Generative Point Tracking and Forecasting

Xuanchen Lu[1,2] Ang Cao[2] Chao Feng[1,2] Andrew Owens[1,2]
[1]Cornell University [2]University of Michigan

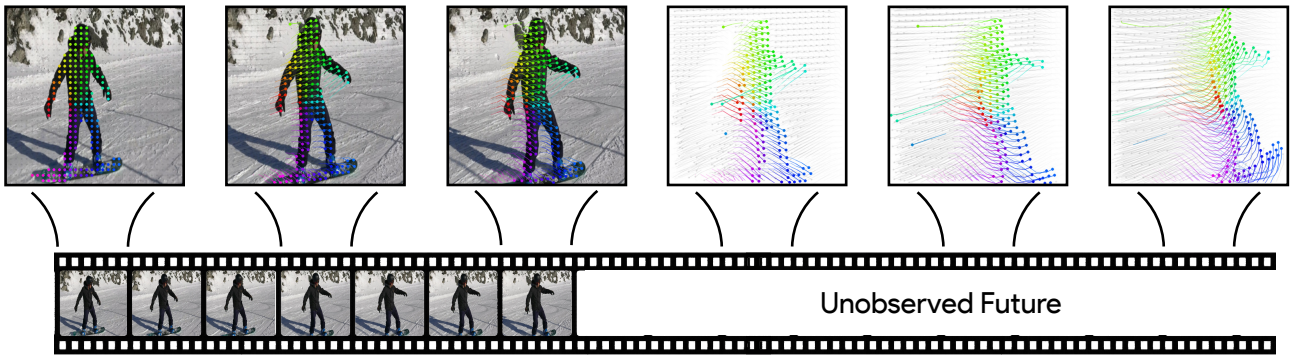


Figure 1. **Unifying point tracking and forecasting by generative modeling.** We train a model that can perform both point tracking and forecasting, by posing both problems as video-conditioned point generation. Here, we condition the model on an incomplete video sequence. When video frames are available, the model performs point tracking (left). For the unobserved frames at the end of the video, the model naturally performs trajectory forecasting (right). Background points are colored gray to highlight the motion of the main subject.

Abstract

Motion forecasting predicts where points will move in the future, while motion tracking predicts where they are in the present. Despite these similarities, existing approaches to the two problems are quite different. In this paper, we propose a unified model that can address both tasks. We train a causal, video-conditioned flow matching model to predict point positions. The resulting model can easily toggle between point tracking to forecasting by changing its visual signal. Despite our model's simplicity, we find that it outperforms prior work in point forecasting and obtains performance that is competitive with the state-of-the-art on the TAP-Vid benchmark.

1. Introduction

The problem of estimating how points move over time is at the core of both motion forecasting and tracking: forecasting predicts where they will be in the future, while tracking predicts where they are in the present. Despite these similarities, the two domains have historically been addressed using different architectures and learning methods. For example, recent point tracking methods are often based on regression with bespoke network architectures, iterative prediction, and robust loss functions [2, 19, 38, 39] and thus do not capture motion priors that are crucial for forecasting. This is in contrast to many other problems in computer vision, which have general-purpose models based on simple network architectures and training formulations.

In this paper, we propose a simple *point generation* model that addresses both point tracking and forecasting. We simply train a generative model to generate point trajectories conditioned on a sequence of video frames. The model performs tracking when visual context is provided and transitions to forecasting when input is absent. We show an example of this in Figure 1, where the model tracks the scene across several frames before forecasting its future motion.

To achieve this, we propose a video-conditioned flow matching model that is designed to perform both tracking and forecasting tasks, run online, model the interaction between query points, and achieve high positional accuracy. We perform generation in point space: in each video frame, we create a token for every query point and use a diffusion transformer [57] to estimate its position. To make the model causal and to enable efficient tracking, we autoregressively predict point positions for a temporal sliding window. Our model's predictions are conditioned on previous point pre-

dictions and (if available) pretrained visual features. When visual information is not provided, the model predicts position in its absence, which provides a natural mechanism for toggling between tracking and forecasting. To reduce error accumulation that occurs in long sequence prediction, we use diffusion forcing [12] and inject noise to previous position predictions at test time.

Despite our model's simplicity, we obtain point forecasting performance that outperforms all previous work on the benchmark of Boduljak et al. [9] and point tracking results that are competitive with state-of-the-art models on both the TAP-Vid DAVIS and Kinetics benchmark [18]. Through our experiments, we find:

- A single generative model can perform both forecasting and tracking.
- Generative modeling objectives can obtain competitive performance with highly tuned state-of-the-art regression-based approaches for point tracking.
- Simple, point-space generative models can outperform existing latent diffusion models on point forecasting [9].
- Jointly generating a trajectory's position with its occlusion indicators improves tracking performance.

