against several baselines using Wan2.1. Among zero-shot methods, ours achieves the highest performance. On TAP-Vid DAVIS, we reach an AJ score of 42.21, outperforming all other zero-shot baselines and even surpassing GMRW (Shrivastava

& Owens, 2024), a strong self-supervised approach. Our occlusion accuracy also exceeds that of both zero-shot and self-supervised methods, approaching supervised performance, highlighting the ability of diffusion models to reason through occlusions.

We include top supervised methods such as CoTracker3 (Karaev et al., 2024b) and TAPNext (Zholus et al., 2025), as well as the best-performing self-supervised baseline, Opt-CWM (Stojanov et al., 2025). While conceptually related, Opt-CWM learns to propagate perturbations through a next-frame predictor supervised by sparse flow. In contrast, our method is entirely zero-shot, using a simple colored dot without training or learned perturbations.

Method	TAP-Vid DAVIS		
	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑
CogVideoX1.5-5B (Yang et al., 2024b)	24.15	34.38	70.79
Wan2.1-1.3B (Wang et al., 2025)	44.58	58.77	85.16
Wan2.1-14B (Wang et al., 2025)	48.60	63.47	85.75
Wan2.2-14B (Wang et al., 2025)	48.78	63.91	86.17

Table 2: **Video Model Ablations.** Wan2.1-1.3B and 14B (Wang et al., 2025) outperform CogVideoX (Yang et al., 2024b), showing that stronger video models improve tracking performance.

Image source	TAP-Vid DAVIS		
	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑
DAVIS (256×256)	42.21	57.29	82.90
DAVIS (256×256 up.)	45.48	60.16	83.49
DAVIS (original res.)	48.60	63.47	85.75

Table 3: **Image Resolution Ablations.** Comparing input resolutions for Wan2.1. Upscaling with (Zhou et al., 2024) improves tracking by better aligning with the model's training distribution.

Method	TAP-Vid DAVIS		
	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑
all	48.60	63.47	85.75
w/o refinement	42.70	59.26	85.14
w/o counterfactual enhancement	22.03	38.53	61.19
w/o color rebalancing	34.86	52.12	82.18
tracker only	11.26	21.07	77.74

Table 4: **Tracking Pipeline Ablations.** Quantitative results on TAP-Vid DAVIS-First showing the impact of each stage in our pipeline (Fig. 3). The last row uses original pixel color instead of the red dot for tracking.

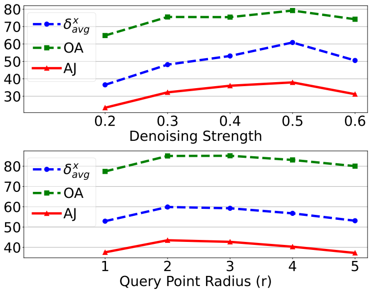


Figure 4: **Effect of denoising strength and radius on tracking performance.**

Different Video Models. Table 2 shows results using Wan2.1 (1.3B and 14B variants), Wan2.2, and CogVideoX (Yang et al., 2024b). Our method performs well across all four models, demonstrating compatibility across different video generation backbones. Wan2.1 and Wan2.2 deliver the strongest results, with the 14B variant outperforming the 1.3B model. We attribute this gain to their higher video generation quality indicating that improved generative fidelity directly enhances tracking accuracy.

Generation Resolution. The TAP-Vid benchmark provides videos at a resolution of 256×256, which we resize to 480×832 to match the input resolution of Wan2.1. To assess the impact of resolution, we first upsample inputs using Upsample-A-Video (Zhou et al., 2024), which improves tracking (Table 3). We then run Wan2.1 on the original high-res DAVIS frames (Perazzi et al., 2016), achieving an AJ score of 48.6, surpassing Opt-CWM. These results show that higher-resolution inputs significantly enhance tracking by improving video generation quality.

Point Propagation Ablations. Table 4 shows ablations of key components. The first row shows our full model with all components enabled. Removing the inpainting-based refinement step reduces positional accuracy due to spatial shifts during denoising which negatively affects tracking precision. Removing counterfactual enhancement guidance causes failure in point propagation where tracking is lost after 5–6 frames, highlighting its critical role in maintaining point consistency across frames. Disabling color rebalancing also degrades performance. Since the tracker relies on detecting red pixels, failure to suppress red tones in the background introduces false positives, especially when the query point is occluded, making tracking less reliable.

