Explicit Object-Centric Video Prediction with Deep Learning Models

Yiliyasi Sulaiman

Submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy

School of Computing Science

College of Science & Engineering

September 2025

To my beloved mother,
in loving memory, who lives forever in my heart.

Abstract

Video prediction is a crucial task for intelligent agents such as robots and autonomous vehicles, it enables them to anticipate and act early on time-critical incidents. Many state-of-the-art video prediction methods typically model the dynamics of a scene jointly and implicitly and seeing it as a single entity, without any explicit decomposition into separate objects. This is sub-optimal, as every object in a dynamic scene has their own pattern of movement, typically somewhat independent of others. Therefore, we hypothesize that explicit modelling of moving objects is crucial for video prediction in limited data and compute scenarios.

We first investigate video prediction with multiple moving and interacting objects in a static camera setting within the context of a latent-transformer as the video predictor. We conduct detailed and carefully-controlled experiments on both synthetic and real-world datasets; our results show that decomposing a dynamic scene leads to higher quality predictions compared with models of a similar capacity that lack such decomposition. We then investigate the trajectory prediction of occluded objects and scenes with background motion which is a common phenomena in real-world scenarios. We introduce explicit motion information, depth map and point flow, to assist the prediction model we proposed previously. We investigate this approach in both synthetic and real-world scenarios. The experimental results shows that with the integration of explicit motion information, the predicted trajectory of dynamic objects is more accurate. We finally investigate the case of deformable objects such as scenes in garment manipulation tasks. We introduced a diffusion variant of our proposed video prediction model to better handle the motion prediction of fully deformable objects because of its continuous nature compared to transformer-based architectures. By testing it on a garment manipulation dataset, we find that our diffusion-based variant outperformed our transformer-based models.

Our findings suggest that for video prediction models to accurately model motion patterns inside a dynamic scene, scaling up holistic models are inefficient and recourse consuming. In contrast, decomposition of objects and modeling with their explicit motion information can be a better and more efficient alternative compared to monolithic models with same capacity. Furthermore, this setting implies that it can be more useful in closed-world settings like robotic manipulation tasks where limited objects are in the scene.

Contents

List of Tables

List of Figures

Acknowledgements

I have my deepest gratitude to my supervisor, Dr. Nicolas Pugeault. He has shown me what it means to be a researcher and taught me to always ask the question "Why?". His patience, encouragement, belief in me, and generosity, whether through providing resources or personal support, made all the difference. I also want to express my heartfelt thanks to my co-supervisor, Dr. Paul Henderson. He has been invaluable throughout my PhD, offering deep insights and constructive advice that continually shaped the direction of my work. Together, without their mentorship and constant support, this PhD would not have been possible, and I could not have asked for better supervisors.

I am also grateful to the School of Computing Science, University of Glasgow, for funding and supporting my PhD for 3.5 years. Without this financial support, it would not have been possible to begin and successfully complete this journey.

I would like to thank my Annual Progression Review panel members, Dr. Fani Deligianni and Dr. Gerardo Aragon Camarasa, for their valuable insights and critical feedback, which guided my research in the right direction.

I am deeply grateful to my friends and colleagues in Glasgow, whose support and companionship enriched my PhD experience. In particular, I wish to thank Stefanos Sagkriotis, Thomas Jänich, Fabricio Mendoza Granada, Yingdong Ru, Zhuo He, Erlend Frayling, Talha Enes Ayrancı, José Rodríguez Bacallado, Javier Sanz-Cruzado Puig, Shiyu Fan, Ozan Bahadır, Luca Löettegen, and Rory Young.

I also want to express my deepest gratitude to my father, Suleyman Halik, and my brother, Yashar Suleyman. Ever since I was a child, their excellence effected me to be a better person. Throughout my PhD, they have constantly supported me both financially and emotionally without a question. Without their strong support and love, this PhD would never have been possible.

Finally, I am especially grateful to my partner, Amina Abulimiti. Her unconditional love, patience, deep belief in me, and simply being there for me during difficult times have always been a source of strength. I cannot thank her enough for her unwavering support in helping me pursue my dreams.

Declaration

I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

Yiliyasi Sulaiman

Abbreviations

- `AE` - Auto-Encoder
- `AAE` - Adverserial Auto-Encoder
- `BERT` - Bidirectional Encoder Representations from Transformers
- `CA` - Cross Attention
- `CCTV` - Closed Circuit Television
- `CUDA` - Compute Unified Device Architecture
- `CNN` - Convolutional Neural Network
- `CPU` - Central Processing Unit
- `CV` - Computer Vision
- `DiT` - Diffusion Transforemr
- `DDIM` - Denoising Diffusion Implicit Model
- `DDPM` - Denoising Diffusion Probabilistic Model
- `ELBO` - Evidence Lower Bound
- `EMD` - Earth Mover's Distances
- `FA-VAE` - Frequency Augmented Variational AutoEncoder
- `FCM` - Frequency Complement Modules
- `FFL` - Focal Frequency Loss
- `FLOPs` - Floating point Operation per second
- `GAN` - Generative Adverserial Network
- `GRU` - Gated Recurrent Unit
- `GPT` - Generative Pre-trained Transformer
- `GPU` - Graphical Processing Unit

- KITTI - Karlsruhe Institute of Technology and Toyota Technological Institute Dataset
- KL Divergence Kullback-Leibler Divergence
- KTH - Kungliga Tekniska Högskolan (Royal Institute of Technology)
- LDM Latent Diffusion Model
- LLM Large Language Model
- LPIPS - Learned Perceptual Image Patch Similarity
- LSTM - Long Short Term Memory
- MAE - Masked Auto-Encoder
- MT - Machine Translation
- MSE - Mean Squared Error
- NLP - Natural Language Processing
- OAAE - Object Aware Auto-Encoder
- OFD - Optical Flow Difference
- PSNR - Peak Signal to Noise Ratio
- Swin - Shifted Window
- ReLU - Rectifying Linear Unit
- RGB - Red Green Blue
- RGB-D - Red Green Blue Depth
- RNN - Recurrent Neural Network
- SA - Self Attention
- SAM - Segment Anything Model
- SCAT - Stochastic Class Attended Transformer
- SSIM - Structural Similarity
- VAE - Variational Auto-Encoder
- ViT - Vision Transformer
- VGG - Visual Geometry Group
- ViViT - Video Vision Transformer
- VQ-VAE - Vector Quantized Variational Auto-Encoder
- xLSTM - Extended Long Short Term Memory

Chapter 1

Introduction

Humans have an inherent desire to anticipate and prepare for the future. From checking the weather forecast before deciding what to wear, to analyzing financial trends when managing resources, we constantly seek ways to reduce uncertainty and make informed decisions. Similarly, humans have the ability to anticipate near-term visual events based on observed cues. For example, when a snowball is thrown toward us, we instinctively estimate its trajectory to adjust our movements to dodge. Similarly, when a football player kicks a ball toward the goal, the goalkeeper anticipates its motion in order to block a potential score. These scenes often involve many surrounding objects, yet the motion of interest typically depends on only a few critical ones. For example, the snowball, the thrower and the person it is aimed at, or the ball, the kicker, and the goalkeeper. This can be generalized that a dynamic scene is the result of sparse interactions of objects despite the presence of multiple objects in the scene.

This fundamental drive to predict and prepare motivates the development of computational models that can forecast future events, such as those in visual environments. The field of video prediction focuses on forecasting the future evolution of visual scenes. Beyond the simple examples we mentioned previously, video prediction has a wide range of applications across different domains. For instance, by analyzing satellite imagery, prediction models can anticipate the future formation of clouds, enabling more accurate weather forecasting (Ravuri et al. 2021). In autonomous driving, such models can anticipate near-future events, such as pedestrians crossing the road, allowing the vehicle to react

in advance by decelerating (Yang et al. 2024a). Similarly, in robotics, video prediction is used in manipulation tasks, where forecasting the future states of objects enables robotic arms to plan and execute actions more effectively (Bharadhwaj et al. 2024), such as in object picking and placement.

The video prediction problem is the task to anticipate and generate a possible future dynamics of an observed visual scene, various deep learning architectures have been employed, including CNNs (Krizhevsky et al. 2012), RNNs (Williams and Zipser 1989), transformers (Vaswani et al. 2017), and diffusion models (Ho et al. 2020). Despite their success, these approaches often model the dynamics of sparsely interacting objects in an implicit manner by seeing the entire scene holistically, relying primarily on scaling the model size rather than explicitly capturing object-level interactions. This leads to heavy and suboptimal models which will require more computational resources to deploy for real world applications.

1.1 Scope of Thesis

Most video prediction approaches focus on modeling the dynamics of an entire scene as a whole (Wang et al. 2022; Yan et al. 2021; Wu et al. 2024; Pallotta et al. 2025). However, not every motion in a scene is equally relevant for predicting future states of interest. For example, when a goalkeeper anticipates the trajectory of a ball, the movements of the audience around the players are largely irrelevant, while the actions of the opposing players are crucial. This highlights the need for approaches that emphasize explicit modeling of key objects and their interactions.

In this thesis, we do not seek to introduce fundamentally new architectures for video prediction. Instead, our contribution lies in systematically studying the trade-offs between implicit and explicit object modeling in dynamic scenes. Specifically, we examine how incorporating object representations, their motions, and their spatial relations can lead to more efficient and interpretable video prediction.

Within this scope, we focus on scenarios where only a subset of objects meaningfully influences the future dynamics of the scene. By concentrating on these sparse interactions, we aim to improve predictive accuracy without unnecessarily increasing the complexity of a video prediction model. Thus, we explicitly model groups of interacting objects while treating background elements and irrelevant motions implicitly.

1.2 Thesis Statement

Accurate video prediction requires explicit modeling of the causes of motion in the scene. This thesis investigates the problem of modeling dynamic scenes with multiple interacting objects and predicting their future evolution over a fixed time horizon. Since interactions among objects in such scenes are typically sparse, modeling the entire scene as a single entity is both suboptimal and inefficient. We hypothesize, **Claim 1:** explicit object decomposition and learning the relationships between decomposed objects improves the quality of predicted future frames. Moreover, incorporating a cross-attention mechanism to capture potential object interactions further enhances prediction quality. Building on this hypothesis, **Claim 2:** integrating explicit motion information such as point flow and depth maps is beneficial for capturing specific dynamics, including occlusions and background motion. Finally, **Claim 3:** we hypothesize that continuous models, such as diffusion models, outperform discrete models in scenarios involving highly deformable objects, such as garments.

1.3 Contributions

The main contribution of this thesis is that it systematically investigated the benefits of decomposition of a scene into objects to predict their future dynamics in various settings such as static and ego-motion camera, rigid, semi-rigid (humans) and fully deformable objects (garments) with sparse interaction between the objects (including occlusions) on the scene. More specifically, the contribution of this thesis is as follows:

- We reveal the limitation of holistic video prediction models that need more parameters to capture the critical motion dynamics. In our research, we develop a family of explicit object-aware video prediction models that needs much less parameters than existing methods, but achieves better or similar performance in a static camera setting (Chapter 3).
- We find that the object decomposition alone is not sufficient for more complex dynamic scenes that features occlusion and background motion. To mitigate this limitation, we integrated explicit motion information, point-flow and depth map, to assist our video prediction model. Our thorough experiments confirmed that these additional modalities increase the motion prediction accuracy (Chapter 4).
- We finally studied the future motion prediction of highly deformable objects and their interaction. We modified the previously proposed approaches into a diffusion model and argue this continuous setting is beneficial for fully deformable objects. We demonstrated that diffusion-based object-aware video prediction model performs better than discrete auto-regressive transformer models (Chapter 5).

1.4 Origins of Material

Most of the material in this thesis is under review by multiple venues in the course of this PhD study:

- In Chapter 3, we present our first attempt of explicit object aware video predictor. This work is currently under review to be published in Transactions in Machine Learning Research (TMLR).
- In Chapter 4, we present the motion information integrated object aware video prediction model. This work is accepted as a workshop paper to be published at British Machine Vision Conference (BMVC 2025) SmartCamera 2025 workshop.
- In Chapter 5, we present the diffusion-based object-aware video prediction model. This work is being prepared to submit to the International Conference of Pattern Recognition (ICPR 2026) in December.

1.5 Outline of Thesis

The rest of this thesis is structured as follows:

In chapter 2, we introduce essential background knowledge used throughout this thesis including Auto-Encoders that we used as a frame encoder to encode the video frames into latent space; Transformer architecture which we used as prediction model in Chapter 3 and chapter 4 and the diffusion models that are used in our Chapter 5's prediction network. We also provide literature review of relevant video prediction models.

In Chapter 3, we will test the main hypothesis of this thesis that the dynamics of a scene is better learned with explicit decomposition of objects in this scene. We propose a transformer decoder-based explicit video prediction model and test this model in both synthetic and real-world scenarios.

In Chapter 4, occlusion and the background motion is studied in detail and a new approach incorporates point flow and depth is used to address the limitations of the previous chapter.

In both of the previous chapters, the main focus is on rigid object motion, therefore in Chapter 5, we will focus on the motion of highly deformable objects such as garments. Furthermore, we address the limitation for the frame encoder in the previous chapter.

In Chapter 6, we will discuss the main contribution of this thesis, and validate the thesis statement. Also, limitations and potential directions of future work are discussed in detail.

Chapter 2

Background

Deep learning has achieved remarkable progress over the past decade, revolutionizing fields such as computer vision and natural language processing. Video prediction, as a prominent problem in computer vision, has similarly benefited from these advances.

The methods proposed in this thesis build upon a range of deep learning architectures to tackle different challenges in video prediction. This chapter introduces the fundamental architectures that form the basis of our approaches and recent video prediction approaches that are relevant to our proposed methods, providing the reader with the necessary background to understand the subsequent technical chapters.

The rest of this chapter is organized as follows:

- Section 2.1 gives brief introduction to the fundamental deep learning architectures we used in this thesis such as Auto-Encoders, Transformer models and Diffusion models.
- Section 2.2 provides with a comprehensive literature review of video prediction models relevant to this thesis.

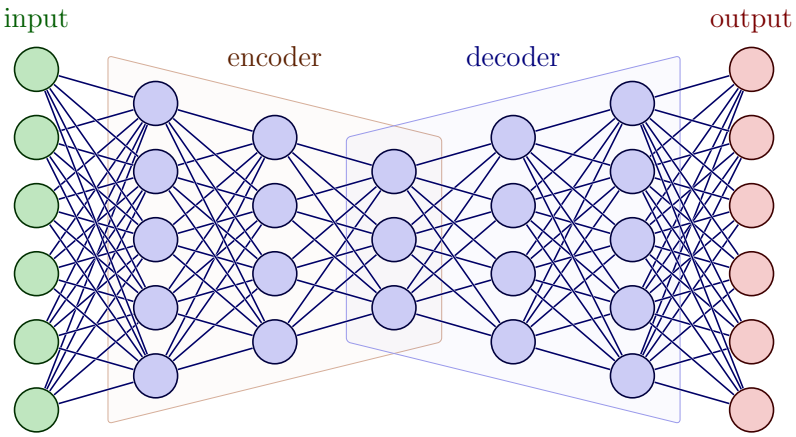

Figure 2.1: Standard Structure of an Auto-Encoder (Neutelings 2015–2025)

2.1 Deep Learning Architectures

2.1.1 Auto-Encoders

Auto-Encoders are very powerful architectures that compress data (e.g., an image) into a compact latent representation of this data. Tasks in the broad computer vision problems deal with images or video data which is more complex than a single image. Since directly processing these visual data is inherently complex, especially space-time correlation in videos, many tasks uses auto-encoders as a pre-processing stage to encode either the images or videos to obtain smaller representations, then perform other downstream tasks. We will first introduce the standard architecture of auto-encoders developed and used in computer vision problems, and its more advanced versions which can learn structured latent representations by using regularization techniques.

2.1.1.1 Standard Auto-Encoders

Auto-Encoder was first introduced in the mid-80's, a neural network to reconstruct its input and learn a latent representation of the input data (Rumelhart et al. 1985) as shown in Figure 2.1, which formalised later by Baldi (2012). With the success of Convolutional Neural Network (CNN) (Krizhevsky et al. 2012) and its reverse De-Convolutiontal Neural

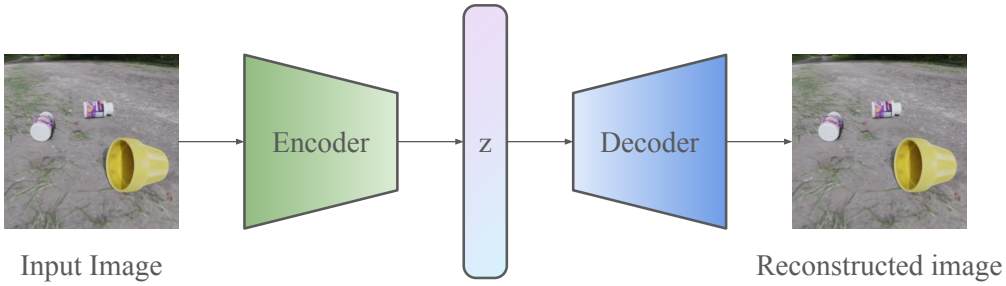

Figure 2.2: Image AutoEncoder

Network (Pu et al. 2016), auto-encoders can successfully encode images into a compact latent representation. Many well-known CNN-based architectures, such as family of ResNet (He et al. 2016) and VGG models (Simonyan and Zisserman 2014) are used as the backbone structure, compared to vanilla CNNs, to improve the quality of reconstructed images. Furthermore, with the recent introduction of Vision Transformers (ViT) (Dosovitskiy et al. 2021), there are also approaches which adopted ViT to encode the images into a latent space such as Masked-AutoEncoder (He et al. 2022), which can reconstruct partially masked images into a full image.

This process can be formally noted as follows; first, an image x is encoded by an encoder Φ to produce a latent representation z:

$$z = \Phi(x) \tag{2.1}$$

Then, z is passed to a decoder Ψ to reconstruct the original image,

$$\hat{x} = \Psi(z) \tag{2.2}$$

The objective of this network is to minimize the reconstruction loss between the original image x and reconstructed image $\hat{x}$ with Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{reconstruction}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{2.3}$$

where x_i is the pixel value in the original image, $\hat{x}_i$ is the pixel value in the reconstructed image, N is the total number of pixels in the image.

Although standard auto-encoders can encode and reconstruct high-quality images, the latent representations learned by this type of models are usually poor without meaningful structure. This is because the latent space itself is not constrained by regularizing terms or conditions. Therefore, it is difficult to perform other downstream tasks by using an unstructured latent space.

2.1.1.2 Regularized Auto-Encoders

To make the latent space more structured, both continuous and discrete regularization methods are used to mitigate the limitation of unstructured latent space and made possible to sample new data, which in our case images. A representative model of a regularized auto-encoder is Variational Auto-Encoder (VAE) (Kingma and Welling 2013). The main difference compared to standard auto-encoder is that VAE assumes the image distribution is Gaussian distribution and each image is a sample from this Gaussian distribution. Thus, instead of producing a single latent representation, the encoding network of VAE produces two latent representations to represent the mean μ and the variance σ of a Gaussian distribution. To obtain the latent representation z, the VAE applies the reparameterization trick, which allows for differentiable sampling. This trick expresses the latent variable z as a function of the mean μ, the standard deviation σ, and a random noise variable $\epsilon \sim \mathcal{N}(0, I)$ sampled from a standard normal distribution. The latent variable z is then obtained as:

$$z = \mu + \sigma \cdot \epsilon \tag{2.4}$$

Then, this latent representation is decoded by a decoder to reconstruct the original image. The network is trained to minimize the distance between the standard Gaussian distribution (e.g., zero mean and identity variance, $\mathcal{N}(0, I)$) and the predicted distribution using Kullback-Leibler divergence (KL Divergence). The Kullback-Leibler divergence between the predicted distribution $q(z|x)$ (parameterized by the mean μ and standard deviation

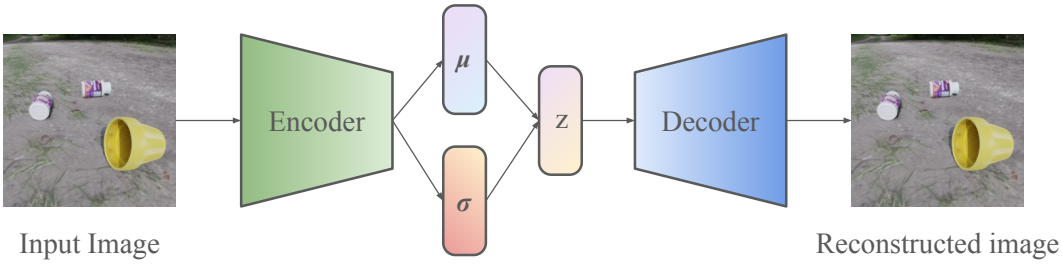

Figure 2.3: Variational Auto-Encoder

σ) and the standard normal distribution $\mathcal{N}(0, I)$ is given by:

$$\mathcal{D}_{\mathrm{KL}}\left(q(z|x)\,||\,\mathcal{N}(0,I)\right) = \frac{1}{2}\sum_{j=1}^{D}\left(\sigma_j^2 + \mu_j^2 - \log(\sigma_j^2) - 1\right) \tag{2.5}$$

where μ_j and σ_j are the mean and standard deviation for each dimension j of the latent variable z, D is the dimensionality of the latent space. The reconstruction performance using negative log-likelihood, which can be represented by the mean squared error (MSE) loss.

$$\mathcal{L}_{\text{reconstruction}} = \mathbb{E}_{q(z|x)}\left[-\log p(x|z)\right] \approx \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2 \tag{2.6}$$

These two terms together forms the evidence lower bound (ELBO), which is the objective function of the network as follows:

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q(z|x)}\left[-\log p(x|z)\right] + \mathcal{D}_{\mathrm{KL}}\left(q(z|x)\,||\,p(z)\right) \tag{2.7}$$

By minimizing the ELBO, the VAE optimizes both the reconstruction accuracy and the structure of the latent space, ensuring that it can generate new, realistic samples by sampling from the latent space.

In this way, the image distribution is mapped to the Gaussian distribution, and it can be used to sample new images. However, VAE still has limitations due to its loss function. It usually produces blurry reconstructions and entanglement of latent representations. Therefore, the main idea of generative adversarial networks (Goodfellow et al. 2014) (GAN) is also used to improve reconstruction performance with joint training of Auto-Encoder network and a discriminator network, such as Adversarial Auto-Encoders (AAE) (Makhzani et al. 2015) and Dist-GAN (Tran et al. 2018).

In contrast to continuous methods, which force the latent representation into a continuous latent space, there are approaches that utilize a set of discrete tokens to represent the latent space. The most representative model of this category is the Vector-Quantized Variational Auto-Encoder (VQ-VAE) (Oord et al. 2017) and its more advanced variant, VQ-VAE-2 (Razavi et al. 2019), as shown in Figure 2.4. Instead of enforcing a prior distribution on the latent representation, VQ-VAE replaces the continuous encoder output with discrete tokens selected from a predefined codebook. This process is referred to as quantization of the latent space. To perform quantization, a codebook $E = \{e_k\}_{k=1}^{K}$ is defined as a learnable embedding matrix, where K denotes the number of discrete codes and each code $e_k \in \mathbb{R}^D$ has the same dimensionality as the encoder output. Given the encoder output z_e, the quantized latent representation z_q is obtained by selecting the closest code in the codebook using nearest-neighbor search:

$$z_q = e_{k^*}, \quad k^* = \arg\min_k \|z_e - e_k\|_2 \tag{2.8}$$

The quantized latent representation z_q is then passed to the decoder to reconstruct the input. Since the quantization operation is non-differentiable, VQ-VAE employs the straight-through estimator to allow gradients to flow from the decoder to the encoder during backpropagation. The training objective of VQ-VAE consists of three components: reconstruction loss which is same as used in the previous auto-encoders, vector quantization loss, and commitment loss. The vector quantization loss updates the codebook embeddings

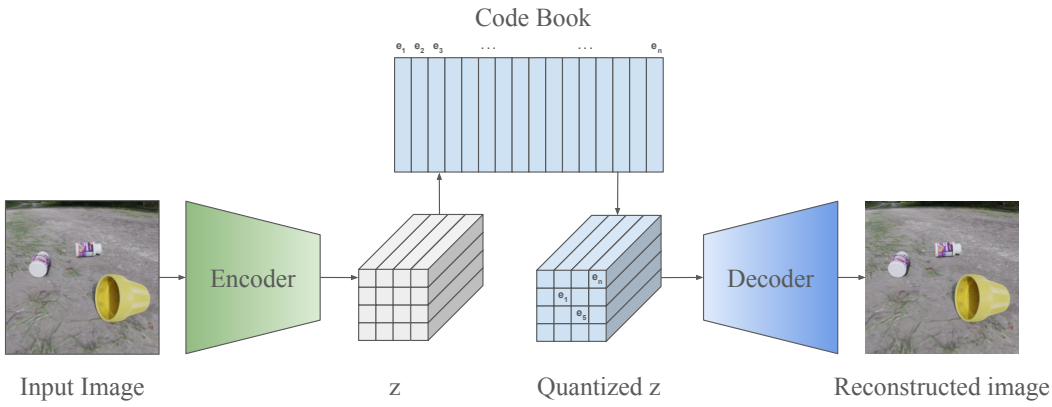

Figure 2.4: Vector-Quantised Variational AutoEncoder

to move closer to the encoder outputs:

$$\mathcal{L}_{\text{VQ}} = \|\text{sg}\left[z_e(x)\right] - e\|_2^2 \tag{2.9}$$

where sg[·] denotes the stop-gradient operator. The commitment loss prevents the encoder outputs from fluctuating excessively and encourages them to commit to a selected code:

$$\mathcal{L}_{\text{commit}} = \beta \left\|z_e(x) - \text{sg}[e]\right\|_2^2 \tag{2.10}$$

where β is a hyperparameter controlling the strength of the commitment loss. The final training objective is given by:

$$\mathcal{L}_{\text{VQ-VAE}} = \mathcal{L}_{\text{reconstruction}} + \mathcal{L}_{\text{VQ}} + \mathcal{L}_{\text{commit}} \tag{2.11}$$

By using a discrete latent space, VQ-VAE avoids posterior collapse and enables learning a compact and interpretable latent representation, which is particularly suitable for high-quality image generation and downstream autoregressive modeling. Therefore, generative models like transformers can use the trained codebook as a dictionary to generate new high quality images (Esser et al. 2021).

In this thesis, because of VQ-VAE's superior performance of producing compact latent space and the high reconstruction quality of the input image, we use VQ-VAE as our video frame encoder in each technical chapter with reasonable adjustments of its structure to meet the needs of a specific chapter.

2.1.2 Transformers

In sequential data modelling, Recurrent Neural Networks (RNN) (Williams and Zipser 1989), and their more sophisticated versions like Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Unit (GRU) (Cho et al. 2014) are used to learn sequential patterns. Since videos are also a type of sequential data, with the combination of CNNs, ConvLSTM networks (Shi et al. 2015) are used to handle video data. However, RNNs based approaches has inherent limitations such as the vanishing gradient problem when a sequence to process is too long, and the sequential computation makes them unable to parallelize.

To mitigate these limitations of RNN, the introduction of the transformer architecture revolutionized the way of modeling sequential data (Vaswani et al. 2017). In this section, we will first introduce the basics of a standard transformer networks and the transformers that is adapted to computer vision tasks.

Although the Extended Long Short-Term Memory (xLSTM) Beck et al. 2024 introduced recently showed superior performance regarding popular transformer-based Large Language Models (LLMs) like Llama (Touvron et al. 2023), they state that xLSTMs are still four times slower than a transformer architecture with similar capacity due to lack of CUDA kernel optimizations for computation efficiency. Therefore, throughout this thesis, we use transformer architecture as our backbone structure due to its stability and its well established performance across different domains. Because our task is to predict future

frames based on observed past frames, we use a decoder only transformer in both Chapter 3 and Chapter 4 and in Chapter 5 we use a transformer-based diffusion model to tackle deformable objects' motion prediction. The diffusion model will be introduced in detail in the Section 2.1.3.

2.1.2.1 Standard Transformers

Transformer architecture was first proposed for Machine Translation (MT) problem in Natural Language Processing (NLP) as shown in Figure 2.5. It consists of an encoder network that learns the patterns of an input sequence with multi-head self-attention, and a decoder that predicts the target sequence with masked (which prevents the model access to the future tokens) multi-head cross-attention between the information from the encoder and the output from its own self-attention. Then the prediction is performed in a causal and auto-regressive manner. Because of the attention mechanism, all of the tokens (e.g., words in language) relations are computed in parallel and without the need of sequential processing. Because transformer model does not process the tokens sequentially, in order to impose the positional relationship to the model, the original transformer model used sine-cosine positional embeddings to let the input sequence have the positional information. This parallelized process made possible to handle long sequences without the risk of vanishing gradients. However, this pair-wise attention computation between tokens makes the architecture computationally heavy.

Because different parts of a transformer play different roles, different transformer variants are introduced to handle different tasks. Encoder-only transformers are usually used to pre-train on a large sequential dataset to learn the over all pattern of the sequences, such as BERT (Devlin et al. 2019). In contrast, because the decoder in transformers is limited to calculate the attention only between the previous tokens with current tokens,

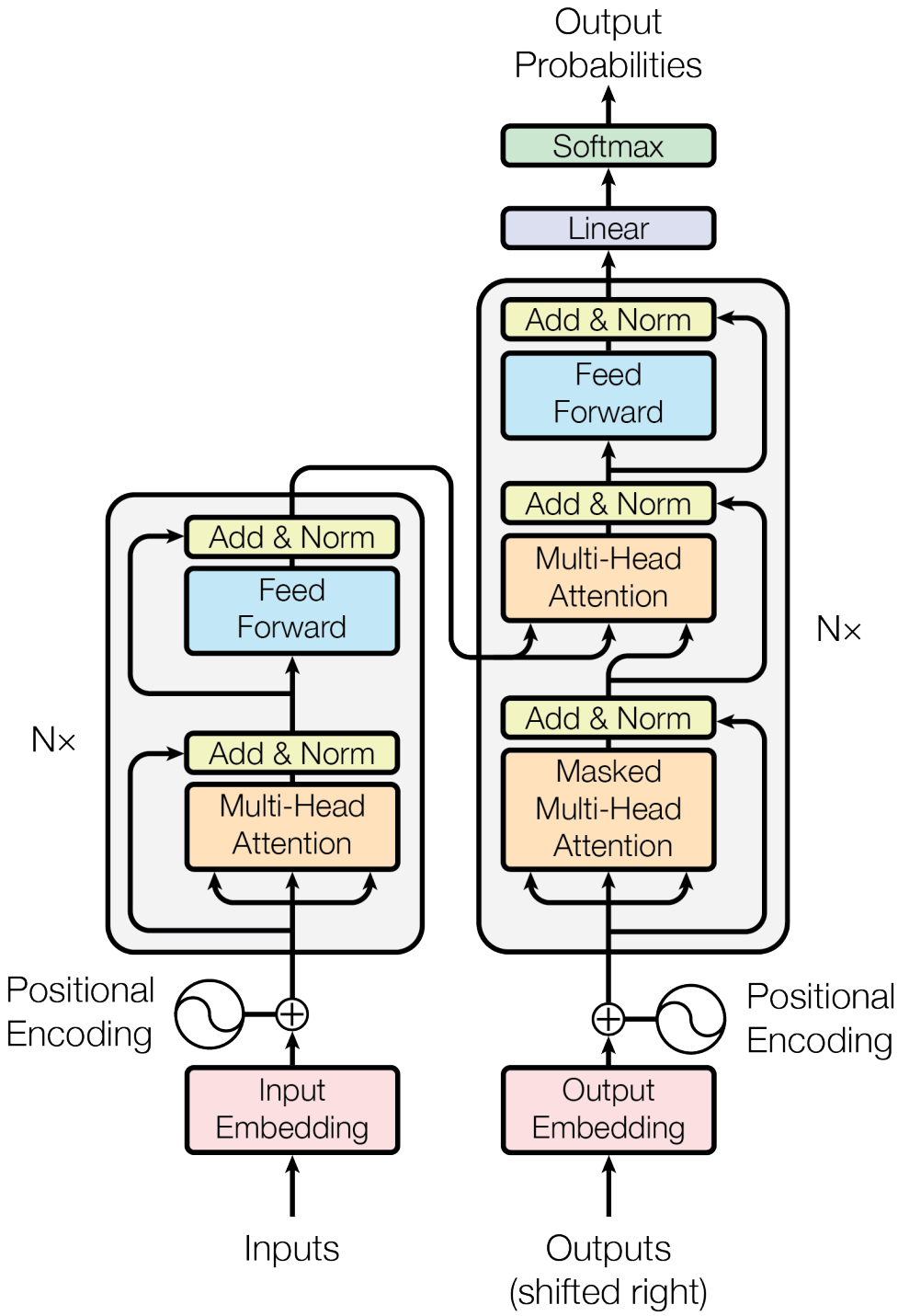

Figure 2.5: The original transformer architecture (Figure is reproduced from (Vaswani et al. 2017))

decoder-only transformers are usually used to generate new tokens (Radford et al. 2018). For example, all modern chatbots like ChatGPT (Achiam et al. 2023), Claude (Anthropic 2025), Gemini (Comanici et al. 2025) and Grok (xAI 2025) use this architecture as their backbone structure.