2. Related Work

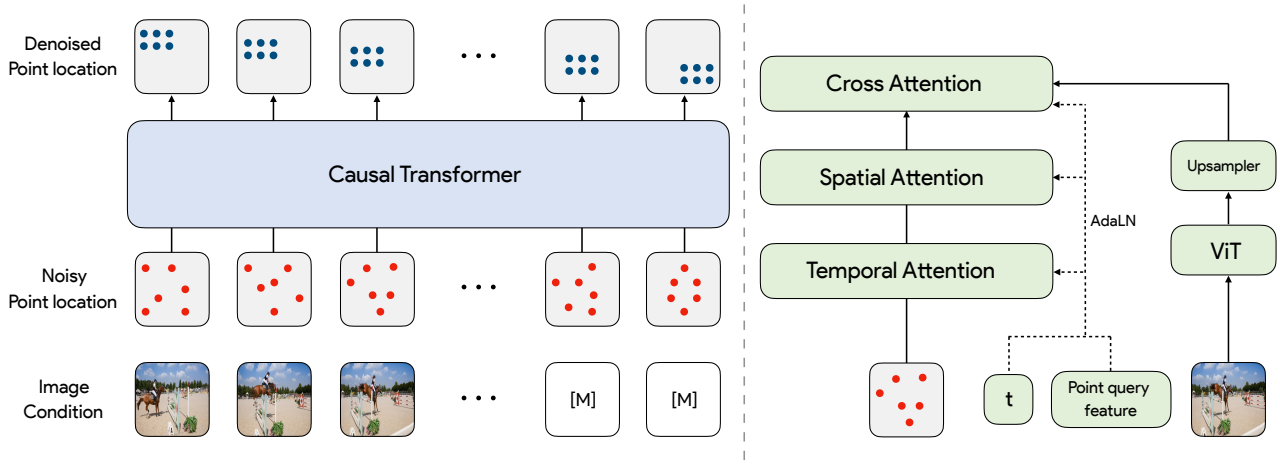
Motion estimation. The problem of estimating motion has a long history in computer vision. Early work formulated the problem as predicting dense optical flow by trading off brightness constancy and smoothness [4, 11, 33, 51, 55, 70], while later work trained models using supervised learning [26, 36, 37, 60, 71, 73, 78, 82]. Another line of work, pioneered by Sand and Teller [64], formulated the problem as tracking a set of points over long time horizons. Later work by Harley et al. [29] and Doersch et al. [18], inspired by these ideas, defined the *point tracking* problem and developed deep learning approaches. Later work proposed new datasets, architectures and loss functions [2, 19, 38, 39, 42, 45, 46] and semi-supervised learning strategies [20]. These approaches treat tracking as a regression problem, with robust loss functions to deal with occlusion. In contrast, our approach poses tracking as a video-to-point generation problem and also addresses point forecasting. Our work is closely related to the very recent work of Zholus et al. [86], which formulates tracking as a next token prediction problem. Like us, this approach uses a generative formulation for tracking. However, they use autoregression instead of flow matching. They also do not capture the joint distribution between points, since the model uses parallel decoding of all points (i.e., all points are conditionally independent given the images and previous point locations). Consequently, it is not possible to directly apply the model to forecasting (only tracking).

Forecasting. The capability to forecasting the future is essential for intelligence, which has many applications, such as robot planning [7, 75]. In general, forecasting can happen in any space: RGB [6, 12, 47, 56, 65], and robotic actions [16, 84]. A variety of recent works have performed point forecasting in the robotics domain [7, 75, 79], by predicting how points will move due to an instruction or action. Like them, we predict future point locations, but we predict them from an initial video signal rather than from robotic actions. Our forecasting approach is most closely related to (and builds on the benchmark of) the point forecasting work of [9]. They train a model that closely resembles an image-based latent diffusion model to predict future point positions, using a variational autoencoder that assigns a latent code to the points within an image patch, conveying their motion. In contrast, our model treats points as tokens and does not assume that the input is grid-based. Moreover, we use a single model to simultaneously solve both point tracking and forecasting by varying our visual conditioning, which is not possible with their architecture without major modifications.

Diffusion Models. Diffusion models [31, 67, 68] or flow matching models [49, 50, 53] are trained to reverse the forward process to traverse from source distribution to target distribution, which has been applied in many domains such as vision [8, 31, 61, 62, 68], audio [14, 15, 34, 52], robotics [16, 48], language [1, 25, 54, 63], and tactile signals [22, 80, 81]. Specifically, the forward process gradually adds noise to data over several time steps, transforming it into pure noise. The reverse process then learns to denoise the data step by step, reconstructing the original data from the noise. Prior work has used various conditions to guide diffusion models for sampling, among which text [24, 61, 62] and image [41, 43, 44] are two significant conditions. Text is a type of flexible control signal, which provides a lot of high-level semantic guidance, for many applications such as text-to-image generation [8, 24, 62] and image editing [10]. Image provides the "context", containing a large amount of high-frequency visual cues, is also useful for many tasks like depth estimation [41], planning [12, 23], or even world modeling [3, 6]. Usually, diffusion models iteratively denoise the whole "sequence" over denoising steps [8, 74]. Recently, there is a line of work [12, 35] combining autoregression and diffusion, where causal attention over the time horizon and iterative denoising are employed. In our paper, we use similarly high-level architecture designs conditioned on images for point sampling.

3. Method

We propose a single, unified generative modeling approach that addresses both point tracking and trajectory forecasting. We frame this as a unified generation problem in *point-*



(a) Method architecture (b) Diffusion block

Figure 2. **Method Overview**. We propose a single generative model that performs both forecasting and tracking using a diffusion transformer, operating with image conditioning for tracking and without image conditioning for forecasting. Noise is added to the point locations across time, and the model is trained to denoise these trajectories. Each diffusion block contains three types of attention: *temporal attention*, applied across the same point over time; *spatial attention*, applied among all points within the same frame; and *cross attention*, applied between point location tokens and image feature tokens.

space, capable of both conditional generation (tracking), by aligning trajectories with visual input, and unconditional generation (forecasting), by sampling from its learned motion prior. This capability is controlled simply by the presence or absence of visual conditioning, allowing the model to toggle between tasks without any architectural changes. We build our model using a diffusion transformer trained with the flow matching objective [49].

