Flow and Depth Assisted Video Prediction with Latent Transformer

Eliyas Suleyman[1]
y.sulaiman.1@research.gla.ac.uk
Paul Henderson[1]
paul.henderson@glasgow.ac.uk
Eksan Firkat[2]
eksan@mail.tsinghua.edu.cn
Nicolas Pugeault[1]
nicolas.pugeault@glasgow.ac.uk

[1] School of Computing Science
University of Glasgow,
Glasgow, UK

[2] Tsinghua Shenzhen International
Graduate School
Tsinghua University
Shenzhen, China

Abstract

Video prediction is a fundamental task for various downstream applications, including robotics and world modeling. Although general video prediction models have achieved remarkable performance in standard scenarios, occlusion is still an inherent challenge in video prediction. We hypothesize that providing explicit information about motion (via point-flow) and geometric structure (via depth-maps) will enable video prediction models to perform better in situations with occlusion and the background motion. To investigate this, we present the first systematic study dedicated to occluded video prediction. We use a standard multi-object latent transformer architecture to predict future frames, but modify this to incorporate information from depth and point-flow. We evaluate this model in a controlled setting on both synthetic and real-world datasets with not only appearance-based metrics but also Wasserstein distances on object masks, which can effectively measure the motion distribution of the prediction. We find that when the prediction model is assisted with point flow and depth, it performs better in occluded scenarios and predicts more accurate background motion compared to models without the help of these modalities.

1 Introduction

Video prediction is a crucial task for intelligent agents, with applications in robotics [19], autonomous driving [52], world models [44], and weather forecasting [34]. Accurately predicting the near future (e.g., 1–2 seconds) is essential, as it directly improves an agent's decision-making efficiency [4]. The development of vision transformers for images [8] and videos [1] have made possible to improve video prediction quality [30, 32, 39, 55]. However, due to the inherent complexity of motion in dynamic scenes with multiple objects, occlusions frequently occur, and latent transformer models can still struggle to accurately estimate the motion of objects that become temporarily invisible [40].

Several approaches aim to improve video prediction by incorporating optical flow estimation [2, 27, 29, 59]. However, two major limitations of optical flow are that it accumulates errors over time, and loses information when objects become fully occluded. As a result, optical flow-based methods struggle to handle complete occlusions effectively. Unlike optical flow, which relies on dense pixel-wise motion estimation, recent progress in point tracking methods enable more robust occlusion handling by tracking and estimating key points on objects even when they are fully occluded [22, 41, 50]. Furthermore, background motion is also well represented by point-flow, which is essential for modeling the motion of an ego-camera (e.g., autonomous cars). Equally critical to occlusion handling, depth maps can provide geometric structure of the scene, allowing for better spatial reasoning in occlusion scenarios.

In this work, we hypothesize that integrating information about depth and the flow of points into a video prediction model will enhance its ability to anticipate object and background motion, particularly in occluded scenarios. While point-flow helps track object motion trajectories, depth maps introduce explicit spatial constraints that improve occlusion-aware prediction. To investigate this, we use latent transformer as our video prediction model, which lacks robustness to occlusions when only relying on RGB images [40], and propose a variant that incorporates both point-flow and depth as additional modalities. Our approach enables the model to retain motion information when objects become temporarily invisible, improving future frame prediction accuracy by leveraging both motion trajectories and spatial structure alongside visual cues. Furthermore, with the assistance of point-flow, the background motion can be predicted more accurately with more precise direction of background motion.

Our main contributions are as follows:

- We provide the first systematic analysis of how depth and point-flow impact the performance of prediction when dynamic scenes have occlusions and background motion.
- We design a video prediction model that can incorporate point-flow and depth as additional modalities to improve RGB frame prediction.
- We conduct extensive experiments on both synthetic and real world datasets, with several model variants and baselines.
- We find that when integrating point flow, the reappearance of occluded objects and the background motion are predicted more accurately.

2 Related Work

General Video Prediction Methods. Various neural network architectures have been explored for video prediction: hybrid models that combine RNNs and CNNs [5, 12, 45]; latent transformer [43] models [40, 47, 51], where the latent is usually encoded by a VQ-VAE or VQ-GAN [10, 35, 42] then the prediction is conducted by a transformer; diffusion [17], latent-diffusion [36] and diffusion-transformer [31] based approaches [26, 56]. Although the overall performance of these approaches is promising, since they lack explicit object or motion information, learning to generate complex motion is very expensive in terms of data and compute.