2.1.2.2 Vision Transformers

Transformer's success is also well adopted by computer vision tasks. The first vision transformer (ViT) was introduced to tackle the image classification problem (Dosovitskiy et al. 2021). Unlike languages, images consist of pixels and are not sequential data, but transformer networks takes sequence of tokens as its input, therefore, ViT's authors introduced a tokenization approach of images that images are segmented to smaller patches, and each patch of pixels are considered as tokens. ViT's network structure is shown in Figure 2.6

Although this patching technique provided a strong baseline for vision tasks, predefined sizes of patches limits its performance on fine-grained local and overall global relationships between patches. Therefore, a more adaptive ViT , Swin Transformer (Liu et al. 2021) is introduced to mitigate this problem. It changes the resolution of patches by combining or dividing adjacent patches gradually by the depth of blocks to mimic the down-sampling process of CNNs architectures, and achieved superior performance in image classification task.

This idea of vision transformer is then used in many different research areas in computer vision, such as semantic segmentation (Zhang et al. 2022, 2024a), unsupervised image representation learning (Caron et al. 2021; Oquab et al. 2023; Siméoni et al. 2025), object detection (Carion et al. 2020), and auto-encoding (He et al. 2022).

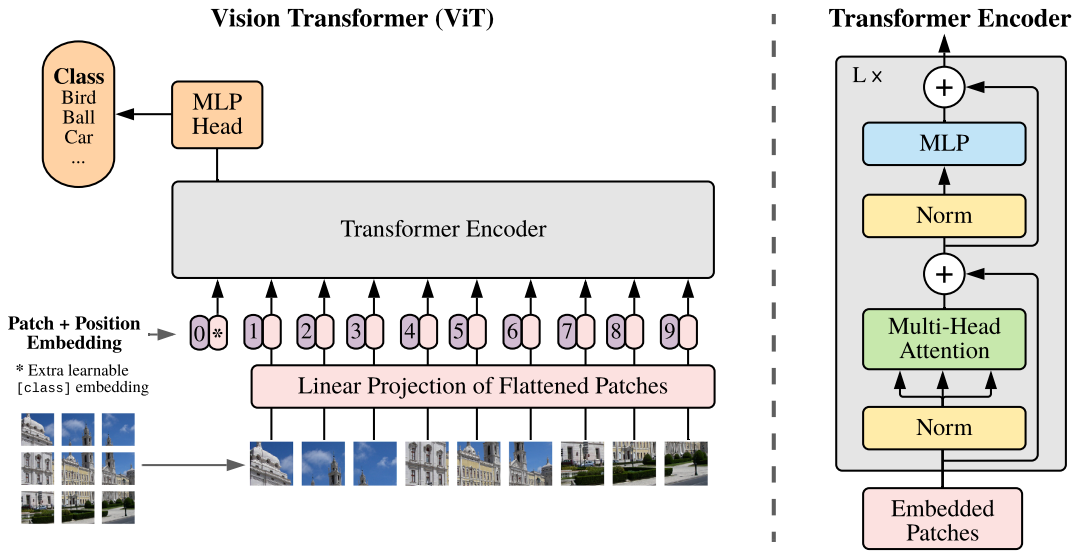

Figure 2.6: Original ViT Architecture (Figure is reproduced from (Dosovitskiy et al. 2021)

Extension to image transformers, the video vision transformer (ViViT) is introduced to perform the video classification task (Arnab et al. 2021). It introduced two different patching approaches. One is to follow the patching process in the ViT and it is applied to each frame to obtain the patches for the video. Another is to patch the same area of adjacent video frames together, so that enabling a patch to have spatiotemporal information. Many other works extended the original images transformers from previous works to video domain. For example, reconstructing the masked patches of a video (Tong et al. 2022; Wang et al. 2023) following MAE; Video Swin Transformer (Liu et al. 2022) is later introduced by following the idea of the original Swin transformer for images.

2.1.3 Diffusion Models

Diffusion is a process that is formally defined in thermodynamics. It refers to spontaneous movements of particles such as atoms, molecules or ions from high- to low-concentration region. For example, if a blue dye dropped into a glass of clear water, the clear water will eventually become blue with the passage of time. Therefore, this process increases the entropy of a system until concentration is equalized throughout the system.

In recent years, the concept of diffusion has been adapted into the field of machine learning, particularly for generative modeling. Inspired by its thermodynamic interpretation, diffusion-based models simulate data generation as the reversal of a gradual noising process. Drawing further motivation from techniques used to transform one probability distribution into another via non-equilibrium Monte Carlo methods (Jarzynski 1997; Neal 2001), Sohl-Dickstein et al. (2015) introduced a framework in which data is incrementally corrupted by Gaussian noise through a forward diffusion process. A neural network is then trained to approximate the reverse process, effectively learning to denoise and reconstruct the original data distribution from pure noise.

2.1.3.1 U-Net Based Diffusion Models

The diffusion framework was further refined by Ho et al. (2020), who introduced the Denoising Diffusion Probabilistic Model (DDPM) to address the image generation problem. Through extensive experiments, the authors demonstrated that DDPM could surpass the image quality of state-of-the-art generative adversarial networks (GANs) of the time, such as the StyleGAN family (Karras et al. 2019, 2020). DDPM employs a U-Net architecture (Ronneberger et al. 2015) as shown in Figure 2.7, originally designed for biomedical image segmentation, as a noise prediction network. This architecture is particularly effective at capturing multi-scale image features, which is crucial for high-quality denoising during the reverse diffusion process.

In DDPM, the task is to learn to map a complex distribution (e.g., images) to a Gaussian distribution, so the model can be used to generate new images from Gaussian noise. The training of a diffusion model involves a forward and reverse diffusion process. In the forward process, a data from a complex distribution is slowly transformed to a pure Gaussian noise by a noise scheduler. Given an original image x_0, and the noise ϵ sampled from Gaussian distribution $\mathcal{N} \sim (0, I)$, with total time step T, we can get the noised image x_t at every time step t from the previous image x_{t-1} at time step $t-1$. This can be represented as a conditional probability distribution of x_t given x_{t-1} as the equation

2.12.

$$q(x_t \mid x_{t-1}) \;=\; \mathcal{N}\Big(x_t\,;\,\sqrt{1-\beta_t}\,x_{t-1},\,\beta_t\mathbf{I}\Big) \tag{2.12}$$

we can write it in equation form as:

$$x_t = \sqrt{\beta_t - 1}x_{t-1} + \sqrt{\beta_t}\epsilon_t \tag{2.13}$$

where β_t is the scheduled amount of noise to be added at time step t. For simplicity, we can represent β_t as:

$$\alpha_t = 1 - \beta_t \tag{2.14}$$

So that:

$$x_t \;=\; \sqrt{\alpha_t}\,x_{t-1} \;+\; \sqrt{1-\alpha_t}\,\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0,\mathbf{I}). \tag{2.15}$$

This shows a single step of a diffusion process. Because x_t is obtained only depending on the previous time step, so this is a Monte Carlo process. Therefore, we can get the noised image x_t at every time step t with the original image x_0 as follows,

$$x_t = \sqrt{\alpha_t\alpha_{t-1}}\,x_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\,\epsilon_{t-1} + \sqrt{1-\alpha_t}\,\epsilon_t \tag{2.16}$$

$$\vdots$$

$$= \sqrt{\bar{\alpha}_t}\,x_0 + \sum_{s=1}^{t}\sqrt{(1-\alpha_s)\prod_{j=s+1}^{t}\alpha_j}\;\epsilon_s. \tag{2.17}$$

Where $\bar{\alpha_t}$ is,

$$\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s \tag{2.18}$$

Based on the properties of the Gaussian distribution, we can derive the mean and the covariance,

$$\mathbb{E}[x_t \mid x_0] \;=\; \sqrt{\bar{\alpha}_t}\,x_0, \qquad \mathrm{Cov}[x_t \mid x_0] \;=\; \left(\sum_{s=1}^{t}(1-\alpha_s)\prod_{j=s+1}^{t}\alpha_j\right)\mathbf{I}. \tag{2.19}$$

Simplifying the covariance as,

$$\sum_{s=1}^{t}(1-\alpha_s)\prod_{j=s+1}^{t}\alpha_j \;=\; 1 - \prod_{j=1}^{t}\alpha_j \;=\; 1-\bar{\alpha}_t. \tag{2.20}$$

then the final Gaussian form is represented as,

$$q(x_t \mid x_0) \;=\; \mathcal{N}\big(x_t\,;\,\sqrt{\bar{\alpha}_t}\,x_0,\,(1-\bar{\alpha}_t)\mathbf{I}\big)\,. \tag{2.21}$$

Because the amount of noise is known at every time step, similarly, this formulation can be used to revert a pure noise sampled from Gaussian distribution back to the data distribution.

In the original DDPM formulation, the forward noising process is controlled by a linear variance schedule over $T = 1000$ diffusion steps, where the noise variance β_t increases linearly from $\beta_0 = 10^{-4}$ at $t = 0$ to $\beta_T = 0.02$ at $t = T$. In contrast, Diffusion Transformers (DiT) (Peebles and Xie 2023) employ a cosine variance schedule. In chapter 5, we will follow DDPM and use a linear scheduler with total time step $T = 1000$.

DDPM is trained to take a noised image x_t as input and predict the noise ϵ that was added to the original image x_0. This formulation allows the model to iteratively denoise a Gaussian noise sample during generation, ultimately producing a realistic sample from pure noise. The training objective of diffusion models is derived from maximizing the evidence lower bound (ELBO) on the log-likelihood of the data distribution $q(x_0)$. Let $q(x_{1:T} \mid x_0)$ denote the forward diffusion process and $p_\theta(x_{0:T})$ the learned reverse process. The variational lower bound can be written as,

$$\log p_\theta(x_0) \;\geq\; \mathbb{E}_q\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right] \;=:\; \mathcal{L}_{\text{ELBO}} \tag{2.22}$$

This objective can be decomposed into a sum of KL divergence terms across time steps. Ho et al. (2020) show that, with appropriate parameterization of the reverse process, all terms reduce to simple forms except the one corresponding to the Gaussian mean prediction. This leads to a simplified training loss of the form

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right], \tag{2.23}$$

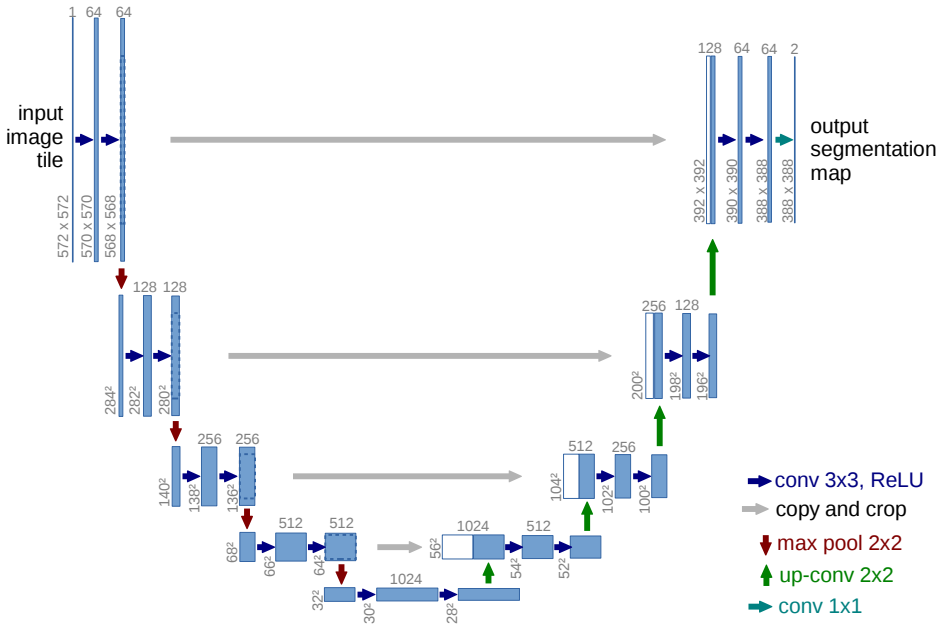

Figure 2.7: Original U-Net structure Ronneberger et al. 2015

Thus, the model is trained to predict the added noise ϵ, which implicitly corresponds to maximizing the ELBO on data likelihood.

However, because DDPM operates directly in the high-dimensional image space, inference is computationally expensive and slow. To address this limitation, the Latent Diffusion Model (LDM) was introduced (Rombach et al. 2022). Instead of performing diffusion on pixel-level data, LDM operates in a compressed latent space learned by a VAE. This significantly reduces the computational burden and speeds up image generation, but the generation quality will be depending on the reconstruction quality of the VAE. Another way to speed up the inference time is to use Denoising Diffusion Implicit Model (DDIM) (Song et al. 2020). Unlike Markovian sampling process of DDPM, it uses deterministic sampling process to make inference time faster without losing the generation quality.

In Chapter 5, we adopt the standard diffusion formulation with a slight modification: instead of operating in pixel space, our model predicts future frames in a learned latent space. Consequently, the diffusion model is trained to predict the noise added to latent video representations, rather than directly to RGB frames.

2.1.3.2 Transformer Based Diffusion Models

While the diffusion models in the previous section rely heavily on the U-Net structure, recent diffusion models explored the integration of a transformer network as the noise predictor of a diffusion model. In Diffusion Transformer (DiT) (Peebles and Xie 2023), the U-Net backbone is replaced by pure transformer backbone. The authors demonstrate that DiT is highly scalable because the attributes of transformer network. However, with the increase of the model size (DiT-XL), it requires significantly higher GFLOPs compared to U-Net based models. Nevertheless, it achieved state-of-the-art performance compared to U-Net based diffusion models like LDM.

2.2 Video Prediction with Deep Learning

2.2.1 Recurrent models for video prediction

Early video prediction models were typically based on the combination of Convolutional Neural Networks (Krizhevsky et al. 2012) and Recurrent Neural Networks, often LSTMs (Shi et al. 2015; Wang et al. 2022, 2018; Chang et al. 2022; Gao et al. 2022; Denton and Fergus 2018). Lee et al. (2021) proposed a method to predict future semantic maps, then used those predicted maps to warp the actual future frames from the past RGB frame. Bei et al. (2021) proposed a similar approach, decomposing the scene with a semantic map, and using separate pathways to model the dynamics of different semantic classes. Of these, some methods are deterministic, i.e., make a single most-likely prediction of the future (Shi et al. 2015; Wang et al. 2018), while others are stochastic, i.e., sample an autoregressive posterior distribution on possible future frames (Denton and Fergus 2018; Lee et al. 2021). We focus on the stochastic setting in this work since the deterministic models tend to predict and converge to the mean of the possible future, as well as stochastic prediction typically producing sharper predictions (Ohayon et al. 2023).

2.2.2 Transformer models for video prediction

Following their success on text (Vaswani et al. 2017) and images (Dosovitskiy et al. 2021), Transformers have also been applied to video prediction. A common approach is to first use an encoder network to map the original video frames into a sequence of lower-dimensional latent vectors. Most models use VQ-VAE (Oord et al. 2017) or VQ-GAN (Esser et al. 2021) as their encoding network due to their high fidelity reconstruction of original frames, and discrete latent space that enables treating the latents similarly to text tokens. Yan et al. (2021) proposed the first autoregressive video prediction model based on VQ-GAN and a decoder transformer to predict future frames; iVideoGPT (Wu et al. 2024) improves the performance further. Gupta et al. (2022) proposed a similar method that uses VQ-VAE and transformer, but trains with iterative masking to let it gradually capture the motion patterns in a video. Sun et al. (2023) proposed a pipeline that decomposes the dynamic scene into motion, object and background, then uses a stochastic transformer to predict future frames in latent space. Our work also uses a latent transformer, but with an explicit decomposition of the latent space into separate objects, and cross-attention to capture object interactions.

2.2.3 Diffusion models for video prediction

The invention of diffusion models (Sohl-Dickstein et al. 2015; Ho et al. 2020) and the computationally faster latent diffusion (Rombach et al. 2022) brought significant improvement on many generative tasks. Latent diffusion was originally designed to generate high-resolution images, but has now been applied to video (Blattmann et al. 2023a,b; Brooks et al. 2024). Ho et al. (2022) use a diffusion model to generate long videos via a joint training paradigm with conditional sampling. Höppe et al. (2022) use a slightly different training process that instead of adding noise to the entire video, randomly keeps some of the input frames without noise. Yu et al. (2023) proposed an interesting way of modeling latent vectors in three directions by slicing 3D feature vectors along different

axes. Pallotta et al. (2025) proposes a very similar framework compared to the proposed method in chapter 5. It uses two modalities RGB and depth map, for each modality separate encoders are used to extract features, and a denoising latent diffusion model based on Unet is used to jointly predict both RGB and depth frames.

For more general purpose video generation models, SORA (Brooks et al. 2024) alongside with Veo3 (DeepMind 2025) is the state-of-the-art video generation model, and can generate extremely realistic videos by using diffusion with a transformer architecture. It is able to accurately generate complex interactions that involve multiple objects (Liu et al. 2024). However, in order to train these kind of models, it is extremely expensive in terms of data and computation power.

2.2.4 Object-centric video prediction

Object-centric representation learning aims to learn decomposed representations of images (Locatello et al. 2020; Engelcke et al. 2020) or videos (Jiang et al. 2019; Zhou et al. 2022) without supervision. This can be used to aid video prediction by learning an object-centric predictor (typically a transformer) over the resulting representations (Kipf et al. 2022; Li et al. 2021; Sajjadi et al. 2022; Singh et al. 2022). (Villar-Corrales et al. 2023) use an attention mechanism to learn the relationship between different objects in the video sequence and achieved good results on the synthetic CLEVRER (Yi* et al. 2020) dataset. Schmeckpeper et al. (2021) use Mask R-CNN (He et al. 2017) to get bounding boxes for each entity in the scene, then predict the next state of each bounding box from a single frame. Henderson and Lampert (2020) and Henderson et al. (2021) proposed self-supervised object-centric approaches that predict frames via latent 3D objects and scene structure from 2D video.

The major differences between this thesis and existing object-centric approaches are threefold. First, instead of learning object-centric representations directly from raw video frames, we explicitly decompose scenes using segmentation masks obtained from pre-trained models. While some prior work relies on semantic segmentation, such approaches decompose scenes into semantic categories rather than individual object instances. This distinction has not been fully explored in existing methods. By leveraging off-the-shelf instance segmentation models, we obtain controlled and reliable access to object-level information, enabling a more explicit and structured scene decomposition Second, while prior object-centric methods demonstrate promising performance, they rarely isolate and evaluate the effect of explicit object decomposition under comparable architectural and capacity settings, particularly in real-world scenarios. Third, existing works primarily emphasize the introduction of novel architectures, whereas this thesis focuses on a systematic study of why and when explicit object decomposition benefits video prediction, independent of architectural novelty.

2.2.5 Optical flow in video prediction

Optical flow is a pixel-wise dense motion estimation between consecutive video frames. FlowNet (Dosovitskiy et al. 2015) and its advanced version (Ilg et al. 2017) is first introduced to estimate the optical flow through CNN network. Recent optical flow estimation approaches used vision transformers to achieve the same goal (Shi et al. 2023; Le Moing et al. 2024; Lu et al. 2023). 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. (2018) first predict the optical flow of future frames by conditioning on a single frame, then warp the RGB frame with predicted flow to achieve video prediction. Shi et al. (2024) used a similar idea to predict the flow first and then use a diffusion model conditioned on flow to generate RGB frames. Bei et al. (2021) proposed a semantic-aware approach that predicts the optical flow directly with a ConvLSTM network, then uses the predicted flow to generate future frames. Wu et al. (2022) used optical flow to optimize the model's frame interpolation ability to improve the future frame prediction quality. Liang

et al. (2024) 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 (Chefer et al. 2025). However, error accumulation over time and the complete loss of information while objects are occluded hamper the effectiveness of optical flow methods when occlusion occurs.

2.3 Auxiliary Structures and Modalities

2.3.1 Cross-attention

Cross-Attention is first introduced with the original transformer. Its purpose is to learn the correlation between the input sequence and the sequence that needs to be predicted. Due to this efficient conditioning mechanism, this idea have been used in many other domains, e.g. Zhu et al. (2022) use pairwise cross-attention to re-identify pedestrians; Shi et al. (2025) use cross-attention to fuse information from audio and video for emotion recognition; Lee et al. (2023) use pairwise cross-attention on video action recognition; Rombach et al. (2022) uses cross attention between image features and text embeddings for conditional image generation. In this thesis, we use cross-attention to model the potential interaction between each object, and also evaluate the impact of using cross-attention to handle object interactions in a dynamic scene.

2.3.2 Point Tracking

Point tracking approaches have recently gained popularity due to their strong performance (Karaev et al. 2025; Tumanyan et al. 2024; Cho et al. 2024; Xiao et al. 2024). Unlike optical flow estimation, which aims to estimate the motion of every pixel in a pair of consecutive images, 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, Bharadhwaj et al. (2024) leveraged point tracking to assist robotic arm control in completing various tasks, achieving superior performance. Point tracking has also been applied to generative tasks. (Jeong et al. 2024) incorporated point tracking into video diffusion models, enabling more realistic motion generation. In this thesis, point tracking is used to provide explicit motion information to assist video prediction.

2.3.3 Depth Estimation

Depth estimation provides spatial information about a scene, which is the relative position and distance of a pixel to the camera. It can help to create a 3D representation of the environment, and this representation can be used on many downstream tasks. Cetinkaya et al. (2022) used depth estimation to perform object detection. Chan et al. (2022) used estimated depth map to create a tri-plane that can reconstruct a 3D scene from a single image. Xiao et al. (2024) extended this idea to track keypoints in a clip in 3D space. In Chapter 4, we will investigate the benefits of using a depth map on the task of video prediction, especially in occluded events where 3D geometry information is crucial for models to learn the occlusion of objects.

2.3.4 Multi-Modal Fusion

Multi-modal fusion, in computer vision, refers to combining different modalities captured by different sensors such as RGB images from digital cameras, depth maps from RGB-D cameras, and point clouds from LiDAR scanners, into a single representation that has all of the characteristics of these modalities. This enables many downstream tasks as 3D

object detection (Chen et al. 2017; Bai et al. 2022; Li et al. 2022) and segmentation (Hazirbas et al. 2016; Guan et al. 2025), 3D scene reconstruction (Azinović et al. 2022), and point tracking (Karaev et al. 2025; Ngo et al. 2025). In chapter 5, we use ideas from multi-modal fusion architectures to integrate different modalities in order to improve the frame reconstruction quality.

Chapter 3

On the Benefits of Instance Decomposition in Video Prediction Models

In the previous chapters, we introduced the task of video prediction, its applications across domains such as robotics and autonomous driving, and the main research gaps motivating this thesis. Besides, we introduced necessary background knowledge such as Auto-Encoders, Transformers and Diffusion models. We also outlined three central research proposals, each designed to address specific limitations of current video prediction models.

In this chapter, we present the first contribution of this thesis: a video prediction framework that explicitly models scenes in an object-centric manner. The goal of this model is to test our first hypothesis that explicit instance decomposition improves the quality of video predictions by enabling per-object motion modeling. We develop and evaluate a transformer-based architecture that encodes and predicts individual object motions, and we assess its benefits over standard non-object-centric models.

The chapter is structured as follows:

- Section 3.1 provides a brief introduction to general video prediction methods and prior work on object-centric predictors;

- Section 3.2 describes the proposed architecture in detail;
- Section 3.3 outlines our experimental setup, implementation details, evaluation metrics, and discusses the results;
- Section 3.4 and 3.5 summarizes the findings and discusses limitations of the proposed method.

3.1 Introduction

Predicting future frames is challenging, since images are high-dimensional and result from the combination of multiple objects' appearances, dynamics and mutual interactions. For example, consider the environment observed while driving a car. How this scene will de-

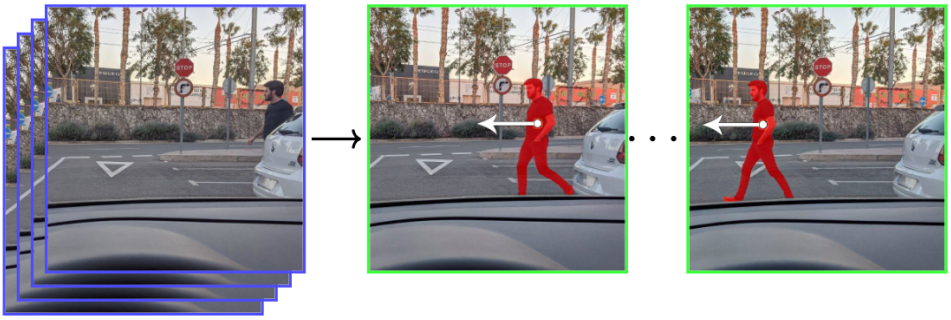

Figure 3.1: Typical scenario of a video prediction task: While we drive a car and want to drive through a cross-road, after we observe a certain period of the past (blue frames) which we see a pedestrian is trying to cross, we will anticipate the future motion (green frames) of this pedestrian and slowdown our car (Oprea et al. 2020).

velop in the immediate future is dependent on critical elements in the scene (e.g., cars, pedestrians, dogs) and their individual pattern of movement, including complex interactions with both static and moving parts of the scene (e.g., a car stopping at a traffic light or a dog following its owner on a leash). Hence, the complexity of the frame prediction task rises quickly as more objects with different motions interact in a scene, and with this, the size and training data required by prediction models.

3.1.1 Joint Vs. Decomposed Modeling of Dynamic Scene

State-of-the-art general video prediction and generation models such as Veo3 (DeepMind 2025) and Sora (Brooks et al. 2024) tend to model the dynamics of a scene jointly with the availability of large-scale datasets and computation power (e.g., High Performance GPUs and CPUs). Although these models achieve remarkable performance, they are not efficient because of the reliance on these resources, which makes them suboptimal. Such big problems can usually be dealt with by decomposing them into smaller pieces. In the case of video prediction, because a dynamic scene often consists of different objects and their pattern of motion, the logic of decomposition is also applicable to video prediction task for more efficient modeling with limited resources. The decomposition enables modeling the appearance and dynamics of each part separately during prediction, thus reducing computational cost and increasing statistical efficiency.

3.1.2 Decomposition Approaches

There are mainly two different ways of decomposing a scene (i.e., an image), implicit and explicit decomposition. **Implicit Decomposition** means the scene is decomposed without any external knowledge such as any form of labels, for example a segmentation map. These approaches use unsupervised learning methods to decompose the scene into individual objects by learning and categorizing similar features. For example, Hsieh et al. (2018) uses DRNet (Denton et al. 2017) to learn a disentangled representation of appearance and 2D pose implicitly with a structured pose and appearance representation. Wu et al. (2023) uses object-centric representation learning (Locatello et al. 2020) to separate objects without supervision, and model the dynamics with a multi-slot transformer. However, training unsupervised learning models also tends to need a well balanced and large dataset, otherwise these kind of models will fail to capture the different semantic information.

Explicit Decomposition uses available label or a pre-trained semantic or panoptic segmentation model when labeled data is unavailable to achieve object decomposition in a scene. For instance, Bei et al. (2021) and Lee et al. (2021) use semantic segmentation models to generate a segmentation map, then predict the future segmentation map; Finally the initial image is warped according to predicted segmentation maps. Compared to implicit decomposition, the latter is more efficient and robust, because there are many ready and available off-the-shelf models already trained on large-scale datasets, for example, family of YOLO (Reis et al. 2023) models and more modern segmentation models: SAM family (Kirillov et al. 2023; Ravi et al. 2024).

While some existing object-centric video prediction approaches we mentioned previously achieved impressive results compared to agnostic video prediction models, they do not focus on measuring the benefits of object decomposition for video prediction models in a scientifically controlled way, i.e., keeping confounding factors such as the number of network parameters, architecture or latent dimensionality constant. Moreover, some of these works (Gao et al. 2022; Wang et al. 2022, 2018) did not use the modern large latent-space Transformer architectures (Vaswani et al. 2017) that now yield excellent results on diverse domains of videos (Yan et al. 2021; Wu et al. 2024); they instead used older, smaller CNN- or RNN-based models.

To fill this gap, we perform a detailed study of the benefits of explicit modeling of separate objects' motions during video prediction, using modern latent transformer models. Rather than introducing an entirely new model, we develop a family of architectures similar to VideoGPT, MOSO and Slotformer (Yan et al. 2021; Sun et al. 2023; Wu et al. 2023), that supports both single-slot (i.e. jointly modeling the whole scene) and multi-slot (i.e. per-object) representations in a unified framework. This allows us to perform controlled experiments on the benefits of object decomposition and on strategies for modeling interactions. Specifically, we adopt a hierarchical approach that explicitly decomposes a dynamic scene into individual objects using an instance segmentation model, before encoding these into separate latent spaces. We assume objects of the same class will have

similar motion patterns, for example different cars or different pedestrians; therefore, we will use same slot (e.g., sharing parameters) across all instances of each class. Nevertheless, each instance can still be modeled separately, but it is inefficient and computationally expensive.

Our main contributions are as follows:

- We present the first systematic and comprehensive analysis of the benefits of explicit object decomposition for latent transformer video prediction models.
- To achieve this, we develop a scalable framework for video prediction that supports both the single- and multi-slot settings.
- We mitigate statistical inefficiencies in object-centric video predictors by sharing weights (and thus knowledge about object dynamics) across slots within each object class.

3.2 Methodology

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 M future frames $X^{T+1:T+M}$, conditioned on the T past frames $X^{1:T}$.

We hypothesise that predicting future frames is more effective when modeling each object or instance separately rather than modeling the entire scene at once. Moreover, when objects are decomposed, we aim to measure the degree to which cross-attention enables learning interactions among objects, thus making prediction more accurate.