3.1. Unified Tracking and Forecasting

We formulate this problem as learning a conditional probability distribution $p_\theta(P, V \mid I_C, Q)$ where $I_C \in \mathbb{R}^{T_C \times H \times W \times 3}$ is the conditioning video sequence of T_C frames, and $Q \in \mathbb{R}^{N \times 2}$ is the set of N query point locations given in the first frame. Our goal is to generate the full point trajectories $P \in \mathbb{R}^{T \times N \times 2}$ and their corresponding visibilities $V \in [0,1]^{T \times N}$ over a target time span T, where $T \geq T_C$.

The model's behavior at any timestep t is determined by the availability of visual conditioning I_t:

- **Tracking (Conditional Generation):** For timesteps $t \leq T_C$, visual features from frame I_t are provided as conditioning. The model is guided to generate a trajectory P_t and visibility V_t that align with this visual input.
- **Forecasting (Unconditional Generation):** For timesteps $t > T_C$, no visual information is provided. The visual condition is replaced by a learned null embedding $\emptyset$. The model seamlessly transitions to forecasting, sampling from its learned internal motion prior to generate a plausible future trajectory P_t and visibility V_t.

This allows our model to perform pure tracking (when $T_C = T$), pure forecasting (when $T_C = 1$), or a combination of tracking followed by forecasting (when $1 < T_C < T$). To learn this conditional distribution, we use conditional flow matching (CFM) [49, 53].

3.2. Point-Space Diffusion Transformer

Our model architecture is a Diffusion Transformer (DiT) [57] that operates directly in point-space. We represent the full trajectory, including positions P and visibilities V, as a sequence of $T \times N$ tokens. Our model learns to map from a Gaussian distribution to the data distribution $\mathbf{x} = (P, V)$. This point-space design follows pixel-space diffusion [32], which avoids the need for complex grid-based VAE encoders and decoders over unstructured point sets. We learn a denoising function $F_\theta(\mathbf{x}_k, k, C)$, where C represents our joint conditioning signal (Q, I_C). We avoid task-specific architectures, such as cost volumes, that are common in state-of-the-art tracking methods.

Transformer Blocks. Our DiT consists of L blocks. Following recent work in conditional transformers [13, 24], each block is composed of three attention modules:

- **Spatial Attention:** A self-attention layer where all point tokens $\mathbf{x}_k^{(t)}$ at the same frame t attend to each other. This models the interactions and joint motion of points within a single timestep.
- **Causal Temporal Attention:** A causal self-attention layer where each point token $\mathbf{x}_k^{(t,n)}$ attends to its own past, $\{\mathbf{x}_k^{(t',n)} \mid t' < t\}$. This allows the model to build a representation of an individual point's motion history.

- **Cross-Attention:** A cross-attention layer where all point tokens $\mathbf{x}_k^{(t)}$ attend to the visual features C_t from the corresponding frame for positional and semantic information.

The cross-attention layer allows us to easily toggle between tracking (by providing C_t) and forecasting (by providing $\emptyset$).

Visual Conditioning. Given the conditioning video I_C, we use a pretrained vision transformer (ViT) [21] as a feature extractor. Each frame I_t is passed through the encoder to obtain a set of feature maps $C_t \in \mathbb{R}^{H' \times W' \times D_{feat}}$ from multiple layers. The set of feature maps is concatenated and projected into the feature dimension of the model. It is then upsampled with an upsampling layer with nearest interpolation and a convolution layer. These are used within cross-attention layers in the transformer blocks.

Query and Timestep Conditioning. Adaptive layer normalization (AdaLN) [58] is commonly used in DiTs to inject conditioning information. In a typical DiT, AdaLN applies a *global* conditioning vector (e.g., representing timestep k or a class label) uniformly to all tokens. Our approach differs by computing a *per-point* conditioning signal, providing each of the N trajectories with its unique starting context, in addition to the global timestep. We bilinearly sample a visual feature from the feature map at the query location Q_n. This is combined with a positional embedding of Q_n and a global embedding of the timestep k. This final vector modulates the transformer blocks using AdaLN layers, informing the model of the query information for each trajectory and the current noise level. For brevity, we omit the query Q from subsequent formulas as it is always used as the conditioning signal.

3.3. Training and Inference

We train a temporally autoregressive flow matching model that predicts both position and visibility.

Flow Matching for Position. We train our model using conditional flow matching (CFM) [49, 53]. CFM frames diffusion as learning a vector field that transports a noise distribution p_0 to the data distribution p_1. We define a probability path $p_k(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid (1-k)\mathbf{x}_0 + k\mathbf{x}_1, \sigma^2)$ and train our network F_θ to predict the flow $\mathbf{v}_k(\mathbf{x}) = \mathbf{x}_1 - \mathbf{x}_0$ by minimizing the L_2 loss:

$$\mathcal{L}_{\text{position}} = \mathbb{E}_{k, p_k(\mathbf{x}|\mathbf{x}_1), \mathbf{c}} \left[\|F_\theta(\mathbf{x}_k, k, \mathbf{c}) - (\mathbf{x}_1 - \mathbf{x}_0)\|_2^2 \right] \quad (1)$$

where $\mathbf{x}_1$ is the ground-truth trajectory, $\mathbf{x}_0 \sim \mathcal{N}(0, I)$ is the noise, and $\mathbf{c}$ represents our optional conditioning signal I_C. While CFM allows for a simple uniform time sampling $t \sim \mathcal{U}(0, 1)$, we find that the choice of distribution of k during training is critical, similar to findings in pixel-space models and high-resolution models. We normalize our trajectories to have zero mean and unit variance. For sampling k, we use a logits-normal distribution [24] and perform a search over the distribution's location and scale to find an optimal distribution for our task.