Optical Flow in Video Prediction. Optical flow is a pixel-wise dense motion estimation between consecutive video frames. FlowNet [9] and its advanced version [20] is first introduced to estimate the optical flow through CNN network. Recent optical flow estimation approaches used vision transformers to achieve the same goal [23, 28, 37]. Because optical flow contains rich motion information of a dynamic scene, it is integrated to many video prediction approaches to predict future frames. Li et al [24] first predict the optical flow of future frames by condition on a single frame, then warp the RGB frame with predicted flow to achieve video prediction. Shi et al [38] used a similar idea to predict the flow first then use a diffusion model conditioned on flow to generate RGB frames. Bei et al [2] proposed a semantic aware approach that predicts the optical flow directly with a ConvLSTM network, then uses the predicted flow to generate the future frames. Wu et al [48] used optical flow to optimize the model's frame interpolation ability to improve the future frame prediction quality. Liang et al [25] generated video frames based on another video's optical flow information. Optical flow has also been integrated with generative diffusion models to guide the motion of generated frames to be more realistic [6]. However, error accumulation over time and the complete loss of information while objects are occluded hampers the effectiveness of optical flow methods when occlusion occurs.

Point Tracking. Point tracking approaches have recently gained popularity due to their strong performance [7, 22, 41, 50]. Unlike optical flow estimation, which aims to estimate the motion of every pixel in an image, point tracking methods typically operate in an encoded latent space and focus on tracking sparse, semantically meaningful features. Rather than modeling dense pixel-level motion, these methods estimate the trajectories of key features across frames, making them more robust to noise, occlusions, and appearance changes. This abstraction allows tracking-based approaches to better capture high-level motion dynamics and structural consistency compared to traditional flow-based methods. Several studies have attempted to integrate point tracking for motion modeling and future trajectory prediction. For instance, [4] leveraged point tracking to assist robotic arm control in completing various tasks, achieving superior performance. Point tracking has also been applied to generative tasks. [21] incorporated point tracking into video diffusion models, enabling more realistic motion generation.

3 Methodology

3.1 Preliminaries

Let $X^{1:T} = \langle x^1, x^2, ..., x^T \rangle$, be a sequence of T RGB frames from a video clip, where $x^t \in \mathbb{R}^{h \times w \times 3}$. Our goal is to learn a probability distribution on future frames $X^{T+1:T+M}$, conditioned on the past frames $X^{1:T}$. We next discuss the base model we build on in this paper, as well as the models used to extract additional modalities—point-flow and depth.

Base Architecture. We use the Stochastic Class-Attended Transformer (SCAT) [40] as the backbone structure for our model. SCAT is a recent latent-transformer-based approach designed for object-centric video prediction. It is a two-staged approach that first trains an object aware auto-encoder (OAAE) to encode the video frames into latent representations. Then, a GPT-style transformer is trained on the past latent frames to predict future latent frames. Finally, the predicted latent frames are decoded via OAAE to reconstruct the predicted frames. SCAT offers a favorable trade-off between performance and computational

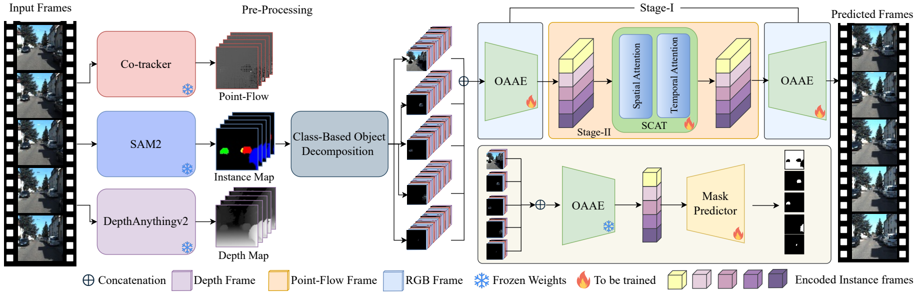

Figure 1: **The overview of the proposed method.** First we obtain different modalities by using Cotracker and DepthAnythingV2; then we use SAM2 to segment the original RGB frames sequence to decompose the objects, segmentation map from SAM2 is also used to decompose the point-flow and depth map; After preprocessing, we first train OAAE to convert the frames into a latent space; then we train SCAT to predict the future latent frames; finally the predicted latent future frames are reconstructed by trained OAAE; The lower right box shows how we train a object mask predictor based on trained OAAE's latent space; after mask predictor is trained, it is then used solely for evaluating EMD.

cost, using a relatively lightweight transformer module for temporal prediction; it achieves higher accuracy for similar parameter count versus similar non-object-centric models.