To test this hypothesis, we designed a family of models that support differing degrees of object decomposition and interaction within a unified framework. We decompose a scene into individual objects using instance segmentation models (Reis et al. 2023; Lüddecke and Ecker 2022). The video prediction models then comprise an *object-aware auto-encoder*

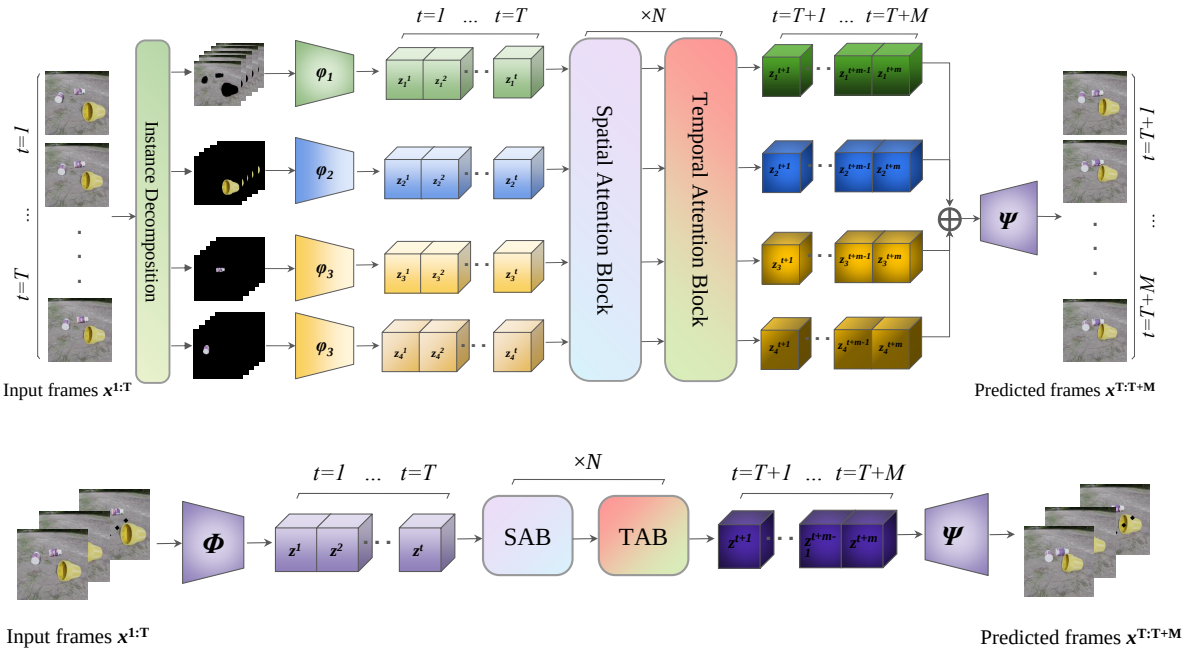

Figure 3.2: **Top:** Our proposed multi-object interacting model **SCAT**. First, the input frames are decomposed via a segmentation model, then each decomposed sequence passes through class-specific encoder to convert the 2D frames into latent representations; then, class-specific transformer blocks learn and predict the dynamics of each instance and its relationships with other instances in latent space; lastly, the predicted latent representation are decoded via joint decoder to reconstruct the predicted RGB frames. **Bottom:** The non-decomposed single-slot variant **SiS** where the scene is modeled globally and jointly.

(OAAE) (Section 3.2.1), which extracts latent representations for each object, and a multi-object transformer (Section 3.2.2) that predicts future latent representations conditioned on previous ones; the OAAE is used to decode these future latents back into video frames. To test our hypotheses, we propose three variants of our overall pipeline:

- **Single Slot (SiS)**: Objects are not modeled separately; frames are encoded with a single encoder, and a standard (not object-centric) transformer network is used to predict future frames; this is similar to VideoGPT (Yan et al. 2021).
- **Stochastic non-Class Attended Transformer (SNCAT)**: The scene is decomposed into instances; both the encoder and predictor have one slot for each object in the scene, with parameters shared across instances of the same class, but no interactions among different object slots in the transformer.

- **Stochastic Class Attended Transformer (SCAT)**: Our full model, which encodes instances separately, then uses a multi-slot transformer for future prediction, with cross-attention to capture object interactions.

The overall pipeline of the fully-interacting decomposed **SCAT** and single slot **SiS** models is shown in Figure 3.2.

3.2.1 Object-aware autoencoder

We now discuss the encoder we use for extracting the latent representation of a video, which will be used in Section 3.2.2 as a lower-dimensional space for future prediction. We first explain the object-aware autoencoder (OAAE) as used in the **SCAT** and **SNCAT** models, then give a brief explanation of the simpler (non-object-centric) variant used in **SiS**.

3.2.1.1 Instance decomposition

Let $x \in \mathbb{R}^{h \times w \times 3}$ be a frame in an RGB video sequence of width w and height h. It is decomposed into a set of N instances with corresponding class labels using an off-the-shelf segmentation model (Reis et al. 2023; Lüddecke and Ecker 2022). The segmentation returns N non-overlapping binary masks, each belonging to one of m object classes $c_k \in \{1, \ldots, m\}$; we then multiply the input frame by the respective masks to isolate each object. The k^{th} masked instance is denoted by $\tilde{x}_k$ for $k \in \{1, 2, \ldots, N\}$, and its class is denoted as c_k. Assuming the segmentation is panoptic and covers all pixels of the frame, the original frame can be reconstructed by recombining all instances of all classes additively as follows:

$$x = \sum_{k=1}^{N} \tilde{x}_k \tag{3.1}$$

3.2.1.2 Instance embedding

We modify the standard VQ-VAE (Oord et al. 2017) model to have a set of encoders $\Phi = \{\phi_1, \phi_2, ..., \phi_m\}$ and a set of embedding code books $E = \{e_1, e_2, ..., e_m\}$, each associated with an individual semantic class. Each instance frame $\tilde{x}_k$ is passed to the corresponding encoder ϕ_{c_k} and quantized with e_{c_k} to produce a latent vector $\tilde{z}_k$:

$$\tilde{z}_k = e^i_{c_k} \text{ where } i = \arg\min_j(\|\phi_{c_k}(\tilde{x}_k) - e^j_{c_k}\|_2) \tag{3.2}$$

The quantized representations are then concatenated into a single vector $z = \bigoplus_{k=1}^{N} \tilde{z}_k$ that encodes the complete frame x (where $\bigoplus$ denotes concatenation operation).

For convenience, we will use the notation $z = \Phi(x)$ to denote the overall encoding operation. This latent representation z can then be passed to a single joint decoder Ψ to reconstruct the full frame, i.e., $\hat{x} = \Psi(z)$. After each up-sampling convolutional layer in the decoder, we incorporate Frequency Complement Modules (FCM) (Lin et al. 2023) to learn not only from the target frame but also from feature maps between encoder and decoder. The FCM module consist of batch normalization layer, ReLU activation, Dropout layer and finally a 2-dimensional Convolutional layer.

3.2.1.3 Loss function

Since our OAAE is a multi-object extended version of the original VQ-VAE (Oord et al. 2017) with some features of FA-VAE (Lin et al. 2023), we also extend the original loss functions correspondingly. There are 4 losses: feature loss, commitment loss, vector quantization loss (VQ loss) and reconstruction loss. Following Lin et al. (ibid.), we impose loss on feature maps, not only on the final pixels; similarly to them, we use focal frequency loss (FFL (Jiang et al. 2021)) between the output of encoder convolution layers and decoder FCM layers:

$$\mathcal{L}_{feature} = \sum_{c=1}^{m} \sum_{l=0}^{L-1} FFL(f^c_l, g_{L-l}) \tag{3.3}$$

where c indexes encoders (recall there is one per class), l indexes over convolutional layers in the c^{th} encoder and $L-l$ over corresponding FCM layers in the decoder (L is the total number of decoder layers), f_l represents the feature map of the l^{th} encoder layer, and g_l that of the l^{th} FCM module in the decoder. The VQ and commitment losses are similar to the original VQ-VAE, except we compute these for each class c and instance k, then sum over these:

$$\mathcal{L}_{VQ} = \sum_{k=1}^{N} \|sg[\phi_{c_k}(\tilde{x}_k)] - e_{c_k}\|^2 \tag{3.4}$$

$$\mathcal{L}_{commitment} = \sum_{k=1}^{N} \|\phi_{c_k}(\tilde{x}_k) - \text{sg}[e_{c_k}]\|_2^2 \tag{3.5}$$

where sg is the stop-gradient operator. Finally, the reconstruction loss is composed of pixel-space and frequency-space terms calculated between the reconstructed and original frames:

$$\mathcal{L}_{recon} = -\log p(x|\Psi(\Phi(x))) + FFL(x, \Psi(\Phi(x))) \tag{3.6}$$

Putting all four terms together yields the final loss function for training OAAE:

$$\mathcal{L}_{oaae} = \mathcal{L}_{recon} + \alpha\mathcal{L}_{feature} + \mathcal{L}_{VQ} + \beta\mathcal{L}_{commitment} \tag{3.7}$$

where α and β weight the different loss terms. Once the OAAE is trained, we denote the latent representation for the frame x^t at time step t as z^t. This provides a structured and disentangled representation, capturing N instances across m classes.

3.2.1.4 Variations of the OAAE

In order to measure whether object decomposition helps with prediction, we also define a non-decomposed version of the VQ-VAE, for use in model **SiS**. This only takes the original non-segmented frame as input. It is processed by a single encoder, with the latent size matched to the total latent size (over all instances) for model **SCAT**. In terms of losses, L_{recon} remains unchanged, L_{VQ}, $L_{commitment}$ and $L_{feature}$ will be a modified to a single

term without summation since there is now a single encoder and codebook, and feature maps from just one instance (e.g. the whole frame). For the **SNCAT** model variant, the OAAE is identical to the main version for **SCAT**, only the subsequent transformer stage is different.

3.2.2 Prediction Model

Using the OAAE, a video clip X is encoded as a sequence of latent representations $Z = \langle z^1, z^2, \ldots, z^T \rangle$. To learn the instance dynamics and its relationship with other instances, we modify the original decoder-only transformer (Vaswani et al. 2017; Radford et al. 2018) into a slot-per-instance auto-regressive transformer that has cross-attention between instances, and shares parameters across instances of each class.

Our transformer consists of alternating attention and feed-forward blocks. However, unlike typical 1D transformers, it includes factored spatial and temporal attention blocks; each of these is applied both for self-attention (i.e., each instance independently attending to other locations / time-points of itself), and cross-attention (i.e., each instance attending to different locations / time-points of all other instances). We use PreNorm (Xiong et al. 2020) in each transformer block. The output vectors for each instance from the last transformer layer are concatenated and passed through a linear layer. The output size matches the number of embeddings in OAAE, allowing the model to predict the probability of possible indices of future frames.

Because the latent vectors produced by the OAAE are a concatenation of each object instance's latent encoding, we can write the sequence of latent encodings in the video for each individual object instance as $\tilde{Z}_k = \langle z_k^1, z_k^2, ..., z_k^T \rangle$ where k denotes the k_{th} instance.

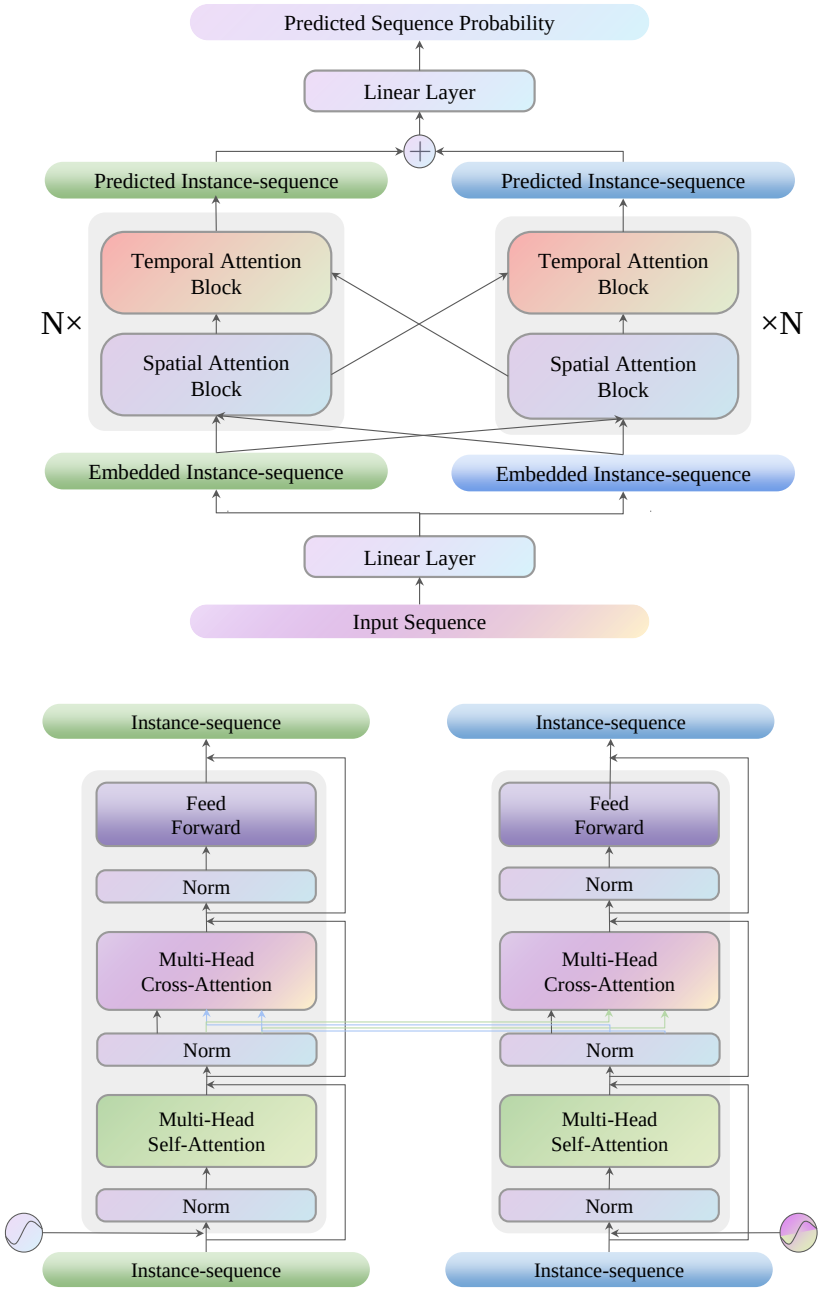

Figure 3.3: **Top:** Architecture of the multi-object latent transformer. **Bottom:** Detail of spatial and temporal attention blocks.

3.2.2.1 Spatial and temporal extensions of attention layers

Since an instance latent sequence $\tilde{Z}_k$ has a 3-dimensional shape $t \times (h \times w) \times c$, where c represents embedding dimension in OAAE, it encompasses both temporal and spatial information. Merely flattening the latent vector to form the video sequence in latent space risks losing crucial spatial details. Hence, inspired by (Sun et al. 2023), all attention layers are applied in both spatial ($h \times w$) and temporal t dimensions. This ensures the model can capture not only the temporal relationships within the sequence but also the important spatial information embedded within each latent representation.

3.2.2.2 Instance-level self-attention

For each latent instance frame z_k^t in the sequence, we first apply learnable positional embeddings. This embedding is added to the input features prior to self-attention to provide the model with information about the position of each instance within the sequence. Scaled self-attention is then applied to each instance sequence separately in order to learn instance-specific dynamics:

$$\mathrm{SA}_c(\tilde{Z}_k) = \mathrm{softmax}\left(\frac{Q_k K_k^T}{\sqrt{d_k}}\right) V_k \tag{3.8}$$

where SA denotes instance-specific self-attention for objects of class c, Q_k, K_k and V_k are the key, query and value calculated by a linear function on $\tilde{Z}_k$; $\frac{1}{\sqrt{d_k}}$ is a scaling factor that prevents excessively large values in the attention score. Following self-attention, we apply a further linear projection layer.

3.2.2.3 Instance-level cross-attention

After the self-attention layer that treats each instance separately, we apply cross-attention between instances to learn the potential relationships and interactions between objects. In this layer, each instance attend the space/time dimensions of each of the other instances:

$$\mathrm{CA}(\tilde{Z}_k) = \bigoplus_{i=1\ldots N,\, i\neq k} \mathrm{softmax}\left(\frac{Q_k {K_i}^T}{\sqrt{d_k}}\right) V_i \tag{3.9}$$

Here CA denotes the cross-attention operation between instance k and the remaining instances. The value V_i and key K_i are derived from $\tilde{Z}_i$, while the query originates from $\tilde{Z}_k$. The cross-attention layer's output, being $n-1$ times larger than the input because of concatenation, is reduced to the original size through a linear layer.

3.2.2.4 Training and inference

The model outputs probabilities over the codebook indices from OAAE, and we use cross-entropy loss to minimize the difference between the predicted and actual distributions. During training, all model variants are trained with teacher forcing on 10-frame clips. Before the forward pass, 10% noise sampled from a standard normal distribution $\mathcal{N}(0,1)$ is added to the input frames. During inference, auto-regressive sampling is used, starting from an initial sequence of conditioning frames, with softmax temperature treated as a hyperparameter.

3.2.2.5 Variants of the transformer

We have described the transformer as used in the full model **SCAT**. In the non-interacting model **SNCAT**, cross-attention is simply replaced by a per-object feed-forward network of similar capacity. The single-slot version **SiS** has a single, larger latent vector for the whole scene instead of separate latents for each object, and we also increase the hidden dimensionality of the transformer (in fact resulting in considerably more parameters). The number of feed-forward and self-attention layers remains the same.

3.3 Experiments

We perform a series of experiments to measure the benefit of separately modeling the dynamics of objects during video prediction. Our focus is on comparing different model variants in a controlled setting, keeping model capacity approximately equal but changing whether the latent representation is decomposed over objects, and whether interactions between objects are modeled if so. In addition, to place our results in context, we perform a comparative evaluation against other recent video prediction models under similar conditions.

3.3.1 Experimental protocol

Each model is given five frames as input, then predicts the following 5–25 frames depending on the dataset. We use 64×64 resolution for all datasets; The models are implemented in PyTorch and trained on a single NVIDIA RTX 3090 GPU, reflecting our emphasis on computational efficiency and model scalability; To ensure a rigorous comparison that focuses on the benefit of instance decomposition, we ensure the numbers of parameters in each model are as similar as possible. Our focus is not on achieving state-of-the-art performance but rather on analyzing the benefits of explicit object-centric modeling within

a balanced and controlled setting. For quantitative evaluation, we report Peak Signal-to-Noise Ratio (PSNR) (Horé and Ziou 2010), Structural Similarity (SSIM) (Wang et al. 2004), and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018). PSNR measures the pixel-wise fidelity between the generated and ground truth image pairs; SSIM evaluates perceptual similarity in terms of luminance, contrast, and structure; LPIPS uses deep network features to capture perceptual similarity. We focus on LPIPS scores in this chapter, because it is more aligned with human perception while PSNR and SSIM are overly sensitive on slight misalignment that leads to poor scores. The results are obtained by sampling with 10 different temperature values ranging from 0.1 to 1.0 with an increment of 0.1 (from low to high stochasticity), and using **argmax** to sample the most likely future indices yielding 11 evaluations in total. For each test video sequence, 25 samples are generated for the same input, which is standard in stochastic prediction tasks (Denton and Fergus 2018; Yan et al. 2021), and the best one is selected in terms of metric score. After evaluating different model variants on each dataset, bootstrapping is used to estimate the spread. We sampled 10000 same-sized evaluation sets with replacement, then calculated the mean and standard deviation of these sets, which are reported in the tables and figures.

3.3.2 Datasets

We conduct experiments on five different datasets characterized by weak and strong interactions. We define weak interactions as scenarios where the dynamics of an instance are unaffected by other instances, or minimally so. In contrast, strong interactions involve instances significantly affecting each other's dynamics, such as during collisions. Since our focus is measuring how the interaction between objects are handled by explicit object decomposition and cross-attention, we do not address the problem of background motion in this chapter, therefore none of the dataset we use features moving background.

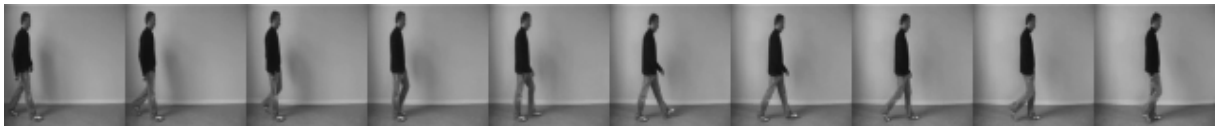

Figure 3.4: An example of **KTH** dataset

Figure 3.5: An Example of **Real-Traffic** dataset

The first weak interaction dataset we use is the **KTH** human action dataset (Schuldt et al. 2004). This includes six action types performed by 25 individuals. Although the primary focus is on the person, there remains some slight interaction between the person and the background, such as shadows cast by the individual on the background. Following MOSO (Sun et al. 2023), we use videos of persons 1-16 for training and 17-25 for testing. We used CLIPSeg (Lüddecke and Ecker 2022) to segment the person and the background with the prompt `'person'` and `'background'`. Each prediction model is conditioned on an input sequence of five observed frames and is required to predict subsequent 15 future frames. This setup is motivated by the characteristics of the running scenes in the KTH dataset, where the person typically exits the frame around the 20th frame after entering a scene. Predicting beyond this point would yield limited meaningful content, as the primary subject of interest is no longer visible.

The second weak interaction dataset is the **Real-Traffic** dataset from Ehrhardt et al. (2020). This comprises video clips taken from a CCTV camera overlooking a highway intersection. The background is static, and only the cars are moving in the scene; there are up to five cars per clip. The original dataset contains 615 video clips with various lengths of total frames, we split the dataset into a more standardized 10 frames per clip with 5,089 clips for training and 2,181 for validation. During inference, the models are given five frames and required to predict five future frames. We used YOLOv8 (Reis et al. 2023) to extract each instance. Each car's motion is independent of other cars most of the time; however, interactions do occur, such as when a car stops before the intersection, causing

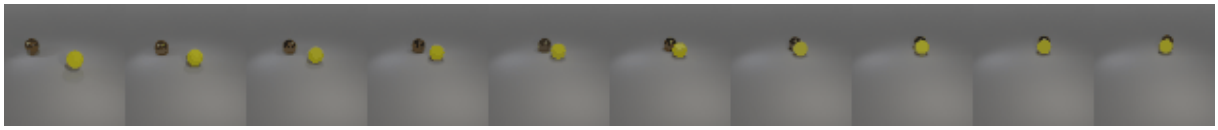

Figure 3.6: An example **CLEVR-2** dataset

Figure 3.7: An example **CLEVR-3** dataset

other cars behind it to slow down. For quantitative evaluation, we therefore identify a subset of video clips from the test set with the strongest interactions. We calculate the distances between centroids of different cars, and select clips where the distance between any pair of cars is less than 25% of the image size; this yields a test set of 807 clips.

For strong interactions, we used Kubric (Greff et al. 2022) to generate a series of synthetic datasets inspired by CLEVRER (Yi* et al. 2020) but exhibiting stronger interactions and more visual complexity. Specifically, **CLEVR-2** contains scenes with two spheres with random velocity sampled such that they will collide; **CLEVR-3** scenes are similar but include another sphere that does not interact with the first two. **Kubric-Real** uses a realistic background and replaces the basic geometric objects with 3D-scanned objects—bottles and pots, since these exhibit interesting dynamics due to their cylindrical shapes. All three datasets use a colliding position range of $[-1, 1]$ and a fixed, static camera looking at $(0, 0, 0)$. The summoning radius is set to 5 for CLEVR datasets and 8 for Kubric-Real, with minimum summoning distances of 2 for CLEVR and 4 for Kubric-Real. CLEVR datasets feature object friction values of 0.4 for metal spheres and 0.8 for rubber spheres, while Kubric-Real has a uniform friction of 1.0. This higher friction in Kubric-Real necessitates a larger maximum initial velocity of 7, compared to 5 in the CLEVR datasets. The number of objects also increases from 2 in CLEVR-2 to 3 in CLEVR-3, and 4 in Kubric-Real. The generation parameters are given in Table 3.1 and the examples are shown in Figure 3.6, Figure 3.7 and Figure 3.8. For all synthetic datasets, the models

Figure 3.8: An example of **Kubric-Real** dataset

	CLEVR-2	**CLEVR-3**	**Kubric-Real**
Colliding Position Range (x, y)	[(-1, 1),(-1, 1)]	[(-1, 1),(-1, 1)]	[(-1, 1),(-1, 1)]
Radius for Summoning Objects	5	5	8
Min Distance When Summoning	2	2	4
Max Initial Velocity	5	5	7
Ground Friction	0.3	0.3	0.3
Object Friction	0.4,0.8	0.4,0.8	1.0
Num Objects	2	3	4
Num Object Class	1	1	2
Camera Position	Fixed Static	Fixed Static	Fixed Static
Camera Looks At (x, y, z)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)

Table 3.1: Parameters for Generating **CLEVR-2**, **CLEVR-3** and **Kubric-Real** Datasets

are required to predict 25 future frames given five observed frames. This is because most interactions in these scenarios complete or stabilize around the 25th frame.

3.3.3 Results

In this section, we first describe the process of how we select the best performing sample of n samples, as well as the best result of the whole evaluation set. Then, we compare our proposed model variants to evaluate the benefit of explicit object-centric modeling in a controlled setting. Finally, we also compare our best performing model variant with other similar approaches.

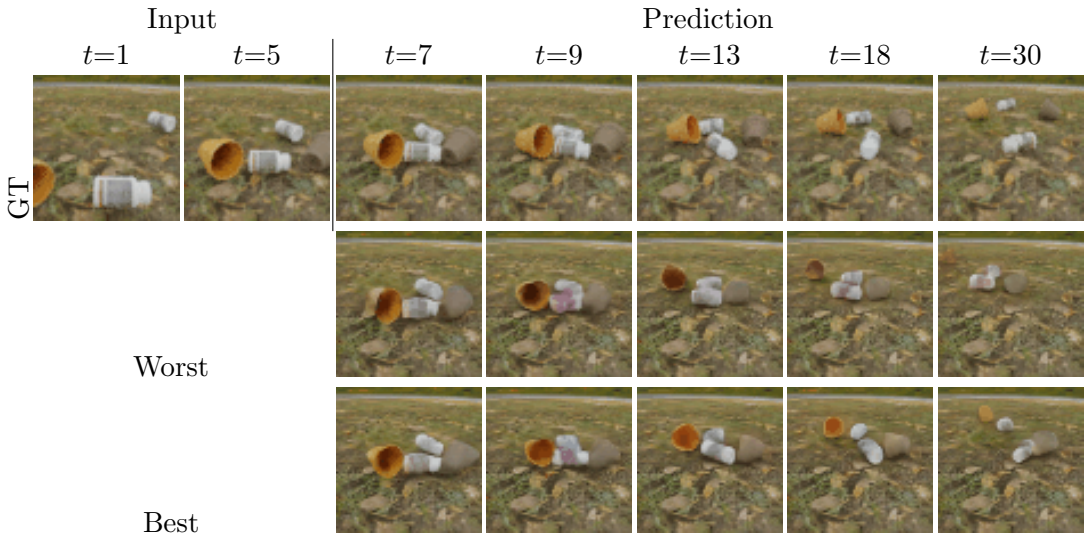

Figure 3.9: Worst and best cases of 25 samples generate by SCAT on Kubric-Real when the temperature equals to 0.7 which performs best among other temperatures.

3.3.3.1 Evaluation Protocol

As described previously in Section 3.3.1, we now provide details on our evaluation strategy. Specifically, we explain: (i) how we select the best-of-N samples for each test sequence, where N=25; and (ii) how we identify the best overall result among 11 evaluations obtained under different temperature values. Figure 3.9 and Figure 3.10 shows the qualitative and quantitative results, which are generated by using **SCAT** model on **Kubric-Real** dataset, of best-of-N samples. Each sample is generated based on the following:

$$P(y_t = k) = \frac{\exp\left(\frac{z_{t,k}}{\tau}\right)}{\sum_{j=1}^{V} \exp\left(\frac{z_{t,j}}{\tau}\right)}, \quad y_t \sim \text{Categorical}(P(y_t = k)) \tag{3.10}$$

where t denotes the timestep in the predicted sequence, k indexes a candidate token from the vocabulary of size V, and $z_{t,k}$ is the logit produced by the model for token k at timestep t. The scalar $\tau > 0$ is a temperature parameter controlling the sharpness of the probability distribution: values $\tau < 1$ make the distribution more peaked, whereas $\tau > 1$ produce a smoother distribution. The term $P(y_t = k)$ represents the probability of selecting token k at timestep t after applying the temperature-scaled softmax transformation to the logits. Finally, $y_t \sim \text{Categorical}(\cdot)$ denotes sampling the discrete token index y_t from the categorical distribution defined by these probabilities. Both the qualitative and quantitative results indicate that the best-case predictions are significantly closer to the ground truth

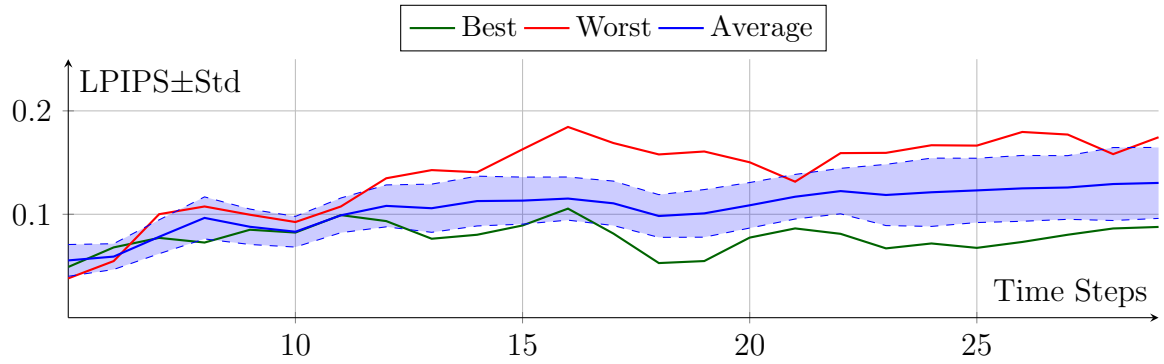

Figure 3.10: Worst, Average and Best cases of the sample shown in Table 3.9; Note that the Standard deviation presented in this figure is obtained without using bootstrapping technique

compared to the average or worst cases. As illustrated in Figure 3.9, the predicted object trajectories in the best-case sample are more accurately aligned with the ground truth, whereas in the worst-case sample, the object positions deviate substantially. However, the worst-case does not indicate our model's performance is poor but the object trajectories are distant from the groun-truth trajectories.

After selecting the best-case prediction for each sequence in the test set of a given dataset, we compute the overall mean and standard deviation of the evaluation metrics using bootstrapping, as described in Section 3.3.1, for each different temperatures. The evaluation results of the SCAT model on the Kubric-Real dataset are presented in Table 3.2 and Figure 3.11. As shown in Figure 3.11, increasing the temperature generally improves the prediction quality, likely due to enhanced sample diversity. However, when the temperature becomes too high (i.e., the sampling becomes overly stochastic), the prediction quality begins to deteriorate. This trend is also evident in Table 3.2, where all metric scores improve steadily as the temperature increases from 0.1 to approximately 0.6-0.7, but degrade beyond that point. These results indicate that moderate stochasticity can help the model avoid overly conservative predictions, whereas excessive randomness leads to unstable or unrealistic outputs.