Binary classification for visibility. We pose visibility prediction as a binary classification task, which is conditioned on previous and current point positions, as well as visibility estimates. Instead of the L_2 flow-matching objective used for positions, our network F_θ is trained to predict the clean ground-truth visibility V_1 at each diffusion step. This direct prediction is supervised with a standard binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{visibility}} = \mathbb{E}_{k, \mathbf{x}_k, \mathbf{c}, V_1} \left[\texttt{BCE}(\hat{V}_1, V_1) \right], \quad (2)$$

where V_1 is the ground-truth visibility and $\hat{V}_1$ is the visibility component predicted by our model $F_\theta(\mathbf{x}_k, k, \mathbf{c})$. Similar to iterative regression work [38], we find it sufficient to use the visibility prediction at the last sampling step as final prediction.

Autoregressive training and inference. To handle long trajectories T that may not fit into memory, we perform generation in a temporally autoregressive manner. During inference, the model generates the first window of W frames and starts to shift with a stride of S. The last $W - S$ generated frames (from the previous window), which contain both predicted positions and their corresponding visibilities, are then used as a causal prefix condition for the next S frames. As a result, the model will track or forecast the entire time span T with a total of $\frac{T-W}{S} + 1$ windows.

Autoregressive generation of continuous data, however, is notoriously prone to error accumulation due to exposure bias [12]. While existing methods often solve this by feeding model predictions back during training or using distillation, these approaches can introduce complex training strategies or difficulties in parallelization. For simplicity, we use diffusion forcing [12], which assigns independent noise levels k to each frame during training. This breaks the temporal dependency on a "perfect" ground truth history and teaches the model to denoise from any context state. During inference, as the window slides forward, the previously predicted part (the $W - S$ prefix) is re-noised with a small noise level. This prevents the model from overrelying on its own past predictions.

Robustness to variable number of points. Point tracking applications must support tracking a user-specified set of points, which can widely vary in size. Diffusion models are also known to be sensitive to the number of tokens, which can differ between training and test time on our task. We observe a degradation in denoising performance when the number of query points N at inference is smaller than the number used during training. To address this, we introduce a simple yet highly effective *factorization* strategy. Given a training sample with a total of N_{train} points, we find all factor pairs (a, b) such that $a \times b = N_{\text{train}}$. At each training step, we randomly sample a pair (a, b) and reshape the N_{train} points into a batch of a samples, each contain-

ing b points. This is equivalent to training the model to denoise a sets of b points separately, teaching it to adapt to diverse point counts in a single forward pass. We find that this approach keeps compute balanced across GPUs in parallel training and leads to significant performance gains.

3.4. Implementation Details

Dataset. For forecasting experiments, we train and evaluate our model on Kubric MOVi-A [27], Physics101 [76], DriveTrack [5], and Kinetics [40], independently. For tracking experiments, we train the tracking task on the *MOVi-E Panning* version of the Kubric dataset, utilizing the data release from [17]. For unified modeling, we train the model jointly on the target forecasting dataset and MOVi-E tracking dataset, with a 50/50 sample mixing.

Architecture. Similar to Aydemir et al. [2], we base our model on pretrained self-supervised visual features. Specifically, we use a pretrained DINO-v3-S [66] vision encoder. To align the size of the feature map to existing methods on tracking [2, 38, 39], we resize each input image such that its shorter side is 768 pixels, which ensures consistent resolution across scenes, and use an upsampling block to provide a $2\times$ larger feature map resolution. We observe that upsampling the image and feature map leads to noticeable performance gains, suggesting that higher-resolution visual features benefit both tracking and prediction tasks.

On top of the DINO-v3-S visual features, we adopt a DiT-style transformer architecture [57] for unified tracking and prediction. Conditioning signals and timestep embeddings are injected through modulation layers. In our main experiments, the transformer consists of 6 blocks with a hidden dimension of 384. Each block contains three key components: temporal attention, spatial attention, and point–image attention, enabling the model to jointly reason over time, space, and cross-modal interactions. We apply RoPE embedding in temporal attention, and axial RoPE embedding indexed by query location in spatial attention [69]. We also use RMS layer normalization [83] and QK-normalization [30].

Training. The model is trained with the flow-matching loss, using a logits-normal noise sampling (loc=-1, scale=1.5). We use AdamW optimizer, gradient accumulation, gradient clipping, and exponential moving average (EMA). For our main experiments, we train the model for 200k steps. For ablation experiments that require training, we set image resolution to 384 pixels, use a smaller EMA parameter, and train for 100k steps.

For tracking-only model, visual condition is always present during training. For forecasting-only model, visual condition is only present for the query frame. For unified model, we randomly select a frame index and mask all visual input after it. We provide more details in the supplementary material.

Inference. Unless otherwise specified, we use a stride of 8 and context noise of 0.15 for tracking, a stride of 1 and context noise 0.02 for forecasting. We provide more details in the supplementary material.

4. Experiments

We evaluate our proposed approach on both point forecasting and tracking.

4.1. Forecasting

4.1.1. Quantitative Results

Synthetic Kubric benchmark. Point-trajectory forecasting is inherently ambiguous, as many reasonable future trajectories can occur and a unique ground truth often does not exist. To ensure a thorough evaluation, we adopt the protocol of Boduljak et al. [9] and employ a suite of complementary metrics, including *FVMD*, *Best of K*, and *LRTL*. These metrics collectively measure forecasting performance from multiple perspectives.