We also evaluate a tracker-only baseline that tracks the query point's color from the initial frame without any point propagation. This performs significantly worse, highlighting that the primary performance gains in our method arise from accurate point propagation through video generation, rather than from the tracker itself, which is intentionally kept simple. Additionally, we ablate key

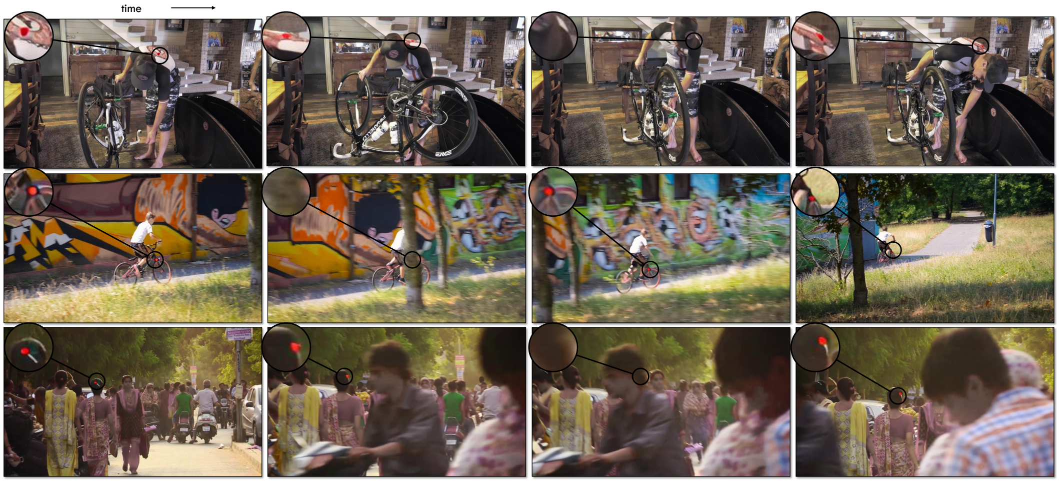


Figure 5: **Point Propagation.** Frames generated from the video diffusion model show consistent red dot tracking. The model recovers the point after long occlusions, showing temporal understanding and object permanence.

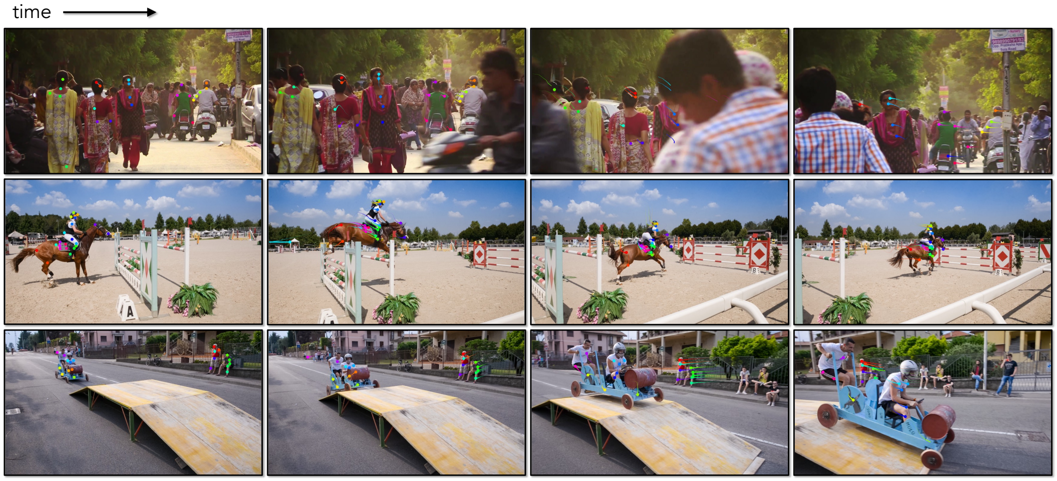


Figure 6: **Tracking results.** Frames show the query point being tracked (circled dot) and its trajectory over the previous 5 frames. When the query point is occluded, only the trajectory tail is displayed without the dot.

hyperparameters in Fig. 4. We observe that a noise strength of 0.5 and a query point radius of 2 pixels yield the best results.