In the first stage, each frame x from a video clip is decomposed by an off-the-shelf instance segmentation model [33] yielding I number of instances with m number of classes. Each instance is extracted by applying the corresponding mask predicted by the instance segmentation model. The full frame can be reconstructed by adding all of the masked instances, i.e. $x = \sum_{k=1}^{I} \tilde{x}_k$, where $\tilde{x}_k$ represents the k^{th} instance. Then, each instance is encoded by a set of class-specific encoders, denoted as $\Phi = \{\phi_1, \phi_2, ..., \phi_m\}$, each of which encodes instances belonging to one class. Then the encoded latent is quantized by class-specific embedding code books $E = \{e_1, e_2, ..., e_m\}$. Each quantized instance is concatenated to generate a structured latent representation z that captures the full frame. A joint decoder Ψ then reconstructs the original video frame x from the latent representation z. To enhance feature extraction and improve performance, we replace the encoder used in SCAT with SlotDiffusion's frame encoder [49], which uses a ResNet-18 [16] architecture. This modification provides stronger feature representations, enabling more effective encoding of object-centric information. Moreover, the latent space is significantly smaller than SCAT's while maintaining similar or better performance

After the OAAE is trained, video clips are converted from RGB images into latent representation using OAAE. Since each frame contains structured information of the instances, we can represent instances sequence as $\tilde{Z}_k = \{z_k^1, z_k^2, ..., z_k^T | k = 1, 2, ..., I\}$, then the sequence representing the full scene can also obtained additively by summing all instance sequences as $Z = \sum_{k=1}^{N} \tilde{Z}_k$, In second stage, SCAT uses class-specific transformer blocks for each semantic class, similarly to the first stage. It models the motion pattern of an instance k with self-attention individually (eq. 1), as well as the potential relationship with other instances via cross-attention (eq. 1) as shown in the equation blow:

$$\mathrm{SA}_c(\tilde{Z}_k) = \mathrm{softmax}\left(\frac{Q_k {K_k}^T}{\sqrt{d_k}}\right) V_k, \quad \mathrm{CA}(\tilde{Z}_k) = \bigoplus_{i=1,\ldots,N,\, i \neq k} \mathrm{softmax}\left(\frac{Q_k {K_i}^T}{\sqrt{d_k}}\right) V_i \tag{1}$$

where $\bigoplus$ denotes concatenation operation; The cross-attention layer's output, being $I-1$ times larger than the input because of concatenation, is reduced to the original size through a linear layer. Additionally, each transformer block's attention mechanism further operates on spatial and temporal dimensions to effectively capture both spatial and temporal dependencies, following [3], both within instance k and across other instances. The final output is a probability distribution over OAAE codebook indices for each instance in each future frame.

Point Tracking with Cotracker. Cotracker [22] is a transformer-based model that tracks 2D points in video sequences. First, the query points are initialized on the first frame of a video clip, with their initial positions and visibility. A point P_i at time step t is represented as $P_i^t = (x_i^t, y_i^t) \in \mathbb{R}^2$,for $t \in \{1, \ldots, T\}$. It is set to make all points visible after it is initialized at the first time step (e.g first frame of a video clip) to reduce ambiguity. After the points are initialized, an end-to-end CNN network is trained to obtain the feature map of the frames. Then each point is projected to the relative position on the feature map, and the corresponding feature is selected for the point. Finally, a transformer model is trained iteratively to learn how these points and selected feature maps are correlated. The objective of this model is to minimize the distance between the predicted and ground truth point locations.