Best of K measures the lowest MSE among K predicted point trajectories and K simulated ground-truth trajectories for the same input image, similar to Chamfer distance.[1] *FVMD* computes the Fréchet distance between generated and ground-truth (simulated) point trajectories using trajectories-based features, like FID or KID in image generation task. Finally, *LRTL* measures the *physical plausibility* of the predicted motions, with a particular focus on rigidity during the generated trajectories.

We jointly train our model on both tracking and forecasting tasks, using a mix of examples in each training mini-batch with (and without) visual conditioning signal (see supplementary material for details). We evaluate and train the model on the benchmark proposed in [9]. The evaluation dataset is a MOVi-A variant synthesized with Kubric, containing 16 scenes, each with 64 trajectories generated under different initial velocities.

We compare our unified model to state-of-the-art trajectory prediction methods in Table 1. We find that it outperforms all existing trajectory forecasting methods, demonstrating the effectiveness of our proposed ideas. These previous methods (proposed by Boduljak et al. [9]) contain generative latent diffusion models that predict future point locations, as well as video generation baselines that predict future frames then run a tracker to obtain point locations. We exhibit qualitative results in the supplementary material.

We also evaluate our model on an out-of-distribution subset of Kubric in Table 2, following the setup of Boduljak et al. [9]. Our model again outperforms all state-of-the-art

[1] We found that there is an error in Boduljak et al. [9]'s implementation of this metric that causes it to overestimate the metric distance. However, it leaves the ranking between methods intact. For consistency with previous work, we use their implementation.

Table 1. **Kubric motion forecasting.** We closely follow the evaluation protocol of Boduljak et al. [9]. Our method shows better adherence to the ground truth motion distribution over multiple metrics. †- model fine-tuned to Kubric dataset. We restate results from Boduljak et al. [9].

Method	FVMD (Scene) ↓	Best of K ↓	LRTL ↓
Video Generators			
WAN 14B [74]	42987	184.6	35.1
Stable Video Diffusion [8]	39494	235.7	37.2
LTX-Video [28]	32019	205.1	17.0
WAN 1.3B [74]	30712	192.6	42.1
DynamicCrafter† [77]	50123	239.9	51.8
Stable Video Diffusion† [8]	22799	152.2	30.1
WAN 1.3B† [74]	20010	162.8	26.6
Trajectory Generators			
Track2Act [7]	22509	250.8	15.8
Boduljak et al [9]	17838	127.0	14.1
Ours (Forecasting Only)	**17786**	**95.6**	14.6
Ours (Unified)	18091	98.6	**13.9**

Table 2. **Kubric motion forecasting (O.O.D.).** We evaluate the forecasting performance on an out-of-distribution version of Kubric following Boduljak et al. [9]. Our method shows better forecasting quality over multiple metrics. †- model fine-tuned to Kubric dataset. We restate results from Boduljak et al. [9].

Method	FVMD (Scene) ↓	Best of K ↓	LRTL ↓
Video Generators			
DynamicCrafter† [77]	49092	230.5	58.8
Stable Video Diffusion† [8]	19780	127.7	31.7
WAN 1.3B† [74]	16547	128.2	27.3
Trajectory Generators			
Track2Act [7]	19608	278.6	19.7
Boduljak et al [9]	14949	127.2	**15.9**
Ours (Forecasting Only)	**14651**	**82.3**	17.9

methods, demonstrating strong strong generalization ability.

Real-world physics. To evaluate our method's performance in real-world physical scenarios, we benchmark on Physics101 [76] following the setup of [9]. Physics101 contains over 10,000 videos of 101 objects interacting through collisions, falling, floating, etc. Since only one ground-truth outcome is provided per scenario, we report Mean Squared Error (MSE) as the primary metric.

Table 3. **Physics101 forecasting errors.**

Method	Fall	Liquid	Multi	Ramp	Spring	Overall
WAN	**16.05**	**4.48**	21.88	37.53	70.48	30.08
Boduljak et al. [9]	19.78	6.00	**15.65**	36.35	65.31	28.62
Ours	27.24	5.78	17.25	**31.01**	**25.76**	**21.41**

Our method outperforms all other baselines on Physics101, demonstrating its ability to forecast complex, non-linear dynamics. Qualitative demonstrations are provided in Fig. 3.

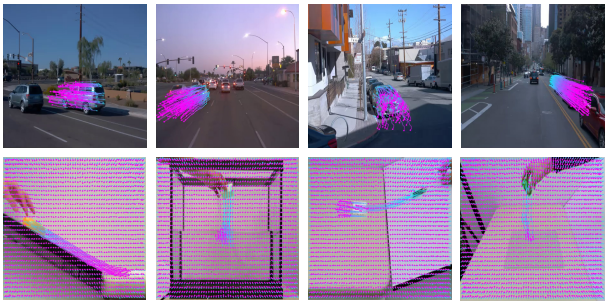


Figure 3. **Qualitative Results on DriveTrack and Physics101**. We visualize the forecasted trajectories on DriveTrack and Physics101 test set. Color from blue → purple indicates time progression. Our model can forecast complex non-linear dynamics and infer motion from scene context in real-world videos.

4.1.2. Qualitative Results in the Real World

Autonomous driving. We qualitatively evaluate our method on DriveTrack [5], a point-tracking dataset built on the Waymo Open dataset [72]. Using the $80/10/10$ splits from [5], we visualize forecasting samples in the test set in Fig. 3. The results show that our method successfully infers motion from scene context in complex driving environments.