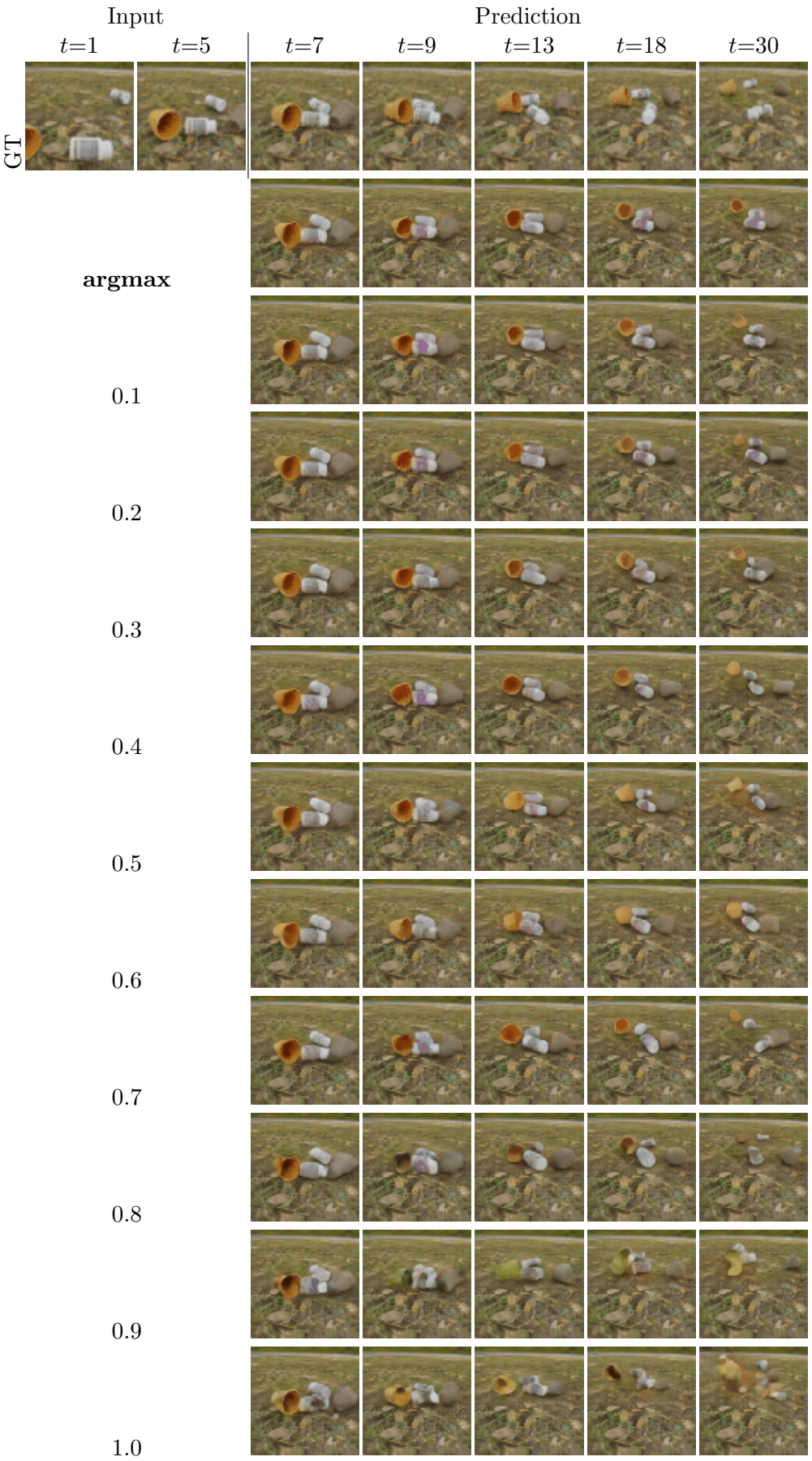

Figure 3.11: Performance of SCAT on the Kubric-Real dataset across temperature values

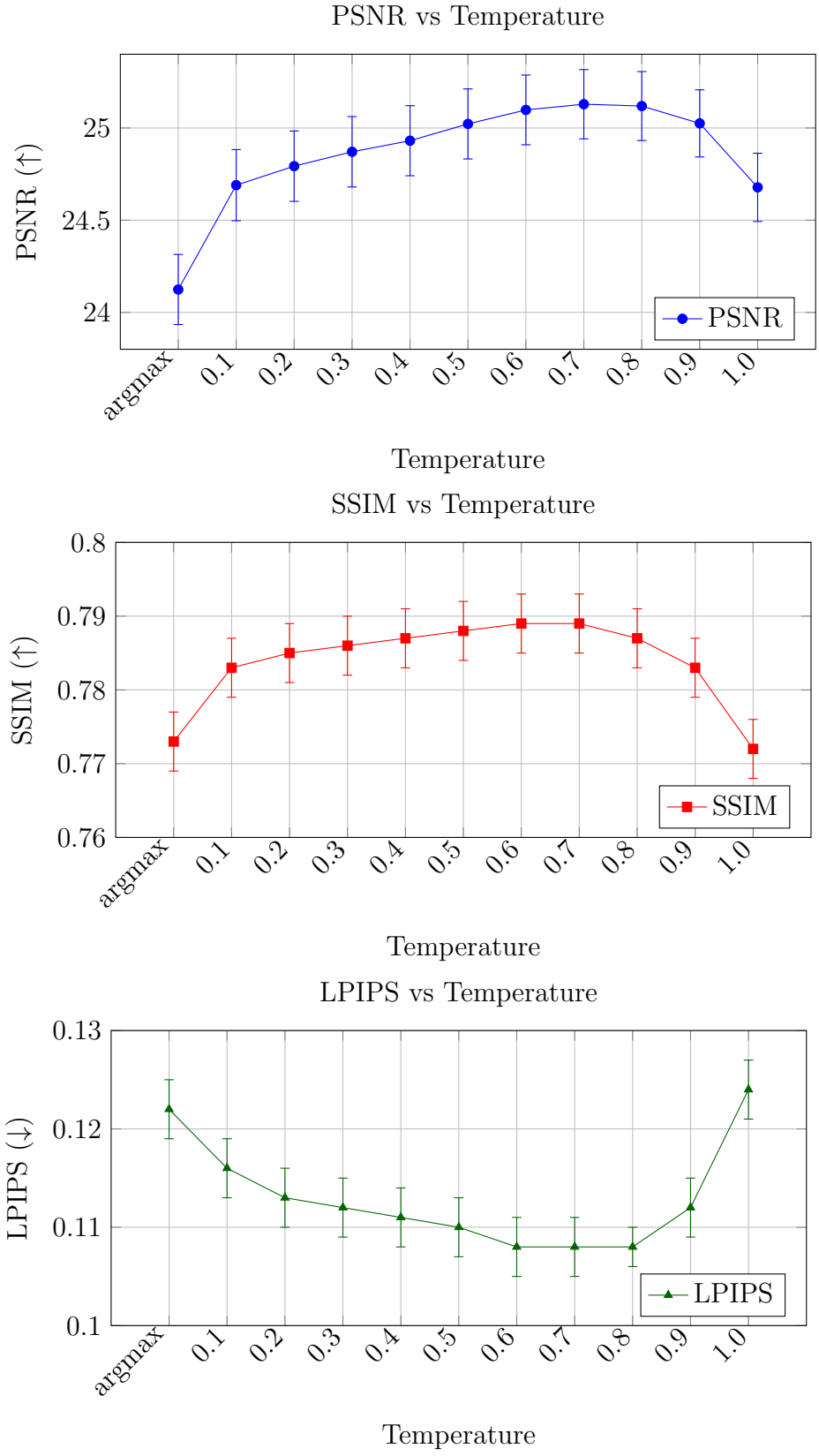

Figure 3.12: Performance of the SCAT model on the Kubric-Real dataset under varying sampling temperatures. Each subplot shows the trend for one evaluation metric. Moderate temperatures improve performance, while both excessive randomness and deterministic sampling (argmax) result in degraded predictions.

temperature	PSNR↑	SSIM↑	LPIPS↓
argmax	24.124±0.190	0.773±0.004	0.122±0.003
0.1	24.690±0.193	0.783±0.004	0.116±0.003
0.2	24.793±0.191	0.785±0.004	0.113±0.003
0.3	24.871±0.191	0.786±0.004	0.112±0.003
0.4	24.931±0.190	0.787±0.004	0.111±0.003
0.5	25.022±0.190	0.788±0.004	0.110±0.003
0.6	25.098±0.189	0.789±0.004	0.108±0.003
0.7	**25.129±0.188**	0.789±0.004	0.108±0.003
0.8	25.119±0.187	0.787±0.004	**0.108±0.002**
0.9	25.025±0.182	0.783±0.004	0.112±0.003
1.0	24.678±0.185	0.772±0.004	0.124±0.003

Table 3.2: LPIPS score of SCAT on **Kubric-Real** dataset with different temperature parameters

3.3.3.2 Internal Evaluation

Table 3.3 shows quantitative results on the two weak-interaction datasets. For **KTH**, the models are given five frames and required to predict 15 frames and for **Real-Traffic**, they are required to predict five frames. In both datasets the SCAT model performs better than the two other variants (SNCAT & SiS). First, modeling the scene separately by segmenting it at the instance level (SNCAT) leads to predictions comparable to modeling the whole scene at once (Single-slot model), while using a much smaller model (25M vs. 48M parameters on **KTH**, 27M vs. 286M parameters on **Real-Traffic**). In **KTH**, we see negligible decrease compared to SiS model, whereas in **Real-Traffic** a slight improvement has been made due to this dataset having more instances and stronger interaction between instance compared to **KTH**. Second, adding cross-attention to the model to handle potential interactions between instances (SCAT) leads to an improvement in performance across all metrics. Since **KTH** features a single instance with negligible interaction, the performance improvement is subtle on each metric: SSIM (+0.003), PSNR (+0.05) and LPIPS (-0.03). On **Real-Traffic**, which has more instances and higher interactions, consistent improvements are observed in all metrics (PSNR: +0.78, SSIM: +0.01, LPIPS: -0.007). These results confirm the computational advantage of both the decomposition and cross-attention components of the approach. From Figure 3.13, we can see that in **Real-Traffic** dataset, improvements are also shown in every time step of the prediction.

In **KTH** dataset, since the interaction level is negligible, the improvement is not obvious. This shows that the proposed video prediction model is more suitable to scenarios where there are multiple object interactions. Table 3.4 provides quantitative results on

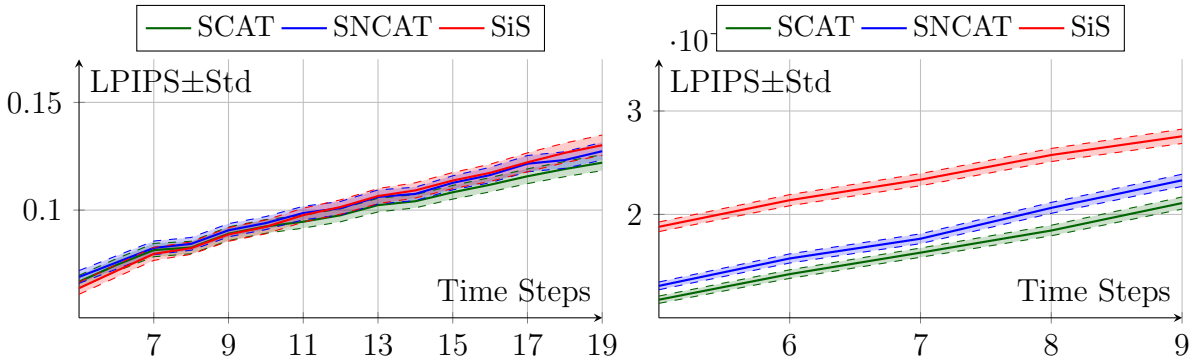

Figure 3.13: Mean and Std of LPIPS metric for **KTH(left)** and **Real-Traffic(right)** datasets

Table 3.3: Quantitative results on **KTH** and **Real-Traffic** datasets

	KTH				**Real-Traffic**			
	PSNR↑	SSIM↑	LPIPS↓	Num-Prms	PSNR↑	SSIM↑	LPIPS↓	Num-Prms
Single-Slot	26.49±0.22	0.786±0.005	0.100±0.003	48M	29.63±0.12	0.939±0.001	0.023±0.0005	286M
SNCAT	26.36±0.17	0.785±0.005	0.101±0.003	25M	30.02±0.12	0.946±0.001	0.018±0.0004	27M
SCAT	**26.54±0.18**	**0.789±0.004**	**0.097±0.003**	23M	**30.41±0.12**	**0.949±0.001**	**0.016±0.0004**	28M

the strong-interaction datasets. On **CLEVR-2**, the SCAT model (PSNR: 31.11) performs similarly to the single-slot model (PSNR: 31.70) but outperforms it on LPIPS (0.047 vs. 0.048). In contrast, SNCAT performs worse than the single-slot model both on **CLEVR-2** and **CLEVR-3** datasets, this is due to the lack of cross-attention to model interactions between objects which lead to deformations of the spheres when collision happens. In **CLEVR-2**, where only two spheres colliding, SiS model can handle this simple interaction. However in **CLEVR-3**, where one sphere is added but not interacted with the original two, SiS model starts to struggle but SCAT performs best by a large margin. This also shows that SCAT's efficiency of modeling multiple objects' motion without the need of big sized model. In **Kubric-Real**, SNCAT preserves object shapes better than the single-slot model, which struggles with deformation after collision. SCAT outperforms both models in LPIPS (0.108 vs. 0.146 for the single-slot model) and SSIM (0.789 vs. 0.748 for the single-slot model), emphasizing the importance of cross-attention in more realistic and complex interaction scenes. Also, From Figure 3.15 we can see that due to the strong interactions, removing cross-attention makes SNCAT unable to beat the single slot model.

In contrast, SCAT performed better than other two variants because of interaction handling with cross-attention. On **Kubric-Real**, note that towards the end of the prediction time frame, the prediction accuracy of **SCAT** and **SNCAT** starts to improve again. This is due to the fact that the moving object has either stopped moving or left the scene entirely. These results confirm our hypothesis that instance segmentation is important for video prediction and that cross-attention is an effective way to encode strong interactions. Moreover, without cross-attention, instance separation on its own is sufficient to achieve similar or better performance compared to the baseline single-slot model on complex scenes (**Real-traffic**, **Kubric-Real**) having more than two instances, with only a fraction of the parameters.

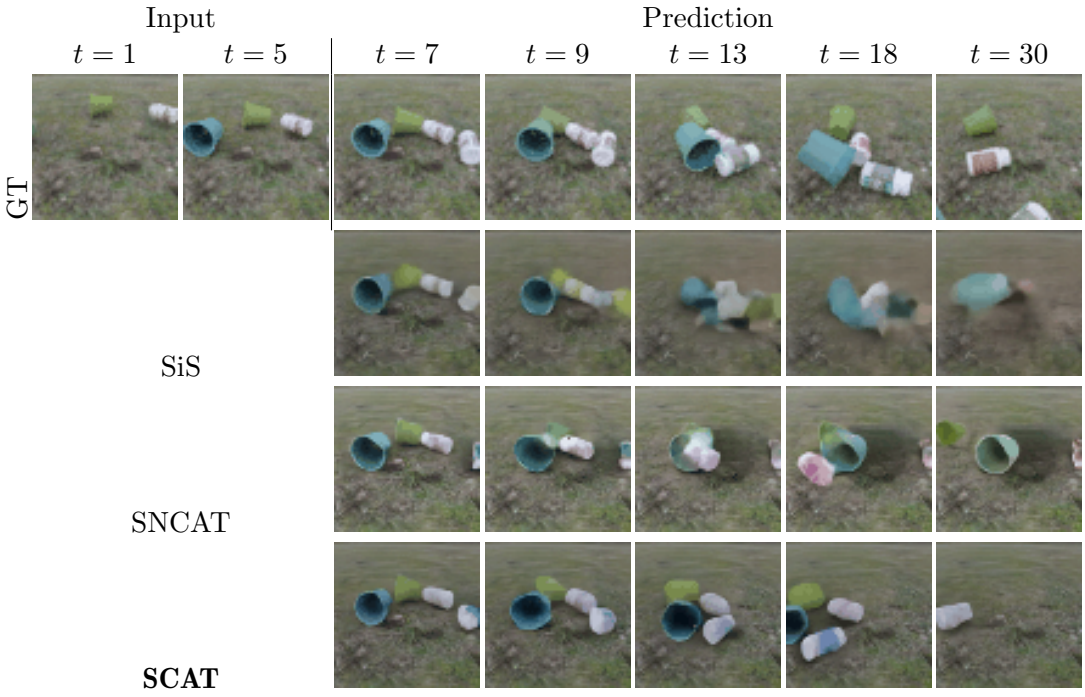

Figure 3.14: Comparison of different model variants on the **Kubric-Real** dataset. SCAT successfully predicted that the blue pot bounced away whereas SNCAT neglected the interaction between other objects and let the blue pot go through from other objects. The single-slot model SiS fails to capture the appearances well, yielding indistinct predictions for later frames.

Table 3.4: Quantitative results on **CLEVR-2**, **CLEVR-3**, and **Kubric-Real** datasets

	CLEVR-2				**CLEVR-3**				**Kubric-Real**			
	PSNR↑	SSIM↑	LPIPS↓	Num-Prms	PSNR↑	SSIM↑	LPIPS↓	Num-Prms	PSNR↑	SSIM↑	LPIPS↓	Num-Prms
Single-Slot	**31.70±0.14**	**0.925±0.001**	0.048±0.001	105M	31.25±0.11	0.911±0.001	0.057±0.001	186M	24.14±0.17	0.748±0.004	0.146±0.002	287M
SNCAT	29.72±0.10	0.908±0.001	0.093±0.002	25M	29.55±0.01	0.898±0.002	0.087±0.002	26M	24.18±0.18	0.759±0.004	0.139±0.003	38M
SCAT	31.11±0.12	0.919±0.001	**0.047±0.001**	25M	**34.42±0.14**	**0.947±0.001**	**0.022±0.001**	26M	**25.13±0.19**	**0.789±0.004**	**0.108±0.003**	40M

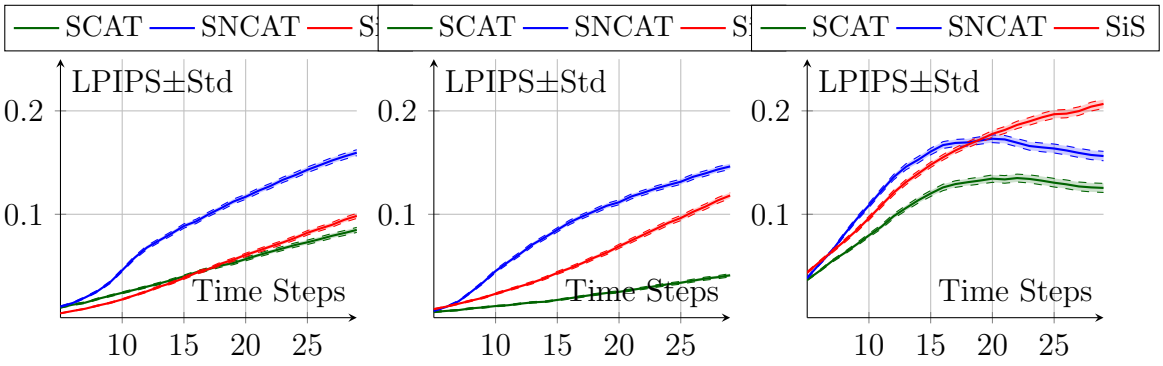

Figure 3.15: Mean and Std of LPIPS metric for **CLEVR-2(left)**, **CLEVR-3(middle)** and **Kubric-Real(right)** datasets

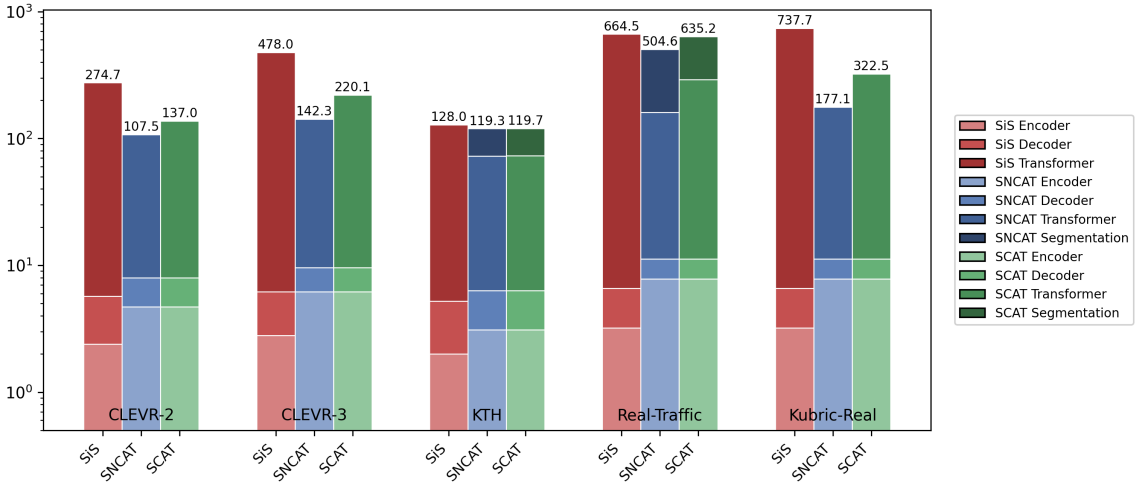

Figure 3.16: FLOPs (GMac) of a single forward pass comparison across different model variants; Note that the Y-axis in this figure uses **log-scale**

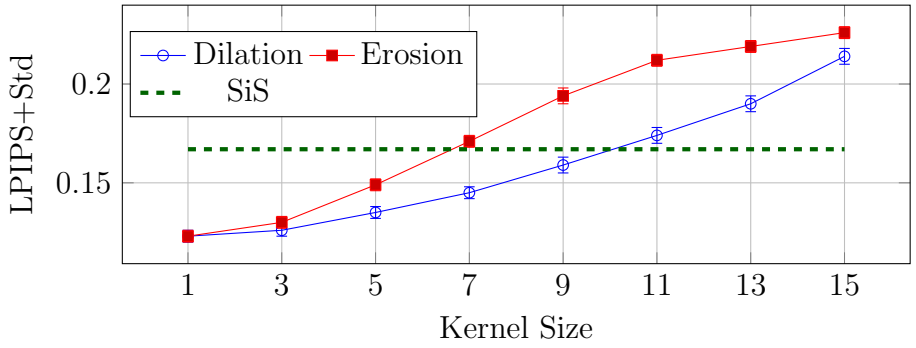

Figure 3.17: Impact of over- and under-segmentation on SCAT performance simulated via dilation and erosion operations on Kubric-Real dataset. We evaluated the samples generated by using **argmax** on logits to isolate the effect of dilation and erosion from stochasticity.

In addition to the model's prediction performance, we also measure the FLOPs of a single forward pass of our proposed variants to evaluate their computational efficiency, which shown in Figure 3.16. It shows both SCAT's and SNCAT's encoder FLOPs are slightly higher compared to SiS's encoder, this is expected because the variants with decompos-

ition have individual encoder for an object class where single-slot encoder only have a single encoder. However, the total FLOPs of decomposed variants are smaller than the one without decomposition across different datasets even when the segmentation model is involved. This suggests the decomposed variants are more computationally efficient than non-decomposed variant. Since SNCAT and SCAT variants depend entirely on the performance of instance segmentation model, we simulate under- and over-segmentation of an instance segmentation model with image processing techniques such as erosion and dilation. Figure 3.17 shows that with the increase of the kernel size, performance of SCAT is decreased. This suggests when the segmentation model's performance is poor, the proposed pipeline's performance will also decrease accordingly. It is worth noting that although both over- and under-segmentation has negative impact on the prediction quality of SCAT, we can see when the objects are over-segmented (dilation), it tends to have smaller effect compared to under-segmentation. This is likely because over-segmentation still provides full information about an instance. More generally, SCAT still performs better or similar to SiS when the kernel sizes of dilation and erosion is relatively small (9 for dilation and 7 for erosion). The implication is that even when the segmentation model makes small errors, explicit models like SCAT will still outperform single-slot models.

3.3.3.3 External Evaluation

Although the main focus of our work is on measuring the benefit of object-centric video modeling in a controlled setting, we also compare our method with other similar methods to better contextualize those results. Our model is designed to be small yet efficient, demonstrating high performance without the need for large-scale resources. In contrast, many existing models rely on significantly larger architectures and large-scale datasets to achieve similar results, which can be resource-intensive and less practical. To ensure a fair and balanced evaluation, we therefore adjusted each method's hyperparameters to match our model's size (i.e., number of weights), providing a level playing field for comparison. We compare against VideoGPT (Yan et al. 2021), which uses a similar architecture, and the CNN-based SimVP (Gao et al. 2022) for a comprehensive evaluation.

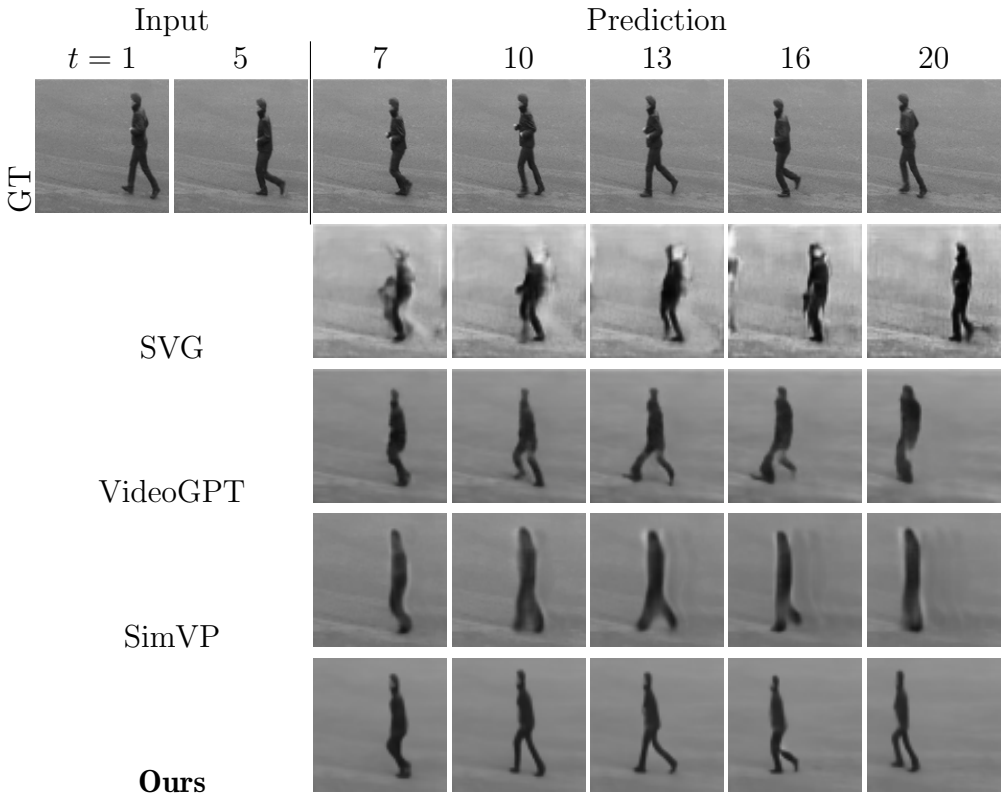

Figure 3.18: Qualitative results from our full model and baselines on the KTH dataset

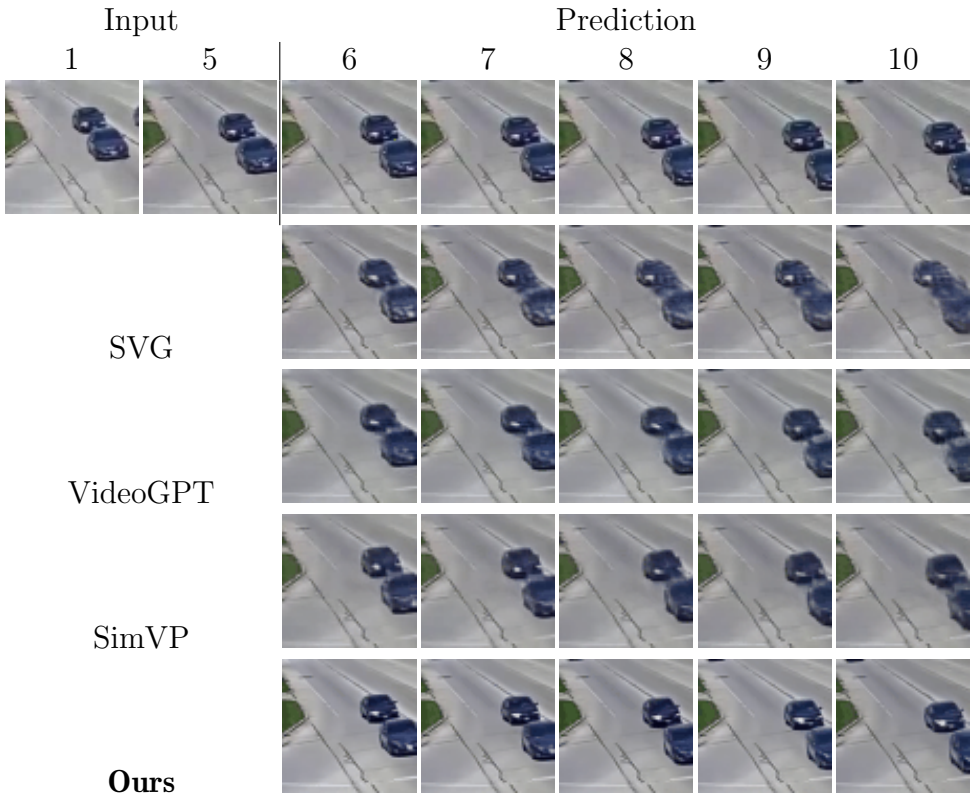

Figure 3.19: Qualitative results from our full model and baselines on the Real-Traffic dataset

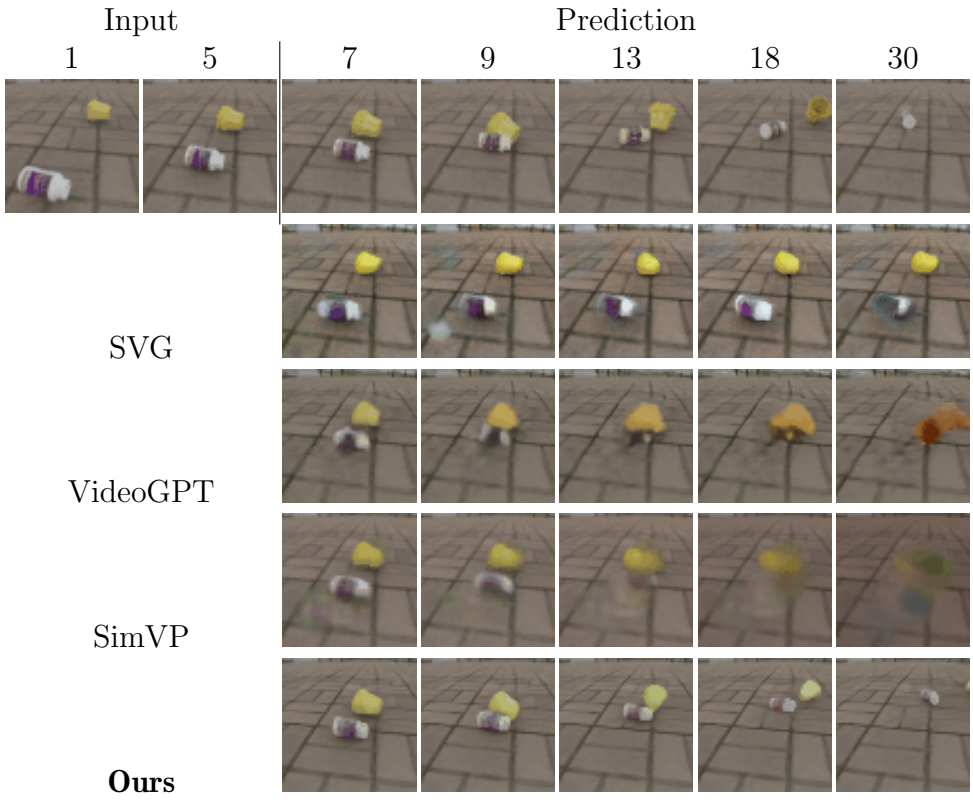

Figure 3.20: Qualitative results from our full model and baselines on the Kubric-Real dataset

Table 3.5: Quantitative results on **KTH**, **Real-Traffic** and **Kubric-Real** datasets

	KTH				**Real-Traffic**				**Kubric-Real**			
	PSNR↑	SSIM↑	LPIPS↓	Num-Prms	PSNR↑	SSIM↑	LPIPS↓	Num-Prms	PSNR↑	SSIM↑	LPIPS↓	Num-Prms
SVG	15.93±0.23	0.614±0.008	0.161±0.004	23M	25.64±0.11	0.900±0.002	0.095±0.0024	31M	16.52±0.13	0.611±0.006	0.699±0.009	41M
VideoGPT	24.44±0.18	0.789±0.004	**0.087±0.002**	41M	29.13±0.10	0.927±0.001	0.023±0.0006	55M	23.62±0.17	0.700±0.005	0.155±0.003	67M
SimVP	25.17±0.22	**0.812±0.005**	0.130±0.004	56M	30.16±0.11	0.949±0.001	0.018±0.0004	31M	22.21±0.15	0.710±0.005	0.213±0.003	59M
SCAT	**26.54±0.18**	0.789±0.004	0.097±0.003	23M	**30.41±0.12**	0.949±0.001	**0.016±0.0004**	28M	**25.13±0.19**	**0.789±0.004**	**0.108±0.003**	40M

Prediction performance on **KTH**, **Real-Traffic** and **Kubric-Real** are presented in Table 3.5 and Figure 3.21. The SCAT model outperforms or is competitive with other models across all three datasets, with a smaller model size, confirming the effectiveness of instance-level segmentation and cross-attention. On the simpler **KTH** dataset, SCAT achieves same SSIM compared to VideoGPT (0.789 vs 0.789) and slightly lower LPIPS than VideoGPT (0.087 vs 0.097), but lower quality according to PSNR (26.54 vs 24.44). Moreover, from Figure 3.18 we can see that only SCAT maintained human posture throughout the prediction. On **Real-Traffic**, SCAT achieved best performance in PSNR metric, with PSNR of 30.41, which is higher than VideoGPT (29.13) and SimVP(30.16). Moreover, SCAT also performs best under the perceptually robust LPIPS metric (0.016), outperforming both VideoGPT (0.023) and SimVP (0.018), indicating better perceptual quality. Also, from Figure 3.19 we can see that when t=9 and t=10, SCAT maintained the distance between

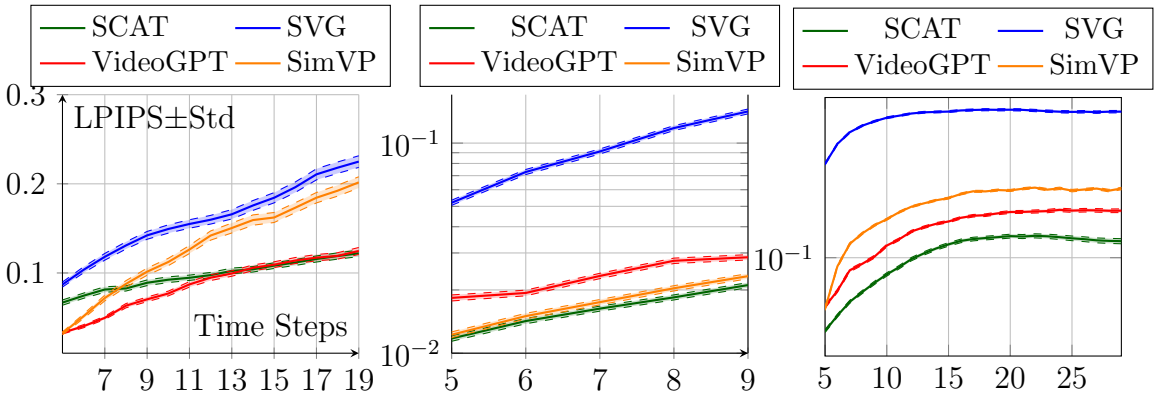

Figure 3.21: Mean and Std of LPIPS metric for **KTH(left)**, **Real-Traffic(middle)** and **Kubric-Real(right)** datasets, where x-axis and y-axis denotes time-step and mean±std, respectively.