Qualitative Results. In Fig. 5, we show video generations from our method, where red dots are successfully propagated across frames, including through occlusions. We extract these points and display the resulting tracks for multiple query points in Fig. 6. Our method reliably tracks points over long temporal range and maintains accuracy even in the presence of occlusions.

6 Conclusion

We have shown that a video diffusion model, when carefully prompted, can mark the location of a point as it moves through a scene over time. We use this idea to create a simple point tracker, which obtains surprisingly effective tracking results, outperforming previous zero-shot approaches. We see our work as opening two new directions. The first is expanding the number of ways that one can adapt large pretrained video diffusion models to new tasks, such as through prompting schemes that go beyond the use of language. Second, our work shows that video generative models are a useful source of pretraining for tracking. We therefore see our work as a step toward unifying video generation and tracking. We will release our code upon acceptance.

References

Kelsey Allen, Carl Doersch, Guangyao Zhou, Mohammed Suhail, Danny Driess, Ignacio Rocco, Yulia Rubanova, Thomas Kipf, Mehdi SM Sajjadi, Kevin Murphy, et al. Direct motion models for assessing generated videos. *arXiv preprint arXiv:2505.00209*, 2025. 1

Pierfrancesco Ardino, Marco De Nadai, Bruno Lepri, Elisa Ricci, and Stéphane Lathuilière. Click to move: Controlling video generation with sparse motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 3

Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22861–22872, 2024. 3

Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022. 3

Daniel M Bear, Kevin Feigelis, Honglin Chen, Wanhee Lee, Rahul Venkatesh, Klemen Kotar, Alex Durango, and Daniel LK Yamins. Unifying (machine) vision via counterfactual world modeling. *arXiv preprint arXiv:2306.01828*, 2023. 1, 3, 6

Anand Bhattad, Konpat Preechakul, and Alexei A. Efros. Visual jenga: Discovering object dependencies via counterfactual inpainting, 2025. URL https://arxiv.org/abs/2503.21770. 3

A. Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, and Dominik Lorenz. Stable video diffusion: Scaling latent video diffusion models to large datasets. *ArXiv*, 2023a. 3, 4

A. 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. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b. 3

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. *arXiv preprint arXiv:2501.08331*, 2025. 1

Duygu Ceylan, Chun-Hao Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion. 2023. 1

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM transactions on Graphics (TOG)*, 42(4):1–10, 2023. 3

Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. Videojam: Joint appearance-motion representations for enhanced motion generation in video models. *ArXiv*, 2025. 1, 3

Siyi Chen, Huijie Zhang, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. Exploring low-dimensional subspaces in diffusion models for controllable image editing. *arXiv preprint arXiv:2409.02374*, 2024. 3

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020. 3

Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video diffusion models with motion prior and reward feedback learning. *arXiv preprint arXiv:2305.13840*, 2023. 3

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

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, 6, 7, 18

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*, pp. 10061–10072, 2023. 2, 7, 18

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Proceedings of the Asian Conference on Computer Vision*, pp. 3257–3274, 2024. 2

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014. 3

Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015. 2

Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. Ccedit: Creative and controllable video editing via diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6712–6722, 2024. 3

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Chen Sun, Oliver Wang, Tobias Pfaff, Tatiana Lopez-Guevara, Carl Doersch, Yusuf Aytar, Michael Rubinstein, Andrew Owens, and Deqing Sun. Motion prompting: Controlling video generation with motion trajectories. *Computer Vision and Pattern Recognition (CVPR)*, 2025. 1

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023. 1

Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7854–7863, 2018. 1, 3

Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pp. 59–75. Springer, 2022. 2

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*, pp. 9729–9738, 2020. 3

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4, 5, 6

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 4

Hsin-Ping Huang, Charles Herrmann, Junhwa Hur, Erika Lu, Kyle Sargent, Austin Stone, Ming-Hsuan Yang, and Deqing Sun. Self-supervised autoflow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11412–11421, 2023. 2

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *arXiv preprint arXiv:2411.13503*, 2024. 19