General videos. To assess performance on unstructured scenes, we pseudo-labeled 30% of the Kinetics training set [40] using CoTracker3 [39] to train our model for general motion forecasting. For this experiment, we employ 12 DiT blocks with 768 channels. We visualize samples from the Kinetics validation set and the DAVIS dataset [59] in Fig. 1 and Fig. 4. We provide the first 8 frames as a visual condition to stabilize camera motion and prevent static generation. The results demonstrate that our model generates plausible motions for humans, objects, and complex 3D scenes alike. We provide more visualizations and failure mode analysis in the supplementary material.

4.2. Tracking

We follow the standard evaluation protocol for point tracking using the TAP-Vid benchmark and its associated metrics: *OA*, δ^{vis}_{avg}, and *AJ*. *OA* (Occlusion Accuracy) measures the correctness of occlusion prediction, treating it as a binary classification task. δ^{vis}_{avg} computes the average visibility accuracy by reporting the fraction of tracked points whose predicted positions fall within 1, 2, 4, 8, and 16 pixels of the ground-truth locations. *AJ* (Average Jaccard) provides a holistic metric that jointly evaluates both tracking accuracy and occlusion prediction quality. For fair evaluation, all the evaluated videos are resized to 256×256 pixels.

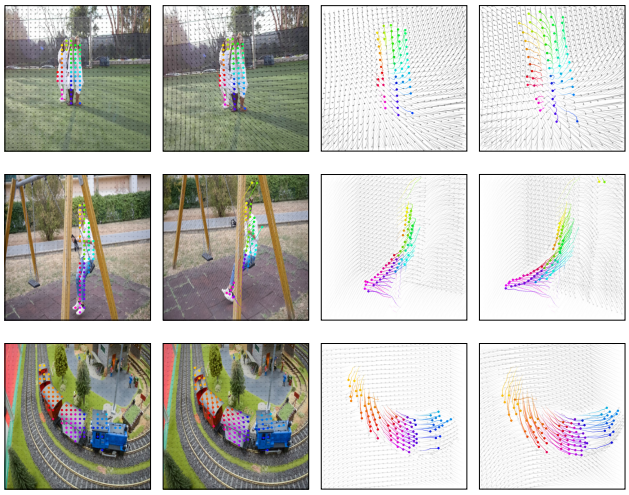


Figure 4. **Qualitative Results on DAVIS**. We train our model on pseudo-labeled Kinetics datasets, and visualize the forecasted trajectories on DAVIS dataset. The first two columns show the conditioning frames with tracking results, and the next two columns show the trajectory forecasting. We color background points in gray to highlight the motion of main subject.

Table 4. **TAP-Vid benchmarks.** Our model without task-specific designs for tracking attains competitive performance compared to current state-of-the-art methods. We restate results from Karaev et al. [39], Zholus et al. [86]. A comprehensive table is provided in the supplementary material.

Method	Train	Kinetics			DAVIS		
		AJ↑	δ^{vis}_{avg} ↑	OA↑	AJ↑	δ^{vis}_{avg} ↑	OA↑
TAPNet [85]	PointOdyssey	—	—	—	33.0	48.6	78.8
PIPs++ [85]	PointOdyssey	—	—	—	—	73.7	—
TAPIR [19]	Kubric	49.6	64.2	85.0	56.2	70.0	86.5
CoTracker [38]	Kubric	49.6	64.3	83.3	61.8	76.1	88.3
TAPTRv2 [45]	Kubric	49.7	64.2	85.7	63.5	75.9	**91.4**
LocoTrack [17]	Kubric	52.9	66.8	85.3	62.9	75.3	87.2
Track-On [2]	Kubric	53.9	67.3	**87.8**	**65.0**	**78.0**	90.8
TAPNext [86]	Kubric	53.3	**67.9**	87.0	62.4	76.6	90.5
CoTracker3 [39]	Kubric	**54.1**	66.6	87.1	64.5	76.7	89.7
Ours (Tracking Only)	Kubric	50.5	65.1	87.6	61.7	75.6	90.6
Ours (Unified)	Kubric	49.8	64.0	87.1	59.9	73.8	89.7

We consider both the performance of a model that solely performs tracking and a unified model that jointly trains on tracking and MOVi-A forecasting (Fig. 4). We find that our tracking-only variant obtains highly competitive performance compared to current state-of-the-art methods on all metrics, despite those methods relying on task-specific designs tailored exclusively for tracking, while its performance on the Kinetics subset is relatively lower. We also found that the unified model obtains similar (though slightly lower) performance than the tracking-only model.

Table 5. **Effect of tracking context on forecasting errors.** Longer context windows provide richer temporal information, allowing the model to generate more realistic forecasts.

# context frame	Kubric	Physics101
1	1059	18.99
2	535	15.82
4	299	14.04
8	174	11.42

4.3. Interplay between Tracking and Forecasting

In this section, we discuss some unique benefits and capabilities of our proposed unified method that a pure tracker or forecaster cannot achieve.

Flexible conditioning for forecasting. Unlike specialized trackers or forecasters, our unified model is able to treat tracking as motion prompts. It can track arbitrary number of frames as condition, then seamlessly switch to forecasting. In Tab. 5, we show that increasing context length directly improves forecasting errors on both Kubric and Physics101. It suggests that the model gathers a stronger context of motion patterns during the tracking phase to guide forecasting.