Depth Estimation with DepthAnything-V2. Depth Anything [53, 54] is a monocular depth estimation model designed to generalize well across diverse real-world scenes. It follows a semi-supervised learning approach, where a teacher-student framework is employed to leverage both synthetic and real data. Initially, a teacher network is trained on a large-scale synthetic dataset with dense ground-truth depth annotations. This teacher is then used to pseudo-label a large corpus of real-world unlabeled images, effectively transferring its knowledge to real data. Finally, a student network is trained on a mixture of these pseudo-labeled real images and a small set of manually labeled real-world samples. The model takes a single RGB frame as input and produces a dense depth map as output. We use the second version as our depth estimator for video frames.

3.2 Proposed Method

SCAT [40] decomposes a video into object instances, and an instance that becomes completely occluded at a certain time step is not visible to the encoder. This makes it difficult to predict the motion of fully occluded objects, even when explicit visual information about these objects is available. We therefore propose incorporating tracked points from Cotracker as point-flows, providing additional information to the prediction model. Point-flows offer a more effective and robust alternative to optical flow for achieving object tracking, as optical flow tends to accumulate errors over time [15]. By utilizing point-flows, the encoder can retain information about an instance's position at a certain time step t, even when its RGB image is entirely absent due to complete occlusion.

We hypothesize that incorporating point-flows alongside RGB frames during encoding will enrich the latent representations with relative location information. Therefore, the motion of occluded objects can be predicted more accurately. Depth images are integrated as a another modality to our model, providing geometric context that is invariant to appearance changes. While point flows capture motion, depth encodes scene structure, aiding in disambiguating object movement and handling occlusions—especially under camera motion—thus improving spatial and temporal reasoning. It is important to note that we do not require any additional or richer information to train our model. Instead, we use pretrained models solely to preprocess the available RGB sequences, generating point-flow and depth images from the same input data used by existing baselines. Following SCAT, we test our hypothesis by designing a family of models with varying input configurations: **SCAT-D**, trained with RGB frames and depth frames; **SCAT-P**, trained with RGB frames and point-flows; and **SCAT-DP** trained with RGB frames, depth images, and point-flows.

Point flow and Depth. We first use Cotracker to track points in a video clip, then calculate the point-flow as the displacements of each point between consecutive frames. For the initial time step ($t = 0$), there are no displacements, as the points are treated as the initial reference positions, represented by a tensor of shape $(T, N, 3)$, where T is the number of frames, N is the number of points, and 3 represents the (h, w) coordinates and visibility. From the second frame and onward ($t \geq 1$), the horizontal and vertical displacements of each point are calculated as the difference between the current and previous positions. Finally, since each point is defined by its (h, w) coordinates, the displacement information is mapped to a grid with the same size as the image, resulting in a tensor of shape $(T, H, W, 3)$, where H and W represent the height and width of the video frame resolution. The last dimension encodes horizontal displacement, vertical displacement, and visibility. We therefore have

$$\mathbf{PointFlow}(T,H,W,3) = \begin{cases} (0,0,1), & \text{if } t = 0, \\ (h_t^n - h_{t-1}^n, w_t^n - w_{t-1}^n, v_t^n), & \text{if } t > 0. \end{cases} \tag{2}$$

where $\mathbf{PointFlow}(T,H,W,3)$ is the displacement tensor, $x_{t,n}$ and $y_{t,n}$ are the (h, w) coordinates of the n^{th} point and $v_{t,n}$ is the visibility of the n^{th} point at time step t. (H, W) corresponds to the pixel grid location in the image, derived from the (h, w) coordinates of each point. This mapping ensures that the point-flows retain spatial correspondence with the video frames, enabling effective integration with the encoder.

For depth images, we employ an off-the-shelf depth estimation model [53] to generate the depth information for non-synthesized datasets. Since a video sequence is composed of instance sequences, the corresponding points and depth information are extracted via segmentation map used to decomposed the instances.

After we obtain these modalities, we concatenate them with the original RGB frame on the channel dimension to form the input of the encoder. Then, all of these information will be encoded together according to different variants of our proposed method. Finally, the model's output is not just a single RGB frames but as well as other modalities. This makes sure that other modalities will be encoded to the latent space.

Loss Function. Since our approach has two stages, we need to train the frame encoder first and then train the temporal predictor. For the frame encoder, we modify the original VQ-loss and Commitment Loss to fit our model design. We extend VQ loss for each semantic class separately because each instance is encoded via a class-specific encoder and codebook, then the overall reconstruction loss for RGB images, depths and point-flows is calculated.