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. 2

Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1745–1753, 2019. 3

Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 2020. 2

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024a. 2

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proc. arXiv:2410.11831*, 2024b. 7, 8, 18

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pp. 18–35. Springer, 2024c. 2

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024. 5

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023. 3

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022. 3

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *European Conference on Computer Vision*, 2018. 1

Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3041–3050, 2023. 3

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2022. URL https://api.semanticscholar.org/CorpusID:252734897. 7

Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, Jing Liao, Bin Jiang, and Wei Liu. Deflocnet: Deep image editing via flexible low-level controls. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10765–10774, 2021. 3

Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4571–4580, 2019. 2

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024. 3

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022. 3, 4, 5

Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *Advances in Neural Information Processing Systems*, 36:47500–47510, 2023. 3

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 1, 2, 3, 4

Jisu Nam, Soowon Son, Dahyun Chung, Jiyoung Kim, Siyoon Jin, Junhwa Hur, and Seungryong Kim. Emergent temporal correspondences from video diffusion transformers. *arXiv preprint arXiv:2506.17220*, 2025. 3

Michal Neoral, Jonáš Šerỳch, and Jiří Matas. Mft: Long-term tracking of every pixel. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6837–6847, 2024. 2

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. 3

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. 3, 7, 18

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016. 8

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 3

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Ki ran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam S. Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali K. Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models. *ArXiv*, abs/2410.13720, 2024. URL https://api.semanticscholar.org/CorpusID:273403698. 3

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021. 3

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 7

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*, pp. 10684–10695, 2022. 3, 4

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*, pp. 22500–22510, 2023. 3

Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International journal of computer vision*, 80:72–91, 2008. 2

Ayush Shrivastava and Andrew Owens. Self-supervised any-point tracking by contrastive random walks. 2024. URL `https://arxiv.org/abs/2409.16288`. 2, 7, 18

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11987–11997, 2023. 3

Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4733–4743, 2024. 3

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015. 4

Stefan Stojanov, David Wendt, Seungwoo Kim, Rahul Venkatesh, Kevin Feigelis, Jiajun Wu, and Daniel LK Yamins. Self-supervised learning of motion concepts by optimizing counterfactuals. *arXiv preprint arXiv:2503.19953*, 2025. 3, 6, 7, 8, 18

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8934–8943, 2018. 2

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. 2, 3, 7, 18

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*, pp. 402–419. Springer, 2020. 2, 7, 18

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024. 3

Rahul Venkatesh, Honglin Chen, Kevin Feigelis, Daniel M Bear, Khaled Jedoui, Klemen Kotar, Felix Binder, Wanhee Lee, Sherry Liu, Kevin A Smith, et al. Understanding physical dynamics with counterfactual world modeling. *arXiv preprint arXiv:2312.06721*, 2023. 1, 3

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)*, September 2018. 2

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 3, 4, 7, 8, 19

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*, 2019. 2

Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6830–6839, 2023. 3

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 3

Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. In *European Conference on Computer Vision*, pp. 378–394. Springer, 2024. 3

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10371–10381, 2024a. 3

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. *ArXiv*, abs/2408.06072, 2024b. URL https://api.semanticscholar.org/CorpusID:271855655. 3, 7, 8, 19

Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *AI Open*, 5:30–38, 2024. 3

Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023. 3

Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems*, 36:45533–45547, 2023a. 2, 3, 7, 18

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023b. 3

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pp. 649–666. Springer, 2016. 3

Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023c. 3

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*, pp. 19855–19865, 2023. 2

Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi S. M. Sajjadi, Sarath Chandar, and Ross Goroshin. Tapnext: Tracking any point (tap) as next token prediction. *arXiv preprint arXiv:2504.05579*, 2025. 2, 7, 8, 18