Table 3.6: Comparison of FLOPs (GMac), Peak vRAM (GB) and Latency (s) of completing the prediction of required future frames (15 for KTH, 5 for Real-Traffic, 25 for Kubric-Real)

Dataset	SVG			VideoGPT			SimVP			SCAT		
	FLOPs	Peak vRAM	Latency	FLOPs	Peak vRAM	Latency	FLOPs	Peak vRAM	Latency	FLOPs	Peak vRAM	Latency
KTH	78.61	0.46	0.31	89497.6	1.27	15.03	31.5	0.80	0.04	1011.06 + (46.6)	0.68 + (0.98)	0.61 + (0.36)
Real-Traffic	32.44	0.50	0.14	35942.4	1.46	5.88	10.5	0.57	0.01	1414.35 + (344.1)	1.05 + (1.19)	1.33 + (1.94)
Kubric-Real	125.62	0.55	0.49	162247.7	1.64	26.23	121.0	1.49	0.12	7796.71 + (0.0)	1.25 + (0.0)	4.06 + (0.0)

two cars and kept them separate while the other models merged the two cars. Finally, on **Kubric-Real**, where strong interactions and realistic objects are present, our model leads by a large margin on every metric. This further demonstrates that the proposed model achieves larger improvements on scenes with more instances and strong interactions. In Figure 3.20, SimVP, VideoGPT and SVG all failed to predict the collision between two objects, while SCAT predicted this accurately and maintained the object shape.

Following internal experiments, we also compare the proposed method in terms of computational efficiency against the baselines. We compare FLOPs of a single forward pass, peak GPU memory usage in inference time and the total time spent to finish predicting the required number of frames for a dataset. From Table 3.6, we can see SCAT's FLOP is higher than SVG and SimVP, and it is scaled up further with the addition of segmentation models. This is an expected limitation of our model that as the number of classes and instances increases, the cross-attention module will be operated between each instance pairs, leading to high computational cost compared to simple and light architectures like SimVP and SVG. However, SCAT is faster and more efficient than VideoGPT due to their

different prediction strategies. VideoGPT uses latent codes to represent the entire video and predicts frames token-by-token over multiple iterations, requiring more FLOPs and time. In contrast, SCAT predicts all tokens for a frame at a single timestep, making it more efficient. It is still worth noting that all of the experiments conducted in this chapter used relatively limited computation power (single NVIDIA RTX 3090 GPU), therefore this approach can be scaled to devices having more computation power to potentially scale up the inference latency.

3.4 Conclusion

In this chapter, we investigated and analyzed the benefits of explicit object-centric decomposition in video prediction. We presented a flexible video prediction pipeline based on an object-aware VQ-VAE and multi-object Transformer, that operates on separate objects extracted via panoptic segmentation; we also defined variants that lack object-decomposition and support for interactions to measure the impact of these design choices in a controlled manner. We evaluated the proposed models on five datasets, finding that when a dynamic scene is explicitly decomposed and encoded into a structured latent vector, prediction quality is better than an equal-capacity model without decomposition, and that this improvement is larger for scenes that involve strong interactions between objects. This confirms that using both object decomposition and cross-attention to handle interactions improves the overall prediction quality when strong interactions occur in a dynamic scene.

3.5 Limitations

Our model has three inherent limitations. First, object decomposition is entirely reliant on the performance of instance segmentation models, this is evident in Figure 3.17 that the proposed model's performance is decreased when the kernel sizes to simulate over- and under-segmentation became bigger. Second, our experiments throughout the chapter focused solely on static camera settings, and additional experiments would be required to evaluate the robustness of the approach to scenarios with moving cameras. Third, the encoder encodes predefined object classes. For example, pots and bottles in Kubric, cars in Real-traffic and spheres in CLEVR datasets. Based on this predefined latent space, the transformer will also learn to predict the dynamics of the given latent space during training. Because each object in a video is first segmented and the instances which belong to the predefined classes are selected to process, if there are novel object classes outside the scope of the predefined classes, then the novel objects are automatically categorized to the background slot. Therefore, this novel object's motion is learned and predicted implicitly. For example in Kubric-Real, the model is trained to predict the motions of pots and bottles, and if we initialize a new object with different characteristics than pre-defined object-class (i.e., a box), its motion is learned in the background slot implicitly.

Chapter 4

Flow and Depth Assissted Video Prediction for Occlusions

In the previous chapter, we discussed how decomposed modeling of a multi-object scene can be beneficial for future frame prediction. It showed promising results on five different datasets includes both synthetic (CLEVR-2, CLEVR-3, Kubric-Real) and real-world scenarios (KTH & Real-Traffic). The proposed pipeline showed significant improvements on handling object interaction compared to non-decomposed standard video prediction models. However, the datasets chosen to conduct the experiments are designed to have limited object occlusion and without background motion. This leads to a critical research question that can object decomposition alone be beneficial for occlusion and background motion? In the occlusion event, a completely occluded object is invisible to the video prediction model making SCAT less practical in these scenarios. Moreover, the motion information is also not explicit with RGB frames. Therefore, SCAT is not directly applicable to prediction of the dynamics of a clip that has occlusion and background motion.

In this Chapter, we will focus on occlusion and background motion prediction problem which theoretically cannot be handled by solely relying on object decomposition. We build upon the model we proposed in the previous chapter by integrating additional modalities that can provide geometrical and motion information. The chapter is structured as follows:

- Section 4.1 gives brief introduction about the importance of video prediction models' ability to work well in occlusion scenarios;

- Section 4.2 introduces the proposed approach in detail;
- Section 4.3 introduces the dataset we used in this chapter, new metrics to better evaluating the motion of prediction, as well as the experimental results and findings.
- In Section 4.4, a comprehensive conclusion is given and briefly discusses the limitation of this chapter.

4.1 Introduction

Occlusion poses a fundamental challenge in video prediction within dynamic scenes. In multi-object environments, where interactions are common, occlusions frequently occur, causing objects to become partially or fully invisible for brief periods. This phenomenon significantly complicates the task of predicting future frames, as models must infer the motion and appearance of occluded objects from limited visual cues. Similarly, in single-object scenarios, deformable objects, such as garments, can exhibit self-occlusion when one part of the object overlaps another, further increasing the complexity of accurate prediction. Therefore, simply using RGB images for complex motion events is not sufficient and explicit motion information is needed to handle this problem more efficiently.

Several approaches aimed to improve video prediction by incorporating optical flow estimation (Bei et al. 2021; Lu et al. 2021; Luo et al. 2021; Zhang et al. 2024b). However, two major limitations of optical flow are that it accumulates errors over time and loses information when objects become fully occluded, e.g., being completely invisible. As a result, optical flow-based video prediction methods struggle to handle complete occlusions effectively.

Unlike optical flow, which relies on dense pixel-wise motion estimation, recent progress in point tracking methods allows more robust motion estimation by tracking and estimating key points on objects even when they are fully occluded (Karaev et al. 2025; Tumanyan et al. 2024; Xiao et al. 2024).

Equally critical to occlusion handling, depth maps provide rich geometric information about the 3D structure of a scene, enabling precise spatial reasoning and occlusion disambiguation (Godard et al. 2017). Depth maps are essential for determining occlusion hierarchies, as they indicate the relative position of objects from the camera, allowing models to identify which objects occlude others.

In this chapter, we hypothesize that integrating information about depth and the flow of points into a video prediction model will enhance its ability to anticipate object motion, particularly in occluded scenarios and the clips has background motion. While key points helps to track object motion trajectories, depth maps introduce explicit spatial constraints that improve occlusion-aware prediction. To investigate this, we build on our method proposed on the previous chapter as our video prediction model, which lacks robustness to occlusions when only relying on RGB images; we define a new modality derived from tracked key points called point flow and propose a variant that incorporates both point flow and depth map as additional information to the model. 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. We want to understand how the model will perform in terms of motion, specifically we want to measure if the model can predict the reappearance of fully occluded objects. Thus, we test our model not only on appearance-based metrics, but also define two motion-based metrics, which are optical flow difference (OFD) that calculates the difference between the optical flow between predicted and ground-truth RGB frames, and earth mover's distance (EMD) on the binary object map of predicted frames between the ground-truth, to evaluate the predicted motion accuracy.

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 occlusion 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 occlusion heavy datasets.
- We find that when integrating point flow, the reappearance of occluded objects is predicted more accurately.

4.2 Methodology

4.2.1 Preliminaries

Our goal remains the same as the previous chapter, that is to learn a probability distribution on future frames $X^{T+1:T+M}$, conditioned on the past frames $X^{1:T}$. However, instead of conditioning on a single modality (RGB), we will jointly encode both depth and point-flow with their corresponding RGB frame. We next discuss the base model we build on in this chapter, as well as the models used to extract additional modalities—point flow and depth.

4.2.1.1 Base Architecture

We use the model we proposed in the previous chapter, Object Aware Auto-Encoder (OAAE) and Stochastic Class Attended Transformer (SCAT), as our base architecture. Our improvements will be described in detail in Section 4.2.2.

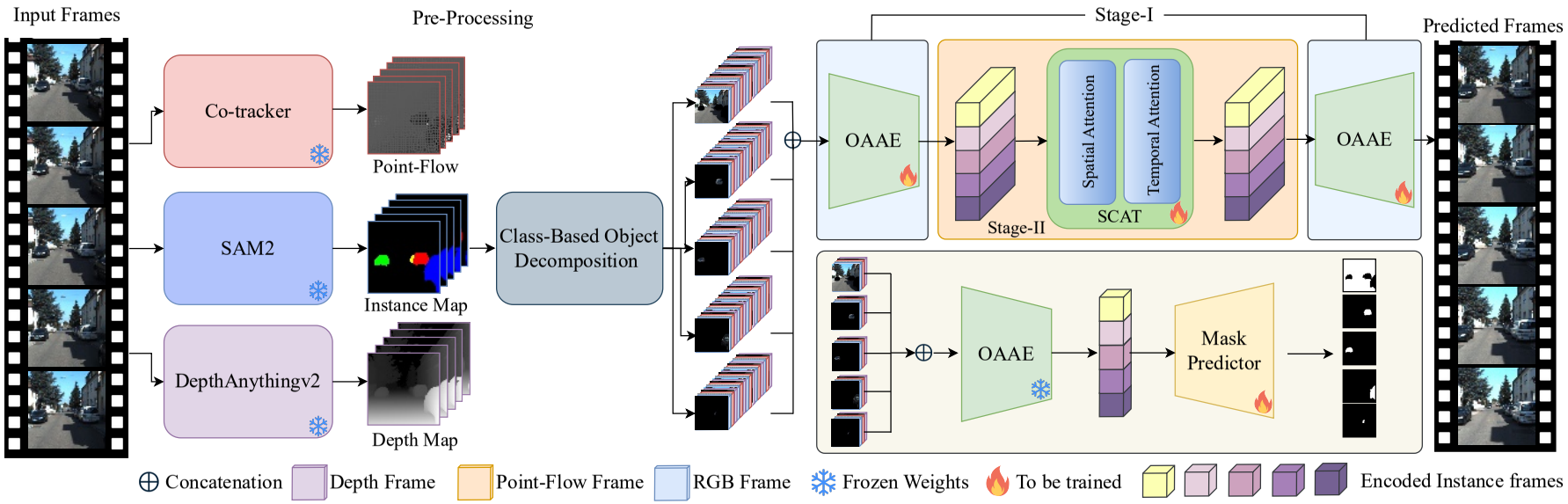

Figure 4.1: **The overview of the proposed method.** First we obtain different modalities by using Cotracker and DepthAnythingV2; then we use SAM2 to segment the original RGB frames sequence to decompose the objects, segmentation map from SAM2 is also used to decompose the point flow and depth map; After 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.

4.2.1.2 Point Tracking with CoTracker

CoTracker (Karaev et al. 2025) 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 $P_i^t = (u_i^t, v_i^t) \in \mathbb{R}^2, \text{for } t \in \{1, \ldots, T\}$. 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 convolutional neural 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 are related the selected features from the encoded representation. The objective of this model is to minimize the distance between the predicted and ground truth point locations.

4.2.1.3 Depth Estimation with DepthAnything-V2

Depth Anything (Yang et al. 2024b,c) 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.

4.2.2 Proposed Method

SCAT (Suleyman et al. 2025) decomposes a video into a set of object instances and models their dynamics in a latent space. However, when an object instance becomes fully occluded at a certain time step, it is no longer visible to the frame encoder. As a result, the corresponding latent representation lacks direct visual evidence, making it difficult to accurately predict the motion of fully occluded objects, even when their prior appearance has been observed.

To address this limitation, we propose incorporating tracked points obtained from Co-Tracker (Karaev et al. 2025) as point flows, providing explicit motion information to the prediction model. Unlike optical flow, which estimates dense pixel-wise motion and is prone to error accumulation over time (Harley et al. 2022), point tracking maintains sparse but temporally consistent trajectories that preserve object identity across frames. When an object becomes occluded, the tracked point trajectories are partially estimated rather than directly observed; however, these estimates are obtained by enforcing temporal motion consistency and leveraging contextual cues from visible regions of the scene. As such, point flows provide informed motion hypotheses rather than arbitrary hallucinations.

By incorporating point flows, the encoder can retain information about an object instances relative position and motion at time step t, even when its RGB appearance is entirely absent due to complete occlusion. We hypothesize that encoding point flows alongside RGB frames enriches the latent representation with explicit relative location and motion cues, thereby enabling more accurate prediction of occluded object dynamics.

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 the model to obtain other modalities. Instead, we use pretrained models solely to pre-process the available RGB sequences, generating point flow and depth images from the same input data used by existing baselines as we discussed previously. Following SCAT, we test our hypothesis by designing a family of models with varying input configurations:

- **SCAT-D**: A model trained with RGB and depth frames;
- **SCAT-P**: A model trained with RGB frames and point flows;
- **SCAT-DP**: A model trained with RGB frames, depth frames, and point flows.

4.2.2.1 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 coordinates of a point and its visibility. From the second frame and onwards ($t \geq 1$), 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

$$\textbf{PointFlow}(T, H, W, 3) = \begin{cases} (0, 0, 1), & \text{if } t = 0, \\ (h_t^n - h_{t-1}^n, w_t^n - w_{t-1}^n, v_t^n), & \text{if } t > 0. \end{cases} \tag{4.1}$$

where $\mathbf{PointFlow}(T, H, W, 3)$ is the displacement tensor, $h_{t,n}$ and $w_{t,n}$ 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, DepthAnythingv2 (Yang et al. 2024c), 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 maps that were used to decompose 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 frame, but also with the reconstructions of other modalities. This make sure that other modalities will be encoded into the latent space.

4.2.2.2 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:

$$\mathcal{L}_{VQ} = \sum_{c=1}^{m} \sum_{k=1}^{n_c} \| \operatorname{sg}[\tilde{z}_k^c] - e_c \|_2^2 \tag{4.2}$$

$$\mathcal{L}_{commitment} = \sum_{c=1}^{m} \sum_{k=1}^{n_c} \| \tilde{z}_k^c - \operatorname{sg}[e_c] \|_2^2 \tag{4.3}$$

$$\mathcal{L}_{recon} = -\log p(x|\Psi(\Phi(x))) \tag{4.4}$$

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 (Zhang et al. 2018) as an additional reconstruction loss:

$$\mathcal{L}_{\text{LPIPS}}(x, \Psi(\Phi(x))) = \sum_l w_l \left\| \phi_l(x) - \phi_l(\Psi(\Phi(x))) \right\|_2^2 \tag{4.5}$$

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 follows:

$$\mathcal{L} = \mathcal{L}_{VQ} + \alpha \mathcal{L}_{commitment} + \mathcal{L}_{recon} + \beta \mathcal{L}_{\text{LPIPS}} \tag{4.6}$$

Where α and β denotes the weights for commitment and LPIPS loss, which are set to 0.25 and 1.0, respectively. 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.

4.3 Experiments

We 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 given five frames and is required to predict

five future frames on KITTI dataset and 20 future frames on Kubric-Occlusion dataset 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.

4.3.1 Datasets

4.3.1.1 Kubric Occlusion

The hypothesis of this chapter is that incorporating point flow and depth map can improve the performance of prediction models, particularly in scenarios involving occlusions and background motion. To test this, we used Kubric (Greff et al. 2022) 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. There are two objects in each clip, we first define a occlusion event location at the range of [-1, 1], then we summon the stationary object at a random location within this range. Second, we summon the moving object with initial velocity to behind of the stationary object at a random position in the circular sector region behind the still object, so that it passes behind the still object. The specific data generation parameters of Kubric is given in Table 4.1. Since, the segmentation and depth maps are automatically calculated by the

	Kubric-Occlusion
Occlusion Event Range (x, y)	[(-1, 1),(-1, 1)]
Radius for Summoning Objects	8
Min Distance When Summoning	4
Max Initial Velocity	7
Ground Friction	0.3
Object Friction	1.0
Num Objects	2
Num Object Class	2 (Bottle & Pot)
Camera Position	Fixed Static
Camera Looks At (x, y, z)	(0, 0, 0)

Table 4.1: Parameters for generating **Kubric-Occlusion** dataset

Kubric generator, it is directly used without using our pre-processing method. However, since point flow is not available from the generator, we used Co-Tracker to track the key points on the scene.

4.3.1.2 KITTI

The **KITTI dataset** (Geiger et al. 2013) is a widely used benchmark for autonomous driving research. It contains various driving scenarios captured in urban, residential, and highway environments. We use a **subset** of KITTI, specifically selecting scenes from *city, residential, and road* categories. We select these scenarios because it features reasonable amount of objects for our model and not extremely complex. Unlike Kubric-Occlusion, a synthetic dataset with all ground-truth labels are available when generated, the segmentation and depth maps are not available for the videos in KITTI. Also, since one video contains various lengths of frames, directly using the proposed pre-processing to to obtain other modalities are not efficient. Most importantly, because the objects in long videos will be replaced regularly and tracking every objects' segmentation map is not particularly helpful for our model. Therefore, we first split the original long video to smaller clips with 10 frames each, then we run our pre-processing method to these clips. Then, we follow the object selection strategy described in Chapter 3, 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; resulting five instances in total. After processing, 2,497 clips are used as training and 639 for testing (each clip contains 10 frames).

4.3.2 Evaluation Metrics

We evaluate the pixel-level quality of predicted frames using standard appearance-based metrics: PSNR(Horé and Ziou 2010), LPIPS(Zhang et al. 2018), and SSIM(Wang et al. 2004). 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. After the model predicts the future frames, optical flow of the predicted future frames is computed using the Gunnar-Farneback method (Farnebäck 2003), then the optical flow of corresponding ground-truth future frames are also calculated using the same method. Finally, 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. Since the segmentation map (e.g., binary mask of an instance) of each instance is available in our dataset, we trained a mask predictor to predict instance masks from the trained OAAE latent space, and use this to estimate masks for predicted frames. We then compute the Earth Movers Distance (EMD) (also known as the Wasserstein distance) between the predicted and 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 thesis is defined as follows: Let $P_t = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\} \subset \mathbb{R}^2$ be the set of pixel coordinates for the predicted mask, and $G_t = \{\mathbf{g}_1, \ldots, \mathbf{g}_n\} \subset \mathbb{R}^2$ be the set of pixel coordinates for the ground truth mask. We define uniform discrete distributions over these sets:

$$\mathbf{a} = \left(\frac{1}{m}, \ldots, \frac{1}{m}\right) \in \Delta^m, \quad \mathbf{b} = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right) \in \Delta^n \tag{4.7}$$

Let $M \in \mathbb{R}^{m \times n}$ be the cost matrix with entries:

$$M_{ij} = \|\mathbf{p}_i - \mathbf{g}_j\|_2 \tag{4.8}$$

The Earth Movers Distance is computed as the optimal transport cost as:

$$\mathrm{EMD}^2(P, G) = \min_{T \in U(\mathbf{a}, \mathbf{b})} \sum_{i=1}^{m} \sum_{j=1}^{n} T_{ij} M_{ij} \tag{4.9}$$

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.

4.3.3 Results

4.3.3.1 Frame Reconstruction Performance

Before evaluating our prediction model, it is crucial to understand what impact is imposed by adding additional modalities to our encoder. Since our strategy is simply concatenating the different modalities on the channel dimension to form the input, only the input channel size will be different and the rest of the VQ-VAE architecture will be exactly the same across different variants of encoders. Therefore, we first evaluate the performance of our autoencoders in reconstructing RGB video frames. Table 4.2 and 4.3 presents the quantitative results and Figure 4.2 and Figure 4.3 shows qualitative results of different variants of the proposed model on both **Kubric-Occlusion** and **KITTI** dataset. In **Kubric-Occlusion**,

	Depth	Point-Flow	PSNR↑	SSIM↑	LPIPS↓
OAAE	✗	✗	26.672±0.133	0.669±0.007	0.052±0.001
OAAE-P	✗	✓	26.737±0.129	0.677±0.006	0.052±0.001
OAAE-D	✓	✗	**27.280±0.127**	**0.712±0.006**	**0.043±0.001**
OAAE-PD	✓	✓	26.194±0.122	0.660±0.007	0.061±0.001

Table 4.2: Autoencoder's frame reconstruction performance on **Kubric-Occlusion** dataset

we observe that integrating a single modality (OAAE-P or OAAE-D) improves overall reconstruction quality. We also noted OAAE-D variant achieves the best results among other variants. This may indicate the ground truth depth map, which is precisely resembling the geometric structure of the RGB image, can provide additional complimentary feature that cannot be captured alone with RGB frame. However, combining both modalities simultaneously (OAAE-PD) leads to decreased reconstruction performance. This suggests a trade-off between incorporating multiple modalities and reconstruction quality under limited latent capacity.

	Depth	Point-Flow	PSNR↑	SSIM↑	LPIPS↓
OAAE	✗	✗	**21.468±0.100**	0.769±0.003	**0.038±0.001**
OAAE-P	✗	✓	20.138±0.093	0.701±0.003	0.056±0.001
OAAE-D	✓	✗	21.316±0.095	**0.770±0.003**	0.040±0.001
OAAE-PD	✓	✓	19.957±0.090	0.695±0.003	0.063±0.001

Table 4.3: Autoencoder's frame reconstruction performance on **KITTI** dataset

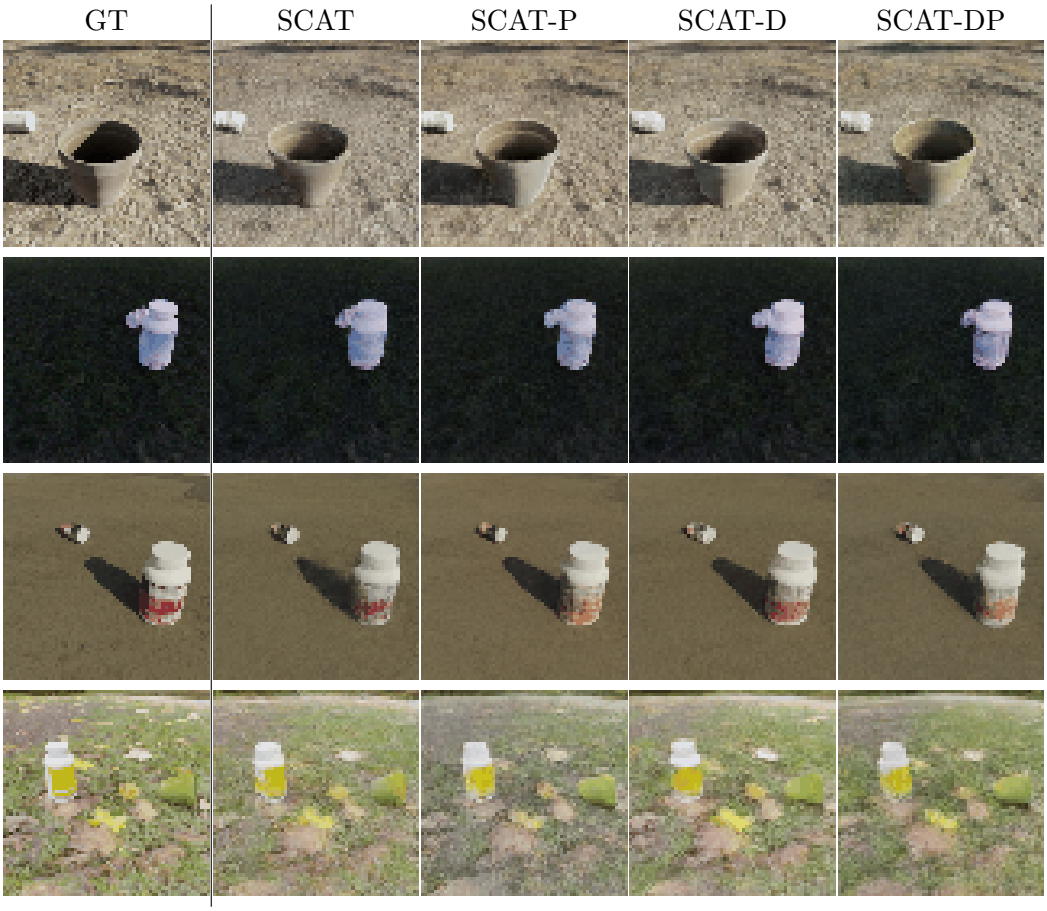

Figure 4.2: Qualitative results on Autoencoder's reconstruction **Kubric-Occlusion** dataset

In **KITTI**, autoencoder's performance significantly degrades when incorporating point flow (OAAE-P & DP). Unlike Kubric-Occlusion, KITTI involves camera motion, making the background non-stationary. As a result, the loss function tasked with reconstructing both RGB frames and additional modalities introduces noise into the RGB output. This is evident in the reconstructed frames shown in Figure 4.3, where small dot-shaped artifacts appear at the point displacement regions. Also in Figure 4.2, we can see that the moving object's appearance and the overall background is poorly reconstructed when point flow is added. These findings suggest that directly concatenating point- flow with RGB frames is not an effective encoding strategy, especially when the background also has its own

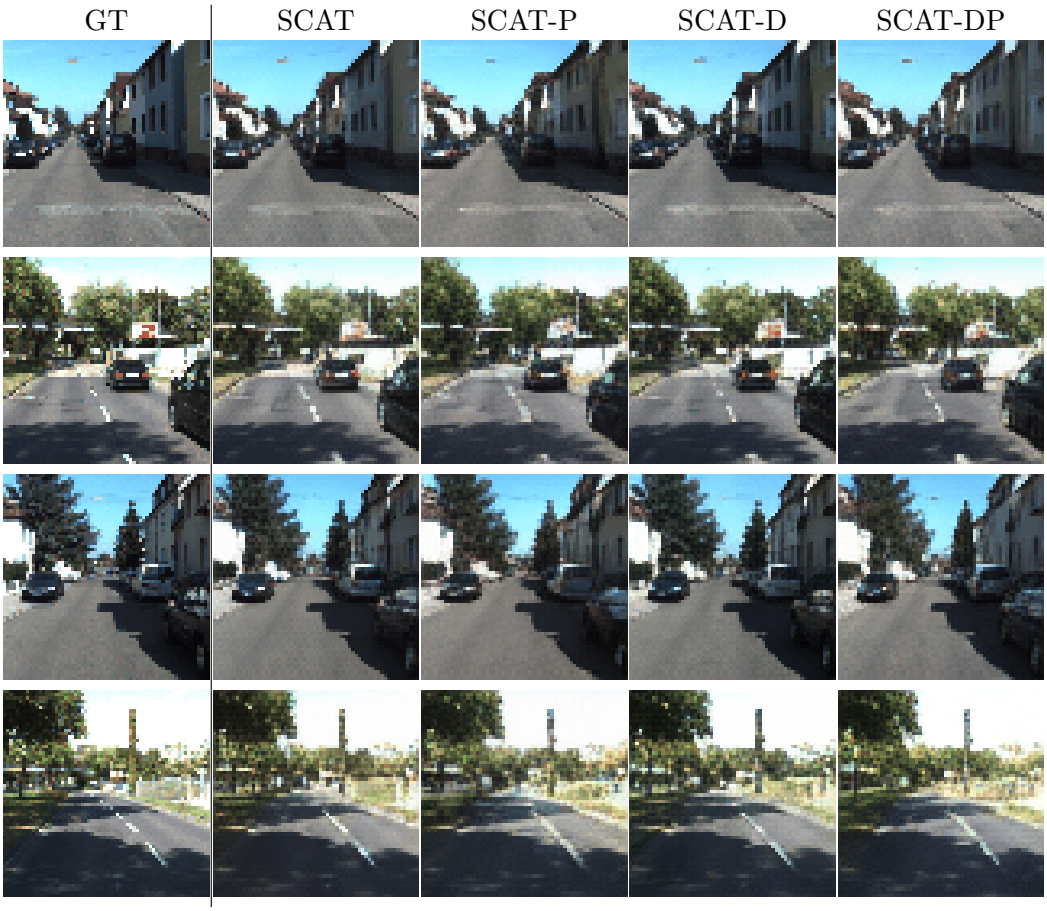

Figure 4.3: Qualitative results on Autoencoder's reconstruction on **KITTI** dataset

motion. However, since the main focus of this chapter is not predicting high quality video frames in terms of appearance, but to investigate and understand the benefits of point flow and depth map brings to motion prediction accuracy especially in the event of occlusions. Therefore, we will ignore this limitation in this chapter and introduce a better encoding method in detail in the next chapter.