L_{VQ}, L_{recon} is shown below:

$$\mathcal{L}_{VQ} = \sum_{c=1}^{m} \sum_{k=1}^{n_c} \|\mathrm{sg}[\tilde{z}_k^c] - e_c\|_2^2, \quad \mathcal{L}_{commitment} = \sum_{c=1}^{m} \sum_{k=1}^{n_c} \|\tilde{z}_k^c - \mathrm{sg}[e_c]\|_2^2$$

$$\mathcal{L}_{recon} = -\log p(x|\Psi(\Phi(x))) \tag{3}$$

where sg denotes the stop-gradient operator, n_c represents the number of instances in class c, and e_c corresponds to the codebook for class c, respectively. We also include LPIPS [57] as an additional reconstruction loss:

$$\mathcal{L}_{\mathrm{LPIPS}}(x, \Psi(\Phi(x))) = \sum_l w_l \, \|\phi_l(x) - \phi_l(\Psi(\Phi(x)))\|_2^2 \tag{4}$$

where $\phi_l(x)$ represents the deep feature maps extracted from the l-th layer of a pretrained network ϕ. The term w_l is a learned weight that adjusts the contribution of each layer to the overall similarity, and $\|\cdot\|_2^2$ denotes the squared Euclidean distance between feature representations. The final objective of our encoder will be summing all loss terms together as $\mathcal{L} = \mathcal{L}_{VQ} + \mathcal{L}_{commitment} + \mathcal{L}_{recon} + \mathcal{L}_{\mathrm{LPIPS}}$.

For the transformer model that predicts future frames in latent space, we use the same formulation as SCAT, i.e. minimizing the cross entropy between target and predicted indices.

4 Experiments

We first conduct a series of experiments to analyze the impact of each additional modality on future frame prediction using the proposed family of models. Our primary focus is on evaluating occluded scenarios under controlled settings, enabling a systematic assessment of how well each modality improves performance in handling occlusions. We focus our evaluation on the predicted RGB frames and moving object's mask but not the other modalities which are simply regarded as guidance for the model. To demonstrate the generality of the proposed method, we also evaluate it on more diverse scenarios and compare its performance against other baselines. In each experiment, we follow SCAT's experimental setups, where the proposed model is required to predict 5 to 20 future frames given five input frames. All experiments are conducted on a single NVIDIA RTX 3090 GPU, and the model sizes (e.g., number of parameters) of other baselines are adjusted accordingly to ensure a fair comparison.

4.1 Datasets

Kubric Occlusion: The hypothesis of this paper is that incorporating point-flow can improve the performance of prediction models, particularly in scenarios involving occlusions. To test this, we used Kubric [14] to generate video clips tailored for our evaluation, which we refer as **Kubric-Occlusion**. A total of 1,800 video clips were generated, with 1,300 used for training and 500 for testing. In each clip, one object remains stationary at a random location, while another object appears at a random position and moves behind the stationary object, creating an occlusion event.

KITTI: The **KITTI dataset** [13] is a widely used benchmark for autonomous driving research. It contains diverse driving scenarios captured in urban, residential, and highway environments. In this work, we use a **subset** of KITTI, specifically selecting scenes from *city, residential, and road* categories. We first preprocess the dataset to obtain all of the car instances; then, we sort the segmented car instances by size and select the largest four as foreground objects; the remainder of the image is categorized as background. After processing, 2,497 clips are used as training and 639 for testing (each clip contains 10 frames).

4.2 Evaluation Metrics

We evaluate the pixel-level quality of predicted frames using standard metrics: PSNR[18], LPIPS[58], and SSIM[46]. However, since the primary focus of our work is on assessing motion in the predicted frames, appearance-based metrics alone are insufficient to capture the dynamic aspects of prediction quality. To address this, we introduce the optical flow difference (OFD), which measures the discrepancy in motion between predicted and ground truth frames. Optical flow is computed using the Gunnar-Farneback method [11], the motion accuracy is then quantified by calculating the mean squared error (L_2 loss) between the predicted and ground truth flows.