Motion prior allows robust tracking. As mentioned above, unified training typically leads to a slight decrease in tracking performance, under model capacity constraints. However, unified training allows the model to learn motion prior from training distribution, which could benefit tracking under challenging scenarios. We compare the tracking performance of our tracking-only and unified model through experiments that (a) vary whether forecasting training and evaluation distribution matches, and (b) introduce synthetic occlusions (black squares of 25% image size at the center of frames). Tab. 6 shows that an out-of-domain motion prior leads to performance degradation. However, when the learned motion prior is in-domain, unified model achieves significant performance gain under occlusion, as it imagines plausible trajectories behind occlusions.

Table 6. **Effect of unified training on tracking performance under occlusion** (a black square of 25% image size placed at the center of frames). We report both δ^{vis}_{avg} and δ^{occ}_{avg}.

		Tracking only		**Unified**	
Train	**Eval**	δ^{vis}_{avg}	δ^{occ}_{avg}	δ^{vis}_{avg}	δ^{occ}_{avg}
Kubric	DAVIS	75.6	61.2	73.8 (-1.8)	60.7 (-0.5)
Kubric	Kubric	89.9	48.0	89.1 (-0.8)	50.5 (+2.5)
Physics101	Physics101	88.5	68.8	88.3 (-0.2)	82.2 (+13.4)
DriveTrack	DriveTrack	86.8	42.6	88.7 (+1.9)	56.9 (+14.3)

4.4. Ablation Experiments

We further investigate our model by addressing a set of key questions through targeted ablation experiments.

Table 7. **Comparison between diffusion loss and regression loss.** We train the same model using a deterministic regression objective and compare it against our diffusion-based formulation, demonstrating the clear effectiveness of the diffusion loss.

Loss	AJ	$\delta^{\text{vis}}_{\text{avg}}$	OA
L2 Regression	30.2	42.5	85.5
L1 Regression	43.3	58.2	86.7
Diffusion	53.7	67.3	89.3

Table 8. **Design Ablations.** We evaluate the effectiveness of the proposed components on the tracking task.

	AJ	$\delta^{\text{vis}}_{\text{avg}}$	OA
Ours	53.7	67.3	89.3
w/o factorization	36.6	46.4	72.3
w/o noise schedule	51.6	65.7	89.3

How does diffusion loss help tracking? We study two related tasks—*tracking* and *forecasting*. While diffusion objectives naturally align with forecasting, it is less clear whether they also benefit tracking, which is typically formulated as a deterministic regression or matching problem.

Table 7 provides preliminary evidence by comparing our diffusion-based model with prior non-diffusion approaches. To more directly isolate the effect of the diffusion objective, we conduct an ablation in which we vary the training loss, using either an $\mathcal{L}_2$ or $\mathcal{L}_1$ regression loss, while keeping the network architecture fixed. This setup disentangles architectural factors from the optimization objective, allowing us to quantify the contribution of the diffusion loss alone.

As shown in Table 7, variants trained with deterministic regression losses perform consistently worse than those trained with the diffusion loss.

How does factorization and noise schedule help? We evaluate the impact of several proposed design choices, including factorization and our noise schedule, in Table 8. These simple modifications are surprisingly effective, collectively yielding substantial performance gains.

How does the sliding window stride affect autogressive generation? To handle longer sequences for both tracking and forecasting, we adopt an autoregressive strategy: the video is divided into overlapping sliding windows, and the predictions from the previous window are used as context for the next. To understand how this design choice affects performance, we investigate the influence of the sliding-window configuration, particularly the stride.

Table 9. **Sliding Window Ablation.** We study the effect of the stride we shift the sliding window forward.

Stride	FVMD (S)	Best of K	LRTL
1	17786	95.6	14.6
2	18820	100.8	16.4
4	21309	100.2	22.5
8	23361	104.1	28.2

As shown in Table 9, smaller strides allow us to scale the computation during inference, consistently leading to better results.

Table 10. **Context Noise Level Ablation.** We investigate the influence of the level we re-noise the context in sliding window.

Context Noise	FVMD (S)	Best of K	LRTL
0.02	17786	95.6	14.6
0.05	18283	92.7	16.6
0.10	19814	95.4	21.0
0.15	21552	96.8	25.8

How does context noise level affect autoregressive generation? In the autoregressive generation setting, although the predictions from the previous window are used as context for the next, we follow diffusion forcing and inject a small amount of noise into these inputs. This encourages the model to be robust to accumulated prediction errors and helps maintain stability over long sequences. As shown in Table 10, we observe that forecasting performance improves with decreased context noise level, suggesting a reliance on a cleaner history for stable generation.

5. Conclusion

We have proposed a unified architecture for *point generation* that can be applied to both point tracking and forecasting. This model is based on a flow matching model for point positions that is conditioned on a visual signal. Despite our model's simplicity, it obtains strong performance on both tasks, obtaining results that outperform previous approaches on the point forecasting benchmark of Boduljak et al. [9] and that are competitive with many recent methods on the TAP-Vid tracking benchmark [18].

We see our model as a step toward creating flexible, general-purpose models for motion, which we anticipate opening two directions. The first is creating motion estimation systems that can accept a wide range of different conditioning signals, such as the goals or possible actions that an agent may take. The second is to create methods that take advantage of the generative structure in our model, such as modeling uncertainty in tracking through occlusion.