4.3.3.2 Prediction Performance

We now evaluate the different variants of our predictors in both datasets and analyze their performance on both appearance and motion based metrics. As we introduced in the previous chapter, we use different temperature parameters to see how well the model will perform under different stochasticity, and then we select the best performing model based on the metric scores. In **Kubric-Occlusion** dataset, the occlusion happens by design. It is less stochastic compared to a collision event that we described in **Kubric-Real** dataset in

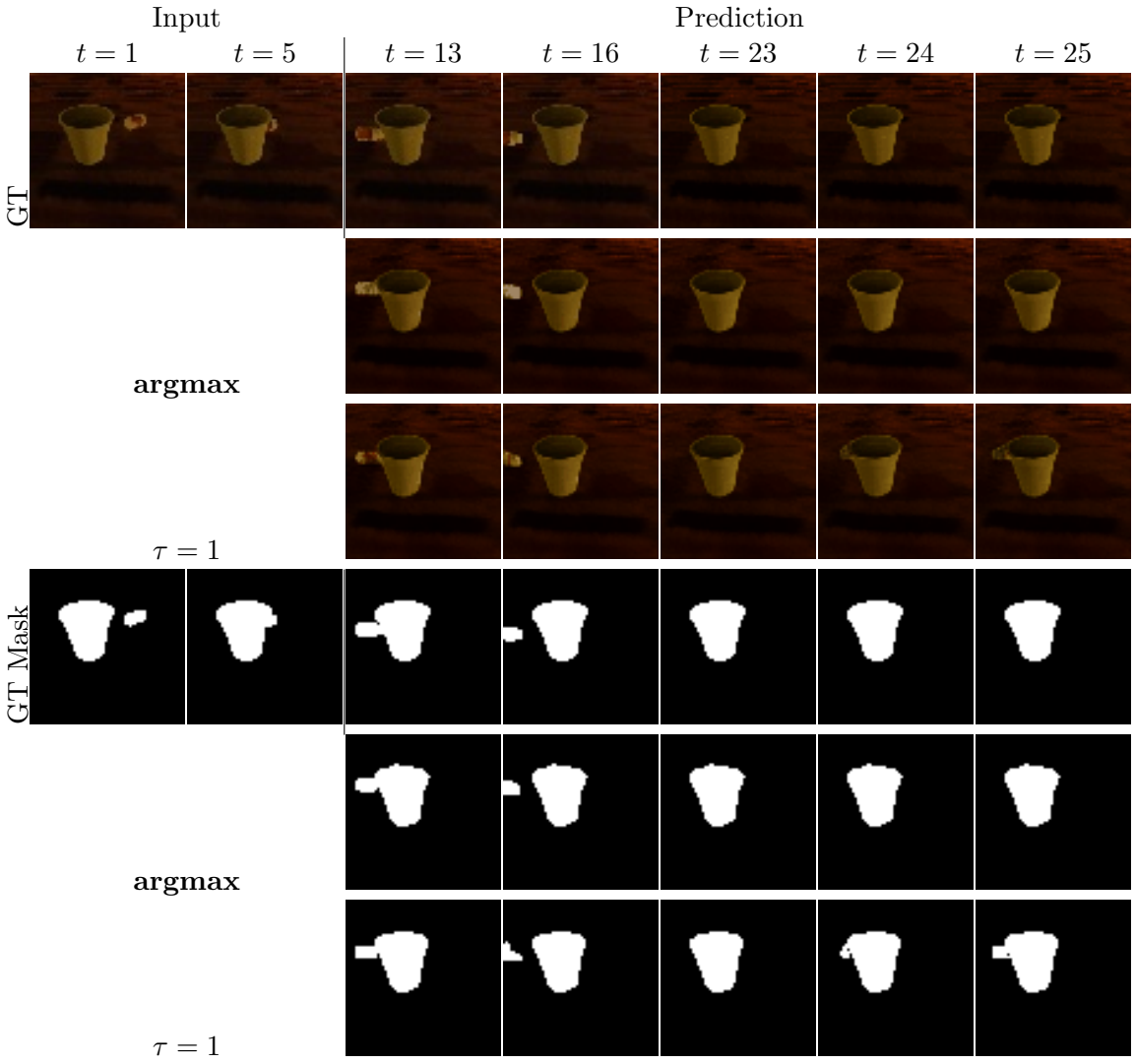

Figure 4.4: Reappearing phenomenon on the **Kubric-Occlusion** dataset when the stochasticity is high

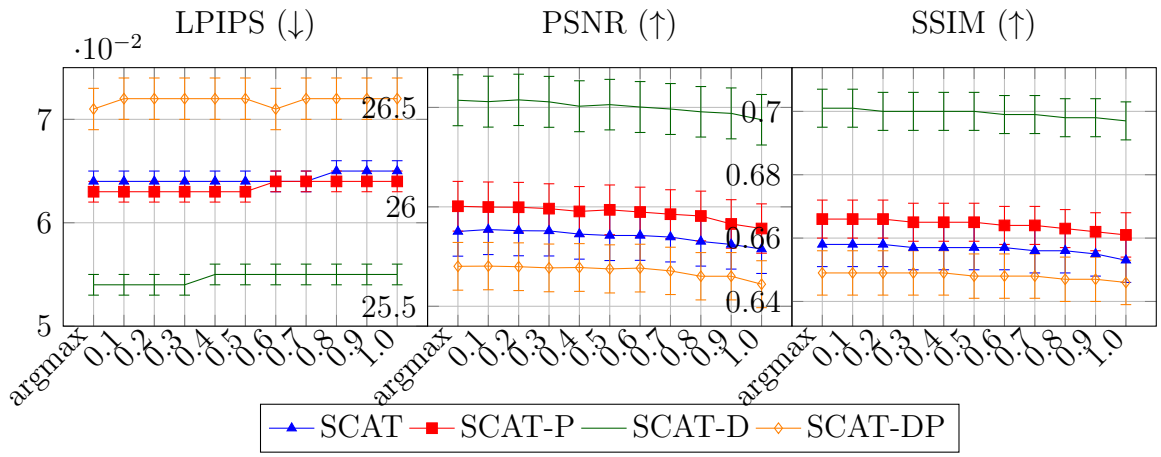

Figure 4.5: Appearance-based metrics (PSNR, SSIM, LPIPS) across temperatures on **Kubric-Occlusion** dataset

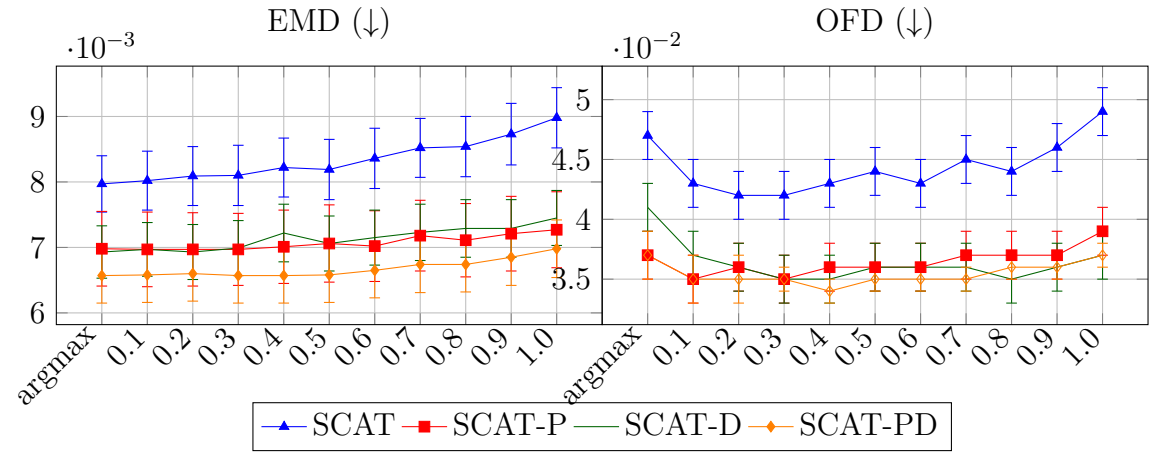

Figure 4.6: Motion-based metrics (EMD & OFD) across temperatures on **Kubric-Occlusion** dataset

the previous chapter. Therefore, by increasing the temperature τ, the prediction model is not expected to improve significantly. This is evident in Figure 4.5 and 4.6. We can see in all appearance based metrics, there are no improvements or very subtle decrease in performance. But in motion based metrics, the models are improved slightly when the temperature value is in between the range of [0.2, 0.3]. The reason for the performance decrease is due to the increasing stochasticity making the models assume there will always be a moving object appearing from the back of stationary object. This is shown in Figure 4.4. From Table 4.4, we can see 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. As consistent in the metric scores reported in Table 4.2, the SCAT-D variant achieves best performance

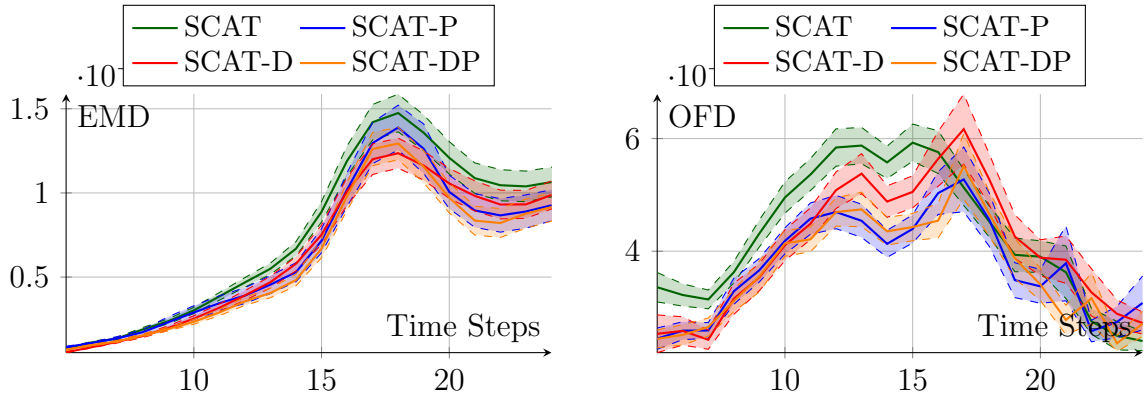

Figure 4.7: Performance of model variants over time on motion metrics, evaluated on the **Kubric-Occlusion** dataset

	Appearance			Motion		
	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
SCAT-P	25.99±0.13	0.665±0.007	0.063±0.001	0.0356±0.0016	0.0070±0.0006	11M
SCAT-D	**26.53±0.13**	**0.701±0.006**	**0.054±0.001**	0.0414±0.0019	0.0069±0.0004	11M
SCAT-PD	25.69±0.12	0.649±0.007	0.072±0.002	**0.0347±0.0014**	**0.0066±0.0004**	11M

Table 4.4: Frame prediction comparison of different SCAT variants on **Kubric-Occlusion** dataset

for appearance metrics (PSNR, SSIM & LPIPS). But in motion based metrics (OFD & EMD), SCAT-PD achieves the best results. 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. Additionally, in Figure 4.7, before the occlusion event happens (i.e., roughly before 10-th frame), all models performed similarly with minimal differences; but the difference is clear when the event of occlusion and the reappearance of the occluded object happens, that the model variants with point flow performed better. 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. Now we evaluate the ability of predicting background motion with **KITTI** dataset Figures 4.8 and 4.9 plots PSNR, SSIM LPIPS, EMD and OFD scores calculated by using different temperature parameters with different variants of our model to predict future frames. We can see that although PNSR and SSIM are decreased with the increase of temperature τ, LPIPS scores are improved overall. The result is consistent with the VQVAE reconstruction performance presented in Table 4.3. When the point flow is added, the prediction quality of the future frames are

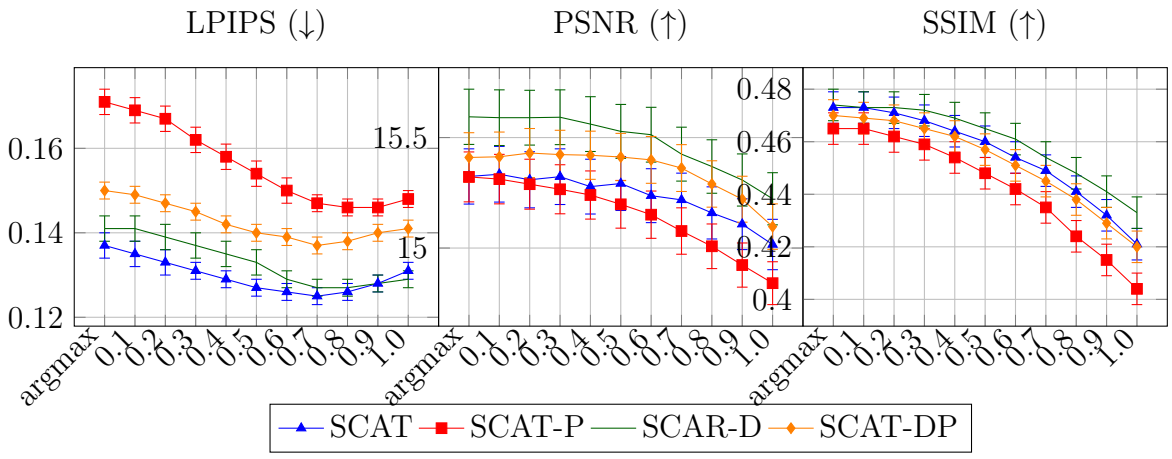

Figure 4.8: Appearance-based metrics (PSNR, SSIM, LPIPS) across temperatures on **KITTI** dataset

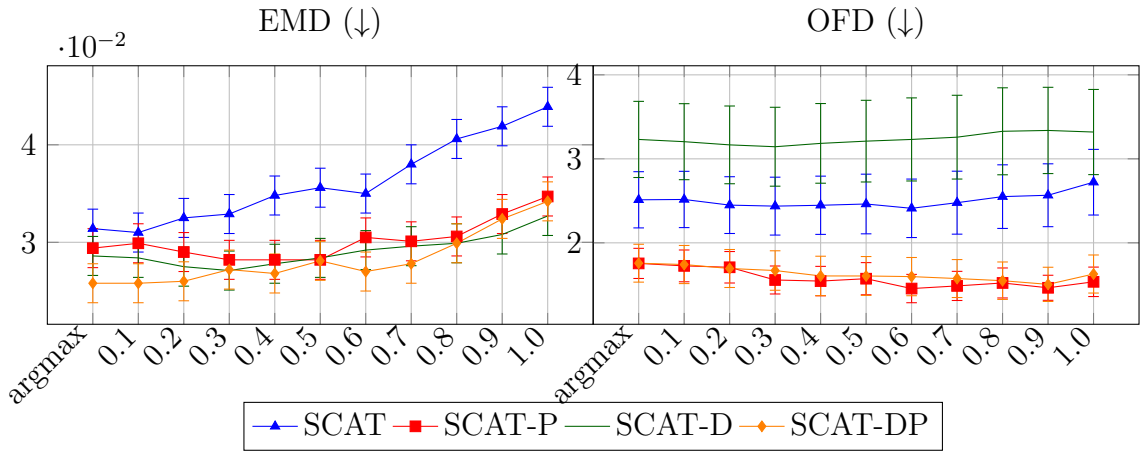

Figure 4.9: Motion-based metrics (EMD & OFD) across temperatures on **KITTI** dataset

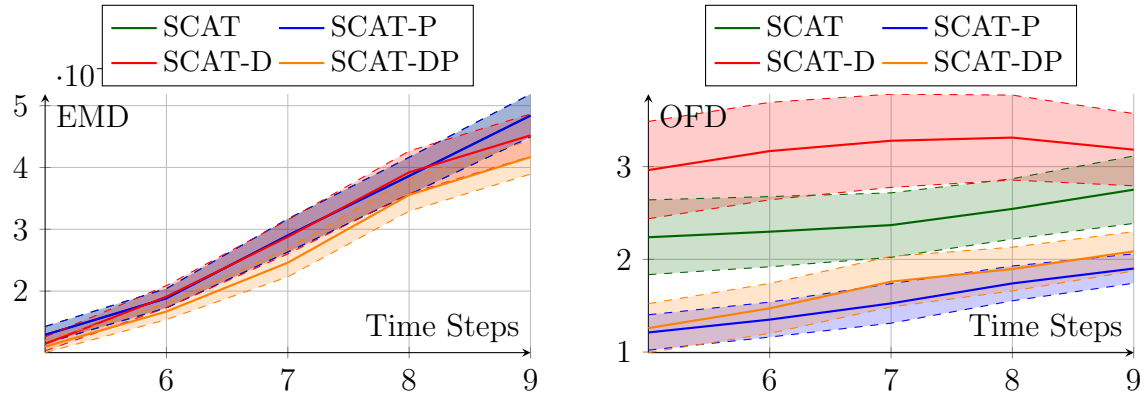

Figure 4.10: Quantitative performance of model variants over time on motion metrics, evaluated on the **KITTI** dataset

	Appearance			Motion		
	PSNR↑	SSIM↑	LPIPS↓	OFD↓	EMD↓	Prms
SCAT	15.33±0.13	0.473±0.006	0.135±0.003	2.5776±0.3341	0.0310±0.0016	8M
SCAT-P	15.20±0.11	0.448±0.006	0.155±0.003	1.6659±0.1939	0.0282±0.0020	8M
SCAT-D	15.53±0.12	0.465±0.006	**0.132±0.003**	3.2781±0.4762	0.0285±0.0022	8M
SCAT-PD	15.36±0.11	0.445±0.006	0.137±0.002	**1.6390±0.2324**	**0.0278±0.0016**	8M

Table 4.5: Frame prediction comparison of different SCAT variants on **Kubric-Occlusion** dataset

decreased in terms of appearances overall. However, when the depth map is added alone, it performs the best in both PSNR and SSIM. In contrast, when we use motion-based metric, EMD and OFD, to measure the performance of the different variants of our models, it shows the opposite. Vanilla SCAT is the worst among other variants. Furthermore, when the point flow is added with or without depth map, the motion prediction quality improves significantly compared to SCAT. Also, when both point flow and depth map are integrated together (SCAT-DP), the predictor performs the best. This evidence further suggests the integration of point flow and depth map are still effective for predicting the motion in moving backgrounds. Tables 4.6 and 4.7 provide a comparison to SimVP.

	PSNR↑	SSIM↑	LPIPS↓	OFD↓	Num-Params
SCAT	25.88±0.13	0.66±0.007	0.064±0.001	0.0423±0.0017	11M
SimVP	**33.05±0.13**	**0.95±0.001**	**0.021±0.001**	0.0626±0.0019	14M
Ours	25.69±0.12	0.65±0.007	0.072±0.002	**0.0347±0.0014**	11M

Table 4.6: Frame prediction comparison on **Kubric-Occlusion** dataset with SimVP

SimVP is a widely adopted and competitive baseline for video prediction that models scene dynamics in an implicit and holistic manner, without relying on explicit object-level

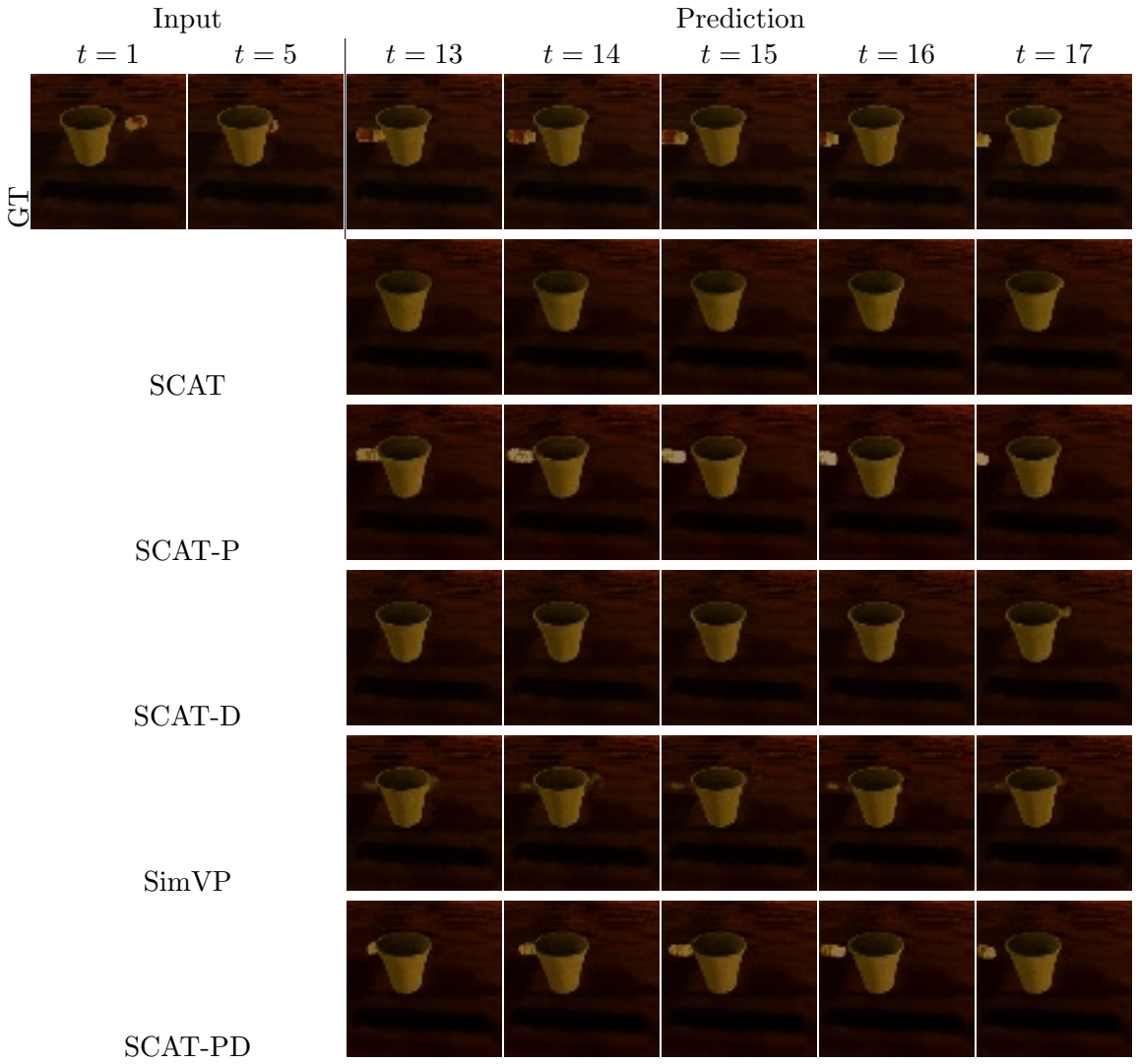

Figure 4.11: Comparison of different model variants on the **Kubric-Occlusion** dataset.

representations. It also has a light structure, similar training objectives and performs well on short term prediction (e.g., 5-10 frames). From a practical perspective, the proposed models in this chapter are computationally expensive to train and evaluate. Each model requires extensive training time across multiple datasets, and running a larger number of baselines would significantly increase the experimental duration without proportionate additional insight. In practice, evaluating four models variants across two datasets already required approximately two months of computation. Therefore, we restrict the comparison to a single representative baseline in order to balance experimental rigor with feasibility.

The proposed models appear to under-perform SimVP when looking at appearance-based metrics on the **Kubric-Occlusion** dataset, however they perform better when looking at motion-based metrics by a large margin. 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 4.11, where SCAT-P & SCAT-DP accurately predicted the motion of moving objects while others fail. Specifically, the trajectory of the moving object in Kubric-Occlusion dataset is correctly predicted only when including point flow information (SCAT-P & SCAT-PD), while SimVP fails to predict the object's reappearance. Moreover, because most of the pixels do not have motion in this dataset, therefore SimVP achieved better performance on appearance based metrics. In contrast,

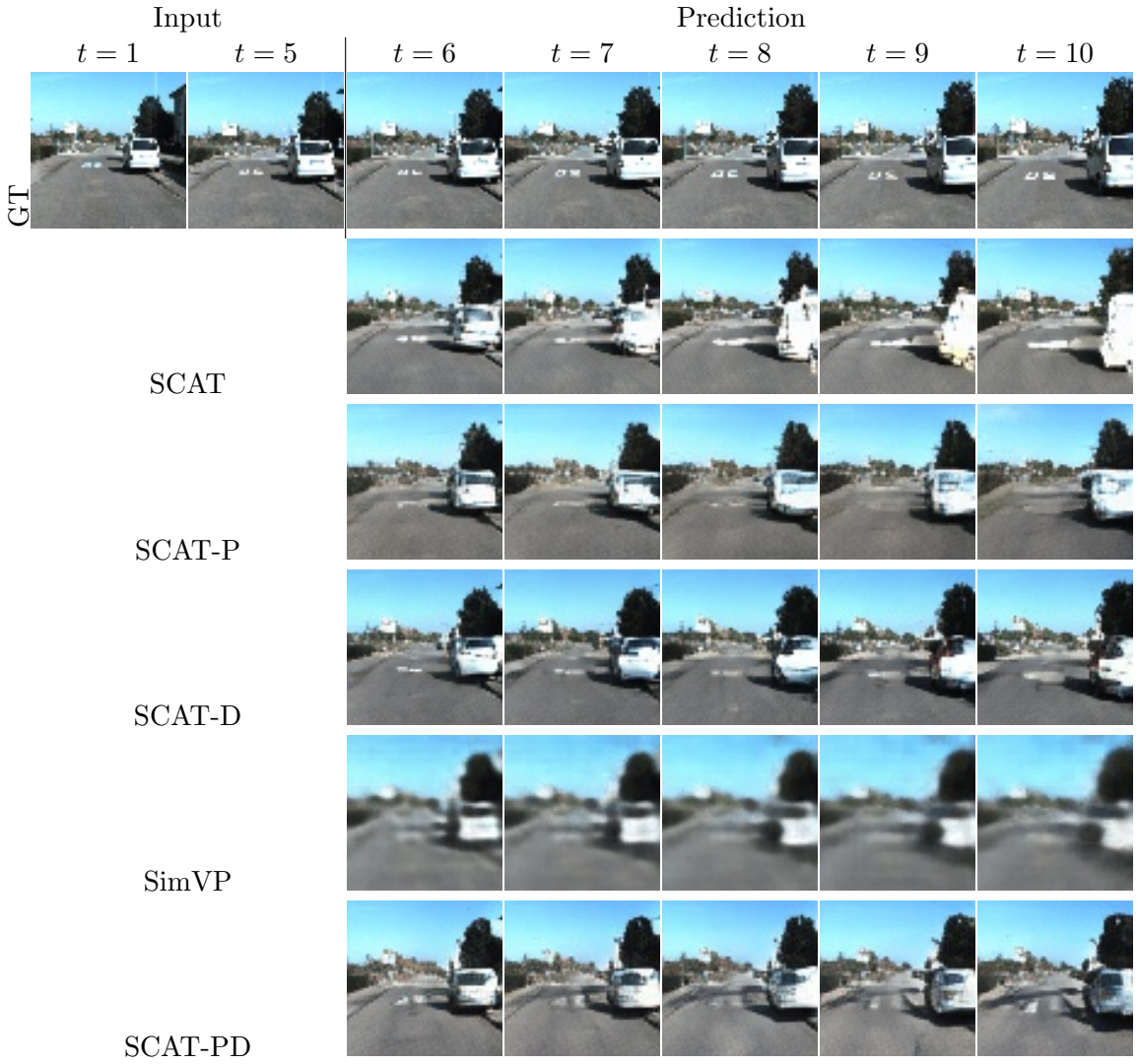

Figure 4.12: Comparison of different model variants on the **KITTI** dataset.

where KITTI features complex real world dynamics, our model outperforms SimVP in

	PSNR↑	SSIM↑	LPIPS↓	OFD↓	Num-Params
SCAT	15.33±0.13	0.47±0.006	**0.135±0.002**	2.49±0.33	8M
SimVP	**17.14±0.10**	**0.49±0.005**	0.332±0.004	1.66±0.11	14M
Ours	15.36±0.11	0.45±0.006	0.137±0.002	**1.64±0.23**	8M

Table 4.7: Frame prediction comparison on **KITTI** dataset with SimVP

LPIPS (0.137 v 0.332). Also, we see that in terms of motion our model also outperformed SimVP (1.64 v 1.66), where this can be seen in Figure 4.12, where the white car's structure across frames is more consistent with point flow and depth variants (SCAT-P, D & PD) versus RGB-only variants, and in particular, SimVP produces very blurry predictions. It is important to note that our SCAT variants are notably smaller models that SimVP and achieved similar or better performance. Another example shows that with the integration of point flow, the motion of the background is accurately predicted as illustrated in Figure 4.13. We can see SCAT and SCAT-D cannot predict the motion of ego-camera's motion which the car is turning left. We can also see that although SimVP correctly predicted the motion, but the predicted frames are very blurry compared to SCAT-P and SCAT-DP. This is also backed by the evidence in Table 4.7 that SCAT-DP performed better in LPIPS metric

4.4 Discussion

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. Experimental results show that point flow contributes to more accurate motion estimation, and in particular can successfully predict the reappearance of occluded moving objects. Furthermore, our approach is also effective in the scenarios of predicting the motion of background.

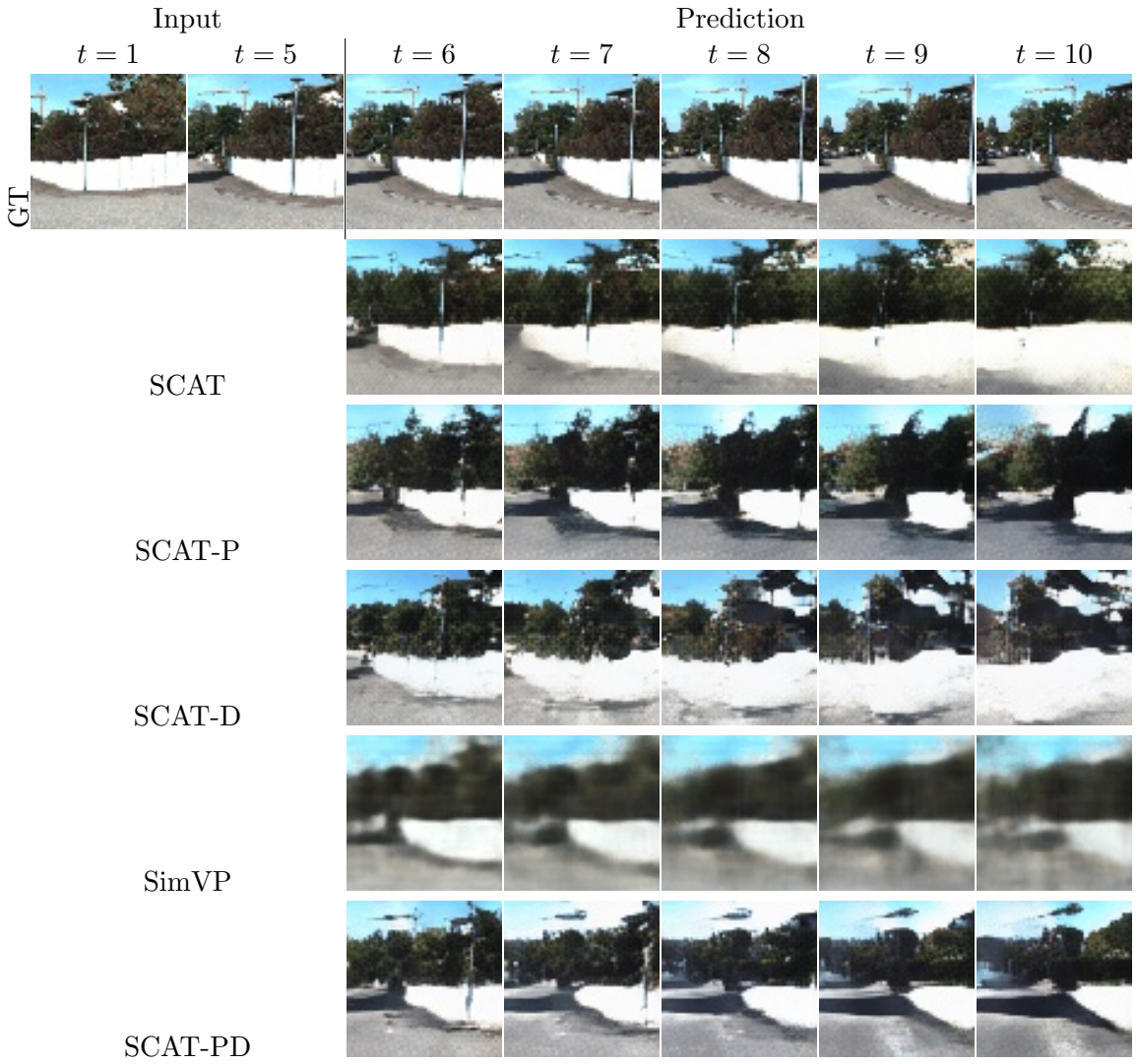

Figure 4.13: Motion accuracy of the predicted frames on KITTI dataset

While the proposed method improves motion modeling through the addition of point flow and depth to RGB inputs, its performance of reconstructing RGB frames degrades when applied to datasets with more complex distributions (e.g., KITTI), as shown in Tables 4.2 and 4.3. A likely cause is that the latent space capacity was kept constant across all variants, and the encoder architecture remained unchanged apart from the input channel size. As a result, with the increasing amount of information, compressing it to a fixed sized latent spaces is not enough to fully represent all of the different modalities. Additionally, our current cross-modal fusion strategy simply concatenates the different modalities, which leads to tradeoffs already at the encoding stages, especially with a moving camera. Another limitation is that our experiments focused mostly on rigid objects except KTH dataset in the previous chapter. Moreover, although KTH features deformable objects (a moving person), interaction is still limited in KTH dataset where only a single instance is on the

scene. Thus, the performance of the proposed models in both this chapter and the previous chapter on soft or highly deformable objects (e.g., garments) remains unknown. In the next chapter, we investigate these limitations by developing a more efficient cross-modal encoding strategy for integrating multiple modalities, and by extending our framework to model and predict the dynamics of deformable objects.