In addition to global motion assessment via OFD, we further evaluate motion quality at the instance level. We train a mask predictor to predict instance masks from the trained VQ-VAE latent space, and use this to estimate masks for predicted frames. We then compute the Earth Mover's Distance (EMD) (also known as Wasserstein distance) between the ground truth masks. While OFD captures overall scene motion, EMD provides a finer-grained analysis of motion distribution differences, offering a more accurate reflection of motion quality in predicted frames. EMD we use in our paper is defined as follows: Let $P_t = \{\mathbf{p}_1, \dots, \mathbf{p}_m\} \subset \mathbb{R}^2$ be the set of pixel coordinates for the predicted mask, and $G_t = \{\mathbf{g}_1, \dots, \mathbf{g}_n\} \subset \mathbb{R}^2$ be the set of pixel coordinates for the ground truth mask. We define uniform discrete distributions over these sets: $\mathbf{a} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right) \in \Delta^m, \quad \mathbf{b} = \left(\frac{1}{n}, \dots, \frac{1}{n}\right) \in \Delta^n$ Let $M \in \mathbb{R}^{m \times n}$ be the cost matrix with entries: $M_{ij} = \|\mathbf{p}_i - \mathbf{g}_j\|_2$ The Earth Mover's Distance (squared Wasserstein distance) is computed as the optimal transport cost:

$$\mathrm{EMD}^2(P, G) = \min_{T \in U(\mathbf{a}, \mathbf{b})} \sum_{i=1}^{m} \sum_{j=1}^{n} T_{ij} M_{ij}$$

where $U(\mathbf{a}, \mathbf{b}) = \{T \in \mathbb{R}_+^{m \times n} \mid T\mathbf{1}_n = \mathbf{a},\ T^\top \mathbf{1}_m = \mathbf{b}\}$ is the set of admissible transport plans. All metrics are computed on a per-frame basis, and the values reported in the table represent the mean over all frames across the clips in the respective dataset.

4.3 Results

In this section, we first evaluate the proposed family of models by comparing their performance internally on occluded and general scenarios with backround motion. Then we select the best performing model to compare against other similar approaches. The goal is to analyze the contribution of each additional modality (point-flow, depth, or both) and determine which variant performs best under different conditions. By conducting these internal experiments, we aim to identify the most effective model configuration before benchmarking it against other similar approaches.

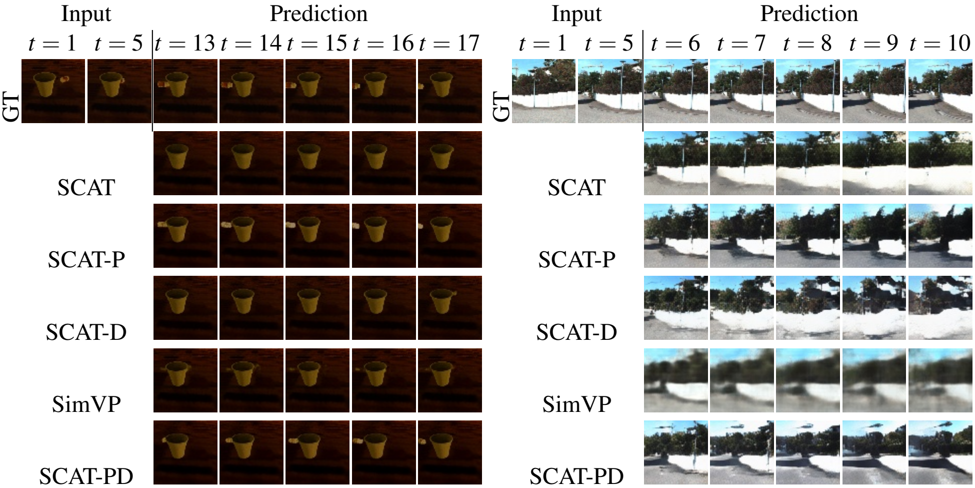

Figure 2: Comparison of different model variants on the **Kubric-Occlusion (Left)** and **KITTI (Right)** dataset.