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3

Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2535–2545, 2024. 8, 19

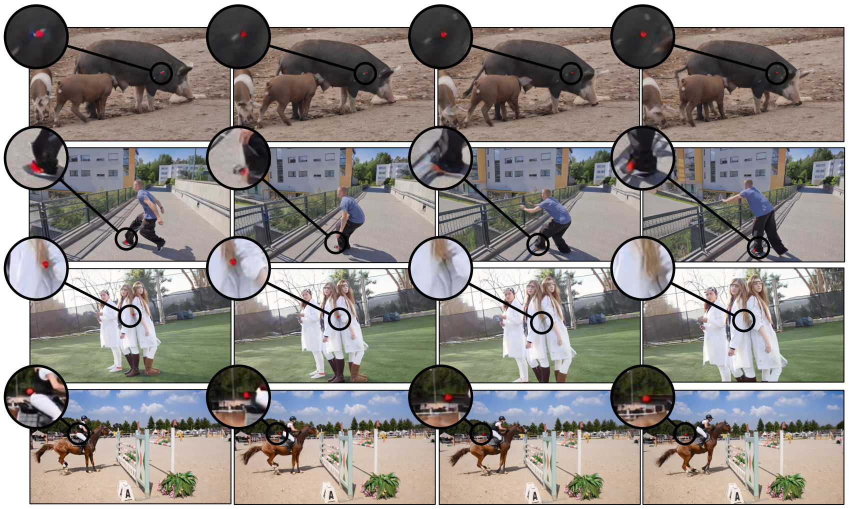
Peiye Zhuang, Oluwasanmi Koyejo, and Alexander G Schwing. Enjoy your editing: Controllable gans for image editing via latent space navigation. *arXiv preprint arXiv:2102.01187*, 2021. 3

A Qualitative Results

We include videos showing point propagation and tracking results at https://iclr-2026-demo.github.io/project-demo .

B Limitations

Our approach requires generating a video for each tracked point. Since our goal is to show that video generators can perform tracking, rather than to perform tracking as an end in itself, we did not attempt to optimize our approach. However, it can potentially be addressed by distilling our model's predictions into a network that directly performs tracking, by considering more efficient generation methods (e.g., one-step sampling), or by tracking multiple points at once. The video generators also sometimes fail to interpret the red dot as being attached to the object surface, especially for (likely out-of-distribution) computer-generated videos (Fig. 7).



Stationary Point: Point remains stationary with respect to the image boundaries (akin to speck of dirt on the lens)

Symmetry: Point drawn on the right foot gets propagated to the left foot

Propagation Failure: Model fails to propagate the point in the last two frames

Ambiguity Near Edges: Point near the boundary snaps to background

Figure 7: **Generation Failures.** Typical failure cases in video generation: (1) *Stationary Point:* The red dot remains fixed relative to image boundaries, resembling lens dirt. (2) *Symmetry Confusion:* Symmetrical objects (e.g., left and right body parts) cause point propagation errors, likely due to compressed latent representations. (3) *Propagation Failure:* The red dot vanishes across consecutive frames. (4) *Edge Ambiguity:* The red dot, near boundaries, shifts to the background.

C Quantitative Results on TAP-Vid

Table 5 presents results on TAP-Vid Kubric (using a subset of 30 videos) with our method based on the Wan2.1-14B model. Our approach outperforms zero-shot baselines, consistent with the results reported in Table 1 of the main paper.

However, the overall performance on Kubric is comparatively lower, likely due to the dataset's synthetic nature. The scenes are generated using a graphics simulator and typically consist of simple environments with basic textures and objects exhibiting non-natural, erratic motion, as illustrated in Fig. 8. These characteristics introduce challenges for faithful video re-generation, which in turn impacts the accuracy of point propagation and tracking.

C.1 Ablations

Tracker Ablations. We ablate key components of our tracking pipeline. First, we run the tracker without any enhancements on the generated videos. Adding a local search window around the previously detected point provides a small improvement, especially under occlusion. Gradually expanding the search radius when the query point becomes occluded yields further gains. We then introduce a position–refinement step that averages the coordinates of all red pixels within a fixed neighborhood around the predicted point, achieving the best overall performance. Finally, replacing