Chapter 5

Diffusion Transformer as Video Predictor

In the previous chapter, we showed that using point flow can improve the predicted object trajectory and depth can improve the appearance of the the predicted frames; integrating point flow and depth map together can enhance both appearance and motion accuracy. So far we mostly used datasets that features the motion of rigid objects such as the systhetic datasets we generated using Kubric, Real-traffic and KITTI. Although we used KTH in the first chapter, which involves a deformable object (a single person), the lack of interaction because of singular moving instance is the main limitation of this dataset. Furthermore, the motion of a human is not fully deformable, our joints srve to constraint the movement of our body to a certain degree. Thus, our proposed architectures' robustness is not tested using a kind of dataset that features the interaction of multiple soft or fully deformable objects. Therefore, we will mainly focus on prediction of the soft objects in this chapter to investigate the performance of our proposed multi-object pipeline.

The rest of the chapter is structured as follows:

- Section 5.1 gives the brief introduction of our motivation and hypothesis;
- Section 5.2 introduces our proposed method in detail;
- Section 5.3 and Section 5.4 discusses the experimental setup and the discussion of the results.

5.1 Introduction

Auto-regressive models such as SCAT have shown effectiveness in modeling structured motion of rigid objects and their interactions as shown in the previous chapters. However, their reliance on predicting discrete tokens may limit their ability to capture the fine-grained, continuous dynamics of deformable objects such as garments, where motion evolves smoothly in space and time.

On the other hand, Diffusion models (Ho et al. 2020; Peebles and Xie 2023; Blattmann et al. 2023a), operate in a continuous latent space and refine entire trajectories through iterative denoising, making them a more natural choice for modeling such continuous non-rigid dynamics. A recent work proposed a U-Net based diffusion model, it used both RGB and depth frames of a video clip to predict future frames (Pallotta et al. 2025). Although they reported promising performance on Real-World scenarios such as CityScapes (Cordts et al. 2016) that features city driving, their performance on motion prediction of deformable objects is not tested.

In this chapter, we hypothesize that diffusion model will better handle the problem of predicting the motion of highly deformable objects such as garments because of its continuous nature compared to auto-regressive GPT-style transformers. To verify this, we propose a transformer-based diffusion video prediction network SCAT-Diffusion. We build on the basis of the transformer network we used in the previous chapters. In addition, due to the limitation of degradation of reconstruction performance of the AutoEncoder we proposed in the previous chapter, we propose a more efficient variant of this autoencoder network to better encode the different modalities and aim improve reconstruction performance. We conduct systematic and detailed experiments to test each of the components we proposed in this chapter on Flat'N'Fold dataset (Zhuang et al. 2024) that features human demonstrations of garment manipulation task. Additionally, in order to test that the proposed method is also suitable for scenes of rigid objects, we test on the widely used benchmark KITTI.

5.2 Methodology

In this chapter, we will use the same problem formalism as in the previous chapters that by conditioning on a sequence of context frames to learn the distribution of future frames. We will first introduce the preliminary concepts of Denoising Diffusion Probabilistic Models (DDPM) (Ho et al. 2020) for images. Then we describe the multi-modal fusion OAAE and our diffusion based video frame predictor in details.

5.2.1 Obtaining Other Modalities

We change our point tracking model to generate point flows from Co-Tracker (Karaev et al. 2025) to Delta-Tracker (Ngo et al. 2025) which is a more recent and powerful point tracker, 10 times faster compared to Co-Tracker, and more robust on tracking the key-points on an occluded object. While Co-Tracker tracks the keypoints in 2D space, Delta-Tracker tracks the keypoints in 3D space which will give our model not only the flow of motion but also 3D geometry information complementing the depth map. For obtaining depth maps, we will use DepthAnythingV2 (Yang et al. 2024c) as in the previous chapter.

5.2.2 Frame Encoder

We build upon the OAAE encoder introduced in the previous chapter, which is based on a ResNet-18 backbone. However, in the previous design, no dedicated module existed to explicitly fuse information from multiple modalities; instead, different modalities (e.g., RGB frane, point flow and depth map) were simply concatenated as the input of OAAE. We observed that this naive fusion strategy degraded RGB reconstruction quality when the number of modalities increased especially when point flow is added.

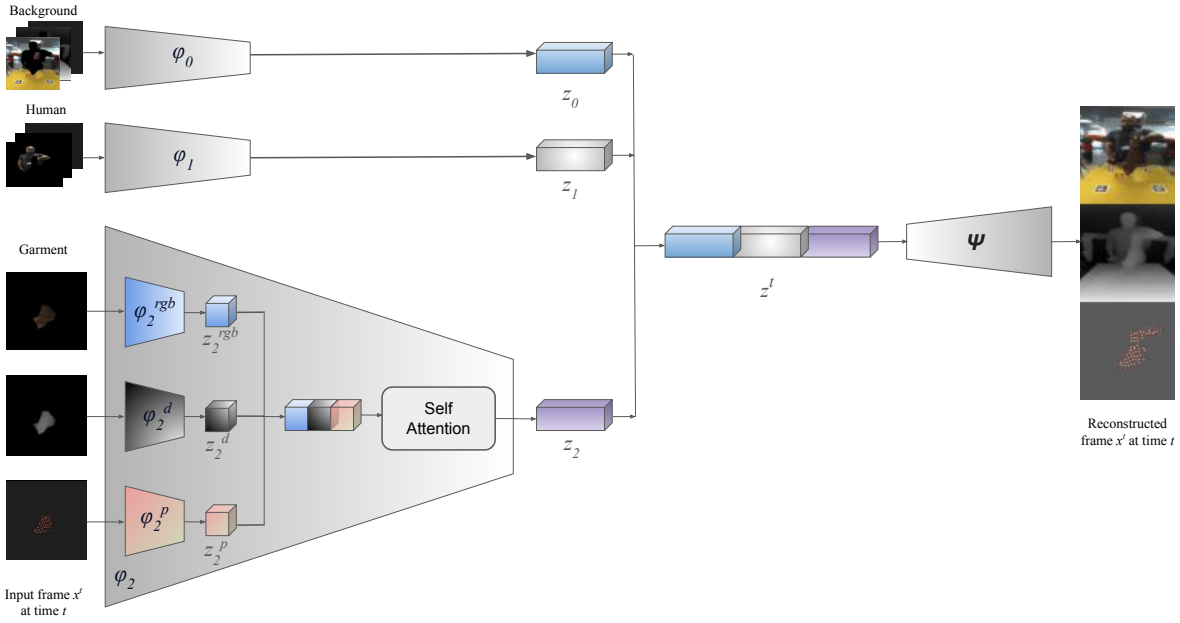

Figure 5.1: Structure of the frame encoder used in this chapter

To address this issue, we propose a multi-modal fusion encoder that more effectively integrates information across modalities. Each modality is first processed by its own dedicated encoder network to produce a set of feature maps. Concretely, for each object class m, the instance-specific encoder ϕ_m now consists of three sub-encoders, one for each modality:

$$\phi_m = \{\phi_m^{\text{rgb}}, \phi_m^d, \phi_m^p\},$$

corresponding to RGB, depth maps, and point-flow, respectively. We denote the full set of object-class encoders as $\Phi = \{\phi_1, \phi_2, \ldots, \phi_m\}$, where m is the number of object classes. Given an object k, its latent representation z_k is obtained by concatenating the latent features from all three modalities:

$$z_k = \bigoplus \left[z_k^{\text{rgb}},\ z_k^{\text{d}},\ z_k^{\text{p}} \right], \tag{5.1}$$

Then z_k is passed through a transformer-based fusion block, which models the interdependencies between each modalities. From this point, the rest of the process is identical to the previous chapters in that each instance latent z_k is concatenated to form z to represent the entire scene and each of the modalities. Then, this full latent representation is passed to the same decoder network Ψ as in the previous chapter to reconstruct all modalities

jointly. We will use the same loss function we described in the previous chapter as follows:

$$\mathcal{L} = \mathcal{L}_{VQ} + \alpha\mathcal{L}_{commitment} + \mathcal{L}_{recon} + \beta\mathcal{L}_{\text{LPIPS}} \tag{5.2}$$

5.2.3 Diffusion-based SCAT

We formulate the problem similarly to the previous chapters: to learn a probability distribution on M future frames $X^{T+1:T+M}$, conditioned on the T past frames $X^{1:T}$. Because the motion of objects like garments is smooth and continuous compared to rigid objects, we hypothesise that predicting the motion of highly deformable objects using a continuous model (SCAT-Diffusion) in a continuous latent space is better than predicting discrete indices using discrete auto-regressive models (SCAT based variants).

Therefore, we adopt a continuous model, a Diffusion Transformer (DiT) (Peebles and Xie 2023), and transform SCAT into a diffusion transformer to directly predict the future frames on the continuous latent space, which will be referred as SCAT-Diffusion in this chapter. We preserved the overall structure that uses self- and cross-attentions to learn objects' motion pattern as well as the potential interaction with other objects. Instead of predicting the probability of possible indices from the learned code-book, we directly predict the future frames based on the context frames in the latent space we learned in the previous step.

We apply the diffusion process to the future latents $Z_{T+1:T+M}$, where the forward process gradually perturbs the clean latents into Gaussian noise:

$$q(Z_t \mid Z_0) = \mathcal{N}\big(Z_t; \sqrt{\bar{\alpha}_t}\, Z_0, (1 - \bar{\alpha}_t)\mathbf{I}\big)\,. \tag{5.3}$$

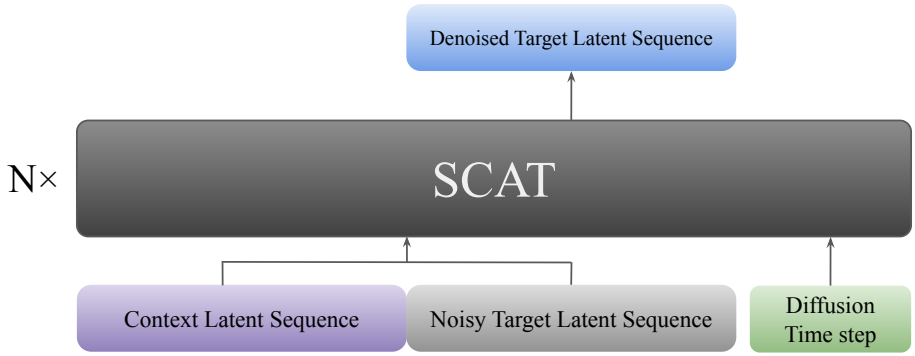

Figure 5.2: Structure of SCAT-Diffusion

At training time, SCAT-Diffusion receives the noised future latents z_t, together with the context latents $z_{1:T}$ and the diffusion timestep t, and learns to predict the added noise ϵ. The conditioning on the past frames is achieved by concatenating the past latent frames $Z_{1:T}$ with the noised future frames along the temporal dimension, allowing the model to process both jointly in a straightforward manner.

The objective of this model is same the DDPM loss, which maximizes the ELBO on the predicted latent frames' likelihood. This can be simplified to minimizing the MSE loss between the predicted noise and the original noise as shown in the equation 5.4.

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{Z_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(Z_t, t)\|^2\right] \tag{5.4}$$

Following DDPM formulation, we use 1000 diffusion steps and a linear schedule to train and validate the proposed model.

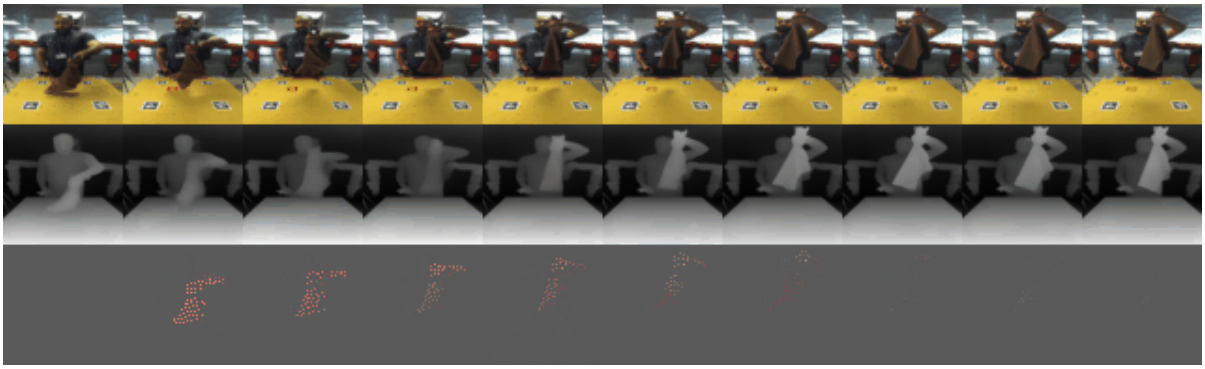

Figure 5.3: An example from the dataset that is showing a person trying to lift a napkin. **Top**: The original RGB sequence; **Middle**: Depth map of corresponding RGB frames; **Bottom**: The flow of tracked key points on RGB frames.

5.3 Experiments

5.3.1 Datasets

Flat'n'Fold is a large scale dataset for garment manipulation task (Zhuang et al. 2024). The main purpose of this dataset is to teach robots to fold a piece of garment with human and machine demonstrations. It contains 1,212 human and 887 robot demonstrations of flattening and folding 44 unique garments across 8 categories, there are 20 different individuals performed the human demonstrations. Each demonstration is stored as a video clip that has 100-500 frames depending on the type of garment being manipulated. The manipulation process is simple, first flatten a randomly placed garment on a table, then fold the garment into a desired shape. However, it involves many different types of garments with different sizes and textures. For example, napkins, T-shirts and trousers each with different texture.

Since this study aims to investigate the performance of SCAT-Diffusion and understand if it will be effective in a fully deformable objects setting, a subset of the entire dataset is selected for testing to limit the randomness. We selected all of the videos that are demonstrated by a human (i.e., no robot motion is involved) to flatten and fold a napkin. Unlike other garment types, a napkin is usually a rectangular shaped garment which is simple in shape compared to other types of garments. Napkins provide a canonical yet challen-

ging example of a fully deformable object: although simple in geometry and topology, they undergo large non-rigid deformations and frequent self-occlusions during manipulation. Moreover, the interaction involves two highly deformable entitiesthe human hands and the garmentallowing us to study rich objectobject interactions while maintaining a controlled experimental setting. Focusing on a single deformable object category therefore allows us to isolate the effect of the proposed diffusion-based predictor and attribute performance differences to the modeling choice rather than dataset heterogeneity.

There are a total of 204 videos of people manipulating napkins, we split the original video into smaller clips to 20 frames per clip, aimed to better pre-process the videos to obtain other modalities. After splitting the data, it yields 4,400 clips in total. We use 4,000 video clips to train our model and the remaining 400 clips as our validation set.

In addition to Flat'n'Fold, we evaluate our model on the KITTI dataset following exactly the same settings as in the previous chapter. Unlike Flat'n'Fold, KITTI primarily contains rigid objects with highly stochastic motion driven by complex scene dynamics and ego-motion. Including KITTI therefore serves a complementary purpose: it allows us to assess whether the proposed encoder and diffusion-based predictor generalize beyond the fully deformable regime, and to highlight the limitations of diffusion-based sampling under increased motion uncertainty. This comparison also enables a direct assessment of the proposed cross-modal fusion encoder against the encoder used in the previous chapter under identical conditions.

5.3.2 Results

The motion metrics introduced earlier for evaluating trajectory accuracy of rigid objects are not suitable for fully deformable objects such as garments in the Flat'n'Fold dataset, because the motion of both human and the garment is subtle compared to the rigid object datasets. Therefore, for Flat'n'Fold we report only appearance-based metrics: PSNR,

SSIM, and LPIPS. For evaluation on the KITTI dataset, we follow the same settings as in the previous chapters. Specifically, we report PSNR, SSIM, and LPIPS for appearance, and OFD and EMD for motion performance. All reported metric scores are obtained using bootstrapping, consistent with the procedure described in Chapter 3.

5.3.2.1 Encoder Performance

In the previous chapter, we found SCAT-PD variant performed the worst, in terms of appearance, among other variants due to the frame encoder's reconstruction performance is poor. To mitigate this problem, we proposed a more efficient way of encoding different modalities that has specific components to fuse these modalities. Therefore, before evaluating the performance of the prediction model, we first evaluate the reconstruction performance of the frame encoder proposed in this chapter and compare against the encoder we used in the previous chapter. Table 5.1 presents the reconstruction performance of the

	Depth	Point-Flow	PSNR↑	SSIM↑	LPIPS↓
OAAE-PD	✓	✓	19.957±0.090	0.695±0.003	0.063±0.001
Fusion-OAAE-PD	✓	✓	**21.013±0.101**	**0.754±0.003**	**0.045±0.001**

Table 5.1: Reconstruction performance comparison of OAAE-PD and Fusion-OAAE-PD on **KITTI** dataset

baseline OAAE-PD encoder and the proposed Fusion-OAAE-PD encoder on the **KITTI** dataset. Across all metrics, the Fusion-OAAE-PD consistently outperforms OAAE-PD, achieving a notable increase in PSNR and SSIM, along with a substantial reduction in LPIPS. Specifically, the Fusion-OAAE-PD improves PSNR by over one point and SSIM by nearly 0.06, while cutting perceptual error (LPIPS) by more than 25%.

These results confirm that the newly introduced fusion module is significantly more effective than the simple concatenation approach used in the previous chapter. By explicitly modeling the relationships across modalities before reconstruction, the Fusion-OAAE-PD leverages complementary information more efficiently, leading to sharper, more perceptually accurate reconstructions. This improvement directly addresses the limitation observed

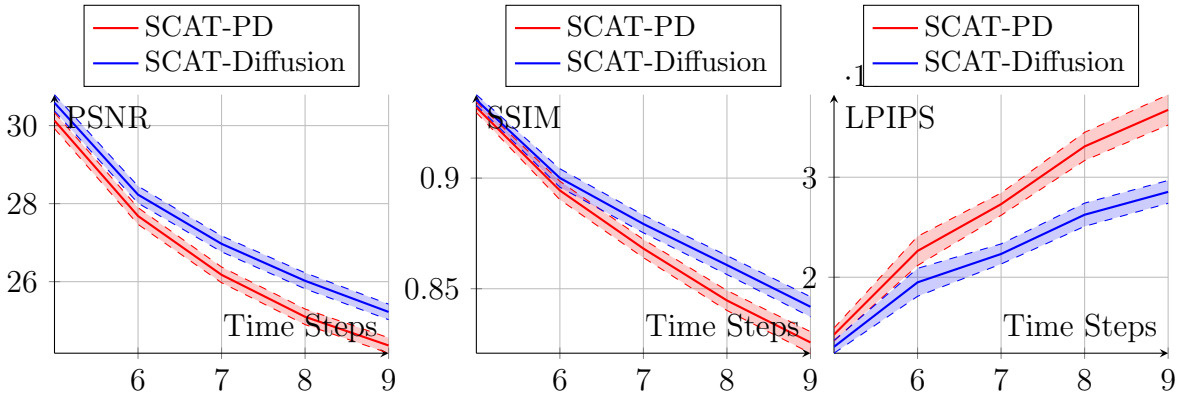

Figure 5.4: Performance of model variants over time on time metrics, evaluated on the **Flat'n'Fold** dataset

in the previous chapter, where the SCAT-PD variant suffered from poor appearance prediction performance due to bad frame reconstructions. Consequently, the enhanced frame encoder provides a stronger foundation for the prediction model, as improved input representations are expected to translate into better temporal dynamics modeling in the subsequent experiments.

5.3.2.2 Prediction performance

After we showed our new encoding strategy is working efficiently, we will now use this new latent space to run our diffusion-based prediction model on Flat'N'Fold dataset to test its future frame prediction performance on highly deformable objects. To ensure fair comparison and isolation of the variable of using diffusion architecture, we used the new latent space for both SCAT-PD, which is proposed in the previous chapter, and SCAT-Diffusion. We let both models take five context frames and they are required to predict five future frames. Because each model is stochastic, we sample 10 times to select the best performing predicted frames for comparison. The diffusion step is set to 1000 steps. The

	Appearance			
	PSNR↑	SSIM↑	LPIPS↓	Prms
SCAT-PD	26.69±0.19	0.873±0.004	0.027±0.001	100M
SCAT-Diffusion	**27.41±0.19**	**0.883±0.003**	**0.022±0.001**	102M

Table 5.2: Comparison of prediction performance on **Flat'n'Fold** dataset

quantitative results on the **FlatnFold** dataset are presented in Table 5.2. SCAT-Diffusion

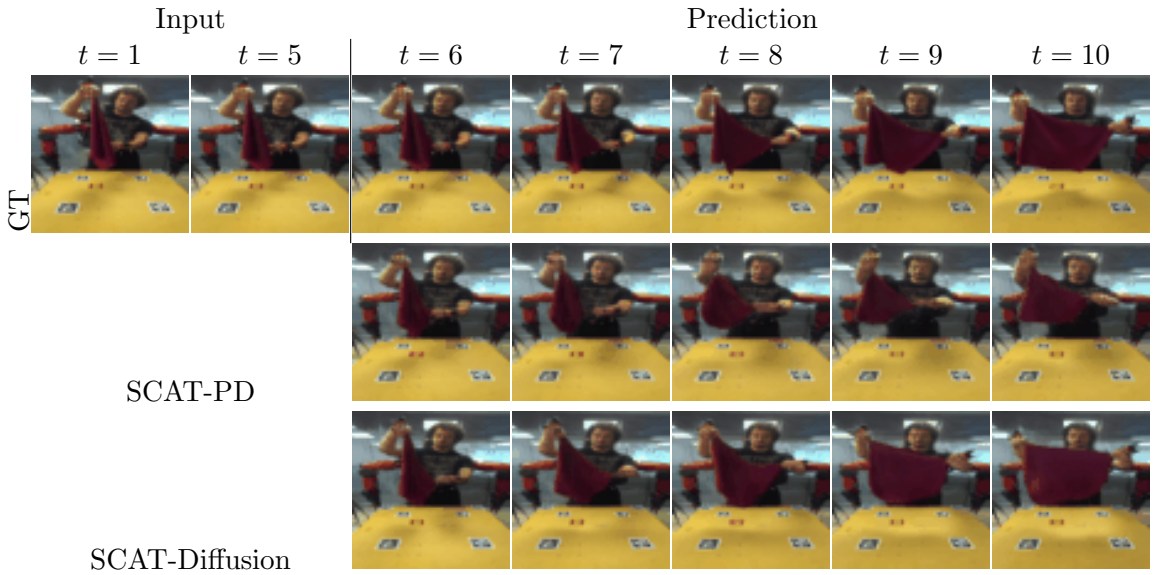

Figure 5.5: Comparison of different model variants on the **Flat'n'Fold** dataset (1)

achieves consistent improvements over SCAT-PD across all appearance-based metrics, with notable gains in PSNR (26.69 vs 27.41) and LPIPS (0.027 vs 0.022), indicating sharper and more perceptually better predictions. The qualitative results further highlight the advantages of SCAT-Diffusion in modeling deformable object dynamics. In Figure 5.5, SCAT-Diffusion accurately predicts that the person unfolds the napkin, producing a prediction close to the ground truth, whereas SCAT-PD struggles to capture this motion and incorrectly predicts that the napkin remains unchanged. Similarly, in Figure 5.6, SCAT-Diffusion successfully predicts the folding motion of the person and the napkin, while SCAT-PD fails to model this interaction and ultimately causes the persons hand to freeze in the predicted sequence. Furthermore, as this napkin has two different colours on different sides, SCAT-Diffusion also correctly predicted the correct white colour in the future frames. These examples demonstrate that SCAT-Diffusion not only generates sharper frames but also captures complex motion patterns of both humans and deformable objects better than its discrete counterpart.

For an additional evaluation on SCAT-Diffusion's prediction performance, we also compare it **KITTI** dataset as this dataset features the opposite motion type compared to **Flat'n'Fold** dataset. In terms of appearance-based metrics, SCAT-Diffusion shows clear improvements in PSNR (15.36 vs 15.72) and LPIPS (0.137 vs 0.111), indicating higher fidelity and perceptual quality. However, it underperforms in SSIM (0.445 vs 0.439), sug-

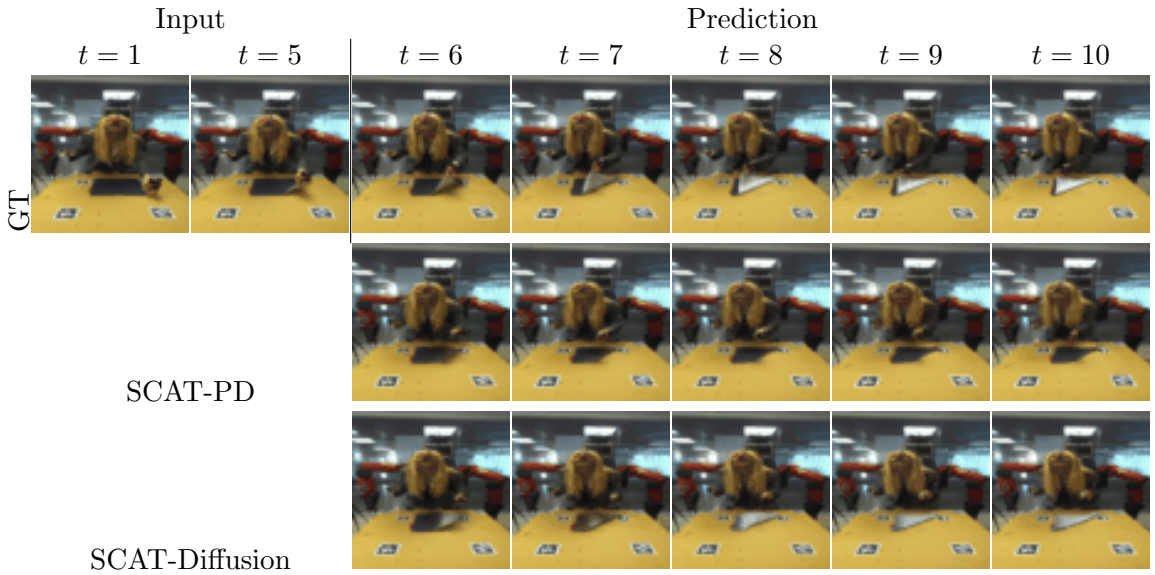

Figure 5.6: Comparison of different model variants on the **Flat'n'Fold** dataset (2)

		Appearance			Motion		
	Fusion-Encoder	PSNR↑	SSIM↑	LPIPS↓	OFD↓	EMD↓	Prms
SCAT-PD	✗	15.36±0.11	0.445±0.006	0.137±0.002	1.64±0.23	0.0278±0.0016	8M
SCAT-PD	✓	**16.02±0.11**	**0.491±0.006**	0.134±0.002	**1.24±0.10**	0.0278±0.0018	8M
SCAT-Diffusion	✓	15.72±0.11	0.439±0.006	**0.111±0.002**	1.29±0.10	0.0402±0.0022	10M

Table 5.3: Comparison of prediction performance on **KITTI** dataset

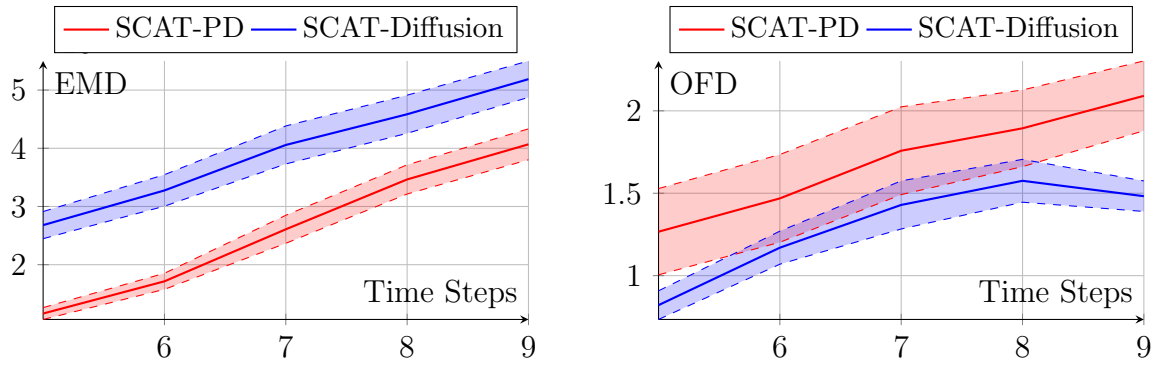

Figure 5.7: Performance of model variants over time on motion metrics, evaluated on the **KITTI** dataset

gesting that structural consistency is not preserved as effectively. For motion-based metrics, SCAT-Diffusion achieves a significantly lower OFD (1.29 vs 1.64), reflecting better alignment of predicted motion with ground truth trajectories. In contrast, on the fine-grained EMD metric, our approach is less effective, performing worse than SCAT-PD. This is evident in Figure 5.9, which shows the decoded masks for each car instance in the scene. We can see that even the predicted RGB frames are visually better than SCAT-PD, the masks decoded by the predicted latent space shows very noisy masks. This is

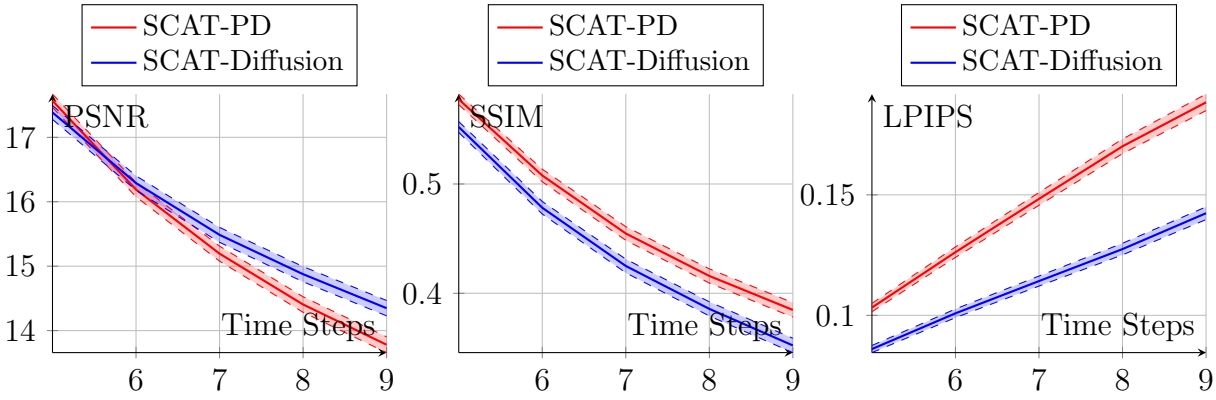

Figure 5.8: Performance of model variants over time on appearance metrics, evaluated on the **Flat'n'Fold** dataset

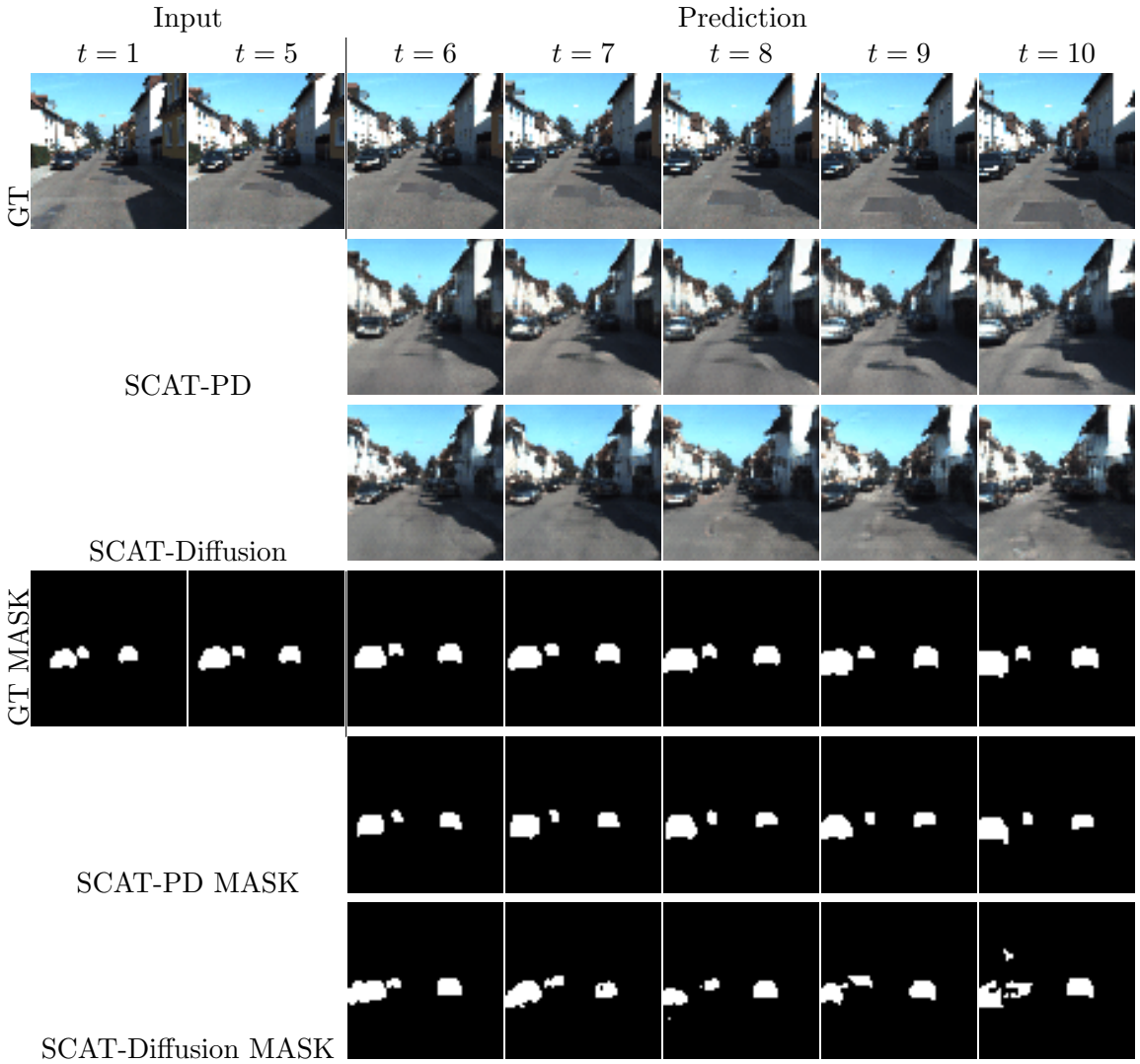

Figure 5.9: Comparison of different model variants on the **KITTI** dataset

likely due to the prediction process of the diffusion model that uses diffusion process to gradually revert a sampled noise back to the latent space. However, as our main focus is testing SCAT-Diffusion in predicting highly deformable objects and our evaluation on these type of dataset showed SCAT-Diffusion is more effective.