	Kubric-Occlusion						**KITTI**					
	PSNR↑	SSIM↑	LPIPS↓	OFD↓	EMD↓	Prms	PSNR↑	SSIM↑	LPIPS↓	OFD↓	EMD↓	Prms
SCAT	25.88±0.13	0.658±0.007	0.064±0.001	0.0423±0.0017	0.0081±0.0005	11M	15.33±0.13	**0.473±0.006**	0.135±0.003	2.5776±0.3341	0.0310±0.0016	8M
SCAT-P	25.99±0.13	0.665±0.007	0.063±0.001	0.0356±0.0016	0.0070±0.0006	11M	15.20±0.11	0.448±0.006	0.155±0.003	1.6659±0.1939	0.0282±0.0020	8M
SCAT-D	**26.53±0.13**	**0.701±0.006**	**0.054±0.001**	0.0414±0.0019	0.0069±0.0004	11M	**15.53±0.12**	0.465±0.006	**0.132±0.003**	3.2781±0.4762	0.0285±0.0022	8M
SCAT-PD	25.69±0.12	0.649±0.007	0.072±0.002	**0.0347±0.0014**	**0.0066±0.0004**	11M	15.36±0.11	0.445±0.006	0.137±0.002	**1.6390±0.2324**	**0.0278±0.0016**	8M

Table 1: Quantitative results on **Kubric-Occlusion** and **KITTI** dataset

From Table 1, we can see that on the Kubric-occlusion dataset, all proposed variants improve on plain SCAT in terms of motion metrics. This confirms our hypothesis that flow and depth modalities are important for occlusion prediction. Interestingly, the SCAT-D variant achieves the best performance for appearance metrics (PSNR, SSIM & LPIPS), and SCAT-P achieves the best results for motion-relevant metrics (OFD & EMD). We found the performance of the SCAT-PD variant to be generally lower than the two other (SCAT-P and SCAT-D), which is likely a consequence of processing larger input data with the same model size. In Figure 2, we can also see the qualitative results reflect the quantitative scores: The occluded object's reappearance is only predicted correctly when point flow is integrated (SCAT-P and -PD), confirming the evidence provided by the OFD and EMD metrics.

Table 2 provide a comparison to SimVP and plain SCAT. The proposed model appear to underperform SimVP when looking at appearance-based metrics on the **Kubric-Occlusion** dataset, however they perform much better when looking at motion-based metrics. This contrast can be explained by the comparatively small impact of moving objects on appearance metrics versus background noise, which is likely reduced by the larger size of the SimVP model. This intuition is confirmed by the qualitative results shown in Figure 2, where SCAT-P & SCAT-DP predict accurately the motion of moving objects while others fail. Specifically,

	Kubric-Occlusion					**KITTI**				
	PSNR↑	SSIM↑	LPIPS↓	OFD↓	Num-Params	PSNR↑	SSIM↑	LPIPS↓	OFD↓	Num-Params
SCAT	25.88±0.13	0.66±0.007	0.064±0.001	0.0423±0.0017	11M	15.33±0.13	0.47±0.006	**0.135±0.002**	2.49±0.33	8M
SimVP	**33.05±0.13**	**0.95±0.001**	**0.021±0.001**	0.0626±0.0019	14M	**17.14±0.10**	**0.49±0.005**	0.332±0.004	1.66±0.11	14M
Ours	25.69±0.12	0.65±0.007	0.072±0.002	**0.0347±0.0014**	11M	15.36±0.11	0.45±0.006	0.137±0.002	**1.64±0.23**	8M

Table 2: Comparison to previous works on **Kubric-Occlusion** and **KITTI** dataset

the trajectory of the moving object in Kubric-Occlusion (left) dataset is correctly predicted only when including point-flow information (SCAT-P & SCAT-PD), while SimVP fails to predict the object's reappearance.

In contrast, where KITTI features complex real world dynamics, our model outperforms SimVP in LPIPS (0.137 v 0.332). Also, we see that in terms of motion our model also outperformed SimVP (1.64 v 1.66), this can be seen in Figure 2 (right). The Figure 2 shows clear evidence that when the point-flow is integrated, the backgorund motion is predicted accurately (SCAT-P, D & PD) versus RGB-only variants. It is important to note that our SCAT variants are nearly two times smaller than SimVP and achieved similar or better performance in terms of motion accuracy.