Method	Supervision	TAP-Vid Kubric		
		AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑
RAFT (Teed & Deng, 2020)	Supervised	68.50	83.01	89.94
TAP-Net (Doersch et al., 2022)		68.22	79.87	93.35
TAPIR (Doersch et al., 2023)		87.88	93.99	96.09
CoTracker3 (Karaev et al., 2024b)		76.99	92.35	92.35
TAPNext (Zholus et al., 2025)		80.91	87.03	97.16
GMRW (Shrivastava & Owens, 2024)	Self-Sup.	55.04	72.22	84.67
Opt-CWM (Stojanov et al., 2025)		60.11	77.24	85.62
DINOv2+NN (Oquab et al., 2023)	Zero-Shot	20.10	40.25	53.27
DIFT (Tang et al., 2023)		25.93	40.12	74.08
SD-DINO (Zhang et al., 2023a)		28.89	47.11	47.10
Ours		31.51	38.42	53.23
Ours (upsampled)		33.55	40.02	54.80

Table 5: **TAP-Vid Kubric Results.** We show results on TAP-Vid Kubric with *first* sampling strategy.

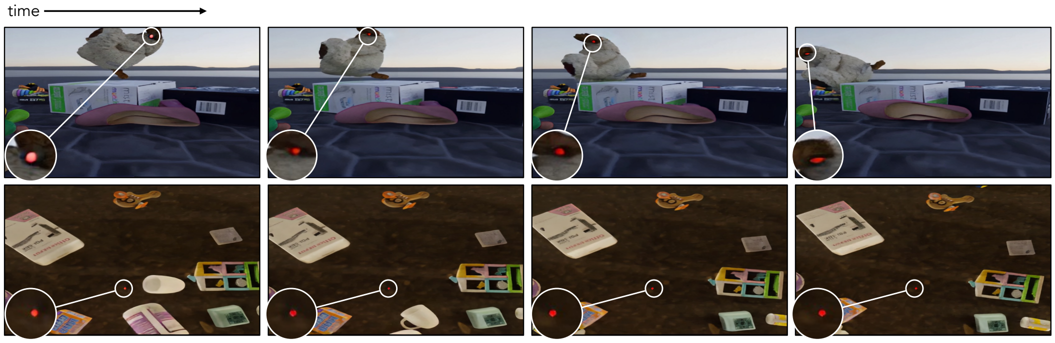


Figure 8: **Qualitative Results on TAP-Vid Kubric.** The top row shows a successful example of point propagation. In contrast, the bottom row illustrates a failure case where the point is not propagated due to the surface having very low texture.

the HSV color space with LAB causes a slight drop in accuracy, indicating that HSV is better suited for red-dot detection in our setup. Results are shown in Table 6.

Color space	Local search window	Occlusion based search radius	Average over color pixels	TAP-Vid DAVIS		
				AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑
HSV				35.80	53.15	81.79
HSV	✓			38.92	53.55	84.92
HSV	✓	✓		39.08	54.57	85.07
HSV	✓	✓	✓	42.70	59.26	85.14
LAB	✓	✓	✓	42.30	57.81	84.84

Table 6: **Tracker Ablations.** (Sec. 3.2). We assess local search window, adaptive radius for occlusions, averaging red pixel positions, and performance across HSV vs. LAB color spaces.

Additional ablations. We further assess model hyperparameters on a subset of TAP-Vid DAVIS videos (Table 7). We ablate the parameter λ (Eq. 5, main paper), which weights the noise estimate from the edited image. The best performance occurs at $\lambda = 8$. Table 8 reports results when varying the marker color. While our approach is robust to different marker colors, using red provides a slight performance gain.

Method	TAP-Vid DAVIS		
	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑
$\lambda = 4$	34.60	52.48	77.94
$\lambda = 8$	35.54	52.98	78.80
$\lambda = 11$	32.82	52.08	75.66
$\lambda = 14$	31.92	52.13	74.09

Table 7: **Counterfactual Enhancement Guidance.**We present ablation results for different values of λ, which controls the influence of the noise estimate from the edited image (with the colored dot) in counterfactual enhancement guidance.

Color	TAP-Vid DAVIS		
	AJ ↑	$< \delta^x_{\text{avg}}$ ↑	OA ↑
red	48.60	63.47	85.75
blue	46.51	60.80	84.08

Table 8: **Marker color.** We use different marker colors as prompt to show that our approach is invariant to marker color.