5.4 Discussion

In this chapter, we hypothesized that the diffusion-based prediction models can better capture the motion patterns of a fully deformable objects. We introduced a fusion module within the frame encoder and demonstrated that it achieves more accurate video frame reconstruction compared to the encoder used in the previous chapter, which lacked a dedicated multi-modality fusion mechanism. Building on this improved encoder, we proposed a diffusion-based video prediction model SCAT-Diffusion, focusing primarily on scenarios involving fully deformable objects using the FlatnFold dataset. We conducted a direct comparison with the SCAT-PD model from the previous chapter while maintaining the same latent space. Our results show that SCAT-Diffusion consistently outperforms SCAT-PD, suggesting that diffusion-based approaches are particularly effective for modeling the future dynamics of fully deformable objects.

Additional experiments on the KITTI dataset revealed more nuanced behavior. Although SCAT-Diffusion improved appearance-based reconstruction metrics compared to SCAT-PD, it performed significantly worse on the fine-grained motion metric (EMD). This performance gap is reflected in the decoded masks, where SCAT-Diffusion produced noisy and less coherent object shapes. We attribute this degradation to two main factors: (i) the KITTI dataset exhibits significantly more stochastic object motions compared to FlatnFold, and (ii) SCAT-Diffusion attempts to approximate the latent vector directly through the reverse diffusion process, whereas SCAT-PD samples discrete indices directly from a constrained codebook, which may provide a more stable representation in highly uncertain scenarios.

5.4.1 Limitations

While SCAT-Diffusion shows promising performance on FlatnFold, our evaluation was restricted to a single scenario, folding a napkin from a random configuration into a target shape. Future work should extend this evaluation to the full FlatnFold dataset, which includes a broader range of manipulation tasks such as folding shirts and trousers, to better assess the models generalization capabilities.

Furthermore, the results on the KITTI dataset reveal another key limitation: when object motion is highly stochastic, the predicted latent space becomes noisy, leading to heavily distorted decoded masks. This highlights the need for more advanced sampling strategies to better control stochasticity during the reverse diffusion process. Finally, the current model used for KITTI experiments is relatively lightweight, containing only 10M parameters. While performance may improve with larger model capacity as we demonstrated in the evaluation on Flat'n'Fold dataset with 100M parameters, it is noteworthy that even with similar capacity to SCAT-PD, SCAT-Diffusion underperforms, indicating that architectural changes may be required rather than simply scaling the model size.

Chapter 6

Conclusions & Discussions

6.1 Validation of Thesis Statement

In Section 1.2, we presented three main claims about the problem we want to focus in this thesis. We will validate these claims in this section.

- **Claim 1:** ***Explicit object decomposition and learning the relationships between decomposed objects improves the quality of predicted future frames. Moreover, incorporating a cross-attention mechanism to capture potential object interactions further enhances prediction quality.*** In Chapter 3, we introduced a family of video prediction models built on a two-stage pipeline: encoding video frames into a latent space and predicting future frames within this space. To validate this claim, we applied off-the-shelf semantic segmentation models to decompose scenes into objects of interest and focused on predicting their dynamics. We then evaluated the proposed models across five datasets that span weak and strong interaction scenarios. Our results consistently showed that SCAT (which uses full decomposition and cross-attention) outperforms the non-decomposed variant (SiS) in both qualitative and quantitative evaluations. Furthermore, we found that SNCAT (which includes decomposition but omits cross-attention) performs worse than SCAT, highlighting the crucial role of cross-attention. These findings collectively confirm our first claim.

- **Claim 2:** ***Integrating explicit motion information such as point flow and depth maps is beneficial for capturing specific dynamics, including occlusions and background motion.*** In Chapter 3, we observed limitations in handling fully occluded objects and background motion when relying solely on RGB information. To address this, Chapter 4 integrated explicit motion information, point flow and depth maps, that provide 3D geometry, relative position, and motion direction. Through systematic experimentation, we found that point flow is the most significant contributor, enabling the model to predict background and occluded object motion more accurately. In contrast, using only RGB or RGB combined with depth yielded inferior motion predictions such as inability to predict the reappearance of fully occluded objects and failure to capture the overall background motion direction. These findings provide clear evidence to our second claim.
- **Claim 3:** ***Continuous models, such as diffusion models, outperform discrete models in scenarios involving highly deformable objects, such as garments.*** To evaluate this claim, we extended the GPT-style autoregressive transformer into a diffusion-based transformer in Chapter 5. We tested this model on the FlatnFold dataset, which features interactions between a person and deformable garments. Using the same frame encoder we proposed in Chapter 5, which could better integrate the different modalities of a single frame, for fair comparison, we changed only the prediction network between the model we proposed in Chapter 4 (SCAT-DP). Our experiments demonstrated that SCAT-Diffusion consistently outperforms the SCAT-DP in both appearance- and motion-based metrics. These results provide strong support for our final claim.

Our findings throughout this thesis establish that object motion in dynamic scenes is fundamentally better modeled through explicit object decomposition using powerful off-the-shelf segmentation models. This explicit object-centric approach is more efficient than monolithic video prediction models in terms of both computational cost and model size, even when segmentation models occasionally make errors. Furthermore, our work demonstrates that multi-modality is essential for capturing complex dynamics such as occlusions and background motion from moving cameras, capabilities that single-modality ap-

proaches fundamentally cannot achieve. Finally, we show that diffusion-based variant of our approach is particularly well-suited for predicting the dynamics of fully deformable objects, revealing important insights about matching model architectures to physical phenomena. These contributions collectively establish a new paradigm for video prediction that leverages the modularity and power of existing foundation models rather than attempting to solve all aspects of the problem end-to-end. Our approach is particularly effective in static-camera settings with a limited number of objects that the scenarios common in robotic arm manipulation tasks. By demonstrating that superior performance can be achieved with lightweight, modular architectures, this work provides a practical foundation for deploying video prediction in real-world applications where computational efficiency and reliability are paramount.

6.2 Limitations of This Thesis

Although we progressively addressed the limitations of each technical chapter throughout the thesis, for example, improving the motion modeling of SCAT in Chapter 4 compared to the vanilla SCAT in Chapter 3, and enhancing the frame encoding strategy to better fuse multi-modal information in Chapter 5 while maintaining reconstruction quality, there remain several inherent challenges that this work does not resolve. The main limitations of this thesis are as follows:

- **Reliance on external semantic segmentation models.** Our methods depend on off-the-shelf segmentation networks for object decomposition. Although we simulated the possible errors made by the segmentation models by dilation and erosion (over- and under-segmentation) and achieved relatively better performance when the kernel size of dilation or erosion is small compared to non-decomposed approaches in Chapter 3, its performance consistently decreases when the size of the kernels becomes bigger.

- **Pre-defined object classes.** The decomposition strategy used in this thesis assumes a fixed set of object categories, which restricts the models ability to generalize to unseen or novel objects. This constraint limits the applicability of the approach to more open-world scenarios where object categories may not be known. For example in Chapter 5, we defined the class of interesting objects are the person and the garment being manipulated by the person, thus everything else goes into the background. Thus, the segmentation model will also only segment the predefined objects. As a result, if the person starts to manipulate a new object that is out of the scope of predefined objects, then this new object will be treated as part of the background and its motion will be learned implicitly with the background. This limits the adaptability of our model in a open-world setting.

6.3 Future Work

Based on the limitations of this thesis, there are several potential future research direction can be continued upon the basis of this work.

- **Efficient interaction modeling.** While the cross-attention module effectively captures pairwise interactions between objects, it remains computationally expensive and will grow exponentially with the increase of the number of objects in a scene. This limits the deployability of our model on a low budget scenario and the inference in the real-time. Future work could investigate more efficient interaction mechanisms, such as restricting attention to spatially or semantically relevant neighbors. For example, a distance-based threshold could be introduced to omit attention between objects that are far apart and unlikely to interact, thereby reducing unnecessary computation. Alternative architectures, such as graph neural networks (Scarselli et al. 2009) or locality-sensitive attention mechanisms Kitaev et al. 2020, may also provide a more efficient yet expressive means of modeling interactions.

- **Improved conditioning on past frames.** In Chapter 5, past frames are incorporated by simple concatenation with the noisy target frames, leaving the burden of learning temporal dependencies to the self-attention mechanism. While effective to some extent, this approach does not explicitly leverage the temporal and spatial structure of the conditioning frames. Future work could explore more principled conditioning strategies, such as cross-attention between observed and target frames. These methods could provide richer and more explicit alignment between past observations and predicted futures, improving both accuracy and sample efficiency.
- **Generalization to open-world settings.** A major limitation of the current framework is its reliance on a pre-defined set of object classes. This restricts the models applicability in real-world environments, where novel or previously unseen objects frequently appear. When such objects are encountered, their dynamics are only captured implicitly as part of the background, limiting interpretability and predictive accuracy.

6.4 Final Remarks

In this thesis, we have demonstrated that explicitly decomposing objects in a dynamic scene and modeling their individual dynamics is not merely a niche idea, but a practical and effective design choice that can bring substantial benefits to video prediction models. Across all proposed approaches, our models were able to generate visually coherent and physically plausible future frame predictions from observed past frames. We believe that the findings of this research provide a solid foundation for advancing video prediction applications. In particular, we highlight the relevance to robotics: as shown in Chapter 5, the SCAT-Diffusion model showed potential in a robotic manipulation setting with a limited number of objects. By accurately forecasting the future states of the manipulated objects, our approach can support more reliable planning and control in robotic systems. However, our model has many inherent limitations as we stated in Section 6.2 which provides promising future research directions.

Bibliography

Achiam, Josh et al. (2023). 'Gpt-4 technical report'. In: *arXiv preprint arXiv:2303.08774*.

Anthropic (2025). *Claude Sonnet 4*. Large language model. URL: `https://claude.ai/`.

Arnab, Anurag et al. (2021). 'Vivit: A video vision transformer'. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846.

Azinović, Dejan et al. (2022). 'Neural rgb-d surface reconstruction'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6290–6301.

Bai, Xuyang et al. (2022). 'Transfusion: Robust lidar-camera fusion for 3d object detection with transformers'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1090–1099.

Baldi, Pierre (2012). 'Autoencoders, unsupervised learning, and deep architectures'. In: *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings, pp. 37–49.

Beck, Maximilian et al. (2024). 'xlstm: Extended long short-term memory'. In: *Advances in Neural Information Processing Systems* 37, pp. 107547–107603.

Bei, Xinzhu, Yanchao Yang and Stefano Soatto (2021). 'Learning semantic-aware dynamics for video prediction'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 902–912.

Bharadhwaj, Homanga et al. (2024). 'Track2Act: Predicting Point Tracks from Internet Videos enables Generalizable Robot Manipulation'. In: *European Conference on Computer Vision (ECCV)*.

Blattmann, Andreas et al. (2023a). 'Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models'. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Blattmann, Andreas et al. (2023b). 'Stable video diffusion: Scaling latent video diffusion models to large datasets'. In: *arXiv preprint arXiv:2311.15127*.

Brooks, Tim et al. (2024). 'Video generation models as world simulators'. In.

Carion, Nicolas et al. (2020). 'End-to-end object detection with transformers'. In: *European conference on computer vision*. Springer, pp. 213–229.

Caron, Mathilde et al. (2021). 'Emerging properties in self-supervised vision transformers'. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660.

Cetinkaya, Bedrettin, Sinan Kalkan and Emre Akbas (2022). 'Does depth estimation help object detection?' In: *Image and vision computing* 122, p. 104427.

Chan, Eric R et al. (2022). 'Efficient geometry-aware 3d generative adversarial networks'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16123–16133.

Chang, Zheng et al. (2022). 'STAM: A SpatioTemporal Attention based Memory for Video Prediction'. In: *IEEE Transactions on Multimedia*, pp. 1–1. DOI: 10.1109/TMM.2022.3146721.

Chefer, Hila et al. (2025). 'VideoJAM: Joint Appearance-Motion Representations for Enhanced Motion Generation in Video Models'. In: *arXiv preprint arXiv:2502.02492*.

Chen, Xiaozhi et al. (2017). 'Multi-view 3d object detection network for autonomous driving'. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915.

Cho, Kyunghyun et al. (2014). 'On the properties of neural machine translation: Encoder-decoder approaches'. In: *arXiv preprint arXiv:1409.1259*.

Cho, Seokju et al. (2024). 'FlowTrack: Revisiting Optical Flow for Long-Range Dense Tracking'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19268–19277.

Comanici, Gheorghe et al. (2025). 'Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities'. In: *arXiv preprint arXiv:2507.06261*.

Cordts, Marius et al. (2016). 'The Cityscapes Dataset for Semantic Urban Scene Understanding'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

DeepMind, Google (2025). *Veo 3*. URL: `https://deepmind.google/models/veo/`.

Denton, Emily and Rob Fergus (2018). 'Stochastic video generation with a learned prior'. In: *International conference on machine learning*. PMLR, pp. 1174–1183.

Denton, Emily L et al. (2017). 'Unsupervised learning of disentangled representations from video'. In: *Advances in neural information processing systems* 30.

Devlin, Jacob et al. (2019). 'Bert: Pre-training of deep bidirectional transformers for language understanding'. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186.

Dosovitskiy, Alexey et al. (2015). 'Flownet: Learning optical flow with convolutional networks'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766.

Dosovitskiy, Alexey et al. (2021). 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. In: *International Conference on Learning Representations*.

Ehrhardt, Sébastien et al. (2020). 'RELATE: Physically Plausible Multi-Object Scene Synthesis Using Structured Latent Spaces'. In: *Advances in Neural Information Processing Systems*.

Engelcke, Martin et al. (2020). 'GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations'. In: *International Conference on Learning Representations*.

Esser, Patrick, Robin Rombach and Bjorn Ommer (2021). 'Taming transformers for high-resolution image synthesis'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883.

Farnebäck, Gunnar (2003). '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*. Springer, pp. 363–370.

Gao, Zhangyang et al. (2022). 'Simvp: Simpler yet better video prediction'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3170–3180.

Geiger, Andreas et al. (2013). 'Vision meets robotics: The kitti dataset'. In: *The International Journal of Robotics Research* 32.11, pp. 1231–1237.

Godard, Clément, Oisin Mac Aodha and Gabriel J Brostow (2017). 'Unsupervised monocular depth estimation with left-right consistency'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 270–279.

Goodfellow, Ian J et al. (2014). 'Generative adversarial nets'. In: *Advances in neural information processing systems* 27.

Greff, Klaus et al. (2022). 'Kubric: A scalable dataset generator'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761.

Guan, He, Chunfeng Song and Zhaoxiang Zhang (2025). 'LiDAR-camera Cooperative Semantic Segmentation'. In: *Machine Intelligence Research*, pp. 1–13.

Gupta, Agrim et al. (2022). 'Maskvit: Masked visual pre-training for video prediction'. In: *arXiv preprint arXiv:2206.11894*.

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

Hazirbas, Caner et al. (2016). 'Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture'. In: *Asian conference on computer vision*. Springer, pp. 213–228.

He, Kaiming et al. (2016). 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, Kaiming et al. (2017). 'Mask r-cnn'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

He, Kaiming et al. (2022). 'Masked autoencoders are scalable vision learners'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009.

Henderson, Paul and Christoph H. Lampert (2020). 'Unsupervised object-centric video generation and decomposition in 3D'. In: *Advances in Neural Information Processing Systems (NeurIPS) 33*.

Henderson, Paul, Christoph H. Lampert and Bernd Bickel (2021). *Unsupervised Video Prediction from a Single Frame by Estimating 3D Dynamic Scene Structure*. arXiv:2106.09051.

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

Ho, Jonathan et al. (2022). 'Video Diffusion Models'. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., pp. 8633–8646.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). 'Long short-term memory'. In: *Neural computation* 9.8, pp. 1735–1780.

Höppe, Tobias et al. (2022). 'Diffusion Models for Video Prediction and Infilling'. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856.

Horé, Alain and Djemel Ziou (2010). 'Image Quality Metrics: PSNR vs. SSIM'. In: *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369. DOI: 10.1109/ICPR.2010.579.

Hsieh, Jun-Ting et al. (2018). 'Learning to Decompose and Disentangle Representations for Video Prediction'. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc.

Ilg, Eddy et al. (2017). 'Flownet 2.0: Evolution of optical flow estimation with deep networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2462–2470.

Jarzynski, Christopher (1997). 'Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach'. In: *Physical Review E* 56.5, p. 5018.

Jeong, Hyeonho et al. (2024). 'Track4Gen: Teaching Video Diffusion Models to Track Points Improves Video Generation'. In: *arXiv preprint arXiv:2412.06016*.

Jiang, Jindong et al. (2019). 'SCALOR: Generative World Models with Scalable Object Representations'. In: *International Conference on Learning Representations*.

Jiang, Liming et al. (2021). 'Focal Frequency Loss for Image Reconstruction and Synthesis'. In: *International Conference on Computer Vision*.

Karaev, Nikita et al. (2025). 'Cotracker: It is better to track together'. In: *European Conference on Computer Vision*. Springer, pp. 18–35.

Karras, Tero, Samuli Laine and Timo Aila (2019). 'A style-based generator architecture for generative adversarial networks'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410.

Karras, Tero et al. (2020). 'Analyzing and improving the image quality of stylegan'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119.

Kingma, Diederik P and Max Welling (2013). 'Auto-encoding variational bayes'. In: *arXiv preprint arXiv:1312.6114*.

Kipf, Thomas et al. (2022). 'Conditional Object-Centric Learning from Video'. In: *International Conference on Learning Representations (ICLR)*.

Kirillov, Alexander et al. (2023). 'Segment anything'. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026.

Kitaev, Nikita, Lukasz Kaiser and Anselm Levskaya (2020). 'Reformer: The Efficient Transformer'. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=rkgNKkHtvB`.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2012). 'Imagenet classification with deep convolutional neural networks'. In: *Advances in neural information processing systems* 25.

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

Lee, Dongho, Jongseo Lee and Jinwoo Choi (2023). 'CAST: Cross-Attention in Space and Time for Video Action Recognition'. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., pp. 79399–79425.

Lee, Wonkwang et al. (2021). 'Revisiting Hierarchical Approach for Persistent Long-Term Video Prediction'. In: *International Conference on Learning Representations*.

Li, Nanbo et al. (2021). 'Object-Centric Representation Learning with Generative Spatial-Temporal Factorization'. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., pp. 10772–10783.

Li, Yijun et al. (2018). 'Flow-grounded spatial-temporal video prediction from still images'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 600–615.

Li, Yingwei et al. (2022). 'Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17182–17191.

Liang, Feng et al. (2024). 'Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8207–8216.

Lin, Xinmiao et al. (2023). 'Catch Missing Details: Image Reconstruction with Frequency Augmented Variational Autoencoder'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Liu, Yixin et al. (2024). 'Sora: A review on background, technology, limitations, and opportunities of large vision models'. In: *arXiv preprint arXiv:2402.17177*.

Liu, Ze et al. (2021). 'Swin transformer: Hierarchical vision transformer using shifted windows'. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022.

Liu, Ze et al. (2022). 'Video swin transformer'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3202–3211.

Locatello, Francesco et al. (2020). 'Object-centric learning with slot attention'. In: *Advances in neural information processing systems* 33, pp. 11525–11538.

Lu, Wei et al. (2021). 'A video prediction method based on optical flow estimation and pixel generation'. In: *IEEE Access* 9, pp. 100395–100406.

Lu, Yawen et al. (2023). 'Transflow: Transformer as flow learner'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18063–18073.

Luo, Weixin et al. (2021). 'Future frame prediction network for video anomaly detection'. In: *IEEE transactions on pattern analysis and machine intelligence* 44.11, pp. 7505–7520.

Lüddecke, Timo and Alexander Ecker (2022). 'Image Segmentation Using Text and Image Prompts'. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7076–7086. DOI: 10.1109/CVPR52688.2022.00695.

Makhzani, Alireza et al. (2015). 'Adversarial autoencoders'. In: *arXiv preprint arXiv:1511.05644*.

Neal, Radford M (2001). 'Annealed importance sampling'. In: *Statistics and computing* 11.2, pp. 125–139.

Neutelings, Izaak (2015–2025). *TikZ Graphs for Neural Networks*. URL: `https://tikz.net/neural_networks/`.

Ngo, Tuan Duc et al. (2025). 'DELTA: DENSE EFFICIENT LONG-RANGE 3D TRACKING FOR ANY VIDEO'. In: *The Thirteenth International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=d9iHI1eimo`.

Ohayon, Guy et al. (2023). 'Reasons for the superiority of stochastic estimators over deterministic ones: Robustness, consistency and perceptual quality'. In: *International Conference on Machine Learning*. PMLR, pp. 26474–26494.

Oord, Aaron van den, Oriol Vinyals and Koray Kavukcuoglu (2017). 'Neural Discrete Representation Learning'. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc.

Oprea, Sergiu et al. (2020). 'A review on deep learning techniques for video prediction'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.6, pp. 2806–2826.

Oquab, Maxime et al. (2023). 'Dinov2: Learning robust visual features without supervision'. In: *arXiv preprint arXiv:2304.07193*.

Pallotta, Enrico et al. (2025). 'SyncVP: Joint Diffusion for Synchronous Multi-Modal Video Prediction'. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13787–13797.

Peebles, William and Saining Xie (2023). 'Scalable diffusion models with transformers'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205.

Pu, Yunchen et al. (2016). 'A Deep Generative Deconvolutional Image Model'. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. Cadiz, Spain: PMLR, pp. 741–750. URL: `https://proceedings.mlr.press/v51/pu16.html`.

Radford, Alec et al. (2018). 'Improving language understanding by generative pre-training'. In.

Ravi, Nikhila et al. (2024). 'SAM 2: Segment Anything in Images and Videos'. In: *arXiv preprint arXiv:2408.00714*.

Ravuri, Suman et al. (2021). 'Skilful precipitation nowcasting using deep generative models of radar'. In: *Nature* 597.7878, pp. 672–677.

Razavi, Ali, Aaron Van den Oord and Oriol Vinyals (2019). 'Generating diverse high-fidelity images with vq-vae-2'. In: *Advances in neural information processing systems* 32.

Reis, Dillon et al. (2023). 'Real-time flying object detection with YOLOv8'. In: *arXiv preprint arXiv:2305.09972*.

Rombach, Robin et al. (2022). 'High-resolution image synthesis with latent diffusion models'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695.

Ronneberger, Olaf, Philipp Fischer and Thomas Brox (2015). 'U-net: Convolutional networks for biomedical image segmentation'. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Rumelhart, David E, Geoffrey E Hinton and Ronald J Williams (1985). *Learning internal representations by error propagation*. Tech. rep.

Sajjadi, Mehdi SM et al. (2022). 'Object scene representation transformer'. In: *Advances in neural information processing systems* 35, pp. 9512–9524.

Scarselli, Franco et al. (2009). 'The Graph Neural Network Model'. In: *IEEE Transactions on Neural Networks* 20.1, pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

Schmeckpeper, Karl, Georgios Georgakis and Kostas Daniilidis (2021). 'Object-centric video prediction without annotation'. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 13604–13610.

Schuldt, Christian, Ivan Laptev and Barbara Caputo (2004). 'Recognizing human actions: a local SVM approach'. In: *Proceedings of the 17th International Conference on Pattern Recognition*. Vol. 3. IEEE, pp. 32–36.

Shi, Tong et al. (2025). 'Detail-Enhanced Intra-and Inter-modal Interaction for Audio-Visual Emotion Recognition'. In: *International Conference on Pattern Recognition*. Springer, pp. 451–465.

Shi, Xiaoyu et al. (2023). 'Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1599–1610.

Shi, Xiaoyu et al. (2024). 'Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling'. In: *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11.

Shi, Xingjian et al. (2015). 'Convolutional LSTM network: A machine learning approach for precipitation nowcasting'. In: *Advances in neural information processing systems* 28.

Siméoni, Oriane et al. (2025). 'Dinov3'. In: *arXiv preprint arXiv:2508.10104*.

Simonyan, Karen and Andrew Zisserman (2014). 'Very deep convolutional networks for large-scale image recognition'. In: *arXiv preprint arXiv:1409.1556*.

Singh, Gautam, Yi-Fu Wu and Sungjin Ahn (2022). 'Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos'. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al.

Sohl-Dickstein, Jascha et al. (2015). 'Deep unsupervised learning using nonequilibrium thermodynamics'. In: *International conference on machine learning*. pmlr, pp. 2256–2265.

Song, Jiaming, Chenlin Meng and Stefano Ermon (2020). 'Denoising diffusion implicit models'. In: *arXiv preprint arXiv:2010.02502*.

Suleyman, Eliyas, Paul Henderson and Nicolas Pugeault (2025). 'On the Benefits of Instance Decomposition in Video Prediction Models'. In: *arXiv preprint arXiv:2501.10562*.

Sun, Mingzhen et al. (2023). 'Moso: Decomposing motion, scene and object for video prediction'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18727–18737.

Tong, Zhan et al. (2022). 'VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training'. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., pp. 10078–10093.

Touvron, Hugo et al. (2023). 'Llama: Open and efficient foundation language models'. In: *arXiv preprint arXiv:2302.13971*.

Tran, Ngoc-Trung, Tuan-Anh Bui and Ngai-Man Cheung (2018). 'Dist-gan: An improved gan using distance constraints'. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 370–385.

Tumanyan, Narek et al. (2024). 'Dino-tracker: Taming dino for self-supervised point tracking in a single video'. In: *European Conference on Computer Vision*. Springer, pp. 367–385.

Vaswani, Ashish et al. (2017). 'Attention is all you need'. In: *Advances in neural information processing systems* 30.

Villar-Corrales, Angel, Ismail Wahdan and Sven Behnke (2023). 'Object-Centric Video Prediction Via Decoupling of Object Dynamics and Interactions'. In: *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 570–574. DOI: 10.1109/ICIP49359.2023.10222810.

Wang, Limin et al. (2023). 'VideoMAE V2: Scaling Video Masked Autoencoders With Dual Masking'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560.

Wang, Yunbo et al. (2018). 'Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning'. In: *International Conference on Machine Learning*. PMLR, pp. 5123–5132.

Wang, Yunbo et al. (2022). 'PredRNN: A recurrent neural network for spatiotemporal predictive learning'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, Zhou et al. (2004). 'Image quality assessment: from error visibility to structural similarity'. In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612. DOI: 10.1109/TIP.2003.819861.

Williams, Ronald J. and David Zipser (June 1989). 'A learning algorithm for continually running fully recurrent neural networks'. In: *Neural Comput.* 1.2, pp. 270280. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.2.270. URL: https://doi.org/10.1162/neco.1989.1.2.270.

Wu, Jialong et al. (2024). 'ivideogpt: Interactive videogpts are scalable world models'. In: *Advances in Neural Information Processing Systems* 37, pp. 68082–68119.

Wu, Yue, Qiang Wen and Qifeng Chen (2022). 'Optimizing video prediction via video frame interpolation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17814–17823.

Wu, Ziyi et al. (2023). 'SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models'. In: *The Eleventh International Conference on Learning Representations*.

xAI (2025). *Grok 4*. URL: `https://x.ai/`.

Xiao, Yuxi et al. (2024). 'SpatialTracker: Tracking Any 2D Pixels in 3D Space'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20406–20417.

Xiong, Ruibin et al. (2020). 'On layer normalization in the transformer architecture'. In: *International Conference on Machine Learning*. PMLR, pp. 10524–10533.

Yan, Wilson et al. (2021). 'Videogpt: Video generation using vq-vae and transformers'. In: *arXiv preprint arXiv:2104.10157*.

Yang, Jiazhi et al. (2024a). 'Generalized Predictive Model for Autonomous Driving'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14662–14672.

Yang, Lihe et al. (2024b). 'Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data'. In: *CVPR*.

Yang, Lihe et al. (2024c). 'Depth Anything V2'. In: *arXiv preprint arXiv:2406.09414*.

Yi*, Kexin et al. (2020). 'CLEVRER: Collision Events for Video Representation and Reasoning'. In: *International Conference on Learning Representations*.

Yu, Sihyun et al. (2023). 'Video probabilistic diffusion models in projected latent space'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18456–18466.

Zhang, Bowen et al. (2022). 'Segvit: Semantic segmentation with plain vision transformers'. In: *Advances in Neural Information Processing Systems* 35, pp. 4971–4982.

Zhang, Bowen et al. (2024a). 'Segvit v2: Exploring efficient and continual semantic segmentation with plain vision transformers'. In: *International Journal of Computer Vision* 132.4, pp. 1126–1147.

Zhang, Richard et al. (2018). 'The unreasonable effectiveness of deep features as a perceptual metric'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595.

Zhang, Zhicheng et al. (2024b). 'Extdm: Distribution extrapolation diffusion model for video prediction'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19310–19320.

Zhou, Yi et al. (2022). 'Slot-vps: Object-centric representation learning for video panoptic segmentation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3093–3103.

Zhu, Haowei et al. (2022). 'Dual Cross-Attention Learning for Fine-Grained Visual Categorization and Object Re-Identification'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4692–4702.

Zhuang, Lipeng et al. (2024). *Flat'n'Fold: A Diverse Multi-Modal Dataset for Garment Perception and Manipulation*. arXiv: 2409.18297 [cs.RO]. URL: https://arxiv.org/abs/2409.18297.