Acknowledgement. We would like to thank Gabrijel Boduljak, Inès Hyeonsu Kim, and Carl Doersch for their assistance with data acquisition, and Ayush Shrivastava and Yan Xu for helpful discussions. This work was supported by the Advanced Research Projects Agency for Health (ARPA-H) under award #1AY2AX000062. This research was funded, in part, by the U.S. Government. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021. 2

[2] Görkay Aydemir, Xiongyi Cai, Weidi Xie, and Fatma Güney. Track-on: Transformer-based online point tracking with memory. *arXiv preprint arXiv:2501.18487*, 2025. 1, 2, 5, 7

[3] Yutong Bai, Danny Tran, Amir Bar, Yann LeCun, Trevor Darrell, and Jitendra Malik. Whole-body conditioned egocentric video prediction. *arXiv preprint arXiv:2506.21552*, 2025. 2

[4] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. In *International Journal of Computer Vision*, pages 221–255, 2004. 2

[5] Arjun Balasingam, Joseph Chandler, Chenning Li, Zhoutong Zhang, and Hari Balakrishnan. Drivetrack: A benchmark for long-range point tracking in real-world videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22488–22497, 2024. 5, 6

[6] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025. 2

[7] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pages 306–324. Springer, 2024. 2, 6

[8] 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. 2, 6

[9] Gabrijel Boduljak, Laurynas Karazija, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. What happens next? anticipating future motion by generating point trajectories. *arXiv preprint arXiv:2509.21592*, 2025. 2, 5, 6, 8

[10] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 2

[11] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004. 2

[12] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024. 2, 4

[13] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 3

[14] Ziyang Chen, Prem Seetharaman, Bryan Russell, Oriol Nieto, David Bourgin, Andrew Owens, and Justin Salamon. Video-guided foley sound generation with multimodal controls. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18770–18781, 2025. 2

[15] Ho Kei Cheng, Masato Ishii, Akio Hayakawa, Takashi Shibuya, Alexander Schwing, and Yuki Mitsufuji. Taming multimodal joint training for high-quality video-to-audio synthesis. *arXiv e-prints*, pages arXiv–2412, 2024. 2

[16] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44 (10-11):1684–1704, 2025. 2

[17] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking, 2024. 5, 7

[18] 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*, 2022. 2, 8

[19] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 1, 2, 7

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

[21] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[22] Yiming Dou, Fengyu Yang, Yi Liu, Antonio Loquercio, and Andrew Owens. Tactile-augmented radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26529–26539, 2024. 2

[23] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023. 2

[24] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 2, 3, 4

[25] Chao Feng, Zihao Wei, and Andrew Owens. Masked diffusion captioning for visual feature learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25247–25263, Suzhou, China, 2025. Association for Computational Linguistics. 2

[26] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick Van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015. 2

[27] 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*, 2022. 5

[28] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024. 6

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

[30] Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4246–4253, 2020. 5

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[32] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023. 3

[33] Berthold Horn and Brain Schunck. Determining optical flow. In *Artificial Intelligence*, pages 185–203, 1981. 2

[34] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, pages 13916–13932. PMLR, 2023. 2

[35] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025. 2

[36] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8981–8989, 2018. 2

[37] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 2

[38] 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*, pages 18–35. Springer, 2024. 1, 2, 4, 5, 7

[39] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 1, 2, 5, 6, 7

[40] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5, 6

[41] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9492–9502, 2024. 2

[42] Inès Hyeonsu Kim, Seokju Cho, Jahyeok Koo, Junghyun Park, Jiahui Huang, Honglak Lee, Joon-Young Lee, and Seungryong Kim. Anthrotap: Learning point tracking with real-world motion. *arXiv preprint arXiv:2507.06233*, 2025. 2

[43] Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. Intrinsic image diffusion for indoor single-view material estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5198–5208, 2024. 2

[44] Peter Kocsis, Lukas Höllein, and Matthias Nießner. Intrinsix: High-quality pbr generation using image priors. *arXiv preprint arXiv:2504.01008*, 2025. 2

[45] Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Feng Li, Bohan Li, Tianhe Ren, and Lei Zhang. Taptrv2: Attention-based position update improves tracking any point. *Advances in Neural Information Processing Systems*, 37: 101074–101095, 2024. 2, 7

[46] Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. Taptr: Tracking any point with transformers as detection. In *European Conference on Computer Vision*, pages 57–75. Springer, 2024. 2

[47] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024. 2

[48] Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5637–5643. IEEE, 2025. 2

[49] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2, 3, 4

[50] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024. 2

[51] Ce Liu et al. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 2

[52] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023. 2

[53] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 2, 3, 4

[54] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023. 2

[55] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. Vancouver, British Columbia, 1981. 2

[56] Karran Pandey, Yannick Hold-Geoffroy, Matheus Gadelha, Niloy J Mitra, Karan Singh, and Paul Guerrero. Motion modes: What could happen next? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2030–2039, 2025. 2

[57] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 1, 3, 5

[58] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 4

[59] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[60] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017. 2

[61] 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. 2

[62] 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. 2

[63] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024. 2

[64] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International journal of computer vision*, 80(1):72–91, 2008. 2

[65] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[66] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 5

[67] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015. 2

[68] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[69] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 5

[70] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2432–2439. IEEE, 2010. 2

[71] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume, 2017. 2

[72] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 6

[73] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020. 2

[74] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2, 6

[75] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 2

[76] Jiajun Wu, Ilker Yildirim, Joseph J Lim, William T Freeman, and Joshua B Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in Neural Information Processing Systems*, pages 127–135, 2015. 5, 6

[77] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2024. 6

[78] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022. 2

[79] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024. 2

[80] Fengyu Yang, Jiacheng Zhang, and Andrew Owens. Generating visual scenes from touch. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22070–22080, 2023. 2

[81] Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26340–26353, 2024. 2

[82] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *Advances in neural information processing systems*, pages 794–805, 2019. 2

[83] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019. 5

[84] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 2

[85] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 7

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