5 Conclusion

We propose a video prediction pipeline that investigates the impact of adding point tracking and depth information on future frame prediction. Our method incorporates point-flow and depth maps to enhance motion prediction, particularly in challenging scenarios with occlusions and background motion. Experimental results show that point-flow contributes to more accurate motion estimation, and in particular can successfully predict the reappearance of occluded moving objects. However, while adding multiple modalities improves general motion prediction, the additional input information can degrade pixel-level appearance quality when keeping model size constant. In future work, we aim to explore strategies for a better integration of diverse modalities and improving reconstruction fidelity.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[2] Xinzhu Bei, Yanchao Yang, and Stefano Soatto. Learning semantic-aware dynamics for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 902–912, 2021.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning (ICML)*, volume 2, page 4, 2021.

[4] 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 (ECCV)*, 2024.

[5] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. Mau: A motion-aware unit for video prediction and beyond. *Advances in Neural Information Processing Systems*, 34:26950–26962, 2021.

[6] 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 preprint arXiv:2502.02492*, 2025.

[7] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. Flowtrack: Revisiting optical flow for long-range dense tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19268–19277, 2024.

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

[9] 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*, pages 2758–2766, 2015.

[10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[11] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pages 363–370. Springer, 2003.

[12] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3170–3180, 2022.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013.

[14] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022.

[15] 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.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[18] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. doi: 10.1109/ICPR.2010.579.

[19] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

[20] 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.

[21] Hyeonho Jeong, Chun-Hao Paul Huang, Jong Chul Ye, Niloy Mitra, and Duygu Ceylan. Track4gen: Teaching video diffusion models to track points improves video generation. *arXiv preprint arXiv:2412.06016*, 2024.

[22] 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, 2025.

[23] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: connecting the dots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2024.

[24] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 600–615, 2018.

[25] Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, et al. Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2024.

[26] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

[27] Wei Lu, Junyun Cui, Yanshuo Chang, and Longmei Zhang. A video prediction method based on optical flow estimation and pixel generation. *IEEE Access*, 9:100395–100406, 2021.

[28] Yawen Lu, Qifan Wang, Siqi Ma, Tong Geng, Yingjie Victor Chen, Huaijin Chen, and Dongfang Liu. Transflow: Transformer as flow learner. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18063–18073, 2023.

[29] Weixin Luo, Wen Liu, Dongze Lian, and Shenghua Gao. Future frame prediction network for video anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7505–7520, 2021.

[30] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.

[31] 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.

[32] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *arXiv preprint arXiv:2006.10704*, 2020.

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[34] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

[35] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[36] 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.

[37] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1599–1610, 2023.

[38] 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.

[39] Binit Singh, Divij Singh, Rohan Kaushal, Agrya Halder, and Pratik Chattopadhyay. Gsstu: Generative spatial self-attention transformer unit for enhanced video prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[40] Eliyas Suleyman, Paul Henderson, and Nicolas Pugeault. On the benefits of instance decomposition in video prediction models. *arXiv preprint arXiv:2501.10562*, 2025.

[41] Narek Tumanyan, Assaf Singer, Shai Bagon, and Tali Dekel. Dino-tracker: Taming dino for self-supervised point tracking in a single video. In *European Conference on Computer Vision*, pages 367–385. Springer, 2024.

[42] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[43] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[44] Xiaofeng Wang, Zheng Zhu, Guan Huang, Boyuan Wang, Xinze Chen, and Jiwen Lu. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv preprint arXiv:2401.09985*, 2024.

[45] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, S Yu Philip, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 2208–2225, 2022.

[46] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

[47] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *arXiv preprint arXiv:2405.15223*, 2024.

[48] Yue Wu, Qiang Wen, and Qifeng Chen. Optimizing video prediction via video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17814–17823, 2022.

[49] Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. *Advances in Neural Information Processing Systems*, 36:50932–50958, 2023.

[50] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024.

[51] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[52] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14662–14672, June 2024.

[53] 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*, pages 10371–10381, 2024.

[54] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024.

[55] Xi Ye and Guillaume-Alexandre Bilodeau. Video prediction by efficient transformers. *Image and Vision Computing*, 130:104612, 2023.

[56] Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346*, 2023.

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[59] Zhicheng Zhang, Junyao Hu, Wentao Cheng, Danda Paudel, and Jufeng Yang. Extdm: Distribution extrapolation diffusion model for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19310–19320, 2024.