C.2 V-Bench Scores

Table 9 shows tracking performance alongside VBench (Huang et al., 2024) scores for Wan2.1 (1.3B and 14B variants), and CogVideoX (Yang et al., 2024b). VBench I2V benchmark evaluates the generation quality of image-conditioned video models. Tracking and generation quality both improve progressively from CogVideoX to Wan2.1-1.3B and further to Wan2.1-14B. We attribute this to the higher video generation quality—reflected in the superior VBench scores—which suggests that better generative models can directly boost tracking accuracy.

Method	TAP-Vid DAVIS	VBench
	AJ ↑	Total Score
CogVideoX1.5-5B (Yang et al., 2024b)	24.15	71.58
Wan2.1-1.3B (Wang et al., 2025)	44.58	83.26
Wan2.1-14B (Wang et al., 2025)	48.60	86.66

Table 9: **VBench (Huang et al., 2024) results.** We show VBench numbers for the different video models used.

D Implementation Details

D.1 Video Preprocessing

Color Rebalancing. Our tracker identifies red pixels in each frame as predicted points. To avoid false positives, we first remove red pixels from the original frame. We convert the frame to the HSV color space and detect pixels whose hue values fall within $[-30°, 10°]$, and whose saturation and value lie inside an ellipse with semi-major and semi-minor axes $r_1 = 80$, $r_2 = 30$, centered at $(255, 255)$. For detected red pixels, we clip the saturation to a maximum of 80, effectively desaturating them.

Padding Input Video. Both Wan and CogVideoX require that the input video contains $4T + 1$ frames. To satisfy this constraint, we pad the input by repeating the last frame until this condition is met. After re-generation, the added frames are removed to restore the original length.

Video Upscaling. We observe that using high-resolution videos improves point propagation, reducing generation artifacts and minimizing drift. To upscale the input videos, we use Upscale-A-Video (Zhou et al., 2024), a diffusion-based video upscaling method. Starting from 256×256 input resolution (from TAP-Vid), we upscale to 1024×1024 using Upscale-A-Video, then downscale to 480×832 to match the video model's expected resolution. For final tracking evaluation, we resize the output back to 256×256.

D.2 Point Propagation

As described in Sec. 3.1, we use SDEdit with a denoising strength $\gamma = 0.5$ to control the signal-to-noise ratio. The diffusion timestep t is calculated based on γ and the total number of diffusion steps T:

$$t = \lfloor \gamma \cdot T \rfloor \tag{7}$$

Counterfactual Enhancement Guidance To enhance the effect of the guidance from the edited image (with a colored dot), we use Eq. 5 (main paper) to compute the noise estimate. In our experiments, we follow the traditional classifier-free guidance scheme, where the guidance weight λ is set to 8.

D.3 Tracker

D.3.1 Red Pixel Detection

We detect red pixels using the `cv2.inRange` function in the HSV color space. OpenCV represents the hue channel in the range [0, 179], so we define two hue intervals to capture red, which wraps around the hue axis:

$$\text{low}_1 = (0, 150, 150), \quad \text{high}_1 = (5, 255, 255)$$
$$\text{low}_2 = (170, 150, 150), \quad \text{high}_2 = (180, 255, 255)$$

A pixel is considered red if it falls within either of these intervals.

D.3.2 Local Search and Occlusion Handling

To effectively locate the marker in each frame, we constrain our search for red pixels to a circular region of radius r centered at the previous detection. By default, this search radius is set to $r_{\text{default}} = 90$. If an occlusion is detected in the previous frame, we expand the search region to accommodate the increased positional uncertainty:

$$r = \min(r_{\text{default}} \times 1.1, r_{\text{max}}) \tag{8}$$

where $r_{\text{max}} = 150$. Once the marker is successfully detected again, we reset r to its default value to maintain efficiency and avoid spurious detections.

D.3.3 Center Estimation

After identifying candidate red pixels, we first select the one closest to the previous detection as an anchor. Around this anchor point, we examine a 20-pixel radius to gather nearby red pixel detections. The final predicted tracking point for the current frame is computed as the average position of these collected pixels. This averaging process produces a stable and consistent estimate for the red blob's center, leading to robust and accurate tracking across frames.

E LLM Usage Statement

We used large language models solely to refine wording and improve readability of the manuscript, for example, polishing sentences and enhancing clarity and flow. The models were not involved in research ideation, experimental design, or substantive content generation, and all scientific claims and results are our own.